Računarske mreže

Prevod četvrtog izdanja

Ostali Tanenbaumovi bestseleri

Distribuirani sistemi: principi i modeli
(Distributed Systems: Principles and Paradigms)
Ova nova knjiga, koju je autor napisao zajedno s Maartenom van Steenom, obuhvata principe i modele savremenih distribuiranih sistema. U prvom delu se detaljno govori o principima komuniciranja, procesima, imenovanju, sinhronizovanju, doslednosti i replikovanju, otpornosti na greške i bezbednosti. Autori zatim u drugom delu razmatraju različite modele distribuiranih sistema, kao što su objektno zasnovani sistemi, distribuirani sistemi datoteka, sistemi zasnovani na dokumentima i sistemi zasnovani na koordinaciji. Detaljno se razmatraju i brojni primeri.

Savremeni operativni sistemi, 2. izdanje
(Modem Operating Systems)
U ovom iscrpnom delu detaljno su obrađeni principi savremenih operativnih sistema, a ilustrovani su brojnim primerima iz prakse. U prvih pet poglavlja (posle uvodnog), autor se bavi osnovnim pojmovima: procesima i nitima, kružnim blokadama, upravljanjem memorijom, ulazno-izlaznim operacijama i sistemima datoteka. U narednih šest poglavlja obrađuje složenije teme: multimedijske sisteme, višeprocesorske sisteme, bezbednost itd. Na kraju detaljno analizira dva konkretna operativna sistema: UNIX/Linux i Windows 2000.

Strukturna organizacija računara, 4. izdanje
(Structured Computer Organization)
Ova rado čitana klasična knjiga, u svom četvrtom izdanju, na idealan način vas uvodi u arhitekturu računara. Svaka tema je razumljivo objašnjena, od osnova naviše. Tu je poglavlje o matematičkoj (binarnoj) logici za početnike, a zatim slede poglavlja o arhitekturi mikroprocesora, nivou arhitekture skupa instrukcija, operativnim sistemima, asembleru i arhitekturama računara s paralelnim procesorima.

Operativni sistemi: projektovanje i realizacija, 2. izdanje
(Operating Systems: Design and Implementation)
Ova popularna knjiga, pisana u saradnji sa Albertom S. Woodhullom, jedina je knjiga koja obuhvata i principe operativnih sistema i primenu tih principa na realne sisteme. U njoj su detaljno opisani svi klasični operativni sistemi. Osim toga, principi su pažljivo ilustrovani pomoću MINX-a - UNIX-u sličnog operativnog sistema zasnovanog na POSIX-u, koji je namenjen za PC računare i dostupan svima. Uz knjigu se dobije i CD s potpunim sistemom MINX, zajedno sa izvornim kodom. Listing izvornog koda priložen je u dodatku na kraju knjige i detaljno je objašnjen u tekstu.

Računarske mreže

Prevod četvrtog izdanja

Andrew S. Tanenbaum
Univerzitet Vrije Amsterdam, Hoiandija

Preveo Dejan Smiljanić

Glavni urednik Olga Milanko Aleksandra
Redaktor i koordinator projekta Stojanović Sanja Tasiđ
Tehnički urednik Realizacija Vladimir Končarević Sanja
korica Prelom teksta i obrada Tasić Milica Dečanski
slika Vladimir Končarević
Izdavač Mikro knjiga, Beograd
Direktor Dragan Tanaskoski

Publikum, Beograd

Štampa

Ako imate pitanja ili komentare, ili ako želite da dobijete besplatan katalog, pišite nam ili se javite:

Mikro knjiga P. fah 20-87 11030 Beograd tel: 011/3540-544 pi sma@rai kroknj i ga.co.yu Mikro knjiga Jevrejska bb

78000 Banja Luka tel: 051/220-960 pi sma@mi kroknj i ga.ba Mikro knjiga Maksimirska 13 10000

Zagreb tel: 01/2344-02.3 pi smaSmi kroknj i ga.hr

CiP - Каталогизатда у публикации Народна библиотека Cpdnje, Београд

004 7

ТАНЕНБАУМ, Ендру С,
Računarske mreže : prevod četvrtog izdanja / Andrew S. Tanenbaum ; preveo Dejan Smiljanić. - Beograd : Mikro knjiga, 2005 (Beograd : Pubiikum). - XIX, 859 str. : iiustr. ; 24 cm

Prevod dela: Computer Networks / Tanenbaum, Andrew S. - Tiraž 1000. - O autoru: str.

[860]. - Registar.

ISBN 86-7555-265-3

a) Рачунарске мреже
COBISS SR-ID 123197708

RMT/254/423044M166P0218538K56/ 5 4 3 2 1

Posvećeno Suzani, Barbari i Murvinu, kao i uspomeni na Brama i Sviti n

SADRŽAJ

6 TRANSPORTNI SLOJ 461

8 BEZBEDNOST NA MREŽI 691

PREDGOVOR

Pred vama je prevod četvrtog izdanja ove knjige. Svako dosadašnje izdanje odgovaralo je različitoj fazi korišćenja računarskih mreža. Kada se 1980. godine pojavilo prvo izdanje, mreže su bile zabava akademskih institucija. U vreme drugog izdanja, 1988. godine, mreže su već koristili univerziteti i velike poslovne organizacije. Kada je 1996. objavljeno treće izdanje, računarske mreže - naročito Internet - postale su svakodnevna realnost miliona ljudi. Novost u četvrtom izdanju je brz razvoj mnogih oblika bežičnih mreža.

Način rada s mrežama radikalno se izmenio od trećeg izdanja ove knjige. Sredinom devedesetih postojale su brojne lokalne (LAN) i regionalne (WAN) mreže, zajedno sa svojim skupovima protokola. Do 2003. godine, od lokalnih mreža koje se povezuju kablovima praktično je ostao samo Ethernet, a skoro sve regionalne mreže preselile su se na Internet. Shodno tome, iz ovog izdanja je uklonjena velika količina materijala o starijim mrežama.

Međutim, u međuvremenu se desilo i štošta novo. Najvažniji je snažan razvoj bežičnih mreža, uključujući standard 802.11, bežične lokalne mreže, 2G i 3G mobilne telefonske mreže, Bluetooth, WAP, mreže sa informacionim režimom rada (i-mode) i druge. Zbog toga se u ovom izdanju našlo i mnogo materijala o bežičnim mrežama. Druga tema koja je naglo dobila na značaju jeste bezbednost, tako da joj je posvećeno celo poglavlje.

Iako prvo poglavlje ima istu namenu kao u prethodnom izdanju knjige, njegov sadržaj je redigovan i osavremenjen. Na primer, u njemu je opširnije opisan nastanak i razvoj Interneta, Etherneta i lokalnih mreža, uz iznošenje istorijskih podataka i motiva. Ukratko su opisane i kućne mreže.

Drugo poglavlje je prerađeno. Posle kratkog uvoda u principe razmenjivanja podataka, slede tri veća odeljka o prenošenju podataka (kroz medijume, bežično i satelitski), a za njima tri odeljka s najvažnijim primerima (javni komutirani telefonski sistem, sistem mobilne telefonije i kablovska televizija). Nove teme u ovom poglavlju obuhvataju ADSL, širokopojasni bežični prenos, bežične gradske mreže (MAN) i kablovski pristup Internetu uz korišćenje specifikacije DOCSIS.

U trećem poglavlju uvek su objašnjavani osnovni principi protokola „od tačke do tačke“. Oni su zaista večni - ne menjaju se već decenijama. Shodno tome, brojni protokoli koji su odabrani za primere uglavnom se nisu promenili od trećeg izdanja.

Nasuprot tome, podsloj MAC je proteklih godina pretrpeo brojne promene, pa su one unete u četvrto poglavlje. Odeljak o Ethemetu je proširen i sada obuhvata i giga- bitni Ethernet. Uneti su novi odeljci o bežičnim lokalnim mrežama, širokopojasnim bežičnim mrežama, Bluetoothu, komutiranju u sloju veze podataka, uključujući i MPLS.

I peto poglavlje je osavremenjeno: uklonjen je sav materijal koji se odnosi na ATM, a dodat je materijal koji se odnosi na Internet. Sada je glavna tema kvalitet usluga, pa je tu i rasprava o integrisanim i diferenciranim uslugama. Razmotrene su i bežične mreže, te usmeravanje u ad hoc mrežama. Od ostalih tema obrađeni su NAT i mreže ravnopravnih računara.

U šestom poglavlju i dalje govorimo o transportnom sloju, ali uz određene izmene. Jedna

od novina je primer o programiranju utičnica. Serverski i klijentski programi, napisani na jeziku C, prikazani su na po jednoj strani i objašnjeni. Ako ih preuzmete s Web strane posvećene ovoj knjizi, možete ih prevesti i izvršavati. Uzeti zajedno, oni predstavljaju elementaran udaljeni server datoteka ili Web server, pogodan za ekspe- rimentisanje. Od novih tema u ovom poglavlju, pominjemo pozive udaljenim procedurama, i protokole RTP i T/TCP.

U sedmom poglavlju o sloju aplikacija, obrađen je manji broj tema. Posle kratkog uvoda u sistem imena domena (DNS), u ostatku poglavlja govori se samo o tri teme: elektronskoj pošti, Webu i multimediji. Međutim, svaka tema je obrađena vrlo detaljno. Web sada zauzima 60 stranica, a razmotrene su statične i dinamične Web strane, HTTP, CGI skriptovi, mreže za isporučivanje sadržaja, kolačići i keširanje Weba. Obuhvaćene su i osnovne tehnologije izrade savremenih Web strana: uvod u XML, XSL, XHTML, PHP itd., zajedno s primerima koji se mogu isprobati. Govori se i o bežičnom Webu, s naglaskom na informacioni režim rada (i-mode) i WAP. Materijal o multimediji sada obuhvata MP3, reprodukovanje zvuka tokom preuzimanja, Internet radio i prenos govora protokolom IP.

Bezbednost je danas postala toliko važna da joj je posvećeno celo poglavlje od preko 100 stranica. Opisani su principi uspostavljanja bezbednosti (algoritmi za simetrične i javne ključeve, digitalni potpisi, i sertifikati X.509), kao i primene ovih principa u praksi (provera identiteta, obezbeđenje poruka e-pošte i obezbeđenje Web sadržaja). Ovo poglavlje je istovremeno i sveobuhvatno (proteže se od kvantne krip- tografije do državne cenzure) i detaljno (npr. kako radi SHA-1).

U devetom poglavlju nalazi se spisak literature za dalje čitanje, kao i iscrpan popis preko 350 bibliografskih jedinica korišćenih pri pisanju ove knjige. Više od 200 navedenih radova i knjiga napisano je 2000. godine i kasnije.

Računarske knjige su prepune akronima, pa ni ova nije izuzetak. Dok je budete čitali, nailazićete na: ADSL, AES, AMPS, AODV, ARP, ATM, BGP, CDMA, CDN, CGI, CIDR, DCF, DES, DHCP, DMCA, FDM, FHSS, GPRS, GSM, HDLC, HFC, HTML, HTTP, ICMP, IMAP, ISP, ITU, LAN, LMDS, MAC, MACA, MIME, MPEG, MPLS, MTU, NAP, NAT, NSA, NTSC, OFDM, OSPF, PCF, PCM, PGP, PHP, PKI, POTS, PPP, PSTN, QAM, QPSK, RED, RFC, RPC, RSA, RSVP, RTP, SSL, TCP, TDM, UDP, URL, UTP, VLAN, VPN, VSAT, WAN, WAP, WDMA, WEP, WWW, i XML. Ali, ne brinite. Svaku skraćenicu ćemo detaljno objasniti pre nego što je upotrebimo.

Za instruktore koji žele da ovu knjigu iskoriste za svoj kurs, pripremili smo brojne pomodne materijale:

° Priručnik s rešenim zadacima.

[8] Datoteke sa slikama u različitim formatima.

• Prezentaciju u PowerPointu zasnovanu na ovoj knjizi.

[8] Simulator (pisan na jeziku C) za primere protokola iz 3. poglavlja.

° Web stranu s hipervezama ka mnogim priručnicima, organizacijama, zbirkama često postavljanih pitanja itd.

Priručnik s rešenim zadacima možete preuzeti direktno od izdavačke kude Prentice Hall (ali samo instruktori, ne i studenti). Sav ostali materijal nalazi se na Web strani posvećenoj

ovoj knjizi:

http://www.prenhall.com/tanenbaum

Kada se nađete na njoj, pritisnite sliku knjige.

Mnogo ljudi mi je pomagalo tokom izrade četvrtog izdanja knjige, a moju zahvalnost naročito zaslužuju: Ross Anderson, Elizabeth Belding-Royer, Steve Bellovin, Chatschilc Bisdikian, Kees Bot, Scott Bradner, Jennifer Bray, Pat Cain, Ed Felten, Warwick Ford, Kevin Fu, Ron Fülle, Jim Geier, Mario Gerla, Natalie Giroux, Steve Hanna, Jeff Hayes, Amir Herzberg, Philip Homburg, Philipp Hoschka, David Green, Bart Jacobs, Frans Kaashoek, Steve Kent, Roger Kermode, Robert Kinicki, Shay Kutten, Rob Lanphier, Marcus Leech, Tom Maufer, Brent Miller, Shivakant Mishra, Thomas Nadeau, Shlomo Ovadia, Kaveh Pahlavan, Radia Perlman, Guillaume Piene, Wayne Pleasant, Patrick Powell, Thomas Robertazzi, Medy Sanadidi, Christian Schmutzer, Henning Schulzrinne, Paul Sevinc, Mihail Sichitiu, Bernard Sklar, Ed Skoudis, Bob Strader, George Swallow, George Thiruvathukal, Peter Tomsu, Patrick Verkaik, Dave Vittali, Spyros Voulgaris, Jan-Mark Wams, Ruediger Weis, Bert Wij- nen, Joseph Wilkes, Leendert van Doorn i Maarten van Steen.

Posebnu zahvalnost dugujem Trudy Levine koja je dokazala da i staramajke odlično mogu da rediguju tehnički materijal. Shivakant Mishra je smislio mnoge izazovne zadatke koji su dati na kraju svakog poglavlja. Andy Dornan je predložio dodatnu literaturu navedenu u 9. poglavlju. Jan Looyen je u kritičnim trenucima uvele imao spreman hardver, a Dr F. de Nies je majstorski uspevao da kopira priloge iz drugih dokumenata. Mary Franz, urednica iz Prentice Halla, zatrpala me je štivom za čitanje obimnijim od onoga što sam pročitao u poslednjih sedam godina, a pružala mi je pomoć i na sve moguće druge načine.

Na kraju, dolazimo do najvažnijih osoba: Suzane, Barbare i Marvina. Zahvalan sam Suzani zbog njene ljubavi, strpljenja i izletničkih korpi prepunih đakonija, a Barbari i Marvinu zbog vedrog raspoloženja koje su sve vreme uspevali da održe (osim kada su se žalili na grozne školske udžbenike, što me je često vraćalo pameti). Hvala im svima.

ANDREW S. TANENBAUM

UVOD

Svakim od protekla tri stoleća vladala je po jedna tehnologija. Osamnaesti vek je bio era velikih mehaničkih sistema koji su doveli do Industrijske revolucije. Devetnaestim vekom je vladala parna mašina, a tokom 20. veka dominantna tehnologija je bila prikupljanje, obrada i distribuiranje podataka. Pored dragih dostignuća, bili smo svedoci uspostavljanja globalnog telefonskog sistema, pronalaska radija i televizije, nastanka i nezapamćenog razvoja računarske industrije i lansiranja telekomunikacionih satelita.

Zbog brzog napretka tehnologije, navedene oblasti su se stopile praktično brišući razlike između prikupljanja, prenosa, skladištenja i obrade podataka. Organizacije sa stotinama kancelarija širom prostranog geografskog područja sada normalno očekuju da pritiskom na dugme budu u stanju da utvrde status i najzabačenije svoje filijale. Naša sposobnost prikupljanja, obrade i distribuiranja podataka svakim danom je sve veća, ali još brže rastu zahtevi za još složenijom obradom informacija.

Iako je računarska industrija još uvek mlada u poređenju s dragim industrijama (npr. sa automobilskom industrijom i vazdušnim transportom), računati su zabeležili neverovatan napredak za srazmerno kratko vreme. Tokom prve dve decenije svog postojanja računarski sistemi su bili strogo centralizovani, obično unutar jedinstvene prostorije. Često je ta prostorija imala staklene zidove kroz koje su posetioci mogli da se dive velikom elektronskom čudu. Kompanije srednje veličine ili univerziteti mogli su imati jedan do dva računara, dok su velike institucije imale najviše par desetina. Pomisao da će za dvadesetak godina računari iste snage biti veličine poštanske marke i proizvoditi se u milionskim serijama bila je čista naučna fantastika.

Stapanje računara s komunikacijama imalo je snažan efekat na način organizo- vanja računarskih sistema. Ideja „računskog centra“ - prostorije s velikim računarom u koju korisnici donose svoje podatke na obradu - potpuno je prevaziđena. Stari model po kome je jedan računar zadovoljavao sve potrebe organizacije zamenjen je modelom u kome posao obavlja veći broj zasebnih, ali međusobno povezanih računara. Takvi sistemi su nazvani **računarske mreže** (engl. *computer networks).* Konstrukcija i organizacija takvih mreža tema su ove knjige.

Izrazom „računarska mreža“ označavaćemo skup nezavisnih računara, međusobno povezanih jedinstvenom tehnologijom. Za dva računara se kaže da su povezana ako mogu

međusobno razmenjivati podatke. Sama veza ne mora da bude izvedena bakarnom žicom; mogu se upotrebiti optičko vlakno, mikrotalasi, infracrveno zračenje i komunikacioni sateliti. Kao što ćemo kasnije videti, mreže mogu biti različite veličine i oblika. Iako mnogima može zvučati čudno, ni Internet, ni World Wide Web nisu računarske mreže. Kada pročitate ovu knjigu, biće vam jasno i zašto. Zasad je kratko objašnjenje sledeće: Internet nije jedinstvena mreža već mreža koja povezuje mnoge mreže, a Web je distribuirani sistem koji se izvršava preko Interneta.

U literaturi se često mešaju izrazi računarska mreža i **distribuirani sistem** (engl. *distributed system)*. Ključno je to što u distribuiranom sistemu, skup nezavisnih računara korisnici vide kao jedinstven, koherentan sistem. On obično korisnicima prikazuje jedinstven model ili paradigmu. Za ugradnju tog modela najčešće je odgovoran poseban sloj softvera **(posrednički softver, midlver** - engl. *middleware)* koji direktno komunicira sa operativnim sistemom. Najpoznatiji distribuirani sistem je **World Wide Web** u kome sve liči na dokument (Web stranu).

U računarskoj mreži, pomenuta koherencija, model i softver ne postoje. Korisnici se sreću sa stvarnim računarima, a sistem ne pokušava da ujednači njihovu pojavu i način rada. Ako računari imaju različit hardver i različite operativne sisteme, korisnik će to videti. Ukoliko korisnik želi da izvršava program na udaljenom računani, treba da se prijavi na njega i da na njemu pokrene program.

Distribuirani sistem je u stvari softverski sistem koji se izvršava u mreži, zaklanja je i daje joj visok stepen ujednačenosti. Prema tome, razliku između mreže i distribuiranog sistema treba tražiti pre u softveru (naročito u operativnom sistemu), nego u hardveru.

Pa ipak, ova dva pojma se poprilično preklapaju. Na primer, i distribuirani sistem i računarska mreža prenose datoteke s jednog mesta na drugo. Razlika je samo u tome ko pokreće prenošenje - sistem ili korisnik. Iako u ovoj knjizi pretežno govorimo o mrežama, mnoge teme se tiču i distribuiranih sistema. Više podataka o distribuiranim sistemima potražite kod Tanenbauma i Van Steena (2002).

1.1 UPOTREBA RAČUNARSKIH MREŽA

Pre nego što detaljno pretresemo tehničke detalje, posvetimo malo vremena razlozima zbog kojih su računarske mreže zanimljive, i mogućnostima njihove upotrebe.
U krajnjoj liniji, da nije bilo zainteresovanih za računarske mreže, malo bi ih bilo napravljeno. Počećemo od klasičnog korišćenja mreža u preduzećima i od strane pojedinaca, pa ćemo postepeno preći na pokretne korisnike i kućne mreže.

1.1.1 Poslovne mreže

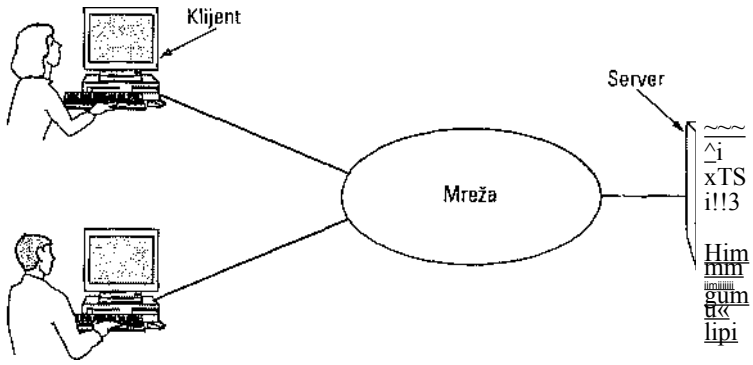
Mnoga preduzeća imaju znatan broj računara. Na primer, preduzeće može da ima posebne računare za praćenje proizvodnje, za inventarisanje i za obračunavanje plata. Na početku je možda svaki od tih računara radio nezavisno od drugih računara, ali je u jednom trenutku uprava odlučila da ih sve poveže kako bi mogla da prikuplja i upo- reduje podatke o čitavom preduzeću.

Ako to malo uopštimo, ovde se radi o **deljenju** resursa (engl. *resource sharing),* čiji je cilj da se svi programi, oprema, a naročito podaci učine dostupnim svima na mreži, bez obzira na stvarnu fizičku lokaciju resursa i korisnika. Očigledan i često ko- rišćen primer je štampač koji deli više službenika. Nijedan službenik nema stvarne potrebe za sopstvenim štampačem, a mrežni štampač visokog kapaciteta često je jeftiniji, brži i lakše se održava od velikog broja pojedinačnih štampača.

Međutim, od deljenja fizičkih resursa - kao što su štampači, skeneri i CD pisači - verovatno je važnije deljenje podataka. Svako veliko i srednje preduzeće, a i mnoga mala, životno zavise od podataka u elektronskom obliku. Većina preduzeća održava spiskove komintenata, inventar, obračune, finansije, poreze i još štošta drugo na sopstvenoj mreži. Kada bi svi njeni računari istovremeno otkazali, banka ne bi preži- vela ni pet minuta. Savremeni proizvodni pogon s kompjuterizovanoin linijom sklapanja proizvoda ne bi izdržao ni toliko. Čak i mala turistička agencija ili advokatslca kancelarija s tri zaposlene osobe danas veoma zavise od računarske mreže koja im omogućava trenutan pristup relevantnim informacijama i dokumentima.

U malim preduzečima svi računari se verovatno nalaze u istoj prostoriji ili možda u istoj zgradi, ali u većim firmama računari i zaposleni mogu biti raštrkani po desetinama kancelarija i pogona u mnogim zemljama. Uprkos tome, prodavcu u Njujorlcu ponekad zatreba pristup bazi podataka s popisom proizvoda u Singapuru. Dragim recima, činjenica da korisnika od podataka deli čak 15.000 kilometara, ne treba da ga spreči da podatke koristi baš kao da su lokalni. Taj cilj se može predstaviti kao pokušaj da se prekine „tiranija geografije".

Informacioni sistem preduzeća u najjednostavnijem slučaju možemo da zamislimo kao jednu ili više baza podataka, i izvestan broj službenika koji pokušavaju da im pristupe daljinski. Po ovom modelu, podaci su uskladišteni na moćnim računarima zvanim serveri (engl. *servers).* Oni su često smešteni najednom mestu, pod budnim okom administratora sistema. Nasuprot tome, službenici na svojim stolovima imaju jednostavnije računare - klijente (engl. *clients*) - pomoću kojih pristupaju udaljenim podacima, da bi ih, na primer, uneli u tabelu na kojoj trenutno rade. (Ponekada ćemo korisnika klijentskog računara nazivati „klijentom", ali će iz konteksta biti jasno da li mislimo na računar ili na korisnika.) Klijentski i serverski računari povezani su preko mreže, kao na slici 1-1. Obratite pažnju na to da smo mrežu predstavili elipsom - bez ikakvih detalja. Koristićemo takvo označavanje kada o mreži govorimo u apstraktnom smislu. Kada bude potrebno više detalja, prikazaćemo i njih.



Slika 1-1. **Mreža s dva klijenta i jednim serverom.**

Opisani sklop se naziva **klijentsko-serverski model** (engl. *client-server model).* On se široko koristi i predstavlja osnovu mnogih mreža. Može se primeniti kada se klijent i server nalaze u istoj zgradi (na primer, pripadaju istom preduzeću), ali i onda kada su međusobno udaljeni. Na primer, kada od kuće pristupite nekoj strani na World Wide Webu, primenjuje se isti model, pri čemu je Web server - server, a vaš računar - klijent. U većini slučajeva isti server može da opsluži veliki broj klijenata.

Ako klijentsko-serverski model ispitamo detaljnije, utvrdićemo da se tu odvijaju dva procesa: jedan na klijentskom, a drugi na serverskom računam. Komunikacija teče tako što klijentski proces preko mreže pošalje poruku serverskom procesu. Klijentski proces zatim čeka odgovor. Kada serverski proces dobije zahtev, on izvršava zahtevani posao ili pronalazi zahtevane podatke i šalje odgovor. Te poruke su prikazane na slici 1-2.

Slika 1-2. **Klijentsko-serverski model obuhvata zahteve i odgovore na njih.**

Dragi motiv za uvođenje računarske mreže ima više veze s ljudima, nego sa informacijama ili čak s računarima. Računarska mreza može da postane moćno **sredstvo komunikacije** (engl. *communication medium)* između zaposlenih. Danas skoro svako preduzeće s dva ili više računara ima **elektronsku** ili **e-poštu** (engl. *electronic mail, e-mail),* koju zaposleni obilato koriste za svakodnevno komuniciranje. U stvari, tokom zajedničke kafe, uglavnom se raspravlja o tome s koliko poruka e-pošte mora svako da se izbori u toku dana, pri čemu su mnoge od njih besmislene jer su šefovi otkrili da jednim pritiskom na dugme mogu da pošalju istu (često praznu) poruku svim svojim nameštenicima.

Međutim, e-pošta nije i jedini oblik poboljšanog komuniciranja koji su omogućile računarske mreže. Pomoću mreže, dve ili više osoba na različitim lokacijama, mogu zajedno da pišu izveštaj. Kada jedna od njih izmeni dokument koji se trenutno nalazi na mreži, svi ostali odmah vide izmenu, umesto da danima čekaju na odštampanu kopiju. Takvo ubrzanje omogućava saradnju između veoma udaljenih grupa, što je ranije bilo neizvodljivo.

Još jedan oblik komuniciranja pomoću računara jesu video-konferencije. Pomoću ove tehnologije, saradnici s više međusobno udaljenih lokacija mogu da drže zajednički sastanak, da vide i čuju jedan dragog, čak i da pišu na zajedničkoj virtuelnoj tabli. Video-konferencije su moćna alatka pomoću koje se eliminišu troškovi i vreme neophodni za putovanja. Ponekad se čuje da se komunikacije i transport međusobno utrkuju: onaj ko pobedi, gurnuće onog drugog u staro gvožđe.

Treći motiv za uvođenje računarskih mreža jeste to što sve veći broj preduzeća obavlja poslovne transakcije s drugim preduzećima elektronskim putem, što naročito važi za isporučioce i korisnike roba. Na primer, proizvođači automobila, aviona i računara nabavljaju podsisteme od brojnih isporučilaca, a zatim sklapaju delove u celinu. Koristeći računarske mreže, proizvođači mogu da naručuju potrebne delove elektronskim putem. Mogućnost da se narudžbina izvrši na vreme (tj., onda kada za određenim de- lom postoji stvarna potreba), smanjuje potrebu za stvaranjem velikih zaliha i povećava efikasnost rada.

Četvrti motiv, koji postaje sve važniji, odnosi se na poslovanje s potrošačima preko Interneta. Avio-kompanije, knjižare i muzičke kuće otkrile su da mnogi ljudi uživaju da kupuju ne ustajući iz fotelje u svom domu. Izlazeći im u susret, brojne kompanije sada drže kataloge svojih proizvoda i usluga na mreži i preko nje primaju narudžbine. Očekuje se dalji brz rast i razvoj ovog sektora i u budućnosti. On se zove elektronska trgovina ili e-trgovina (engl. *electronic commerce, e-commerce).*

1.1.2 Kućne mreže

Ken Olsen je 1977. godine bio predsednik korporacije DEC (Digital Equipment Corporation), tada dragog svetskog prodavca računara (odmaih iza IBM-a). Upitan zašto DEC ne ulazi velikim koracima na tržište personalnih računara, odgovorio je da nema nikakvog razloga da iko ima računar kod kuće. Istorija je demantovala njegove reči i korporacija DEC više ne postoji. Zašto ljudi uopšte kupuju računare za kuću? U početku, to je bilo zbog obrade teksta i igranja, ali se u poslednje vreme ta slika drastično promenila. Danas je verovatno najjači razlog pristup Internetu. Evo nekoliko najpopularnijih razloga za korišćenje Interneta od kuće:

1. Pristupanje udaljenim informacijama.
2. Komuniciranje između korisnika.
3. Interaktivna zabava.
4. Elektronska trgovina.

Pristupanje udaljenim informacijama ispoljava se kroz različite forme. To može da bude lutanje po World Wide Webu u potrazi za određenim informacijama ili samo radi zabave. Raspoložive informacije obuhvataju umetnost, poslove, kuvanje, državnu upravu, zdravlje,

istoriju, hobije, rekreaciju, nauku, sport, putovanja i štošta drugo. Zabava se pojavljuje u tako mnogo oblika da ih je nemoguće sve navesti, a i u oblicima koje radije ne bismo pominjali.

Mnogi dnevni listovi imaju svoja izdanja na mreži koja se mogu personalizovati. tj. podesiti prema ličnim potrebama korisnika. Na primer, ponekad je moguće zahte- vati samo vesti o korumpiranim političarima, požarima, skandalima poznatih ličnosti i epidemijama, ali ne i o fudbalu, na primer. Ponekad je moguće da odabrane članke dobijete direktno na svoj čvrsti disk dok se zasluženo odmarate, ili odštampane na štampaču - neposredno pre prve jutarnje kafe. Ako se ovakav trend nastavi, izazvaće masovno otpuštanje prodavača novina, ali novinske kuće ga podržavaju jer je distribucija izdanja uvek bila najslabija karika u proizvodnom lancu.

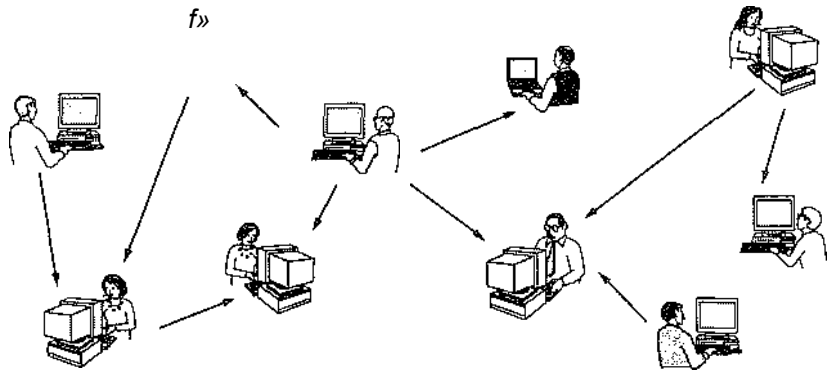
Posle novinskih izdanja (uključujući opšte i naučne časopise), sledeći korak su mrežne digitalne biblioteke. Mnoge profesionalne organizacije, npr. ACM (www.acm.org) i IEEE Computer Society (www.computer.org), već imaju na mreži mnoge časopise i zbornike radova sa održanih skupova. Druge organizacije ih ubrzano slede. Možda će - u zavisnosti od cene, veličine i težine prenosivih računara, takozvanih digitalnih be- ležnica - štampana izdanja otići u zaborav. Oni koji sumnjaju treba da se sete efekta koji je prva štamparska presa imala na srednjevekovnu rukom pisanu literatura.

Sve navedene primene podrazumevaju interakciju između korisnika i udaljene baze podataka. Druga široka kategorija je upotreba mreže za lično međusobno komuniciranje - odgovor 21. stoleća na telefon 19. veka. E-poštu već svakodnevno širom sveta koriste milioni ljudi i njen promet je sve veći. Poruke e-pošte, pored teksta i slika, već uobičajeno sadrže audio i video priloge. Na mirise ćemo još malo pričekati.

Svaki tinejdžer koji drži do sebe upražnjava **trenutno razmenjivanje poruka** (engl. *instant messaging).* Ta mogućnost, izvedena od UNIX-ovog programa *talk* koji se koristio oko 1970. godine, nudi priliku da dvoje ljudi međusobno razmenjuju poruke u realnom vremenu. Višekorisnička verzija ovog koncepta je **pričaonica** (engl. *chat room),* u kojoj svako iz grupe korisnika može da šalje poruke čitavoj grupi.

Tzv. **diskusione grupe** (engl. *newsgroups),* u kojima se raspravlja o svim zami- slivim temama, već su postale uobičajene u nekim sredinama, a sve ih je više i obuhvataju sve šire slojeve korisnika. Rasprave, u kojima jedna osoba istakne poruku koju svi članovi grupe mogu da pročitaju, često su duhovite, ali ponekad i ostrašćene. Za razliku od pričaonica, razmena poruka u diskusionim grupama ne odvija se u realnom vremenu; poruke se snimaju, tako da onoga ko se uključi posle dužeg odsustvovanja čekaju sve poruke koje su u međuvremenu pristigle.

Drugu vrstu međusobne komunikacije predstavlja tzv. komuniciranje između **ravnopravnih korisnika** (engl. *peer-to-peer),* tako nazvano da bi se podvukla razlika u odnosu na klijentsko-serverski model (Parameswaran et al., 2001). Po ovom modelu, korisnici koji čine labavu grupu mogu da komuniciraju s dragim članovima grupe (slika 1-.3). Svaki od njih u načelu može da komunicira s jednim ili više korisnika; nema fiksne podele na klijente i servere.



Slika 1-3. U sistemu ravnopravnih računara nema fiksiranih klijenata i servera.

Komuniciranje između ravnopravnih računara imalo je svoje „zlatno doba“ oko 2000. godine, kada je pomoću usluge Napster preko 50 miliona muzičkih zaljubljenika međusobno razmenjivalo datoteke kršeći autorska prava u do tada nezabeleženom stepenu (Lam iTan, 2001; Macedonia, 2000). Sama ideja je bila prilično jednostavna. Članovi su u centralnoj bazi podataka održavanoj na servera Napster registrovali muzičke numere koje su posedovali na svojim računarima. Ako je neki član želeo određenu numeru, proveravao je bazu podataka da bi utvrdio ko je ima i direktno se obraćao na tu adresu. Vlasnici Napstera su tvrdili da ne krše ničija autorska prava jer na svom serveru nemaju nikavu muziku. Sud, međutim, nije uvažio taj argument i Napster je zatvoren.

U sledećoj generaciji sistema ravnopravnih računara, centralna baza podataka zamenjena je lokalnim bazama koje održava svaki korisnik, a obezbeđenaje i lista okolnih korisnika - članova sistema. Nov korisnik tako može da se obrati bilo kom aktivnom korisniku, da pregleda šta on ima i da od njega dobije listu dragih članova koji će mu pomoći da proširi svoj spisak muzičkih numera i drugih članova. Taj postupale postupnog pretraživanja može da se ponavlja neograničen broj puta, a rezultat je ogromna lokalna baza podataka o muzičkim numerama, smeštena kod korisnika. Takva aktivnost bi za ljude bula mučna, ali računari u njoj briljiraju.

Postoje i legalne primene komuniciranja između ravnopravnih računara. Na primer, korisnici koji razmenjuju muzičke numere u javnom vlasništvu ili muzičke spotove koje su razne grupe objavile radi sticanja publiciteta, zatim porodice koje razmenjuju fotografije, filmove i rodoslovne podatke, kao i tinejdžeri koji učestvuju u višekorisničkim igrama na mreži. U stvari, jedna od najpopularnijih primena Interneta, e-pošta, u suštini koristi sistem ravnopravnih računara. U budućnosti se očekuje znatan razvoj ovakve vrste komuniciranja.

Elektronski kriminal nije ograničen samo na kršenje autorskih prava. Drago privlačno područje je elektronsko kockanje. Računari već decenijama uspešno oponašaju različite stvari, pa zašto ne bi mogli da oponašaju automate za kockanje, rulet, bakaru i druge kockarske igre? Pa, odgovor je da je kocka na mnogim mestima zabranjena.

Problem je u tome što je kockanje na mnogim dragim mestima dozvoljeno (u Engleskoj, na primer), a tamošnji vlasnici kazina dobro su shvatili potencijal kockanja preko Interneta. Sta se događa ako se kockar i kazino nalaze u različitim državama, s različitim zakonima? Dobro pitanje.

Drage primene komuniciranja odnose se na upotrebu Interneta za prenošenje telefonskih razgovora, videofonskih seansi i Internet radija - tri područja koja se brzo razvijaju. Još jedna primena je učenje na daljinu, što znači da možete prisustvovati času koji počinje u 8 sati izjutra, a da prethodno ne morate ustati iz postelje. Upotreba mreža za poboljšanje komunikacije među ljudima može se na duge staze pokazati kao najvažnija od svih primena.

Naša treća kategorija je zabava - ogromna industrija koja neprekidno raste. Ovde je glavna stvar tzv. video na zahtev (engl. *video on demand)*. Za desetak godina ver- ovatno ćete modi da odaberete bilo koji film ili TV emisiju koji su ikada napravljeni u bilo kojoj zemlji i da ih trenutno dobijete na monitora vašeg računara. Novi filmovi će možda biti interaktivni u tom smislu da će se od korisnika povremeno zahtevati da usmeri radnju filma (da li da Magbet ubije Dankana ili da i dalje čeka priliku?), pri čemu će na raspolaganju biti različiti alternativni scenariji. Živi TV prenosi takođe mogu da postanu interaktivni: publika učestvuje u kvizu, bira takmičare itd.

S drage strane, možda video na zahtev neće biti glavna stvar. Možda će to biti ig- rice. Već imamo višekorisničke simulacije u realnom vremenu, kao što su žmurke u virtuelnoj tamnici i simulatore leta u kojima jedan tim pokušava da upuca i obori igrače protivničkog tima. Ako se igrice budu igrale uz digitalne naočare, u tri dimenzije i u realnom vremenu, sa animacijama fotografskog kvaliteta, imaćemo neku vrstu globalne virtuelne stvarnosti.

Naša četvrta kategorija je elektronska trgovina u svom najširem značenju. Kupovina od kuće je već postala popularna; ona omogućuje kupcima da na mreži pregledaju kataloge hiljada firmi. Neki od tih kataloga uskoro će nuditi mogućnost trenutnog uključivanja video sekvence koja prikazuje određeni proizvod, jednostavnim pritiskom na ime tog proizvoda. Kada korisnik kupi proizvod elektronskim putem, pa onda shvati da ne zna kako ga treba koristiti, moći će preko mreže da se obrati odgovarajućoj službi za podršku.

Drugo područje u kome se elektronska trgovina već zahuktala jeste pristup finan- sijskim institucijama. Mnogi ljudi već plaćaju račune elektronskim putem, upravljaju svojim bankovnim računima i prate svoje investicije. Ta primena će se sigurno proširiti kada mreže budu postale bezbednije.

Područje koje izgleda niko nije predvideo jeste elektronska buvlja pijaca (e-bu- vljak?). Mrežne rasprodaje (engl. *on-line auctions)* korišćene robe postale su masovna pojava. Za razliku od klasične e-trgovine, koja sledi klijentsko-serverski model, mrežne rasprodaje se više drže sistema ravnopravnih računara, međusobno povezujući potrošače. Neki od oblika elektronske trgovine dobili su i posebne oznake zahvaljujući činjenici da se engleski predlog „to“ i broj 2 izgovaraju isto. Neki od najpopularnijih navedeni su na slici 1-4.

Oznaka	Puno ime	Primer
B2C	Business-to-consumer (između proizvođača i potrošača)	Naručivanje knjiga preko mreže
B2B	Bussines-to-bussiness (između više proizvođača)	Naručivanje delova za sklapanje automobila od kooperanata
G2C	Government-to-consumer (između državne uprave i potrošača)	Država distribuira poreske obrasce elektronskim putem
C2C	Consumer-to-consumer (između više potrošača)	Rasprodaja korišćene robe preko mreže
P2P	Peer-to-peer (između korisnika)	Razmena datoteka

Slika 1-4. Neki oblici e-trgovine.

Nema sumnje da će se područje primene računarskih mreža brzo širiti u budućnosti, verovatno na način koji niko ne može sada da predvidi. U krajnjoj liniji, lco je 1990. mogao da predvidi da će tinejdžeri dok se vozikaju gradskim prevozom ukucavati kratke poruke u svoje mobilne telefone i tako u narednih 10 godina stvoriti veliki prihod telefonskim kompanijama? Pa ipak, usluga kratkih (SMS) poruka veoma je profitabilna.

Računarske mreže mogu da postanu izuzetno važne ljudima koji se sticajem okolnosti nalaze u geografski izolovanim područjima, nudeći im pristup istim uslugama koje su na raspolaganju stanovniku metropole. Učenje na daljinu može da ima veliki uticaj na obrazovanje; univerziteti mogu da postanu nacionalnog ili internacionalnog značaja. Telemedicina je tek u povoju (npr. daljinski pregled pacijenta), ali ima veliki potencijalan značaj. Međutim, udarna stvar će možda biti nešto sasvim obično, svakodnevno, npr. da pomoću Web kamere postavljene u unutrašnjosti vašeg frižidera vidite da li treba da kupite mleko dok se budete vraćali s posla.

1.1.3 Pokretni korisnici

Prenosivi računan i lični digitalni asistenti (LDA), jedan su od segmenata računa- ske industrije koji najbrže raste. Mnogi vlasnici takvih računara imaju stone računate u kancelariji, ali žele da ostanu u vezi sa svojom matičnom bazom čak i kada nisu kod kuće ili su na putu. Pošto je kablovsko povezivanje nemoguće iz automobila ili aviona, postoji veliko zanimanje za bežične mreže. U ovom odeljku kratko ćemo se pozabaviti nekim prirnenama bežičnih mreža.

Zašto bi neko uopšte poželeo takvu mrežu? Uobičajeni razlog je pokretna kancelarija. Oni koji stalno putuju često žele mogućnost da pomoću elektronske opreme koju nose sa sobom šalju i primaju elektronske pozive, faksove i e-poštu, da lutaju Webom, pristupaju udaljenim datotekama i prijavljuju se na udaljene računare. Oni žele da imaju takvu mogućnost dok su na zemlji, na vodi ili u vazduhu. Na primer, na skupovima o računarima, organizatori ovih dana često uspostavljaju bežičnu mrežu nad prostorom na kome se skup održava. Svako ko ima prenosivi računat i bežični modem može da ih uključi i da se poveže na Internet, baš kao pomoću računara u kablovskoj mreži.

Slično tome, neki univerziteti su nad svojim područjem uspostavili bežičnu mrežu tako da studenti, udobno zavaljeni ispod nekog drveta, mogu da pregledaju katalog bi- bliotečkih jedinica ili da čitaju svoju e-poštu.

Bežične mreže imaju veliki značaj za teretna vozila, taksi-službu, vozila za snab- devanje i majstore koji sve vreme treba da su u kontaktu sa svojom maticom (kućom). Na primer, iako postoje taksi-službe, u mnogim gradovima taksisti su najčešće privatna lica. U nekim gradovima, u taksiju postoji displej koji vozač može da vidi. Kada mušterija zatraži uslugu, centralni dispečer otkuca njenu trenutnu lokaciju i odredište. Ti podaci se prikazuju na displeju uz zvučni signal. Prvi taksista koji pritisne dugme na displeju, preuzima tu mušteriju.

Bežične mreže su važne i za vojsku. Ako želite da uspešno i pravovremeno izve- dete vojnu akciju bilo gde na zemaljskoj kugli, verovatno se nećete oslanjati na lokalnu telekomunikacionu strukturu. Bolje je da takvu strukturu ponesete sa sobom.

Iako su rad u bežičnoj mreži i bežično umrežavanje prenosivih računara srodni sistemi, ipak nisu identični, kao što se vidi sa slike 1-5. Tu vidimo u čemu se razlikuju **fiksni bežični sistem** (engl. *fixed wireless)* i **mobilni bežični sistem** (engl. *mobile wireless)*. Čak su i prenosivi računari ponekad povezani kablom. Na primer, ako putnik uključi svoj prenosivi računar u telefonski priključak u hotelskoj sobi, on je mobilan korisnik i bez bežične mreže.

Bežični	Mobilni	Primene
Ne	Ne	Stoni računari u kancelarijama
Ne	Da	Prenosivi računar u hotelskoj sobi
Da	Ne	Mreže u starim zgradama, bez instalacija
Da	Da	Pokretna kancelarija; LDA za inventarisanje skladišta

Slika 1-5. **Kombinacije bežičnih mreža i pokretnih računara.**

S druge strane, neki bežično umreženi računari nisu pokretni. Značajan primer je preduzeće smešteno u staroj zgradi bez mrežnih instalacija, koje želi da poveže svoje računare. Instaliranje bežične mreže ne mora da bude išta više od kupovine nešto elektronike, raspakivanja i povezivanja. Takvo rešenje može da bude jeftinije od oži- čavanja zgrade.

Naravno da postoje i prave mobilne, bežične primene - počev od pokretne kancelarije, do službenika koji hoda kroz skladište inventarišući robu. Na mnogim prometnim aerodromima, službenici rentakara koji rade na parkingu za vraćena vozila imaju bežične prenosive računare sa štampačem. Kada vozilo stigne na parking, službenik u računar unosi broj vozila. Taj broj se bežičnim putem prenosi centralnom računaru, odakle se šalju podaci o rentiranju na osnovu kojih se odmah štampa račun.

Kako se bežična tehnologija bude širila, pojavljivaće se i nove primene. Razmotrimo neke mogućnosti. Bežični parking-satovi imaju prednosti i za korisnike i za gradsku upravu. Satovi mogu da prihvataju platne kartice i da im odmah proveravaju stanje bežičnim putem. Kada (plaćeno ili dozvoljeno) vreme parkiranja istekne, sat bi slanjem signala u pravcu kola mogao da proveri da li su još uvek tu i da - ukoliko jesu - obavesti policiju o prekoračenju. Procenjeno je da bi gradske uprave samo u SAD mogle na ovaj način da sakupe dodatnih 10

milijardi dolara (Harte et al., 2000). Osim toga, bolja disciplina parkiranja doprinela bi zaštiti okoline jer bi se vozači koji una- pred znaju da će njihovo ilegalno parkiranje biti otkriveno, inožda ipak odlučili da koriste gradski prevoz.

Svuda se mogu nači automati za hranu, napitke i drage artikle. Međutim, roba ne dolazi u automat sama, već se s vremena na vreme pojavi snabdevač s kamionetom i dopuni ga. Kada bi automati za prodaju jednom dnevno bežičnim putem javljali stanje svojih zaliha, snabdevač bi znao koje automate treba da opsluži i šta da ponese. Takav podatak bi mu umnogome olakšao obilazak. Naravno, takav podatak bi se mogao poslati i putem standardne telefonske linije, ali dodeljivanje fiksnog telefonskog priključka svakom automatu da bi on jednom dnevno poslao izveštaj, bilo bi preskupo.

Još jedno područje u kome bi bežični prenos doneo uštede jeste očitavanje brojila u domaćinstvima. Kada bi strujomer, vodomer, gasomer i dragi „satovi“ koje ljudi imaju po stanovima saopštavali svoje stanje bežičnim putem, ne bi trebalo da postoji osoblje koje obilazi domaćinstva i to radi ručno. Slično tome, bežični javljači požara mogli bi odmah da zovu vatrogasce umesto da se oglašavaju zvučnim signalom (što je besmisleno ako nema nikoga kod kuće). Kada cena radio-uređaja i emitovanja opadne, sve će više rezultata takvih daljinskih merenja biti prenošeno bežičnim putem.

Potpuno drugačije područje primene bežičnih mreža jeste povezivanje mobilnih telefona i LDA u male bežične računare. Prvi takav pokušaj rezultovao je malim LDA uređajem koji je mogao da prikaže uprošćene Web strane na još manjem ekranu. Taj sistem, zvan WAP 1.0 (protokol za bežične aplikacije, engl. *Wireless Application Protocol),* nije uspeo, uglavnom zbog mikroskopskog ekrana, uskog propusnog opsega i loše usluge. Međutim WAP 2.0 obećava bolje uređaje i uslugu.

Oblast u kojoj ovakvi uređaji mogu da dostignu svoju punu moć jeste pokretna ili m-trgovina (engl. *m -commerce, mobile commerce*) (Senn, 2000). Pogonska sila ove aktivnosti je udruživanje proizvođača bežičnih LDA uređaja i mežnih operatera koji pokušavaju da smisle način da iz elektronske trgovine i oni izvuku svoj deo. Oni se nadaju da će bežični LDA uređaji moći da se koriste za obavljanje bankarskih poslova i kupovinu. Jedna ideja je da se bežični LDA iskoristi kao svojevrstan elektronski novčanik preko koga bi se plaćalo u prodavnicama, umesto gotovinom ili kreditnim karticama. Iznos bi se pojavljivao na računu za mobilni telefon. Sa stanovišta prođav- nice, takva šema bi im uštedela troškove obrade kreditnih kartica, što može da iznese i nekoliko procenata. Naravno, ova šema može da ima i negativan efekat za prodavce, pošto kupci preko LDA uređaja mogu da uporede cenu iste robe u dragim prodavnicama i da odu tamo. A da i ne pominjemo da telefonske kompanije mogu LDA uređaje da opreme s čitačem bar-koda tako da kupac u samoj prodavnici može da skenira određeni artikal i da trenutno dobije detaljan izveštaj o tome gde se još on može na-baviti i po kojoj ceni.

Pošto operater mreže zna gde se korisnik nalazi, neke usluge su namerno vezane za lokaciju. Na primer, biće moguće tražiti adresu najbliže knjižare ili ldneskog restorana.
Još jedna mogućnost su mobilne mape, kao i vrlo lokalne vremenske prognoze, u stilu „Kada će prestati da pada Idša u mom dvorištu?“. Nema sumnje da će se pojaviti i mnoge drage primene kako bežični uređaji budu masovnije ulazili u upotrebu.

Značajan princip na koji se oslanja m-trgovina jeste činjenica da su korisnici mobilnih telefona naviknli da plaćaju svaku uslugu (za razliku od korisnika Interneta koji očekuju da je sve besplatno). Kada bi neka Web lokacija objavila da će korisnici ubuduće moći da plaćaju robu i usluge svojim kreditnim karticama uz malu naknadu, korisnici bi podigli veliku galamu. S druge strane, kada bi operater mobilne telefonije omogućio korisnicima da u prodavnicama plaćaju pomoću telefona i za tu uslugu im zaračunao malu naknadu, to bi verovatno prošlo kao normalna stvar. Vreme će pokazati da li je tako.

Nešto dalje u budućnosti nalaze se lične mreže i računari koji se nose kao odevni predmeti. IBM je napravio ručni sat koji radi pod Linuxom (uključujući i grafički sistem XII) i bežično se povezuje na Internet radi slanja i primanja poruka e-pošte (Narayanaswami et al., 2002). U budućnosti će ljudi možda razmenjivati svoje „poset- nice“ samo tako što će međusobno suočiti svoje ručne satove. Bežični računari koji se nose kao deo odeće omogućiće ljudima da ulaze u obezbeđene prostorije na isti način kao što to sada omogućavaju magnetne kartice (možda u kombinaciji sa PIN kodom ili biometrijskim parametrima). Takvi satovi mogu biti u stanju i da prikupljaju informacije relevantne za korisnikovu trenutnu lokaciju (npr. adrese lokalnih restorana). Mogućnosti su beskrajne.

Inteligentne časovnike s radiom upoznali smo još 1946. u stripu DikTrejsi. Ali, šta reći o „inteligentnom prahu“? Istraživači sa univerziteta Berkli spakovali su bežični računar u kockicu veličine 1 mm (Warneke et al., 2001). Potencijalne primene takvih računara obuhvataju praćenje opreme, paketa, čak i malih ptica, glodara i insekata.

1.1.4 Društveni aspekti

Sve veće korišćenje mreža donelo je sa sobom nove društvene, etičke i političke probleme. Pomenimo ukratko samo nekoliko takvih problema jer bi za njihovo detaljno razmatranje bila potrebna čitava knjiga. Popularna aktivnost u mnogim mrežama je korišćenje diskusionih grupa i oglasnih tabli pomoću kojih osobe sličnih intereso- vanja međusobno razmenjuju poruke. Sve dok se interesovanja kreću oko tehničkih tema ili hobija, kao što je baštovanstvo, nema problema.

Problemi nastaju kada se formiraju diskusione grupe na teme koje direktno pogađaju ljude, kao što su politika, religija ili seks. Gledišta izneta u takvim diskusionim grupama mogu duboko da vređaju osećanja nekih ljudi. Štaviše, ona često nisu politički ispravna. Poruke nisu ograničene samo na tekst, jer se preko savremenih mreža mogu preneti i visokokvalitetne fotografije u boji, čak i video-sekvence. Neki učesnici diskusija drže se pravila „živi i pusti drage da žive“, ali drugi smatraju da je isticanje određenog materijala (npr. napada na određene zemlje ili religije, pornografije itd.) neprihvatljivo i da se mora cenzurisati. Različite zemlje imaju različite, često suprotstavljene zakone u ovoj oblasti. Na taj način, rasprava se samo rasplamsava.

Ljudi su tužili operatere, smatrajući da su oni odgovorni za sadržaj na mreži, baš kao što su urednici odgovorni za sadržaj novina i časopisa, ali su neizbežno dobij ali odgovor da mreža radi slično telefonskoj kompaniji ili pošti, i da ne može da ograničava ono što korisnici na nju iznesu. Staviše, kada bi operateri cenzurisali poruke, najverovatnije se na mreži ne bi pojavilo ništa što bi i u najmanjoj meri moglo da ih optuži, pa bi na taj način bilo prekršeno

osnovno pravo svakog građanina na slobodu govora. Nema sumnje da će se ova rasprava nastaviti.

Drugo zanimljivo pitanje odnosi se na prava zaposlenih u odnosu na prava poslodavaca. Mnogi ljudi razmenjuju e-poštu na radnom mestu. Poslodavci često ističu svoje pravo da čitaju i, možda, cenzurišu poruke svojih zaposlenih, uključujući i poruke poslate s kućnih računara nakon završetka radnog vremena. Ne slažu se svi zaposleni sa ovakvom politikom firme.

Možda možemo da prihvatimo da poslodavci imaju određenu moć nad svojim zaposlenima, ali da li to važi i za univerzitete i njihove studente? Za srednje škole i njihove učenike? Godine 1994. Univerzitet Carnegie-Mellon je odlučio da prekine tok dolazećih poruka za više diskusionih grupa koje su raspravljale o seksu, smatrajući da je takav materijal nepodoban za mlađe studente (onih nekoliko ispod 18 godina). Prašini koja se potom digla trebale su godine da se slegne.

Sledeća vruća tema je odnos vlasti i građana. FBI je kod mnogih davalaca Internet usluga instalirao sistem pregledanja svih dolaznih i odlaznih poruka e-pošte u cilju traženja određenih reči koje su mogle ukazivati na nelegalnu aktivnost (Blaže i Bel- lovin, 2000; Sobel, 2001; i Zacks, 2001). Sistem je prvobitno nazvan **Carnivore** (mesožder), ali je zbog lošeg utiska koji je takvo ime ostavilo, promenio ime u mnogo nevinije, DCS1000. Međutim, svrha sistema je i dalje ostala ista: da špijunira milione ljudi u nadi da će pronaći informacije o ilegalnoj aktivnosti. Na nesreću po FBI, 4. amandman Ustava Sjedinjenih Država zabranjuje takva istraživanja bez sudskog naloga. Da li te 54 reči, stavljene na papir u 18. veku, imaju neku težinu i u 21. veku, pitanje je koje će možda zapošljavati sudove sve do 22. veka.

Nema samo vlada monopol na ugrožavanje privatnosti građana - isto se ponaša i privatni sektor. Na primer, male datoteke zvane kolačići (engl. *cookies),* koje čitači Weba đeponuju u računarima korisnika, omogućavaju kompanijama da prate aktivnosti korisnika u kibernetskom prostoru, a mogu da izazovu i „curenje“ poverljivih podataka - kao što su brojevi kreditnih kartica - na Internet (Berghel, 2001).

Pomoću računarskih mreža lako se mogu slati anonimne poruke. U izvesnim slučajevima, takva mogućnost je čak poželjna. Na primer, ona omogućava studentima, vojnicima i građanima da zazvone na uzbunu povodom nelegalnog ponašanja određenih profesora, oficira, nadređenih osoba i političara, bez straha od represalija. S druge strane, u Sjedinjenim Državama i većini drugih demokratskih zemalja, zakon izričito priznaje optuženim osobama pravo da se s tužiteljem suoče i rasprave na sudu. Anonimne optužbe ne mogu se prihvatiti kao dokazni materijal.

Jednom reči, računarske mreže - slično štamparskoj presi pre 500 godina - omogućavaju običnim ljudima da šire svoja gledišta na različitije načine i pred drugačijom publikom nego ranije. Ova novostečena sloboda donosi sa sobom mnoge nerazrešene društvene, političke i moralne probleme.

Uporedo s dobrim uvek ide i ono loše, kako to već u životu biva. Internet omogućava brzo pronalaženje informacija, ali su mnoge od njih polovične, varljive ili potpuno pogrešne. Medicinski savet koji preuzmete sa Interneta mogao je objaviti dobitnik Nobelove nagrade za medicinu, ali i neki večiti student. S računarskim mrežama pojavile su se i nove vrste antisocijalnog i kriminalnog ponašanja. Elektronska neželjena pošta (engl. *spam)* postala je deo svakodnevice nakon što su prikupljeni mi- lioni e-adresa i na CD diskovima prodati marketinškim kvazistručnjacima. Poruke e-pošte sa aktivnim sadržajem (programima ili makroima koji se izvršavaju na korisnikovom računaru) mogu da sadrže viruse koji izazivaju haos.

Mnogi pomenuti problemi bi nestali kada bi se računarska industrija ozbiljno pozabavila bezbednošću. Kada bi sve poruke bile šifrovane i s proverenim identitetom pošiljaoca, bilo bi mnogo manje nerviranja. Bezbednosna tehnologija je dobro razrađena i detaljno ćemo je opisati u 8. poglavlju. Problem je u tome što prodavci hardvera i softvera znaju da obezbeđenje ima svoju cenu, i što kupci ne zahtevaju da im se obezbeđenje ugradi. Osim toga, ne mali broj problema izaziva neispravan softver, zato što proizvođači u programe stalno ugrađuju nove mogućnosti, neprestano povećavajući kod, samim tim unoseći u njega sve više grešaka. Možda bi pomoglo uvođenje takse na nove mogućnosti, ali bi to verovatno otežalo prodaju u nekim sredinama. Bilo bi lepo kada bi se korisnicima plaćala nadoknada za neispravan softver, ali bi takva mera za godinu dana dovela do bankrotstva celokupnu industriju softvera.

1.2 MREŽNI HARDVER

Vreme je da našu pažnju sa primene i društvenih aspekata rada u mreži (zabavniji deo) prebacimo na tehničke aspekte projektovanja mreža (teži deo). Ne postoji opšte- prihvaćen sistem klasifikacije računarskih mreža, ali se ističu dva njihova najvažnija aspekta: tehnologija prenosa podataka i veličina. Posvetićemo se prvo jednom, pa drugom aspektu.

U načelu, postoje dva tipa najčešće korišćenih tehnologija za prenos podataka:

1. Veze za neusmereno (difuzno) emitovanje.
2. Veze od tačke do tačke.

Mreže **s neusmerenim (difuznim)** emitovanjem (engl. *broadcast networks*) imaju jedinstven komunikacioni kanal koji dele svi umreženi računari. Kratice poruke, pone- kada zvane **paketi** (engl. *packets*), koje emituje bilo koji računar, primaju svi ostali umreženi računari. Polje za adresu unutar paketa određuje primaoca (računar kome je paket namenjen). Kada računar primi paket i utvrdi daje namenjen njemu, on ga obrađuje; ako utvrdi daje namenjen nekom drugom računaru, jednostavno ga zanemaruje.

Zamislimo, kao analogiju, nekoga ko je izišao u hodnik iz koga vode vrata u mnoge kancelarije i ko glasno viče: „Milane, dođi. Hoću odmah da te vidim.“ Iako je poruku primilo (čulo) mnogo osoba, samo Milan odgovara na nju (izlazi). Ostali je zanemaruju. Draga analogija je poziv preko aerodromskog razglasa da se svi putnici na letu 644 upute ka izlazu 12.

Sistemi za difuzno emitovanje najčešće imaju mogućnost da pakete usmere na sva odredišta pomoću specijalnog koda u adresnom polju. Kada se paket s takvim kodom emituje u mrežu, prima ga i obrađuje svaki umreženi računar. Opisani režim rada naziva se

neusmereno (difuzno) **emitovanje** (engl. *broadcasting).* Neki takvi sistemi podržavaju i usmeravanje paketa samo na određeni podskup računara, što se ponekad naziva **višesmerno** emitovanje (engl. *multicasting).* Jedna mogućnost je da se u adresnom polju rezerviše jedan bit za označavanje višesmernog emitovanja. Preostalih n-1 bitova adrese mogu da sadrže broj grupe. Svaki računar može da se „uključi" u jednu ili više grupa. Kada se paket pošalje određenoj grupi, on se isporučuje svim računarima uključenim u tu grupu.

Za razliku od prethodnog opisa, mreže „od tačke do tačke" (engl. *point-to-point networks)* sadrže brojne veze između pojedinih parova računara. Da bi od polazišta stigao do odredišta, paket na ovom tipu mreže možda mora da prođe kroz jedan ili više drugih računara. Cesto postoji više putanja različite dužine, tako da je pronalaženje optimalne putanje važna stavka u mrežama tipa „od tačke do tačke". Iako postoje mnogi izuzeci, u načelu se u manjim, geografski lolcalizovanim mrežama koristi difuzno emitovanje, dok veće mreže uglavnom koriste povezivanje od tačke do tačke. Prenos poruka od tačke do tačke (od jednog pošiljaoca do jednog primaoca), često se naziva **jednosmerno emitovanje** (engl. *unicasting).*

Mreže se mogu klasifikovati i po veličini. Na slici 1-6, višeprocesorski sistemi svrstani su prema svojoj fizičkoj veličini. Na vrhu liste su lične mreže (engl. *personal area networks),* namenjene jednoj osobi. Takva je, na primer, bežična mreža koja povezuje računar s mišem, tastaturom i štampačem. I LDAlcoji upravlja slušnim aparatom korisnika ili njegovim pejsmejkerom spada u ovu kategoriju. Izvan kategorije ličnih mreža nalaze se mreže većeg dometa. One se dele na lokalne, gradske i regionalne. Konačno, spoj dve ili više mreža naziva se kombinovana mreža (engl. *internetwork).*

Razdaljina između sistema	Sistemi se nalaze	
1 m	na istom kvadratnom metru	Lična mreža
10 m	u istoj prostoriji	
100 m	u istoj zgradi	* Lokalna mreža
1 km	na istom organizacionom području	
10 km	u istom gradu	Gradska mreža
100 km	u istoj državi	Regionalna mreža
1000 km	na istom kontinentu	
10.000 km	na istoj planeti	Internet
		Primer

Slika 1-6. Klasifikacija povezanih sistema prema veličini.

Globalni Internet je dobr o poznati primer kombinovane mreže. Razdaljina je važno merilo za klasifikovanje mreža jer se za razne razdaljine koriste različite tehnologije. U ovoj knjizi bavićemo se mrežama svih veličina. U nastavku ukratko opisujemo mrežni hardver.

1.2.1 Lokalne mreže

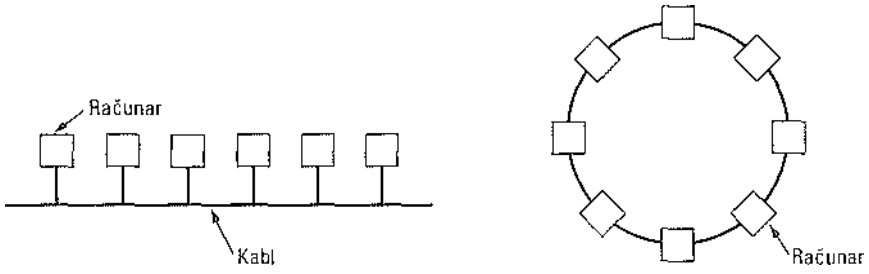
Lokalne mreže (engl. *Local Area Networks, LAN)* jesu privatne mreže unutar jedne zgrade ili jednog organizacionog područja raspona do 5 km. Široko se koriste za povezivanje ličnih računara i radnih stanica u kancelarijama i pogonima firmi radi zajedničkog korišćenja resursa (npr. štampača) i razmene informacija. Lokalne mreže se razlikuju od drugih mreža po tri kriterijuma: (1) veličini; (2) tehnologiji prenosa podataka i (3) topologiji.

Lokalne mreže su ograničene veličine, što znači da je u njima vreme prenosa informacija u najgorem slučaju takođe ograničeno i unapred poznato. Kada poznajemo tu granicu, možemo da upotrebimo način projelctovanja koji inače ne bi bio moguć. Poznavanje ograničenja pojednostavljuje i upravljanje mrežom.

U lokalnim mrežama prenos podataka može se ostvariti pomoću kabla za koji su priključeni svi računali, slično koncepciji telefonske mreže u seoskim područjima. Brzina prenosa u klasičnim lokalnim mrežama kreće se od 10 Mb/s do 100 Mb/s, kašnjenje je malo (meri se milcro ili nano sekundama), a greške retke. Nove lokalne mreže rade brzinama i do 10 Gb/s. U ovoj knjizi držaćemo se klasičnih mreža i izražavati brzinu prenosa u megabitima u sekundi (1 Mb/s je brzina od 1.000.000 bitova u sekundi) i gigabitima u sekundi (1 Gb/s iznosi 1.000.000.000 bitova u sekundi).

Za lokalne mreže s neusmerenim (difuznim) emitovanjem moguće su različite topologije, od kojih su dve prikazane na slici 1-7. U mreži s topologijom magistrale (engl. *bus),* tj. sa linearnim kablom, u jednom trenutku je najviše jedan računar „na vlasti“ i u mogućnosti da emituje. .Svi ostali računari dobijaju zahtev da se uzdrže od slanja poruka. Neophodan je mehanizam odlučivanja za slučaj kada dva ili više računara zahtevaju da istovremeno emituju. Taj mehanizam može da bude centralizovan ili distribuiran. Na primer, sistem DEEE 802..3, popularno zvan Ethernet, predstavlja mrežu s topologijom magistrale, neusmerenim emitovanjem, i decentralizovanim upravljanjem, koja obično radi brzinom između 10 Mb/s i 10 Gb/s. Računari na Ethemetu mogu da emituju poruke kad god požele; ako se dva paketa sukobe, svaki računar pauzira tokom na- sumično izabranog perioda, a onda pokušava ponovo da emituje.

Drugi tip sistema za neusmereno emitovanje jeste topologija prstena (engl. *ring).* U prstenu svaki bit kruži nezavisno od ostatka paketa kome pripada. Cesto bit obiđe ceo prsten pre nego što se emituje čitav paket. Kao u svim sistemima za neusmerano emitovanje, mora postojati neko pravilo za odlučivanje u slučaju istovremenog pristupanja prstenu. U upotrebi su različite metode, npr. omogućavanje računalima da pristupaju redom. IBM-ova token ring mreža IEEE 802.5 predstavlja prstenastu lokalnu mrežu brzine 4 i 16 Mb/s. IFDDI je prstenasta mreža.



(a) (b)

Slika 1-7. Dve mreže sa neusmerenim emitovanjem. (a) magistrala (b) prsten.

Mreže za neusmereno emitovanje mogu se dalje deliti na statičke i dinamičke, u zavisnosti od toga kako se dodeljuje kanal. Pri statičkom dodeljivanju najčešće se vreme izdeli na kratke intervale koji se u krug dodeljuju pojedinim računalima u cilju emitovanja. Statičkim dodeljivanjem kanal se koristi neefikasno jer računar često nema šta da emituje kada na njega dođe red, pa većina sistema kanal dodeljuje dinamički (na zahtev).

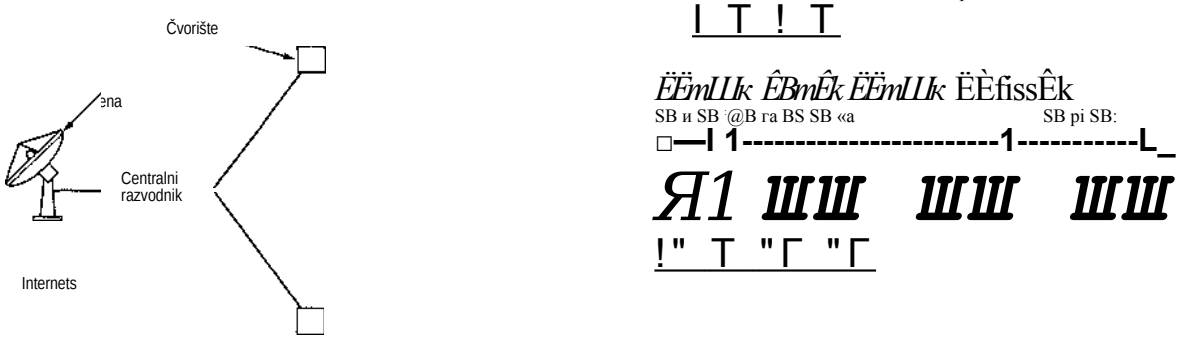
Metode dinamičkog dodeljivanja zajedničkog kanala mogu da budu centralizova- ne i decentralizovane. U metodi centralizovanog dodeljivanja postoji jedinstvena jedinica za odlučivanje koja određuje redosled pristupanja računara magistrali. Ona to čini primajući zahteve i donoseći odluku na osnovu ugrađenog algoritma. U decentra- lizovanom dodeljivanju ne postoji jedinstvena jedinica za odlučivanje; svaki računar mora sam odlučiti da li će da emituje. Možda mislite daje to direktan put do haosa, ali nije tako. Kasnije ćemo proučiti više algoritama namenjenih uvođenju reda u ovaj prividan haos.

1.2.2 Gradske mreže

Gradska mreža (engl. *Metropolitan Area Network, MAN),* kako joj i ime kaže, pokriva gradsko područje. Najpoznatija takva mreža je mreža kablovske televizije, koja postoji u mnogim gradovima. Taj sistem je izrastao iz ranijeg sistema televizije sa zajedničkom antenom u područjima u kojima je postojao loš vazdušni prijem signala. U takvim sistemima, velika zajednička antena postavljana je na vrh obližnjeg uzvišenja, odakle je signal kablovima razvođen po kućama.

Na početku su to bili lokalni ad hoc sistemi, a zatim su kompanije uskočile u posao sklapajući ugovore s gradskim vladama za ožičenje čitavog gradskog područja. Posle toga je došlo programiranje TV kanala, pri čemu su mnogi kanali bili predviđeni isključivo za kablovsku televiziju. Oni su često bili specijalizovani: vesti, sport, ku- vanje, baštovanstvo itd., ali u periodu od njihovog nastanka do kasnih devedesetih godina koristili su se isključivo za prijem TV programa.

Kako je Internet počeo da zaokuplja svetsku javnost, operateri kablovske televizije su shvatili da malim izmenama u sistemu mogu da obezbede i dvosmerne Internet usluge u nekorišćenim delovima frekventnog područja. U tom trenutku, sistem kablovske televizije počeo je da se pretvara iz specijalizovane TV usluge u pravu gradsku mrežu. U svojoj najjednostavnijoj varijanti, MAN mreža može da se prikaže šemom sa slike 1-8. Tu vidimo da se i TV signal i Internet dovode do **centralnog razdvodni- ka** (engl. *head end),* odakle se dalje distribuiraju do kuća korisnika. Ovoj temi ćemo se detaljnije vratiti u 2. poglavlju.



Slika 1-8. **Gradska mreža zasnovana na sistemu kablovske televizije.**

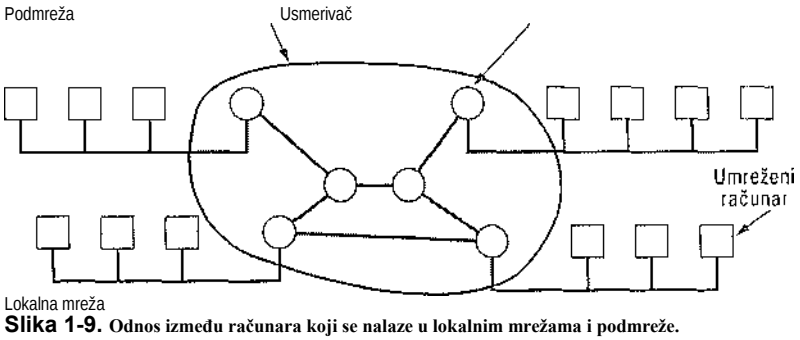
Kablovska televizija nije jedina gradska mreža. Nedavni razvoj visokobrzinskog bežičnog pristupa Internetu rezultovao je drugom vrstom gradske mreže, koja je standardizovana pod oznakom IEEE 802.16. Vratićemo se na nju u 2. poglavlju.

1.2.3 Regionalne mreže

Mreža širokog područja ili **regionalna mreža** (engl. *Wide Area Network, WAN)* pokriva veliko geografsko područje, često čitavu državu ili čak kontinent. Ona sadrži skup računara namenjenih za izvršavanje korisničkih programa (aplikacija). Držaćemo se uobičajenog načina korišćenja ovih mreža sa skupom umreženih računara (engl. *hosts).* Umreženi računari su povezani komunikacionom **podmrežom** (engl. *communication subnet)* ili kratko, podmrežom (engl. *subnet).* Računari su vlasništvo korisnika (to su njihovi lični računari), dok je komunikaciona podmreža najčešće vlasništvo telefonske kompanije ili davaoca Internet usluga; oni je i održavaju. Zadatak podmreže je da prenosi poruke od jednog do dragog računara, kao što telefonski sistem prenosi reci od govornika do slušaoca. Razdvajanje čisto komunikacione uloge mreže (podmreža) od aplikativnog aspekta (računari) umnogome uprošćava projektovanje mreže.

U vec'ini regionalnih mreža, podmreža se sastoji od dve jasno razgraničene komponente: prenosnih linija i prekidačkih elemenata. **Linije prenosa** (engl. *transmission lines*) propuštaju bitove od jednog računaraka drugom. One mogu biti od bakarne žice, optičkog vlakna ili radio-veza. **Prekidački elementi** (engl. *switching elements*) specijalizovani su računari koji spajaju tri i više linija prenosa. Kada podaci stignu jednom linijom, prekidački element mora da odluči kojom linijom da ih dalje uputi. Ovi prekidački računari u prošlosti su različito nazivani, ali se danas za njih ustalio naziv **usmerivači** (engl. *routers)*. Iako naziv usmerivač jasno ukazuje na funkciju ovakvih računara, oni se i dalje u našem računarskom žargonu nazivaju „ruteri".

U opisanom modelu, prikazanom na slici 1-9, često je svaki umreženi računar deo lokalne *mreže* povezane preko usmerivača, iako u izvesnim slučajevima preko usme- rivača može da bude priključen samo jedan računar. Skup linija prenosa i usmerivača (bez umreženih računara) čini podmrežu.



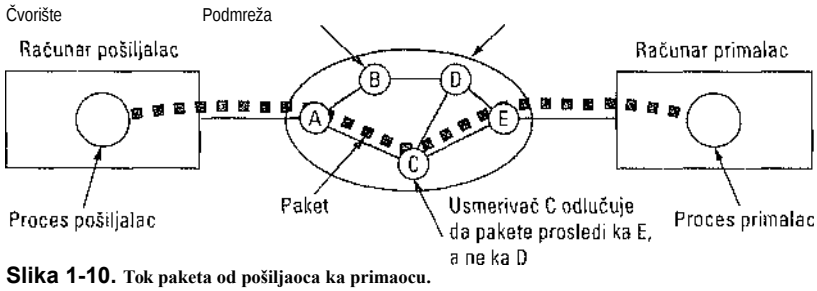
Slika 1-9. Odnos između računara koji se nalaze u lokalnim mrežama i podmreže.

Na ovom mestu treba da prokomentarišemo izraz „podmreža". Prvobitno, podmreža je označavala **isključivo** skup usmerivača i komunikacionih linija za prenošenje paketa od polaznog do odredišnog računara. Nešto kasnije, izraz je dobio i drugo značenje u vezi s mrežnim adresiranjem (o čemu ćemo govoriti u 5. poglavlju). Nažalost, nema šire prihvaćene alternative za prvobitno značenje, tako da ćemo isti izraz oprezno koristiti u oba značenja. Iz konteksta bi trebalo da bude jasno na koje značenje mislimo.

Većina regionalnih mreža ima mnogo linija prenosa, od kojih svaka povezuje dva usmerivača. Ako dva usmerivača koji nisu povezani istom linijom prenosa žele da komuniciraju, moraju to da urade posredno, preko drugih usmerivača. Kada se paket šalje od jednog usmerivača ka dragom preko jednog ili više međuusmerivača, svaki međuusmerivač prima ceo paket, čuva ga dok se ne oslobodi odgovarajuća linija prenosa, a zatim ga prosleđuje dalje. Podmreža koja je organizovana na opisanom principu naziva se podmreža „čuvaj i **prosledi"** (engl. *store-and-forward)* ili podmreža **s komutiranjem paketa** (engl. *packet-switched)*. Skoro sve regionalne mreže (izuzev satelitskih) imaju podmreže s komutiranjem paketa. Kada su paketi mali i iste veličine, često se nazivaju ćelije (engl. *cells).*

Princip komutiranja paketa u regionalnim mrežama toliko je važan da ćemo mu posvetiti još malo vremena. U načelu, kada proces na jednom umreženom računani želi da pošalje

poruku procesu na drugom umreženom računani, računar koji šalje najpre deli poraku na pakete, dodeljujući svakom paketu redni broj. Paketi se tada šalju u mrežu pojedinačno, jedan za drugim. Paketi se nezavisno prenose mrežom i skupljaju u odredišnom računam gde se ponovo od njih sklapa prvobitna poraka i isporučuje procesu kome je namenjena. Tok paketa potekao od jedne poruke prikazan je na slici 1-10.



Slika 1-10. **Tok paketa od pošiljaoca ka primaocu.**

Na ovoj slici, svi paketi slede putanju ACE, umesto ABDE ili ACDE. U nekim mrežama, svi paketi jedne poruke moraju da prate istu putanju; u drugim se svaki paket nezavisno usmerava. Naravno, ako je ACE optimalna putanja, svi paketi mogu proći njom čak i kada se nezavisno usmeravaju.

Odluku o usmeravanju donosi lokalni usmerivač. Kada paket pristigne usmerivaču A, taj usmerivač treba da odluči da li da ga prosledi ka usmerivaču B ili ka usmerivaču C. On tu odluku donosi na osnovu ugrađenog algoritma za usmeravanje (engl. *routing algorithm).* Postoje mnogi takvi algoritmi i o njima ćemo govoriti detaljnije u 5. poglavlju.

Ne rade sve regionalne mreže s komutiranjem paketa. Druga mogućnost je satelitski sistem. Svaki usmerivač je snabdeven primopredajnom antenom. Svi usmerivači mogu da uhvate signal sa satelita, a u nekim slučajevima mogu da čuju i emitovanje okolnih usmerivača ka satelitu. Ponekad su usmerivači uglavnom povezani pod- mrežom od tačke do tačke, s tim što samo neki od njih imaju satelitske antene. Satelitske mreže po svojoj prirodi emituju neusmereno i najkorisnije su kada je za mrežu važno to svojstvo.

1.2.4 Bežične mreže

Digitalno bežično komuniciranje nije nova ideja. Još 1901. godine italijanski fizičar Guljelmo Markoni demonstrirao je princip bežičnog telegrafa između broda i obale koristeći Morzeovu azbuku (u krajnjoj liniji, tačke i crte su binarni kod). Savremeni digitalni bežični sistemi imaju bolje performanse, ali je osnovna ideja ostala ista.

Sasvim grubo, bežične mreže se mogu podeliti u tri osnovne kategorije:

1. Mreže za povezivanje sistema.
2. Bežične lokalne mreže.
3. Bežične regionalne mreže.

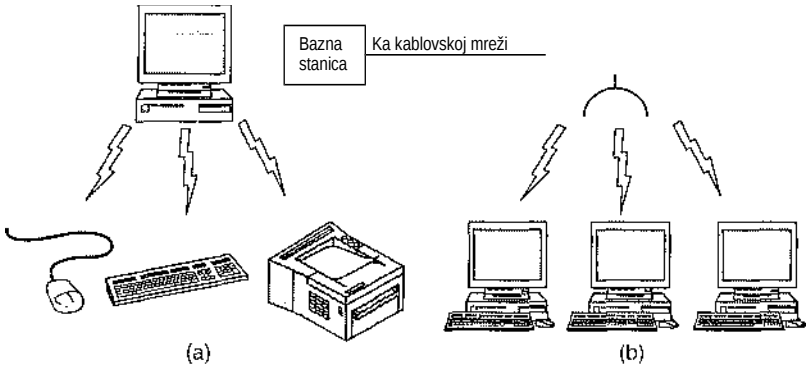
Pod povezivanjem sistema podrazumeva se povezivanje komponenata računara radiotalasima kratkog dometa. Skoro svaki računar ima monitor, tastaturu, miša i štampač povezane s glavnom jedinicom pomoću kablova. Novi korisnici imaju toliko problema da uključe prave kablove u prave priključke (čak i kada su odgovarajući parovi označeni istom

bojom), da većina prodavača nudi da pošalje tehničara da to uradi. Zbog toga su se neke kompanije udružile i projektovale bežičnu mrežu kratkog dometa, zvanu Bluetooth, da bi sve te komponente povezali bez kablova. Sistem Bluetooth omogućava priključivanje i digitalnih kamera, slušalica, skenera i drugih uređaja tako što se jednostavno dovedu u domet emitovanja mreže. Nema kablova, nema instaliranja upravljačkih programa, samo sve skupite na jedno mesto, uključite računar i sve radi! Mnogi smatraju da ih je ova tehnologija preporodila.

U svom najjednostavnijem obliku, mreže za povezivanje sistema koriste obrazac nadređenog i podređenog uređaja, prikazan na slici 1-1 l(a). Sistemska jedinica je obično nadređena i ona upravlja svojim podređenima: mišem, tastaturom itd. Ona im saopštava adrese koje treba da koriste, kada mogu da emituju neusmereno, koliko dugo sme da traje emitovanje, koje frekvencije da koriste itd. O sistemu Bluetooth go- vorićemo detaljnije u 4. poglavlju.

Sledeći korak u bežičnom umrežavanju jesu bežične lokalne mreže. To su sistemi u kojima svaki računar ima radio-modem i antenu pomoću kojih može da komunicira s drugim sistemima. Često na tavanici prostorije postoji antena s kojom računan mogu da komuniciraju, kao na slici 1-1 l(b). Međutim, ako su sistemi međusobno dovoljno blizu, oni mogu komunicirati i direktno između sebe u konfiguraciji ravnopravnih računaia. Bežične lokalne mreže sve su češće u malim kancelarijama i u kućama, gde instaliranje Etherneta predstavlja suviše veliku teškoću, kao i u starijim poslovnim zgradama, konferencijskim salama i na drugim mestima. Za bežične lokalne mreže postoji standard **IEEE 802.11,** koji uglavnom ugrađuju svi sistemi pa se brzo širi. **O** njemu ćemo govoriti u 4. poglavlju.

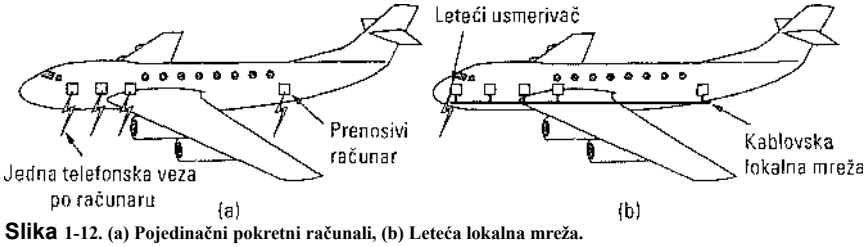
Treća vrsta bežične mreže koristi se u regionalnim mrežama. Radiotalasna mreža koja se koristi za mobilnu telefoniju primer je bežičnog sistema niske propusne moći. Taj sistem je već doživeo tri generacije. Prva generacija je bila analogna i samo za prenos govora. Druga generacija je bila digitalna, ali je i i ona prenosila samo govor. Treća generacija je takođe digitalna, ali prenosi i govor i podatke. Bežične mreže mobilne telefonije u izvesnom smislu liče na bežične lokalne mreže, osim što su razdaljine mnogo veće, a brzine prenosa mnogo manje. Bežične lokalne mreže mogu da rade brzinom do oko 50 Mb/s na udaljenosti od nekoliko desetina metara. Sistemi mobilne telefonije rade brzinom manjom od 1 Mb/s, ali se razdaljina između baze i računara ili telefona meri kilometrima, umesto metrima. Imaćemo štošta da kažemo o ovim mrežama u 2. poglavlju.



Slika 1-11. (a) Konfiguracija sistema Bluetooth, (b) Bežični LAN.

Osim pomenutih sporih mreža, razvijaju se i bežične regionalne mreže visoke propusne modi. Kod njih je glavni cilj da se privatni i poslovni korisnici povežu na Internet pomoću bežičnog priključka visoke brzine prenosa koji zaobilazi sistem tele- fonije. Ta usluga se obično zove lokalna distributivna usluga za više korisnika (engl. *local multipoint distribution Service, LMDS)* i opisacemo je kasnije. Za nju je razvijen i poseban standard, IEEE 802.16, opisan u 4. poglavlju.

Skoro sve bežične mreže se u nekoj tački priključuju na ožičenu mrežu da bi se omogućio pristup datotekama, bazama podataka i Internetu. Takvi priključci se mogu ostvariti na više načina, što zavisi od konkretnih okolnosti. Na primer, na slici l-12(a) prikazujemo avion u kome se više osoba pomoću modema i telefona ugrađenih u sedi- šta povezuju sa svojim kancelarijama. Svaka telefonska veza se uspostavlja pojedinačno. Mnogo efikasnije rešenje je, međutim, „leteća lokalna mreža“, prikazana na slici 1- 12(b). Ovde je svako sedište opremljeno Ethernet priključkom u koji putnici mogu da uključe svoje prenosive računare. Jedan jedini usmerivač u avionu održava vezu s više usmerivača na zemlji koji se smenjuju tokom leta aviona. Takva konfiguracija se ne razlikuje od klasične lokalne mreže, osim što je njena veza sa svetom bežična.



Slika 1-12. (a) Pojedinačni pokretni računali, (b) Leteća lokalna mreža.

Mnogi smatraju da bežičnoj vezi pripada budućnost (npr. Bi i sar,, 2001; Leeper, 2001; Varshey i Vetter, 2000), ali ima i suprotnih mišljenja. Tvorac Etherneta, Bob Metcalfe, napisao je: „Pokretni bežični računari su kao pokretni toaleti - nužno zlo.

Ustaliće se u vozilima, na gradilištima i na rok-koncertima. Savetujem vam da svoju kuću pristojno ožičite i ostanete u njoj“ (Metcalfe, 1995). Istorija će ovaj savet svrstati uz komentar predsednika IBM-a T.J. Watsona, koji je 1945. godine, na pitanje zašto IBM ne ulazi u posao s računalima odgovorio: „Četiri ili pet računara će do 2000. godine biti dovoljno za čitav svet“.

1.2.5 *Kućne* mreže

Kućne računarske mreže su na pomolu. Već se razmišlja o tome da u bliskoj budućnosti većina kuća bude opremljena sopstvenim mrežama. Svi uređaji u kući moći će međusobno da komuniciraju, a svima će se moći pristupiti preko Interneta. Ovo je jedna od onih vizionarskih stvari za kojom nije postojala stvarna potreba (slično daljinskim TV upravljačima i mobilnim telefonima), ali kada se jednom pojavila, svako se čudio kako je bez nje uopšte mogao da živi.

Mnogi uređaji se mogu umrežiti. Navodimo neke od najočiglednijih kategorija (i primere) takvih uređaja:

1. Računari (stoni PC računari, prenosivi PC računati, LDA, deljeni periferijski uređaji).
2. Audio i video oprema (TV, DVD, VCR, kamkorder, kamera, stereo-uređaj, MP3).
3. Telekomunikacije (telefon, mobilni telefon, interfon, faks).
4. Kućni aparati (mikrotalasna rerna, frižider, sat, peć, klima-uređaj, osvetljenje).
5. Telemetrija (električno brojilo, vodomer, protivpožarni i alarmni sistem, termostati, sistem nadgledanja).

Rudimentarno umrežavanje kućnih računara već postoji. U mnogim domovima se više računara posebnim uređajem povezuje s brzim priključkom na Internet. Audio i video oprema još nije umrežena, ali kako se sve više filmova i muzičkih numera preuzima sa Interneta, tako rastu i zahtevi da se na njega povežu stereo-uređaj i i TV prijemnici. Isto tako, mnogi žele da svoje video-snimke podele s rođacima i prijateljima, pa komunikacija mora da bude dvosmerna. Telekomunikaciona oprema je već povezana sa spoljnim svetom, ali će uskoro postati digitalna i ići preko Interneta. U prosečnoj kući verovatno ima desetak satova (računajući i one u raznim uređajima), koji se moraju barem dvaput godišnje podešavati (onda kada se menja vreme sa zimskog na letnje i obrnuto). Kada bi svi satovi bili povezani sa Internetom, mogli bi se podešavati automatski. Na kraju, verovatno je najprivlačnija ideja daljinskog nadgledanja kuće. Mnogi roditelji bi bili voljni da utroše nešto novca kako bi mogli iz restorana da preko LDA uređaja pogledaju šta radi njihova beba, čak i kada je u kući neko ko pazi na nju. Iako možete da razmišljate o posebnim mrežama za svalio od pomenutih područja pri- mene, verovatno je bolja ideja da se svi sistemi povežu u jedinstvenu mrežu.

Kućne mreže imaju neka fundamentalno drugačija svojstva od drugih tipova mreža. Na prvom mestu, mreža i uređaji moraju se lako instalirati. Autor ove knjige imao je priliku da tokom više godina instalira brojne hardverske i softverske komponente na različite računare i susretao se s mnogim problemima. Zahtevajući tehničku podršku od prodavca opreme, dobijao je uglavnom ovakve odgovore: (1) Pročitajte uputstvo, (2) Ponovo pokrenite računar, (3) Uklonite sve hardverske i softverske komponente koje nisu naše i pokušajte ponovo, (4)

Preuzmite najnoviju verziju upravljačkog programa s naše Web lokacije i - ako ništa od pobrojanog ne uspe - (5) Formatirajte čvrsti disk i ponovo instalirajte Windows sa CD-a. Ako kupcu Internet frižidera savetujete da preuzme i instalira novu verziju operativnog sistema frižidera, verovatno ćete regruto- vati četu besnih mušterija. Korisnici računara su navikli da se bore s proizvodima koji ne rade kako bi trebalo; kupci automobila, televizora i frižidera mnogo su manje tolerantni. Oni očekuju da proizvod odmah radi 100% ispravno.

Drugo, upravljanje radom mreže i uređaja mora da bude jednostavno i potpuno pouzdano. Raniji klima-uređaji imali su jedno dugme sa četiri položaja: ISKLJUČENO, SLABO, SREDNJE, JAKO. Danas se uređaji isporučuju s priručnicima od tridesetak stranica. Kada budu umreženi, očekujte da će toliko stranica imati samo poglavlje sa uputstvima o bezbednom rukovanju. Takvo nešto prevazilazi strpljenje većine korisnika.

Treće, za uspeh je neophodno da cena bude niska. Ljudi neće plaćati dodatnih 50 dolara za Internet termostat, zato što ih je malo koji pridaju toliku važnost nadgledanju temperature u svom stanu dok su na poslu. S druge strane, ako treba doplatiti samo 5 dolara, možda će se zainteresovati.

Četvrto, glavna primena će verovatno biti prenos multimedijskih sadržaja, tako da mreža mora imati dovoljnu propusnu moć. Nikada neće postojati tržište za televizore povezane na Internet koji prikazuju trepereće filmove u rezoluciji 320 x 240 piksela, brzinom od 10 slika u sekundi. Brzi Ethernet, koji obavlja glavninu posla u većini kancelarija, nije dovoljno brz za multimediju. Shodno tome, za kućne mreže su potrebne bolje performanse od performansi postojećih kancelarijskih mreža, a cena im mora biti niža da bi postali proizvodi široke potrošnje.

Peto, sistem mora da omogući poČinjanje s jednim do dva povezana uređaja, kao i naknadno postupno širenje mreže. To znači da ne sme doći do sukobljavanja formata. Reći danas mušterijama da kupe periferijske uređaje sa interfejsima po standardu IEEE 1394 (FireWire), a par godina kasnije reklamirati interfejs USB 2.0, vodi u sigurnu propast. Mrežni interfejs ne sme da se menja tokom više godina; ožičenje (ako postoji) mora da ostane isto decenijama.

Šesto, poseban značaj imaće bezbednost i pouzdanost u radu. Izgubiti zbog virusa jednu ili dve poruke e-pošte baš i nije strašno, ali ako provalnik pomoću svog IDA uređaja onesposobi vaš alarmni sistem i isprazni vam kuću, to je nešto sasvim drugo.

Zanimljivo pitanje je da li kućne mreže treba da budu izvedene kablovima ili da budu bežične. U većini kuća već je instalirano šest različitih mreža: električna, telefonska, kablovska TV, vodovodna, gasna i kanalizaciona. Dodati još jednu mrežu tokom građenja novih kuća nije veliki problem, ali je ugradnja sedme mreže u postojeće kuće skupa. Troškovi diktiraju upotrebu bežičnih mreža, dok bezbednost leži na strani ožičenih. Problem s bežičnim prenosom leži u tome što radio-talasi koji se za njega koriste prilično dobro prolaze kroz prepreke. Mnogi se pribojavaju da komšija može neovlašćeno da koristi njihov priključak na Internet ili da špijunira njihovu e-poštu dok se bežičnim putem šalje štampaču. U 8. poglavlju ćemo govoriti o tome kako se bezbednost može poboljšati šifrovanjem, ali u okviru kućnih mreža ona mora da bude stoprocentna, čak i kod neiskusnih korisnika. To je lakše reći nego uraditi, čak i kada korisnici imaju veliko iskustvo.

Sve u svemu, kućno umrežavanje nudi velike mogućnosti, ali se i suočava s mnogim izazovima. Većina njih proističe iz potrebe za lakim rukovanjem, pouzdanošću i bezbednošću takvih sistema, naročito u rukama korisnika koji nemaju tehničkih znanja, kao i iz potrebe da

se po razumnoj ceni dobiju odlične performanse.

1.2.6 Kombinovane mreže

Širom sveta postoje mnoge mreže, sastavljene od različitih hardverskih i softverskih komponenata. Osobe povezane u jednu mrežu često žele da ostvare komunikaciju sa osobama koje su povezane u neku drugu mrežu. Za ispunjenje ove želje potrebno je da se različite, često nekompatibilne mreže međusobno povežu, ponekada pomoću uređaja zvanih mrežni prolazi (engl. *gateways),* koji fizički povezuju i istovremeno usuglašavaju različite hardverske i softverske komponente dve mreže. Skup međusobno povezanih mreža naziva se **kombinovana mreža** ili **međumreža** (engl. *internetwork* ili *internet)*. Ti engleski izrazi se koriste u opštem smislu, za razliku od globalnog Interneta (koji je samo jedna posebna međumreža), koji se najčešće piše s velikim početnim slovom.

Čest oblik međumreže je WAN koji povezuje više LAN-ova. U stvari, ako bismo na slici 1-9 oznaku „podmreža“ zamenili oznakom „WAN“, ništa drugo na slici ne bi trebalo menjati. Jedina stvarna tehnička razlika između podmreže i WAN-a u ovom slučaju jeste pitanje da li u mreži postoje računali. Ako sistem unutar sivog područja sadrži samo usmerivače, to je podmreža; ako sadrži i usmerivače i računare, onda je WAN. Stvarne razlike su vlasništvo i način korišćenja.

Pojmovi podmreže, mreže i međumreže često se mešaju. Podmreža ima najviše smisla unutar regionalne mreže, gde se odnosi na skup usmerivača i komunikacionih linija u vlasništvu operatera mreže. Analogni primer je telefonski sistem koji se sastoji od više centrala međusobno povezanih brzim linijama, a s pojedinačnim korisnicima preko sporih linija. Ove linije i oprema koje poseduje i održava telefonska kompanija, predstavljaju podmrežu telefonskog sistema. Sami telefonski aparati (analogno računalima) nisu deo podmreže. Kombinacija podmreže i telefonskih aparata (tj. računara) obrazuje mrežu. U slučaju lokalne mreže, mrežu obrazuju kabl i računari - podmreža stvarno ne postoji.

Međumreža se obrazuje kada se međusobno povežu jasno ograničene mreže. Smatraćemo da međumreža nastaje kada se povežu LAN i WAN ili kada se povežu dva LAN-a, premda u ovom pogledu postoje mnoga terminološka neslaganja. U slučaju kada više organizacija investira u izgradnju različitih delova mreže i svaka održava svoj deo, iskustveno pravilo kaže da je to međumreža, a ne jedinstvena mreža. Isto tako, ako se u različitim delovima mreže koriste različite tehnologije (npr. difuzno emitovanje i prenos od tačke do tačke), verovatno se ne radi o jednoj, već o više međusobno povezanih mreža.

1.3 MREŽNI SOFTVER

U projektima prvih računarskih mreža hardver je imao glavnu ulogu, a softver sporednu. Takva strategija vise ne prolazi. Struktura današnjeg mrežnog softvera veoma je složena. U narednim odeljcima bavic'emo se detaljnije tehnikama strukturiranja softvera. Metoda koju opisujemo predstavlja kamen temeljac čitave ove knjige i često đemo se na nju vraćati.

1.3.1 Hijerarhije protokola

Da bi projektovanje bilo jednostavnije, mreže se većinom organizuju kao skup **slojeva** (engl. *layers*) ili **nivoa** (engl. *levels).* Broj slojeva, njihova imena, sadržaj i funkcija razlikuju se od mreže do mreže. Svaki sloj nudi određene usluge višim slojevima, ne opterećujući ih detaljima njihove realizacije. Svaki sloj je u izvesnom smislu vir- tuelna mašina koja nudi određene usluge sloju iznad sebe.

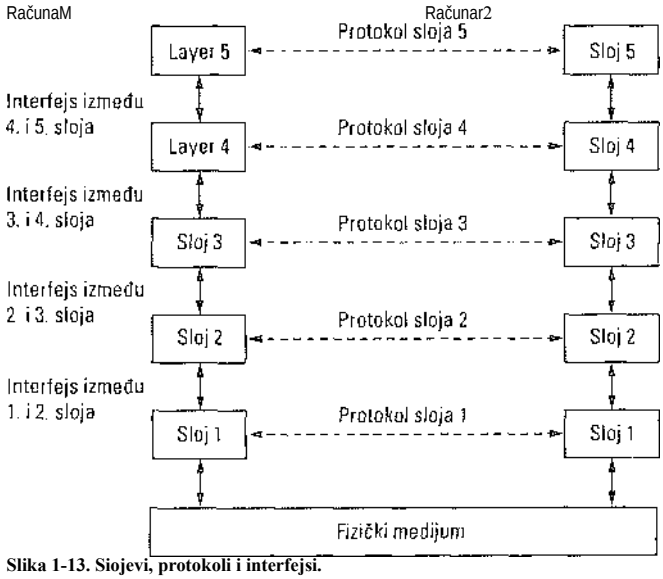
Ovaj koncept je već poznat u računarskim naukama, gde se različito naziva: skrivanje informacija, apstraktni tipovi podataka, kapsuliranje podataka i objektno ori- jentisano programiranje. Osnovna ideja je da određena softverska (ili hardverska) komponenta obezbedi usluge svojim korisnicima, a da od njih sakrije detalje svog unutrašnjeg stanja i primenjenih algoritama.

Sloj *n* najednom računaru komunicira sa slojem *n* na dragom računani. Pravila i konvencije koji se koriste u komuniciranju poznati su pod zajedničkim imenom protokol sloja *n.* U osnovi, **protokol** (engl. *protocol)* predstavlja dogovor između dve jedinke o tome kako treba da teče njihova međusobna komunikacija. Na primer, kada se žena predstavlja muškarcu, ona može da pruži raku. On, potom, može da tu ruku prihvati i stegne ili da je poljubi, u zavisnosti od toga da li je ona američka poslovna žena na nekom profesionalnom skupu ili evropska princeza na zvaničnom balu. Narušavanje protokola otežaće komuniciranje, čak ga i onemogućiti.

Na slici 1-13 prikazana je mreža s pet slojeva. Z,a elemente odgovarajućih slojeva na različitim računalima kaže se da su ravnopravni (engl. *peers).* Ravnopravni elementi mogu da budu procesi, hardverski uređaji, čak i ljudi. Dragim recima, protokolarno komuniciranje se odvija između ravnopravnih strana.

U stvarnosti, nikada se podaci ne prenose direktno od sloja *n* na jednom računam ka sloju *n* na dragom računani, već svaki sloj prosleđuje podatke i upravljačke informacije sloju neposredno ispod sebe, sve dok se ne dostigne najniži sloj. Ispod sloja 1 je **fizički međijum** (engl. *physical medium)* kroz koji se stvarno odvija komunikacija. Tokovi prividne komunikacije označeni su na slici 1-13 tačkastim linijama, a stvarna komunikacija punim.

Između svaka dva susedna sloja nalazi se **interfejs** (engl. *interface).* Interfejs određuje osnovne operacije i usluge koje donji sloj nudi gornjem. Kada projektanti odlučuju o broju slojeva u mreži i njihovoj funkciji, najvažnije je da definišu jasne interfejse između slojeva. Da bi se to postiglo, svaki sloj mora da izvršava određen skup funkcija s tačno definisanom namenom. Osim što smanjuje količinu podataka koja se mora prosleđivati između slojeva, precizno definisan interfejs olakšava i izmenu konstrukcije slojeva (na primer, u slučaju kada se sve telefonske linije zamene satelitskim kanalima), jer se od nove verzije sloja zahteva samo da sloju iznad sebe ponudi isti skup usluga kao i ranije.



Slika 1-13. Siojevi, protokoli i interfejsi.

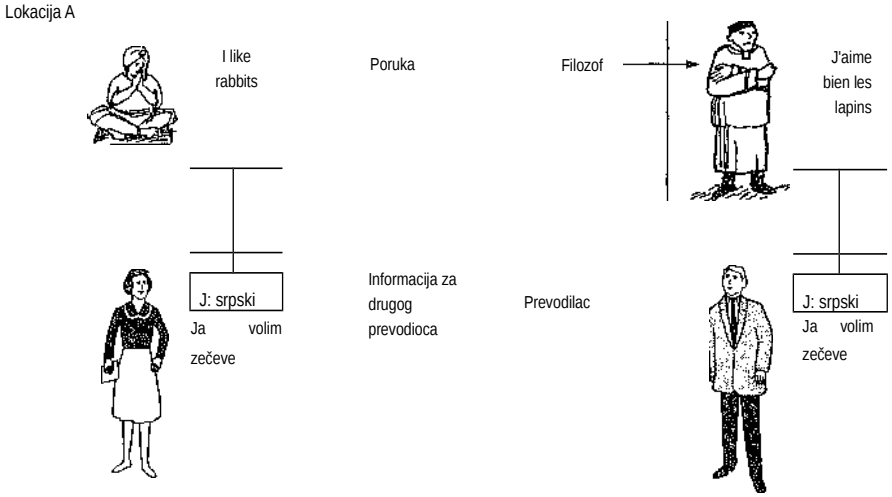
Skup slojeva i protokola naziva se zajedničkim imenom arhitektura mreže (engl. *network architecture)*. Specifikacija arhitekture mora da sadrži dovoljno informacija kako bi realizator mogao da za svaki sloj napiše program ili projektuje hardver koji će slediti pravila odgovarajućeg protokola. Ni detalji realizacije ni specifikacija interfe- jsa nisu deo arhitekture jer se ne vide spolja - oni su skriveni u računalima. Čak nije neophodno da interfejsi na svim umreženim računalima budu isti, pod uslovom da svaki računar ispravno koristi sve protokole. Lista protokola koju koristi određeni sistem (jedan protokol po sloju), naziva se skup protokola (engl. *protocol stack)*. Elementi arhitekture mreže, skupovi protokola i sami protokoli predstavljaju glavne teme ove knjige.

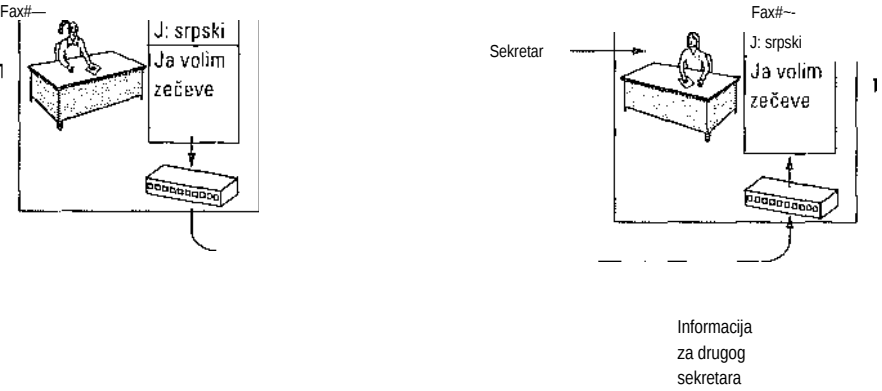
Dademo jednu analogiju koja će vam pomoći da bolje razumete koncept komuniciranja između slojeva. Zamislite dva filozofa (ravnopravni procesi u sloju 3) - jednog koji govori urdu (zvanični jezik Pakistana, prim, prev.) i engleski, i drugog koji govori kineski i francuski. Pošto nijedan od pobrojanih jezika ne poznaju obojica, svaki od njih angažuje prevodioca (ravnopravni procesi u sloju 2), a svaki prevodilac angažuje sekretara (ravnopravni procesi u sloju 1). Prvi filozof želi da kolegi prenese svoja osećanja prema vrsti *oryctolagus cuniculus* (zečevima). U tom cilju, on poruku (na engleskom), „1 like rabbits“, prosleđuje prevodiocu kroz interfejs 2/3, kao što je

prikazano na slici 1-14. Prevodioci su se dogovorili da koriste jezik koji obojica znaju,

Lokacija B

srpski, pa poruka postaje „Ja volim zečeve“. Izbor zajedničkog jezika zavisi od protokola i ravnopravnih procesa u sloju 2.

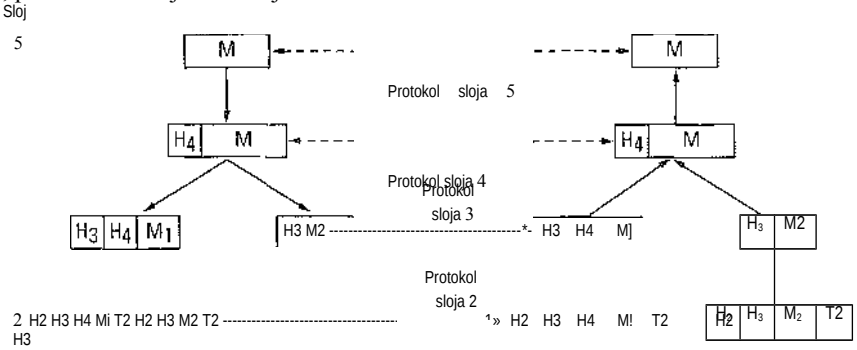




Slika 1-14. **Filozofsko-prevodilačko-sekretarska arhitektura.**

Prevodilac zatim poruku daje sekretara da je pošalje, na primer, faksom (protokol sloja 1). Kada poruka stigne na odredište, ona se pevodi na francuski i kroz interfejs 2/3 prosleđuje drugom filozofu. Obratite pažnju na to da su protokoli potpuno nezavisni jedan od drugog sve dok se interfejsi ne izmene. Prevodioci mogu da sa srpskog jezika pređu na finski, pod uslovom da se oko toga slože, i nijedan od njih neće time izmeniti svoje interfejse pema slojevima 1 i 3. Slično tome, sekretari umesto faksom, poruku mogu da razmene elektronskom poštom ili telefonom, a da time ne poremete druge slojeve; čak ih ne moraju ni obavestiti o tome. Svaki proces može poruci da priključi dodatne podatke namenjene isključivo svom „kolegi“ (ravnopravnom procesu na drugom računaru). Ti podaci se ne prosleđuju gornjem sloju.

Razmotrimo sada jedan „tehničkiji“ primer: kako obezbediti komuniciranje između najviših slojeva petoslojne mreže sa slike 1-15. Poruka M koju generiše proces aplikacije u sloju 5, prenosi se u sloj 4 za slanje.



Slika 1-15. **Primer toka podataka u virtuelnoj komunikaciji između slojeva 5.**

Sloj 4 ispred poruke postavlja identifikaciono zaglavlje (engl. *header)* i sve zajedno prosleđuje sloju 3. Zaglavlje sadrži upravljačke podatke, npr. redne brojeve koji omogućavaju sloju 4 na drugom računam da poruke isporuči ispravnim redosledom, za slučaj da niži slojevi taj redosled ne poštuju. U nekim slojevima, zaglavlja mogu da sadrže i veličine, vremena i druga polja.

U mnogim mrežama nije ograničena veličina poruke koja se može preneti protokolom sloja 4, ali takvo ograničenje skoro uvek postoji u protokolu sloja 3. Zbog toga sloj 3 mor a da izdeli dolazne poruke na manje jedinice - pakete - i da svakom paketu doda zaglavlje sloja 3. U našem primeru, M se deli na dve jedinice, Ml i M2.

Sloj 3 bira liniju za slanje i prosleđuje pakete sloju 2. Sloj 2 svakom paketu dodaje ne samo zaglavlje, već i završetak i sve zajedno prosleđuje sloju 1 za stvarno (fizičko) slanje. Na računam koji poruku primi, ona se kreće uzlazno od jednog do drugog sloja, pri čemu se u svakom sloju s nje uklanja odgovarajuće zaglavlje. Nijedno zaglavlje slojeva ispod sloja *n* ne dolazi do tog sloja.

Na slici 1-15 važno je zapaziti razliku između virtuelne i stvarne komunikacije, kao i razliku između protokola i interfejsa. Ravnopravni procesi u sloju 4, na primer, ponašaju se kao da međusobno komuniciraju „horizontalno“, pomoću protokola sloja

4. Svaki od njih najverovatnije ima neke procedure tipa PošaljiDrugomRačunaru i PreuzmiSaDrugogRačunara, iako te procedure u stvari komuniciraju s nižim slojevima preko interfejsa 3/4, a ne direktno s drugim računarom.

Pojam apstraktnih ravnopravnih procesa igra ključnu ulogu u projektima svih mreža. Pomoću njega se nesavladiv posao projektovanja celokupne mreže može razbiti na više manjih, manipulativnijih projektnih zadataka, tj. na projektovanje pojedinačnih slojeva.

Iako je naslov odeljka 1.3 „Mrežni softver“, treba naglasiti da se niži slojevi hijerarhije protokola često realizuju hardverski ili u obliku softversko-hardverskog upravljačkog sklopa (firmvera). Bez obzira na to što oni delom ili u celini mogu biti realizovani hardverski, za protokole se koriste složeni algoritmi.

1.3.2 Problematika projektovanja slojeva

Neke od glavnih stavki u projelctovanju računarskih mreža zajedničke su za više slojeva. U nastavku ćemo ukratko razmotriti najvažnije.

Svaki sloj mora imati mehanizam za raspoznavanje pošiljalaca i primalaca. Pošto se mreža obično sastoji od više računara na kojima se istovremeno može izvršavati više procesa, mora postojati način da proces na jednom računaru odredi s kim želi da komunicira. Zbog postojanja više mogućih odredišta, neophodan je izvestan oblik njihovog **adresiranja** (engl. *addressing)*.

Drugi skup odluka pri projektovanju odnosi se na pravila prenosa podataka. U nekim sistemima podaci putuju uvek u jednom smeru, dok u drugima putuju u oba sme- ra. Protokolom se mora odrediti broj logičkih kanala po vezi i njihov prioritet. U mnogim mrežama svaka veza ima barem dva logička kanala: jedan za obične, drugi za hitne podatke.

Kontrola grešaka (engl. *error control)* je važna stavka jer su fizički komunikacioni medijumi daleko od savršenstva. Postoje mnogi kodovi za otkrivanje i ispravljanje grešaka, ali se dve strane moraju dogovoriti o kodu koji će koristiti. Osim toga, primalac mora imati mehanizam da saopšti pošiljaocu koje su poruke primljene ispravno, a koje nisu.

Ne održavaju svi komunikacioni kanali redosled poruka koje se kroz njih šalju. Da bi se redosled očuvao, protokolom se mora predvideti mehanizam pomoću koga će primalac ponovo uspostaviti eventualno poremećen redosled poruka. Rešenje koje se nameće jeste numerisanje paketa, ali i dalje ostaje otvoreno pitanje šta raditi s paketima koji su stigli u neispravnom stanju.

Na svakom nivou se javlja problem neusaglašenosti brzine slanja i primanja, pri čemu primalac često bude zatrpan porukama koje ne može da obradi. Predložena su različita rešenja ovog problema i njih ćemo razmotriti kasnije. Neka od njih predviđaju slanje direktne ili indirektne povratne poruke pošiljaocu o trenutnom stanju primaoca. Druga, pak, ograničavaju brzinu slanja na unapred dogovorenu meru. Cela ova problematika naziva se **kontrola toka** (engl*, flow control)*.

Još jedan problem koji je zajednički mnogim slojevima proizlazi iz nemogućnosti svih procesa da prihvate poruke neograničene dužine. Zbog toga postoje mehanizmi rastavljanja, prenošenja i ponovnog sastavljanja poruka. Sličan problem nastaje kada neki proces uporno radi s tako malim paketima podataka da je njihovo pojedinačno prenošenje neefikasno. Ovde je rešenje da se više malih poruka za isto odredište kom- binuje u jedinstvenu veliku poruku, a zatim na odredištu ponovo rastavi na polazne komponente.

Kada je nepogodno ili skupo da se za svaki par procesa koji međusobno komuniciraju uspostavlja zasebna veza, odgovarajući sloj može da istu vezu upotrebi za više istovremenih,

nezavisnih konverzacija. Sve dok je ovo **multipleksiranje** (engl. *multiplexing)* i **demultipleksiranje** (engl. *demultiplexing*) nevidljivo, može ga koristiti svaki sloj. Multipleksiranje je, na primer, neophodno u fizičkom sloju, gde se saobraćaj za sve veze mora preneti preko najviše nekoliko fizičkih linija.

Kada između pošiljaoca i primaoca postoji više putanja, mora se izabrati jedna od njih. Ponekada u odlučivanju o putanji učestvuju dva ili više slojeva. Na primer, da bi se podaci poslali iz Londona u Rim, možda treba doneti odluku na vrhu o tome da li da se šalju preko Francuske ili preko Nemačke, u zavisnosti od odgovarajućih zakonskih propisa. Zatim se na nižem nivou donosi odluka o korišćenju jedne od raspoloživih putanja, u zavisnosti od trenutne gustine saobraćaja. Ova tematika se naziva **usmeravanje** (engl. *routing)*.

1.3.3 Usluge sa uspostavljanjem direktne veze i bez nje

Sloj može sloju iznad sebe da ponudi dve različite vrste usluga: sa uspostavljanjem direktne veze i bez uspostavljanja direktne veze. U ovom odeljku ćemo razmotriti oba tipa usluga i utvrditi razlike između njih.

Usluga sa uspostavljanjem direktne veze (engl. *connection-oriented service)* oblikovana je po modelu telefonskog sistema. Da biste s nekim razgovarali, podižete slušalicu, birate broj, razgovarate, zatim spuštate slušalicu. Slično tome, kada koristite uslugu sa uspostavljanjem direktne veze, najpre uspostavljate vezu, koristite je, a zatim je prekidate. Bitan aspekt veze je da ona liči na cev: pošiljalac najednom njenom kraju ubacuje objekte (bitove), a primalac ih na drugom kraju preuzima. Većinom se pri tome čuva redosled tako da primalac poruke prima redom kojim su poslate.

U nekim slučajevima, pri uspostavljanju veze, pošiljalac, primalac i podmreža **pregovaraju** (engl. *negotiate)* o parametrima koje će koristiti, dogovaraju npr. maksimalnu veličinu poruke, kvalitet usluge i drugo. Najčešće jedna strana daje predlog koji druga strana može da prihvati, da ga odbije ili da ponudi nov predlog.

Nasuprot tome, **usluga bez uspostavljanja direktne veze** (engl. *connectionless service)* modelovana je prema poštanskom sistemu. Svaka poruka (pismo) nosi potpunu adresu odredišta i svaka se na nju nezavisno usmerava. U normalnim situacijama, kada se na isto odredište šalju dve poruke, prva na njega stiže ona koja je prva poslata. Međutim, moguće je da se prva poruka na putu zadrži tako da na odredište najpre stigne druga poruka.

Svaka usluga se može opisati svojim **kvalitetom** (engl. *quality of service)*. Neke usluge su pouzdane u smislu da nikada ne gube podatke. Pouzdana usluga se obično ugrađuje uz zahtev da primalac mora da potvrdi prijem poruke tako da pošiljalac bude siguran daje ona stigla. Proces potvrđivanja unosi dodatni saobraćaj u mrežu i izaziva zastoje, što ima svoje opravdanje, ali je ponekada i nepoželjno.

Najčešća situacija u kojoj je uspostavljanje direktne veze opravdano jeste prenos datoteka. Vlasnik datoteke želi da bude siguran da su svi njeni bitovi stigli u ispravnom stanju i istim redom kako su poslati. Malo je onih koji bi zbog veće brzine pre- nosa prihvatili oštećene datoteke.

Pouzdana usluga sa uspostavljanjem direktne veze postoji u dve varijante: kao tok poruka i kao tok bajtova. U prvoj varijanti se čuvaju granice poruka. Kada se pošalju dve poruke, svaka od 1024 bajta, one na odredište stižu kao dve jasno odeljene poruke od po 1024 bajta, nikada kao jedna poruka od 2048 bajtova. U drugom slučaju, veza predstavlja jednostavan

tok bajtova bez granica poruka. Kada 2048 bajtova stigne primaocu, nema načina da se utvrdi da lije u pitanju jedna poruka od 2048 bajtova, 2 poruke po 1024 bajta ili 2048 poruka od 1 bajta. Ako se stranice knjige šalju mrežom slovoslagaču, onda ih možda treba slati kao zasebne poruke. S druge strane, kada se korisnik prijavljuje na udaljeni server, serveru je potrebno poslati samo niz bajtova. Granice poruka nisu bitne.

Kao što smo već pomenuli, za.neke primene je neprihvatljivo usporavanje prenosa izazvano potvrđivanjem prijema poruka. Jedna takva primena je prenos digitalizova- nog glasa. Korisnici telefona će radije prihvatiti da vezu povremeno remeti šum koji se čuje u pozadini nego da svog sagovornika stalno čuju sa zadrškom. Slično tome, kada se prenose video-konferencije, nekoliko neispravnih piksela neće predstavljati problem, ali stalno zamrzavanje slike dok se greške ne isprave hoće.

Nije za sve aplikacije potrebna direktna veza. Na primer, porast popularnosti elektronske pošte pratio je i porast broja neželjenih poruka. Onaj ko takve poruke šalje verovatno ne želi da uspostavlja i kasnije raskida vezu samo zato da bi poslao jednu poruku, niti muje bitna stoprocentno pouzdana isporuka, naročito ako je takva usluga skuplja. Njemu je potrebno da poruka na cilj stigne s visokom verovatnocom, ali bez garancije. Nepouzdana usluga bez uspostavljanja direktne veze (tj. usluga bez potvrđivanja prijema) često se naziva **usluga datagrama** ili **datagrafska usluga** (engl. *datagram service*), po analogiji s telegrafskim uslugama slanja telegrama gde pošiljalac takođe ne dobija potvrdu o prijemu poruke.

Postoje i situacije u kojima je poželjno ne uspostavljati direktnu vezu da bi se poslala jedna kratka poruka, ali je pouzdanost usluge bitna. U takvim slučajevima može se koristiti **usluga datagrama s potvrdom o prijemu** (engl. *acknowledged datagram service).* To je kao da šaljete preporučenu pošiljku s povratnicom. Kada dobije povratnicu, pošiljalac je apsolutno siguran da pošiljka nije negde zalutala, već da je stigla na ruke primaoca.

Postoji i **usluga odgovaranja na zahteve** (engl. *request-reply service).* Pošiljalac Šalje jedinstven datagram sa zahtevom, a od servera dobija odgovor. Na primer, upit lokalnoj biblioteci s pitanjem gde se govori ujgur spada u tu kategoriju. Usluga odgovaranja na zahteve često se koristi za realizovanje komunikacije po modelu klijent—**server** : klijent ispostavlja zahtev, na koji odgovara server. Na slici 1-16 sabrane su vrste usluga o kojima smo govorili.

	Usluga	Primer
Sa uspostavljanjem direktne veze	Pouzdan tok poruka	Niz stranica
	Pouzdan tok bajtova	Daljinsko prijavljivanje
	Nepouzdana veza	Digitalizovani glas
Bez uspostavljanja direktne veze	Nepouzdan datagram	Neželjena elektronska pošta
	Datagram s potvrdom o prijemu	Preporučena pošiljka
	Zahtev - odgovor	Pretraživanje baze podataka

Slika 1-16. Šest vrsta usluga.

Koncept korišćenja nepouzdanih komunikacija može na prvi pogled da vas začudi. Zašto bi iko više voleo nepouzdanu, nego pouzdanu komunikaciju? Pre svega, pouzdana komunikacija (tj. kada se šalje potvrda o prijemu) možda nije na raspolaganju. Na primer, Ethernet ne nudi pouzdanu komunikaciju. Paketi se često mogu oštetiti pri prenosu. O tom problemu treba da vode računa protokoli višeg nivoa. Zatim, zadrške izazvane slanjem potvrda o prijemu poruka mogu da budu neprihvatljive, naročito za aplikacije koje se izvršavaju u realnom vremenu, npr. pri prenosu multimedijskih sadržaja. Zbog toga, uporedo postoje i pouzdane i nepouzdane komunikacije.

1.3.4 Osnovne operacije za definisanje usluge

Usluga se formalno zadaje skupom osnovnih operacija (engl. *primitives),* preko kojih korisnički proces može da pristupi usluzi. Osnovne operacije nalažu usluzi da izvrši određenu akciju ili da izvesti o akciji koju je izvršio ravnopravan proces. Ako je skup protokola smešten u operativni sistem, kao što je često slučaj, osnovne operacije pedstavljaju normalne pozive sistemu. Ti pozivi izazivaju prekid rada programa i prelazak u režim jezgra koje potoni vraća kontrolu operativnom sistemu da bi mogao da pošalje potrebne pakete.

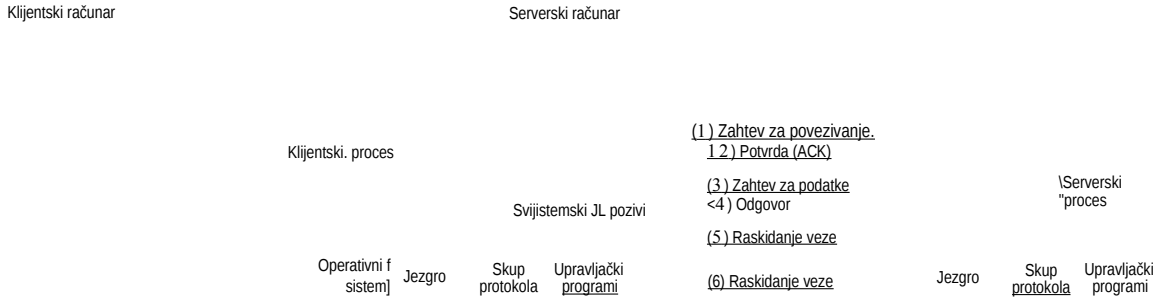
Skup raspoloživih osnovnih operacija zavisi od vrste usluge. Osnovne operacije u uslugama sa uspostavljanjem direktne veze razlikuju se od osnovnih operacija u uslugama bez uspostavljanja veze. Na slici 1-17 prikazanje minimalan skup osnovnih operacija za realizovanje toka bajtova u klijentsko-serverskom okruženju.

Osnovna operacija	**Značenje**
LISTEN	Blokada i čekanje na zahtev za uspostavljanje dolazne veze
CONNECT	Uspostavljanje veze s ravnopravnim procesom koji čeka
RECEIVE	Blokada i čekanje na dolaznu poruku
SEND	Slanje poruke ravnopravnom procesu
DISCONNECT	Prekidanje veze

Slika 1-17. Pet osnovnih operacija za realizovanje jednostavne usluge sa uspostavljanjem direktne veze.

Navedene osnovne operacije mogu se koristiti na sledeći način. Najpre server izvršava operaciju LISTEN, naznačujući da je spreman da prihvati dolazne veze. Operacija LISTEN se obično realizuje kao blokirajući sistemski poziv. Pošto se osnovna operacija izvrši, serverski proces se blokira sve dok ne stigne zahtev za uspostavljanje veze.

Potom klijentski proces izvršava operaciju CONNECT da bi uspostavio vezu sa ser- verom. Poziv CONNECT mora znati na koga treba da se priključi, zato obično sadrži parametar sa adresom servera. Posle toga operativni sistem najčešće šalje paket drugom računaru sa zahtevom za povezivanje, kao što je prikazano procesom (1) na slici 1-18. Klijentski proces se privremeno zaustavlja dok ne stigne odgovor. Kada paket stigne serveru, tamo ga obrađuje operativni sistem. Kada sistem utvrdi daje u pitanju zahtev za povezivanje, on proverava da li server osluškuje. Ako to utvrdi, operativni sistem radi dve stvari: deblokira server i klijentu šalje potvrdan odgovor (2). Kada potvrda stigne do klijenta, ona ponovo pokreće privremeno zaustavljeni klijentski proces. U tom trenutku rade i server i klijent i između njih je uspostavljena veza. Treba naglasiti da se potvrda (2) generiše samim kodom protokola, a ne kao odgovor na upotrebljenu osnovnu operaciju. Ako pristigne zahtev za povezivanje, a server ne osluškuje, rezultat je neodređen. U nekim sistemima paket se nakratko smešta u red čekanja dok ne stigne (ako stigne) signal LISTEN.



Slika 1-18. **Slanje paketa tokom jednostavne komunikacije između klijenta i servera na mreži sa uspostavljanjem direktne veze,**

Možemo da napravimo analogiju između ovog protokola i stvarnog života. To bi bila mušterija (klijent) koja telefonom poziva predstavnika službe za odnose s kupcima nekog preduzeđa. Predstavnik (server) počinje tako što se nalazi u blizini telefona za slučaj da zazvoni. Kada klijent pozove, predstavnik diže slušalicu i veza je uspostavljena.

U sledećem koraku server izvršava operaciju RECEIVE da bi se pripremio za prijem prvog zahteva. Server to obično radi odmah po deblokiranju od poziva LISTEN, pre nego što potvrda signe do klijenta. Poziv RECEIVE ponovo blokira server.

Tada klijent izvršava operaciju SEND da bi poslao svoj zahtev (3), a zatim RECEIVE da bi dobio odgovor.

Pristizanje paketa sa zahtevom deblokira serverski proces tako da može da obradi zahtev.

Pošto to obavi, serverski proces poziva SEND da bi odgovor poslao klijentu (4). Pristizanjem tog paketa deblokira se klijent koji sada može da očekuje odgovor.

Ako klijent ima još zahteva, može sad da ih prosledi. Kada se sve završi, klijent može pozivom DISCONNECT da raskine vezu. Obično je prvi poziv DISCONNECT blokirajući, tako da se klijent privremeno zaustavlja šaljući serveru paket sa obaveštenjem da mu veza više nije potrebna (5). Kada server dobije paket, i on izvršava operaciju DISCONNECT, šaljući potvrdan odgovor klijentu i raskidajući vezu. Kada serverski paket (6) stigne klijentu, klijentski proces se ponovo pokreće, a veza raskida. Ovo je suština načina na koji funlccioniše komunikacija sa uspostavljanjem direktne veze.

Naravno, u stvarnom životu nije sve tako jednostavno. Mnogo štošta može da krene nizbrdo. Sinhronizacija može da bude loša (npr. da se CONNECT izvrši pre LISTEN), paketi mogu na putu nestati, i svašta drugo. Takvim problemima ćemo se pozabaviti kasnije, a za sada smatrajmo da slika 1-18 u osnovnim crtama opisuje komunikaciju između klijenta i servera u mreži sa uspostavljanjem direktne veze.

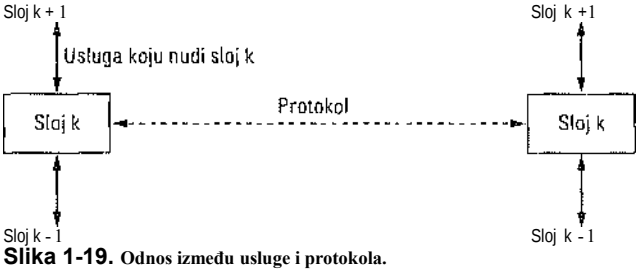
Imajući u vidu da ovaj protokol zahteva razmenjivanje šest paketa, neko može da upita zašto se umesto njega ne koristi neki protokol bez uspostavljanja direktne veze. Kada bi sve radilo savršeno, pitanje bi bilo na mestu jer bi tada bila potrebna samo dva paketa: jedan za zahtev ijedan za odgovor. Međutim, kada se suočimo sa ogromnim porukama (npr. veličine megabajta), greškama u prenosu i izgubljenim paketima, situacija nije tako ružičasta. Ako se odgovor sastoji od stotina paketa, od kojih se neki izgube u transportu, kako će primalac znati da neki paketi nedostaju? Kako će klijent znati da je poslednji paket koji je primio zaista poslednji paket koji je poslat? Pretpostavimo da klijent želi i drugu datoteku. Kako će on razlikovati prvi paket druge datoteke od prvog paketa prve datoteke koji se privremeno izgubio i onda iznenada pronašao put do kuće? Ukratko, u stvarnom svetu, jednostavan protokol odgovaranja na zahtev preko nepouzdane mreže često ne zadovoljava. U 3. poglavlju ćemo detaljno razmotriti više protokola pomoću kojih se prevazilaze pomenuti i drugi problemi. Zasad recimo samo da je pouzdan, uređen tok bajtova između procesa ponekad veoma koristan.

1.3.5 Odnos između usluga i protokola

Usluge i protokoli su različiti pojmovi koji se često mešaju. Pa ipak, razlika među njima je toliko važna, da ćemo je ovde ponovo naglasiti. *Usluga* (engl. *service)* je skup osnovnih operacija koje sloj obezbeđuje sloju iznad sebe. Uslugom se definišu operacije koje sloj izvršava za račun korisnika, ali se potpuno skriva način izvršavanja tih operacija. Usluga se vezuje za interfejs između slojeva, pri čemu je donji sloj davalac, a gornji korisnik usluge.

Protokol je skup pravila o formatu i značenju paketa ili poruka koji se razmenjuju između procesa istog sloja. Procesi koriste protokole da bi realizovali definisane usluge. Oni mogu menjati protokole po želji, pod uslovom da usluge vidljive njihovim korisnicima ostanu neizmenjene. Na ovaj način, usluge i protokoli potpuno su razgraničeni.

Dragim recima, usluge se odnose na interfejse između slojeva, kao što je prikazano na slici 1-19. Nasuprot tome, protokoli se odnose na pakete koji se razmenjuju između ravnopravnih procesa na različitim računarima. Važno je da se ova dva pojma ne mešaju.



Slika 1-19. Odnos između usluge i protokola.

Vredi pomenuti analogiju s programskim jezicima. Usluga je slična apstraktnom tipu podataka ili objektu u objektno orijentisanom jeziku. Ona definiše operacije koje se mogu izvesti sa objektom, ali ne objašnjava kako. Protokol se odnosi na *realizaciju* (implementaciju) usluge i zato nije vidljiv korisniku usluge.

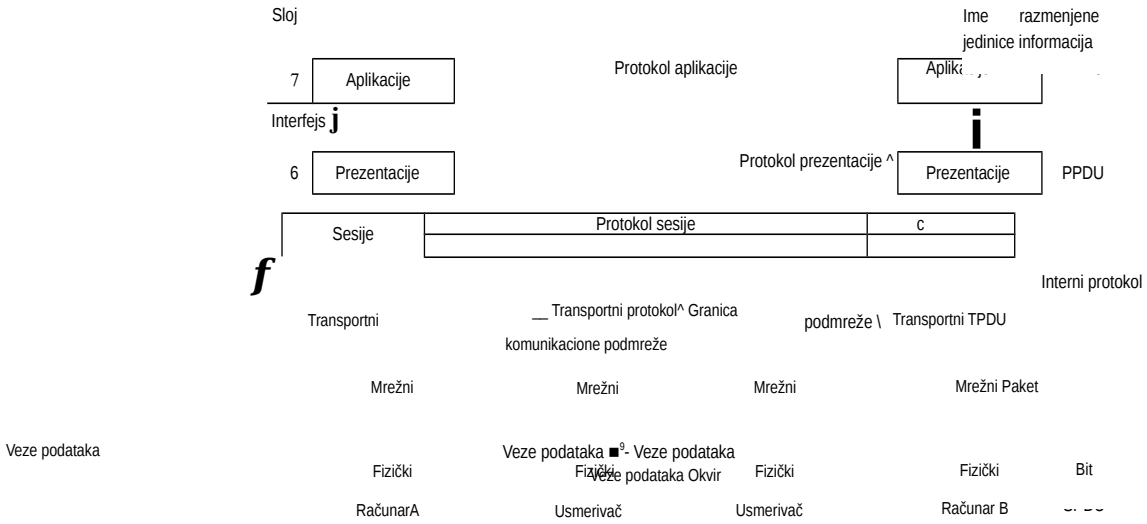
Mnogi stariji protokoli ne prave razliku između protokola i usluge. Na primer, tipičan sloj je inogao imati osnovnu operaciju usluge SEND PACKET, pri čemu je korisnik morao zadavati pokazivač na paket pripremljen za slanje. Uz takav postupale, korisnik je odmah uočavao svaku izmenu protokola. Većina današnjih projektanata mreža to smatra ozbiljnim propustom.

1.4 REFERENTNI MODELI

Pošto smo teorijski opisali slojevite mreže, pređimo na nekoliko primera. U naredna dva odeljka obradićemo dve važne arhitekture mreže - referentne modele OSI i TCP/IP. Iako se protokoli povezani s modelom OSI danas retko koriste, sam model je sveobuhvatan i još uvek važeći, a svojstva svakog sloja i dalje veoma važna. TCP/IP se na neki način nalazi na suprotnom kraju: sam model nema širu primenu, ali se njegovi protokoli nalaze svuda. Zbog navedenih razloga obradićemo detaljno oba modela, jer ne treba zaboraviti da često možete više da naučite na greškama nego na uspesima.

1.4.1 Referentni model OSI

Na slici 1-20 prikazanje model OSI (bez fizičkog medijuma). Model se zasniva na predlogu Međunarodne organizacije za standardizaciju (ISO) i trebalo je da bude prvi korak ka međunarodnom standardizovanju protokola koji se koriste u različitim slojevima (Day i Zimmermann, 1983). Prepravljen je 1995 (Day, 199.5). Model se zove Referentni sistem ISO OSI (engl. *International Standards Organization - Open Systems Interconnection),* zato što treba da poveže otvorene sisteme - one koji su otvoreni za komuniciranje s drugim sistemima. Mi ćemo ga zvati, kratko, model OSI.



9 Protokol mrežnog sloja za povezivanje računara i usmerivača
■ Protokol sloja veze podataka za povezivanje računara i usmerivača
■ Protokol fizičkog sloja za povezivanje računara i usmerivača

Slika 1-20. Referentni model OSI,

Model OSI ima sedam slojeva. Principi koji su doveli do obrazovanja sedam slojeva mogu se sažeti na sledeći način:

1. Treba napraviti nov sloj kad god je neophodna nova apstrakcija.
2. Svaki sloj treba da ima jasno definisanu funkciju.
3. Funkciju svakog sloja treba izabrati imajući u vidu definisanje međunarodno standardizovanih protokola.
4. Granice slojeva treba izabrati tako da se minirnizuje protok informacija između slojeva.
5. Broj slojeva treba da bude dovoljno veliki da se funkcije čije se namene jasno razlikuju ne bi na silu trpale u isti sloj, a ipak dovoljno mali da arhitektura ne postane previše složena.

U nastavku ćemo obraditi svaki sloj redom, počinjući od najnižeg. Obratite pažnju na to da model OSI ne predstavlja arhitekturu mreže jer se njime ne zadaju konkretne usluge i protokoli za svaki sloj. Međutim, organizacija ISO je predvidela i standarde za sve slojeve, premda oni nisu deo modela. Svaki od njih je objavljen kao zaseban međunarodni standard.

Fizički sloj

Uloga **fizičkog sloja** (engl. *physical layer)* jeste da dobijeni niz bitova prenese duž komunikacionog kanala. U probleme njegovog projektovanja spada obezbeđivanje da kada jedna strana pošalje bit 1, druga strana takođe primi bit 1, a ne bit 0. Obično se razmišlja o tome koliki napon treba da predstavlja jedinicu, a koliki nulu, koliko nanosekundi treba da traje bit, da li se prenos može istovremeno obavljati u oba sme- ra, kako se na početku uspostavlja veza i kako se prekida kada oba učesnika obave poslove, koliko kontakata treba da ima mrežni priključak, i za šta se koji kontakt koristi. Projektanti se ovde uglavnom bave mehaničkim, električnim i sinhronizujućim međusklopovima, kao i fizičkim medijumom za prenos, koji leži ispod fizičkog sloja.

Sloj veze podataka

Glavni zadatak **sloja veze podataka** (engl. *data link layer*) jeste da za (gornji) mrežni sloj „pretvori" grabi prenosni uređaj u transportnu liniju koja niz bitova prenosi bez greške. To se radi tako što pošiljalac ulazne podatke deli na **okvire podataka** (engl. *data frames),* najčešće od po nekoliko stotina do nekoliko hiljada bajtova, i okvire šalje jedan za dragim. Ako je usluga pouzdana, primalac potvrđuje ispravan prijem svakog okvira šaljući pošiljaocu **okvir za potvrdu** (engl. *acknowledgement frame).*

Jedan od problema koji se javlja u sloju veze podataka (a i u većini viših slojeva) jeste neusaglašenost brzine slanja i brzine primanja podataka. Često je neophodan nekakav mehanizam regulisanja saobraćaja kako bi pošiljalac znao kolikom privremenom memorijom primalac u svakom momentu raspolaže. Nije retko da se ovo regulisanje toka integriše sa obradom grešaka.

Mreže s difuznim emitovanjem poruka imaju i dodatan problem u sloju veze podataka: kako upravljati pristupom zajedničkom kanalu. Tim problemom se bavi specijalan podsloj sloja veze podataka - podsloj za upravljanje pristupom medijumima (engl. *medium access control sublayer, MAC).*

Mrežni sloj

Mrežni sloj (engl. *network layer)* upravlja radom podmreže. Pri njegovom pro- jektovanju ključno je odrediti kako se paketi upućuju od izvora ka odredištu. Putanje se mogu zasnivati na statičnim tabelama koje su „ugrađene" u mrežu i retko se menja- ju. One se mogu utvrđivati i na početku svake konverzacije, na primer, pre svake ter- minalske sesije (daljinskog prijavljivanja). Najzad, one mogu da se određuju potpuno dinamički za svaki paket, u zavisnosti od trenutnog opterećenja mreže.

Ako se u podmreži istovremeno nalazi previše paketa, oni će se međusobno ometati, stvarajući uska grla. Mrežni sloj treba da kontroliše i takva zagušenja saobraćaja, Rečju, mrežni sloj treba da vodi računa o kvalitetu ponuđene usluge (zadrškama, vremenu prolaska, neravnomernosti pristizanja paketa itd.).

Kada paket, da bi stigao na odredište, treba da pređe s jedne mreže na drugu, mogu da nastanu mnogi problemi. Načini adresiranja u dve mreže mogu da se razlikuju. Draga mreža može i da ne prihvati paket zbog njegove veličine. I protokoli se mogu razlikovati, kao i mnoge druge stvari. Zadatak mrežnog sloja je da prevaziđe navedene probleme i omogući povezivanje heterogenih mreža.

U mrežama s neusmerenim (difuznim) emitovanjem usmeravanje je jednostavno, tako

daje u njima mrežni sloj rudimentaran ili čak ne postoji.

Transportni sloj

Transportni sloj (engl. *transport layer*) ima osnovni zadatak da prihvata podatke „odozgo“, da ih po potrebi razvrstava u manje grupe i da ih prosleđuje mrežnom sloju, obezbeđujući da svi delovi ispravno stignu na odredište. Štaviše, on sve to treba da uradi efikasno i na takav način da od viših slojeva ostanu skrivene neizbežne izmene hardvera.

Transportni sloj takođe defmiše usluge koje se nude sloju sesije - u krajnjoj liniji, korisnicima mreže. Najpopularnija vrsta transportne veze je kanal „od tačke do tačke“ sa ispravljanjem grešaka, koji isporučuje poruke ili tok bajtova redom kojim su poslati. Postoje i druge vrste transportnih usluga, na primer, prenošenje izolovanih poruka bez garancije redosleda pristizanja, kao i difuzno slanje poruka na više odredišta. Vrsta usluge se određuje kada se uspostavi veza. (Pomenimo uzgred da se kanal s potpunim ispravljanjem grešaka u stvarnosti ne može postići; ovim izrazom se podrazumeva da je učestalost pojave grešaka dovoljno niska da se u praksi može tolerisati.)

Transportni sloj potpuno povezuje dva kraja: izvor i odredište. Drugim recima, program na izvornom računani vodi konverzaciju sa sličnim programom na odredišnom računani koristeći pri tome zaglavlja poruka i upravljačke poruke. U nižim slojevima, protokoli povezuju svaki računar s njegovim najbližim susedima, dok između izvornog i odredišnog računara može postojati više usmerivača. Razlika između slojeva 1 do 3, koji su lančano povezani, i slojeva 4 do 7 - koji se direktno protežu od jednog do drugog kraja - prikazana je na slici 1-20.

Sloj sesije

Sloj sesije (engl. *session layer)* omogućava korisnicima na različitim računarima da međusobno uspostave **sesiju** (engl. *session).* Sesije nude različite usluge, uključujući **upravljanje dijalogom** (engl. *dialog control),* tj. vođenje računa o tome na koga je red da šalje poruke, **rad sa žetonima** (engl. *token management),* tj. sprečavanje učesnika da istovremeno pokrenu istu kritičnu operaciju i **sinhronizovanje** (engl. *synchronization),* tj. proveravanje dugačkog niza podataka tokom prenosa da bi se omogućilo nastavljanje od tačke prekida u slučaju pada sistema.

Sloj prezentacije

Za razliku od nižih slojeva, koji uglavnom premeštaju bitove s jednog mesta na drugo, **sloj prezentacije** (engl. *presentation layer)* bavi se sintaksom i semantikom prenetih informacija. Da bi računari koji podatke predstavljaju na različit način mogli međusobno da komuniciraju, strukture podataka koji se prenose mogu se definisati na apstraktan način i standardno kodirati u cilju prenosa. Sloj prezentacije obrađuje te apstraktne strukture podataka i omogućava da se definišu i razmenjuju strukture podataka višeg nivoa (npr. bankarski podaci).

Sloj aplikacija

Sloj aplikacija (engl. *application layer*) sadrži više protokola najčešće potrebnih korisnicima. Jedan takav široko korišćen protokol jeste protokol za prenos hiper- teksta (engl. *Hypertext Transfer Protocol, HTTP),* koji čini osnovu World Wide Weba. Kada korisnik želi da otvori Web stranu u čitaču, on serveru šalje ime te strane koristeći HTTP.

Server tada šalje stranu. Za prenos datoteka, elektronske pošte i poruka diskusionih grupa, koriste se drugi protokoli aplikacija.

1.4.2 Referentni inodel TCP/IP

Pređimo sada s modela OSI na referentni model koji je koristio predale svih regionalnih računarskih mreža, ARPANET, i koji danas koristi njegov naslednik, globalni Internet. Iako ćemo se kasnije ukratko pozabaviti istorijom ARPANET-a, zgodno je da o toj mreži nešto kažemo i sada. ARPANET je bila istraživačka mreža koju je sponzorisalo Ministarstvo odbrane SAD. Ona je povezivala stotine univerziteta i državnih ustanova putem iznajmljenih telefonskih linija. Kada su se kasnije pojavile radio i satelitske mreže, postojeći protokoli su naišli na teškoće, pa je bila neophodna nova referentna arhitektura. Bešavno povezivanje više mreža očigledno je predstavljalo problem od samoga početka. Opisana arhitektura kasnije je postala poznata pod imenom referentni model TCP/IP (engl. *TCP/IP Reference Model),* nastalo kombi- novanjem imena njegova dva osnovna protokola. Prvi su ga definisali Cerf i Kahn (1974), a kasnije razradili Leiner i saradnici (1985). Filozofiju projelctovanja modela naći ćete kod Clarka (1988).

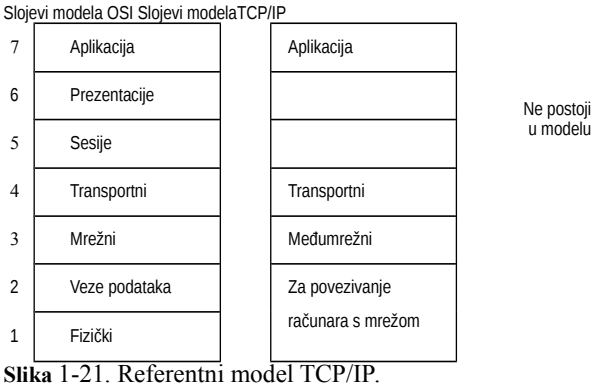
Ministarstvo odbrane je strahovalo da u slučaju napada neki od njegovih drago- cenih računara, usmerivača i međumrežnih prolaza mogu da budu pogođeni ili uništeni, pa je jedan od zadataka projekta ARPANET bio da održi konverzaciju preko mreže i pri eventualnim gubicima podmrežnog hardvera. Drugim recima, Ministarstvo je želelo da se veza može održati sve dok rade izvorni i odredišni računar, čak i ako bi neki od računara između njih bili uništeni. Sem toga, bilo je potrebno osmisliti elastičnu arhitekturu sposobnu da zadovolji različite zahteve, počev od jednostavnog prenosa datoteka, pa do prenosa govora u realnom vremenu.

Međumrežni sloj

Svi pomenuti zahtevi uticali su na to da se izabere mreža s komutiranjem paketa, zasnovana na međumrežnom sloju bez direktnog uspostavljanja veze. Međumrežni sloj (engl. *internet layer)* predstavlja „spajalicu“ koja drži na okupu čitavu arhitekturu mreže. Njegov zadatak je da pakete koje računali ubacuju u bilo koju mrežu upućuje nezavisno na odredište (moguće i na drugu mrežu). Paketi na odredište mogu da stignu redosledom drugačijim od onog kojim su poslati, pa je zadatak viših slojeva da ih dovede u red ako je to neophodno. Međumrežni sloj postoji i na Internetu.

Ovde je pogodno upotrebiti analogiju sa zemaljskom poštom. Određena osoba u jednoj državi može da spusti u poštansko sanduče više pisama za inostranstvo i uz malo sreće da očekuje da sva ona stignu na odredišne adrese. Pisma najčešće putuju od jedne do druge poštanske ustanove u raznim državama, ali pošiljalac to ne vidi. Pošiljalac ne mora da zna ni to da se u svakoj državi (tj. mreži) koriste drugačije poštanske marke, drugačiji format koverata i drugačiji način isporuke.

Međumrežni sloj definiše zvanični format paketa i tzv. **Internet protokol** (engl. *Internet Protocol, IP).* Zadatak međumrežnog sloja je da isporuči IP pakete tamo gde treba da stignu. Jasno je da su ovde najveći problemi usmeravanje i izbegavanje zagušenja. Zbog toga izgleda da se međumrežni TCP/IP sloj može po funkcionalnosti uporediti s mrežnim OSI slojem. Slika 1-21 prikazuje ove sličnosti.

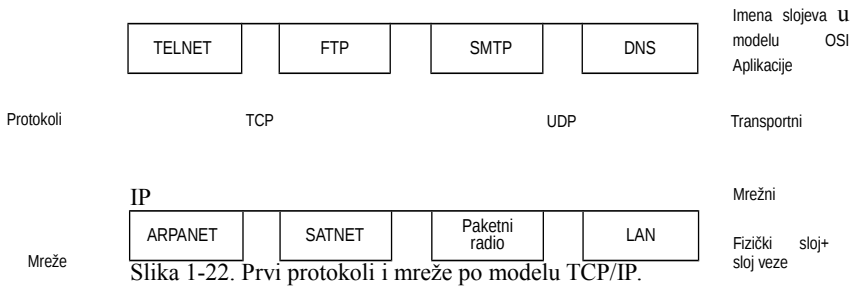


Slika 1-21. Referentni model TCP/IP.

Transportni sloj

Sloj iznad međumrežnog sloja u modelu TCP/IP danas se obično naziva **transportni sloj** (engl. *transport layer).* On je namenjen konverzaciji između ravnopravnih procesa na izvornom i odredišnom računaru, baš kao i transportni OSI sloj. Ovde su definisana dva protokola koji spajaju dva kraja. Prvi, **protokol za upravljanje prenosom** (engl. *Transmission Control Protocol, TCP),* predstavlja pouzdan protokol sa uspostavljanjem direktne veze, koji omogućava da se tok bajtova potekao s jednog računara bez greške dovede do bilo kog odredišta u međumreži. On deli početni tok bajtova na zasebne poruke i svaku prosleđuje međumrežnom sloju. Prihvatni TCP proces na odredištu uređuje primljene poruke i od njih ponovo obrazuje tok bajtova. TCP takođe upravlja tokom podataka tako da brzi pošiljalac ne može da zatrpa sporog primaoca s više poruka nego što ovaj može da obradi.

Drugi protokol ovog sloja, **protokol za korisničke datagrame** (engl. *User Datagram Protocol, UDP),* predstavlja nepouzdan protokol bez uspostavljanja direktne veze, namenjen aplikacijama koje same, umesto protokola TCP, uređuju svoje pakete i upravljaju tokom podataka. On se široko koristi i za jednostavne upite (zahtev-od- govorj klijentsko-serverskog tipa, kao i za aplikacije kod kojih hitnost isporuke ima prednost nad tačnošću, npr. prenos govora ili videa. Odnosi između protokola IP, TCP i UDP prikazani su na slici 1-22. Od nastanka modela, protokol IP je ugrađen i u mnoge druge mreže.



Slika 1-22. Prvi protokoli i mreže po modelu TCP/IP.

Sloj aplikacija

U modelu TCP/IP nema sloja sesije, niti sloja prezentacije. Za njima nije bilo potrebe, pa nisu ni uključeni u model. Iskustvo s modelom OSI potvrđuje takvo gledište: ti slojevi su većini aplikacija od male koristi.

Iznad transportnog sloja nalazi se **sloj aplikacija** (engl. *application layer)*. On sadrži sve protokole višeg nivoa. Na početku su to bili protokoli za virtuelni terminal (TELNET), za prenos datoteka (FTP) i za elektronsku poštu (SMTP), kao na slici 1-22. Protokol za virtuelni terminal omogućava korisniku da se sa svog računara daljinski prijavi na drugi računar i da na njemu radi. Protokol za prenos datoteka omogućava efikasno prenošenje podataka s jednog računara na drugi. Elektronska pošta je na početku ličila na prenos datoteka, ali je kasnije za nju razvijen poseban protokol (SMTP). Tokom godina, sloju aplikacija dodati su mnogi drugi protokoli: sistem imenovanja domena (DNS) za preslikavanje (prevođenje) imena računara u njihove mrežne adrese, protokol za prenošenje poruka USENET-ovih diskusionih grupa (NNTP), protokol za preuzimanje strana s World Wide Weba (HTTP) i mnogi drugi.

Sloj za povezivanje računara s mrežom

Ispod međumrežnog sloja zjapi velika praznina. Referentni model TCP/IP ne objašnjava detaljno šta se tu događa, osim što ističe da računar mora da se poveže s mrežom pomoću nekog protokola kako bi mogao da joj šalje IP pakete. Sam protokol za povezivanje s mrežom nije definisan i menja se od računara do računara i od jedne mreže do druge. Knjige i radovi o modelu TCP/IP retko se dotiču ove teme.

1.4.3 **Poređenje** referentnih modela OSI i TCP/IP

Referentni modeli OSI i TCP/IP imaju mnogo zajedničkog. Oba se zasnivaju na konceptu skupa nezavisnih protokola. Isto tako, funkcionalnost slojeva je prilično slična. Na primer, u oba modela svi slojevi zaključno s transportnim slojem treba da obezbede transportnu uslugu, koja nezavisno od mreže povezuje oba kraja i obrađuje njihove zahteve za komuniciranjem. Ti slojevi su davaoci usluge transporta. Dalji slojevi iznad transportnog predstavljaju aplikacije koje su korisnici transportnih usluga.

Uprkos navedenim načelnim sličnostima, između modela postoje i mnoge razlike, pa ćemo se u ovom odeljku pozabaviti onim ključnim. Treba naglasiti da ćemo pore- diti referentne modele, a ne skupove protokola. O protokolima ćemo govoriti kasnije. Poređenju modela OSI i TCP/IP i razlikama između njih posvećena je i čitava knjiga (Piscitello i Chapin, 1993).

Za model OSI su ključna tri koncepta:

1. Usluge.
2. Interfejsi.
3. Protokoli.

Verovatno je najveći doprinos modela OSI to što je povukao jasne granice između ova tri koncepta. Svaki sloj obavlja određene usluge za sloj iznad sebe. Definicija *usluge* ukazuje na ono šta sloj radi, a ne kako će joj elementi gornjeg sloja pristupati ili kako radi sam sloj u kome se nalazi. Definicija usluge sadrži semantiku sloja.

Interfejs između slojeva ukazuje procesima iz gornjeg sloja kako da pristupe donjem sloju. On određuje koje parametre treba upotrebiti i kakvi se rezultati mogu očekivati. Međutim, ni on ne otkriva ništa o tome kako donji sloj radi.

I na kraju, ravnopravni *protokoli* koji se koriste unutar sloja tiču se samo tog sloja. Sloj može da koristi kakve god hoće protokole, sve dok obavlja predviđene zadatke (tj. izvršava usluge koje nudi). On ih takođe može proizvoljno menjati a da to ne utiče na softver u višim slojevima.

Navedeni pristup se veoma dobro slaže sa suvremenim pristupom objektno orijen- tisanom programiranju. Slično sloju, objekat ima skup metoda (operacija) koje mogu da pozovu spoljni procesi. Semantika ovih metoda definiše skup usluga koji objekat nudi. Parametri metoda i rezultati obrazuju interfejs objekta. Interni kod objekta predstavlja njegov protokol koji se spolja ne vidi, niti ima značaja izvan objekta.

Model TCP/IP na početku nije povukao jasnu razliku između usluge, interfejsa i protokola, mada su kasnije činjeni pokušaji da se on približi modelu OSI. Na primer, jedine stvarne usluge koje nudi njegov među mrežni sloj jesu usluge SEND IP PACKET i RECEIVE IP PACKET.

Zbog toga su protokoli u modelu OSI bolje skriveni nego u modelu TCP/IP i mogu se s napretkom tehnologije lakše zameniti. Mogućnost takvih zamena je i jedan od glavnih ciljeva arhitekture s protokolima raspoređenim po slojevima.

Referentni model OSI je razvijen pre nastanka odgovarajućih protokola. To znači da model nije pravljen prema određenom skupu protokola, što ga čini opštijim. Ovo ima i svoju lošu stranu jer su projektanti, nemajući previše iskustva s takvim stvarima, često lutali pri dodeljivanju funkcionalnosti pojedinim slojevima.

Na primer, sloj veze podataka prvobitno je radio samo s mrežama od tačke do tačke. Kada su se pojavile mreže s difuznim emitovanjem poruka, modelu se morao „prikačiti“ nov podsloj. Kada je počela gradnja stvarnih mreža na osnovu modela OSI i postojećih protokola, otkriveno je da one ne odgovaraju zahtevanim specifikacijama usluga (o, čuda!), pa su morali biti dodati podslojevi konvergencije da bi se ove razlike prevazišle. Navedimo na kraju i to da su autori modela prvobitno očekivali da će svaka država imati jednu mrežu kojom upravlja vlada i koristiti OSI protokole, tako da niko nije ni razmišljao o radu u kombinovanoj mreži. Da bismo skratili priču, recimo samo da stvari nisu išle tim tokom.

S modelom TCP/IP dogodilo se upravo suprotno: najpre su se pojavili protokoli, pa model

- koji je zapravo bio samo opis postojećih protokola. Sa uklapanjem protokola u model nije bilo problema - uklapali su se savršeno. Problem je bio u tome što se sam model nije mogao uklopiti ni u jedan drugi skup protokola. Zbog toga on baš nije naročito koristan za opisivanje drugačijih mreža.

Ako se od filozofiranja okrenemo praktičnijim stvarima, očiglednu razliku između dva modela čini broj slojeva: model OSI ima sedam slojeva, a model TCP/IP četiri. Oba imaju (među)mrežni sloj, transportni sloj i sloj aplikacija, ali se ostali slojevi razlikuju.

Druga razlika se odnosi na komunikaciju: sa uspostavljanjem direktne veze ili bez nje. Model OSI u mrežnom sloju podržava obe vrste komunikacije, ali u transportnom sloju - gde je ona i najhitnija (jer transportnu uslugu korisnici vide) - podržava samo komunikaciju sa uspostavljanjem direktne veze. Model TCP/IP u mrežnom sloju podržava samo jedan režim komunikacije (bez uspostavljanja direktne veze), ali podržava oba režima u transportnom sloju, nudeći korisnicima izbor. Takav izbor je posebno važan za jednostavne protokole odgovaranja na upite.

1.4.4 Kritika modela OSI i njegovih protokola

Ni model OSI sa svojim protokolima, ni model TCP/IP sa svojim, nisu savršeni. I jednom i drugom je upućeno dosta kritike. U ovom i sledecem odeljlcu razmotrićemo neke od tih zamerki. Počećemo s modelom OSI, dok ćemo TCP/IP ostaviti za kasnije.

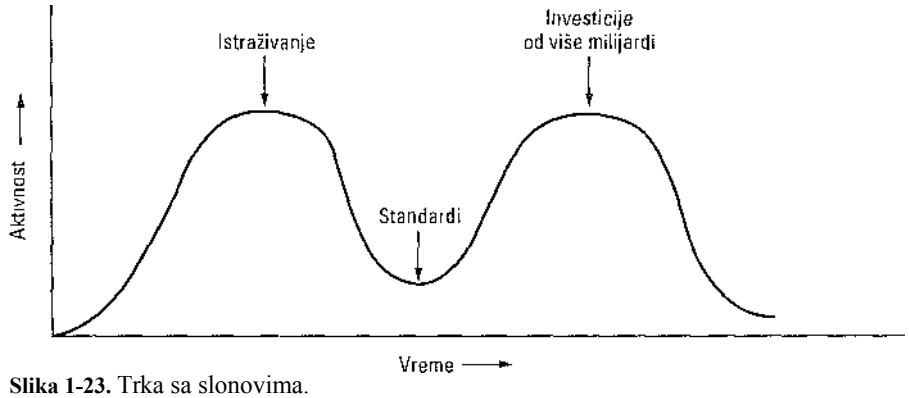
U trenutku kada se pojavilo drugo izdanje ove knjige (1989), mnogim stručnjacima je izgledalo da će se model OSI i njegovi protokoli proširiti svetom i potisnuti sve drugo na tom polju. To se ipak nije dogodilo. Zašto? Spisak onoga što se stvarno dogodilo može da bude poučan:

1. Loša sinhronizacija.
2. Loša tehnologija.
3. Loša realizacija.
4. Loša politika.

Loša sinhronizacija

Razmotrimo najpre prvi razlog: biranje lošeg trenutka za objavljivanje modela. Izbor pravog trenutka pojave standarda apsolutno je ključan za njegov uspeh. David Clark s Masačusetskog tehničkog instituta ima teoriju poznatu kao trka sa slonovima, koja je ilustrovana slikom 1-23.

Dijagram na slici prikazuje količinu aktivnosti ispoljenu u vezi s nekom novotarijom. Kada se otkrije nešto novo, nastaje eksplozija istraživačke aktivnosti u obliku diskusija, naučnih radova i skupova. Posle nekog vremena ta aktivnost jenjava, a industrijske korporacije shvataju da se radi o nečem novom i počinje poplava investicija.



Slika 1-23. Trka sa slonovima.

Suštinski je važno da se standardi umetnu u uzak prostor između dva „slona“ - dva maksimuma na dijagramu. Ako se objave prerano, pre završenih istraživanja, novost neće u potpunosti biti shvaćena, a rezultat su loši standardi. Ako se objave prekasno, kompanije su već investirale velik novac u pravcu koji se razlikuje od standarda, pa se standardi ne poštuju. Ako je rastojanje između slonova vrlo malo (pošto su svi u žurbi da otpočnu investiranje), autorima standarda (trkačima) preti opasnost da budu zgnječeni.

Danas je jasno da su standardni OSI protokoli upravo tako smrvljeni. U trenutku kada su se pojavili OSI protokoli, konkurentni TCP/IP protokoli već su bili u širokoj upotrebi u akademskom okruženju. Iako veliko investiranje još nije uhvatilo maha, akademsko tržište bilo je za proizvođače dovoljno veliko, pa su mnogi počeli oprezno da nude TCP/IP proizvode. Kada se pojavio model OSI, oni nisu našli računa da dobrovoljno podržavaju još jedan paralelni skup protokola, pa na početku nisu ni nudili nove proizvode. U situaciji kada svaka kompanija čeka da neko drugi probije led, model OSI nije ni zaživeo.

Loša tehnologija

Drugi razlog što model OSI nije zaživeo jeste to što je bio prepun nedostataka, baš kao i njegovi protokoli. Izbor sedam slojeva bio je više političke, nego tehničke prirode, a njegova dva sloja (sesije i prezentacije) gotovo su prazni, dok su druga dva (veze podataka i mrežni) pretrpani.

Model OSI, zajedno s definicijama svojih usluga i protokolima, izuzetno je složen. Štampana verzija standarda predstavlja brdo papira visoko skoro metar. Usluge i protokoli teško se ugrađuju i neefikasni su u radu. Ovde se i nehotice setim pitalice koju je postavio Paul Mockapetris (Rose, 1993):

Pitanje: Šta dobijete kada ukrstite gangstera s međunarodnim standardom?
Odgovor: Nekoga ko vam predlaže nešto što ne razumete.

Osim što je nerazumljiv, u modelu OSI se neke funkcije, npr. za adresiranje, upravljanje tokom i kontrolu grešaka, ponavljaju u svakom sloju. Salzer i saradnici (1984) istakli su, na primer, da se efikasnost može postići ako se kontrola grešaka ugradi u najviši sloj; njeno ponavljanje u svakom od nižih slojeva često je nepotrebno i neefikasno.

Loša realizacija

Imajući u vidu složenost modela i njegovih protokola, ne čudi što su prve njegove realizacije bile ogromne, nezgrapne i spore. Svako ko se usudio da ih isproba, dobro se opekao. Nije trebalo dugo da se pojam „OSI“ izjednači s „niskim kvalitetom“. Iako su pojedine komponente vremenom usavršavane, početni utisak je ostao.

Nasuprot tome, prva realizacija modela TCP/IP bila je deo Berkeley UNIX-a i bila je prilično dobra (da i ne pominjemo da je bila besplatna). Ljudi su je brzo prihvatali, što je stvorilo veliku zajednicu korisnika, koja je vodila ka daljim poboljšanjima. To je sa svoje strane dovelo do proširivanja zajednice i tako ukrug. Ovde je istorija tekla uzlaznom spiralom, umesto obrnuto.

Loša politika

Zbog prvobitne realizacije modela, mnogi su, naročito u akademskim krugovima, smatrali daje TCP/IP deo TJNIX-a, a UNIX je osamdesetih godina bio pravo utočište za američke akademce.

Za OSI se, s druge strane, smatralo da predstavlja kreaciju evropskih ministarstava za telekomunikacije, Evropske unije i kasnije, Američke vlade. To je samo delimično tačno, ali je za negativan stav bila dovoljna i sama pomisao na gomilu državnih birokrata koji pokušavaju da tehnički inferioran standard uvale neispavanim istraživačima i programerima koji na prvoj borbenoj liniji mukotrpno sklapaju računaraske mreže. Neki su ovaj standard poredili sa izjavom rukovodilaca IBM-a iz šezdesetih godina da će PL/I postati programski jezik budućnosti i kasnijom ispravkom Ministarstva odbrane SAD, da se u stvari radi o jeziku Ada.

1.4.5 Kritika referentnog modela TCP/IP

Model TCP/IP i njegovi protokoli takođe imaju svoje probleme. Prvo, model ne razgraničava jasno koncepte usluga, interfejsa i protokola. Dobra praksa softverskog inženjerstva zahteva razlikovanje specifikacije i realizacije, nešto što je u modelu OSI urađeno vrlo pažljivo, a u modelu TCP/IP traljavo. Zbog toga model TCP/IP nije od velike pomoći kada treba projelctovati nove mreže s novim tehnologijama.

Drugo, model TCP/IP ni izbliza nije dovoljno uopšten i veoma šturo može da opiše bilo koji skup protokola osim skupa TCP/IP protokola. Opisati, na primer, Bluetooth pomoću modela TCP/IP sasvim je nemoguće.

Treće, sloj za povezivanje računara s mrežom u stvari nije sloj u uobičajenom značenju koje se koristi kod protokola raspoređenih po slojevima. To je interfejs (između mreže i slojeva veze podataka). Između interfejsa i sloja postoji suštinska razlika i tu nema mesta približnim definicijama.

Četvrto, Model TCP/IP ne razdvaja (čak i ne pominje) fizički sloj i sloj veze podataka. Ta dva sloja su potpuno različita. Fizički sloj se bavi prenosnim svojstvima bakarne žice, optičkog vlakna i bežičnih komunikacija. Zadatak sloja veze podataka jeste da označi početak i kraj okvira podataka i da ga prenese s jedne na drugu stranu uz željen stepen pouzdanosti. Jedan potpun model morao bi ove slojeve da ima kao zasebne, dok se u modelu TCP/IP oni i ne pominju.

Na kraju, iako su protokoli IP i TCP brižljivo projektovani i dobro realizovani, mnogi drugi protokoli su urađeni na brzinu - najčešće ih je pravila grupa diplomaca koji su primenjivali programerske trikove dok im sve nije dosadilo. Realizacije protokola su zatim distribuirane besplatno, zbog čega su bile široko korišćene, potpuno prihvaćene i nerado zamenjivane. Neke od njih su danas više neprijatnost, nego korist. Protokol za virtuelni terminal (TELNET), na primer, projektovan je za mehanički teleprinterski terminal brzine deset znakova u sekundi. On ništa ne zna o grafičkom korisničkom okruženju i miševima. Pa ipak, posle 25 godina, još uvek je u širokoj upotrebi.

Sve u svemu, uprkos problemima, model OSI (bez slojeva sesije i prezentacije) pokazao se izuzetno korisnim u razmatranju računarskih mreža. Međutim, njegovi protokoli nisu naišli na šire prihvatanje. Nasuprot tome, model TCP/IP praktično ne postoji, ali se njegovi protokoli masovno koriste. Pošto računardžije takođe žele da dobiju deo kolača, u ovoj knjizi ćemo kao model koristiti modifikovani model OSI, a istovremeno se koncentrisati na TCP/IP i srodne protokole, kao i na novije protokole, kao što su 802, SONET i Bluetooth. U stvari, kao okvir za teme ove knjige koristiće- mo hibridni model prikazan na slici 1-24.

sloj aplikacija
transportni sloj
mrežni sloj sloj
veze podataka
fizički sloj

5 4 3 **2** 1

Slika 1-24. Hibridni referentni model koji će biti korišćen u knjizi.

1.5 PRIMERI MREŽA

Tema umrežavanja računara obuhvata mnoge vrste mreža, malih i velikih, poznatih i manje poznatih. One se grade s različitom svrhom, različite su veličine i zasnovane su na različitim tehnologijama. U narednim odeljcima razmotrićemo nekoliko primera računarskih mreža da bismo stekli utisak o njihovoj različitosti.

Počećemo sa Internetom, verovatno najpoznatijom mrežom, opisati njegovu isto- riju, razvoj i tehnologiju. Zatim ćemo razmotriti ATM, mrežu koja se često nalazi u središtu velikih (telefonskih) sistema. Ona se tehnički veoma razlikuje od Interneta, pa je zgodno da ih uporedimo. Posle toga ćemo govoriti u Ethernetu, najčešćoj lokalnoj mreži, i na kraju ćemo razmotriti IEEE 802.11, standard za bežične lokalne mreže.

1.5.1 Internet

Internet uopšte nije mreža, već ogroman skup različitih mreža u kojima se koriste neki zajednički protokoli i obezbeđuju neke zajedničke usluge. Internet je jedinstven po tome što ga niko nije planirao i što niko njime ne upravlja. Da bismo ovo bolje ra- zumeli, krenimo od početka i pogledajmo kako i zašto je nastao. Za upoznavanje sa sveobuhvatnom istorijom Interneta, toplo preporučujemo knjigu Johna Naughtona
(2000) . To je jedna od onih retkih knjiga koja nije samo zabavna za čitanje, već je i prepuna oznaka *ibid,* i *op. cit.,* bez kojih ne mogu ozbiljni istoričari. Nešto od onoga o čemu govorimo u nastavku zasniva se na materijalu iz te knjige.

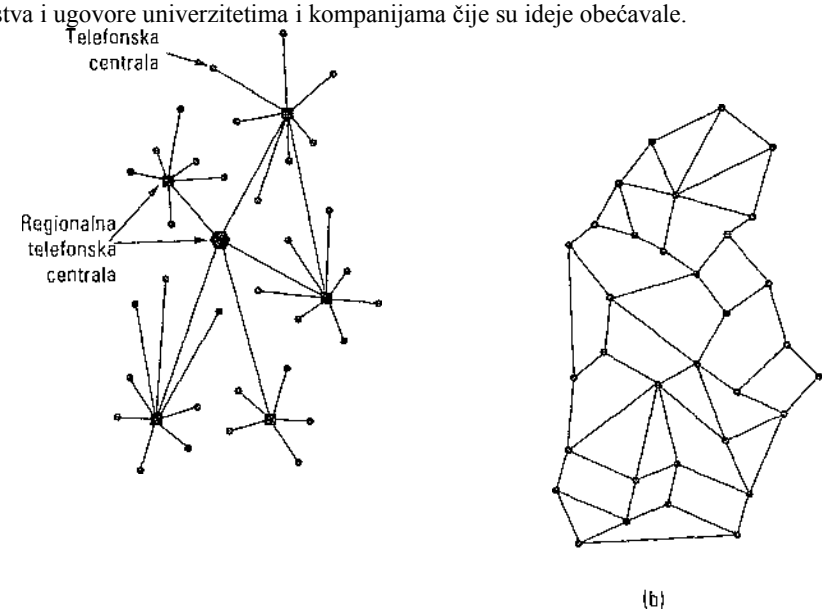
ARPANET

Priča počinje krajem pedesetih godina. Na vrhuncu hladnog rata, Ministarstvo odbrane SAD poželelo je da ima komandno-kontrolnu mrežu koja bi mogla da izdrži nuklearni udar. U to doba su sve vojne komunikacije koristile javni telefonski sistem koji je smatran ranjivim. To ćete lako shvatiti ako pogledate sliku 1-25(a). Na njoj su tačkama predstavljene telefonske centrale, od kojih je svaka povezana s hiljadama telefona. Te centrale su, sa svoje strane, povezane za centrale višeg nivoa (regionalne centrale), čineći državnu hijerarhiju telefonskog sistema u kome ima malo rezervnih komponenata. Ranjivost sistema ogleda se u tome što se razaranjem nekoliko ključnih regionalnih centrala on raspada na više izolovanih ostrvaca.

Oko 1960. godine, Ministarstvo odbrane je ugovorom obavezalo korporaciju RAND da pronađe rešenje. Jedan od zaposlenih u korporaciji, Paul Baran, izišao je s predlogom projekta široko distribuirane mreže otporne na greške, prikazane na slici l-25(b). Pošto su, po projektu, razdaljine između susednih centrala bile prevelike da bi analogni signali putovali bez izobličenja, Baran je za ceo sistem predložio tehnologiju komutiranja digitalnih paketa. Baran je za Ministarstvo napisao više izveštaja u kojima je svoju ideju obrazložio do detalja. Njegov koncept se svideo zvaničnicima u Pentagonu, pa su zatražili od tadašnje nacionalne monopolske telefonske kompanije u SAD, korporacije AT&T, da izgradi prototip. Korporacija AT&T nije želela čak ni da razmotri Baranovu ideju. Najveća i najbogatija korporacija na svetu nije mogla dozvoliti da joj neki uobraženi žutokljunac diktira kako da izgradi telefonski sistem. Izvestili su Ministarstvo da se Baranova mreža ne može napraviti i projekat je stavljen ad acta.

Prošlo je više godina, a Ministarstvo odbrane još uvek nije imalo bolji lcomandno- -kontrolni sistem. Da bismo razumeli šta se tada dogodilo, treba da se vratimo na oktobar 1957, kada je Sovjetski Savez porazio SAD u svemirskoj trci lansirajući prvi veštač- ki satelit - Sputnjik. Kada je predsednik Ajzenhauer pokušao da utvrdi „ko se uspavao“, zapanjio se videvši kako se Armija, Mornarica i Vazduhoplovstvo otimaju za istraživački budžet Pentagona. Njegova neposredna reakcija bila je da osnuje jedinstvenu istraživačku organizaciju za poslove odbrane, ARPA, Advanced Research Project Agency (Agencija za napredne istraživačke projekte). Organizacija ARPA nije imala svoje

istraživače, ni laboratorije; u stvari, imala je samo jednu kancelariju i mali budžet (po metilima Pentagona). Zadatak zbog kojeg je osnovana obavljala je dodeljujući finan- sijska sredstva i ugovore univerzitetima i kompanijama čije su ideje obećavale.



Slika 1-25. (a) Struktura telefonskog sistema, (b) Baranov predlog distribuiranog sistema komutacije.

Prvih nekoliko godina ARPA je pokušavala da ustanovi šta joj je zadatak, ali je 1967. pažnju njenog tadašnjeg direktora, Larryja Robertsa, privuklo umrežavanje. On se posavetovao s više stručnjaka da bi odlučio šta da radi. Jedan od njih, Wesley Clark, predložio je izgradnju podmreže s komutiranjem paketa, pri čemu bi svaki umreženi računar imao svoj usmerivač (slika 1-10).

Posle izvesnog oklevanja, Roberts je prihvatio ideju i podneo o njoj pomalo uopšteno saopštenje na Simpozijumu ACM SIGOPS o principima operativnih sistema u Gatlinburgu u državi Tenesi (Roberts, 1967). Na Robertsovo iznenađenje, na simpozijumu se pojavilo i saopštenje o vrlo sličnom sistemu koji ne samo što je bio projektovan, već je bio i realizovan pod rukovodstvom Donalda Daviesa u Nacionalnoj laboratoriji za fiziku (National Physical Laboatory, NPL) u Engleskoj. NPL sistem nije bio nacionalnog značaja (povezivao je samo nekoliko laboratorijskih računara), ali je dokazao da komutiranje paketa radi. Staviše, u saopštenju se citirao Baranov rad koji je Ministarstvo odbrane odbacilo. Roberts je otišao iz Gatlinburga čvrsto naumivši da napravi ono što je kasnije postalo poznato kao ARPANET.

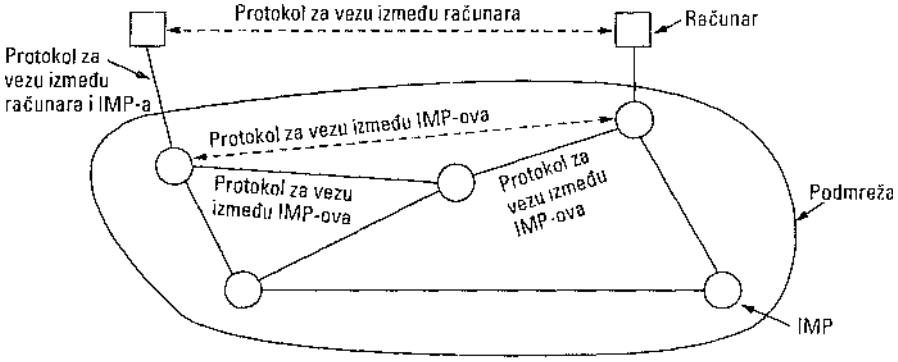
Podmreža bi se sastojala od miniračunara zvanih **obrađivači poruka na interfejsu** (engl. *Interface Message Processors*, *IMPs),* povezanih linijama brzine prenosa 56 kb/s. Zbog veće pouzdanosti rada, svaki IMP bi bio povezan s barem još dva druga IMP-a. Podmreža bi bila datagramska, kako bi u slučaju uništenja nekih linija i IMP-a poruka mogla da se automatski

preusmeri dragom putanjom.

Svaki čvor mreže sastojao bi se od IMP-a i umreženog računara, smeštenih u istu prostoriju i povezanih kratkim kablom. Računar bi IMP-u mogao da šalje poruke veličine do 8063 bita koje bi ovaj razbijao na pakete veličine 1008 bitova i nezavisno ih prosleđivao ka odredištu. Svaki paket bi pre daljeg prosleđivanja morao biti primljen u celini, tako da podmreža koju je zamislio Roberts predstavlja prvu mrežu po sistemu komutiranja „čuvaj i prosledi".

ARPA je tada objavila tender za izgradnju podmreže na koji se prijavilo dvanaest kompanija. Posle razmatranja ponuda, ARPA je izabrala kompaniju BBN, konsul- tantsku firmu iz Kembridža u Masačusetsu i decembra 1968. sklopila s njom ugovor da izgradi podmrežu i napravi odgovarajući softver. Kompanija BBN je za obrađi- vače poruka na interfejsu izabrala Honeywellove miniračunare DDP-316 sa osnovnom memorijom od 12 KB 16-bitnih reči. Miniračunari nisu imali diskove jer su pokretni delovi smatrani nepouzdanim. Bili su međusobno povezani linijama brzine prenosa 56 kb/s iznajmljenim od telefonskih kompanija. Iako brzina od 56 kb/s danas predstavlja izbor tinejdžera koji nema novca za ADSL ili kablovsku vezu, u to vreme je to bio maksimum.

Softver je bio podeljen u dve celine: za podmrežu i za računare. Softver za podmrežu sadržao je IMP kraj veze između računara i IMP-a, protokol za vezu između dva uzastopna IMP-a i, radi veće pouzdanosti, protokol za vezu između izvornog i odredišnog IMP-a. Prvobitni projekat ARPANET-a prikazanje na slici 1-26.



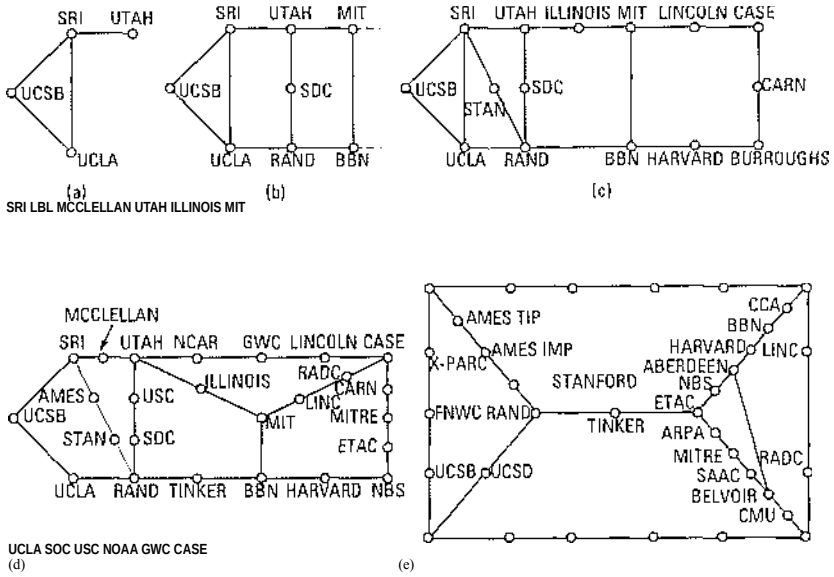
Slika 1-26. Prvobitni projekat ARPANET-a.

Softver je bio neophodan i izvan podmreže: računarski kraj veze IMP-računar; protokol za vezu između računara i aplikacioni softver. Ubrzo je postalo jasno da kompanija BBN, čim je uspela da poruke poslate s veze između računara i IMP-a na jednom kraju dostavi na odredišnu vezu između računara i IMP-a, smatra da je time njen posao završen.

Robertsu je tako ostao problem računarskog softvera. Da bi ga rešio, sazvao je leta 1969. skup istraživača mreža, uglavnom tek diplomiranih studenata, u Snowbirdu u državi Juta. Diplomirani studenti su očekivali da će im neki ekspert za mreže objasniti

grandiozni projekat mreže i njenog softvera, a zatim svakome dodeliti deo koji treba da uradi. Bili su neprijatno iznenađeni kada su shvatili da nema ni eksperta, ni projekta. Morali su sami da utvrde šta im je posao.

Pa ipak, eksperimentalna mreža je nekako puštena u rad decembra 1969; imala je četiri čvora: UCLA, UCSB, SRI i Univerzitet Jute. Te četiri ustanove izabrane su jer su sve imale više ugovora sa ARPA-om i sve su imale različite, međusobno nekompatibilne računare (kako bi sve bilo zabavnije). Mreža je brzo rasla sa svakim isporučenim i instaliranim IMP-om i ubrzo je pokrila čitave Sjedinjene Države. Slika 1-27 prikazuje brzinu rasta ARPANET-a tokom prve 3 godine.



Slika 1-27. Razvoj ARPANET-a. (a) Decembar 1969. (b) Jul 1970. (c) Mart 1971. (d) April 1972. (e) September 1972.

Osim pomaganja razvoja ARPANET-a, organizacija ARPA je finansirala i istraživanja upotrebe satelitskih mreža za mobilni paketni radio. U jednoj od sada čuvenih demonstracija, kamion koji je krstario Kalifornijom koristio je mrežu palcetnog radija da bi slao poruke Istraživačkom institutu u Stenfordu (SRI), odakle su one ARPA- NET-om prosleđivane na istočnu obalu, a zatim isporučivane Univerzitetskom koledžu u Londonu satelitskom mrežom. To je istraživačima koji su kamionom krstarili Kalifornijom omogućilo da istovremeno koriste računar u Londonu.

Eksperiment je istovremeno pokazao da postojeći ARPANET protokoli nisu pogodni za rad u više mreža. To zapažanje je podstaklo razvoj protokola, čiji je vrhunac bio stvaranje modela TCP/IP i njegovih protokola. (Cerf i Khan, 1974). TCP/IPje posebno projektovan za međumrežni rad, što je postajalo sve važnije kako su se ARPA- NET-u priključivale i drage mreže.

Da bi podstakla prihvatanje novih protokola, ARPA je kompaniji BBN i Univerzitetu Kalifornije u Berkliju dodelila više ugovora da protokole integrišu u Berkeley UNIX. Istraživači u Berkliju su smislili odgovarajući programski interfejs za mrežu (utičnice, engl. *sockets)* i napisali mnoge aplikacije, uslužne programe i programe za održavanje sistema kako bi olakšali rad s mrežom.

Trenutak je bio dobro izabran. Mnogi univerziteti su upravo dobili drugi ili treći VAX računar i LAN da ih povežu, ali nije bilo mrežnog softvera. Kada se pojavila verzija 4.2 Berklijevog softvera (4.2BSD), zajedno s TCP/IP protokolima, utičnicama i mnogim mrežnim uslužnim programima, softverski paket je odmah prihvaćen. Sta- više, lokalna mreža se pomoću TCP/IP protokola lako povezivala na ARPANET, pa je to u većini slučajeva i činjeno.

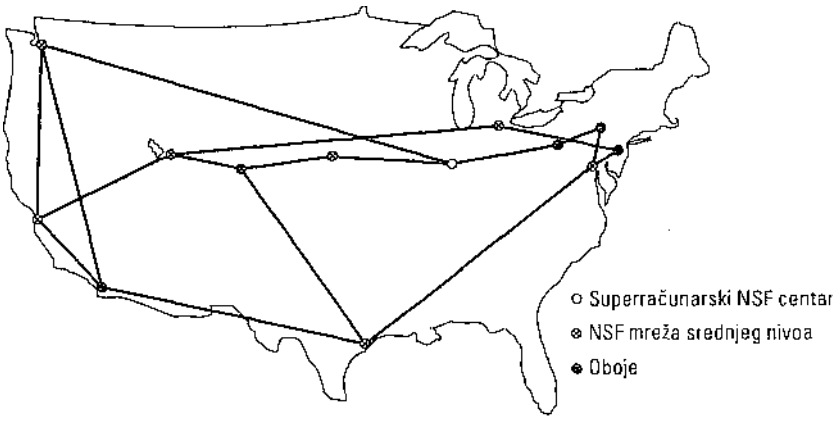
Tokom osamdesetih godina, ARPANET-u su se priključivale dodatne mreže, naročito lokalne. Kako je mreža rasla, pronalaženje računara na mreži bilo je sve teže, pa je napravljen **sistem imenovanja domena** (engl. *Domain Name System, DNS)* da bi se računari svrstali u domene i omogućilo prevođenje imena računara u njihove IP adrese. Od tada je DNS prerastao u opšti sistem distriburianih baza podataka za skladištenje brojnih informacija u vezi sa imenovanjem. Detaljno ćemo ga obraditi u 7. poglavlju.

NSFNET

Krajem sedamdesetih godina NSF (Američka nacionalna fondacija za nauku) shvatila je koliki uticaj ima mreža ARPANET na istraživački rad na univerzitetima time što omogućava i naučnicima širom zemlje da razmenjuju podatke i sarađuju na istraživačkim projektima. Međutim, da bi se priključio ARPANET-u, univerzitet je morao da ima ugovor s Ministarstvom odbrane, a njega mnogi nisu imali. NSF je zato rešila da projektuje naslednika ARPANET-a, mrežu na koju bi se slobodno mogle povezati sve univerzitetske istraživačke grupe. Konkretan temelj ovoj zamisli postavili su kada su odlučili da izgrade mrežnu okosnicu (engl. *backbone)* koja će povezati šest centara sa superračunarima: u San Dijegu, Bulderu, Šampanji, Pitsburgu, Itaki i Prinstonu. Svaki superračunar je dobio „malog brata“ - mikroračunar LSI-11, nazvan **čupava loptica** (engl.*fuzzball)*. Ti „čupavci“ su bili međusobno povezani iznajmljenim linijama brzine prenosa 56 kb/s, čineći podmrežu, a upotrebljena je ista hardverska tehnologija koja je korišćena i za ARPANET. Softverska tehnologija je, međutim, bila drugačija: čupave loptice su se od početka sporazumevale protokolom TCP/IP, tako da je to bila prva regionalna TCP/IP mreža.

NSF je finansirala i izgradnju dvadesetak regionalnih mreža koje su spajane sa okosnicom, kako bi pomogla hiljadama univerziteta, istraživačkih laboratorija, biblioteka i muzeja da pristupe svakom superračunaru i da međusobno razmenjuju poruke. Cela mreža, okosnica i regionalne mreže, nazvana je NSFNET. Ona se povezivala sa ARPANET-om preko veze između jednog IMP-a i mikroračunara u računarskom centra Univerziteta Camegie-Mellon. Prva okosnica NSFNET-a prikazana je na slici 1-28.

Mreža NSFNET pokazala se odmah tako uspešnom da je ubrzo bila zagušena. NSF je odmah počela da planira njenog naslednika i dodelila ugovor za izgradnju konzorcijumu MERIT iz Mičigena. Za 2. verziju okosnice od kompanije MCI (pošto se ujedinila s WorldComom) iznajmljeni su optički kanali brzine prenosa 448 kb/s, a za usmerivače su iskorišćeni IBM-ovi personalni računari zasnovani na RISC tehnologiji (PC-RT). I ovaj kapacitet je uskoro prevaziđen, pa je 1990. propusna moć okosnice povećana na 1,5 Mb/s.



Slika 1-28. Okosnica mreže NSFNET 1988. godine.

Kako je mreža rasla, NSF je shvatila da država neće stalno moći da je finansira. Komercijalnim organizacijama koje su želele da se priključe, to su zabranjivali propisi Fondacije. Zbog toga je NSF nagovorila kompanije MERIT, MCI i IBM da formiraju neprofitnu korporaciju ANS (Advanced Networks and Services), kao prvi korak ka komercijalizovanju. Godine 1990, ANS je preuzela NSFNET i unapredila njenu 1,5 megabitnu vezu do brzine od 45 Mb/s, nazvavši novu mrežu ANSNET. Ta meža je radila pet godina, a onda je prodata organizaciji American Online. U to vrerne su već različite kompanije nudile komercijalne IP usluge i postalo je jasno da država treba da se izvlači iz ovog posla.

Da bi olakšala prelazni period i obezbedila da svaka regionalna mreža može da komunicira sa svakom drugom regionalnom mrežom, NSF je ugovorima obavezala četiri različita mrežna operatera da uspostave tačke pristupa mreži (engl. *Network Access Points, NAPs).* To su bili PacBell (San Francislco), Ameritech (Čikago), MFS (distrikt Vašington) i Sprint (Njujork, gde se za svrhe povezivanja Pennsaulcen u Nju Džerziju računao u područje Njujorka). Svaki mrežni operater koji je regionalnim NSF mrežama želeo da ponudi usluge povezivanja sa okosnicom, morao je da se poveže sa svim ostalim pristupnim tačkama.

Takva konstrukcija je značila da paket koji krene s bilo koje regionalne mreže može da bira centrale na okosnici da bi stigao od svoje do odredišne pristupne tačke. Shodno tome, kompanije koje su održavale centrale bile su prinuđene da se međusobno nadmeću u pogledu usluge i cene, što je i bila osnovna ideja. Rezultat toga je bio da je jedinstvena podrazumevana okosnica zamenjena infrastrukturom zasnovanom na komercijalnim principima i konkurenciji. Mnogi su skloni da kritilcuju Saveznu vladu što nije preuzela inicijativu, ali treba biti iskren pa priznati da su Ministarstvo odbrane i Fondacija za nauku prvi stvorili infrastrukturu koja predstavlja osnovu Interneta, a zatim je predali industriji daje koristi i unapređuje.

Tokom devedesetih godina, mnoge druge države i regioni izgradili su nacionalne istraživačke mreže, često po obrascu ARPANET-a i NSFNET-a. Među njima su u Evropi mreže EuropaNET i EBONE, koje su počele sa 2 Mb/s, a zatim poboljšavane do brzine 34 Mb/s. Na kraju je i u Evropi mrežna infrastruktura predata industriji.

Korišćenje Interneta

Broj mreža, računara i korisnika priključenih na ARPANET brzo je rastao nakon što je 1. januara 1983. TCP/IP postao jedini zvanični skup protokola. Kada su NSF- NET i ARPANET međusobno povezani, rast mreže je postao eksponencijalan. Priključile su se mnoge regionalne mreže, a uspostavljene su veze i s mrežama u Kanadi, Evropi i na Pacifiku.

Sredinom osamdesetih godina neki su takav skup mreža počeli da posmatraju kao veliku međumrežu (engl. *internet)* koja je kasnije dobila ime Internet, mada nikakav visoki zvaničnik nije tom prilikom razbio bocu šampanjca o računar.

Vezivo koje drži Internet na okupu jeste model TCP/IP i njegov skup protokola. TCP/IP omogućava davanje univerzalne usluge i može se uporediti sa standardom o širini koloseka koji su usvojile železnice u 19. veku ili sa standardnim protokolom za pozivanje koji su usvojile sve telefonske kompanije.

Sta zapravo znači biti na Internetu? Prema našoj definiciji, računar je na Internetu ako izvršava skup protokola TCP/IP, ima IP adresu i može da šalje IP pakete svim drugim računarima na Internetu. Sposobnost da se šalje i prima elektronska pošta nije dovoljna, pošto se elektronska pošta preko mrežnih prolaza usmerava i na mnoge mreže izvan Interneta. Međutim, ova stvar nije dovoljno jasna zbog činjenice da mi- lioni personalnih računara mogu da pomoću modema pozovu davaoca Internet usluga, da dobiju privremenu IP adresu i da pošalju IP pakete dragim računarima na Internetu. Ima smisla da se takvi računari smatraju delom Interneta sve dok su povezani sa usmerivačem davaoca Internet usluga.

Između 1970. i 1990. godine, postojale su četiri glavne primene Interneta i njegovih prethodnika:

1. **E-pošta** (engl. *e-mail).* Mogućnost da se sastavi, pošalje i primi elektronska poruka postojala je od prvih dana ARPANET-a i izuzetno je popularna. Mnogi dobijaju desetine poruka dnevno i smatraju elektronsku poštu glavnim sredstvom komunikacije sa spoljnim svetom, koje daleko prevazilazi mogućnosti telefonskog razgovora ili klasične pošte. Danas programi za elektronsku poštu postoje na skoro svakoj vrsti računara.
2. Diskusione grupe (engl. *newsgroups).* Specijalizovani forumi kroz koje korisnici istih interesovanja mogu da razmene poruke. Postoje hiljade diskusionih grupa, posvećenih tehničkim i drugim temama, uključujući računare, nauku, rekreaciju i politiku. Svaka diskusiona grupa ima sopstvenu etikeciju, stil i običaje, i teško onome ko ih naruši.
3. Daljinsko prijavljivanje (engl. *remote login).* Pomoću programa telnet, rlo- gin ili ssh korisnici koji se nalaze na Internetu mogu da se prijave na bilo koji drugi računar na kome imaju otvoren nalog.
4. Prenos datoteka *(engl. file transfer).* Pomoću programa FTP, korisnici Interneta mogu da kopiraju datoteke s jednog računara na drugi. Na taj način im je dostupno mnoštvo članaka, baza podataka i drugih informacija.

Sve do početka devedesetih godina, Internet su uglavnom naseljavali istraživači iz akademskih, državnih i industrijskih krugova. Sve to je izmenila jedna nova oblast primene, **WWW (World Wide Web),** dovodeći na mrežu milione novih, neakademskih korisnika. Ta oblast, koju je osmislio fizičar Tim Berners-Lee iz C'ERN-a, nije menjala suštinu mreže, već je samo olakšala njeno korišćenje. Zajedno sa Mosaicom, čitačem Weba koji je napravio Marc Andreessen iz Nacionalnog centra za superra- čunarske aplikacije u Urbani (Ilinois), WWW je omogućio da se na mrežnoj lokaciji napravi skup informativnih strana s tekstom, slikama, zvukom, čak i videom, i sa ugrađenim vezama ka drugim stranama. Pritiskajući

vezu, korisnik se odmah prebacuje na stranu na koju ona ukazuje. Na primer, mnoge kompanije imaju početnu (matičnu) stranu (engl. *home page)* sa stavkama koje ukazuju na druge strane na kojima se nalaze informacije o proizvodima, cenama, uslovima prodaje, tehničkoj podršci, vezama ka zaposlenima, berzanskim informacijama i drugim temama.

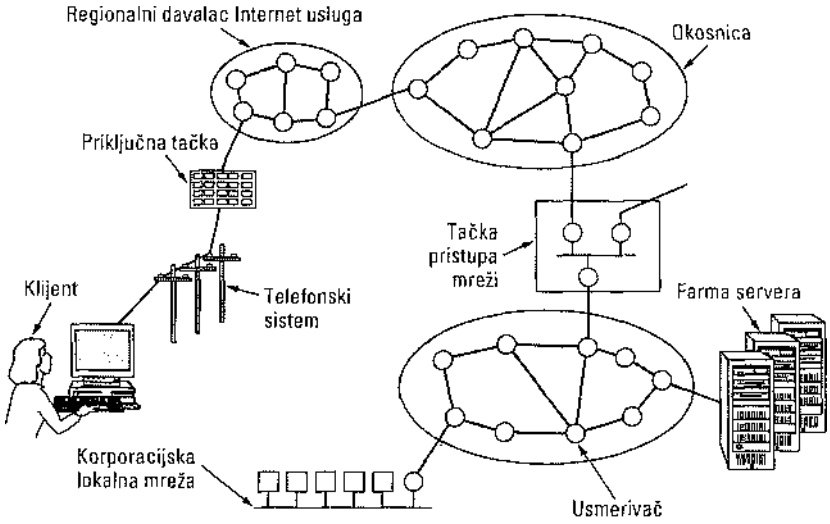
Ubrzo su se pojavile brojne druge vrste strana, kao što su mape, berzanske tabele, bibliotečki katalozi, snimljeni radio-programi, čak i strane s vezama ka potpunom tekstu mnogih knjiga čija su autorska prava istekla (Mark Tven, Čarls Dikens itd.). Mnogi pojedinci imaju i svoje lične strane na kojima se predstavljaju javnosti.

Tokom devedesetih godina, veliko učešće u razvoju Interneta imali su davaoci Internet usluga (engl. *Internet Service Providers, ISP).* To su kompanije koje privatnim licima omogućavaju da se od kuće povežu s jednim od njihovih računara i tako izađu na Internet, tj. dobijaju pristup elektronskoj pošti, Webu i drugim uslugama Interneta. Te kompanije su krajem devedesetih godina imale na desetine miliona novih pretplatnika godišnje, što je potpuno izmenilo karakter mreže - od akademskog igrališta, odnosno vojnog poligona, do javne službe, slične telefonskom sistemu. Sadašnji broj korisnika Interneta nije poznat, ali se sigurno meri stotinama miliona i verovatno će uskoro premašiti milijardu.

Arhitektura Interneta

U ovom odeljku pokušaćemo da damo kratak pregled savremenog Interneta. Brojne integracije telefonskih kompanija i davalaca Internet usluga, dovele su do prilično složene situacije, pa je često teško utvrditi šta ko radi. Zbog toga je neophodno da ovaj opis bude jednostavniji od stvarnog stanja. Opšti prikaz Interneta naći ćete na slici 1-29. Ispitajmo tu sliku, deo po deo.

Počnimo od klijenta koji se nalazi u svojoj kući. Pretpostavimo da klijent uspostavlja vezu sa svojim davaocem Internet usluga (ISP) telefonskim putem, kao na slici 1-29. Modem je posebna kartica koja digitalne signale proizvedene u računam pretvara u analogne signale koji se neometano mogu prenositi telefonom. Tako pretvoreni signali dovode se do davaočeve **priključne tačke** (engl. *Point ofPresence, POP),* gde se izvlače iz telefonskog sistema i ubacuju u davaočevu regionalnu mrežu. Od te tačke sistem je potpuno digitalan i radi uz komutiranje paketa. Ako je davalac Internet usluga lokalna telefonska kompanija, priključna tačlca će verovatno biti smeštena u centrali u kojoj se završava klijentova telefonska linija. Ako davalac nije lokalna telefonska kompanija, onda priključna tačka može da bude smeštena u nekoj daljoj telefonskoj centrali.



Slika 1-29. Opšti prikaz Interneta.

Davaočeva regionalna mreža sastoji se od međusobno povezanih usmerivača u gradovima koje davalac opslužuje. Ako je paket namenjen računaru koji je direktno vezan za davaoca, onda mu se odmah isporučuje. U suprotnom, on se prosleđuje da- vaočevom operateru okosnice.

Na vrhu hijerarhije su glavni operateri mrežnih okosnica - kompanije kao što su AT&T i Sprint. Oni održavaju velike međunarodne mrežne okosnice, s hiljadama usmerivača povezanih optičkim kablovima visoke propusne moći. Velike korporacije i uslužne kompanije koje održavaju farme servera (računare koji mogu da šalju hiljade Web strana u sekundi), često se direktno povezuju na okosnicu. Operateri okosnice ohrabruju ovakav pristup iznajmljujući prostor u tzv. **telekomunikacionim hotelima** (engl. *car- rier hotels),* U osnovi, to su ormani za opremu, koji se nalaze u istoj prostoriji sa usme- rivačem kako bi se ostvarila kratka i brza veza između farmi servera i okosnice mreže.

Ako je paket na okosnici namenjen nekom davaocu Internet usluga ili kompaniji direktno vezanoj na okosnicu, on se šalje najbližem usmerivaču i tamo isporučuje. Međutim, na svetu postoje brojne okosnice različitih veličina, pa paket može da pređe na drugu okosnicu. Da bi se omogućilo preskakanje paketa s jedne okosnice na drugu, sve glavne okosnice spajaju se u tačlcama pristupa, o kojima smo već govorili. Tačka pristupa mreži (NAP) u osnovi je prostorija prepuna usmerivača - barem po jednim za svaku okosnicu. Lokalna mreža unutar prostorije povezuje sve usmerivače, tako da se paketi mogu usmeriti s bilo koje okosnice na bilo koju drugu okosnicu. Osim što su povezane preko pristupnih tačalca, veće okosnice se međusobno povezuju direktnim vezama preko usmerivača, a ta tehnika je poznata kao **privatno povezivanje ravnopravnih usmerivača** (engl. *private peering).* Jedan od mnogih paradoksa Interneta jeste i to što davaoci Internet usluga koji se javno nadmeću za klijente, često privatno sarađuju povezujući se na ovaj način (Metz, 2001).

Ovime se završava naš kratki prikaz Interneta. O pojedinim njegovim komponentama, njihovom projektovanju, primenjenim algoritmima i protokolima imaćemo dosta da kažemo u narednim poglavljima. Treba pomenuti i to da su neke kompanije međusobno povezale sve

svoje interne mreže, često koristeći tehnologiju sličnu Internetu. Tako dobijenom **intranetu** (engl. *intranet)* najčešće se može pristupiti samo unutar kompanije, dok sve drugo radi kao Internet.

1.5.2 Mreže sa uspostavljanjem direktne veze: X.25, štafetni prenos okvira i ATM

Još od nastanka prvih mreža vodi se rat između onih koji podržavaju (datagramske) podmreže bez uspostavljanja direktne veze i onih koji se zalažu za podmreže sa uspostavljanjem direktne veze. Glavni zagovornici podmreža bez uspostavljanja direktne veze dolaze iz zajednice korisnika ARPANET-a/Interneta. Setite se da je prvobitna želja Ministarstva odbrane SAD bila da ARPANET nastavi da radi i kada pretpostavljeno nuklearno oružje ošteti i izbaci iz rada brojne usmerivače i prenosne linije. Zbog toga se otpornost na greške nalazila u vrhu njihovih prioriteta; naplaćivanje usluga korisnicima uopšte im nije bilo na umu. Takav pristup je doveo do projekta mreže bez uspostavljanja direktne veze, u kojoj se svaki paket usmerava nezavisno od drugih paketa. Kada tokom sesije otkaže neki usmerivač, to ne utiče na prenos sve dok je sistem u stanju da se automatski prilagodi situaciji šaljući pakete alternativnim putanjama, koje se mogu i razlikovati od prvobitnih.

Zagovornici mreža sa uspostavljanjem direktne' veze jesu, prirodno, telefonske kompanije. U telefonskom sistemu pozivalac mora da izabere broj sagovornika i da čeka na uspostavljanje veze pre nego što pošalje podatke. Uspostavljanje veze znači uspostavljanje putanje za prenos podataka kroz telefonski sistem, koja se održava sve dok se veza ne raskine. Sve reči ili paketi slede istu putanju. Ako linija ili skretnica (engl. *switch)* na toj putanji otkaže, veza se prekida. To je upravo ono što se Ministarstvu odbrane nije dopadalo u ovom konceptu.

Zašto se onda takav sistem dopada telefonskim kompanijama? Za to postoje dva razloga:

1. Kvalitet usluge.
2. Mogućnost naplate usluge.

Kada vezu uspostavite unapred, podmreža može da rezerviše resurse (privremenu memoriju, kapacitet procesora usmerivača itd.). Ako pokušate da uspostavite vezu bez dovoljno raspoloživih resursa, dobićete signal zauzeća. S druge strane, kada se veza uspostavi, ona obezbeđuje kvalitetnu uslugu. U mreži bez direktnog uspostavljanja veze, ako previše paketa stigne istovremeno na isti usmerivač, on će se zagušiti i verovatno izgubiti neke pakete. Pošiljalac de to odmah primetiti i ponovo ih poslati, ali de prenos paketa biti neravnomeran i nepodesan za audio ili video, osim ako je mreža sasvim slabo opterećena. Ne treba ni pominjati daje kvalitet zvuka nešto o čemu telefonske kompanije veoma brinu, pa odatle i sledi njihova sklonost ka direktnim vezama.

Telefonske kompanije vole usluge sa uspostavljanjem direktne veze i zato što uslugu najčešće naplaćuju na osnovu vremena korišćenja - kada pozovete telefonski broj u drugom gradu ili državi, meri se vreme razgovora i na osnovu njega vam se zaračunava naknada. Od samog svog nastanka, mreže su težile modelu u kome bi se olakšalo naplaćivanje prema vremenu korišćenja. Ako morate da uspostavljate vezu pre nego što pošaljete podatke, vreme teče od trenutka njenog uspostavljanja. Ako se veza ne uspostavi, to vam ne mogu naplatiti.

Paradoksalno je to što je praćenje vremenskog korišćenja veze veoma skupo. Kada bi telefonska kompanija uspostavila paušalnu mesečnu pretplatu za svoje usluge, uz neograničen broj poziva i time izbegla registrovanje i zaračunavanje pojedinačnih poziva,

verovatno bi uštedela grdan novac, uprkos porastu telefonskog saobraćaja koji bi takav potez izazvao. To se, međutim, uglavnom ne čini iz raznoraznih političkih, zakonskih i drugih razloga. Zanimljivo je da paušalno naplaćivanje usluga postoji u drugim oblastima komunikacija. Na primer, kablovska televizija se naplaćuje paušalno, bez obzira na to koliko programa dobijate. I tu je mogla biti organizovana naplata po programu, ali nije, delom i zbog skupoće takvog pristupa (a delom i zato što bi - kada se uzme u obzir kvalitet većine programa - to mnogima bilo neprijatno). Slično tome, u mnogim zabavnim parkovima možete da kupite dnevnu ulaznicu i da se tokom čitavog dana provodite kako i koliko znate i umete, dok na vašarištima koja se sele iz mesta u mesto morate posebno da platite svaku atrakciju: vožnju na autodromu, ringišpilu itd.

Posle svega ne čudi što sve mreže koje su projektovale telefonske kompanije rade sa uspostavljanjem direktne veze. Iznenađuje, međutim, to što se i Internet kreće u tom smeru, želeći da obezbedi kvalitetnije audio i video usluge, nešto o čemu ćemo govoriti u 5. poglavlju. Zasad ćemo se zadržati na nekoliko konkretnih mreža koje rade sa uspostavljanjem direktne veze.

X.25 i štafetni prenos okvira

Prvi primer mreže sa uspostavljanjem direktne veze jeste X.25 - prva javna mreža. Ona je puštena u rad sedamdesetih godina, u vreme kada je telefonija svuda imala monopol i kada su telefonske kompanije očekivale da svaka država ima po jednu mrežu za prenos podataka - njihovu. Za rad u mreži X.25 računar prvo mora da uspostavi vezu s drugim računarom, tj. da „okrene“ njegov telefon. Toj vezi se pridružuje broj veze koji se koristi pri prenosu paketa podataka (jer istovremeno može da bude uspostavljeno više veza). Paketi podataka su veoma jednostavni - sadrže zaglavlje od 3 bajta i najviše 128 bajtova podataka. Zaglavlje se sastoji od 12-bitnog broja veze (engl. *connection number),* rednog broja paketa (engl. *packet sequence number),* broja za potvrđivanje (engl. *acknowledgement number)* i nekoliko drugih bitova različite namene. Mreže X.25 korišc'ene su tokom desetak godina s različitim uspehom.

Osamdesetih godina, mreže X.25 skoro su u potpunosti zamenjene mrežama s tzv. štafetnim prenosom okvira (engl. *frame relay).* Njihova osnovna karakteristika je da rade sa uspostavljanjem direktne veze i da u njima ne postoji kontrola grešaka, niti upravljanje tokom podataka. Pošto se uspostavlja direktna veza, paketi se isporučuju strogim redosledom (ako se uopšte isporučuju). Održavanje redosleda isporučenih paketa, nepostojanje kontrole grešaka i upravljanja tokom podataka, čine štafetni pre- nos okvira sličnim lokalnoj mreži šireg područja. Najvažniju primenu štafetni prenos okvira našao je u povezivanju lokalnih mreža koje se nalaze u različitim poslovnim prostorijama iste kompanije. Štafetni prenos okvira postigao je umeren uspeh, a i danas se ponegde koristi.

Režim asinhronog prenosa

Jedna druga, mnogo važnija mreža u kojoj se veza direktno uspostavlja jeste mreža koja radi u asinhronom režimu prenosa (engl. *Asynchronous Transfer Mode, ATM),* tzv. ATM mreža. Ovakvo neobično ime dobila je da bi se istakla suprotnost načinu rada telefonskog sistema, koji je uglavnom sinhron (tesno vezan za otkucaje časovnika).

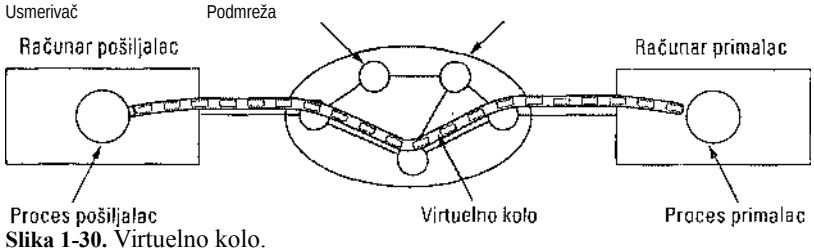
ATM mreža je projektovana početkom devedesetih godina i lansirana uz veliku pompu (Ginsburg, 1996; Goralski, 1995; Ibe, 1997; Kim et al,, 1994; Stallings, 2000). Njenom pojavom trebalo je da nestanu svi svetski mrežni i telekomunikacioni problemi tako što će se

govor, podaci, kablovska televizija, teleks, telegraf, golubovi pismonoše, konzerve povezane kanapom, tam-tamovi, dimni signali i sve ostalo integrisati u jedinstven sistem koji može da uradi bilo šta za bilo koga. Naravno, to se nije dogodilo, velikim delom zbog problema koje smo opisali u vezi s modelom OSI (loš trenutak, loša tehnologija, loša realizacija i loša politika). Upravo pobedivši telefonske kompanije u prvoj rundi, mnogi zagovornici Interneta shvatili su ATM mrežu kao nov izazov i dočekali je „na nož“. Pokazalo se da to ipak nije tako, a i zagriženi poklonici datagramskih usluga morali su da priznaju da mnogo štošta nedostaje kvalitetu usluga na Internetu. Ukratko, ATM mreža je postigla mnogo veći uspeh od modela OSI, a danas radi u telefonskim sistemima, često za prenos IP paketa. Pošto je telefonske centrale uglavnom koriste za interni prenos podataka, za korisnike je najčešće nevidljiva, ali je definitivno živa i radi.

Virtuelna ATM kola

Pošto ATM mreže rade sa uspostavljanjem direktne veze, da biste poslali podatke, morate prvo da pošaljete paket za uspostavljanje veze. Dok se paket probija kroz podmrežu, svi usinerivači na njegovom putu beleže u svoje interne tabele uspostavljenu vezu i rezervišu resurse koji su joj potrebni. Veze se često zovu virtuelna kola (engl. *virtual circuits),* po analogiji s fizičkim kolima u telefonskom sistemu. Većina ATM mreža podržava i trajna virtuelna kola (engl. *permanent virtual circuits),* lcoja

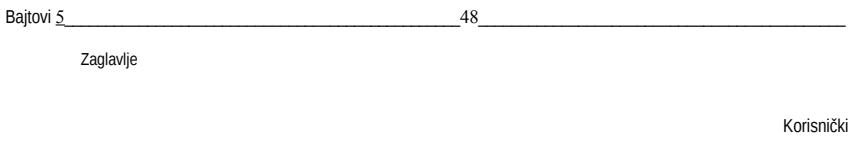
trajno povezuju dva udaljena računara. Ona liče na iznajmljene telefonske linije. Svaka veza, privremena ili trajna, ima jedinstven identifikator. Virtuelno kolo je prikazano na slici 1-30.



Slika 1-30. Virtuelno kolo.

Pošto se veza uspostavi, svaka od dve strane može početi da šalje podatke. U ATM mrežama osnovni princip je da se podaci šalju u malim paketima fiksne veličine, zvanim ćelije (engl. *cells).* Ćelije su dužine 53 bajta (.5 bajtova zaglavlja, 48 bajtova podataka), kao na slici 1-31. Deo zaglavlja zauzima identifikator veze, tako da računar pošiljalac, računar primalac i svi usmerivači na putu mogu svaki paket da pridruže odgovarajućoj vezi. Taj podatak omogućava svakom usmerivaču da odredi kako će usme- riti pridošlu ćeliju. Ćelije se usmeravaju hardverski, velikom brzinom. U stvari, glavni razlog za izbor ćelija fiksne veličine bio je taj što se hardverski usmerivači za obradu kratkih ćelija iste dužine prave lako. IP paketi promenljive dužine moraju se usmeravati softverskim putem, što je sporije. Prednost ATM mreže je i to što se pridošla ćelija može hardverski kopirati na više izlaznih linija - svojstvo koje je neophodno pri difuznom emitovanju TV programa mnogim korisnicima. Naglasimo i to da male ćelije ne mogu da blokiraju liniju tokom dužeg vremena, što olakšava garantovanje kvaliteta usluge.

Sve ćelije slede istu putanju do odredišta. Isporuka ćelija nije garantovana, ali njihov redosled jeste. Ako se ćelije 1 i 2 pošalju tim redom, one će tim redom stići i na odredište, pod uslovom da stignu obe. Međutim, na putu se može izgubiti jedna od njih ili i jedna i druga. U takvim slučajevima, postupak se određuje protokolom višeg nivoa. Treba primetiti da ovde ipak postoji nekakva garancija. Na Internetu paketi mogu ne samo da se izgube, već i da na odredište stignu pogrešnim redosledom, dok ATM uvek. garantuje ispravan redosled pristizanja.



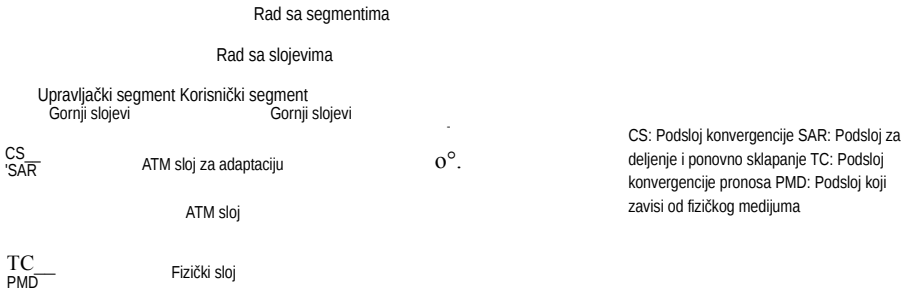
Slika 1-31. ATM ćelija.

ATM mreže se organizuju kao klasične regionalne mreže, sa svojim linijama i skret- nicama (usmerivačima). Najčešća brzina prenosa u ATM mrežama iznosi 1.5.5 Mb/s ili 622 Mb/s, ali su podržane i veće brzine. Manja od dve navedene brzine izabrana je kao

minimum za prenos televizijskog signala visoke rezolucije. Njena tačna vrednost je 155,52 Mb/s zbog kompatibilnosti s prenosnim sistemom SONET korporacije AT&T, o čemu ćemo govoriti u 2. poglavlju. Veća brzina (622 Mb/s) omogućava da se ista linija iskoristi za četiri kanala brzine 155 Mb/s.

Referentni model ATM

ATM ima sopstveni model koji se razlikuje i od modela OSI i od modela TCP/IP. On je prikazan na slici 1-32. Sastoji se od tri sloja, fizičkog, ATM sloja i ATM sloja za adaptaciju, a korisnik iznad njih može da doda šta god mu treba.



Slika 1-32. Referentni model ATM.

Fizički sloj radi s fizičkim medijumom: naponima, sinhronizovanjem bitova i sličnim stvarima. ATM ne propisuje određen skup pravila, već samo nalaže da se ćelije kao takve mogu slati žicom ili optičkim kablom, ali i da se mogu pakovati kao koristan teret (engl. *payload)* dragih sistema prenosa. Drugim recima, ATM je pro- jektovan da bude nezavistan od prenosnog medijuma.

ATM sloj (engl. *ATM layer)* bavi se ćelijama i njihovim prenosom. On definiše organizaciju ćelije i daje značenje poljima zaglavlja. ATM sloj takođe uspostavlja i raskida virtuelna kola, a upravlja i zagušenjima saobraćaja na mreži.

Pošto većina aplikacija ne radi neposredno sa ćelijama (premda neke to mogu), definisan je sloj iznad ATM sloja koji korisnicima omogućava da šalju pakete veće od ćelija. ATM interfejs deli te pakete, prenosi pojedinačne ćelije i ponovo sklapa pakete na dragom kraju. To je ATM sloj za adaptaciju (engl. *ATM Adaptation Layer, AAL).*

Za razliku od prethodnih, dvodimenzionalnih referentnih modela, model ATM je trodimenzionalan (slika 1-32). Korisnički segment (engl. *user plane)* prenosi podatke, upravlja tokom, ispravlja greške i izvršava drage korisničke funkcije. Upravljački segment (engl. *control plane)* upravlja vezom. Funkcije rada sa slojevima i segmentima namenjene su upravljanju resursima i koordinaciji između slojeva.

I fizički i AAL sloj imaju dva podsloja: donji, koji obavlja stvarni posao, i gornji - podsloj konvergencije - koji predstavlja odgovarajući interfejs ka sloju iznad sebe. Funkcije

pojedinih slojeva i podslojeva prikazane su na slici 1-33.

Sloj po modelu OSI	ATM sloj	ATM podsloj	Namena
3/4	AAL	CS	Obezbeđuje standardni interfejs (konvergenciju)
		SAR	Deljenje i ponovno sklapanje
2/3	ATM		Kontrola toka Generisanje/uklanjanje zaglavlja ćelije Rad s virtuelnim kolom/putanjom Multipleksiranje/demultipleksiranje delije
2	Fizički	TC	Regulisanje brzine slanja ćelija Generisanje kontrolnog zbira u zaglavlju i njegova provera Generisanje ćelija Pakovanje/raspakivanje ćelija iz omotnice Generisanje okvira
1		PMD	Sinhronizovanje bitova Pristupanje fizičkoj mreži

Slika 1-33. Slojevi i podslojevi modela ATM i njihove funkcije.

Podsloj koji zavisi od fizičkog medij uma (engl. *Physical Medium Dependent, PMD)* predstavlja interfejs ka kablu. On šalje i prihvata bitove iz kabla, obezbeđujući odgovarajuće sinhronizovanje operacija. Ovaj podsloj će se razlikovati za različite nosioce podataka i različite kablove.

Dnigi podsloj fizičkog sloja je **podsloj konvergencije prenosa** (engl. *Transmission Convergence, TC).* Tokom prenosa ćelija, TC ih šalje PMD podsloju kao tok bitova, što je jednostavno. S druge strane, i on ih u tom obliku prima od PMD podsloja. Njegov zadatak je da primljeni tok bitova pretvori u tok ćelija pre nego što ih prosledi ATM sloju, što znači da u toku bitova propisno obeležava početak i kraj svake ćelije. U modelu ATM, ovu funkciju ima fizički sloj. U modelu OSI i u prilično mnogo drugih mreža, uokviravanje (engl. *framing),* tj. pretvaranje toka bitova u niz ćelija ili okvira, predstavlja zadatak sloja veze podataka.

Kao što smo već pomenuli, ATM sloj radi sa ćelijama, uključujući njihovo generisanje i prenos. Većina zanimljivih aspekata modela ATM smeštena je u njemu. Taj sloj je mešavina sloja veze i mrežnog sloja modela OST, on nije izdeljen na podslojeve.

AAL sloj ima dva podsloja: podsloj za **deljenje i ponovno sklapanje** (engl. *Segmentation And Reassembly, SAR)* i **podsloj konvergencije** (engl. *Convergence Sublayer, CS).* Donji podsloj deli pakete u ćelije pri slanju i ponovo ih sklapa na odredištu. Gornji podsloj omogućava ATM sistemima da različlitim aplikacijama ponude različite usluge (npr. prenos datoteka i video na zahtev imaju drugačije zahteve u pogledu obrade grešaka, sinhronizacije i slično).

Pošto ATM sistem polako nestaje, nećemo ga više razmatrati u ovoj knjizi. Pa ipak, budući da su instalirani mnogi takvi sistemi, verovatno ćemo ga gledati još nekoliko godina. Više detalja o ATM mrežama potražite kod Dobrovvskog i Grisea
(2001) i kod Gadeckog i Heckarta (1997).

1.5.3 Ethernet

I Internet i ATM projektovani su za rad u regionalnim mrežama. Međutim, mnoge kompanije, univerziteti i druge organizacije imaju mnoštvo računara koje međusobno treba povezati. Iz te potrebe izrasle su lokalne računarske mreže. U ovom odeljku go- vorićemo nešto o najpopularnijoj lokalnoj mreži, Ethernetu.

Priča počinje na devičanskim Havajima početkom sedamdesetih godina. Izraz „devičanski“ u ovom kontekstu znači „bez telefonskog sistema“. Iako zasluženi odmor turista ne prekida zvrjanje telefona, istraživaču Normanu Abramsonu i njegovim kolegama s Havajskog univerziteta, nepostojanje telefonskog sistema samo je zagorčavalo život dok su pokušavali da povežu korisnike na udaljenim ostrvima s glavnim računarom u Honoluluu. Nije dolazilo u obzir da rastežu sopstvene kablove ispod Pacifika, pa su zato tražili drugačije rešenje.

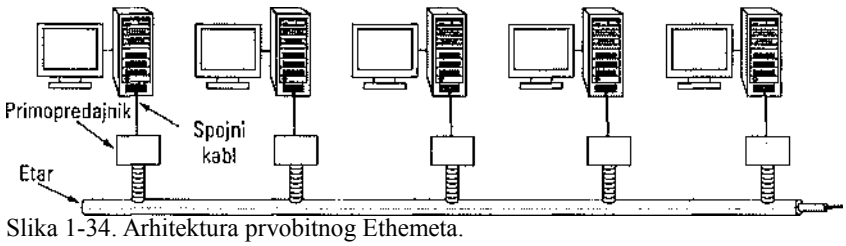
Našli su ga u radiju kratkog dometa. Svaki korisnički terminal opremili su primo- predajnikom sa samo dve frekvencije: jednom za emitovanje ka centralnom računam i drugom - za prijem podataka sa centralnog računara. Kada je korisnik želeo da pristupi računam, jednostavno je slao paket podataka prvom frekvencijom. Ako niko drugi u tom trenutku nije emitovao, paket je nalazio svoj cilj i o tome je korisnik do- bijao potvrdu drugom pomenutom frekvencijom. Ako je bilo sukobljavanja na kanalu za slanje podataka, korisnik ne bi dobio potvrdu o prijemu paketa i slao bi ga ponovo. Pošto je na strani centralnog računara bio samo jedan pošiljalac, na kanalu kojim je on slao poruke nije bilo sukobljavanja. Opisani sistem, nazvan ALOHANET, radio je prilično dobro u uslovima retkog saobraćaja, ali se zaglavljivao kada je saobraćaj ka centralnom računara bio gust.

Nekako u isto vreme, na Masačusetskom tehničkom institutu (MIT) u Kembridžu diplomirao je student Bob Metcalfe i zatim prešao na drugu stranu Čarlsove reke da bi radio doktorat na Harvardu. Tokom studija se upoznao sa Abramsonovim radom i tako zainteresovao za njega daje posle doktoriranja na Harvardu odlučio da provede mesec dana na Havajima radeći sa Abramsonom, pre nego što stupi na nov posao u Istraživačkom centru u Palo Altu (Xeroxov PARC). Kada je stigao u PARC, utvrdio je da su tamošnji istraživači projektovali i izgradili ono što će kasnije generacije zvati personalni računar. Računari su, međutim, bili izolovani. Koristeći znanje stečeno u radu sa Abramsonom, on je zajedno s kolegom Davidom Boggsom projektovao i rea- lizovao prvu lokalnu računarsku mrežu (Metcalfe i Boggs, 1976).

Sistem su nazvali Ethernet prema *luminifernom etru* - nosiocu svetlosti, za koji se nekada smatralo da prenosi elektromagnetno zračenje. (Kada je britanski fizičar Džerns Klerk Malcsvel u 19. velcu otkrio da se elektromagnetno zračenje može opisati talasnom jednačinom, naučnici su zaključili da svemir mora biti ispunjen nekim eteričnim medijumom kroz koji se prostiru talasi. Tek nakon čuvenog Majkelson-Mor- lijevog eksperimenta 1887. godine, fizičari su shvatili da se elektromagnetno zračenje može prostirati kroz vakuum.)

Kod Ethemeta, prenosni medijum nije bio vakuum, već debeo koaksijalni kabl („etar“), dugačak do 2,5 km (s repetitorima na svakih 500 m). Pomoću primopredajni- ka ugrađenih u kabl, na njega se moglo povezati do 256 računara. Kabl na koji je paralelno povezano više računara naziva se kabl s više priključaka (engl. *multidrop cable).* Sistem je radio s brzinom prenosa 2,94 Mb/s. Skica njegove arhitekture prikazana je na slici 1-34. Ethernet je u odnosu na ALOHANET imao brojna poboljšanja: pre slanja, računar je najpre osluškivao kabl. Ako bi otkrio da neko već emituje, povlačio bi se dok se tekuće emitovanje ne okonča. Na taj način je izbegavano sukobljavanje, što je doprinelo efikasnosti. Sistem ALOHANET nije

mogao da radi tako jer nije bilo načina da terminal najednom ostrvu otkrije emitovanje terminala na drugom ostrvu. Na jedinstvenom kablu takav problem nije postojao.



Slika 1-34. Arhitektura prvobitnog Ethemeta.

Uprkos tome što su računari osluškivali pre emitovanja, postojao je još jedan problem: šta ako dva ili više računara istovremeno očekuju da se završi tekuće emitovanje, a onda istovremeno krenu na mrežu? Rešenje je nađeno u tome da svaki računar pri pokušaju emitovanja istovremeno osluškuje kabl; ako otkrije mešanje, upozorava sve pošiljaoce i povlači se, a zatim - posle proizvoljnog vremenskog intervala - ponovo pokušava da emituje. Ako i drugi put dođe do sukobljavanja, proizvoljni vremenski interval se udvaja i tako redom, da bi se konkurentne emisije vremenski razdvojile i dala šansa jednom računaru da prvi emituje.

Xeroxov Ethernet bio je tako uspešan da su DEC, Intel i Xerox 1978. uspostavili standard za Ethernet brzine prenosa 10 Mb/s, pod imenom standard DIX. Uz dve manje izmene, standard DIX je 1983. postao standard IEEE 802.3.

Firma Xerox je, nažalost, već imala dugu istoriju sopstvenih pronalazaka (npr. personalnog računara) koje nije uspela da komercijalizuje - priča opisana u knjizi *Propuštanje budućnosti (Fumbling the Future,* Smith i Alexander, 1988). Kada je firma Xerox, osim što je pomogla da se standardizuje, konačno pokazala malo više zanimanja za Ethernet, Metcalfe je već bio osnovao sopstvenu kompaniju, 3Com, za prodaju mrežnih Ethernet kartica za personalne računare. Kompanija je do sada prodala preko sto miliona takvih kartica.

Ethernet je nastavio da se razvija i to čini i danas. Pojavile su se njegove nove verzije s brzinom prenosa 100 Mb/s, 1000 Mb/s i brže. Kabliranje je takođe poboljšano, a dodato je komutiranje i štošta drugo. Ethernet ćemo detaljno obraditi u 4. poglavlju.

Kada već govorimo o tome, treba naglasiti da Ethernet (IEEE 802.3) nije jedini standard za lokalne mreže. Komitet je standardizovao i mreže token bus (802.4) i token ring (802.5). Pojava tri manje-više nekompatibilna standarda nema ništa s tehnologijom, već pre s politikom. U vreme standardizacije, General Motors je gurao lokalnu mrežu koja je imala istu topologiju kao i Ethernet (linearan kabl), samo što su se računali uzastopno smenjivali u emitovanju prosleđujući jedan drugom mali paket, tzv. žeton (engl. *token).* Računar je mogao da emituje samo kada poseduje žeton - na taj način je izbeg- nuto sukobljavanje. General Motors je izjavio daje takva šema neophodna u proizvodnji automobila i nije hteo da odstupi. Bez obzira na tu izjavu, standard 802.4 praktično je nestao.

Slično tome, i IBM je imao svog pulena: sopstveni sistem token ring. Žeton je kružio prstenom i računar kod koga se trenutno našao mogao je da emituje pre nego što vrati žeton u prsten. Za razliku od standarda 802.4, ova šema - standardizovana pod oznakom 802.5 - još uvek se koristi na nekim IBM-ovim lokacijama, ali nigde izvan njih. Rad se nastavlja na gigabitnoj verziji (802.5v), ali ne izgleda verovatno da će ona ikada dostići Ethernet.

Ukratko, vodio se rat između Etherneta, token busa i token ringa, i Ethernet je izišao kao pobednik, najviše zato što je prvi rešio problem i zato što izazivači nisu bili tako dobri.

1.5.4 Bežični LAN: 802.11

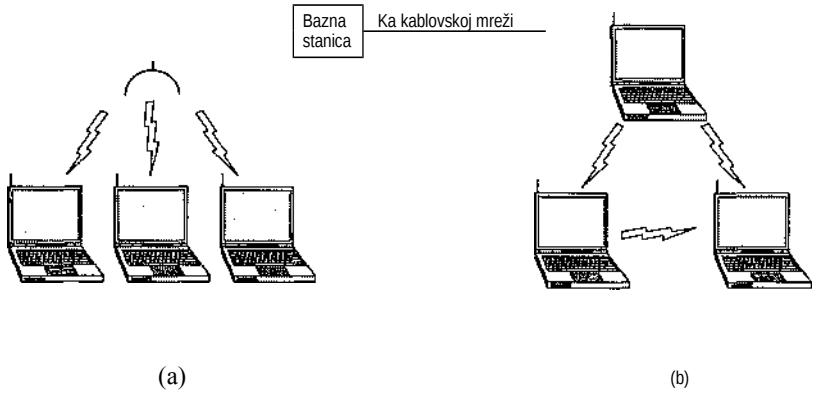
Čim su se pojavili prenosni računari, mnogi su počeli da sanjaju o tome da jednostavno uhodaju u neku kancelariju i priključe ga na Internet. Zbog toga su brojne grupe počele da rade na ostvarenju tog cilja. Najpraktičnije rešenje bilo je da se i kancelarija i prenosni računar opreme radio-predajnicima kratkog dometa pomoću kojih bi mogli da komuniciraju. Taj pristup je ubrzo doveo do stvaranja bežičnih lokalnih mreža koje su nudile mnoge kompanije.

Problem je bio u tome što se među njima nisu mogle naći ni dve međusobno kompatibilne mreže. Takvo zanemarivanje standarda značilo je da računar s radio-predaj- nikom marke X neće raditi u prostoriji u kojoj je instalirana bazna stanica marke Y. Na kraju je industrija zaključila da bi standard za bežične lokalne mreže mogao biti dobra ideja, pa je IEEE komitetu koji je standardizovao ožičenu lokalnu mrežu dat zadatak da napravi standard i za bežični LAN. Standard koji je ovaj institut predložio dobio je oznaku 802.11, ali je odmah stekao i popularno ime **WiFi.** Pošto je standard važan i zaslužuje poštovanje, zvaćemo ga pravim imenom, 802.11.

Predloženi standard je predvideo dva radna režima:

1. U prisustvu bazne stanice.
2. U odsustvu bazne stanice.

U prvom slučaju, sva komunikacija treba da se odvija preko bazne stanice, u terminologiji standarda 802.11 zvane **pristupna tačka** (engl. *access point)*. U drugom slučaju, računari bi uspostavljali direktnu međusobnu vezu. Takav režim rada se ponekad naziva **ad hoc umrežavanje** (engl. *ad hoc networking).* Tipičan primer je direktna komunikacija između računara dva ili više korisnika u prostoriji koja nema bežičnu lokalnu mrežu. Radni režimi su prikazani na slici 1-35.



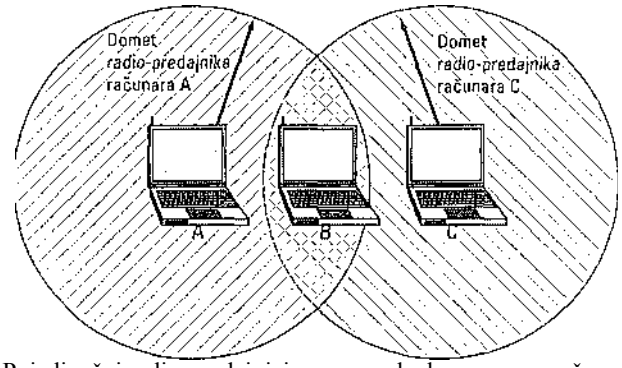
Slika 1-35. (a) Rad u bežičnoj mreži s baznom stanicom, (b) Ad hoc umrežavanje.

Najlakše je bilo doneti prvu odluku: ime standarda. Svi drugi standardi za lokalne mreže nosili su oznake 802.1, 802.2, 802.3 i tako sve do 802.10, pa je bilo prirodno da standard za bežični LAN dobije oznaku 802.11. Ostatak posla bio je teži.

Trebalo je resiti mnoge probleme: pronaći pogodnu frekvenciju - po mogućstvu, globalno raspoloživu; uvažiti činjenicu da radio-talasi imaju ograničen domet; obez- bediti privatnost korisnika; uzeti u obzir ograničen kapacitet baterija; misliti o bez- bednosti korisnika (da li radio-talasi izazivaju rak?); razumeti sve implikacije mobilnosti računara; i konačno, izgraditi sistem dovoljne propusne moći da bude ekonomski prihvatljiv.

U vreme kada je započet proces standardizacije (sredina devedesetih), Ethernet je već dominirao područjem lokalnih mreža, pa je IEEE komitet odlučio da standard 802.11 učini kompatibilnim s Ethernetom iznad sloja veze podataka. Naročito je trebalo omogućiti da se IP paket preko bežične lokalne mreže šalje na isti način kao i preko Etherneta. Pored toga, u fizičkom sloju i sloju veze podataka postoji više razlika u odnosu na Ethernet koje je standardom trebalo razrešiti.

Prvo, računar na Ethernetu uvek osluškuje kabl pre nego što počne da emituje. On počinje da emituje tek kada utvrdi da na kablu nema nikoga. U bežičnim mrežama, takav pristup se ne ostvaruje lako. Objašnjenje pruža slika 1-36. Pretpostavimo da računar A šalje poruku računaru B, ali je domet radio-predajnika računara A suviše mali da dosegne računar C. Ukoliko računar C želi da pošalje poruku računaru B, on može da osluškuje etar pre slanja ali, ako ništa ne čuje, to ne znači da će uspešno poslati poruku. Standard 802.11 morao je da razreši ovaj problem.

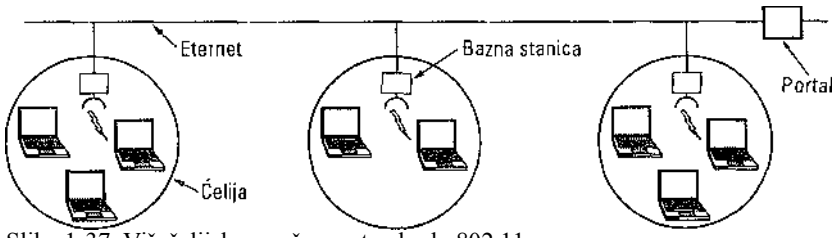


Slika 1-36. Pojedinačni radio-predajnici ne mogu da dosegnu sve računare u sistemu.

Dragi problem je to što se radio-signal može odbijati od čvrstih prepreka, tako da prijemnik može isti signal primiti više puta (direktan i odbijen različitim putanjama). Rezultujuća interferencija signala naziva se slabljenje zbog različitih putanja (engl. *multipath fading)*.

Treći problem nastaje zato što veliki deo softvera nije svestan pokretljivosti raču- nara. Na primer, mnogi programi za obradu teksta imaju listu iz koje korisnik bira štampač da bi odštampao datoteku. Kada se računar s takvom listom premesti u drugo okruženje, ugrađena lista štampača postaje beskorisna.

Četvrti problem nastaje kada prenosivi računar premestite iz dometa jedne bazne stanice u domet drage stanice; mora postojati mehanizam kojim jedna bazna stanica predaje upravljanje računarom dragoj stanici. Mada se isti problem javlja s mobilnim telefonima, ne javlja se u Ethernet mrežama i trebalo gaje resiti. Konkretno, zamišljena mreža imala je više ćelija, svaka sa svojom baznom stanicom, pri čemu su baze povezane Ethernet mrežom (slika 1-37). Sistem je spolja izgledao kao i svaki dragi Ethernet. Veza sistema 802.11 sa spoljnim svetom nazvana je portal.



Slika 1-37. Višečelijska mreža po standardu 802.11.

Komitet je 1997. objavio standard koji je rešavao pomenute i druge probleme. Predviđena brzina rada bežičnog LAN-a bila je 1 ili 2 Mb/s, na šta su odmah usledile žalbe korisnika i rad je nastavljen u cilju ubrzavanja rada u mreži. U komitetu je nastao

rascep, zbog čega su 1999. objavljena dva nova standarda. Standard 802.11a predviđa širi frekventni opseg i rad pri brzinama do 54 Mb/s. Standard 802.11b koristi istu frekvenciju kao i osnovni standard 802.11, ali uz drugačiju tehniku modulacije postiže brzinu 11 Mb/s. Neki smatraju daje to psihološki važno jer je postignuta brzina (11 Mb/s) veća nego u kablovskom Ethernetu. Jasno je da će prvobitni standard 802.11, koji omogućava brzinu 1 Mb/s, nestati, ali nije jasno šta će ga naslediti.

Da bi sve još više zapetljao, komitet je objavio i dodatnu varijantu, 802.1 lg, koja koristi tehniku modulacije iz standarda 802.11a, i opseg frekvencija iz standarda 802.11b. Standard 802.11 obradićemo detaljno u 4. poglavlju.

Da će standard 802.11 izazvati revoluciju u računarstvu i Internetu, to je sada bez svake sumnje. Bežične mreže se ubrzano instaliraju na aerodromima, železničkim stanicama, u hotelima, tržnim centrima i na univerzitetima. Čak i mondenski kafići instaliraju mreže 802.11 da bi japiji mogli da krstare Webom dok ispijaju svoju jutarnju kafu s mlekom. Izgleda da će bežične mreže učiniti za Internet ono što su prenosivi računari učinili za računarstvo: učiniće ga dostupnim sa svakog mesta.

1.6 STANDARDIZOVANJE MREŽA

Postoji mnogo proizvođača i prodavača mreža, a svaki ima sopstveni stav o tome kako stvari treba da izgledaju. Bez koordinacije nastao bi potpun haos, od koga bi korisnici imali samo štetu. Jedini izlaz je dogovaranje o nekakvim mrežnim standardima.

Standardi ne samo što omogućavaju da različiti računari međusobno komuniciraju, već i prošiniju tržište proizvoda koji su usaglašeni sa standardom. Šire tržište znači ma- sovniju i ekonomičniju proizvodnju, primenu tehnologije VLSI i druge prednosti koje snižavaju cenu i povećavaju prihvatljivost proizvoda. U narednim odeljcima bacićemo kratak pogled na važno, ali slabo poznato područje međunarodne standardizacije.

Standardi se dele u dve kategorije: de facto i de jure. De facto (lat. *činjenički, faktički)* standardi su oni koji su prihvaćeni bez prethodnog planiranja. IBM PC i njegovi naslednici predstavljaju de facto standarde računara za male firme i kućnu upotrebu zato što su desetine proizvođača vrlo precizno „iskopirali“ IBM-ove proizvode. Slično tome, UNIX je de facto standard za operativne sisteme u univerzitetskim računarskim centrima.

Nasuprot tome, de jure (lat. *zakonski)* standardi predstavljaju formalne, zakonske standarde koje je objavilo neko telo ovlašćeno za standardizaciju. Organizacije ovla- šćene za međunarodne standarde u načelu se dele na dve klase: one koje su osnovane sporazumom više država i one dobrovoljne, izvan sporazuma. Na području standardizacije računarskih mreža ima više organizacija oba tipa i o njima govorimo u nastavku.

1.6.1 Ko je ko u svetu telekomunikacija

Zakonski status telefonskih kompanija u svetu razlikuje se od jedne zemlje do dmge. Jedna krajnost su Sjedinjene Države sa 1500 zasebnih, privatnih telefonskih kompanija. Scenom je do svog raspada, 1984, dominirao AT&T, tada najveća svetska korporacija. Ona je obezbeđivala telefonske usluge za 80 posto američkih telefona i prostirala se na više od pola Amerike, dok su sve ostale kompanije zajedno zadovoljavale preostalih 20 odsto potreba

(većinom seoskog) stanovništva. Posle rasprada korporacije, AT&T je nastavila da održava međumesni saobraćaj, mada u konkurenciji s drugim kompanijama. Sedam regionalnih Bellovih telefonskih centrala koje su se odvojile od korporacije AT&T, kao i brojne druge nezavisne kompanije, obezbeđuju lokalne telefonske usluge i usluge mobilne telefonije. Zahvaljujući integracijama i drugim promenama, stanje u ovoj oblasti neprestano se menja.

Kompanije u SAD koje pružaju komunikacione usluge javnom sektora nazivaju se **javne telekomunikacione službe** (engl. *common carriers).* Njihova ponuda i cenov- nici navode se u dokumentu zvanom **tarifa** (engl. *tariff),* koji mora da odobri Savezna komisija za komunikacije kada se radi o međudržavnom (unutar SAD) i međunarodnom saobraćaju, a Državna komisija kada se radi o saobraćaju unutar pojedinih država (unutar SAD).

Draga krajnost su države u kojima vlada ima potpun monopol nad komunikacijama, uključujući poštu, telegraf, telefon, a često i radio i TV. Veći deo sveta spada u ovu kategoriju. U nekim slučajevima, telekomunikacije drži nacionalizovana kompanija, a u dragim je to jednostavno vladin sektor, obično poznat kao **Poštanska, telegrafska i telefonska uprava** (engl. *Post, Telegraph & Téléphoné administration, PTT*). U čitavom svetu oseća se trend ka liberalizaciji, konkurenciji i slamanju državnog monopola na ovom polju. Većina evropskih država već je (delom) privatizovala svoje PTT uprave, ali je negde taj proces tek počeo.

Uz toliko različitih davalaca usluga, postoji jasna potreba za kompatibilnošću na globalnom planu da bi ljudi (ili računari) u jednoj državi mogli da komuniciraju sa svojim parnjacima u dragoj. Takva potreba, u stvari, postoji odavno. Godine 1865, predstavnici mnogih evropskih vlada sastali su se da osnuju pretka današnjeg **Međunarodnog saveza za telekomunikacije** (engl. *International Télécommunication Union, ITU).* Zadatak saveza bio je da standardizuje međunarodne telekomunikacije, što je u ono doba značilo telegrafiju. Čak i onda je bilo jasno da će nastati problemi ako polovina zemalja koristi Morzeovu azbuku, a druga polovina neki drugi kôd. Kada je telefonski saobraćaj prešao granice pojedinih zemalja, ITU je preuzeo da standardizuje i telefoniju. Godine 1947, ITU je postao agencija Ujedinjenih Nacija.

ITU ima tri glavna sektora:

1. Sektor radiokomunikacija (ITU-R).
2. Sektor za standardizovanje telekomunikacija (ITU-T).
3. Sektor za razvoj (ITU-D).

Sektor ITU-R širom sveta dodeljuje frekventna područja konkurentskim zainteresovanim stranama. Mi ćemo pretežno govoriti o sektora ITU-T, u čiji đelokiug spadaju telefonski sistemi i sistemi prenosa podataka. Između 1956. i 1993. godine ITU-T je bio poznat kao **CCTTT,** što je skraćenica od francuskog imena Comité Consultatif International Télégraphique et Téléphonique. Prvog marta 1993, CCITT se re- organizovao, postao je manje birokratski i uzeo novo ime da označi svoju novu ulogu.

IITU-T i CCITT su davali preporuke na području telefonije i sistema prenosa podataka. I sada ćete često naleteti na preporuke organizacije CCITT, npr. na preporuku CCITT X.25, iako od 1993. preporuke nose oznaku ITU-T.

Sektor ITU-T ima četiri vrste članova:

1. Državne vlade.
2. Članove sektora.

3. Pridružene članove.
4. Agencije odgovorne za propise.

ITU-T ima oko 200 predstavnika vlada, uključujući skoro svakog člana Ujedinjenih Nacija. Pošto Sjedinjene Države nemaju PTT, neko drugi je morao da ih predstavlja u sektoru ITU-T. Taj zadatak je dodeljen Ministarstvu inostranih poslova, verovatno zato što se sektor ITU-T bavi stranim zemljama, a to je specijalnost ovog ministarstva. Postoji oko 500 članova sektora, među kojima su telefonske kompanije (npr. AT&T, Vodafone, WorldCom), proizvođači telekomunikacione opreme (npr. Cisco, Nokia, Nortel), proizvođači računara (npr. Compaq, Sun, Toshiba), proizvođači čipova (npr. Intel, Motorola, TI), medijske kompanije (npr. AOLTime Warner, CBS, Sony) i drage zainteresovane kompanije (npr. Boeing, Samsung, Xerox). Različite neprofitne naučne organizacije i industrijski konzorcijumi takođe su članovi sektora (npr. EFIP i IATA). Pridruženi članovi su manje organizacije koje su zainteresovane za Studijske grupe uže tematike. U agencije odgovorne za propise spadaju oni koji nadgledaju poslovni aspekt telekomunikacija, kao što je Američka savezna komisija za komunikacije.

Zadatak sektora ITU-T jeste da daje tehničke preporuke za interfejse koji se koriste u telefoniji, telegrafiji i prenosu podataka. Te preporuke često postaju međunarodno priznati standardi, npr. V.24 (poznat u SAD i kao EIA RS-232), koji određuje raspored i ulogu kontakata priključka za većinu asinhronih terminala i spoljnih modema.

Treba naglasiti da su preporuke sektora ITU-T formalno samo preporuke, koje pojedine države - po nahođenju - mogu da prihvate ili odbace (jer se vlade ponašaju slično pubertetlijama - ne prihvataju da im se izdaju komande). To u praksi znači da država koja želi da koristi neki drugi standard to slobodno može da učini, ali po cenu izolovanja od ostalog sveta. Takav stav možda može da prođe u Severnoj Koreji, ali na svakom drugom mestu predstavljaće problem. Održavanje iluzije da su ITU-T standardi samo „preporuke“ bilo je - a i sada je - neophodno da se umire nacionalisti u mnogim zemljama.

ITU-T obavlja svoje zadatke kroz Studijske grupe, koje često broje i 400 članova. Trenutno postoji 14 takvih grupa. One pokrivaju različite oblasti, počev od naplaćivanja telefonskih usluga, do multimedije. Da bi u Studijskim grupama išta moglo da se uradi, one se dele na Radne grupe koje - sa svoje strane - obrazuju Timove stručnjaka, a ovi ad hoc grupe. Jednom birokrata - uvek birokrata.

Uprlcos svemu, ITU-T uspešno ispunjava zadatke. Od svog nastanka, ITU-T je izdao oko 3.000 preporaka na oko 60.000 stranica. Mnoge od njih su široko prihvaćene u praksi. Na primer, popularni standard V.90 za modeme brzine 56 kb/s predstavlja ITU preporuku.

Kako telekomunikacije, počev od osamdesetih godina, prolaze kroz tranziciju od isključivo nacionalnih ka potpuno globalnim, standardi postaju sve važniji, a sve je više organizacija koje izražavaju želju da učestvuju u njihovoj izradi. Više podataka o organizaciji ITU naci čete kod Irmera (1994.).

1.6.2 Ko je ko u svetu međunarodnih standarda

Međunarodne standarde piše i objavljuje **Međunarodna organizacija za standardizaciju** (engl. *International Standards Organization*, *ISO**), dobrovoljna organizacija, osnovana 1946. mimo međudržavnih sporazuma. Njeni članovi su nacionalne organizacije za

standardizovanje iz 89 zemalja. Među članovima su i ANSI (SAD), BSI (Velika Britanija), AFNOR (Francuska), DIN (Nemačka) itd.

ISO pravi standarde za sve i svašta, počev od šrafova i navrtki, do zaštitne boje za telefonske bandere [da i ne pominjemo zrna kakaovca (ISO 2451), ribarske mreže (ISO 1530), ženski donji veš (ISO 4416), kao i dosta drugih stavki za koje biste pomislili da nisu podložne standardizovanju]. Izdato je preko 13.000 standarda, uključujući i OSI standarde. ISO ima skoro 200 Tehničkih komiteta (engl. *Technical Committees, TCs*), nabrojanih redosledom njihovog nastanka, a svaki se bavi specifičnom tematikom. TC1 se bavi grafovima i navrtkama (standardizovanje koraka navoja). TC97 razmišlja o računarima i obradi informacija. Svaki TC ima potkomitete (engl, *subcommittees, SCs),* izdeljene u radne grupe (engl. *working groups, WGs).*

Stvarni posao u radnim grupama uglavnom obavlja preko 100.000 dobrovoljaca širom sveta. Mnogima od ovih „dobrovoljaca" baš je njihov poslodavac dao u zadatak da prate standarde koji se odnose na proizvode njegove firme. Dragi su državni činovnici koji žele da svoju zemlju izvedu na put međunarodne standardizacije. Akademski stručnjaci su takođe aktivni u mnogim radnim grupama.

ISO i ITU-T (ISO je član organizacije ITU-T) često sarađuju po pitanju standardi- zovanja telekomunikacija da bi se izbegao apsurd nastajanja dva zvanična, međusobno nekompatibilna međunarodna standarda.

Predstavnik SAD **u** organizaciji ISO je **Američki institut za nacionalne standarde** (engl. *American National Standards Institute, ANSI),* koji, uprkos svom imenu, predstavlja privatnu, nevladinu, neprofitnu organizaciju. Njeni članovi su proizvođači, javne telekomunikacione službe i drage zainteresovane strane. ANSI standarde organizacija ISO često prihvata kao međunarodne.

Postupale koji ISO koristi tokom prihvatanja standarda obezbeđuje najširi mogući konsenzus mišljenja. Postupak počinje tako što jedna od nacionalnih organizacija za standardizaciju oseti potrebu za međunarodnim standardom u određenoj oblasti. Tada se obrazuje radna grupa koja napravi **prednacrt standarda** (engl. *Committee Draft, CD)*. Prednacrt se prosledi svim članicama da u roku od šest meseci stave primedbe. Ako se većina članica složi, pravi se redigovan **nacrt međunarodnog standarda** (engl. *Draft International Standard, DIS)* i ponovo šalje članicama na diskusiju i

[1] Za perfekcioniste: pravo ime organizacije ISO glasi: International Organization for Standardization.

glasanje. U zavisnosti od ishoda ovog kruga, priprema se konačan tekst **Međunarodnog standarda** (engl. *International Standard, IS),* koji se zatim odobrava i objavljuje. U slučaju velikih primedaba, CD i DIS mogu da pretrpe brojne izmene pre nego što se dovoljno članica složi oko teksta, a postupak može da potraje godinama.

Nacionalni institut za standarde i tehnologiju (National Institute of Standards and Technology, NIST) deo je Američkog ministarstva trgovine. Ranije je to bio Nacionalni biro za standarizaciju. Institut objavljuje standarde koji su obavezni za nabavke Američke vlade, osim za Ministarstvo odbrane, koje ima sopstvene standarde.

Još jedan veliki igrač u svetu standarda je i **Institut inženjera elektrotehnike i elektronike** (engl. *Institute of Electrical and Electronic Engineers, IEEE),* najveća profesionalna organizacija na svetu. Osim što objavljuje više stručnih časopisa i održava

nekoliko stotina konferencija svake godine, IEEE ima grupu za izradu standarda u oblasti elektrotehnike i računarstva. Njihov komitet 802 standardizovao je mnoge vrste lokalnih mreža. Neke rezultate ovog komiteta razmotrićemo kasnije. Posao obavljaju radne grupe, pobrojane na slici 1-38. Uspešnost mnogih radnih grupa komiteta 802 nije bila velika; to što ispred sebe držite broj 802.x nije garancija uspeha. Ali su zato uspešno obavljeni poslovi (naročito na standardima 802.3 i 802.11) ostavili za sobom neizbrisiv trag.

Broj	Tema
802.1	Opšti pregled i arhitektura lokalnih mreža
802.2 i	Upravljanje logičkim vezama
802.3 *	Ethernet
802.4 4-	Token bus (kratko korišćena u proizvodnim pogonima)
802.5	Token ring (IBM-ova ulaznica u svet lokalnih mreža)
802.6 4-	Dvostruki red čekanja s dvostrukom sabirnicom (prve gradske mreže)
802.7 -i	Tehnička konsultantska grupa za širokopojasne tehnologije
802.8 f	Tehnička konsultantska grupa za tehnologiju optičkih vlakana
•	Izohrone lokalne mreže (za aplikacije koje se izvršavaju u realnom vremenu)
802.10 4-	Virtuelne lokalne mreže i bezbednost
802.11 *	Bežične lokalne mreže
•	Prioritet zahteva (Hevvlett-Packardov AnyLAN)
802.13	Nesrećan broj. Niko ga nije hteo
• i	Kablovski modemi (prestala s radom: u oblast je prvi uskočio jedan industrijski konzorcijum)
802.15 *	Lične mreže (Bluetooth)
802.16 *	Širokopojasne bežične mreže
802.17	Prsten sa elastičnim paketima

Slika 1-38. Radne grupe komiteta 802. One važne su označene zvezdicom (*). One koje nose oznaku -i hibemiraju. Grupe sa oznakom t prestale su s radom i raspuštene su.

1.6.3 Ko je ko u svetu standarda za Internet

Globalni Internet ima sopstvene mehanizme standarđizovanja, veoma različite od postupaka koje primenjuju ITU-T i ISO. Razlika se grubo može ilustrovati ako kažemo da ljudi koji se okupljaju na sastancima organizacija ITU i ISO nose odela. Ljudi koji se okupljaju na sastancima posvećenim standardizovanju Interneta obučeni su u džins (osim ako je sastanak u San Dijegu, kada nose šorceve i majice).

ITU i ISO skupovima prisustvuju korporacijski i državni zvaničnici kojima je standardizovanje posao. Oni smatraju da su standardi „dobra stvar“ i posvećuju im čitav život. Korisnici Interneta, s druge strane, anarhiju smatraju gotovo principom. Međutim, ako svaki od miliona ljudi radi što mu se prohte, teško je ostvariti međusobnu komunikaciju. Zato, ma kako to bolno bilo, ponekad nešto treba i standardizovati.

Kada je mreža ARPANET puštena u rad, Ministarstvo odbrane je obrazovalo formalni komitet daje nadgleda. Godine 1983, komitet je preimenovan u **Odbor za aktivnosti na Internetu** (engl. *Internet Activities Board, IAB)* i dodeljena mu je malo šira misija: da i dalje

podstiče istraživače da mrežu razvijaju, i da Internet održava u približno istom smeru - aktivnost koja se može porediti s kroćenjem čopora mačaka. Ista skraćenica (IAB) korišćena je i onda kada je ime Odbora promenjeno u **Odbor za arhitekturu Interneta** (engl. *Internet Architecture Board).*

Svaki od desetak članova IAB-a rukovodi radnom grupom koja se bavi nekim trenutno važnim problemom. IAB se okuplja više puta godišnje da razmotri rezultate i da o njima izvesti Ministarstvo odbrane i NSF, organizacije koje su uglavnom finan- sirale IAB. Kada se ukazala potreba za novim standardom (npr. za novim algoritmom za usmeravanje), članovi IAB-a su ga razmatrali, a zatim objavljivali izmene, tako da su studenti koji su tek diplomirali (glavna softverska radna snaga Interneta) mogli da ih ugrade. Izmene su objavljivane u nizu tehničkih izveštaja zvanih **Zahtevi za komentare** (engl. *Request For Comments, RFCs).* RFC dokumenti su skladišteni na mreži tako da ih svaki zainteresovan korisnik može peuzeti sa adrese www.ietf.org/ *rfc.* Oni su numerisani hronološkim redom pojavljivanja i danas ih ima preko 3.000. U ovoj knjizi ćemo se pozivati na mnoge RFC dokumente.

Do 1989. godine Internet je tako narastao da opisani neformalan stil rada više nije bio primenljiv. Mnogi prodavci su već nudili TCP/IP proizvode i nisu želeli da ih menjaju samo zato što šačica istraživača ima „bolju ideju". U leto 1989. godine IAB je ponovo reorganizovan. Istraživači su prebačeni u **Istraživačke snage Interneta** (engl. *Internet Research Task Force, IRTF),* telo podređeno IAB-u, kao i paralelno telo **Inženjerske snage Interneta** (engl. *Internet Engineering Task Force, IETF).* IAB je ponovo popunjen predstavnicima organizacija koje nisu više bile samo akademske i istraživačke. U početku je to bila grupa koja se sama obnavljala tako što su posle dvogodišnjeg mandata stari članovi imenovali nove. Kasnije je od osoba zain- teresovanih za Internet obrazovano **Internet društvo** (engl. *Internet Society).* Internet društvo je na taj način uporedivo sa organizacijama ACM ili IEEE. Njime upravljaju izabrani poverenici koji imenuju članove IAB-a.

Cilj ovakve podele bio je da se IRTF koncentriše na dugoročna istraživanja, dok bi se IETF bavio kratkoročnim inženjerskim poslovima. IETF je podeljen na radne grupe za rešavanje određenih problema. Predsednici grupa se na početku sastaju kao inicijativni odbor da bi usmerili inženjerske napore u pojedinim grupama i u celini. U delokrug radnih grupa spadaju nove aplikacije, korisničke informacije, OSI integri- sanje, usmeravanje i adresiranje, bezbednost, održavanje mreže i standardi. Na kraju se namnožilo toliko radnih grupa (više od 70) da su grupisane po oblastima, a inicijativni odbor čine predsednici pojedinih oblasti.

Osim toga, prihvaćenje formalniji postupak standardizacije, po ugledu na ISO. Da bi se došlo do **predloga standarda** (engl. *ProposedStandard),* osnovna ideja mora da bude potpuno objašnjena u RFC dokumentima i mora da bude dovoljno zaintereso- vanih korisnika da bi se razmatranje predloga isplatilo. Da bi se stiglo do **nacrta standarda** (engl. *Draft Standard),* funkcionalna verzija njegove realizacije mora biti rigorozno proverena barem na dve lokacije tokom najmanje 4 meseca. Kada se IAB uveri daje ideja dobra i da softver radi, on može odgovarajući RFC dokument proglasiti Standardom za Internet. Neki Internet standardi postali su standardi Ministarstva odbrane (MIL-STD), obavezni za dobavljače Ministarstva. David Clark je jednom dao (sada čuvenu) primedbu da se standardizovanje Interneta sastoji od „kakvog-tak- vog konsenzusa i funkcionalnog koda".

1.7 METRIČKE JEDINICE

Da bi se izbegla zabuna, treba naglasiti da se u ovoj knjizi, kao i uopšte u računarskim naukama, koriste metričke jedinice, a ne tradicionalne engleske jedinice (funte, pinte i ostalo). Osnovni prefiksi metričkih jedinica navedeni su na slici 1-39. Oni se najčešće skraćuju na početna slova, s tim što se prefiks koji označava jedinicu veću od osnovne piše velikim slovima (KB, MB itd.). Izuzetak je (iz istorijskih razloga) kb/s za kilobitove u sekundi. Shodno tome, komunikaciona linija brzine prenosa 1 Mb/s prenosi IO^6 bitova u sekundi, a interval od 100 ps (100 pikosekundi) iznosi IO'^{10} sekundi. Pošto prefiksi „mili“ i „mikro“ počinju slovom „m“, „mili“ se skraćeno piše „m“, a „mikro“ grčkim slovom „p“ (mi).

Eksp.	Dekadni množiteij	Prefiks	Eksp.	Dekadni množiteij	Prefiks
10^{-3}	0,001	mili	10^3	1.000	Kilo
10^{-6}	0,000001	mikro	10^6	1.000.000	Меда
$10'^9$	0,000000001	nano	10^9	1.000.000.000	Giga
10-12	0,000000000001	piko	10^{12}	1.000.000.000.000	Tera
$1C\Gamma^{15}$	0,000000000000001	femto	10^{15}	1.000.000.000.000.000	Peta
10-18	0,000000000000000001	ato	10^{18}	1.000.000.000.000.000.000	Eksa
$1C\Gamma^{21}$	0,000000000000000000001	cepto	10^{21}	1.000.000.000.000.000.000.000	Ceta
10-24	0,000000000000000000000001	jokto	10^{24}	1.000.000.000.000.000.000.000.000	Jota

Slika 1-39. Osnovni prefiksi metričkih jedinica.

Takođe treba naglasiti da se za merenje veličina memorije, diska, datoteka i baza podataka u industrijskoj praksi koriste jedinice s nešto drugačijim značenjem. Tamo kilo označava 2^{10} (1024) a ne 10^3 (1000), zato što se za izražavanje veličine memorije u računarstvu uvek umesto dekadnog koristi binarni brojčani sistem. Tako memorija od 1 KB sadrži 1024, a ne 1000 bajtova. Slično tome, memorija od 1 MB sadrži 2 (1.048.576) bajtova, od 1 GB sadrži 2^{30} (1.07.3.741.824) bajta, a baza podataka od 1 TB sadrži 2^{40} (1.099.511.627.776) bajtova. Međutim, komunikaciona linija brzine 1 kb/s prenosi 1000 bitova u sekundi, a lokalna mreža od 10 Mb/s radi brzinom 10.000.000 bitova u sekundi zato što se ove brzine izražavaju u dekadnom sistemu. Nažalost, mnogi mešaju ova dva brojčana sistema, naročito kada razmatraju veličinu diska. Da bismo izbegli nedoumice, u ovoj knjizi čemo koristiti simbole KB, MB i GB za 2^{10},2^{20}, odnosno 2^{30} bajtova, a simbole kb/s, Mb/s i Gb/s za 10^3, IO^6, odnosno IO^9 bitova u sekundi.

1.8 PREGLED OSTATKA KNJIGE

Knjiga obrađuje principe i praksu umrežavanja računara. Poglavlja većinom počinju razmatranjem odgovarajućih principa, a zatim sledi više primera koji ilustru- ju te principe. Prirneri su obično uzeti sa Interneta i iz bežičnih mreža, zato što su ove dve mreže veoma važne, a međusobno vrlo različite. Gde je pogodno, dajemo i druge primere.

U knjizi se držimo hibridnog modela prikazanog na slici 1-24. Počev od 2. poglavlja krećemo se uz hijerarhiju protokola polazeći od samog dna. U drugom poglavlju razmatramo osnove prenosa podataka. Tu govorimo o kablovskim, bežičnim i satelitskim sistemima prenosa. Izneti materijal se tiče fizičkog sloja, iako više govorimo o njegovoj arhitekturi, nego o hardveru. Kroz više primera obrađujemo javnu komutiranu telefonsku mrežu, mobilnu telefoniju i kablovsku televiziju.

U trećem poglavlju govorimo o sloju veze podataka i njegovim protokolima, razmatrajući složenije primere. Tu je i potpuna analiza protokola. Posle toga govorimo o nekim važnim protokolima iz stvarnog života, uključujući HDLC (koji se koristi u mrežama niske i srednje brzine) i PPP (koji se koristi na Internetu).

Četvrto poglavlje tiče se podsloja za upravljanje pristupom medijumima, koji je deo sloja veze podataka. Osnovno pitanje koje razmatramo jeste redosled korišćenja mreže sačinjene od jednog zajedničkog kanala, kao većina lokalnih mreža i neke satelitske mreže. Prikazano je mnogo primera kablovskih i bežičnih lokalnih mreža (naročito Etherneta), bežičnih gradskih mreža, Bluetootha i satelitskih mreža. U ovom poglavlju govorimo i o mostovima (engl. *bridges*) i skretnicama (engl. *switches)* sloja veze, koji se koriste za povezivanje lokalnih mreža.

Peto poglavlje se bavi mrežnim slojem, naročito usmeravanjem, pri čemu obrađujemo mnoge statičke i dinamičke algoritme. Čak i uz dobre algoritme za usmeravanje, ako je saobraćaj neprimeren mreži, može doći do zagušenja, pa govorimo i o zagušenju i o načinima da se ono spreči. Sprečiti zagušenje je poželjno, ali je još bolje garantovati određen kvalitet usluge. I o tome govorimo u ovom poglavlju, kao i o međusobnom povezivanju heterogenih mreža i problemima koji tu nastaju. Mrežnom sloju na Internetu posvećen je veliki deo poglavlja.

Šesto poglavlje se odnosi na transportni sloj. Naglasak je stavljen na protokole za rad sa

direktnom vezom jer to zahtevaju mnoge aplikacije. Dat je primer jedne transportne usluge i prikazan je njen stvarni kod da bi se pokazalo kako se ona može reali- zovati. Detaljno su opisani protokoli za prenos podataka na Internetu, TJDP i TCP, kao i problematika njihovih performansi. Opisani su i problemi koji se tiču bežičnih mreža.

U sedmom poglavlju govorimo o sloju aplikacija, njegovim protokolima i prime- ni. Prva tema je DNS, koji predstavlja „telefonski imenik“ Interneta. Sledi e-pošta i objašnjenje njenih protokola. Posle toga prelazimo na Web, detaljno opisujući statični i dinamični sadržaj Web strana, šta se događa kod klijenta a šta na serveru, protokole, performanse, bežični Web i mnogo dragih stvari. Na kraju govorimo o bežičnom pre- nosu multimedijskih sadržaja, uključujući reprodukovanje zvuka tokom preuzimanja (tj. u realnom vremenu), Internet radio i video na zahtev.

U osmom poglavlju obrađujemo bezbednost na mreži. Tema se odnosi na sve slojeve, pa je pogodnije da o njoj govorimo tek kada ih sve objasnimo. Poglavlje počinje uvodom u kriptografiju, a kasnije se ukazuje kako se ona može iskoristiti za ostvarivanje bezbedne komunikacije, e-pošte i Weba. Knjiga se završava raspravom o nekim područjima u kojima bezbednost zadire u privatnost, slobodu govora - govorimo o cenzuri i dragim društvenim aspektima.

Deveto poglavlje sadrži komentarisanu listu referenci predloženih za dalje čitanje, uređenu prema poglavljima. Ona je namenjena korisnicima koji svoja znanja o mrežama žele da prodube. Dat je i abecedno uređen spisak svih referenci navedenih u knjizi.

Adresa autorovog kutka na Web lokaciji izdavačke kuće Prentice Hali glasi:

http://www. prenhall. com/tanenbaum

Tamo ćete naći stranu pepunu veza ka mnogim priručnicima, zbirkama često postavljanih pitanja, industrijskim konzorcijumima, profesionalnim organizacijama, organizacijama za standardizovanje, tehnologijama, radovima itd.

1.9 SAŽETAK

Računarske mreže se mogu koristiti za pružanje brojnih usluga, kako kompanijama, tako i pojedincima. U kompanijama, mreže personalnih računara vezanih za zajednički server omogućavaju pristupanje korporacijskim podacima. Najčešće one slede klijent- sko-serverski model, pri čemu klijentske radne stanice sa stolova zaposlenih pristupaju snažnim serverima u računskom centru kompanije. Pojedinačnim korisnicima, mreže nude pristup različitim informacijama i izvorima zabave. Pojedinačni korisnici često pristupaju Internetu telefonskim putem, pozivajući davaoca Internet usluga pomoću modema, ali je sve više korisnika koji i kod kuće imaju stalnu vezu sa Internetom. Područje koje se ubrzano razvija jesu bežične mreže s novim aplikacijama, kao što je pristup e-pošti sa raznih prenosivih uređaja i m-trgovina.

Mreže se grubo mogu podeliti na lokalne, gradske, regionalne i međumreže, svaka sa svojim svojstvima, tehnologijama, brzinom i područjem primene. Lokalne mreže se nalaze unutar zgrade i rade velikom brzinom. Gradska mreža pokriva gradsko područje - npr. sistem kablovske televizije preko koga mnogi sada pristupaju i Internetu. Regionalna mreža pokriva državu ili kontinent. Lokalne i gradske mreže su nekomu- tirane (nemaju usmerivače); regionalne mreže su komutirane. Bežične mreže postaju sve popularnije, naročito bežične

lokalne mreže. Mreže se međusobno mogu povezati u kombinovane mreže (međumreže).

Mrežni softver obuhvata protokole s pravilima komunikacije između procesa. Postoje protokoli koji rade sa uspostavljanjem direktne veze i oni koji rade bez uspostavljanja direktne veze. Mreže većinom podržavaju hijerarhije protokola, pri čemu svaki sloj obezbeđuje usluge za sloj iznad sebe, štedeci ga detalja protokola koji koristi. Skupovi protokola se najčešće zasnivaju na modelima OSI ili TCP/IP. Oba modela imaju mrežni sloj, transportni sloj i sloj aplikacija, ali se razlikuju po drugim slojevima. Problematika projektovanja obuhvata multipleksiranje, upravljanje tokom podataka, kontrolu grešaka i drago. Veći deo ove knjige bavi se protokolima i njihovim projektovanjem.

Mreže korisnicima pružaju usluge koje se izvršavaju uz uspostavljanje direktne veze ili bez nje. U nekim mrežama, jedan sloj pruža usluge bez uspostavljanja veze, dok sloj iznad njega radi uz uspostavljanje direktne veze.

Najpoznatije mreže su Internet, ATM mreže, Ethernet i bežične lokalne mreže IEEE 802.11. Internet se razvio iz ARPANET-a, kome su, u cilju stvaranja međumreže, dodavane drage mreže. Sadašnji Internet nije jedinstvena mreža, već kombinova- na mreža sastavljena od više hiljada mreža. Ono što ga izdvaja od dragih mreža jeste jedinstveno korišćenje skupa protokola TCP/IP u čitavom sistemu. ATM se široko koristi u sistemu telefonije, za prenos podataka na velike daljine. Ethernet je najpopularnija lokalna mreža koja postoji u većini velikih kompanija i na univerzitetima. I, na kraju, počele su da se šire bežične lokalne mreže iznenađujuće velike brzine prenosa (do 54 Mb/s).

Da bi više računara moglo međusobno da komunicira, neophodno je štošta standardizovati, kako u oblasti hardvera, tako i softvera. Organizacije, kao što su ITU-T, ISO, IEEE i IAB sprovode različite delove postupka standardizacije.

ZADACI

1. Zamislite da ste svog bernardinca Bernija naučili da umesto bočice brendija nosi kutiju s tri 8-milimetarske trake. (Kada vam se disk napuni podacima, za vas je to hitna situacija.) Svaka traka može da primi 7 gigabajta podataka. Gde god da ste, pas može da dođe do vas brzinom 18 km/h. U kom intervalu razdaljina Berni prenosi podatke brže (zanemarujući prateće podatke) nego prenosna linija brzine 150 Mb/s?
2. Alternativa lokalnoj mreži je veliki sistem s deljenjem vremena, gde svaki korisnik ima terminal. Navedite dve prednosti lokalne mreže s klijentsko-serverskim sistemom.
3. Na performanse klijentsko-serverskog sistema utiču dva parametra mreže: njen propusni opseg (najveći broj bitova koje mreža može da prenese u sekundi) i kašnjenje (broj sekundi potreban da prvi bit stigne od klijenta do servera). Navedite primer mreže velikog propusnog opsega i velikog kašnjenja. Zatim navedite primer mreže malog propusnog opsega i malog kašnjenja.
4. Osim propusnog opsega i kašnjenja, kojim drugim parametrima se opisuje kvalitet usluge koju nudi mreža za prenos digitalizovanog govora.
5. Činilac koji utiče na kašnjenje u komutiranom sistemu paketa „čuvaj i prosledi“ jeste vreme potrebno da se paket sačuva i prosledi kim skretnicu. Ako je vreme zadržavanja paketa u skretnici 10 ps, da li će ono bitno uticati na odgovor klijentsko- -serverskog sistema u kome je jedan klijent u Njujorku, a drugi u Kaliforniji? Pretpostavite da brzina prostiranja signala kroz bakarnu žicu i optičko vlakno iznosi 2/3 brzine svetlosti u vakuumu.

6. Klijentsko-serverski sistem koristi satelitsku mrežu čiji se satelit nalazi na visini od 40.000 km. Koliko je kašnjenje odgovora na upit u najboljem slučaju?
7. U budućnosti, kada svako bude imao kućni terminal povezan s računarskom mrežom, moći će trenutno da se sprovode referendumi o važnim zakonima. Na taj način bi se mogle ukinuti sve institucije koje razmatraju zakone, jer se u vrlo kratkom roku može sagledati volja naroda. Pozitivni aspekti takve neposredne demokratije sasvim su očiti; razmotrite i njene negativne aspekte.
8. Skup od pet usmerivača treba povezati s podmrežom tipa od tačke do tačke. Svaki par usmerivača projektant može da poveže linijom visoke, srednje ili niske brzine, ili da ih uopšte ne poveže. Ako računaru treba 100 ms da generiše i proveri svaku to- pologiju, za koliko vremena će ih sve ispitati?
9. Grupa od 2"-l usmerivača povezana je u centralizovano binarno stablo, s usmeri- vačem u svakom čvoru stabla. Usmerivač *i* komunicira sa usmerivačem *j* tako što šalje poruku korenu stabla. Koren tada vraća poruku usmerivaču *j*. Izvedite približan izraz za srednji broj skokova po poruci za veliko *n,* pretpostavljajući jednaku vero- vatnoću izbora svih parova usmerivača.
10. Nedostatak podmreže s difuznim emitovanjem ogleda se u smanjenju njenog kapaciteta kada više računara istovremeno pokuša da pristupi kanalu. Pretpostavite daje vreme izdeljeno na diskretne intervale i da tokom svakog od tih intervala svaki od *n* računara pokušava da pristupi kanalu s verovatnoćom *p.* Koliki se udeo vremenskih intervala protraći zbog sukobljavanja?
11. Navedite dva razloga u prilog korišćenja protokola razmeštenih po slojevima.
12. Predsednik kompanije Specijalne Boje došao je na ideju da lokalnoj pivari ponudi saradnju u proizvodnji nevidljivih limenki (kako bi se izbeglo zagađenje okoline). Predsednik je naložio svojoj pravnoj službi da preduzme odgovarajuće korake, a služba je zatražila pomoć od inženjerskog sektora. Glavni inženjer je tada pozvao svog kolegu u pivari i s njim razmotrio tehničku stranu saradnje. Oba inženjera su o ovom razgovoru obavestila svoje pravne službe koje su se preko telefona dogovorile o pravnim aspektima saradnje. Na kraju su dva predsednika dogovorili finansijske uslove poduhvata. Da lije ovo primer protokola koji se izvršavaju po slojevima modela OSI?
13. Koja je osnovna razlika između komunikacije sa uspostavljanjem direktne veze i komunikacije bez uspostavljanja direktne veze?
14. Svaka od dve mreže obezbeđuje pouzdanu uslugu sa uspostavljanjem direktne veze. Jedna od njih nudi pouzdan tok bajtova, a druga pouzdan tok poruka. Jesu li mreže identične? Ako jesu, zašto se između njih pravi razlika? Ako nisu, navedite u čemu se razlikuju.
15. Šta znači „dogovaranje“ kada se govori o mrežnim protokolima? Ponudite primer.
16. Na slici 1-19 prikazana je usluga. Da li se na slici podrazumevaju i druge usluge? Ako je tako, gde su? Ako nije, zašto ih nema?
17. U nekim mrežama, sloj veze podataka obrađuje greške pri prenosu tako što zahteva ponovan prerios oštećenih okvira. Ako je verovatnoća oštećivanja okvira *p,* koliko se prosečno puta mora poslati okvir da bi sigurno stigao neoštećen? Pretpostavite da se potvrde o pri jemu okvira nikada ne gube.
18. Koji slojevi modela OSI rade sledeće:
    (a) Deljenje toka bitova u okvire.
    (b) Određivanje putanje kroz podrnrežu.
19. Ako se jedinica podataka razmenjenih na nivou veze podataka zove okvir, a jedinica podataka razmenjenih na mrežnom nivou zove paket, da li okviri kapsuliraju pakete ili paketi kapsuliraju okvire? Obrazložite odgovor.

20. Sistem ima hijerarhiju protokola u *n* slojeva. Aplikacije generišu poruke dužine *M* bajtova. Poruci se u svakom sloju dodaje zaglavlje od *h* bajtova. Koji deo propusnog opsega mreže zauzimaju zaglavlja?
21. Navedite dve karakteristike po kojima su referentni modeli OSI i TCP/IP isti. Zatim navedite dve karakteristike po kojima se oni razlikuju.
22. U čemu je osnovna razlika između protokola TCP i UDP?
23. Podmreža na slici l-25(b) projektovana je da izdrži nuklearni udar. Koliko je bombi potrebno da od nje naprave dva nepovezana dela? Pretpostavite da svaka bomba uništava čvor i sve veze koje se pružaju od njega.
24. Veličina Interneta udvostručava se približno svakih 18 meseci. Iako niko ne zna tačnu cifru, pretpostavlja se da gaje 2001. činilo 100 miliona računara. Na osnovu ovih podataka izračunajte očekivani broj umreženih računara 2010. godine. Verujete li daje rezultat tačan? Ako verujete, objasnite zašto; ako ne verujete, i to objasnite.
25. Kada se datoteka prenosi između dva računara, moguće su dve strategije potvrde prijema. Prema prvoj, datoteka se deli na pakete čije dospeće primalac nezavisno potvrđuje, ali ne šalje potvrdu daje datoteka primljena u celini. Prema drugoj, ne potvrđuje se prijem pojedinačnih paketa, ali se šalje potvrda kada cela poruka stigne na odredište. Prokomentarišite ova dva pristupa.
26. Zašto se u ATM mrežama koriste male ćelije fiksne dužine?
27. Koliko je (u metrima) dugačak jedan bit na mreži izgrađenoj prema prvobitnom standardu 802.3? Računajte s brzinom prenosa 10 Mb/s i pretpostavite daje brzina signala u koaksijalnom kablu 2/3 brzine svetlosti u vakuumu.
28. Slika je veličine 1024 x 768 piksela, sa 3 bajta po pikselu. Pretpostavite da je slika nekomprimovana. Koliko ce trajati njen prenos modemskim kanalom brzine 56 kb/s? A kablovskim modemom brzine 1 Mb/s? Ethemetom brzine 10 Mb/s? Kroz Ethernet brzine 100 Mb/s?
29. Ethernet i bežične mreže imaju izvesne sličnosti i neke razlike. Jedna od karakteristika Etherneta jeste to da se kroz mrežu može Slati samo jedan okvir u jednom trenutku. Da lije to tako i u mreži 802.11? Objasnite odgovor.
30. Bežične mreže se lako instaliraju, zbog čega su jeftine u poređenju s ožičenim mrežama gde troškovi instaliranja daleko nadmašuju cenu opreme. Pa ipak, i one imaju mana. Navedite dve.
31. Navedite dve prednosti i dve mane postojanja međunarodnih standarda za mrežne protokole.
32. Kada se komponenta sistema sastoji od fiksnog i izmenjivog dela (npr. CD čitač i CD), važno je da bude standardizovana, tako da različite kompanije mogu da proizvode fiksne i izmenjive delove, a da sve ipak radi dobro. Navedite tri primera takvih međunarodnih standarda izvan računarske industrije. Zatim navedite tri područja izvan računarske industrije gde takvi standardi ne postoje.
33. Napravite spisak svakodnevnih aktivnosti u kojima koristite računarske mreže. Koliko bi se vaš život izmenio kada bi se sve te mreže odjednom isključile?
34. Ispitajte koje se sve mreže koriste u vašoj školi, na fakultetu ili na radnom mestu. Opišite vrste mreža, njihove topologije i metode komutiranja.
35. Program *ping* omogućava da na datu lokaciju pošaljete probni paket i da utvrdite koliko mu treba da do nje stigne i da se vrati. Upotrebite *ping* da biste utvrdili koliko paketu treba da se vrati kada ga pošaljete na nekoliko dobro poznatih lokacija. Iz do- bijenih podataka uspostavite zavisnost između vremena putovanja preko Interneta u jednom smeru i razdaljine. Najbolje je da za to koristite univerzitete jer su lokacije njihovih servera tačno poznate. Na primer, *berkeley.edu* je u Berkliju u Kaliforniji, *mit.edu* je u

Kembridžu u Masačusetsu, *vu.nl* je u Amsterdamu u Holandiji, www.usyd.edu.au je u Sidneju u Australiji, a www.uct.ac.za je u Kejptaunu u Južnoj Africi.

36. Povežite se s Web lokacijom IETF-a, www.ietf.org, i pogledajte na čemu rade. Izaberite jedan projekat i napišite na pola strane izveštaj o datom problemu i predloženom rešenju.
37. Standardizacija je veoma važna za mreže. ITU i ISO su glavne zvanične organizacije za standardizovanje. Povežite se s njihovim lokacijama na Webu (www.itu.org i www.iso.org) i upoznajte se sa standardima na kojima rade. Napišite kratak izveštaj o vrstama proizvoda koje su ove organizacije standardizovale.
38. Internet je sastavljen od velikog broja mreža. Njihov raspored određuje topologiju Interneta o kojoj možete da nađete mnogo podataka na samoj Mreži. Pomoću pretraživača Weba pronađite podatke o topologiji Interneta i napišite kratak izveštaj o tome.

1

F1Z1CKS SLOJ

U ovom poglavlju bavićemo se najnižim slojem iz hijerarhije prikazane na slici 1-24. On definiše mehanički i električni interfejs prema mreži, kao i interfejs za sinhronizovanje. Opisivanje ćemo započeti teorijskom analizom prenosa podataka i odmah otkriti daje Majka Priroda postavila neka ograničenja u pogledu onoga što se može poslati kanalom.

Zatim ćemo opisati tri vrste prenosnih medijuma: fizičke (bakarnu žicu i optičko vlakno), bežične (zemaljski radio) i satelitske. Oni predstavljaju podlogu za glavne tehnologije prenosa u savremenim mrežama.

Ostatak poglavlja posvećenje opisu tri primera komunikacionih sistema koji se u praksi koriste za regionalne računarske mreže: (fiksnog) telefonskog sistema, sistema mobilne telefonije i sistema kablovske televizije. U sva tri sistema se za okosnicu koristi optičko vlakno, ali je svaki drugačije organizovan i koristi drugačiju tehnologiju za povezivanje s krajnjim korisnikom.

2.1 TEORIJSKE OSNOVE PRENOSA PODATAKA

Podaci se mogu prenositi žicom tako što se menja neko njeno fizičko svojstvo, npr. napon ili jačina struje. Kada predstavimo vrednost tog napona ili te jačine struje kao jednoznačnu funkciju vremena,/(f), možemo da modeliramo ponašanje signala i da ga analiziramo služeći se matematičkim metodama. Takva analiza je tema narednih odeljalta.

2.1.1 Furijeova **analiza**

Početkom 19. veka, francuski matematičar Žan-Baptist Furije dokazao je da se svaka normalna periodična funkcija *g(t)* periode *T* može predstaviti kao zbir (možda beskonačnog) broja sinusnih i kosinusnih funkcija:

$$g(t) = -c + \sum_{n=1} a_n sm\{2\%nft) + \sum_{n=1} b_n cos(2nnft) \qquad (2\text{-}1)$$

gde *jef=l/T* osnovna frekvencija, a_n i b_n su amplitude n-tog **harmonika** (člana) si- nusne i kosinusne funkcije, a *c* je konstanta. Tako razložena periodična funkcija naziva se **Furijeov niz** (engl. *Fourier seri.es).* Iz njega se može rekonstruisati prvobitna funkcija; to znači, ako je poznata perioda Ti ako su zadate amplitude, prvobitna funkcija se dobija sabiranjem niza iz jednačine (2-1).

Signal podataka koji ima ograničeno trajanje (kao i svi signali) može se rastaviti u Furijeov niz ako zamislimo da se njegov profil stalno ponavlja, tj. da se profil iz intervala od *0* do *T* istovetno ponavlja u intervalu od *T* do *2T,* itd.

Amplitude a_n mogu se izračunati za svaku finkciju *g(t)* množenjem obe strane jednačine (2-1) činiocem *smilnkft),* a zatim integrisanjem jednačine u intervalu od *0* do *T.* Pošto je

$$\int_0 \sin(271\ kft) \sin(2TC72/\Gamma)\, dt = \begin{cases} Q\ m\ k * n \\ (772 \text{ za } \kappa = n \end{cases}$$

posle integracije preostaje samo jedan član: a_n. Zbir sa amplitudom b_n potpuno iščezava. Slično tome, množecijednačinu (2-1) činiocem *cos(2T(kft)* i integrišuci je između *0* i *T,* možemo da dobijemo amplitudu b_n. Neposrednom integracijom obe strane jednačine (2-1) dobijamo *c.* Navedenim operacijama dobijaju se sledeđi rezultati: $m \qquad m \quad m$

$$a_n = -jg(t)sm(2nnft)dt \; b_n = -jg(t)cos(2nnft)dt \; c = j_{,}jg(t)dt \; ooo$$

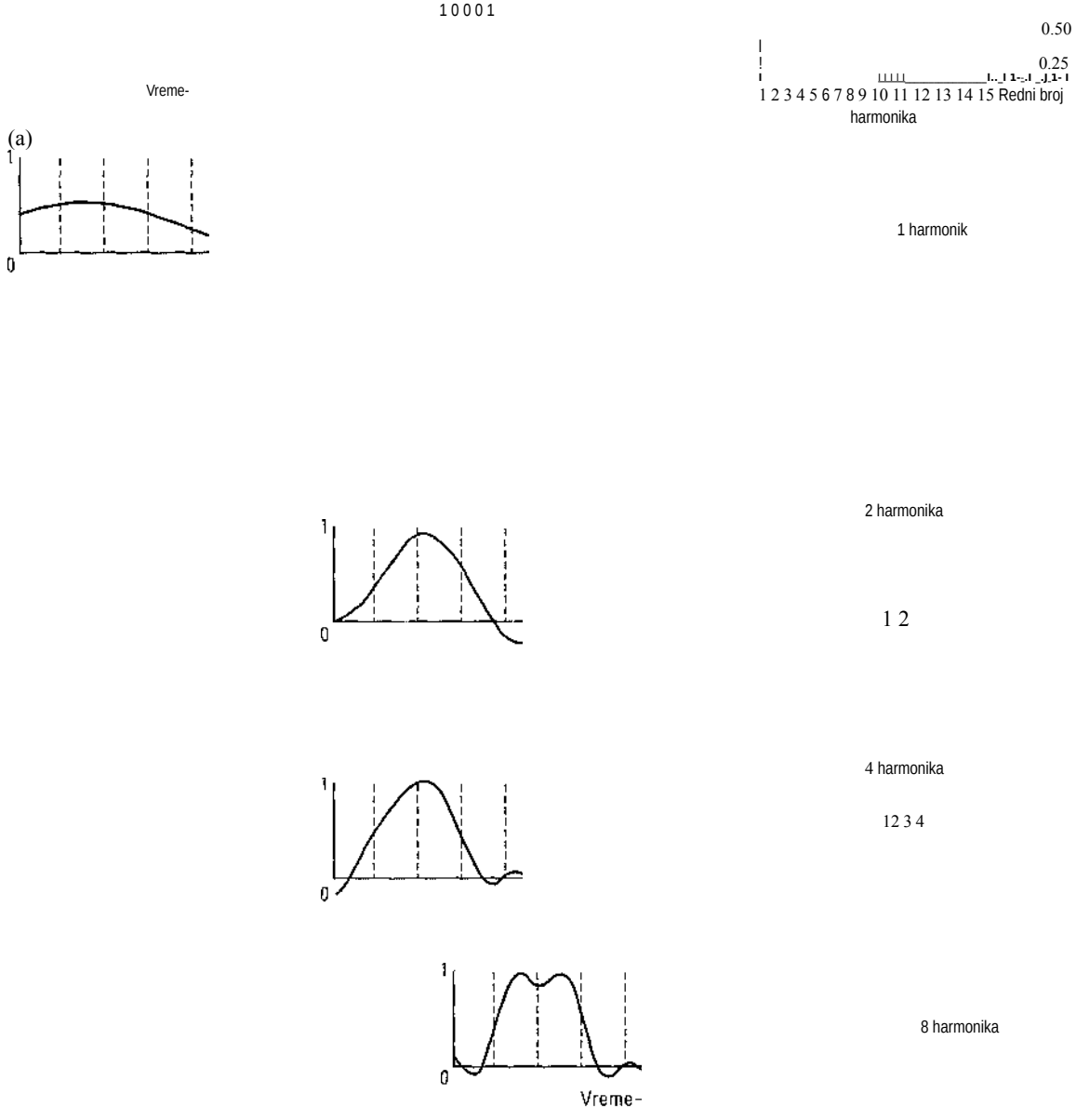
2.1.2 Signali ograničeni propusnim opsegom

Da bismo razumeli kakve veze sve ovo ima s prenosom podataka, razmotrimo jedan primer: slanje slova „b“ kodiranog pomoću 8 bitova (1 bajt). Niz bitova koje treba preneti izgleda ovako: 01100010. Leva strana slike 2-1 (a) prikazuje napon signala koji šalje računar. Furijeova analiza signala daje sledece koeficijente:

$$a_n = \sum[COS(7t72/4) - COS(37T72/4) + COS(6*n/4) - COS(7T\Gamma 72/4)] \quad b_n = \text{—}$$

[sin(37in/4)-sin(nn/4) + sin(77tn/4)-(sin67tn/4)]

c = 3/4



Slika 2-1. (a) Binarni signal i njegove srednjekvadratne Furijeove amplitude.
(b) -(e) Uzastopne aproksimacije prvobitnog signala.

Srednjekvadratne amplitude, *Ja%* + , za prvih nekoliko članova niza, prikazane

su na desnoj strani slike 2-1 (a). One su važne jer su njihovi kvadrati proporcionalni energiji koja se pri određenoj frekvenciji prenese.

Nema transportnog medijuma koji prenosi signale bez gubitaka. Kada bi sve komponente Furijeovog niza podjednako slabile, amplituda signala bila bi takođe manja, ali se signal ne bi izobličio, tj. imao bi isti pravougaoni oblik kao na slici 2-1 (a). Nažalost, u svim prenosnim medijumima različite komponente Furijeovog niza različito slabe, što dovodi do izobličenja signala. Amplitude signala obično se prenose bez slabljenja, počev od frekvencije 0, pa do neke frekvencije/,, a iznad te granične frekvencije (merene u broju ciklusa u sekundi, tj. u hercima, Hz) sve amplitude se smanjuju. Opseg frekvencija koje se prenose bez većeg slabljenja naziva se **propusni opseg** (engl. *bandwidth)*. Granična frekvencija u praksi nije tako oštra, pa se propusni opseg često definiše kao opseg frekvencija od 0 do frekvencije pri kojoj snaga signala opadne na polovinu.

Propusni opseg je fizičko svojstvo transportnog medijuma i obično zavisi od konstrukcije, debljine i dužine medijuma. U nekim slučajevima, u kolo se uključuje filtar koji korisnicima ograničava propusni opseg. Na primer, telefonska linija može na kratkim rastojanjima da ima propusni opseg od 1 MHz, ali telefonske kompanije postavljaju filtar koji korisnicima ograničava propusni opseg na oko 3100 Hz. Takav propusni opseg je dovoljan za razumljivo prenošenje govora, a istovremeno se povećava efikasnost sistema jer korisnici manje opterećuju resurse.

Razmotrimo sada kako bi izgledao signal sa slike 2-1(a) kada bi medijum propuštao samo najniže frekvencije, tj. kada bi funkcija bila približno predstavljena samo pomoću prvih nekoliko članova jednačine (2-1). Slika 2-1(b) prikazuje izgled signala u medijumu koji propušta samo prvi njegov harmonik - osnovnu frekvenciju/. Slično tome, slike 2-l(c)-(e) prikazuju spektre i rekonstruisane funkcije za kanale sa sve većim propusnim opsegom.

Pri zadatoj brzini prenosa *b* (bitova u sekundi), vreme potrebno da se, na primer, jedan za dragim pošalje 8 bitova, iznosi 8/7? sekundi, tako da je frekvencija prvog harmonika *bi*8 Hz. Obična **govorna telefonska linija** (engl. *voice-grade line)* ima ve- štački ugrađenu graničnu frekvenciju od oko 3000 Hz. To ograničenje znači da redni broj najvišeg harmonika koji takva linija propušta iznosi oko 3000/(h/8) ili 24.000/£> (granica nije oštra).

Na slici 2-2 prikazani su rezultati proračuna za neke brzine prenosa. Iz njih je jasno da će pokušaj slanja brzinom 9600 b/s preko govorne telefonske linije transformisati signal sa slike 2-l(a) u signal na slici 2-l(c), što će veoma otežati tačan prijem sekvence bitova. Takođe je očigledno da pri brzinama prenosa znatno višim od 38,4 kb/s uopšte nema šanse da se prenesu *binarni* signali, pa ni linijom u kojoj ne postoji šum. Drugim recima, ograničavanje propusnog opsega ograničava i brzinu prenosa, čak i kroz savršene kanale. Postoje, međutim, složeni sistemi kodiranja u kojima se koristi više naponskih nivoa i pomoću kojih se mogu postići veće brzine prenosa. O njima ćemo govoriti u nastavku poglavlja.

Brzina prenosa (b/s)	T (ms)	Prvi harmonik (Hz)	Broj harmonika propuštenih kanalom
300	26,67	37,5	80
600	13,33	75	40
1200	6,67	150	20
2400	3,33	300	10
4800	1,67	600	5
9600	0,83	1200	2
19200	0,42	2400	1
38400	0,21	4800	0

Slika 2-2. Odnos između brzine prenosa i broja harmonika.

2.1.3 Najveća brzina prenosa kroz kanal

Još 1924. godihe, Henri Nikvist (Henry Nyquist), inženjer korporacije AT&T, otkrio je da čak i savršen kanal ima ograničen kapacitet prenosa. On je izveo jednačinu za maksimalnu brzinu prenosa kroz bešumni kanal ograničene propusne moči. Klod Šenon (Claude Shannon) je 1948. godine proširio Nikvistovu jednačinu na kanale sa slučajnim (termodinamičkim) šumom (Shannon, 1948). Na ovome mestu ukratko čemo sumirati njihove rezultate koji se sada smatraju fundamentalnim.

Za slučaj prenošenja proizvoljnog signala kroz ograničavajući filtar propusnog opsega *H,* Nikvist je dokazao da se filtrirani signal može potpuno rekonstruisati ako se uzorkuje brzinom od (tačno) *2H* uzoraka u sekundi. Brže uzorkovanje nema smisla jer su viši harmonici koji bi se takvim uzorkovanjem mogli pojaviti *već* uklonjeni filtriranjem. Ako se signal sastoji od *V* diskretnih nivoa, Nikvistova teorema glasi:

$$\text{najveća brzina prenosa (b/s)} = 2H \log_2 V$$

Na primer, bešumni kanal propusnog opsega 3 kHz ne može da prenosi binarne signale (tj. s dva naponska nivoa) brzinom većom od 6000 b/s.

Dosad smo razmatrali samo bešumne kanale. Situacija se naglo pogoršava ako postoji slučajni (termički) šum, a on uvek postoji zbog kretanja molekula u sistemu. Termički šum se izražava kao količnik snage signala i snage šuma i naziva se odnos signala i šuma (engl. *signal-to-noise ratio*). Ako snagu signala označimo sa *S,* a snagu šuma sa *N,* odnos signala i šuma je *S/N.* Obično se ne navodi sam odnos, već vrednost 10 logio *S/N,* izražena u decibelima (dB). Vrednosti odnosa *S/N* od 10 odgovara 10 dB, vrednosti 100 odgovara 20 dB, vrednosti 1000 odgovara 30 dB itd. Proizvođači stereo-pojačivača često određuju propusni opseg (frekventno područje) u kome njihov uređaj radi linearno, tako što krajeve opsega definišu frekvencijom od 3 dB, tj. tačlca- ma u kojima se faktor pojačanja smanjuje približno na polovinu (jer je $\log_{10}3 \sim 0{,}5$).

Osnovni rezultat Šenonovog rada jeste jednačina koja prikazuje maksimalnu brzinu prenosa kroz kanal propusnog opsega *H* Hz i odnosa signala i suma *S/N:*

$$\text{najveća brzina prenosa (b/s)} = H \log_2 (1 + S/N)$$

Na primer, kanal propusnog opsega 3000 Hz, sa odnosom signala i termičkog šuma 30 dB (tipični parametri analognog dela telefonskog sistema) neće nikada moći da prenosi podatke brzinom mnogo većom od 30.000 b/s, bez obzira na broj naponskih nivoa signala ili učestalost uzorkovanja. Senonov rezultat je izveden na osnovu teorije informacija i važi za sve kanale s termičkim šumom. Protivargumente ovom rezultatu treba posmatrati kao spekulacije koje podržavaju *perpetuum mobile.* Treba naglasiti da je Senon postavio gornju teorijsku granicu - realni sistemi je retko dostižu.

2.2 FIZIČKI MEDIJUMI ZA PRENOS PODATAKA

Osnovna svrha fizičkog sloja je da niz bitova prenese bez greške s jednog računara na dragi. Za prenos se mogu koristiti različiti fizički medijumi, a svaki od njih ima svoje mesto u pogledu propusnog opsega, kašnjenja, cene i lakoće instaliranja i održavanja. Medijumi se okvirno dele na materijalne, kao što su bakama žica i optičko vlakno, i nematerijalne, kao što su radio-talasi i laserski snopovi. Sve ćemo ih redom opisati u narednim odeljcima.

2.2.1 Magnetni medijumi

Jedan od najčešćih načina za prenos podataka s jednog računara na dragi sastoji se u tome da se podaci upišu na magnetnu traku ili izmenjivi medijum (npr. na upisivi DVD), da se traka ili diskovi fizički dopreme do odredišnog računara i da se podaci s njih učitaju u taj računar. Iako opisani postupak nije tako elegantan kao prenos podataka sistemom geosinhronih komunikacionih satelita, on je često jeftiniji, naročito u slučajevima u kojima su ključni činioci brzina prenosa ili cena prenetog bita podataka.

To se može objasniti jednostavnom računicom. Magnetna traka izrađena po industrijskom standardu Ultrium može da primi 200 gigabajta podataka. Kutija veličine 60 x 60 x 60 cm može da primi oko 1000 takvih traka, ukupnog kapaciteta 200 tera- bajta ili 1600 terabita (1,6 petabita). Kutija s trakama može da se dostavi ekspresnom poštom na bilo koju adresu unutar SAD u roku od 24 časa. Efektivna brzina ovog prenosa je 1600 terabita/86.400 s, ili 19 Gb/s. Ako je odredište udaljeno samo jedan sat vožnje dramom, brzina prenosa se povećava na preko 400 Gb/s. Nema računarske mreže koja se ovoj vrednosti može i približiti.

Za banku koja dnevno pravi gigabajte rezervnih kopija podataka na drugom računani (da bi mogla da nastavi s radom i u slučaju elementarnih nepogoda - poplava, zemljotresa), nije verovatno da će se prenosu podataka na magnetnoj traci po performansama uskoro približiti ijedna druga tehnologija. Mreže su, naravno, sve brže i brže, ali se povećava i gustina zapisivanja podataka na traku.

Ako razmotrimo cenu takvog prenosa podataka, zaključicemo slično. Cena jedne Ultrium trake iznosi oko 40 dolara kada se kupi na veliko. Traka se može koristiti barem deset puta, tako da je cena kutije s trakama, po jednoj upotrebi, oko 4000 dolara. Dodajte na to oko 1000 dolara za poštanske troškove (verovatno su oni mnogo niži), i imamo cenu od oko 5000 dolara za prenos 200 TB podataka, odakle se može izračunati da prenos gigabajta podataka

košta manje od 3 centa. Nijedna mreža ne može da radi tako jeftino. Naravoučenije:

Nikada ne potcenjujte brzinu prenosa podataka pomoću šlepera prepunog traka koji grabi niz drum.

2.2.2 Upredena parica

Iako je brzina prenosa podataka na magnetnoj traci zapanjujuća, ogromno je i kašnjenje koje se pojavljuje pri takvom prenosu. Vreme prenosa podataka ne meri se milisekundama, već minutima i satima. Za mnoge primene je neophodna direktna veza, a jedan od najstarijih i još uvele čestih prenosnih medijuma koji je omogućuje jeste upredena parica (engl. *twisted pair)*. Upredenu paricu čine dve izolovane bakarne žice, najčešće prečnilea oko 1 mm. Žice su međusobno spiralno uvijene (upredene), baš kao molekul DNK. Žice se upredaju zato što dve paralelne žice predstavljaju odličnu antenu. Kada se žice upredu, poništavaju se talasi generisani u različitim navojima, tako da ceo sklop zrači mnogo manje.

Upredena parica se najčešće koristi u telefonskom sistemu. Skoro svi telefoni su s telefonskom centralom povezani pomoću upredene parice. Upredena parica može da se proteže više kilometara bez pojačivača, ali su oni neophodni za veće razdaljine. Kada se veliki broj paralelnih parica protežu na veću daljinu - na primer, sve linije iz jednog stambenog bloka do telefonske centrale - one se povezuju u snop i obavijaju zaštitnim omotačem. Da nisu upredene, pojedinačne parice bi ometale jedna drugu. U područjima u kojima se telefonske linije razvode pomoću bandera, nije retkost videti njihove snopove debele više santirnetara.

Upredenom paricom se mogu prenositi i analogni i digitalni signali. Brzina prenosa zavisi od debljine žice i rastojanja, ali se za daljine od nekoliko kilometara u većini slučajeva može postići brzina od više megabita u sekundi. Zahvaljujući niskoj ceni i prihvatljivim performansama, upredena parica se masovno koristi i sva je prilika da će ostati na sceni još mnogo godina.

Upredena parica se realizuje u više oblika, od kojih su dva posebno važna za računarske mreže. Upredena parica 3. kategorije sastoji se od dve blago uvijene izolovane žice. Četiri takva para obično se grupišu u zajedničkom plastičnom omotaču koji ih drži zajedno i štiti. Približno do 1988. godine većina poslovnih zgrada imala je jedan kabl 3. kategorije koji se protezao iz centralnog razvodnog ormana (engl. *wiring closet)* na svakom spratu, do svake kacelarije. Taj sistem je dozvoljavao da se preko razvodnog ormana u svakoj kancelariji povezu četiri obična telefona ili dva telefona s više linija.

Počev od 1988. godine, uvedene su savršenije parice 5. kategorije. One su ličile na parice 3. kategorije, ali su bile gušće upredene, čime je smanjeno preslušavanje (engl. *crosstalk)* između parica i omogućen kvalitetniji signal na većim razdaljinama, zbog čega su bile pogodnije za brzu komunikaciju između računara. Upravo se pojavljuju parice 6. i 7. kategorije, propusnog opsega 250 MHz, odnosno 600 MHz (u odnosu na propusni opseg od 16 MHz za paricu 3. kategorije, odnosno 100 MHz za paricu 5. kategorije).

Sve ove vrste žica spadaju u tzv. **neoklopljene upredene parice** (engl. *Unshielded Twisted Pair, UTP),* da bi se razlikovale od robusnih, skupih kablova sa oklopljenim upredenim paricama koje je IBM uveo početkom osamdesetih godina i koji se ipak nisu proširili izvan IBM-ovih instalacija. Realizacija upredene parice prikazana je na slici 2-3.

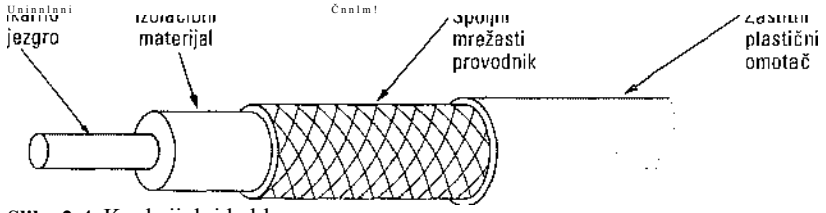
(a) (b)

Slika 2-3. (a) UTP 3. kategorije, (b) UTP **5.** kategorije.

2.2.3 Koaksijalni kabl

Drugi često korišđen prenosni medijum je **koaksijalni kabl** (u žargonu poznat i kao „koaks"). Bolje je oklopljen od upredene parice pa podatke može da prenosi većom brzinom i na veće daljine. Široko se koriste dve vrste koaksijalnog kabla. Jedan, 50-omski kabl, obično se koristi kada se od početka namenjuje digitalnom prenosu podataka. Drugi, 75-omski, obično se koristi za prenos analognih podataka i za kablovsku televiziju, ali postaje sve važniji i za kablovski Internet. Razlika između dve vrste koaksijalnih kablovauslovljena je pre istorijskim, nego tehničkim razlozima (na primer, uz prve dipolne antene impedanse 300 oma odmah su se mogli koristiti postojeći transformatori odnosa impedansi 4:1).

Koaksijalni kabl ima jezgro od čvrste bakarne žice oko koje se nalazi izolator. Oko izolatora je cilindrični provodnik napravljen od gusto upletene bakarne mrežice. Preko njega dolazi zaštitni plastični omotač. Izvučeni presek koaksijalnog kabla prikazan je na slici 2-4.



Slika 2-4. Koaksijalni kabl.

Konstrukcija i električna zaštita koaksijalnog kabla omogućavaju mu dobra kombinaciju velikog propusnog opsega i otpornosti na smetnje. Postignut propusni opseg zavisi od kvaliteta kabla, njegove dužine, i odnosa signala prema šumu. Savremeni koaksijalni kablovi imaju propusni opseg blizu 1 GHz. Koaksijalni kablovi su ranije široko korišćeni za međugradske i međudržavne veze u telefonskim sistemima, ali su danas na tim vezama u velikoj meri zamenjeni optičkim kablovima. Međutim, koaksijalni kablovi se i dalje masovno koriste za kablovsku televiziju i gradske mreže.

2.2.4 Optičko vlakno

Mnogi se u računarskoj industriji ponose brzim razvojem njenih tehnologija. Prvobitni (1981) IBM-ov PC radio je s taktom 4,77 MHz. Dvadeset godina kasnije, PC je radio na 2 GHz, što je ubrzanje od 20 puta u svakoj deceniji. Nije loše.

U istom periodu, prenos podataka na širem području ubrzan je sa 56 kb/s (ARPA- NET) na 1 Gb/s (savremene optičke komunikacije), što je ubrzanje od 125 puta u jednoj deceniji, dok je istovremeno učestalost grešaka smanjena sa 10^{15} praktično na nulu.

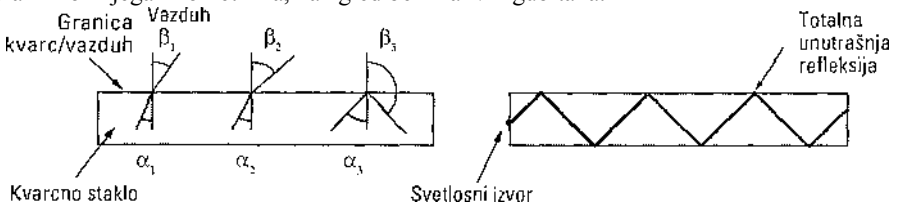
Osim toga, pojedinačni procesori već su počeli da se primiču svojim fizičkim granicama uslovljenim brzinom prostiranja svetlosti i problemima disipacije toplote. Nasuprot tome, brzina prenosa podataka optičkim vlaknima uz *današnju* tehnologiju prevazilazi 50.000 Gb/s (50 Tb/s), a mnogi već uveliko istražuju bolje tehnologije i materijale. Današnje praktično ograničenje brzine prenosa signala na vrednost od oko 10 Gb/s proizlazi iz naše nesposobnosti da većom brzinom pretvaramo optičke signale u električne i obrnuto, iako se u laboratoriji brzina 10 Gb/s postiže u jednom jedinom vlaknu.

U trci između računarstva i komunikacija, pobedile su komunikacije. Sve posledice raspolaganja suštinski neograničenim propusnim opsegom (iako ne besplatnog raspolaganja) nisu se još slegle u glavama računarskih naučnika i inženjera obučenih da poštuju Nikvistova i Šenonova ograničenja izvedena za bakarni provodnik. Prema novom nepisanom pravilu, svi računari su beznadežno spori i u mrežama ih po svaku cenu treba izbegavati, bez obzira na smanjenje propusnog opsega. U ovom odeljku ćemo proučiti optička vlakna da bismo utvrdili kako radi ta tehnologija prenosa podataka.

Optički sistem za prenos podataka sadrži tri glavne komponente: svetlosni izvor, prenosni medijum i detektor. Po konvenciji, svetlosni impuls označava bit 1, a odsustvo impulsa - bit 0. Prenosni medijum je ultratanko stakleno vlakno. Detektor proizvodi električni impuls kada na njega padne svetlosni zrak. Spajajući svetlosni izvor s jednim krajem optičkog vlakna, a detektor s njegovim drugim krajem, dobijamo jed- nosmerni sistem prenosa podataka koji prihvata električni signal, pretvara ga u svetlosni impuls i prenosi, a zatim ga na drugom kraju ponovo pretvara u električni signal.

Iz takvog prenosnog sistema svetlost bi „curila“ na sve strane i on bi, osim kao zanimljiva laboratorijska demonstracija, bio neupotrebljiv u praksi. Kada svetlosni zrak prelazi iz jedne u drugu materijalnu sredinu, na primer, iz kvarcnog stakla u vazduh, on se prelama (savija) na granici kvarc/vazduh, kao na slici 2-5(a). Tu vidimo svetlosni zrak koji na granicu između medijuma dolazi pod upadnim uglom pj, a napušta je pod uglom otj. Ugao prelamanja zavisi od prirode dva medijuma (zapravo, od njihovih indeksa prelamanja). Kada upadni ugao zraka pređe određenu kritičnu vrednost, zrak

uopšte ne prelazi u vazduh, već se vraća u kvare. Na taj način, zrak sa upadnim uglom većim ili jednakim kritičnom, zauvek je zarobljen u vlaknu, kao na slici 2-5(b), i može da prolazi kroz njega kilometrima, naizgled bez ikakvih gubitaka.



Slika 2-5. (a) Primer tri svetlosna zraka unutar kvarenog vlakna koji na granicu kvarc/vazduh stižu pod različitim uglom. (b) Svetlost zarobljena zahvaljujući totalnoj unutrašnjoj refleksiji.

Slika 2-5(b) prikazuje samo jedan zarobljeni zrak, ali kroz vlakno može istovremeno prolaziti više svetlosnih zrakova od kojih se svaki odbija pod drugačijim uglom, uvek većim od kritične vrednosti. Za svaki zrak se kaže da kroz vlakno prolazi drugačijim **režimom** (engl. *mode),* pa se vlakno sa ovim svojstvom naziva **multi- modno** ili **višerežimsko vlakno** (engl. *multimode fiber).*

Međutim, ako se prečnilc vlakna svede na nekoliko talasnih dužina svetlosti, vlakno radi kao talasovod i svetlost se kroz njega prostire samo pravolinijski, bez odbijanja; to je **monomodno** ili **jednorežimsko vlakno** (engl. *single-mode fiber).* Jednorežimska vlakna su skuplja, ali se široko koriste za veća rastojanja. Savremena jednorežimska vlakna mogu da prenose signale na daljinu od 100 km, brzinom 50 Gb/s, bez pojačavanja. U laboratoriji su na kraćim rastojanjima postignute još vec'e brzine prenosa.

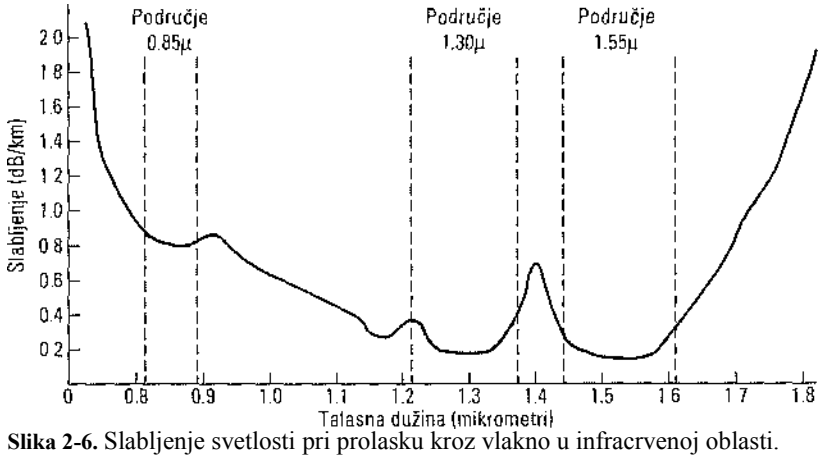
Prolazak svetlosti kroz vlakno

Optička vlakna se prave od stakla koje se, kao što je poznato, pravi od peska - jef'tine sirovine raspoložive u neograničenim količinama. Staklo su znali da prave još stari Egipćani, ali je njihovo staklo moralo biti tanje od 1 mm da bi propuštalo svetlost. Staklo koje je bilo dovoljno prozračno da bi se od njega mogli praviti prozori, stvoreno je tek tokom Renesanse. Staklo koje se koristi za savremena optička vlakna tako je prozračno da biste mogli videti okeansko dno kada bi voda bila njime za- menjena, baš kao što iz aviona po vedrom danu vidite zemlju.

Slabljenje svetlosti pri prolasku kroz staklo zavisi od talasne dužine svetlosti (kao i od izvesnih fizičkih svojstava stakla). Za vrstu stakla od koga su napravljena optička vlakna, na slici 2-6 prikazano je slabljenje u decibelima po dužnom kilometru vlakna. Slabljenje se u decibelima može izračunati pomoću formule

$$\text{Slabljenje u decibelima} = 10 \log \frac{\text{propuštena snaga}}{\text{ulazna snaga}}$$

Na primer, gubici za faktor dva odgovaraju slabljenju od 10 $\log_{10} 2 = 3$ dB. Slika prikazuje bliski infracrveni deo spektra koji se koristi u praksi. Vidljiva svetlost je nešto manjih talasnih dužina, između 0,4 i 0,7 mikrometara (1 mikrometar je 10^{-6} m). Zagriženi metrički čistunac bi ove talasne dužine izrazio kao 400 nm, odnosno 700 nm, ali ćemo se mi držati tradicionalnog označavanja.



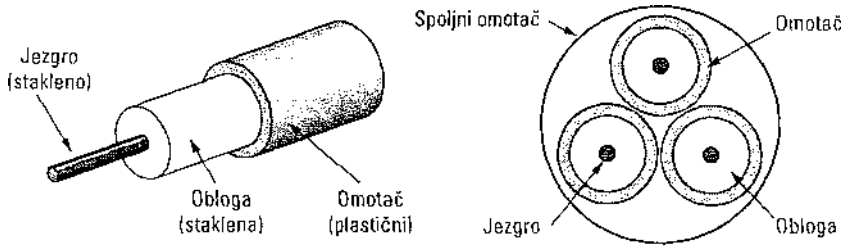
Slika 2-6. Slabljenje svetlosti pri prolasku kroz vlakno u infracrvenoj oblasti.

Za optičke komunikacije koriste se tri uska područja talasnih dužina, centrirana oko 0,85, 1,30 i 1,55 mikrometara. Poslednja dva obezbeđuju malo slabljenje (manje od *5%* gubitaka po kilometru). U području oko 0,85 mikrometara slabljenje je veće, ali se za njega laseri i elektronika mogu izraditi od istog materijala (galijum-arseni- da). Sva tri područja su širine od 25.000 do 30.000 GHz.

Svetlosni impulsi (različitih talasnih dužina) tokom prolaska kroz vlakno napreduju različitom brzinom. To rasipanje se naziva **hromatska disperzija** (engl. *chromatic dispersion).* Stepen rasipanja zavisi od talasne dužine svetlosti. Jedan način da se iz- begne preklapanje rasutih impulsa jeste da se poveća rastojanje između njih, ali to smanjuje brzinu slanja signala. Srećom, otkriveno je da se u impulsu specijalnog profila (izvedenog iz recipročne vrednosti hiperboličkog kosinusa), skoro svi disperzioni efekti poništavaju, tako daje moguće slati impulse desetinama hiljada kilometara bez primetnog izobličenja. Takvi impulsi se nazivaju **solitoni** (engl. *solitons).* Upravo se ulažu znatni istraživački napori da se solitoni iz laboratorije izvedu u svet.

Optički kablovi

Optički kablovi su slični koaksijalnim kablovima, samo što nemaju mrežasti provodnik. Slika 2-7(a) prikazuje izgled optičkog kabla s jednim vlaknom. Duž njegove ose proteže se stakleno jezgro kroz koje prolazi svetlosni zrak. U višerežimskim vla- kanima, pečnik jezgra je najčešće 50 mikrometara, što odgovara debljini vlasi kose. U jednorežimskim vlaknima, debljina jezgra je 8 do 10 mikrometara.



Slika 2-7. Izgled optičkog kabla s jednim vlaknom, (b) Presek optičkog kabla s tri vlakna.

Jezgro je okruženo oblogom od stakla čiji je indeks prelamanja manji od indeksa prelamanja jezgra, kako bi se sva svetlost zadržala u jezgru. Oko svega se nalazi plastični omotač koji štiti oblogu. Vlakna se najčešće grupišu u snopove i zaštićuju dodatnim spoljnim omotačem. Slika 2-7(b) prikazuje optički kabl s tri vlakna.

Optički kablovi se obično polažu na dubini od jednog metra ispod površine tla, gde ih povremeno oštećuju rovokopači i krtice. U blizini obale, prekookeanski optički kablovi polažu se u rovove pomoću neke vrste morskog pluga, a u dubokoj vodi oni jednostavno leže na dnu, gde ih mogu oštetiti ribati vukući parangal ili neka džinovska hobotnica.

Optička vlakna se mogu spajati na tri načina. Prvo, ona se mogu završavati konek- torima koji se priključuju u odgovarajuće utičnice. U konektorima se gubi oko 10 do 20% svetlosti, ali se zato lako može menjati konfiguracija sistema.

Drugo, optička vlakna se mogu spajati mehanički. Pri mehaničkom spajanju, paralelno isečeni krajevi vlakana postave se jedan uz drugi pomoću specijalnog rukavca i stegnu stezaljkom. Paralelnost preseka se može poboljšati ako se kroz vlakno propusti svetlost i pri tome spoj podešava sve dok se ne dobije najjači signal. Uvežbani tehničar može da napravi mehanički spoj za oko 5 minuta, a gubici svetlosti u njemu iznose oko 10 procenata.

Treći način je da se dva kraja vlakna međusobno stope i tako ostvare čvrst spoj. Vlakno spojeno stapanjem skoro se ne razlikuje od vlakna izvučenog u jednom komadu, mada i ovde dolazi do malog slabljenja svetlosti.

U sve tri vrste spojeva može da dođe do reflektovanja svetlosti na spoju, što ometa signal.

Za emitovanje signala obično se koriste dve vrste svetlosnih izvora, svetlosne diode (engl. *LightEmitting Diodes, LEDs)* i poluprovodnički laseri, koji imaju različita svojstva (slika 2-8). Talasna dužina svetlosti koju emituju može se podesiti pomoću Fabri- -Peroovog (Fabry-Perot) ili Mah-Zenderovog (Mach-Zehnder) interferometra posta vljenog između svetlosnog izvora i optičkog vlakna. Fabri-Peroov interferometar, je jednostavna rezonantna šupljina koju čine dva paralelna ogledala. Svetlost se uvodi normalno na ogledala. Unutar šupljine rezoniraju (pojačavaju se) talasi čija je talasna dužina celobrojni umnožak rastojanja između ogledala, dok svi ostali talasi slabe; na

taj način se bira željena talasna dužina svetlosti. Mah-Zenderov interferometar razlaže svetlost na dva snopa i upućuje ih putanjama čija se dužina neznatno razlikuje. Kada se na kraju snopovi opet spoje, pojačavaju se samo talasi određenih talasnih dužina - onih koje su u fazi.

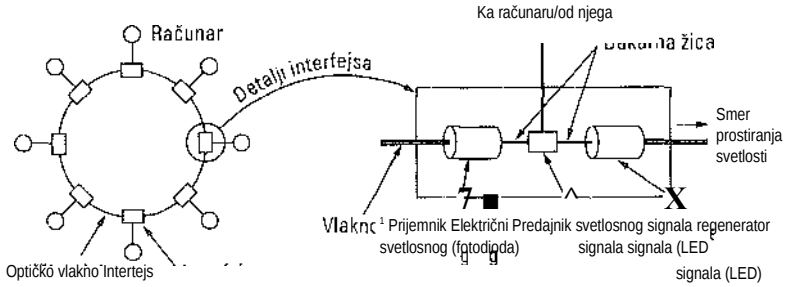
Na kraju optičkog vlakna suprotnom od svetlosnog izvora, nalazi se fotodioda koja generiše električni signal kada je pogodi zrak svetlosti. Tipično vreme reakcije foto- diode iznosi 1 *ns,* što ograničava brzinu prenosa podataka na oko 1 Gb/s. Termički šum takođe utiče, pa svetlosni impuls mora da ima dovoljnu energiju da bi bio detektovan. Kada su impulsi dovoljno snažni, učestalost grešaka se može proizvoljno smanjiti.

Svojstvo	Svetlosna dioda	Poluprovodnički laser
Brzina prenosa	mala	velika
Vrsta optičkog vlakna	jednorežimsko	jednorežimsko ili višerežimsko
Rastojanje	malo	veliko
Životni vek	dug	kratak
Osetljivost na temperaturu	zanemarljiva	znatna
Gena	niska	visoka

Slika 2-8. Poređenje poluprovodničkih lasera i svetlosnih dioda (LED) kao izvora svetlosti.

Mreže od optičkih vlakana

Optička vlakna se mogu koristiti i za lokalne mreže i za prenos podataka na velika rastojanja, iako je povezivanje s takvom mrežom složenije nego povezivanje sa Ethernetom. Sve je lakše kada shvatite da je prstenasta mreža u stvari skup veza od tačke do tačke, kao na slici 2-9. Interfejs svakog računara treba da propušta tok impulsa ka sledećoj vezi, a služi i kao T-spoj (račva) koji omogućuje da računar prima i šalje poruke.



Slika 2-9. Prsten od optičkog vlakna sa aktivnim repetitorima.

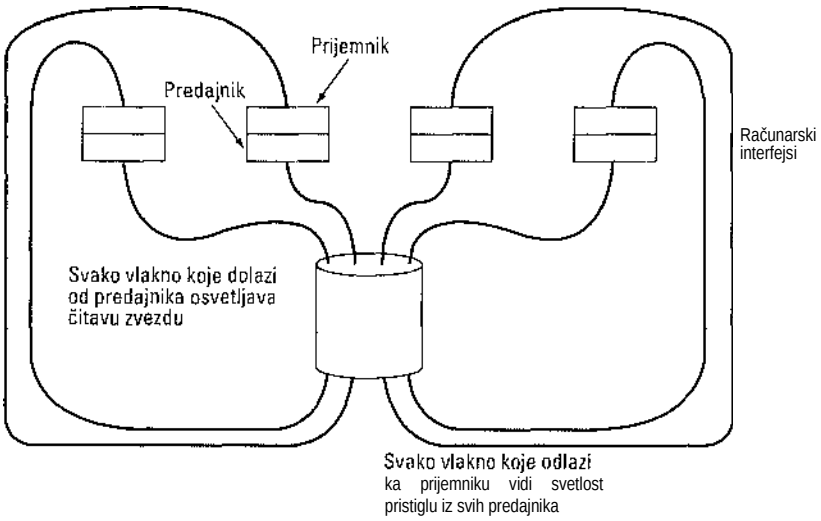
Koriste se dve vrste interfejsa. Pasivni interfejs se sastoji od dva priključna elemenata stapanjem spojena s glavnim vlaknom: jedan je odašiljač signala od računara ka vlaknu (LED ili laserska dioda), a drugi je prijemnik signala od vlakna ka računam

(fotodioda). Priključci rade potpuno pasivno i zato su pouzdani, jer kada „crkne“ fo- todioda ili LED, time se ne prekida prstenasta mreža, već se iz nje isključuje samo jedan računar.

Druga vrsta interfejsa, prikazana na slici 2-9, predstavlja **aktivan repetitor** (engl. *active repeater)*. Upadna svetlost se pretvara u električni signal koji se regeneriše do pune jačine ukoliko je oslabljen, a zatim ponovo pretvara u svetlost. Interfejs ka računam je obična bakarna žica koja ga spaja s regeneratorom signala. Danas se koriste i isključivo optički repetitori. U tim uređajima nema pretvaranja svetlosnog signala u električni i ponovo u svetlosni, što znači da imaju veliki propusni opseg.

Ako otkaže aktivni repetitor, prsten se prekida i mreža prestaje da radi. S druge strane, posto se signal regeneriše na svakom interfejsu, veze između susednih raču- nara mogu da budu kilometrima dugačke, a veličina same mreže nije naizgled ničim ograničena. Kod pasivnih interfejsa na svakom spoju se javljaju gubici, tako da su ukupan broj računara i veličina mreže veoma ograničeni.

Lokalna mreža od optičkih vlakana ne mora biti samo topologije prstena. Moguće je ostvariti hardversko difuzno emitovanje koristeći **topologiju pasivne zvezde** (engl. *passive star topology)* sa slike 2-10. U takvoj topologiji, svaki interfejs ima vlakno koje se od predajnika proteže do cilindra od kvarcnog stakla, gde su sva takva vlakna stapanjem spojena s jednim njegovim krajem. Slično tome, vlakna spojena s njegovim drugim krajem protežu se do svakog prijemnika istih interfejsa. Kad god jedan interfejs emituje svetlosni impuls, on se difuzno širi kroz topologiju pasivne zvezde osvetljavajući sve prijemnike, čime se ostvaruje neusmereno (difuzno) emitovanje. U stvari, pasivna zvezda kombinuje sve primljene signale i ukupan rezultat šalje svim računarima. Pošto se primljena energija deli na sve linije koje polaze od cilindra, broj čvorova mreže je ograničen osetljivošću fotodioda.



Slika 2-10. Topologija pasivne zvezde u mreži od optičkih vlakana.

Poređenje optičkih vlakana i bakarne žice

Poređenje staklenih vlakana s bakrom je poučno. Vlakna imaju brojne prednosti. Na prvom mestu, ona obezbeđuju mnogo vec'i propusni opseg od bakra. Samo zbog te osobine trebalo bi ih koristiti u najsavremenijim mrežama. Zbog malog slabljenja, repetitore treba postavljati tek na razdaljinama od oko 50 km, što u odnosu na svakih 5 km u slučaju bakarne žice, predstavlja veliku uštedu. Vlakna su u prednosti i zato što su neosetljiva na naponske udare, elektromagnetna polja i nestanke struje. Ona ta- kođe ne korodiraju u vazdušnoj sredini, zbog čega su idealna za teške fabričke uslove.

Ironično, ali telefonske kompanije cene optička vlakna iz sasvim drugih razloga: ona su tanka i laka. Mnogi postojeći kablovodi već su prepuni i ne mogu se proširivati. Uklanjanjem svog bakra iz kablovoda i njegovim zamenjivanjem optičkim vlaknima kablovod se rasterećuje, a bakarna žica se dobro može prodati topionicama koje je cene kao veoma bogatu bakarnu radu. Vlakna su i mnogo lakša od bakra. Hiljadu upredenih parica dužine 1 km imaju masu od 8000 kg. Dva vlakna imaju mnogo veći kapacitet, a masa im je samo 100 kg, što znatno snižava troškove mehaničkog potpornog sistema (na primer, bandera) koji se mora i održavati. U novoizgrađenim sistemima vlakno odnosi pobedu zbog mnogo nižih troškova instaliranja.

I na kraju, vlakno praktično ne rasipa svetlost u okolinu, a „kačenje“ na njega je teško, pa je skoro potpuno obezbeđeno od krađe informacija.

Vlakno ima i svoje loše strane - to je relativno nova tehnologija koju mnogi nedovoljno poznaju, a vlakna se lako oštećuju kada se previše saviju. Postoje prenos svet- losnih signala po prirodi jednosmeran, za dvosmernu komunikaciju su potrebna ili dva vlakna ili dve frekvencije u istom vlaknu. Najzad, interfejsi za optička vlakna koštaju mnogo više od električnih. Bez obzira na sve, potpuno je jasno da budućnost fiksnih sistema za prenos podataka pripada optičkim vlaknima čim se radi o rasto- janjima većim od nekoliko metara. Detaljno razmatranje svih aspekata optičkih vlakana i mreža napravljenih od njih naći ćete kod Hechta (2001).

2.3 BEŽIČNI PRENOS PODATAKA

Naše doba je izrodilo informatičke fanatike: pokretne korisnike koji na svakom mestu moraju da imaju pristup mreži. Za njih su beskorisne upredene parice, koaksijalni kablovi i optička vlakna. Oni moraju da dobijaju podatke na svoj prenosivi uređaj, pa bila to digitalna beležnica, džepni ili ručni računar, ili računar u obliku ručnog sata, bez ograničenja koja postavlja zemaljska komunikaciona infrastruktura. Za takve korisnike jedini odgovor su bežične komunikacije. U sledećim odeljcima detaljno ćemo razmotriti bežične komunikacije, pošto one imaju i mnoge druge primene osim da povezuju korisnike koji žele da lutaju Webom dok su na plaži.

Neki smatraju da budućnost imaju samo dve vrste komunikacija: optičke i bežične. Svi fiksni (tj. nepokretni) računati, telefoni, faksovi koristiće optičko vlakno, a njihovi pokretni parnjaci bežičnu vezu.

Bežična veza ima ponekad prednosti i za fiksne uređaje. Na primer, ako je dovođenje optičkog kabla do zgrade skopčano s teškoćama u vezi sa okolnim terenom (brda, džungle, močvare...), bežična veza može da reši problem. U tom smislu treba podsetiti daje razvoj savremenih bežičnih digitalnih komunikacija započeo na Havajima, gde su ostrva razdvojena

velikim vodenim prostranstvima, a telefonski sistem neodgovarajući.

2.3.1 Elektromagnetni spektar

Kada se elektroni kreću, oni proizvode elektromagnetne talase koji se prostiru kroz okolinu (čak i u vakuumu). Te talase je još 1865. teorijski predvideo britanski fizičar Džems Klerk Maksvel, a eksperimentalno dokazao 1887. nemački fizičar Hajn- rih Herc. Broj oscilacija talasa u sekundi naziva se **frekvencija** *(f)* i meri hercima (Hz) u čast Hajnriha Herca. Rastojanje između dva talasna maksimuma (ili minimuma) zove se **talasna dužina,** koja se uvek označava grčkim slovom lambda (A).

Kada se električno kolo spoji sa antenom odgovarajuće veličine, elektromagnetni talasi se mogu slati kroz prostor, gde ih na nekoj udaljenosti može primiti odgovarajući prijemnik. Sve bežične komunikacije zasnivaju se na navedenom principu.

Svi elektromagnetni talasi, bez obzira na frekvenciju, u vakuumu se kreću istom brzinom. Ta brzina, obično zvana **brzina svetlosti** (c), iznosi približno 3 x IO^8 m/s ili oko 1 stopu (30 cm) za nanosekundu. (Trebalo bi uspostaviti novu definiciju stope kao rastojanje koje svetlost pređe za 1 ns, a ne da se ona definiše prema veličini stopala nekog davno upokojenog kralja.) U bakarnom ili optičkom medijumu, brzina prostiranja elektromagnetnih talasa smanjuje se na oko 2/3 ove vrednosti i počinje neznatno da zavisi od frekvencije. Brzina svetlosti je univerzalna granica brzine. Nikakav materijalni objekat, niti signal ne mogu da prevaziđu tu brzinu.

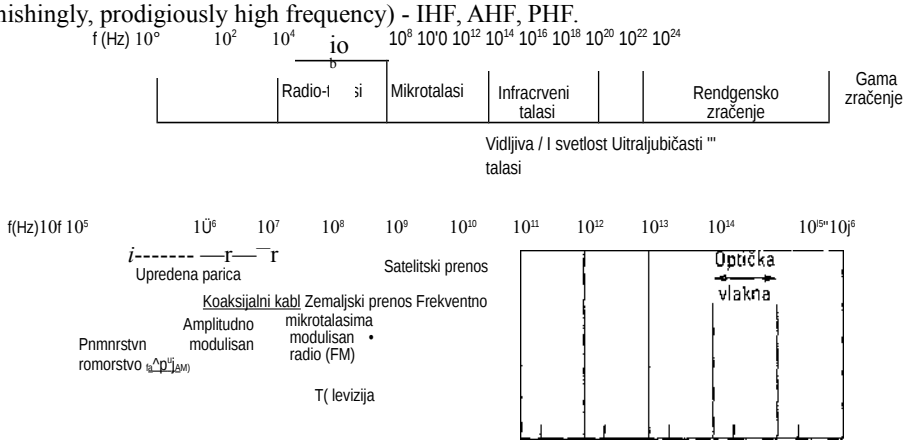
Osnovni odnos između/, A i c (u vakuumu) glasi

$$\text{Ŧ} = c \tag{2-2}$$

Pošto je *c* konstanta, ako znamo /, možemo da izračunamo A i obrnuto. Prema uprošćenom pravilu, kada je A u metrima, a/u MHz, onda je A/~ 300. Na primer, talasi frekvencije 100 MHz dugački su oko 3 metra, talasi frekvencije 1000 MHz dužine su 0,3 m, a talasi dužine 0,1 m imaju frekvenciju 3000 MHz.

Elektromagnetni spektar je prikazan na slici 2-11. Svi njegovi delovi: radio-talasi, mikrotalasi, infracrveni talasi i vidljivi svetlosni talasi mogu se koristiti za prenos informacija ako im se modulira amplituda, frekvencija ili faza. Još bolje bi bilo raditi sa ultraljubičastom svetlošću, rendgenskim i gama-zracima jer su im frekvencije više, ali se takvi talasi teže proizvode i teže moduliraju, ne prolaze lako kroz čvrsta tela i predstavljaju opasnost za živa bića. Oznake navedene u dnu slike 2-11 predstavljaju zva- nična ITU imena područja elektromagnetnog spektra različitih talasnih dužina. Tako se područje LF proteže između 1 i 10 km (približno od 30 kHz do 300 kHz). Nazivi LF, MF i HF skraćenice su za nisku, srednju i visoku frekvenciju (engl. /ow, medium, /tigh ,/requency). Naravno, u trenutku kada su imena davana, niko nije pretpostavljao da će se ići preko 10 MHz, tako da su dalja područja (VHF, UHF, DHF, EHF, THF)

označavana kao veoma, ultra, super, izuzetno i užasno visoke frekvencije (engl. very, ultra, super, extremely, tremendously *high*/requency). Posle toga im je ponestalo imena, ali bi lepo zvučale i neverovatno, zapanjujuće i monstruozno visoke frekvencije (engl. incredibly, astonishingly, prodigiously high frequency) - IHF, AHF, PHF.



Talasno LF MF HF VHF UHF SHF EHF THF područje

Slika 2-11. Elektromagnetni spektar i mogućnost njegove primene u komunikacijama.

Količina informacija koju može da prenese elektromagnetni talas zavisi od njegove frekvencije. Pri nižim frekvencijama, moguće je da se pomoću savremene tehnologije kodira nekoliko bitova po hercu, a pri visokim frekvencijama čak i do 8 bitova, tako da koaksijalni kabl propusnog opsega 750 MHz može da prenosi više gigabitova podataka u sekundi. Ako ponovo pogledamo sliku 2-11, jasno je zašto projektanti mreža toliko vole optička vlakna.

Ukoliko jednačinu (2-2) resimo po /i diferenciramo po A, dobijamo

$$\frac{df}{d\,l} \sim \sim \frac{c}{X^{2}}$$

Ako diferencijale zamenimo konačnim priraštajima i posmatramo samo apsolutne vrednosti, imamo

$$A/ = \qquad (2\text{-}3)$$

Za datu širinu područja talasnih dužina AA možemo da izračunamo odgovarajuće područje frekvencija A/, a odatle brzinu prenosa koja se ostvaruje u tom području. Sto je šire područje, brži je prenos. Razmotrimo, na primer, područje oko 1,30 mikrome- tara na slici 2-6. Ovde imamo A = 1,3 x IO'⁶ i AA = 0,17 x IO'⁶, tako da A/iznosi oko 30 THz. Pri kodiranju od, recimo, 8 b/Hz, dobijamo brzinu prenosa 240 Tb/s.

Za prenos se uglavnom koristi usko frekventno područje (Af/f « 1) da bi se postigao bolji prijem (više W/Hz). Međutim, u nekim slučajevima se koristi široko frekventno područje, uz dve varijante. Kod **skokovitog frekventnog širenja spektra**

(engl. *frequency hopping spread spectrum),* predajnik skače s jedne frekvencije na drugu više stotina puta u sekundi. Takvo emitovanje je popularno u vojsci jer se teško otkriva i gotovo da se ne može ometati. Pri takvom emitovanju nema slabljenja zbog različitih putanja talasa jer direktan signal uvek prvi stiže do prijemnika. Odbijeni signali slede dužu putanju i do prijemnika stižu kasnije, u momentu kada je on možda već promenio frekvenciju i ne prima signale na staroj frekvenciji, čime se eliminiše interferencija direktnih i odbijenih signala. Ova tehnologija se poslednjih godina pri- menjuje i u komercijalnim mrežama - koriste je i 802.11 i Bluetooth.

Kao zanimljivost navedimo da je u razradi opisane tehnike aktivno učestvovala seks-bomba Hedi Lamar, poreklom Austrijanka, prva žena koja se svukla **u** igranom filmu (*Ekstaza* - čehoslovački film iz 1933.). Njen prvi muž, proizvođač vojne opreme, pričao joj je kako se lako mogu blokirati radio-signali koji su tada korišćeni za navođenje torpeda. Kada je otkrila da on prodaje oružje Hitleru, užasnula se, prerušila se u kućnu pomoćnicu da bi mu utekla i pobegla u Holivud, gde je nastavila svoju karijera filmske glumice. Tamo je u dokolici smislila „skokovito frekventno širenje spektra" da bi pomogla saveznicima. Njen sistem je koristio 88 frekvencija - broj dirki (i frekvencija) na klaviru. Pronalazak, koji je razradila zajedno sa svojim prijateljem, kompozitorom Džordžom Antejem (George Antheil), zaštitila je patentom (U.S. 2,292,387). Međutim, nikako nisu mogli da ubede Američku mornaricu da njihov patent ima ikakvu praktičnu primenu, tako da na njemu nisu zaradili. On je postao popularan tek godinama posle isteka njihovih autorskih prava.

Druga varijanta, **direktno sekvencijalno širenje spektra** (engl. *direct sequential spread spectrum),* u kojoj se signal istovremeno prostire kroz široko frekventno područje, takođe je postala komercijalno privlačna. Koriste je neki sistemi mobilne tele- fonije druge generacije, a postaće dominantna u trećoj generaciji zahvaljujući svojoj efikasnosti u iskorišćavanju frekventnog spektra, otpornosti na šum i dragim svojstvima. Ona se koristi i u nekim bežičnim lokalnim mrežama. Na Širenje spektra ćemo se vratiti u dragom delu ovog poglavlja. Fascinirajući i detaljan prikaz istorije komunikacija uz širenje frekventnog spektra naći ćete kod Scholtza (1982).

Zasad ćemo pretpostaviti da se u svim prenosima koristi usko područje frekvencija i preći na razmatranje korišćenja pojedinih oblasti elektromagnetnog spektra sa slike 2- 11, počinjući s radio-talasima.

2.3.2 Prenos podataka radio-talasima

Radio-talase je lako generisati, oni mogu da prelaze velika rastojanja i lako prolaze kroz zgrade, zbog čega se naveliko koriste u komunikacijama, kako unutrašnjim, tako i spoljnim. Radio-talasi se prostiru na sve strane od izvora, tako da položaj predajnika i prijemnika nije od velikog značaja.

Ponekada je svojstvo širenja radio-talasa u svim pravcima poželjno, a ponekada nije. Sedamdesetih godina, General Motors je odlučio da opremi sve svoje nove kadi- lake računarski nadziranim kočnicama koje se ne blokiraju. Kada vozač nagazi pedalu kočnice, računar preuzima da je pritiska s prekidima, umesto konstantnom silom. Ali, jednog lepog dana, kada je saobraćajac na autoputu kroz Ohajo uključio svoj novi mobilni telefon da bi se javio stanici, jedan kadilak koji gaje mimoilazio poče da se „rita ko mlado ždrebe". Saobracajac gaje konačno primorao da se zaustavi, a vozač se kleo da nije radio baš ništa,

već daje automobil jednostavno poludeo.

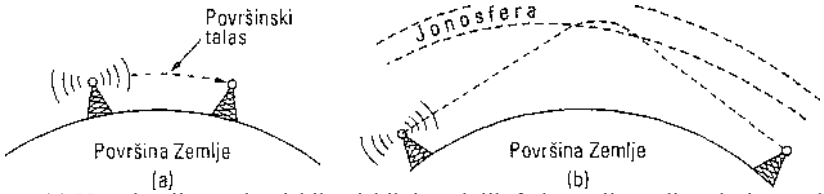
Ubrzo je počeo da se nazire uzrok: kadilaci bi ponekada podivljali, ali samo na glavnim putevima kroz Ohajo, a i tada samo onda kada bi na njih motrio neki saobracajac. Dugo vremena se General Motors pitao zašto kadilaci odlično rade u svim drugim državama i na manje značajnim putevima u Ohaju. Tek posle iscrpljujućeg istraživanja utvrdili su da je kadilakovo ožičenje predstavljalo odličan antenski sistem za frekvencije novog radio-sistema koji je koristila saobraćajna policija na autoputu kroz Ohajo.

Svojstva radio-talasa zavise od njihove frekvencije. Pri nižim frekvencijama, oni lako prolaze kroz prepreke, ali im snaga naglo opada s rastojanjem od izvora, približno s njegovim kvadratom (u vazduhu). Pri višim frekvencijama, radio-talasi teže da se prostiru pravolinijski i da se odbijaju od prepreka. Talcođe ih apsorbuje kiša. Bez obzira na frekvenciju, ometaju ih zračenja elektromotora i drugih električnih uređaja.

Zbog sposobnosti radio-talasa da prelaze velika rastojanja, javlja se problem interferencije između korisnika. Iz tog razloga, sve države škrto dele licence za korišćenje radio-predajnika, osim u jednom slučaju koji opisujemo u nastavku.

U području veoma niskih, niskih i srednjih frekvencija, radio-talasi prate krivinu Zemlje, kao na slici 2-12(a). Takve talase je moguće detektovati na razdaljini od možda 1000 km ako su frekvencije niske; rastojanje je manje ako su frekvencije više. Amplitudno moduliran radio koristi područje srednjih frekvencija (MF), zbog čega se Radio Boston slabo čuje u Njujorku. Radio-talasi iz ovog područja lako prolaze kroz zgrade, što omogućuje da se u njima slušaju tranzistorski prijemnici. Glavni problem s primenom radio-talasa iz ovog područja za prenos podataka jeste njihov mali propusni opseg [jednačina (2-3)].

U područjima visokih (HF) i vrlo visokih (VHF) frekvencija, talase koji se prostiru površinom apsorbuje tlo. Međutim, talasi koji dosegnu jonosferu - sloj naelektrisanih čestica koji okružuje Zemlju na visini od 100 do 500 km - odbijaju se od nje i vraćaju na Zemlju, kao na slici 2-12(b). U određenim atmosferskim prilikama, signali se tako mogu odbijati više puta. Radio-amateri koriste ovo talasno područje da bi međusobno komunicirali na velikim rastojanjima. I vojska za svoje potrebe koristi područja visokih i vrlo visokih frekvencija.



Slika 2-12. (a) U područjima vrlo niskih, niskih i srednjih frekvencija, radio-talasi prate krivinu zemljine površine, (b) U području visokih frekvencija, oni se odbijaju od jonosfere.

2.3.3 Prenos podataka mikrotalasima

Iznad frekvencije od oko 100 MHz, talasi se prostiru skoro pravolinijski i zato se mogu fokusirati. Koncentrisanjem sve energije u uzak snop pomoću parabolične antene (slične popularnom satelitskom tanjiru) postiže se mnogo bolji odnos signala i šuma, ali predajna i prijemna antena moraju biti strogo u liniji. Osim toga, ova usme- renost omogućava da svaki od niza međusobno bliskih predajnika bez ometanja komunicira sa svojim prijemnikom iz

niza međusobno bliskih prijemnika, pod uslo- vom da se poštuje mimimalno rastojanje između dva predajnika (prijemnika) u grupi. Pre uvođenja optičkih kablova, opisani mikrotalasni primopredajnici decenijama su predstavljali samu srž telefonskih komunikacija na velikim rastojanjima. U stvari, kompanija MCI, jedan od glavnih konkurenata korporacije AT&T, nakon što se oslobodila državne uprave, izgradila je čitav sistem mikrotalasnih komunikacija, sa antenskim tornjevima na međusobnom rastojanju od nekoliko desetina kilometara. Samo ime kompanije podseća na njenu misiju (MCI je skraćeno od Microwave Communications, Inc. - Združene mikrotalasne komunikacije). Kasnije je kompanija MCI prešla na optička vlakna i spojila se s WorldComom.

Pošto se mikrotalasi prostiru pravolinijski, zbog zakrivljenosti zemlje tornjevi ne smeju biti međusobno previše udaljeni (pomislite na vezu između San Franciska i Amsterdama). Sto su tornjevi viši, rastojanje između njih može biti veće. To rastojanje se povećava približno s kvadratnim korenom visine tornjeva. Tornjevi visine 100 m mogu se postavljati na međusobnom rastojanju od 80 km.

Za razliku od radio-talasa niskih frekvencija, mikrotalasi loše prolaze kroz zidove. Osim toga, i kada se talasi dobro usmere na predajniku, oni se ipak na putu mogu ra suti. Neki talasi se mogu odbiti od niskih atmosferskih slojeva i zato na cilj stići ka snije od direktnih talasa. Zakasneli talas može da dođe u suprotnu fazu s direktnim talasom i da ga tako poništi. Taj efekat se naziva **slabljenje zbog različitih putanja** (engl. *multipath fading)* i često stvara ozbiljne probleme. On zavisi od frekvencije i menja se s vremenskim prilikama. Neki operateri stalno drže u rezervi 10 procenata svojih kanala da bi na njih mogli da se prebace kada zbog privremenog slabljenja potpuno nestane signal na nekoj frekvenciji.

Narastajuće potrebe za komunikacijama teraju operatere da koriste sve više i više frekvencije. Danas se uobičajeno koriste područja do 10 GHz, ali se na frekvencijama od oko 4 GHz javlja nov problem: apsorpcija elektromagnetnih talasa u vodi. Te tala- se iz santimetarskog područja apsorbije najobičnija kiša. Efekat bi se dobro mogao iskoristiti kao spoljna mikrotalasna rerna za termičku obradu ptica u letu, ali u tele komunikacijama samo pravi probleme. Slično postupku koji se primenjuje pri slabljenju signala zbog različitih putanja, i ovde je jedino rešenje da se isključe veze pod kišom i da se signal pošalje zaobilaznim putem.

Sve u svemu, mikrotalasne komunikacije se u tolikoj meri koriste za telefoniju na velikim rastojanjima, za mobilnu telefoniju, televiziju i druge svrhe, daje ponestalo raspoloživih talasnih područja. Prenos podataka mikrotalasima ima više značajnih prednosti u odnosu na prenos optičkim vlaknima. Glavna je to što se ne mora tražiti pravo prolaska, već je dovoljno kupiti malu parcelu na svakih 50 km i na nju postaviti mikrotalasnu antenu da bi se zaobišao postojeći telefonski sistem i uspostavila direktna komunikacija. Na taj način je MCI tako brzo postala značajna dugolinijska telefonska kompanija. (Kompanija Sprint sledila je drugu logiku: nju su osnovale Južnopacifičke železnice koje su već imale veliki deo prava prolaska, tako da su optički kablovi samo sledili železničke šine.)

Mikrotalasna tehnologija je i srazmerno jeftina. Izgradnja dva jednostavna tornja (možda samo dva stuba sa po četiri držača) i postavljanje antena na svaki od njih može da ispadne jeftinije od zakopavanja 50 km dugačkog optičkog kabla koji se provlači kroz gusto gradsko područje ili ide preko planinskih vrhova, a može da bude jeftinije i od iznajmljivanja optičkog kabla od telefonske kompanije, naročito ako još nije naplatila sav bakar koji je

zamenila optikom.

Pravila dodeljivanja frekventnih područja

Da bi se izbegla potpuna zbrka, postoje državni i međudržavni dogovori o tome lco može da koristi koje frekvencije. Pošto svako želi veću brzinu prenosa, svi žele i vec'i deo spektra. Državne vlade dodeljuju frekvencije za amplitudno i frekventno moduli- san radio, za televiziju i mobilnu telefoniju, kao i telefonskim kompanijama, policiji, pomorskom saobraćaju, službama za navigaciju, armiji, vladi i mnogim drugim konkurentskim korisnicima. Dodeljivanje frekvencija na svetskom planu pokušava da koordinira jedna agencija sektora ITU-R (WARC), kako bi se mogli proizvoditi uređaji koji rade u različitim državama. Države, međutim, nisu vezane preporukama organizacije ITU-R, a i sama Američka savezna komisija za komunikacije (FCC) koja do- deljuje frekvencije unutar SAD, povremeno odbacuje te preporuke (obično zato što bi tada trebalo da neka grupa moćnika ustupi deo svog frekventnog područja).

Čak i kada se deo frekventnog područja nameni za određenu svrhu, na primer, za mobilnu telefoniju, nastaju dodatni problemi oko toga koji će operater koristiti koju frekvenciju. Dosad su za odlučivanje o tome korišćena tri algoritma. Najstariji, tzv. konkurs za izbor lepotice (engl. *beauty contest*), zahteva da svaki operater objasni zašto baš njegov predlog najbolje služi javnim interesima. Vladini zvaničnici tada odlučuju koja im se od lepih pričica najviše dopada. Kada „zvanična nagrada“ na konkursu vredi više milijardi dolara, onda na površinu isplivavaju mito, korupcija, nepotizam itd. Čak i kada neki pošten i skrupulozan zvaničnik uvidi da će strana kompanija posao uraditi bolje od ijedne domaće kompanije, svestan je da će taj stav teško moći da odbrani.

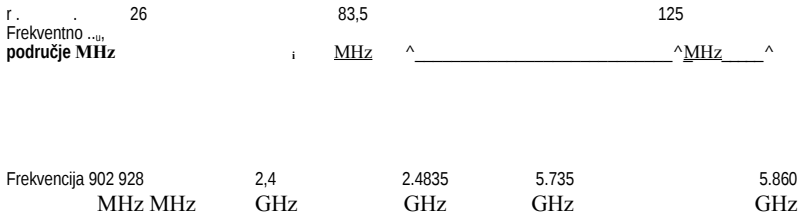
Navedeno zapažanje izrodilo je algoritam 2 - izvlačenje pobednika kockom između zainteresovanih kompanija. Ovde je problem to što se za izvlačenje mogu prijaviti i kompanije koje nisu zainteresovane za korišćenje eventualno dobijenih frekvencija. Ako, recimo, pobedi restoran brze hrane ili neki lanac prodavnica obuće, svaki od njih može da frekvencije preproda telekomunikacionim kompanijama uz dobra zaradu i bez ikakvog rizika.

Mnogi su kritikovali mogućnost da nagradu dobije kompanija koja je naizgled svuda prisutna, ali je u stvari beznačajna, što je dovelo do algoritma 3 - licitacije frekventnih područja do trenutka kad neko ponudi najvišu cenu. Kada je Engleska 2000.
godine izvršila aukciju frekvencija neophodnih za treću generaciju mobilnih telefona, vlada je očekivala da prihoduje oko 4 milijarde dolara. Zaradili su u stvari oko 40 milijardi jer su se telekomunikacione kompanije nadmetale do krvi, smrtno uplašene da će biti izgurane s tržišta mobilne telefonije. Izgled na dobra zaradu naveo ih je da drže i sopstvene aukcije, što je funkcionisalo, ali je i neke od telekomunikacionih kompanija ostavilo u dugovima koji se graniče s bankrotstvom. I u najboljim slučajevima, trebalo je više godina da se kompanija povrati od troškova licenciranja.

Potpuno drugačiji pristup od dodeljivanja frekvencija jeste da se one uopšte ne do- deljuju. Dakle, da se pusti da svako po volji emituje, ali da se snaga emitovanja ograniči na tako mali radijus da emisije ne ometaju jedna drugu. U tom smislu, većina zemalja je odvojila određena talasna područja, zvana **industrijska, naučna i medicinska područja frekvencija (engl.** *Industrial, Scientific, Medical - ISM)* za nelicen- cirano korišćenje.

Daljinski upravljači za garažna vrata, bežični telefoni, igračke teledirigovane radiom, bežični miševi i brojni drugi bežični kućni aparati koriste ISM područja. Da bi se smanjilo međusobno ometanje ovih uređaja, FCC obavezuje da svi uređaji u ISM područjima koriste tehnike širenja spektra. Slična pravila postoje i u zemljama izvan SAD.

ISM područja pomalo variraju od jedne do druge zemlje. U Sjedinjenim Državama, na primer, uređaji snage ispod 1 W mogu da koriste područja navedena na slici 2-13 bez FCC licence. Područje na 900 MHz radi najbolje, ali je pretrpano i nije svuda raspoloživo. Područje oko 2,4 GHz raspoloživo je u većini zemalja, ali je podložno interferenciji sa zračenjem mikrotalasnih rerni i radarskih instalacija. Bluetooth i neke od bežičnih mreža 802.11 rade u ovom području. Područje oko 5,7 GHz je novo i srazmerno nerazvijeno, pa je oprema za njega skupa. Međutim, pošto ga koristi mreža 801.11a, ubrzo će postati popularnije.



Slika 2-13. ISM područja u SAD.

2.3.4 Infracrveni i milimetarski talasi

Infracrveni i milimetarski talasi široko se koriste za bežične komunikacije kratkog dometa. Na primer, daljinski upravljači za televizore, video-rikordere i stereo-uređaje koriste infracrvene talase. Oni se srazmerno lako usmeravaju, a uređaji za njihovo generisanje i prijem su jeftini, ali imaju i ozbiljnu manu: ne prolaze kroz čvrsta tela (stanite između daljinskog upravljača i televizora i proverite da li on još uvek radi). U načelu, kada se od dugih radio-talasa krećete prema vidljivoj svetlosti, oni se sve više ponašaju kao svetlost, a sve manje kao radio-talasi.

S druge strane, to što infracrveni talasi loše prolaze kroz čvrsta tela takođe je i dobrodošlo. To znači da infracrveni sistem u jednoj prostoriji zgrade neće ometati isti takav sistem u susednim prostorijama ili zgradama: nećete moći da petljate po komšijskom televizoru pomoću svog daljinskog upravljača. Osim toga, bezbednost infracrvenih sistema od spoljnog prisluškivanja veća je upravo iz istog razloga. Prema tome, za razliku od radio-sistema za koji se mora dobiti zvanična dozvola za rad izvan ISM područja, za infracrvene sisteme to nije potrebno. Komunikacija infracr- venim talasima ima ograničenu primenu za povezivanje komponenata računara, ali je i tu prevazilaze druge vrste komuniciranja.

2.3.5 Prenos podataka vidljivom svetlošću

Svetlosna signalizacija poznata je stolećima. Pol Rever je koristio binarnu svetlo- snu signalizaciju pre svoje čuvene jahačke ture (Paul Revere, 1735-1818, američki patriota koji je 18. aprila 1775, jašući iz Bostona za Leksington upozorio revolucionare na približavanje britanskih trupa i na taj način ih pripremio za čuvenu bitku kojom je započeo Rat za

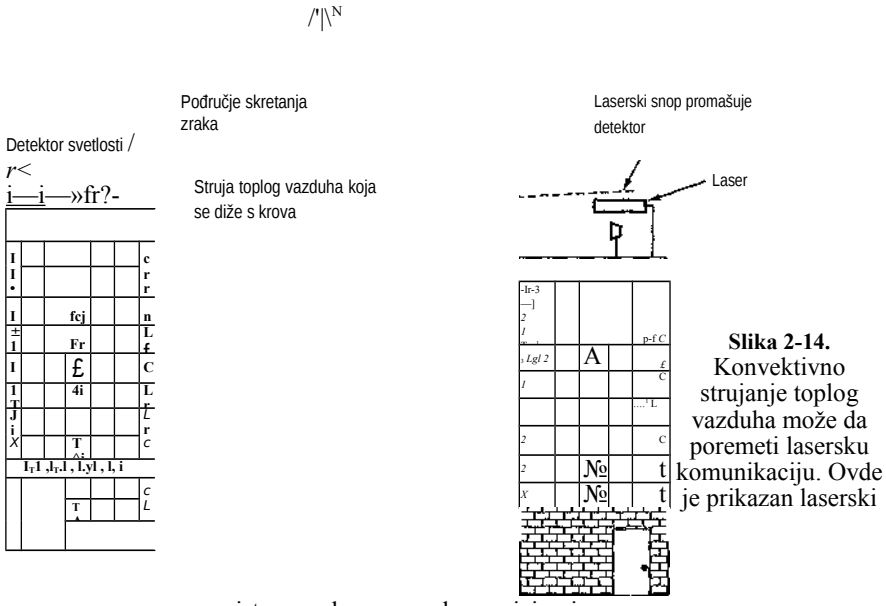
nezavisnost. „Binarna signalizacija“ sa Stare severne crkve bila je: jedna lampa ako idu kopnom, dve lampe ako idu morem. Ovaj podvig je 1863. opevao Longfelou u svojoj poemi „Paul Revere’s Ride“ - prim. prev.). Savremenija primena signalizacije vidljivom svetlošću ogleda se u povezivanju lokalnih mreža u dve zgrade pomoću laserskih uređaja na kr ovovima. Koherentni laserski zrak je po prirodi usme- ren, tako da na svakoj od dve zgrade mora postojati i laser i fotodetektor. Takav sistem ima veoma veliku propusnu moć i izuzetno je jeftin. Srazmemo se lako instalira i za razliku od mikrotalasa, za njega nije potrebno tražiti zvaničnu dozvolu vlasti.

Velika prednost lasera - njegov veoma uzak snop - ovde je i njegova mana. Pogoditi laserskim snopom širine 1 milimetra metu veličine glave čiode na rastojanju od 500 metara bio bi ozbiljan zadatak i za Jasnu Šekarić. Zbog toga se u sistem obično ugrađuju rasipna sočiva koja proširuju snop.

Nedostatak laserskog snopa je njegova nemogućnost da prodre kroz kišu ili maglu, ali tokom sunčanih dana on obično radi odlično. Međutim, autor ove knjige je prisustvovao skupu u modernom evropskom hotelu u kome su organizatori jednu prostoriju opremili terminalima za čitanje e-pošte. Pošto je lokalna PTT služba odbila da izda i instalira brojne telefonske priključke na kratak rok od 3 dana koliko je trajao skup, organizatori su na krov hotela postavili laser i usmerili ga na zgradu svog univerziteta udaljenu nekoliko kilometara. Sistem su isprobali tokom noći uoči skupa i on se pokazao savršenim. U 9 sati sledećeg divnog sunčanog jutra sistem je potpuno zakazao i nije se oporavio čitavog dana. Iste večeri organizatori su ga ponovo pažljivo isprobali i on je ponovo radio nikako drugačije do savršeno. Međutim, potpuno ista priča ponovila se i sledeća dva dana.

Organizatori su tek posle završenog skupa uspeli da otkriju u čemu je stvar. Tokom dana, dok je sunce zagrevalo krov, s njega se podizala struja toplog vazduha (slika 2-14). Ti vazdušni vrtlozi stalno su skretali zrak tako daje on samo slučajno mogao da pogodi detektor. Zbog takvih atmosferskih smetnji čini nam se da zvezde trepere, a

astronomi podižu teleskope na najviše vrhove planina u težnji da koliko je god moguće „izađu“ iz atmosfere. Isti efekat je odgovoran za „lelujanje“ pejzaža iznad autoputa tokom vrelog letnjeg dana ili ako ga zimi posmatramo kroz prozor iznad toplog radijatora.



Slika 2-14. Konvektivno strujanje toplog vazduha može da poremeti lasersku komunikaciju. Ovde je prikazan laserski sistem za dvosmemo komuniciranje.

2.4 KOMUNIKACIONI SATELITI

Tokom pedesetih i početkom šezdesetih godina, vršeni su pokušaji da se uspostavi komunikacija pomoću metaliziranih meteoroloških balona koji bi odbijali signale. Nažalost, odbijeni signali bili su previše slabi za bilo kakvu praktičnu primenu. Tada je Američka mornarica uočila da na nebu postoji neka vrsta stalnog „meteorološkog balona“ i izgradila sistem za komuniciranje između brodova na moru i kopnenih stanica koji se zasnivao na odbijanju signala od površine Meseca.

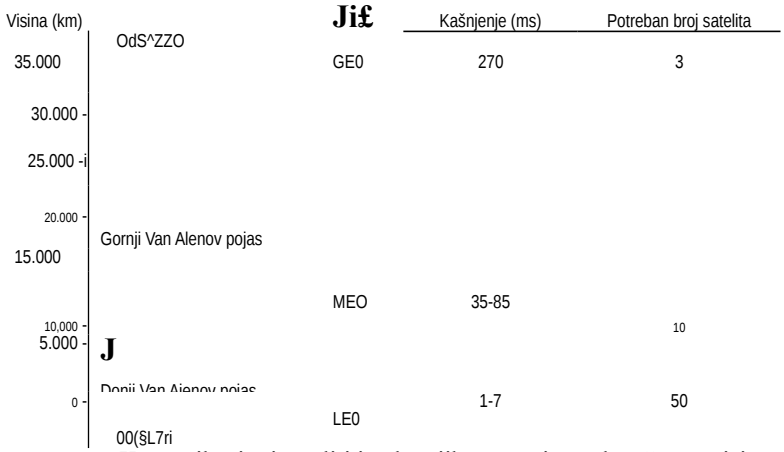
Dalji napredak na polju nebeskih komunikacija nije zapažen sve do lansiranja prvog komunikacionog satelita. Osnovna razlika između veštačkog i prirodnog satelita ogleda se u tome što veštački satelit može da pojača primljeni signal pre nego što ga ponovo pošalje. Na taj način je jedan naučno-tehnički poduhvat koji je uglavnom demonstrirao mogućnosti, prerastao u moćan komunikacioni sistem.

Komunikacioni sateliti imaju izvesna zanimljiva svojstva zbog kojih su privlačni za mnoge primene. U njegovom najjednostavnijem obliku, komunikacioni satelit možete zamisliti kao veliki repetitor na nebu. On u stvari sadrži više **transpondera** (engl. *transponders)* od kojih svaki osluškuje određeni deo elektromagnetnog spektra,

pojačava primljeni signal i ponovo ga emituje na drugoj frekvenciji da bi izbegao interferenciju s dolaznim signalom. Snop emitovanih signala može da bude širok i da pokriva priličan deo zemljine površine ili da bude uzan i da pokriva područje od samo nekoliko stotina kilometara u prečniku. Ovaj način rada se naziva **režim savijene cevi** (engl. *bent pipe)*.

Prema jednom od Keplerovih zakona, period obilaska satelita proporcionalan je poluprečniku orbite na stepen 3/2. Što je viša orbita satelita, duži je i period njegovog obilaska. U blizini zemljine površine period je oko 90 minuta. Zbog toga sateliti niske orbite brzo odlaze s vidika, tako da ih je za stalno pokrivanje određenog područja potreban veliki broj. Na visini od oko 35.800 km, period je 24 časa. Na visini od 384.000 km period iznosi oko mesec dana, što može da potvrdi svako ko je iole pažljivije po- smatrao kretanje Meseca.

Period obilaska satelita oko Zemlje je važan, ali nije i jedino što treba uzeti u obzir kada se razmatra visina satelita. Ovde konkretno mislimo na prisustvo Van Alenovih pojaseva slojeva naelektrisanih čestica koje na okupu drži zemljino magnetno polje. Satelit koji bi se kretao kroz njih ubrzo bi bio razoren udarima ovih visokoenergetskih čestica. Zbog toga se sateliti smeštaju u tri bezbednija visinska područja. Ta područja i neke osobine satelitskih sistema u njima prikazana su na slici 2-15. U nastavku ćemo kratko opisati satelite koji naseljavaju svako od tri područja.



Slika 2-15. Komunikacioni sateliti i neka njihova svojstva, kao što su visina iznad zemljine površine, kašnjenje povratnog signala i broj satelita potreban za pokrivanje cele planete.

2.4.1 Sateliti s **geostacionarnom** orbitom

Artur Klark, pisac naučne fantastike, izračunao je još 1945. godine da bi satelit koji kruži iznad ekvatora na visini od 35.800 km izgledao kao prikovan za nebeski svod, tako da bi se mogao pratiti stacionarnim antenama (Clarke, 1945). Zatim je opisao potpun komunikacioni sistem **geostacionarnih** satelita (s ljudskom posadom), njihove orbite, panele sa solarnim ćelijama, radio-frekvencije i postupke lansiranja. On je, nažalost, zaključio da bi takav sistem

bio nepraktičan zbog nemogućnosti da se osetlji- vi, energetski zahtevni pojačivači sa elektronskim cevima bezbedno dovedu u orbitu, pa ideju nije dalje razvijao, iako je na njoj zasnovao nekoliko naučnofantastičnih priča.

Pronalazak tranzistora je sve promenio i prvi veštački komunikacioni satelit, Tel- star, lansiran je jula 1962. Otada su telekomunikacioni sateliti postali unosan posao vredan više milijardi dolara i istovremeno jedini visokoprofitabilan aspekt svemirskih istraživanja. Sateliti koji kruže na velikoj visini dobili su naziv GEO (engl. *Geostationary Earth Orbit)* sateliti.

Uz današnju tehnologiju, da bi se izbegla interferencija, najbolje je da se geosta- cionarni sateliti ne raspoređuju oko ekvatorijalne ravni od 360 stepeni na rastojanjima manjim od 2 stepena. Uz međusobno rastojanje od 2 lučna stepena, istovremeno ih na nebu može biti samo 360/2 = 180. Međutim, svaki satelitski transponder može da koristi više frekvencija i različitu polarizaciju da bi proširio propusni opseg.

Da bi sprečila totalnu zbrku na nebu, organizacija ITU dodeljuje pojedine intervale orbite. Njene odluke su izuzetno ispolitizirane jer neke države koje su takoreći juče izišle iz kamenog doba zahtevaju „svoj“ interval orbite (ne bi li ga skupo prodale). Druge, međutim, smatraju da se državni (svemirski) prostor ne prostire uvis sve do Meseca i da nijedna država ne sme da polaže pravo na orbitu koja prolazi iznad njene teritorije. Temperatura podiže i to što se sateliti ne koriste samo za komercijalne tele- komunikacione svrhe: TV stanice, vlade i armije pojedinih država takođe zahtevaju svoj deo orbitalnog kolača.

Savremeni sateliti mogu da budu prilično veliki, mase do 4000 kg i snage od više kilovata, koju generišu solarni paneli. Gravitaciona sila Sunca, Meseca i velikih planeta teži da ih pomeri iz dodeljenih orbitalnih intervala i da im promeni orijentaciju, zbog čega sateliti imaju ugrađene raketne motore za korekciju putanje. Takve korek- tivne aktivnosti nazivaju se **održavanjem stanice u orbiti.** Međutim, kada se (obično za oko 10 godina) potroši gorivo za raketne motore, satelit počinje da bespomoćno leluja pa se mora isključiti. Tokom vemena, njegova orbita postaje sve niža, dok ne uđe u gornje slojeve atmosfere gde sagori ili se ponekad i sunovrati na zemlju.

Orbitalni intervali nisu i jedini razlog za prepirke. To su i frekvencije, jer emisija sa satelita može da ometa postojeće korisnike mikrotalasnih frekvencija. Zbog toga je organizacija ITU dodelila određena frekventna područja za satelitski prenos, od kojih su glavna prikazana na slici 2-16. Za komercijalni satelitski prenos prvo je dodeljeno područje C s dve frekvencije: niža je za prenos od satelita, a viša za prenos ka satelitu. Da bi se saobraćaj mogao odvijati istovremeno u oba smera, bila su potrebna dva nezavisna kanala. Ti kanali su već zagušeni jer ih koriste i javne telekomunikacione službe za zemaljske mikrotalasne veze. Područja L i S dodata su 2000. godine na osnovu međunarodnog sporazuma. Međutim, ona su uska i pretrpana.

Sledeće područje viših frekvencija raspoloživo komercijalnim telelcomunikacionim službama jeste područje Ku (K donje, engl. *K under).* Ono (još uvele) nije pretrpano, a na ovoj frekvenciji sateliti se mogu raspoređivati na međusobnom rastojanju od 1 lučnog stepena. Međutim, ovde se javlja dragi problem: kiša. Voda je izuzetno dobar apsorber ovako kratkih mikrotalasa. Na sreću, velike nepogode su obično lokalizovane na usko područje, pa se problem može zaobići ako se umesto jedne zemaljske stanice upotrebi nekoliko međusobno udaljenih stanica, naravno, uz povećane troškove zbog više antena, više kablova,

i elektronike potrebne za pravovremeno prebacivanje rada s jedne stanice na drugu. Za komercijalni satelitski saobraćaj dodeljeno je i područje Ka (K gornje, engl. *K above),* ali je oprema za njegovo korišćenje još uvek skupa. Osim navedenih komercijalnih područja, postoje i mnoga zvanična državna i vojna područja frekvencija.

Područje	Veza od satelita	Veza ka satelitu	Propusni opseg	Problemi
L	1,5 GHz	1,6 GHz	15 MHz	Uzak propusni opseg; pretrpanost
S	1,9 GHz	2,2 GHz	70 MHz	Uzak propusni opseg; pretrpanost
C	4,0 GHz	6,0 GHz	500 MHz	Interferencija sa zemaljskim vezama
Ku	11 GHz	14 GHz	500 MHz	Kiša
Ka	20 GHz	30 GHz	3500 MHz	Kiša; cena opreme

Slika 2-16. Osnovna frekventna područja za satelitski prenos.

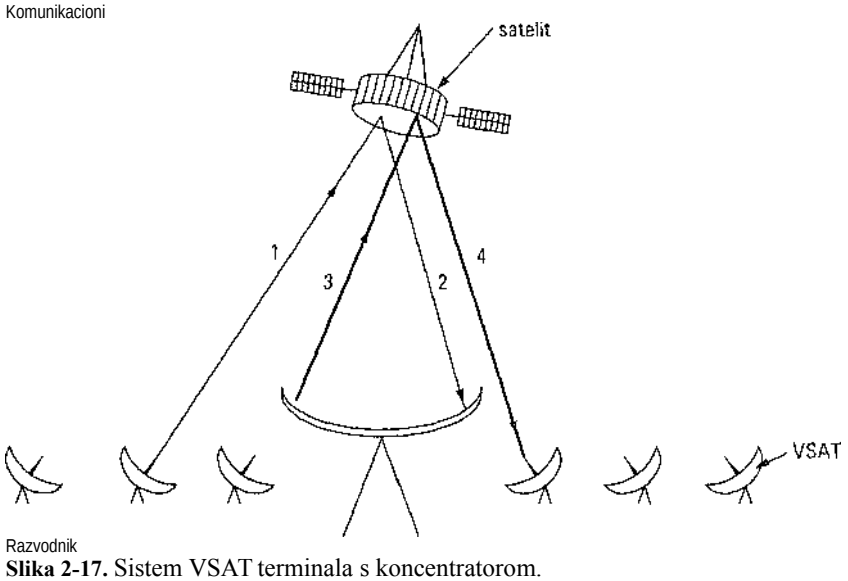
Savremeni satelit nosi oko 40 transpondera, svaki propusnog opsega 80 MHz. Transponderi obično rade u režimu savijene cevi, ali najnoviji sateliti mogu da u iz- vesnoj meri obrade signal i tako ponude složenije mogućnosti rada. Na prvim satelitima, transponderi su bili statički podeljeni na kanale: propusni opseg je jednostavno deljen na fiksna frekventna područja. Danas se svaki satelitski snop deli na vremenske intervale, koje korisnici naizmenično upotrebljavaju. Dve takve tehnike (multipleksi- ranje podelom frekvencije i multipleksiranje podelom vremena) obradićemo detaljno u drugom delu ovog poglavlja.

Prvi geostacionarni sateliti imali su jedinstven prostorni snop koji je svoj **„otisak“** (engl.*footprint)* ostavljao na oko 1/3 zemljine površine. Uz neverovatan pad cene, veličine i energetskih zahteva koje je donela mikroelektronika, postale su moguće i daleko složenije strategije emitovanja. Sada svaki satelit ima više antena i više transpondera. Svaki snop emitovan sa satelita može se usmeriti na malo geografsko područje, tako da se istovremeno može ostvariti više prenosa ka satelitu i od njega. Ovi, tzv. **usmereni snopovi** (engl. *spot beams)* ostavljaju eliptičan otisak i mogu da pokriju područje preč- nika samo nekoliko stotina kilometara. Komunikacioni satelit namenjen za SAD obično ima jedan širok snop koji pokriva glavnu teritoriju u centralnom delu Severne Amerike i dva usmerena snopa koji komuniciraju sa Aljaskom i Havajima.

Najnovije dostignuće na ovom polju jesu jeftine mikrostanice - **terminali s vrlo uskim emisionim snopom** (engl. *Very Small Aperture Terminals, VSATs)* (Abramson, 2000). Oni imaju antene prečnilca 1 metra ili manje (za razliku od antene prečnilca 10 metara za standardne GEO sisteme) i emituju snagom od oko 1 vata. Veza ka satelitu je u načelu dobra kada radi brzinom 19,2 kb/s, ali veza od satelita dostiže brzinu 512 kb/s i više. Ova tehnologija se koristi za jednosmeran prenos u satelitskoj TV difuziji.

U mnogim VSAT sistemima, mikrostanice nemaju dovoljnu snagu da direktno ko- municiraju jedna s drugom (naravno, preko satelita), već se saobraćaj između njih odvija preko tzv. **razvodnika** (engl. *huh)* - zemaljske antene visokog učinka (slika 2-17). U ovakvom režimu rada, veliku antenu i snažan pojačivač ima ili pošiljalac ili primalac.

Prednost male snage mikrostanica ima i svoju cenu - u takvom režimu rada kašnjenje signala je veće.



Slika 2-17. Sistem VSAT terminala s koncentratorom.

Sistem VSAT ima veliku potencijalnu primenu u seoskim područjima. Ta mogućnost nije široko sagledana uprkos činjenici da polovina ljudi na našoj planeti živi na udaljenosti od barem sat hoda do najbližeg telefona. Razvlačenje telefonskih žica do hiljada zabačenih sela uveliko prevazilazi mogućnosti većine država Trećeg sveta, ali instaliranje metarskih VSAT tanjirića napajanih solarnim ćelijama često je izvodljivo. Sistem VSAT obezbeđuje tehnologiju koja može da poveže sve ljude na svetu.

Komunikacioni sateliti imaju više osobina koje ih potpuno razlikuju od zemaljskih veza tipa „od tačke do tačke“. Najpre, iako signali ka satelitu i od njega putuju brzinom svetlosti (oko 300.000 km/s), velika udaljenost GEO satelita dovodi do značajnog kašnjenja. U zavisnosti od rastojanja između korisnika i zemaljske stanice i visine satelita iznad horizonta, vreme putovanja od jednog do drugog kraja iznosi između 250 i 300 ms. Tipična vrednost je 270 ms (.540 ms za sistem VSAT s razvodnikom).

Poređenja radi, zemaljske mikrotalasne veze kasne oko 3 ps/km, a veze ostvarene koaksijalnim ili optičkim kablovima kasne *5* ps/km. Veze vođene kroz materijalni medijum sporije su jer se elektromagnetni talasi brže prostiru kroz vazduh nego kroz čvrsta tela.

Drugo važno svojstvo satelita je to što su oni po prirodi medijum za neusmereno (difuzno) emitovanje. Slanje signala hiljadama stanica unutar otiska satelitskog snopa ne košta ni pare više od slanja signala samo jednoj stanici. Za neke primene ovo svojstvo je veoma dobrodošlo. Na primer, možemo da zamislimo satelit koji difuzno emituje popularne Web strane u privremenu memoriju brojnih racunara rasutih po širem području. Čak i kada se difuzno emitovanje simulira vezama od tačke do tačke, satelitsko emitovanje može da bude mnogo jefitinije. S druge strane, sa aspekta bez- bednosti i privatnosti, sateliti su potpuna

katastrofa: svako može sve da čuje. Ako se insistira na bezbednosti, neophodno je šifrovati podatke.

Svojstvo satelitskog prenosa je i to što cena prenete poruke ne zavisi od udaljenosti. Prekookeanska veza košta isto kao i veza s komšijom preko puta. Satelitski pre- nos je takođe izuzetno imun na greške i sistem se može uspostaviti gotovo trenutno, što je bitno za vojne komunikacije.

2.4.2 Zemljini sateliti srednje orbite

Na mnogo manjim visinama, između dva Van Alenova pojasa, nalaze se **sateliti srednje orbite** (engl. *Medium-Earth Orbit, MEO)*. Posmatrani sa zemlje, oni pomalo menjaju meridijan po kome obilaze Zemlju za 6 časova. Zbog svog relativnog kretanja u odnosu na Zemlju, moraju se pratiti. Pošto kruže na manjoj visini od GEO satelita, na Zemlji ostavljaju manji otisak, a zemaljska stanica može da bude manje snage. Trenutno se ne koriste za telekomunikacije, pa ih nećemo dalje razmatrati. Primer MEO sistema su 24 satelita **globalnog sistema za pozicioniranje** (engl. *Global Positioning System, GPS),* koji kruže na visini od 18.000 km.

2.4.3 Zemljini sateliti niske orbite

Na još manjim visinama **kruže sateliti niske orbite** (engl. *Low-Earth Orbit, LEO)*. Zbog toga što se brzo kreću, sistem koji teži da pokrije čitavu površinu Zemlje mora da ima mnogo satelita. S druge strane, pošto su sateliti vrlo blizu površine Zemlje, nisu potrebne snažne zemaljske stanice, a kašnjenje signala iznosi nekoliko milise- kundi. U ovom odeljku ćemo razmotriti tri primera ovakvih sistema, dva koja su na- menjena prenosu glasa ijedan namenjen Internetu.

Iridium

Kao što je već naglašeno, sateliti niske orbite retko su korišćeni tokom prvih 30 *godina* satelitske ere upravo zato što se naglo pojavljuju i brzo zamiču za horizont. Godine 1990. Motorola je napravila prvi korak tražeći od Savezne komisije za komunikacije (FCC) dozvolu da lansira 77 satelita niske orbite za svoj projekat Iridium (iridijum je 77. element periodnog sistema). Kasnije je plan izmenjen u smislu korišćenja samo 66 satelita, tako da bi po istoj logici projekat trebalo nazvati Dysprosium (po 66. elementu), ali je to ime previše podsećalo na ime neke bolesti. Osnovna zamisao projekta bila je da se na nebu pojavi nov satelit čim prethodni zamakne za horizont. Predlog je izazvao bezobzirne napade drugih telekomunikacionih kompanija. I odjednom, svako je želeo da lansira lanac satelita niske orbite.

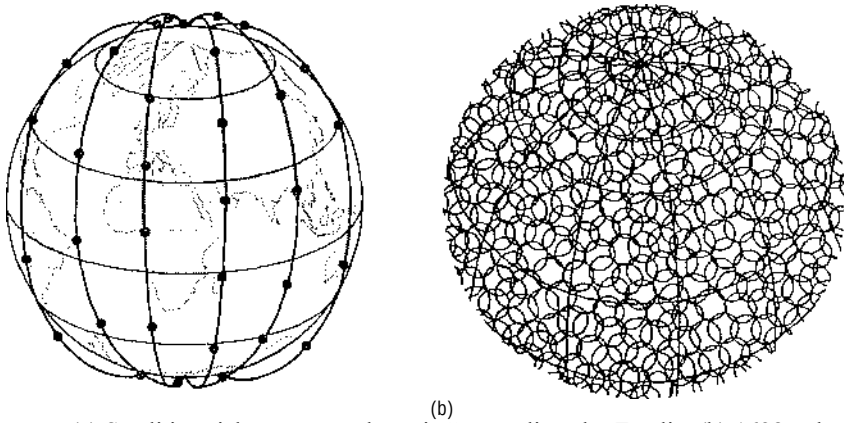
Partneri i finansije prikupljani su tokom sedam godina, a zatim su sateliti projekta Iridium lansirani 1997. godine. Komunikaciona služba je počela da radi novembra 1998. Nažalost, izostala je veća komercijalna potražnja za glomaznim satelitskim telefonima pošto se mreža mobilne telefonije neverovatno proširila počev od 1990. Iridium zato nije doneo profit i čitav projekat je ugašen avgusta 1999. u jednoj od najspektakularnijih katastrofa koje prate bankrotiranje korporacija. Sateliti i draga oprema (vedni 5 milijardi dolara) kasnije su prodati za 5 miliona dolara javnim nadmetanjem. Novi kupac je Iridium ponovo pustio u rad marta

2001.

Cilj projekta Iridium bio je, a i sada je, da obezbedi globalne telekomunikacione usluge pomoću ručnih uređaja koji se direktno povezuju s njegovim satelitima. Iridium obezbeđuje usluge prenosa glasa i podataka, pejdžinga, faksa i navigacije na svakoj tački kopna, na mora i u vazduhu. Korisnici su pomorski saobraćaj, vazduhoplovstvo i okeanske platforme za istraživanje naftnih ležišta, kao i pojedinci koji putuju delovi- ma sveta u kojima nema telekomunikacione infrastrukture (npr. u pustinjama, na planinama, u džunglama i u nekim zemljama Trećeg sveta).

Sateliti projekta Iridium kreću se na visini od 750 km, po kružnim polarnim orbitama. Oni u smeru sever-jug formiraju „ogrlice“, s jednim „zrnom“ na svaka 32 ste- pena geografske širine. Šest takvih ogrlica dovoljno je da pokrije celu Zemlju, što se bolje vidi na slici 2-18(a). Oni koji samo površno poznaju herniju, ovakav sistem mogu da porede sa ogromnim atomom disprozijuma, pri čemu Zemlja igra ulogu jezgra, a sateliti su elektroni.

Svaki satelit ima najviše 48 ćelija (usmerenih snopova). Ukupno ih je 1628 i pokrivaju čitavu Zemlju, kao na slici 2-18(b). Kapacitet svakog satelita je 3840 kanala, što sve zajedno iznosi 253.440 kanala. Neki od njih se koriste za pejdžing i navigaciju, a ostali za prenos glasa i podataka.



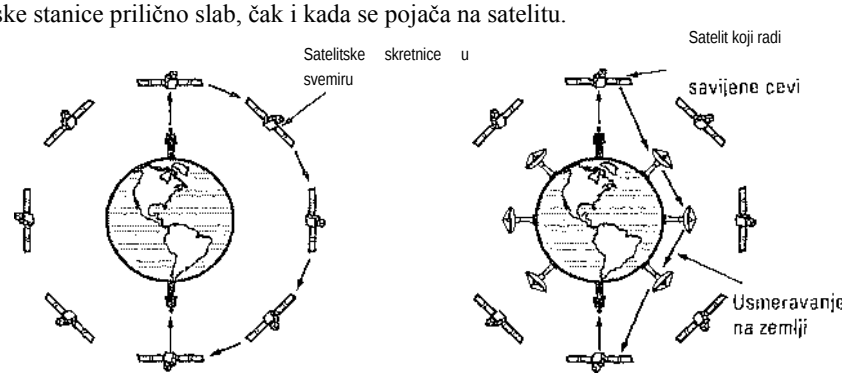
(a) (b)

Slika 2-18. (a) Sateliti projekta **Iridium** obrazuju šest ogrlica oko Zemlje, (b) 1628 pokretnih ćelija pokriva čitavu Zemlju.

Zanimljivo je da se u sistemu Iridium komunikacija između udaljenih korisnika odvija kroz svemir, pri čemu sateliti prosleđuju poruke između sebe, kao na slici 2- 19(a). Tu vidimo da pozivalac sa Severnog pola uspostavlja vezu sa satelitom neposredno iznad sebe. Poziv prosleđuju drugi sateliti i na kraju se on isporučuje na željenu adresu na Južnom polu.

Globalstar

Alternativa Iridiumu je sistem Globalstar. On ima 48 LEO satelita, ali s drugačijim komutiranjem. Dok se u Iridiumu pozivi prosleđuju od jednog do drugog satelita, zbog čega oni moraju imati složenu opremu za komutiranje, Globalstar koristi klasični režim savijene cevi. Poziv upućen sa Severnog pola, slika 2-19(b), ponovo se vraća na Zemlju, gde ga dočekuje velika zemaljska stanica locirana u Deda-Mrazovoj radionici. Poziv se tada upućuje zemaljskim linijama do stanice najbliže odredištu i na njega isporučuje talcođe pomoću savijene cevi. Prednost ovakvog sistema je to što je njegov složeniji deo uglavnom na zemlji, gde ga je lakše održavati. Isto tako, upotreba snažnih zemaljskih antena koje pojačavaju primljene signale otklanja potrebu za snažnim telefonskim aparatima. U krajnjoj liniji, telefoni emituju snagom od samo nekoliko milivata, pa je signal koji sa satelita stigne do zemaljske stanice prilično slab, čak i kada se pojača na satelitu.



<a| (b)

Slika 2-19. (a) Prosleđivanje poziva kroz svemir, (b) Prosleđivanje poziva po Zemlji.

Teledesic

Iridium je namenjen korisnicima telefona koji se nalaze na zabačenim mestima. Naš sledeći primer, Teledesic, namenjen je korisnicima Interneta širom sveta, koji stalno vape za

širim propusnim opsegom. Sistem su 1990. zamislili Krejg Mekau (Craig McCaw), pionir mobilne telefonije, i osnivač Microsofta, Bil Gejts, jer su bili razočarani tempom kojim su telefonske kompanije povećavale propusni opseg za korisnike Interneta. Cilj sistema Teledesic bio je da se za milione korisnika Interneta obezbedi istovremena veza ka satelitu brzine do 100 Mb/s, a od satelita - do 720 Mb/s, uz

korišćenje malih, fiksnih antena tipa VSAT, čime bi se potpuno zaobišao postojeći telefonski sistem. Telefonske kompanije su o tako nečem mogle samo da sanjare.

Prvobitnim projektom predviđen je sistem od 288 satelita s malim otiskom (tragom), postavljenih u 12 ravni neposredno ispod donjeg Van Alenovog pojasa, na visini od 1350 km. Kasnijim izmenama broj satelita je smanjen na 30, ali im je otisak na zemlji povećan. Predviđeno je da sistem koristi srazmemo prazno frekventno područje Ka, velikog propusnog opsega. Sistem u svemiru radi s komutiranjem paketa, pri čemu svaki satelit može da usmerava pakete ka svojim susedima. Kada korisniku zatreba propusni opseg za slanje paketa, on mu se dinamički dodeljuje na zahtev, što traje oko 50 ms. Ukoliko sve protekne po planu, sistem će biti pušten u rad 2005.

2.4.4 Poređenje satelitskih i optičkih veza

Iz poređenja satelitskih i zemaljskih komunikacija mogu da se izvuku poučni zaključci. Još pre 20 godina izgledalo je potpuno izvesno da budućnost komunikacija leži u komunikacionim satelitima. Telefonski sistemi se poslednjih 100 godina nisu praktično ništa izmenili, a sva je prilika da se ništa u njima neće promeniti ni za sle- dećih 100 godina. Takvoj tehnološkoj učmalosti znatno su doprineli i zakonski propisi koji su telefonske kompanije obavezivali da obezbede kvalitetne govorne usluge po razumnim cenama (što su i obezbeđivale), dobijajući zauzvrat garantovanu zaradu. Oni koji su želeli da prenose podatke, mogli su da koriste modeme brzine 1200 kb/s. I to je otprilike bilo sve.

Kada je 1984. u SAD uvedena konkurencija između telefonskih kompanija, što se nešto kasnije dogodilo i u Evropi, situacija se potpuno izmenila. Telefonske kompanije su svoje međugradske linije počele da zamenjuju optičkim kablovima i da uvode usluge velike propusne moći, kao što je asimetrična digitalna pretplatnička linija (engl. *Asymmetric Digital Subscriber Line, ADSL)*. Takođe su prestali da veštački podižu cene međugradskim razgovorima i da od tog novca flnansiraju lokalni saobraćaj.

Najednom je izgledalo da je u zemaljskim komunikacijama na dugi rok pobedio optički kabl. Bez obzira na to, komunikacionim satelitima je ostalo značajan deo tržišta koji optički kabl nije želeo (ili nije mogao) da zauzme. Sada ćemo detaljnije pretresti to područje.

Prvo, iako samo jedno optičko vlakno u načelu ima veću propusnu moć od svih ikada lansiranih satelita, taj propusni opseg većini korisnika nije dostupan. Optička vlakna koja se sada instaliraju koriste se u telefonskim sistemima za istovremeno vođenje više međugradskih razgovora, a ne za povećavanje propusnog opsega pojedinačnim korisnicima. Kod satelita je zgodno to što svaki korisnik može da instalira antenu na krovu svog doma i da - zaobišavši telefonski sistem - dobije veliki propusni opseg. Teledesic je zasnovan na toj ideji.

Drugo područje su mobilne komunikacije. Mnogi danas žele da komuniciraju dok džogiraju, dok se voze automobilom, dok jedre ili lete. Zemaljske veze, makar i s optičkim kablovima, nisu im tu od pomoći, ali satelitske veze potencijalno jesu. Moguće je, međutim, da će kombinacija mobilnog radija i optičkog vlakna zadovoljiti većinu korisnika (ali verovatno ne one koji su na moru ili u vazduhu).

Treće područje obuhvata primene čija je suština neusmereno (difuzno) emito- vanje. Poruku poslatu sa satelita može istovremeno da primi na hiljade zemaljskih stanica. Na primer, za organizaciju koja distribuira berzanske, bankarske ili robne informacije hiljadama

dilera, satelitski sistem može da se pokaže jeftinijim od odgovarajućeg zemaljskog sistema.

Četvrto područje su komunikacije na negostoljubivoin terenu ili na mestima bez razvijene zemaljske komunikacione infrastrukture. Indonezija, na primer, ima sopstve- ni satelit za održavanje domaćeg telefonskog saobraćaja. Lansiranje jednog satelita je jeftinije od razvlačenja hiljada podmorskih kablova između 13.677 ostrva arhipelaga.

Peto tržišno područje rezervisano za satelitski prenos odnosi se na oblasti u kojima je dobij anje prava prolaska teško ili bezobrazno skupo.

Šesto, kada je potrebno da se sistem brzo uspostavi, kao u vojnim komunikacionim sistemima tokom rata, sateliti lako dobijaju trku.

Sve u svemu, izgleda da će u budućnosti glavni razvoj komunikacija ići u smeru zemaljskih optičkih kablova kombinovanih s mobilnim radiom, ali će za neke specijalizovane primene sateliti biti pogodniji. Međutim, postoji činilac koji jednako pogađa sve tehnologije: ekonomija. Iako vlakno omogućava veći propusni opseg, gotovo je izvesno da će se zemaljske i satelitske komunikacije međusobno nadmetati u pogledu cene. Ukoliko se napretkom tehnologije znatnije snizi cena lansiranja satelita (npr. ako neki budući šatl uspe da u jednom lansiranju postavi tuce satelita) ili se proširi primena satelita niske orbite, nije sigurno da će optičko vlakno pobediti na svim poljima.

2.5 JAVNA KOMUTIRANA TELEFONSKA MREŽA

Kada dva bliska računara u istoj kompaniji ili organizaciji treba da međusobno komuniciraju, često je najlakše da se povežu kablom. Lokalne mreže rade na taj način. Međutim, kada su računari na većoj razdaljini ili ih ima više, ili onda kada u cilju njihovog povezivanja treba koristiti javni put ili preći područje u tuđem vlasništvu, cena postavljanja privatnih kablova može da bude ograničavajući činilac. Osim toga, u skoro svakoj državi na svetu nelegalno je razvlačiti privatne prenosite linije preko (ili ispod) javnog područja. Upravo zbog toga projektanti mreža moraju da se oslanjaju na postojeću telekomunikacionu strukturu.

Ta struktura, naročito javna komutirana telefonska mreža (engl. *Public Switched Telephone Network, PSTN),* obično je projektovana davno, s potpuno drugim ciljem: da manje-više razumljivo prenosi ljudski glas. Njena podobnost za komunikaciju između računara često je nepredvidiva, ali se situacija na tom polju naglo menja uvođenjem optičkih vlakana i digitalne tehnologije. U svakom slučaju, telefonski sistem je tako tesno isprepletan s (regionalnim) računarskim mrežama, da mu moramo posvetiti određen prostor u ovoj knjizi.

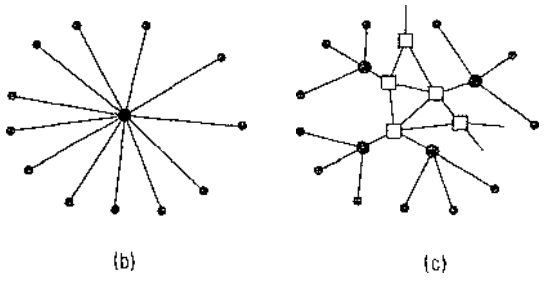
Da bismo sagledali značaj problema, uporedimo na grub, ali ilustrativan način svojstva tipične veze između dva računara ostvarene lokalnim kablom, i pomoću modema i telefonske linije. Kabl koji povezuje računare može da prenosi podatke brzinom IO[9]

b/s, možda i brže. Nasuprot tome, modemska telefonska linija radi maksimalnom brzinom 56 kb/s, tj. skoro 20.000 puta sporije. To je razlika između patke lcoja se lenjo gega livadom i rakete koja stremi Mesecu. Ako se modemska telefonska linija zameni ADSL linijom, ona će ipak biti 1000-2000 puta sporija od kablovske veze.

Problem je, naravno, u tome što su projektanti računarskih sistema navikli da rade s računarskim sistemima i kada se odjednom suoče sa sistemima čije su performanse (s njihovog gledišta) 3 do 4 reda veličine lošije, oni prirodno troše mnogo vremena i napora pokušavajući da dokuče kako da ih efikasno koriste. U narednim odeljcima opisaćemo telefonski sistem i njegov rad. Detaljnija objašnjanja potražite kod Bella- myja (2000).

2.5.1 Struktura telefonskog sistema

Ubrzo pošto je Aleksandar Grejem Bel patentirao telefon 1876. (samo nekoliko časova pre svog rivala Elise Greja), nastala je velika potražnja za njegovim izumom. Prvo tržište bilo je tržište telefonskih aparata koji su se nudili u paru, pri čemu je (jednu) žicu između njih trebalo da razvuče kupac jer su se elektroni vraćali kroz zemlju. Ako je vlasnik telefona želeo da razgovara sa *n* drugih vlasnika telefona, morao je da razvuče *n* posebnih žica ka njihovim domovima. Vremenom je gradove prekrila šuma žica razvučenih preko krovova i drveća. Odmah je postala jasna nepraktičnost sistema u kome je svaki telefon spojen sa svakim drugim telefonom, kao na slici 2-20(a).



Slika 2-20. (a) Mreža s direktnim vezama između svih telefona, (b) Centralizovana skretnica.

(c) Dvostepena hijerarhija.

Bel je, što mu treba priznati, ovo odmah uočio i osnovao Belovu telefonsku kompaniju koja je 1878. uspostavila prvu telefonsku centralu u Nju Hejvnu u državi Ko- nektikat. Od centrale su sprovedene žice do domova ili kancelarija svih korisnika. Kad je želeo nekoga da pozove, korisnik bi okrenuo ručicu aparata i aktivirao zvonce u telefonskoj kompaniji, posle čega bi ga operater ručno spojio sa željenim sagovor- nilcom. Model jedinstvene telefonske centrale prikazan je na slici 2-20(b).

Vrlo brzo su telefonske centrale po Belovom sistemu počele svuda da niču i korisnici su poželeli da vode i međugradske razgovore, pa je kompanija počela da međusobno povezuje centrale. Ponovo se pojavio prvobitni problem: nemogućnost da se

svaka centrala poveže sa svim drugim centralama, pa su uvedene telefonske centrale višeg nivoa. Uskoro su se i te centrale namnožile, pa je trebalo uvoditi centrale još višeg nivoa, kao na slici 2-20(c). Hijerarhija je tako porasla sve do petog nivoa.

Godine 1890, telefonski sistem je imao tri glavna delai telefonske centrale, sra- zmerno kratke žice između korisnika i telefonskih centrala (sada standardizovane, izolovane i upredene parice umesto prvobitnih golih jednostrukih žica, dok je druga žica bila zemlja), i dugačke veze između telefonskih centrala. Iako je svaki navedeni deo otada poboljšavan, osnovni model Belovog sistema ostao je isti više od 100 godina. Kratku tehničku istoriju telefonskog sistema naći ćete kod Hawleyja (1991).

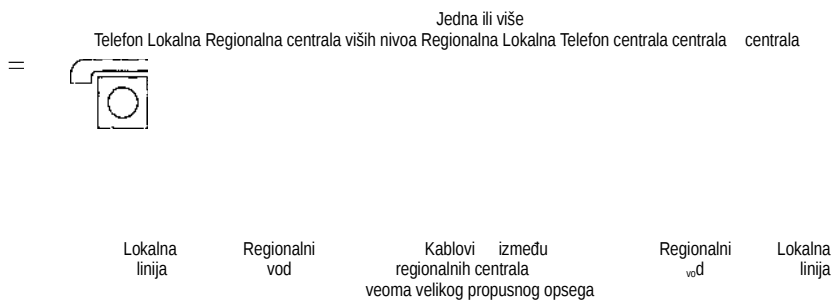
Pre raspada korporacije AT&T 1984. godine, telefonski sistem je bio organizovan kao visokoredundantna hijerarhija s više nivoa. Opis koji sledi prilično je uprošćen, pa ipak ćete steći pravu sliku. Svaki telefon je bio snabdeven s dve bakarne žice koje su vodile direktno do **lokalne telefonske centrale** (engl. *end office),* zvane kraće **lokalna centrala** (engl. *local central office).* Razdaljina od telefona do centrale obično je bila između 1 i 10 km, manja u gradovima, veća na seoskom području. Samo u Sjedinjenim Državama bilo je oko 22.000 lokalnih centrala. Dvožična veza između korisnika i centrale poznata je i kao **lokalna linija** (engl. *local loop).* Kada bi se sve lokalne veze na svetu nadovezale jedna na drugu, dobili biste žicu 1000 puta dužu od rastojanja između Zemlje i Meseca.

U jednom periodu, oko 80% kapitala korporacije AT&T bilo je sadržano u lokalnim vezama. AT&T je, u stvari, tada bila najveći rudnik bakra na svetu. Na sreću, investitori nisu znali za taj podatak. Da su znali, neki od njih bi možda kupio AT&T, stavio tačku na sve telefonske usluge unutar SAD, pokupio svu žicu i prodao je topionicama ostvai'ujući brzu zaradu.

Kada pretplatnik koji je povezan sa određenom centralom pozove drugog pretplatnika povezanog sa istom centralom, mehanizam za uključivanje u centrali uspostavlja direktnu električnu vezu između dve lokalne linije. Ta veza se ne prekida dok traje razgovor.

Kada je traženi pretplatnik vezan za drugu lokalnu centralu, mora se postupiti drugačije. Od svake lokalne centrale vodi više linija ka obližnjim centralama višeg nivoa, tzv. **regionalnim telefonskim centralama** (engl. *toli offices),* zvanim i **tandem centralama** (engl. *tandem offices)* ako se nalaze u istom lokalnom području. Te linije se zovu **regionalni vodovi** (engl. *toli connecting trunks).* Ako od lokalnih centrala pozivaoca i pozvanog korisnika vode magistale ka istoj regionalnoj centrali (što je verovatno kada su korisnici srazmerno blizu), veza se može uspostaviti u regionalnoj centrali. Telefonska mreža koja se sastoji samo od telefona (male tačke), lokalnih centrala (veće tačke) i regionalnih centrala (kvadratići) prikazana je na slici 2-20(c).

Ako pozivalac i potencijalni sagovornilc nemaju istu regionalnu centralu, veza se mora uspostaviti na nekom višem nivou. **Regionalne centrale svih nivoa** međusobno komuniciraju pomoću odgovarajućih **kablova** (engl. *intertoll trunks, interojfice trunks).* Broj centrala raznih nivoa i njihova topologija (npr. mogu li dve centrale istog nivoa da se direktno po vežu ili se veza mora ostvariti preko centrale višeg nivoa?) varira od jedne do drage države, zavisno od gustine telefonskog saobraćaja u njima. Slika 2-21 prikazuje kako se može uspostaviti telefonska veza na srednjim rastojanjima.



Slika 2-21. Tipično uspostavljanje veze srednje dužine.

Za telekomunikacije se koriste veoma različiti prenosni medijumi. Lokalne linije su danas upredene parice 3. kategorije, iako su na počecima telefonije preko bandera bile rastezane gole žice, međusobno udaljene 25 cm. Za vezu između centrala široko se koriste koaksijalni kablovi, mikrotalasi i, naročito, optički kablovi.

Ranije su podaci kroz telefonski sistem prenošeni analogno, pri čemu je glas što vernije preslikavan u pulsirajući električni napon koji je prenošen sa izvora do odredišta. S napretkom tehnologije optičkih vlakana, digitalne elektronike i računara, sve linije i centrale su sada digitalne, osim lokalne koja je ostala jedini analogni deo u sistemu. Prenos podataka u digitalnom obliku je pogodniji jer se tu ne mora tačno re- produkovati oblik analognog talasa pri prolasku kroz razne pojačivače na dugom putu. Dovoljno je da se tačno prenesu jedinice i nule. Zbog toga je digitalni prenos podataka pouzdaniji od analognog. Takođe je jeftiniji i lakše se održava.

Sve u svemu, telefonski sistem se sastoji od tri glavne komponente:

1. Lokalnih linija (analogna upredena parica koja dolazi do stanova korisnika i poslovnih zgrada).
2. Telefonskih kablova (digitalnih optičkih kablova koji povezuju centrale).
3. Centrala (u kojima se poziv prebacuje s jednog kabla na drugi).

Posle kratke epizode o politici telefonije, vratićemo se detaljnije na svaku od navedenih komponenata. Lokalne linije svakome obezbeđuju pristup čitavom sistemu i zato su važne. Nažalost, one su i najslabija tačka sistema. Kod međumesnih telefonskih kablova glavno pitanje je kako prikupiti sve pozive i poslati ih istovremeno istim optičkim kablom. To se zove multipleksiranje i mi ćemo proučiti njegove tri varijante. Na kraju, komutiranje veza se može izvesti na dva u osnovi različita načina, a mi ćemo opisati i jedan i drugi.

2.5.2 Politika telefonije

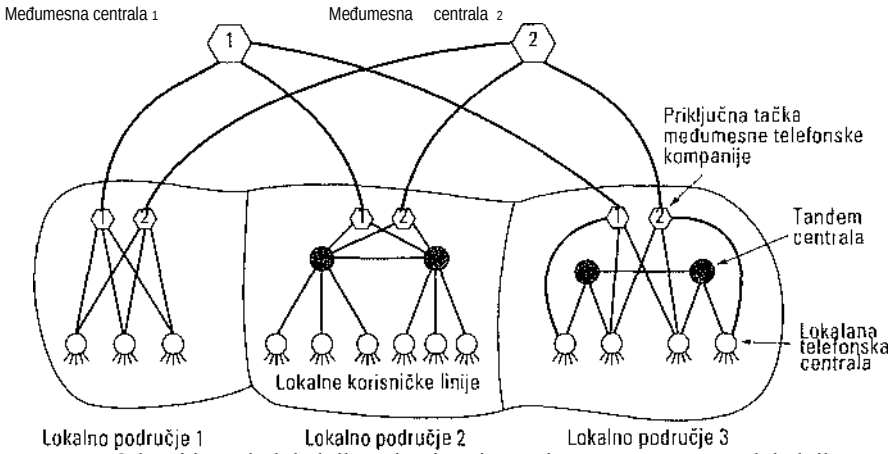
Decenijama pre 1984. godine, Belov sistem je na većem delu teritorije SAD obezbeđivao kako lokalne, tako i međugradske veze. Sedamdesetih godina, Savezna vlada je došla do zaključka da je to vid nelegalnog monopola i resila da ga ukine. U svome nastojanju uspela je da 1. januara 1984. razbije korporaciju AT&T na kompaniju AT&T Long Lines, **23 Belove telefonske centrale** (engl. *Bell Telephone Companies, BOCs)* i nekoliko drugih kompanija. Zbog ekonomičnosti, Belove telefonske centrale

su grupisane u sedam regionalnih centrala (engl. *RBOCs).* Ćela struktura telekomunikacija u Sjedinjenim Državama promenila se odlukom suda preko noći (Kongres *nije* doneo nikakvu odluku).

Detalji postupka oduzimanja prava opisani su u tzv. **Izmenjenom konačnom zaključku** (engl. *Modified Final Judgement, MFJ),* svojevrsnom oksimoronu - ako se zaključak mogao izmeniti, nikako nije mogao biti konačan. Taj događaj je doveo do povećane konkurencije, poboljšanja usluga i nižih cena međugradskog saobraćaja za pojedince i firme. Međutim, cene lokalnih usluga su rasle jer se u njih više nije pre- livao prihod iz međugradskih usluga, pa su sada morale same da se izdržavaju. I mnoge druge države su na sličan način uvele konkurenciju u ovu oblast.

Da bi bilo potpuno jasno šta ko radi, Sjedinjene Države su podeljene na 164 **lokalna područja pristupanja i transporta** (engl. *Local Access and Transport Areas, LATAs).* LATA je otprilike područje koje pokriva jedan pozivni broj. Unutar njega se nalazi **lokalna telefonska centrala** (engl. *Local Exchange Office, LEC),* koja ima monopol na klasične telefonske usluge unutar njega. Najvažnije lokalne centrale bile su Belove telefonske kompanije, iako u nekim područjima tu ulogu obavljaju od jedne, pa čak do preko 1500 nezavisnih telefonskih kompanija.

Sav saobraćaj između područja LATA održava druga vrsta kompanije, **međume- sna telefonska centrala** (engl. *IntereXchange Carrier, IXC).* Prvobitno je kompanija AT&T Long Lines bila jedini ozbiljan IXC, ali su sada u tom poslu WorldCom i Sprint takođe jaki konkurenti. Jedan od glavnih ciljeva pri raspodeljivanju poslova bio je da se sve međumesne centrale tretiraju na isti način u pogledu kvaliteta veze, tarife i broja cifara koje korisnik mora da bira da bi ih koristio. Način na koji je ovo urađeno prikazan je na slici 2-22. Tu vidimo tri primera lokalnih područja, svako s više lokalnih centrala. Područja 2 i 3 imaju i malu hijerarhiju s tandem centralama (interne regionalne centrale).



Slika 2-22. Odnosi između lokalnih područja pristupa i transporta **(LATA),** lokalnih **(LEC)** i međumesnih **(IXC)** telefonskih centrala. Svaki šestougao pripada međumesnoj telefonskoj kompaniji čiji se broj nalazi u njemu.

Svaka međumesna centrala koja želi da prenosi pozive potekle iz lokalnog područja, može

u njemu da izgradi posebnu centralu, zvanu **priključna tačka** (engl. *Point ofPre- sence, POP)*. Od lokalne telefonske kompanije se zahteva da poveže svaku međumesnu centralu sa svakom lokalnom centralom, bilo direktno, kao u područjima 1 i 3, ili indirektno, kao u području 2. Štaviše, uslovi povezivanja, i tehnički i finansijski, moraju biti jednaki za sve međumesne centrale. Na taj način, na primer, korisnik u području 1 može da bira između više međumesnih centrala da bi pozvao sagovornilca iz područja 3.

Deo Izmenjenog konačnog zaključka bio je i zabrana međumesnim centralama da nude lokalne usluge, kao i zabrana lokalnim kompanijama da nude međugradske telefonske usluge, iako su svi mogli da se bave i drugim delatnostima, npr. da drže restorane brze hrane. Tako je 1984. godine status učesnika prilično nedvosmisleno definisan. Nažalost, tehnologija ima nezgodnu osobinu da stalno grabi napredj zbog čega propisi brzo zastarevaju. Navedeni sporazum nije obuhvatao ni kablovsku televiziju ni mobilnu telefoniju. Kada se kablovska televizija iz jednosmerne pretvorila u dvosmernu, a mobilna telefonija zadobila izuzetnu popularnost, i lokalne i međumesne telefonske kompanije počele su da kupuju kablovske i mobilne operatere ili da se integrišu s njima.

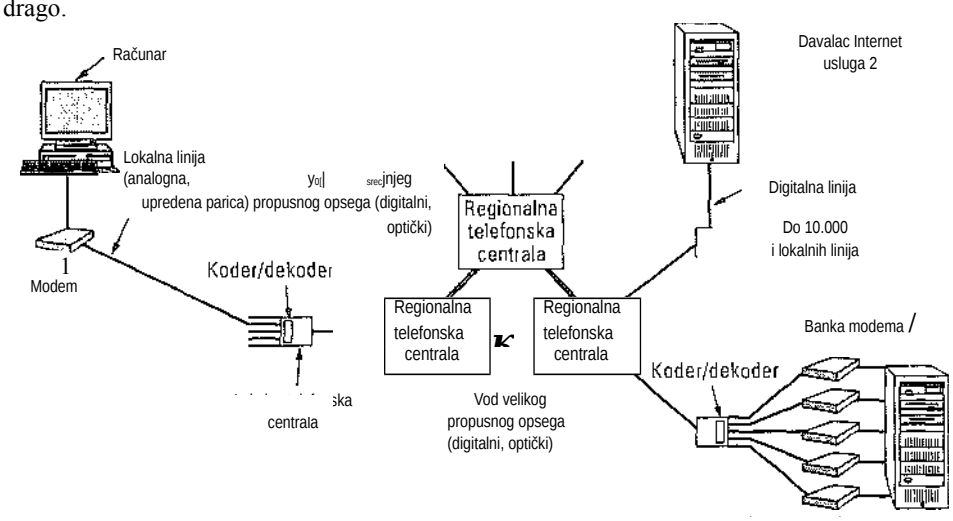
Kongres je 1995. uvideo da je neodrživo insistiranje na razlikama između dve vrste kompanija i doneo je propis kojim se dozvoljava kablovskoj televiziji, lokalnim telefonskim kompanijama, međumesnim telefonskim kompanijama i mobilnim operaterima da međusobno konkurišu jedni dragima u svim navedenim delatnostima. Zamisao je bila da se svakoj kompaniji omogući da korisnicima ponudi TV, telefonske usluge i usluge prenosa podataka u jedinstvenom paketu i da tako konkurišu jedna drugoj raznovrsnošću usluga, njihovim kvalitetom i cenom. Taj propis je februara 1996. postao zakon. Posle toga su neke Belove telefonske kompanije postale međugradske centrale, a neke drage kompanije, kao kablovska TV, počele su da nude i lokalne telefonske usluge.

Jedna zanimljiva odredba zakona iz 1996. obavezivala je lokalne telefonske centrale da omoguće prenosivost korisničkih telefonskih brojeva. To znači da je korisnik mogao da promeni lokalnu telefonsku kompaniju, a da zadrži svoj stari broj. Korisnici su sada mnogo bezbolnije mogli da biraju lokalnu telefonsku kompaniju koja im pruža najbolje usluge i time između njih zaoštrili konkurenciju. Kao rezultat primene ovog zakona, oblast telekomunikacija u SAD danas trpi velike promene. I opet, primer su sledile i mnoge drage države. One često čekaju da vide kako će neki eksperiment uspeti u SAD, pa ako se pokaže dobro, i same ga primenjuju; ako propadne, one probaju nešto drugo.

2.5.3 Lokalne veze: modemi, ADSL i bežične linije

Vreme je da počnemo detaljno da proučavamo rad telefonskog sistema. Njegovi glavni delovi prikazani su na slici 2-23. Tu vidimo lokalne linije, regionalne vodove, kao i regionalne i lokalne centrale, od kojih svaka ima opremu za komutiranje poziva. Lokalna telefonska centrala drži do 10.000 lokalnih linija (u SAD i dragim velikim zemljama). U stvari, donedavno je međugradski telefonski broj upućivao na lokalnu

centralu, tako da je broj (212) 601-xxxx označavao određenu lokalnu centralu s 10.000 pretplatnika, s brojevima od 0000 do 9999. S pojavom konkurencije u oblasti lokalnih usluga taj sistem se više nije mogao održati jer je previše kompanija zahte- valo da im se dodeli broj za lokalnu centralu, a oni su praktično svi bili iskorišćeni, tako da se moralo smisliti nešto drago.



Slika 2-23. Korišćenje analogno-digitalne veze za prenos podataka s računara na računar. Podaci se konvertuju posebnim uređajima: modemima i koderima/dekoderima.

Počnimo od onoga što poznaje većina korisnika: dvožične lokalne linije koja spaja lokalnu centralu s telefonima privatnih i poslovnih korisnika. Lokalna linija se često naziva i „poslednjim kilometrom“, iako njena stvarna dužina može da bude i više kilometara. Kroz nju su poslednjih 100 godina putovali analogni signali i tako će vero- vatno još neko vreme ostati jer cena digitalne opreme nije mala. Pa ipak, promene se događaju i u tom poslednjem uporištu analognog prenosa. U ovom odeljku ćemo proučiti klasičnu lokalnu liniju, kao i

novije izmene koje se u nju uvode, obraćajući posebnu pažnju na prenos podataka s kućnih računara.

Kada računar poželi da pošalje digitalne podatke preko analogne telefonske linije, podaci se prvo moraju pretvoriti u analogan oblik za prenos lokalnom linijom. To pretvaranje obavlja modem, uređaj koji ćemo uskoro proučiti. U lokalnoj telefonskoj centrali podaci se opet pretvaraju u digitalan oblik za slanje preko glavnih vodova.

Ako se na dragom kraju nalazi računar opremljen modemom, potrebno je ponovo izvršiti digitalno-analogno pretvaranje da bi se prošla lokalna linija na odredištu. Takva se veza na slici 2-23 uspostavlja s prvim davaocem Internet usluga koji ima banku modema, od kojih je svaki povezan s drugom lokalnom linijom. On može istovremeno da održava onoliko veza koliko ima modema (uz odgovarajuću snagu servera). Takav sistem je bio uobičajen sve dok se nisu pojavili modemi brzine 56 Icb/s, a razloge ćemo ubrzo objasniti.

Analogni signal predstavlja nepravilan naponski talas. Uz savršen prenosni me- dijuin, druga strana će primiti signal u obliku u kojem je poslat. Nažalost, medijumi nisu savršeni, tako da poslati i primljeni signal nisu isti. Kod digitalnih signala takva razlika se ne može tolerisati.

Prenosne linije pate od tri glavne boljke: slabljenja, kašnjenja i šuma. **Slabljenje** (engl. *attenuation)* označava smanjenje energije signala na putu zbog gubitaka. Gubici se izražavaju brojem decibela po kilometru i zavise od frekvencije. Da biste ra- zumeli uticaj frekvencije, nemojte zamišljati signal kao talas, već kao skup Furijeovih komponenata. Svaka komponenta slabi u drugačijoj meri, tako da na odredište stiže izobličen spektar Furijeovih komponenata.

Još gore **je** to što se pojedine Furijeove komponente različitom brzinom kredu kroz provodnik. Zbog toga na drugi kraj provodnika signal stiže **izobličen** (engl. *distorted)*

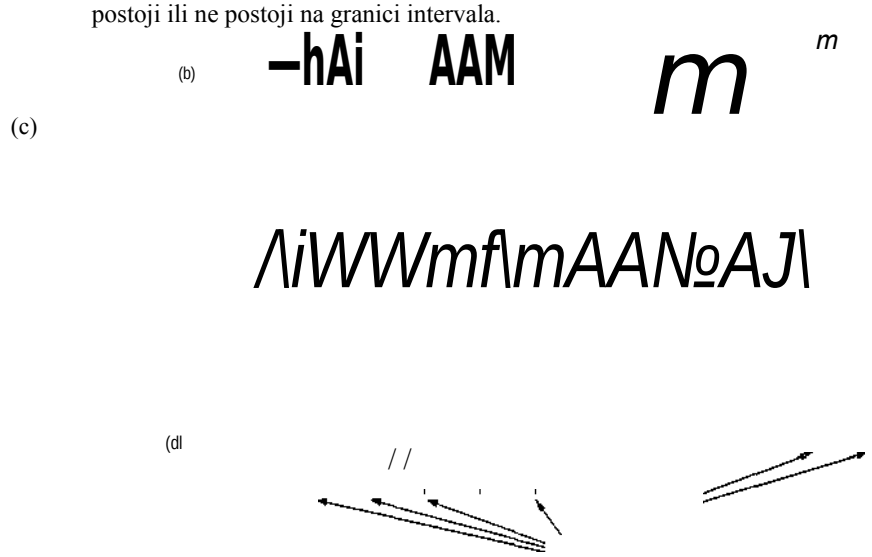
Tu je i problem **šuma** (engl. *noise),* energije koja ne potiče od izvora. Termički šum je prouzrokovan haotičnim kretanjem elektrona u provodniku i on se ne može iz- beći. Preslušavanje je, pak, izazvano induktivnom spregom između dva bliska provodnika. Ponekada, kada razgovarate telefonom, u pozadini možete da čujete dragi razgovor. To je preslušavanje. Na kraju, postoji i impulsni šum, izazvan naponskim udarima i drugim uzrocima. Takav impuls može da obriše jedan ili više bitova digitalnih podataka.

Modemi

Zbog problema koje smo upravo naveli, a naročito zbog činjenice da i slabljenje i brzina prostiranja signala zavise od frekvencije, nije poželjno da se signal prostire u širokom rasponu frekvencija. Nažalost, talasi pravougaonog oblika od kojih se sastoje digitalni signali zahvataju širok opseg frekvencija i zato su podložni jakom slabljenju i izobličenju zbog nejednakog kašnjenja. Zbog toga je slanje (jednosmernih) signala osnovnom frekvencijom nepogodno, osim malom brzinom i na kratka rastojanja.

Da bi se prevazišli problemi u vezi sa slanjem signala jednosmernom strujom, naročito na telefonskim vezama, signali su slati pomoću naizmenične struje. Uveden je stalan ton frekvencije od 1000 do 2000 Hz, zvan **sinusni noseći talas** (engl. *sine wave carrier),* čija su se amplituda, frekvencija ili faza mogli modulisati da bi se prenosili podaci. U **amplitudnoj modulaciji** (engl. *amplitude modulation),* korišćene su dve različite amplitude za predstavljanje nule, odnosno jedinice. U **frekventnoj modulaciji** (engl. *frequency modulation),* poznatoj i kao **modulacija s frekventnim pome- ranjem** (engl. *frequency shift keying),* koriste se dva (ili više) različitih tonova. (Engleski izraz „*keying*“ često se koristi kao sinonim za **modulisanje.) U** najjednostavnijem vidu **fazne modulacije** (engl. *phase modulation),* nosecem talasu se sistematski obrće faza za 180 stepeni u jednakim vremenskim intervalima. Bolje je kada se faza menja za 45, 135, 225 ili 315 stepeni jer se tada po vremenskom intervalu može preneti 2 bita podataka. Isto tako, kada se faza menja na kraju svakog vremenskog intervala, primalac lakše prepoznaje granice intervala.

Slika 2-24 prikazuje tri vrste modulacije. Na slici 2-24(a) jedna amplituda je različita od nule, dok je druga nula. Na slici 2-24(b) koriste se dve frekvencije; ona koja predstavlja binarnu nulu takođe je nula. Na slici 2-24(c) koriste se takođe dve frekvencije za predstavljanje binarne jedinice i nule i obe su različite od nule. Na slici 2-24(d) fazni pomak postoji ili ne postoji na granici intervala.



Slika 2-24. (a) Binarni signal, (b) Amplitudna modulacija, (c) Frekventna modulacija. (d) Fazna modulacija.

Uređaj koji prihvata niz bitova kao ulazne podatke i generiše noseći talas moduli- san pomoću jedne ili više opisanih metoda, ili modulisani noseći talas pretvara u niz bitova, naziva se modem (skraćeno od modulator-r/emodulator). Modem se smešta između (digitalnog) računara i (analognog) telefonskog sistema.

U cilju postizanja sve većih brzina, nije dovoljno samo povećavati brzinu uzorko- vanja. Nikvistova teorema tvrdi da ni uz savršenu liniju frekventnog opsega 3000 Hz (što modemska telefonska linija izvesno nije), nema smisla uzorkovati brže od 6000 puta u sekundi. U praksi, modemi uzorkuju brzinom 2400 puta u sekundi i trude se da uzorkom zahvate više bitova.

Jedinica za broj uzoraka u sekundi zove se **bod** (engl. *baud).* Tokom svakog boda šalje se jedan **simbol.** Tako linija od *n* bodova prenosi *n* simbola u sekundi. Na primer, linija od 2400 bodova šalje jedan simbol svakih 416,667 ps. Ako je simbol napon od 0 volti za logičku nulu i napon od 1 volta za logičku jedinicu, brzina prenosa je 2400 b/s. Ukoliko se, međutim, koriste naponi od 0,1,2 i 3 volta, svaki simbol sadrži 2 bita, tako da će linija od 2400 bodova prenositi 2400 simbola u sekundi, tj. 4800 b/s. Slično tome, kada postoje četiri različita fazna pomaka, i tu simbol sadrži 2 bita, pa je brzina prenosa podataka (b/s) dvostruko veća od brzine uzorkovanja u bodovima. Ova pos- lednja tehnika često se koristi pod nazivom **kvadraturna modulacija faznim pomakom** (engl. *Quadrature Phase Shift Keying, QPSK).*

Pojmovi propusni opseg, bod, simbol, i brzina prenosa često se brkaju, pa ćemo ih ponovo definisati. Propusni opseg medijuma je raspon frekvencija koje kroz njega prolaze uz minimalno slabljenje. To je fizičko svojstvo medijuma - navodi se obično kao raspon od 0 do neke maksimalne frekvencije (Hz). Bodovima se izražava broj napravljenih uzoraka u sekundi. Svaki uzorak nosi deo informacije, tj. jedan simbol. Broj bodova i broj simbola u sekundi su, dakle, jednaki. Tehnika modulisanja (npr. QPSK) određuje broj bitova po simbolu. Brzina prenosa predstavlja količinu podataka (bitova) prenesenih kanalom u sekundi i jednaka je broju simbola u sekundi pomnoženom brojem bitova po simbolu.

Svi noviji modemi koriste kombinaciju tehnika modulisanja da bi preneli više bitova po bodu. Cesto se kombinuje višestruka amplitudna i fazna modulacija da bi se prenelo nekoliko bitova po simbolu. Na slici 2-25(a) vidimo tačke pri 45, 135, 225 i 315 stepeni koje imaju jednaku amplitudu (udaljenost od koordinatnog početka). Faza tačke je određena uglom između prave koja spaja tačku s koordinatnim početkom i pozitivnog smera x-ose. Na slici 2-2.5(a) prikazane su četiri dozvoljene kombinacije koje se mogu koristiti za prenos 2 bita po simbolu. To je tehnika QPSK.

Na slici 2-25(b) vidimo drugačiji sistem modulisanja u kome se koriste četiri amplitude i četiri faze, ukupno 16 različitih kombinacija. Taj sistem modulisanja može se koristiti za prenos 4 bita po simbolu. On se zove **kvadraturna amplitudna modulacija** ili **QAM-16** (engl. *Quadrature Amplitude Modulation*). Ponekada se koristi i izraz **16-QAM.** QAM-16 se može koristiti, na primer, za prenošenje 9600 b/s kroz liniju koja uzorkuje brzinom 2400 boda.

90 90

90

9 9

180 ■

•o 180 ■

•o 180 ■

9 9

270 270 270

(a) (b) (c)

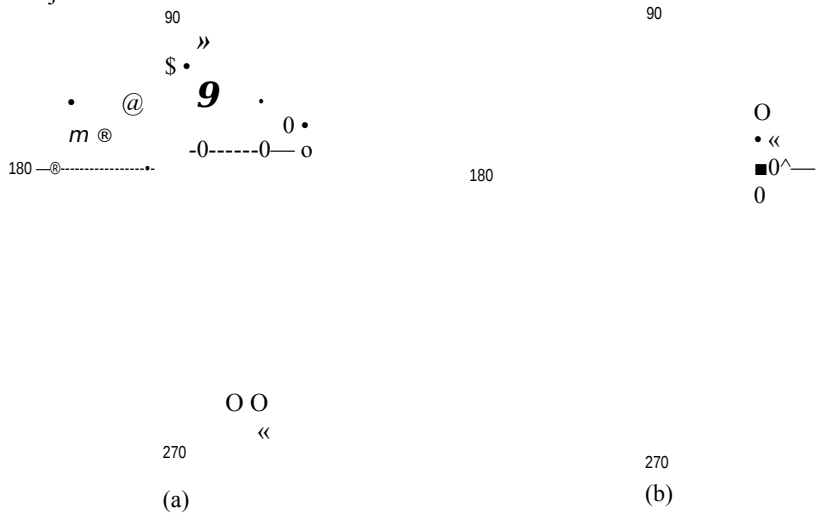
Slika 2-25. (a) QPSK. (b) QAM-16. (c) QAM-64.

Slika 2-25(c) prikazuje još jedan sistem gde se kombinuju višestruke amplitudne i fazne modulacije. On omogućava 64 različite kombinacije, tako da se po simbolu može preneti 6 bitova. On se zove **QAM-64,** a postoje i QAM sistemi višeg reda.

Dijagrami slični onom na slici 2-25, koji pokazuju dozvoljene kombinacije amplitude i faze, nazivaju se **konstelacioni dijagrami** (engl. *constellation diagram).* Svaki standard za brze modeme ima svoj konstelacioni dijagram i može da komunicira samo s modemom koji ima isti takav dijagram (iako većina modema može da emulira svoje sporije rođake).

Kada u konstelacionom dijagramu ima toliko tačaka, čak i mali šum u amplitudi ili fazi može da izazove grešku i da proizvede mnoge pogrešne bitove. Da bi se smanjila takva mogućnost, standardi za brze modeme predviđaju ispravljanje grešaka, zbog čega u svaki uzorak unose dodatni bit. Ta šerna je poznata kao **rešetkasto kodirana modulacija** (engl. *Trellis Coded Modulation, TCM).* Tako, na primer, modemski standard V.32 koristi 32 konstelacione tačke za prenošenje četiri bita podataka i jednog bita parnosti po simbolu, pri 2400 bodova, i postiže brzinu prenosa 9600 b/s. Njegov konstelacioni dijagram prikazan je na slici 2-26(a). Odluka da se dijagram rotira za 45 stepeni doneta je iz tehničkih razloga; rotirane i nerotirane konstelacije imaju isti informacioni kapacitet.

Sledeći standard **V.32 bis** opisuje modem brzine 14.400 b/s. Ta brzina prenosa postiže se prenošenjem šest bitova podataka i jednog bita parnosti po uzorku, pri 2400 bodova. Njegov konstelacioni dijagram ima 128 tačaka kad se koristi QAM-128 i prikazan je na slici 2-26(b). Faks modemi koriste ovu brzinu za prenos stranica slcenira- nih u bit mape. QAM-256 se ne koristi u standardnim telefonskim modemima, ali se koristi u kablovskim mrežama, kao što ćemo kasnije videti.



Slika 2-26. (a) V.32 za 9600 b/s. (b) V.32 bis za 14.400 b/s.

Posle standarda V.32 dolazi standard **V.34** za modeme brzine 28.800 b/s koji uz 2400 bodova prenose 12 bitova podataka po simbolu. Poslednji modem u ovoj seriji je **V.34 bis** koji prenosi 14 bitova podataka po simbolu pri 2400 bodova i postiže brzinu prenosa 33.600 b/s.

Da bi se efektivna brzina prenosa dalje povećala, u mnogim modemima se podaci komprimuju pre nego što se pošalju, tako da efektivna brzina prenosa premašuje po- menutih 33.600 b/s. S druge strane, skoro svi modemi proveravaju liniju pre nego što počnu da šalju korisničke podatke i, ako utvrde da je ona niskog kvaliteta, smanjuju brzinu prenosa. Zbog toga *efektivna* brzina prenosa koju zapaža korisnik može da bude manja, jednaka ili veća od nominalne brzine modema.

Svi savremeni modemi omogućavaju istovremeni saobraćaj u oba smera (koristeći za svaki smer drugu frekvenciju). Veza koja omogućava istovremeni saobraćaj u oba smera naziva se **potpuni dupleks** (engl. *full duplex)*. Autoput s dve razdvojene trake predstavlja potpuni dupleks. Veza koja omogućava saobraćaj u oba smera, ali u jednom trenutku samo u jednom, naziva se **poludupleks** (engl. *half duplex)*. Jedan že- lenički kolosek je poludupleks. Veza koja omogućava saobraćaj samo u jednom smeru naziva se **jednosmerna veza** (engl. *simplex)*. Takva je jednosmerna ulica. Drugi primer je optičko vlakno s laserom na jednom kraju i detektorom na dragom.

Standardi za modeme zaustavljaju se na brzini 33.600 b/s, na tzv. Šenonovoj granici (35 kb/s) - brzine veće od ove narušavaju zakone termodinamike. Ako vas zanima da li su teorijski mogući modemi brzine prenosa 56 kb/s, nastavite da čitate.

Ali zasto je teorijska granica baš 35 kb/s? To proizlazi iz prosečne dužine i kvaliteta lokalnih telefonskih veza. Granica od 35 kb/s određena je na osnovu prosečne dužine lokalnih linija. Na slici 2-23, poziv koji šalje računar na levoj strani i koji završava kod prvog davaoca Internet usluga, prolazi kroz dve lokalne linije kao analogni signal, jednom na izvoru, drugi put na odredištu. Svaka analogna linija pojačava šum signala. Kada bismo mogli da se oslobodimo jedne od ovih linija, maksimalna brzina prenosa bi se udvostručila.

Drugi davalac Internet usluga upravo tako radi. On ima čistu digitalnu liniju ka najbližoj telefonskoj centrali. On direktno dobija digitalni signal koji prolazi regionalnim vodovima, zbog čega kod njega nema kodera/dekodera, modema, niti analognog prenosa. Na taj način, kada je jedan kraj veze potpuno digitalan, kao što je slučaj sa skoro svim današnjim davaocima Internet usluga, maksimalna brzina prenosa može da bude i 70 lcb/s. Između dva kućna korisnika s modemima i analognim linijama, maksimum je 33,6 kb/s.

Nikvistova teorema je odgovorna za činjenicu da se koriste modemi brzine 56 kb/s. Telefonski kanal ima opseg 4000 Hz (uključujući i zaštitne margine). Maksimalan broj uzoraka u sekundi je na taj način 8000. Broj bitova po uzorku u Americi je 8; jedan se koristi za upravljanje, što ostavlja 56.000 b/s za korisničke podatke. U Evropi je svih 8 bitova raspoloživo korisnicima, tako da se mogu upotrebljavati modemi brzine 64.000 b/s. Međutim, da bi standard bio međunarodno priznat, dogovorena je brzina 56.000 b/s.

Opisani standard nosi oznaku V.90. On obezbeđuje prenos podataka od korisnika ka davaocu usluga brzinom 33,6 kb/s, a od davaoca ka korisniku brzinom 56 kb/s, jer se obično više podataka prenosi od davaoca ka korisniku nego obratno (na primer, zahtev za preuzimanje Web strane dugačak je samo nekoliko bajtova, dok sama strana može biti veličine i više megabajta). Teorijski bi se prema davaocu mogao uspostaviti kanal većeg propusnog opsega, ali lokalne linije najčešće imaju prejak šum i za brzinu od 33,6 lcb/s, pa se veći propusni opseg dodeljuje kanalu od davaoca ka korisniku, čime se povećavaju šanse da on stvarno radi brzinom 56 kb/s.

Sledeći brži standard je V.92. Modemi po tom standardu mogu ka davaocu Internet usluga da prenose podatke brzinom 48 kb/s ukoliko to može da prihvati linija. Njima je takođe potrebno upola manje vremena da odrede odgovarajuću brzinu prenosa nego starijim

modemima (oko 15 s). Najzad, oni omogućavaju da vas - dok ste na Internetu - neko pozove telefonom, pod uslovom da linija obezbeđuje uslugu čekanja na poziv.

Digitalne pretplatničke linije

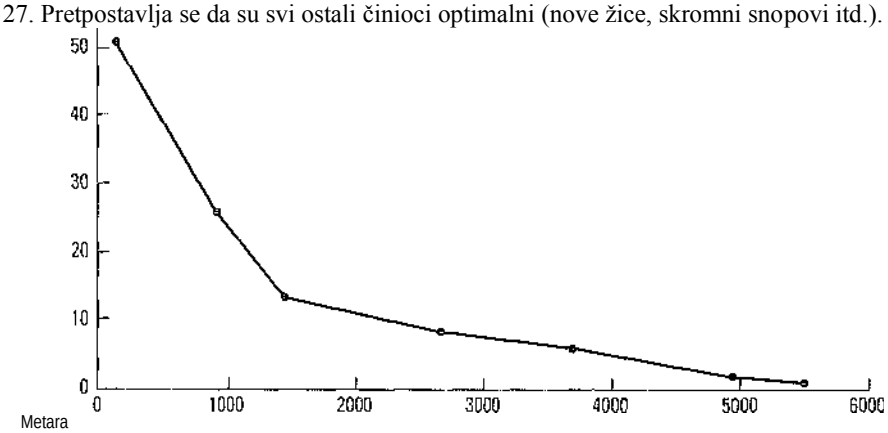
Kada je telefonija dostigla 56 kb/s, svi su sebi čestitali na dobro obavljenom poslu. U međuvremenu, kablovska televizija je već nudila 10 Mb/s, dok su satelitske kompanije planirale brzine i do 50 Mb/s. Kako je pristup Internetu postajao sve važnija stavka u njihovom poslovanju, lokalne telefonske kompanije su shvatile da moraju da razmišljaju o mnogo konkurentnijem proizvodu. I počele su da nude nove digitalne usluge preko lokalnih linija. Usluge s većim propusnim opsegom od standardnih telefonskih usluga ponekada se zovu **širokopojasne** (engl. *broadband),* iako izraz ima više komercijalni, nego tehnički značaj.

Na početku je bilo mnogo sličnih ponuda pod opštim imenom **digitalne pretplatničke linije** (engl. *Digital Subscriber Line, xDSL),* kod kojih se razlikovao samo prefiks *x.* U nastavku ćemo ih razmotriti, ali ćemo se uglavnom zadržati na verovatno najpopularnijoj usluzi - **asimetričnoj digitalnoj pretplatničkoj liniji** (engl. *Asymmetric Digital Subscriber Line, ADSL).* Pošto se ADSL linija još razvija i nije potpuno standardizovana, nešto od onoga što ćemo izneti možda će se vremenom promeniti, ali će osnovni opis i dalje važiti. Više informacija o ADSL linijama potražite kod Sum- mersa (1999) i Vettera i saradnika (2000).

Modemi su spori jer su telefoni smišljeni za prenos ljudskog glasa, a telefonski sistem optimizovan za istu svrhu, dok su podaci uvek bili siročići. U tačlci gde lokalna linija stiže u centralu, ona prolazi kroz filtar koji prigušuje sve frekvencije ispod 300 Hz i iznad 3400 Hz. Prigušivanje nije oštro - na granicama od 300 i 3400 Hz slabljenje je 3 dB, tako da se obično navodi propusni opseg od 4000 Hz, iako je područje između pomenutih granica široko 3100 Hz. Prenos podataka je tako ograničen na ovu usku oblast.

Trik koji omogućava da linija xDSL radi sastoji se u tome što se ulazna linija spaja sa skretnicom koja nema filtar i tako stavlja na raspolaganje čitav propusni opseg lokalne linije. On sada nije veštački postignutih 3100 Hz, već je određen samo fizičkim stanjem linije.

Nažalost, kapacitet lokalne linije zavisi od više činilaca: dužine, debljine i opšteg kvaliteta. Zavisnost potencijalnog propusnog opsega od dužine linije prikazana je na slici 2-27. Pretpostavlja se da su svi ostali činioci optimalni (nove žice, skromni snopovi itd.).



Slika 2-27. Propusni opseg za DSL u zavisnosti od dužine neoklopljene upredene parice 3. kategorije.

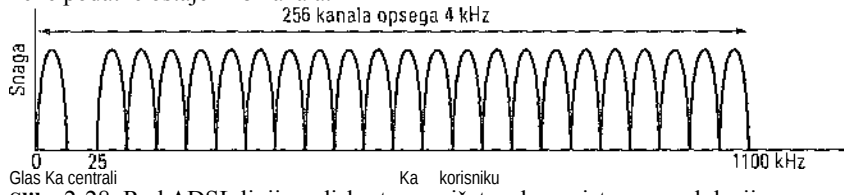
Stanje koje ovaj dijagram implicira, stvara probleme telefonskim kompanijama. Kada

kompanija bira brzinu koju želi da ponudi, ona istovremeno bira i područje oko svojih lokalnih centrala izvan koga ne može da ponudi uslugu. Kada udaljeni korisnik poželi da se prijavi za uslugu, to znači da će ponekad dobiti odgovor: „Zahvaljujemo što ste nam se obratili, ali vi stanujete 100 m predaleko od najbliže lokalne centrale. Da li možete da se preselite negde bliže?“ Što kompanija izabere sporiji prenos, područje oko lokalnih centrala biće šire i više će korisnika moći da usluži. Ali, što je brzina manja, i usluga je manje atraktivna, pa će manje korisnika biti voljno da je plati. Ovde se sukobljavaju tehnologija i biznis. (Mogu se graditi i mini lokalne centrale u susedstvu, ali je to rešenje skupo.)

Sve xDSL linije pravljene su sa određenim ciljevima. Prvo, usluga mora da radi pomoću postojeće lokalne upredene parice 3. kategorije. Drugo, ona ne srne da ometa ko- lisnikove postojeće telefone i faksove. Treće, brzina prenosa mora biti mnogo veća od 56 kb/s. Četvrto, xDSL linija mora stalno biti aktivna uz paušalnu mesečnu naknadu.

Prvu ADSL uslugu ponudila je korporacija AT&T, deleći raspoloživi frekventni opseg lokalne linije od oko 1,1 MHz u tri frekventna područja: za staru dobru telefonsku uslugu (engl. *Plain Old Telephone Seiyice, POTS),* za prenos od korisnika ka centrali i za prenos od centrale ka korisniku. Tehnika korišćenja više frekventnih područja zove se multipleksiranje podelom frekvencije; proučićemo je u jednom od narednih odeljaka. Ponude dragih davalaca ADSL usluga koje su ubrzo usledile imale su drugačiji prilaz i pošto izgleda da će taj pristup prevagnuti, detaljnije ćemo ga opisati.

Alternativni pristup, zvan diskretan višetonski sistem (engl. *Discrete MultiTone, DMT),* prikazanje na slici 2-28. U njemu se, u stvari, raspoloživi opseg lokalne linije od 1,1 MHz deli na *256* nezavisnih kanala, svaki opsega 4312,5 Hz. Kanal 0 se koristi za POTS. Kanali 1-5 se ne koriste da bi se govorni signal što bolje odvojio od signala podataka. Od preostalih 250 kanala, dva se koriste za upravljanje (po jedan za svaki smer prenosa), tako da za korisničke podatke ostaje 248 kanala.



Slika 2-28. Rad ADSL linije u diskretnom višetonskom sistemu modulacije.

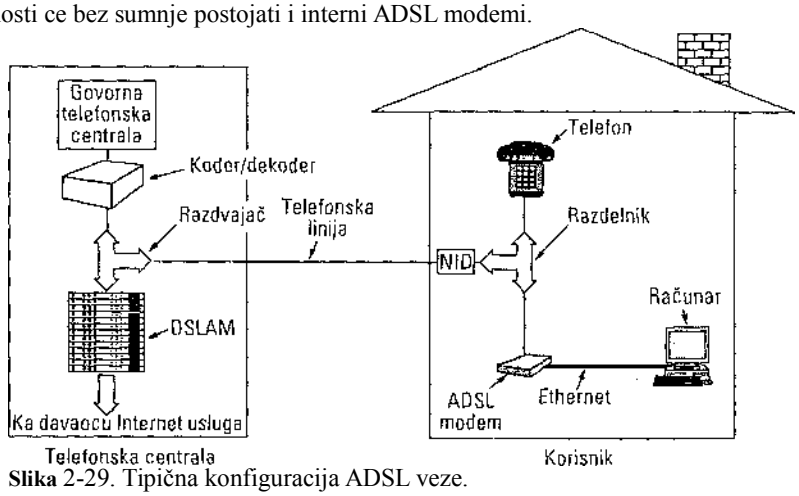
Svaki od preostalih kanala u načelu se može koristiti za potpun dupleksni prenos podataka, ali ce viši harmonici, preslušavanje i drugi efekti ograničiti brzinu prenosa na vrednost znatno nižu od teorijske. Davalac usluge određuje koliko ce kanala koristiti za prenos ka centrali, a koliko za prenos ka korisniku. Odnos 50:50 je tehnički izvodljiv, ali davaoci ADSL usluga većinom rezervisti 80-90% kanala za prenos ka korisniku jer se u praksi pokazalo da taj smer zahteva vec'i kapacitet prenosa. Zbog takvog izbora, usluga ADSL počinje slovom „A“ (asimetrična). Obično se 32 kanala rezerviše za prenos ka centrali, a ostatak za prenos ka korisniku. Izvodljivo je i to da se nekoliko gornjih kanala rezervisanih za prenos ka centrali učine dvosmemim radi širenja opsega, ali ta optimizacija zahteva ugradnju specijalnog kola koje poništava odjeke.

Standard za ADSL prenos (ANSI T1.413 i ITU G.992.1) dozvoljava brzine čak do 8 Mb/s ka korisniku i 1 Mb/s ka centrali. Međutim, malo davalaca nudi ovakve brzine prenosa. Obično se nude brzine od 512 kb/s ka korisniku i 64 lcb/s ka centrali (standardna usluga) i 1 Mb/s ka korisniku, odnosno 256 kb/s ka centrali (prvoklasna usluga).

Unutar svakog kanala koristi se sistem modulacije sličan standardu V.34, iako brzina

uzorkovanja nije 2400, već 4000 bodova. U svakom kanalu neprestano se nadzire kvalitet linije i po potrebi podešava brzina prenosa podataka, tako da se brzina prenosa može razlikovati od jednog kanala do dragog. Podaci se moduliraju tehnikom QAM sa 15 bitova po bodu, prema konstelacionom dijagramu analognom onom sa slike 2~25(b). Uz, recimo, 224 kanala za prenos ka korisniku i 15 bitova po bodu pri 4000 bodova, propusni opseg za prenos ka korisniku iznosi 13,44 Mb/s. U praksi, odnos signala i šuma ne dozvoljava tako veliku brzinu prenosa, ali se u kraćim vremenskim intervalima i kroz visokokvalitetne linije postiže 8 Mb/s, što objašnjava visoku vrednost predviđenu standardom.

Na slici 2-29 prikazan je tipičan sistem ADSL. Telefonska kompanija mora da instalira mrežni **interfejs** (engl. *Network Interface Device, N1D)* kod korisnika. Ta mala plastična kutija označava granicu između vlasništva kompanije i vlasništva korisnika. Odmah pored NID-a (ponekad i u kombinaciji s njim) nalazi se **razdelnik** (engl. *splitter),* analogni filtar- koji razdvaja područje CM-000 Hz - koje koristi POTS - od podataka. POTS signal se usmerava ka postojećem telefonu ili faksu, a signal podataka ka ADSL modemu. ADSL modem je u stvari uređaj za obradu digitalnog signala, podešen tako da radi kao paralelan skup od 250 QAM modema različitih frekvencija. Pošto su savremeni ADSL modemi uglavnom spoljni, računar s modemom mora biti povezan brzom vezom. Obično se računar opremi mrežnom karticom i uspostavi vrlo kratka Ethernet mreža s dva čvora: računarom i ADSL modemom. Ponekada se ume- sto Ethernet kartice koristi USB priključak. U budućnosti ce bez sumnje postojati i interni ADSL modemi.



Slika 2-29. Tipična konfiguracija ADSL veze.

Na drugom kraju žice, u telefonskoj centrali, takođe se instalira razdelnik. Ovde se izdvaja glasovni deo signala i šalje klasičnoj govornoj telefonskoj centrali. Signal iznad 26 kHz usmerava se ka **multipleksoru pristupa digitalnoj pretplatničkoj liniji** (engl. *Digital Subscriber Line Access Multiplexer, DSLAM),* koji sadrži istu vrstu uređaja za obradu digitalnog signala kao ADSL modem. Pošto se signal prevede u tok bitova, obrazuju se paketi i šalju davaocu Internet usluga.

Ovo potpuno razdvajanje sistema prenosa glasa od ADSL-a olakšava telefonskoj kompaniji instaliranje ADSL linija novim korisnicima. Potrebno je samo dokupiti DSLAM i razdelnik, i spojiti ADSL pretplatnika na razdelnik. Za druge širokopojasne usluge (npr. za ISDN) potrebne su radikalnije izmene telekomunikacione opreme.

Nedostatak šeme sa slike 2-29 jeste potreba za mrežnim interfejsom i razdelnikom kod korisnika. Njih može da instalira samo telefonska kompanija, što prouzrokuje troškove

dolaska u kuću. Zbog toga je standardizovan i alternativan sistem u kome nema razdelnika. Nezvanično ga zovu G.lite, ali ima i svoje standardno ITU ime: G.992,2. On izgleda isto kao na slici 2-29, osim što nema razdelnik - koristi se postojeća telefonska linija. Razlika je u tome što u telefonski priključak, između telefona i linije ili između ADSL modema i linije, treba umetnuti mikrofiltar. Mikrofiltar za telefon je niskopropusni filtar koji prigušuje frekvencije iznad 3400 Hz; mikrofiltar za ADSL modem je visokopropusni filtar koji prigušuje frekvencije ispod 26 kHz. Međutim, ovaj sistem nije toliko pouzdan kao sistem s razdelnikom, tako da se G.lite može koristiti do brzina 1,5 Mb/s (u odnosu na 8 Mb/s za ADSL liniju s razdelnikom). Za sistem G.lite ipak je potreban razdelnik u centrali, ali otpadaju troškovi za odlazak kod korisnika.

ADSL je samo standard za fizički sloj. Šta se odvija iznad njega zavisi od centrale. Često izbor pada na ATM jer ATM može da podešava kvalitet usluge i zato što mnoge telefonske kompanije interno koriste taj režim.

Bežične lokalne linije

Počev od 1996. godine, u SAD su (a nešto kasnije i u drugim državama) kompanije koje su želele dakonkurišu „ušančenoj" lokalnoj telefonskoj kompaniji (bivšem monopolisti), zvanoj **obavezni** LEG (engl. *Incumbent LEC, ILEC),* dobile u tom pogledu odrešene ruke. Najpodobniji kandidati za takvu delatnost bile su međugradske telefonske kompanije (IXC). Svaka takva kompanija koja želi da uđe u posao pružanja lokalnih telefonskih usluga u nekom gradu, mora da uradi sledeće. Prvo, treba da kupi ili da iznajmi zgradu za svoju prvu lokalnu centralu u tom gradu. Drugo, centralu treba da ispuni komutatorima i drugom opremom koja se odmah može kupiti od različitih proizvođača. Treće, treba da razvuče optički kabl između lokalne i najbliže regionalne centrale da bi novi lokalni korisnici imali vezu s državnom mrežom. Četvrto, treba da privuče nove korisnike, najčešće boljom uslugom i nižom cenom od postojećeg ILEC-a.

Tada počinje onaj teži deo. Pretpostavimo da se stvarno javljaju novi potencijalni korisnici. Kako će nova telefonska kompanija, zvana **konkurentski** LEC (engl. *Competitive LEC, CLEQ* povezati telefone i računare korisnika sa svojom blistavom novom centralom? Kupovanje neophodnih prava prolaska i razvlačenje žica ili optičkih kablova ne dolazi u obzir jer je preskupo. Mnoge konkurentske telefonske kompanije otkrile su jeftinu alternativu klasičnoj upredenoj parici: **bežičnu lokalnu liniju** (engl. *Wireless Local Loop, WLL).*

Fiksni telefon koji koristi bežičnu lokalnu liniju u izvesnoj meri podseća na mobilni telefon, ali između njih postoje i tri ključne tehničke razlike. Prvo, korisnik bežične lokalne linije često želi brzu vezu sa Internetom, obično brzinom koja odgovara ADSL liniji. Drugo, novi korisnik verovatno neće zameriti što će mu tehničar na krovu instalirati veliku antenu usmerenu ka CLEC centrali. Treće, korisnik ne menja mesto, što otklanja sve probleme sa kretanjem i preuzimanjem upravljanja mobilnim telefonima, o čemu ćemo govoriti u drugom delu poglavlja. Tako je rođena nova oblast telekomunikacija: **fiksni bežični** (engl .*fixed wireless)* prenos, u kome za lokalne telefonske usluge i usluge Interneta konkurentska telefonska kompanija koristi bežičnu lokalnu liniju.

Iako su bežične lokalne linije ozbiljno zaživele tek 1998. godine, vratićemo se u 1969. godinu da bismo proučili njihov nastanak. Te godine je Savezna komisija za komunikacije (FCC) dodelila dva kanala (opsega po 6 MHz) za obrazovnu televiziju na frekvenciji od 2.1 GHz. Narednih godina je dodeljen još 31 kanal na frekvenciji od 2,5 GHz, tako daje ukupan opseg bio 198 MHz.

Obrazovna televizija nikada nije počela s radom i 1998. godine FCC je povukla sve

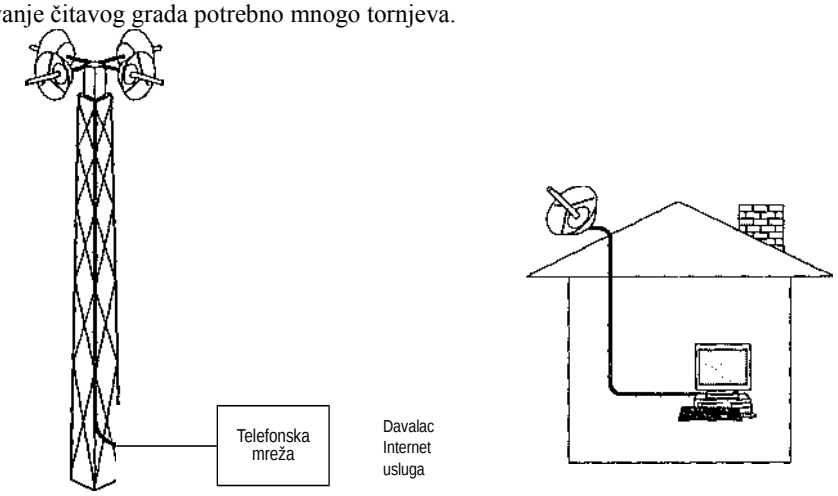
dodeljene frekvencije i namenila ih dvosmernom radio-saobraćaju. Mikrotalasi ovih frekvencija imaju dužinu 10-12 cm. Domet im je oko 50 km i relativno dobro

prolaze kroz vegetaciju i kišu. Opseg od 198 MHz novog spektra odmah je iskorišćen za bežične lokalne linije u vidu usluge zvane višekanalna distributivna usluga za više korisnika (engl. *Multichannel Multipoint Distribution Service, MMDS)*. MMDS se može smatrati gradskom mrežom, kao i njoj bliska usluga LMDS o kojoj govorimo u nastavku.

Velika prednost ove usluge je to što za nju postoji proverena tehnologija, a potrebna oprema se lako može nabaviti. Mana joj je skroman ukupni propusni opseg koji dele mnogi korisnici s prilično prostranog geografskog područja.

Zbog malog propusnog opsega usluge MMDS, zanimanje su privukli milimetarski talasi. Frekventna područja između 28 i 31 GHz u SAD i pri 40 GHz u Evropi ostala su nedodeljena jer je bilo teško napraviti silicijumska integrisana kola koja rade tako brzo. Problem su resila integrisana kola s galijum-arsenidom koja su otvorila milimetarska područja za radio-komunikacije. FFC je odgovorila na zahteve dodeljujući opseg od 1,3 GHz novoj usluzi bežične lokalne linije, zvanoj lokalna distributivna usluga za više korisnika (engl. *Local Multipoint Distribution Service, LMDS)*. Taj opseg je najveći koji je FFC ikada dodelila. Sličan opseg je dodeljen i u Evropi, ali pri frekvenciji od 40 GHz.

Slika 2-30 prikazuje rad usluge LMDS. Na njoj je prikazan toranj s više antena usmerenih u različitim pravcima. Pošto su milimetarski talasi strogo usmereni, svaka antena definiše sektor nezavistan od drugih sektora. Pri ovoj frekvenciji, domet je 2-5 km, što znači da je za pokrivanje čitavog grada potrebno mnogo tornjeva.



Slika 2-30. Arhitektura sistema LMDS.

U LMDS vezi, slično kao u ADSL vezi, propusni opseg se dodeljuje asimetrično, uz davanje prednosti prenosu podataka ka korisniku. Uz sadašnju tehnologiju, moguće je u sektoru ostvariti ukupnu brzinu prenosa 36 Gb/s ka korisnicima i 1 Mb/s ka centrali. Ako svaki aktivni korisnik preuzima tri strane po .5 KB u minutu, time u prošeku

zauzima 2000 b/s propusnog opsega, što omogućava da 18.000 korisnika po sektoru rade istovremeno. Da bi se kašnjenje svelo na razumnu meru, ne treba dozvoliti da istovremeno radi više od 9000 korisnika. Ako postoje četiri sektora, kao na slici 2-30, biće omogućen istovremeni rad 36.000 korisnika. Pretpostavljajući da će samo jedan od tri pretplatnika koristiti vezu u saobraćajnom špicu, jedan telekomunikacioni toranj sa četiri antene moći će da opsluži 100.000 korisnika unutar područja na 5 km od tornja. Proračune slične navedenom izradile su mnoge potencijalne konkurentske kompanije i neke od njih su zaključile da uz umereno investiranje u tehnologiju milimetarskih talasa mogu da uđu u posao pružanja lokalnih telefonskih usluga i pristupa Internetu i da korisnicima ponude brzinu prenosa podataka uporedivu sa onom koju nudi kablovska TV, a po nižoj ceni.

Sistem LMDS, međutim, ima i svoje mane. Kao prvo, milimetarski talasi se prostiru pravolinijski, pa između tornja i antena na krovovima korisnika mora postojati optička vidljivost. Drugo, vegetacija dobro apsorbuje ove talase, pa toranj mora biti dovoljno visok da ona ne bi ometala vezu. (Ono stoje optički „čista“ putanja u decembru, možda ne izgleda tako leti kada sve drveće ozeleni.) Milimetarske talase apsorbuje i kiša. Greške prouzrokovane kišom mogu se u izvesnoj meri ublažiti pri- menom koda za ispravljanje grešaka ili pojačanjem snage emitovanja tokom kišnih dana. Pa ipak, za uslugu LMDS prvenstveno je pogodna suva klima, tako da ćemo je videti pre u Sahari nego u Indiji.

Bežične lokalne linije neće privući veću pažnju sve dok se ne uspostave standardi na osnovu kojih proizvođači mogu da prave opremu koja će korisnicima omogućiti da menjaju davaoca usluga (lokalnu telefonsku kompaniju), a da pri tom ne moraju da kupuju novu opremu. U tom smislu, IEEE je formirao komitet pod imenom 802.16 da napravi standard za LMDS. Standard 802.16 objavljen je aprila 2002. IEEE ovu mrežu zove **bežična gradska mreža** (engl. *wireless MAN).*

Sistem IEEE 802.16 namenjen je za digitalnu telefoniju, za pristup Internetu, za povezivanje dve udaljene lokalne mreže, za emitovanje TV i radio programa itd. Posvetićemo mu više vremena u 4. poglavlju.

2.5.4 Vodovi i multipleksiranje

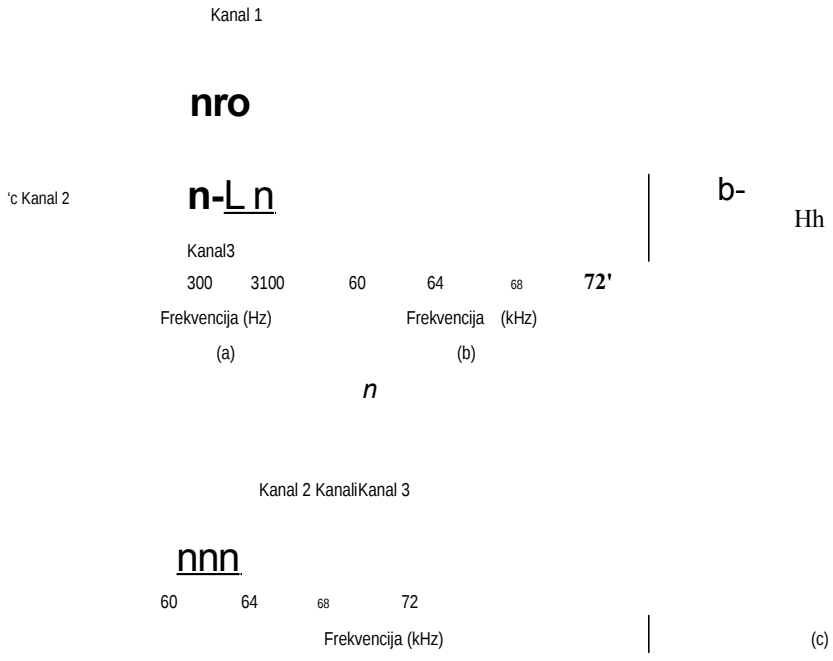
Veličina sistema igra važnu ulogu u troškovima telefonskih kompanija. Cena instaliranja i održavanja ista je i za širokopojasni i za uskopojasni vod između dve centrale (tj. glavnina troškova otpada na prokopavanje kanala, a samo mali deo na bakarni ili optički kabl). Zbog toga su telefonske kompanije razvile složene sisteme multipleksiranja, tj. sisteme za istovremeno prenošenje više razgovora istim vodom. Sistemi multipleksiranja mogu se podeliti u dve glavne kategorije: **multipleksiranje podelom frekvencije** (engl. *Frequency Division Multiplexing, FDM)* i **multipleksiranje podelom vremena** (engl. *Time Division Multiplexing, TDM).* U sistemu FDM, spektar frekvencija se deli na područja i svako područje se dodeljuje na isključivo ko- rišćenje jednom pretplatniku. U sistemu TDM, korisnici se smenjuju u kratkim vremenskim intervalima u kojima imaju na raspolaganju ceo propusni opseg.

Amplitudna radio-difuzija ilustruje obe vrste multipleksiranja. Dodeljeno područje frekvencija je oko 1 MHz, otprilike između 500 i 1500 kHz. Pojedinim logičkim kanalima (radio-stanicama) dodeljuju se različiti delovi frekventnog spektra, dovoljno međusobno udaljeni da se izbegne interferencija. Taj sistem je primer multipleksiranja podelom frekvencija. Pored njega (u nekim zemljama), pojedinačne stanice imaju dva logička potkanala: za muziku i za oglase. Oni se smenjuju u kratkim vremenskim intervalima na istoj frekvenciji, muzika, oglasi, muzika itd. To je primer multipleksiranja podelom vremena.

U nastavku ćemo opisati multipleksiranje podelom frekvencija, a zatim razmotriti kako se FDM može primeniti na optička vlakna (multipleksiranje podelom talasne dužine). Potom ćemo se okrenuti sistemu TDM koji se koristi za optička vlakna (SONET).

Multipleksiranje podelom frekvencija

Slika 2-31 prikazuje kako se sistemom FDM multipleksiraju tri govorna telefonska kanala. Filtar ograničava propusni opseg svakog kanala na oko 3100 Hz. Kada se multipleksira više kanala, svakom kanalu se dodeljuje po 4000 Hz da bi bili dobro razdvojeni. Kada se kanali raspodele po frekvencijama, mogu se kombinovati jer svaki zauzima odvojeno područje spektra. Obratite pažnju na to da se uprkos postojanju razmaka (zaštitnih margina) između kanala, susedni kanali malo preklapaju pošto filtri ne odsecaju oštro frekventna područja. To znači da će jak. signal na ivici frekventnog područja jednog kanala biti registrovan u susednom kanalu kao netermički šum.

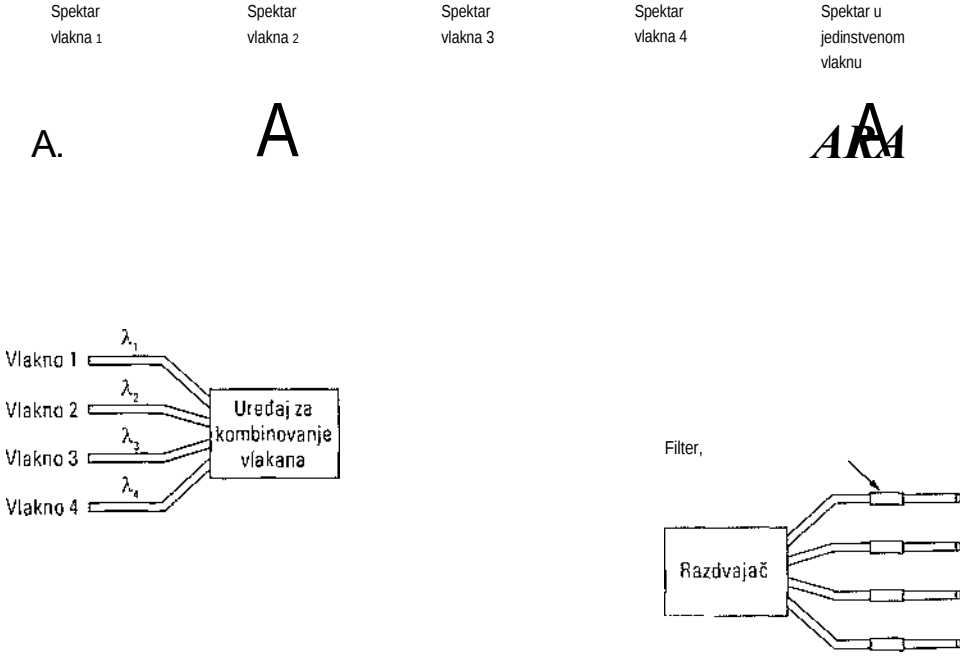


Slika 2-31. Multipleksiranje podelom frekvencija, (a) Prvobitni kanali. (b) Kanali premešteni u dodeljena frekventna područja, (c) Multipleksirani kanal.

Sistem FDM se širom sveta realizuje na prilično standardan način. Najčešće se dvanaest govornih kanala opsega 4000 Hz raspodeljuju unutar frekventnog područja od 60 do 108 kHz. Takav skup se naziva **grupa** (govornih kanala). Područje između 12 i 60 kHz ponekad se koristi za drugu grupu. Mnoge kompanije za takve grupe govornih kanala nude iznajmljene linije brzine od 48 do 56 kb/s. Pet grupa (60 govornih kanala) mogu se multipleksirati u **supergrupu.** Sledeći stepen u hijerarhiji je **matična grupa** (engl. *mastergroup)* koja okuplja pet (standard CCITT) ili deset supergrupa (Belov sistem). Postoje i standardi za okupljanje do 230.000 govornih kanala.

Multipleksiranje podelom talasne dužine

Za kanale izvedene optičkim kablom primenjuje se FDM varijanta: **multipleksiranje podelom talasne dužine** (engl. *Wavelength Division Multiplexing, WDM).* Osnovni princip WDM-a u optičkom vlaknu prikazan je na slici 2-32. Četiri optička vlakna, svako sa signalom drugačije talasne ružine, kombinuju se u jedinstveno vlakno koje signale prenosi na udaljeno odredište. Na odredištu se jedinstveno vlakno razgranava u onoliko pojedinačnih vlakana koliko ih je bilo i na izvoru. Svako izlazno vlakno ima kratko, specijalno projektovano jezgro koje propušta samo jednu talasnu dužinu. Rezultujući signali u pojedinačnim vlaknima mogu se odvesti do svojih odredišta ili ponovo kombinovati na drugačiji način za dalji multiplelcsirani prenos.



Jedinstveno vlakno za daljinski prenos **Slika 2-32.** Multipleksiranje podelom talasne dužine.

Ovde nema ničeg suštinski novog. U pitanju je poznato multipleksiranje deljenjem vrlo visokih frekvencija. Sve dok svaki kanal ima svoj frekventni opseg (opseg talas- nih dužina) jasno razdvojen od drugih sličnih opsega, kanali se mogu multipleksirati u zajedničkom vlaknu za daljinski prenos. Jedina razlika u odnosu na električnu varijantu sistema FDM ogleda se u tome što se u optičkom sistemu kao filtri koriste di- frakcione rešetke - pasivni elementi visoke pouzdanosti.

Tehnologija WDM napreduje tempom koji računarsku tehnologiju potiskuje u drugi plan. Ona se pojavila oko 1990. godine. Prvi komercijalni sistemi imali su osam kanala, svaki s propusnim opsegom 2,5 Gb/s. Već 1998. godine na tržištu su se pojavili sistemi sa 40 kanala po 2,5 Gb/s. Godine 2001. postojali su sistemi sa 96 kanala po 10 Gb/s, ukupnog propusnog opsega 960 Gb/s. To je dovoljno velika propusna moć za istovremeno prenošenje 30 standardnih igranih filmova u sekundi (uz tehnologiju MPEG-2). U laboratoriji sada rade i sistemi sa 200 kanala. Kada ima vrlo mnogo kanala, a njihove talasne dužine su tesno zbijene jedna uz dragu - na primer, na odstojanju 0,1 nm - sistem se često naziva gusti WDM (engl. *Dense WDM, DWDM).*

Treba primetiti da je razlog popularnosti sistema WDM to što je širina signala u optičkom vlaknu obično samo nekoliko GHz jer današnji opto-električni i elektro-op- tički pretvarači ne mogu da rade brže. Kada se prenos različitim talasnim dužinama obavlja paralelno kroz više kanala, ukupni propusni opseg raste linearno s brojem kanala. Pošto je propusni opseg jedinstvenog optičkog vlakna oko 25.000 GHz (pogledajte sliku 2-6), postoji teorijska mogućnost za obrazovanje 2500 kanala po 10 Gb/s, čak. i uz 1 b/Hz (a moguće su i veće brzine).

Sledeća novina su optički pojačivači. Ranije je na svakih 100 km multipleksirani signal morao biti razbijen na pojedinačne kanale, signal svakog kanala morao je biti pretvoren u odgovarajući električni signal, ovaj pojačan, a zatim opet pretvoren u optički signal i multipleksiran sa sličnim optičkim signalima za dalji prenos. Danas ukupan optički signal treba pojačati tek na svakih 1000 km, pojačivačem koji sadrži samo optičke komponente, i bez potrebe da se signal pretvara iz optičkog u električni i obratno.

Sistem sa slike 2-32 radi s fiksnim talasnim dužinama. Podaci iz ulaznog vlakna 1 upuću ju se izlaznom vlaknu 3, podaci iz ulaznog vlakna 2 idu u izlazno vlakno 1 itd. Međutim, moguće je napraviti i komutirani WDM sistem. U njemu se izlazna vlakna mogu birati pomoću Fabri-Peroovog ili Mah-Zenderovog interferometra. Više detalja

0 sistemu WDM i njegovim primenama za komutiranje paketa na Internetu potražite kod Elmirghanija i Mouftaha (2000), Huntera i Andonovica (2000), kao i kod Lista- nija i saradnika (2001).

Multipleksiranje podelom vremena

WDM tehnologija je odlična, ali u telefonskom sistemu ima još mnogo bakarne žice, pa ćemo se sada malo vratiti na nju. Iako tehnologija FDM koristi bakarne žice

1 mikrotalasne kanale, za nju su neophodne analogne veze zbog čega se ne slaže dobro s računarima. Nasuprot tome, tehnologija TDM može se u potpnosti realizovati po- moću digitalne elektronike, zbog čega je poslednjih godina daleko šire prihvaćena. Nažalost, ona se može koristiti samo za digitalne podatke. Pošto lokalne telefonske linije prenose analogne signale, u centrali su neophodni analogno-digitalni pretvarači da bi svi lokalni signali mogli biti mutipleksirani i prosleđeni telefonskim vodom.

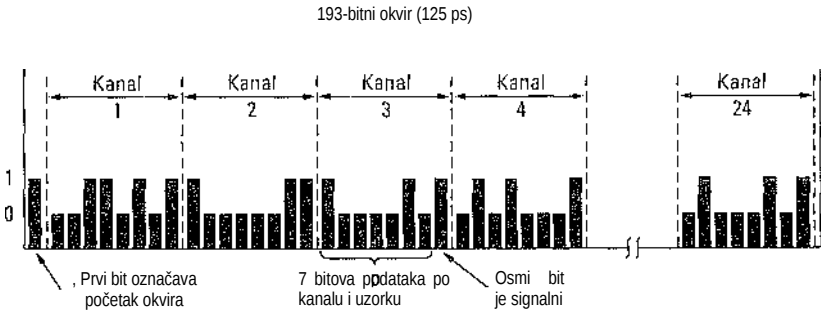
Sada ćemo razmotriti kako se više analognih govornih signala digitalizuje i kom- binuje u jedinstven digitalni signal za slanje telefonskim vodom. Računarski podaci poslati pomoću modema takođe su analogni, tako da sledeći opis važi i za njih.

Analogni signal se u lokalnoj telefonskoj centrali digitalizuje pomoću **kodera/deko-** dera (engl. *codec)* koji ga pretvara u niz 8-bitnih brojeva. Uređaj uzorkuje analogni signal 8000 puta u sekundi (125 ps po uzorku) jer Nikvistova teorema tvrdi daje ta brzina dovoljna za prikupljanje svih informacija iz telefonskog kanala propusnog opsega 4 kHz. Manjom

brzinom uzorkovanja ne bi se mogle prikupiti sve informacije, a uzorkovati većom brzinom nema smisla. Opisana tehnika zove se **impulsno-kodna modulacija** (engl. *Pulse Code Modulation, PCM).* PCM je jezgro savremene telefo- nije. Zbog njega su praktično svi vremenski intervali koji se koriste unutar telefonskog sistema umnošci od 125 ps.

Kada se digitalni prenos podataka pojavio kao obećavajuća tehnologija, organizacija CCITT nije uspela da postigne saglasnost u pogledu međunarodnog standarda za PCM. Zbog toga danas u zemljama širom sveta postoji više takvih, međusobno nekompatibilnih sistema.

U Sevemoj Americi i Japanu koristi se metoda tzv. **nosioca TI** (engl. *TI carrier),* prikazana na slici 2-33. (Ispravnije bi bilo reći daje format DS1, a daje nosilac TI, no držaćemo se stručnog žargona.) Nosilac T1 sadrži 24 međusobno multipleksirana govorna kanala. Analogni signali se obično uzorkuju ciklično, a rezultujući signal se upućuje u zajednički koder/dekoder (umesto u 24 kodera/dekodera za svaki pojedinačni signal); tako se odmah dobija zbirni digitalni signal, umesto da se 24 pojedinačna digitalna signala međusobno kombinuju. Svaki od 24 kanala ima mogućnost da u digitalni tok unese 8 bitova; sedam bitova podataka i jedan kontrolni bit, što ukupno daje 7 x 8000 = 56.000 b/s podataka i 1 x 8000 = 8000 b/s signalnih podataka po kanalu.



Slika 2-33. Nosilac TI (1,544 Mb/s).

Okvir sadrži 24 x 8 = 192 bita plus jedan bit za označavanje početka okvira, što ukupno iznosi 193 bita svakih 125 ps. Time dobijamo ukupnu brzinu 1,544 Mb/s. Dodatni sto devedeset treći (u stvari, prvi) bit služi za sinhronizovanje okvira i sledi šemu 0101010101... Primalac ga obično neprekidno proverava da bi bio siguran da nije narušena sinhronizacija. Ako iskoči iz sinhronizacije, primalac može ponovo da potraži ovu šemu i tako se opet sinhronizuje. Analogno povezani korisnici uopšte ne mogu da generišu taj niz bitova jer on odgovara sinusnom talasu frekvencije 4000 Hz koji se uklanja filtrom. Digitalno povezani korisnici mogu da ga generišu, ali je velika verovatnoća da on i ne postoji u okviru koji im promakne. Kada se sistem TI koristi isključivo za podatke, njima se namenjuje 23 kanala. Dvadeset četvrti kanal služi isključivo za sinhronizovanje i brži oporavak u slučaju propuštanja okvira.

Kada je CCITT konačno postigao saglasnost, shvaćeno je da 8000 b/s signalnih podataka previše opterećuje sistem, pa je standard brzine prenosa 1,544 Mb/s, umesto na 7-bitnom, zasnovan na 8-bitnom kanalu za podatke; drugim recima, analogni signal je, umesto na 128, izdeljen na 256 diskretnih nivoa. Predviđene su dve (međusobno nekompatibilne) varijante. Kod tzv. **zajedničkog signaliziranja za sve kanale** (engl. *common-channel signaling),* dodatni bit (koji je, umesto na početak, stavljan nakraj 193-bitnog okvira) sledi šemu 10101010... u neparnim okvirima, i sadrži sig- nalne informacije za sve kanale u parnim

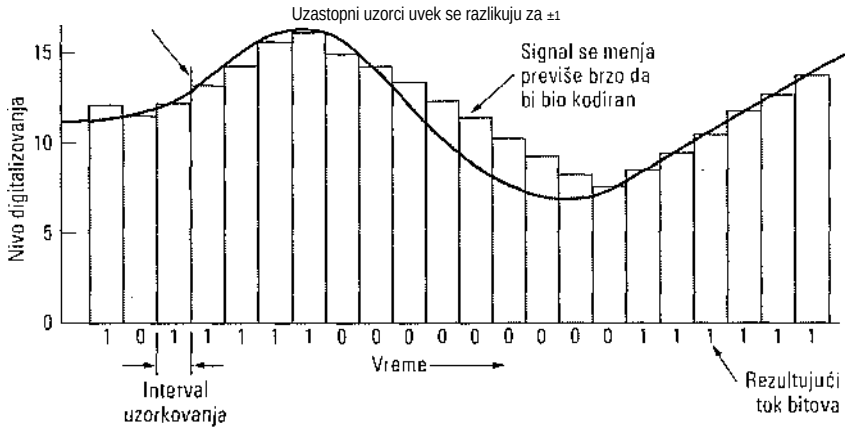
okvirima.

U drugoj varijanti, **signaliziranju za pojedinačne kanale** (engl. *channel-associated signaling),* svaki kanal ima sopstveni, privatni signalni potkanal. Privatni kanal se obrazuje tako što se jedan od osam bitova korisničkih podataka u svakom šestom okviru dodeli za signaliziranje, tako da su pet od šest uzoraka širine 8 bitova, a jedan širine 7 bitova. Organizacija CCITT preporučila je i PCM nosilac brzine prenosa 2,048 Mb/s, pod imenom El. Taj nosilac sadrži 32 8-bitna uzorka podataka spakovana u osnovni okvir od 125 ps. Trideset kanala se koristi za podatke, a dva za signalizaciju. Za svaku grupu od četiri okvira predviđena su 64 signalna bita, od kojih se polovina koristi za signalizaciju u pojedinačnim kanalima, a polovina za sinhronizovanje okvira ili kao rezerva koju svaka država može da upotrebi po svojoj volji. Izvan Severne Amerike i Japana, umesto nosioca TI koristi se nosilac El brzine 2,048 Mb/s.

Pošto se govorni signal pretvori u digitalni oblik, pokušava se da se pomoću statističkih tehnika smanji broj bitova potrebnih po kanalu. Tim tehnikama se može kodirati ne samo govor, već i bilo koji drugi analogni signal. Sve metode komprimovanja zasnivaju se na činjenici da se signal menja srazmerno sporo u odnosu na učestalost uzorkovanja, tako da znatan deo 7-bitnih ili 8-bitnih digitalnih informacija praktično predstavlja višak.

Pomoću metode zvane **diferencijalna impulsno-kodna modulacija** (engl. *differential pulse code modulation),* na izlazu se ne generiše digitalizovana amplituda, već razlika između njene trenutne vrednosti i prethodne. Pošto je mala verovatnoća pojave skokova od ±16 jedinica na skali od 128 jedinica, za skoro potpunu informaciju je umesto 7 bitova dovoljno samo 5. Kada signal povremeno „podivlja“, za ponovno uhodavanje potrebno je nekoliko ciklusa uzorkovanja. Kada se radi o govornom signalu, time se unosi zanemarljiva greška.

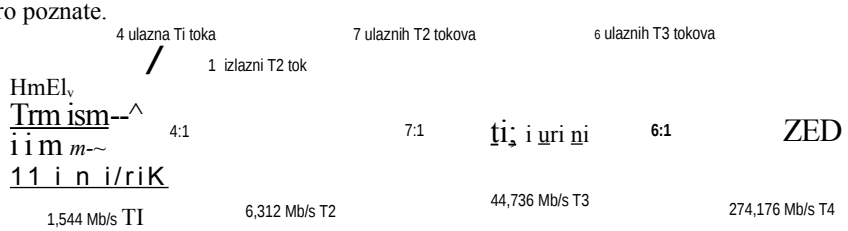
U jednoj varijanti opisane metode komprimovanja svaka uzorkovana vrednost mora da se razlikuje od prethodne za +1 ili za -1. U tim okolnostima, potrebno je poslati samo jedan bit koji označava da li uzorak ima veću ili manju vrednost od prethodnog uzorka. Ta tehnika, **delta modulacija,** prikazana je na slici 2-34. Kao i sve tehnike komprimovanja koje pretpostavljaju mali nivo promena između uzastopnih uzoraka, delta kodiranje može da ne uspe ako se signal menja prebrzo, baš kao što je prikazano na slici. U takvim situacijama, deo informacija se gubi.



Slika 2-34. Delta modulacija.

Postoji poboljšana tehnika impulsno-kodne modulacije, u kojoj se prethodnih nekoliko vrednosti ekstrapoliraju da bi se predvidela sledec'a vrednost i potom kodirala razlika između aktuelnog i predviđenog signala. Naravno da predajnik i prijemnik moraju da koriste isti algoritam za predviđanje. Takvi sistemi se zovu **prediktivno kodiranje** (engl. *predictive encoding).* Privlačni su zato što smanjuju veličinu brojeva koje treba kodirati, pa tako i broj bitova koji se šalju.

Multipleksiranje podelom vremena omogućava da se više TI nosilaca multiplek- sira u nosilac višeg reda. Slika 2-35 prikazuje kako se to može izvesti. Na levoj strani vidimo četiri TI kanala koji se multipleksiraju u jedan T2 kanal. Multipleksiranje na nivou T2 i višim nivoima ne vrši se bajt po bajt, već bit po bit, za 24 govorna kanala koji sačinjavaju TI okvir. Od četiri TI toka brzine 1,544 Mb/s treba da se dobije brzina prenosa 6,176 Mb/s, ali nosilac T2 u stvari ima veću brzinu - 6,312 Mb/s. Višak bitova se koristi za označavanje okvira i oporavak u slučaju propuštanja nosioca. Korisnici često upotrebljavaju nosioce TI i T3, dok se nosioci T2 i T4 koriste isključivo unutar telefonskog sistema i stoga njihove osobine nisu dobro poznate.



Slika 2-35. Multipleksiranje TI tokova u nosioce višeg reda.

Na sledecem nivou, sedam T2 tokova kombinuju se po bitovima obrazujući T3 tok. Zatim se šestT3 tokova stapaju u T4 tolc. Na svakom stupnju se dodaju bitovi za označavanje okvira i za oporavak u slučaju gubitka sinhronizacije između pošiljaoca i primaoca.

Kao stoje malo zajedničkog u pogledu osnovnog nosioca između Sjedinjenih Država i ostatka sveta, isto tako se razlikuju i mišljenja o tome kako nosioce treba multiplek- sirati na višim nivoima. Američka šema sa 4,7 i 6 sjedinjavanja na uzastopno rastućim nivoima nije recept za svakoga, tako da CCITT standard preporučuje multipleksiranje četiri toka na svakom nivou. U američkoj šemi i CCITT standardu razlikuju se i bitovi za označavanje okvira, odnosno za oporavljanje posle gubitka sinhronizacije. Hijerarhija koju preporučuje CCITT sadrži 32,128,512, 2048 i 8192 kanala sa odgovarajućim brzinama prenosa podataka 2,048 Mb/s, 8,848 Mb/s, 34,304 Mb/s, 139,264 Mb/s i 565,148 Mb/s.

SONET/SDH

Na početku uvođenja optičkih vlakana, svaka telefonska kompanija je imala sopst- veni optički TDM sistem. Posle raspada korporacije AT&T 1984. godine, lokalne telefonske kompanije su morale da se povezuju s međugradskim centralama od kojih je svaka imala drugačiji optički TDM sistem, tako da je potreba za standardizovanjem ove oblasti postala urgentna. Godine 1985, Bellcore, istraživački sektor regionalnih Belo- vih telefonskih centrala, počeo da radi na standardu sinhrone optičke mreže (engl. *Synchronous Optical NETwork, SONE7*Kasnije se poslu pridružila i organizacija CCITT, odakle je 1989. proizišao standard SONET i skup paralelnih CCITT preporuka (G.707, G.708 i G.709), poznatih pod imenom sinhrona digitalna hijerarhija (engl. *Synchronous Digital Hierarchy, SDH),* koje se od SONET-a samo neznatno razlikuju. Danas skoro sav međugradski telefonski saobraćaj u Sjedinjenim Državama - i ne samo u njima - ide vodovima koji u fizičkom sloju koriste SONET. Detaljnije podatke o SONET-u potražite kod Bellamyja (2000), Goralskog (2000) i Sheparda (2001).

Projekat SONET trebalo je da ostvari četiri glavna cilja. Prvo i najvažnije, trebalo je da obezbedi međusobnu saradnju različitih nosilaca podataka. Zbog toga je bilo neophodno usvojiti opšti standard za signaliziranje u pogledu talasne dužine, vremenskog raspoređivanja, strukture okvira i sličnih stavki.

Drugo, bilo je potrebno pronaći način da se objedine američki, evropski i japanski digitalni sistemi, koji doduše svi koriste PCM kanale brzine 64 kb/s, ali ih svako kom- binuje na drugačiji način (nekompatibilan sa ostalima).

Treće, SONET treba da ponudi način za multipleksiranje višestrukih digitalnih kanala. U vreme nastanka SONET-a, najbrži digitalni nosilac podataka u SAD bio je T3, brzine 44,736 Mb/s. T4 je već bio definisan, ali ne i mnogo korišćen, a iznad toga ništa nije bilo ni definisano. Deo misije SONET-a bio je i da produži hijerarhiju do gi- gabajtnih brzina, a i dalje. Trebalo je ponuditi i standardnu metodu multipleksiranja sporijih kanala u jedinstven SONET kanal.

Četvrto, SONET je morao da obezbedi podršku za rad, administriranje i održavanje (engl. *operations, administration, maintenance, OAM),* bolje od ranijih sistema.

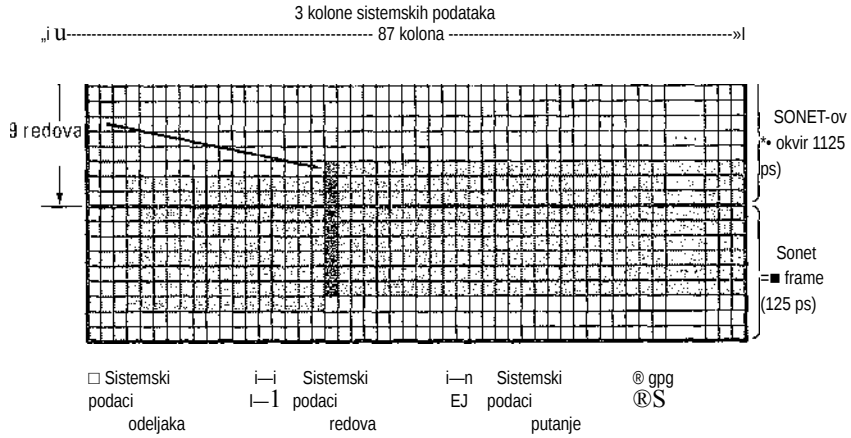
Srazmerno rano je odlučeno da SONET postane klasični TDM sistem u kome ceo propusni opseg vlakna zauzima jedan kanal vremenski raspodeljen na više potkanala. Tako zamišljen, SONET je predstavljao sinhroni sistem kojim upravlja glavni časov- nik, tačnosti

$1:10^9$. Bitovi se SONET linijom šalju u izuzetno preciznim vremenskim intervalima koje određuje časovnik. Kada je kasnije predloženo komutiranje ćelija kao osnova ATM-a, zbog mogućnosti neravnomernog pristizanja delija taj režim je označen kao *asinkroni* režim prenosa da bi se jasno razlikovao od sinhronog rada SO- NET-a. U SONET-u se rad pošiljaoca i primaoca koordinira zajedničkim časovni- korn, dok u asinhronom prenosu to nije slučaj.

Osnovni okvir sistema SONET predstavlja blok od 810 bajtova koji se isporučuje svakih 125 ps. Pošto SONET radi sinhrono, okviri se emituju bez obzira na to da li postoje podaci koje treba poslati. Brzina od 8000 okvira u sekundi tačno odgovara brzini uzorkovanja PCM kanala koji se koriste u svim digitalnim telefonskim sistemima.

SONET-ove okvire od 810 bajtova najlakše je shvatiti kao tabelu sa 90 kolona i 9 redova. Na taj način se 8 x 810 = 6480 bitova prenosi 8000 puta u sekundi, dajući ukupnu brzinu prenosa 51,84 Mb/s. Taj osnovni kanal sistema SONET zove se **STS-1 (Synchronous Transport Signal-1).** Svi vodovi SONET-a sadrže više ovih osnovnih kanala.

Prve tri kolone svakog okvira rezervisane su za sistemske podatke (slika 2-36). U njima prva tri reda sadrže sistemske podatke odeljaka (engl. *section overhead*), a preostalih šest sistemske podatke redova (engl. *line overhead).* Sistemski podaci odeljaka generišu se i proveravaju na početku i na kraju svakog odeljka, a sistemski podaci redova generišu se i proveravaju na početku i na kraju svakog reda.



Slika 2-36. Dva susedna SONET-ova okvira.

Predajnik u sistemu SONET šalje okvire dužine 810 bajtova, jedan za drugim, bez razmaka, čak i kada nema podataka za slanje (tada šalje prazne podatke). Prijemnik vidi samo neprekidan tok bajtova, koji logički deli na okvire tako što traži fiksni šablon sadržan u prva dva bajta svakog okvira. Kada šablon pronađe na istom mestu u velikom broju uzastopnih okvira, on tada smatra daje sinhronizovan s pošiljaocem. Teorijski bi i korisnik ovaj šablon mogao da redovno ugrađuje u svoje podatke, ali bi od toga bilo male praktične koristi zbog multipleksiranja podataka više korisnika u istom okviru i drugih razloga.

Preostalih 87 kolona sadrže korisničke podatke koji se, dakle, prenose brzinom od 87 x 9 x 8 x 8000 = 50,112 Mb/s. Međutim, tzv. **sinhroni korisnički podaci** (engl. *Synchronous*

Payload Envelope, SPE) ne počinju uvek u 4. koloni prvog reda; oni mogu početi bilo gde unutar okvira. Pokazivač na prvi bajt podataka nalazi se u prvom redu sistemskih podataka koji se odnose na redove. Prva kolona SPE podataka sadrži sistemske podatke koji se odnose na putanju (zaglavlje protokola podsloja koji povezuje dve strane).

Fleksibilnosti sistema SONET doprinosi i mogućnost da SPE podaci počnu bilo gde unutar okvira i da se rastegnu i na sledeći okvir, kao što je prikazano na slici 2-36. Na primer, ako se na izvoru pojave stvarni podaci u trenutku dok SONET formira okvir s praznim podacima, oni mogu početi da se unose i u njega, umesto da čekaju formiranje sledećeg okvira.

Hijerarhija multipleksiranja u sistemu SONET prikazana je na slici 2-37. Defini- sane su brzine prenosa od STS-1 do STS-192. Optički nosilac koji odgovara nosiocu STS-n ima oznaku OC-n, ali su im svi bitovi isti, osim što su u izvesnoj meri drugačije raspoređeni zbog sinhronizovan)a. Odgovarajuća SDH imena su drugačija - počinju sa OC-3 jer sistemi zasnovani na preporukama organizacije CCITT ne rade brzinama bliskim .51,84 Mb/s. Naveden je i nosilac OC-9 zato što dobro odgovara brzini glavnih visokobrzinskih vodova u Japanu. U Japanu se koriste i nosioci OC-18, odnosno OC-36. Ukupna brzina prenosa obuhvata i prenos svih sistemskih podataka. U brzinu prenosa SPE podataka nisu uključeni sistemski podaci odeljaka i redova. Iz brzine prenosa korisničkih podataka isključeni su svi sistemski podaci (sistemski podaci koji se odnose na putanju) - oni se prostim samo na 86 kolona.

SONET		SDH	Brzina prenosa (Mb/s)		
Električni nosilac	Optički nosilac	Optički nosilac	Ukupna	SPE	Korisničkih podataka
STS-1	OC-1		51,84	50,112	49,536
STS-3	OC-3	STM-1	155,52	150,336	148,608
STS-9	OC-9	STM-3	466,56	451,008	445,824
STS-12	OC-12	STM-4	622,08	601,344	594,432
STS-18	OC-18	STM-6	933,12	902,016	891,648
STS-24	OC-24	STM-8	1244,16	1202,688	1188,864
STS-36	OC-36	STM-12	1866,24	1804,032	1783,296
STS-48	OC-48	STM-16	2488,32	2405,376	2377,728
STS-192	OC-192	STM-64	9953,28	9621,504	9510,912

Slika 2-37. Multipleksna brzina prenosa u sistemima SONET i SDH.

Kada nosilac, kao što je OC-3, nije multipleksiran, već nosi podatke samo iz jednog izvora, oznaci se dodaje jedno *c* (od engl. *concatenated* - nadovezan), tako da OC-3 označava nosioca brzine 155,52 Mb/s koji se sastoji od tri odvojena nosioca OC-1, a OC-3c označava tok podataka iz jedinstvenog izvora, brzine 155,52 Mb/s. Unutar okvira OC-3c, tri toka OC-1 preplicu se redom, po kolonama: prvo dolazi kolona 1 iz toka 1, zatim kolona 1 iz toka 2, zatim kolona 1 iz toka 3, pa kolona 2 iz toka 1 itd., obrazujući okvir sa 270 kolona i 9 redova podataka.

2.5.5 Komutiran je

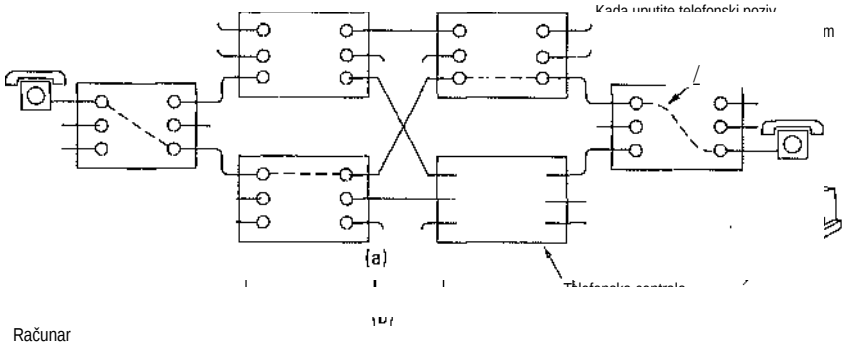
Sa gledišta prosečnog inženjera telefonije, telefonski sistem ima dva glavna dela: spoljne uređaje (lokalne linije i vodove) koji su izvan telefonske centrale i unutrašnje uređaje (komutatore), koji su smešteni unutar telefonske centrale. Upravo smo pretresli spoljne uređaje, pa je vreme da se pozabavimo i unutrašnjim.

Danas se uglavnom koriste dve tehnike komutiranja: komutiranje električnih kola i komutiranje samih paketa. U nastavku ćemo ukratko opisati svaku od njih, a zatim ćemo preći na detaljno izlaganje o komutiranju električnih kola jer tako radi telefonski sistem. Komutiranje paketa proučićemo u narednim poglavljima.

Komutiranje električnih kola

Kada lično ili preko računara uputite telefonski poziv, komutatori unutar telefonskog sistema uspostavljaju fizičku vezu između vašeg i sagovornikovog telefonskog aparata. Ta tehnika se zove komutiranje električnih kola (engl. *circuit switching*) i prikazana je šematski na slici 2-38(a). Svaki od šest pravougaonilca na slici predstavlja po jednu telefonsku centralu (lokalnu, regionalnu itd.). U ovom primeru, svaka centrala ima tri ulazne i tri izlazne linije. Kada poziv prođe kroz centralu, uspostavlja se fizička veza (prikazana isprekidano) između linije kojom je stigao poziv i jedne od izlaznih linija.

Na početku primene telefonije, vezu je uspostavljao operater povezujući ulaznu i izlaznu utičnicu kablom. U stvari, pronalazak uređaja za automatsko komutiranje prati čudna anegdota. Taj uređaj je u 19. velcu napravio pogrebnik Alrnon B. Strowger iz države Misuri. Ubrzo posle pojave telefona, kada bi neko umro, osoba iz pokojnikove okoline pozvala bi gradskog operatera i rekla „Molim vas da me povežete s nekim pogrebnikom“. Na nesreću g. Strowgera, u gradu su postojala dva pogrebnika, a supruga onog drugog bila je gradski telefonski operater. G. Strowger je odmah shvatio da mora da izmisli uređaj za automatsko komutiranje poziva ili da potpuno napusti posao kojim se bavi. Odabrao je ono prvo. Uređaj koji je napravio gotovo 100 godina je bio poznat kao Strowgerov uređaj. (Istorija nije zabeležila da lije dotadašnji telefonski operater postao operater službe za informacije, gde bi mogao da odgovara na pitanje „Dajte mi telefonski broj nekog pogrebnika“.)



Slika 2-38. (a) Komutiranje električnih kola, (b) Komutiranje paketa.

Model prikazan na slici 2-3 8(a) veoma je uprošćen, naravno, pošto delovi fizičke putanje između dva telefonska aparata mogu da se ostvaruju mikrotalasima ili optičkim kablom kroz koji prolaze hiljade multipleksiranih poziva. Pa ipak, prikazani princip je ispravan: kada se uputi poziv, uspostavlja se namenska putanja između dva telefonska aparata koja se ne menja sve dok se razgovor ne završi.

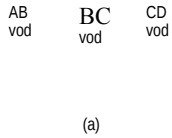
Alternativa komutiranju električnih kola je komutiranje paketa, prikazano na slici 2-38(b). Prema toj tehnologiji, pojedinačni paketi se šalju kada je potrebno, putanjom koja nije unapred određena. Svaki paket samostalno pronalazi put do odredišta.

Važna osobina komutiranja električnih kola jeste potreba da se putanja između dva kraja uspostavi *pre* nego što se pošalju bilo kakvi podaci. Vreme koje protekne između biranja poslednje cifre telefonskog broja i trenutka kada telefon na drugom kraju zazvoni može da bude i 10 s, naročito na međugradskim ih međudržavnim linijama. Tokom tog vremena, telefonski sistem pronalazi putanju, što je prikazano na slici 2-39(a). Obratite pažnju na to da signal za uspostavljanje poziva mora da stigne do odredišta i da potvrda o tome mora da stigne nazad pre nego što se pošalju bilo kakvi podaci. Za mnoge računarske primene (na primer, za proveru statusa kreditnih kartica u prodav- nicama), dugo vreme uspostavljanja veze nije prihvatljivo.

Zbog toga što se veza između dva kraja rezerviše samo za taj razgovor, kada se ona uspostavi, podaci se razmenjuju brzinom prostiranja elektromagnetnih talasa kroz medijum - tu je kašnjenje oko 5 ms na 1000 km. Isto tako, ovde nema bojazni od zagušenja: kada se veza uspostavi, više ne možete čuti signal zauzeća do kraja razgovora. Naravno, signal zauzeća možete da dobijete dok se veza ne uspostavi, ukoliko u tom trenutku nema slobodnih kapaciteta za komutiranje ili prenos.

Komutiranje poruka

Alternativna strategija je **komutiranje poruka** (engl. *message switching),* prikazano na slici 2-39(b). Ovde se putanja između pošiljaoca i primaoca ne uspostavlja una- pred. Umesto toga, kada pošiljalac ima spreman blok podataka za slanje, on ih sldadišti u prvoj telefonskoj centrali (tj. u usmerivaču), odakle se kasnije skokovito prosleđuju. Kao što smo pomenuli u 1. poglavlju, takva mreža radi po principu **„čuvaj i prosledi“** (engl. *store-anđ-forwarđ).*



Slika 2-39. Vremenski sled događaja pri (a) komutiranju električnih kola, (b) komutiranju poruka, (c) komutiranju paketa.

Prvi elektromehanički telekomunikacioni sistemi koristili su komutiranje poruka, konkretno, za slanje telegrama. Poruka je u polaznoj telegrafskoj stanici prenošena na papirnu traku (bušenjem), zatim je traka provlačena kroz specijalan uređaj za čitanje i podaci su slati komunikacionom linijom do sledeće telegrafske stanice, gde su opet bušenjem prenošeni na papirnu traku. Operater u toj stanici otkinuo bi izbušenu traku i podatke s nje učitao ponovo pomoću jednog od više čitača, povezanih sa izlaznim vodovima. To su bile **telegrafske stanice s čitačem/bušaćem trake** (engl. *tom tape office)*. Papirna traka je davno zaboravljena i komutiranje poruka se više ne koristi, pa ćemo ovde završiti njegovo opisivanje.

Komutiranje paketa

Pri komutiranju poruka, veličina bloka nije ograničena, što znači da usmerivači (u savremenim sistemima) moraju imati diskove za privremeno skladištenje dugačkih blokova. Takođe se može desiti da dugačak blok minutima zauzme liniju između dva usmerivača i time učini beskorisnim komutiranje poruka u interaktivnom saobraćaju. Radi rešavanja takvih problema uvedeno je **komutiranje paketa** (engl. *packet switching),* kao što je opisano u 1. poglavlju. Mreže koje rade s komutiranjem paketa strogo ograničavaju veličinu bloka i time omogućuju da se paketi skladište u glavnoj memoriji usmerivača, a ne na disku. Sprečavajući korisnike da zauzmu liniju tokom dužeg vremena (merenog milisekundama), mreže s komutiranjem paketa omogućuju interaktivan saobraćaj. Druge prednosti komutiranja paketa u odnosu na komutiranje poruka prikazane su na slikama 2-39(b) i (c): prvi paket poruke od više paketa može se proslediti pre nego što drugi paket bude u potpunosti primljen, čime se smanjuje kašnjenje i ubrzava saobraćaj. Iz ovih razloga, računarske mreže obično koriste komutiranje paketa, ponekad i komutiranje električnih kola, ali nikada komutiranje poruka.

Između komutiranja električnih kola i komutiranja paketa postoje mnoge razlike. Prvo, pri komutiranju električnih kola neophodno je da se pre stvarne komunikacije uspostavi električno kolo od jednog do dragog kraja. Za komutiranje paketa ne treba unapred uspostavljati vezu. Prvi paket se može poslati čim za tim nastane potreba.

Prethodnim uspostavljanjem veze u mreži koja radi s komutiranjem električnih kola, rezerviše se propusni opseg na čitavom putu od pošiljaoca do primaoca. Svi paketi slede taj put. Zbog toga, između ostalog, oni ne mogu da na odredište stignu bilo kako, već redom kojim su poslati. Pri radu s komutiranjem paketa ne postoji unapred određena putanja, pa različiti paketi mogu da slede različite putanje u zavisnosti od aktuelne gustine saobraćaja na mreži. Zbog toga oni na odredište mogu da stignu bilo kojim redosledom.

Komutiranje paketa je otpornije na greške nego komutiranje električnih kola. To je i glavni razlog zbog koga je komutiranje paketa smišljeno. Ako otkaže skretnica, prekidaju se sva električna kola koja je koriste i preko njih se više ne mogu slati poruke. Komutirani paketi, međutim, mogu pronaći put koji zaobilazi pokvarenu skretnicu.

Prethodno određivanje putanje omogućava i da se propusni opseg rezerviše unapred. Kada je propusni opseg rezervisan, svaki pridošli paket odmah se prosleđuje. Za komutirane pakete se, međutim, ne rezerviše propusni opseg, pa pojedini paketi mogu da budu prosleđeni sa zadrškom.

Kada rezervišete propusni opseg, ne može doći do zagušenja (osim ako pokušate da istovremeno prosledite više od očekivanog broja paketa). S drage strane, pokušaj da se uspostavi električna veza od kraja do kraja može da ne uspe upravo zbog zagušenja linija. Dakle, problem zagušenja se može javiti i pri komutiranju električnih kola (tokom uspostavljanja veze), i pri komutiranju paketa (pri slanju paketa).

Ako je električno kolo rezervisano za određenog korisnika, a nema podataka za slanje, propusni opseg tog kola uludo je straćen i ne može se iskoristiti za drugi saobraćaj. Pri komutiranju paketa, s propusnim opsegom se postupa racionalnije, pa je - sa aspekta sistema - komutiranje paketa efikasnija metoda. Uočavanje ovog kompromisa ključno je za razumevanje razlike između komutiranja električnih kola i komutiranja paketa. Dakle, možete garantovati uslugu i razbacivati se resursima, ili pružati uslugu bez garancije i štedeti

resurse.

Komutiranje paketa se izvodi po principu „čuvaj i prosledi“. Paketi se skupljaju u memoriji usmerivača, a zatim prosleđuju sledećem usmerivaču. Tehnika „čuvaj i prosledi“ dovodi do kašnjenja paketa.

Druga razlika između ove dve vrste komutiranja ogleda se u tome stoje komutiranje električnih kola potpuno transparentno - između dve strane ostvaruje se direktna veza. Pošiljalac i primalac mogu da rade bilo kojom brzinom, da koriste bilo koji format ili način formiranja okvira. Nosilac podataka o tome ne vodi brigu. Pri komutiranju paketa, međutim, nosilac određuje osnovne parametre prenosa. Razlika između ove dve situacije može se grubo uporediti s razlikom između autoputa i železničkog koloseka. Na autoputu korisnik određuje veličinu, brzinu i prirodu vozila, dok na železničkoj prazi kolosek određuje šta će i kako biti prevoženo. Zbog pomenute transparentnosti, kroz telefonski sistem se ravnopravno mogu prenositi govor, faks poruke ili podaci.

Posleđnju razliku između komutiranja električnih kola i komutiranja paketa predstavlja algoritam za naplaćivanje usluga. Kod komutiranja kola, naplaćivanje je tradicionalno zavisilo od udaljenosti i vremena. Za mobilnu telefoniju udaljenost ne igra nikakvu ulogu, osim kod međunarodnih razgovora, a vreme korišćenja usluge je manje značajno (npr. pretplata za 2000 minuta korišćenja veća je od pretplate za 1000 minuta, a ponekad su pozivi upućeni vikendom i noću jeftiniji od uobičajenih). Kod komutiranja paketa, vreme provedeno na vezi ne igra nikakvu ulogu, ali je obim saobraćaja ponekad važan. Kućnim korisnicima davaoci Internet usluga obično nude paušalno mesečno plaćanje usluga jer ono znači manje posla za davaoca, a korisniku je način korišćenja usluge potpuno jasan. S drage strane, kompanije koje održavaju okosnicu zahtevaju od regionalnih mreža naknadu zasnovanu na obimu saobraćaja. Razlike između dve vrste komutiranja sažete su na slici 2-40.

Tehnike komutiranja električnih kola i komutiranja paketa ipak su veoma važne, pa ćemo se na njih uskoro vratiti i detaljno opisati tehnologije koje se za njih koriste.

Stavka	Komutiranje električnih kola	Komutiranje paketa
Uspostavljanje veze	Obavezno	Nepotrebno
Unapred određena fizička putanja	Da	Ne
Svaki paket sledi istu putanju	Da	Ne
Paketi stižu redom kojim se šalju	Da	Ne
Otkazivanje skretnice	Pogubno po vezu	Nebitno
Raspoloživ propusni opseg	Unapred određen	Promenljiv
Moguće zagušenje	Tokom uspostavljanja veze	Pri prenosu svakog paketa
Zauzimanje propusnog opsega	Rasipničko	Štedljivo
Prenos tipa „čuvaj i prosledi“	Ne	Da
Transparentnost	Da	Ne
Naplaćivanje usluge	Po minutu	Po paketu

Slika 2-40. Poređenje mreža koje rade s komutiranjem električnih kola, i onih s komutiranjem paketa.

2.6 SISTEM MOBILNE TELEFONIJE

Klasičan telefonski sistem (čak i ako jednoga dana bude sav od optičkih vlakana brzine prenosa više gigabita u sekundi) i dalje neće moći da zadovolji grupu koja svakim danom postaje sve veća: pokretne korisnike. Ljudi očekuju da sada mogu da telefoniraju iz aviona, iz automobila, iz bazena i iz parka dok džogiraju. Za par godina će tražiti da sa istih lokacija šalju e-poštu, krstare Webom i ko zna šta još. Iz tih razloga stalno raste zanimanje za bežičnu telefoniju. U narednim odeljcima ovu temu ćemo detaljnije proučiti.

Postoje dve osnovne varijante bežičnih telefona: fiksni i mobilni. **Fiksni bežični telefoni** (engl. *cordless phones*) imaju fiksnu bazu koja je u bežičnoj vezi s preno- sivom slušalicom, i namenjeni su kućnom korišćenju. Pošto se ovakvi telefoni nikada ne koriste u mrežama, nećemo se više na njima zadržavati, već ćemo preći na mobilne sisteme lcoji se koriste za regionalni prenos glasa i podataka.

Postoje već tri generacije **mobilnih telefona** (engl. *mobile phones, cell phones),* svaka s drugačijom tehnologijom:

1. Analogni prenos glasa.
2. Digitalni prenos glasa.
3. Digitalni prenos glasa i podataka (Internet, e-pošta itd.).

Iako ćemo uglavnom govoriti o tehnologiji ovih sistema, zanimljivo je ukazati na uticaj koji na njih imaju politika i male tržišne razlike. Prvi sistem mobilne telefonije u SAD ostvarila je korporacija AT&T, a organizacija FCC je sistem proglasila obaveznim za čitavu zemlju. Tako su SAD dobile jedinstven (analogni) sistem na čitavoj teritoriji i mobilni telefoni kupljeni u Kaliforniji radili su i u Njujorku. Nasuprot tome, kada je mobilna telefonija stigla u Evropu, svaka država je smislila sopstveni sistem, što je rezultovalo komunikacionom katastrofom.

Evropa je tako na teži način naučila šta treba činiti, i kada se pojavio digitalni sistem mobilne telefonije, državne PTT organizacije su se sastale i standardizovale jedinstven sistem (GSM), tako da svaki mobilni telefon proizveden u Evropi radi u svakoj evropskoj državi. U to vreme je u SAD zaključeno da vlada ne treba da propisuje standarde, pa je standardizacija digitalnog sistema mobilne telefonije ostavljena tržištu. Odmah su se pojavili različiti proizvođači koji su nudili različite tipove mobilnih telefona. Zbog toga danas u Sjedinjenim Državama postoje dva glavna, međusobno nekompatibilna sistema mobilne telefonije (i još jedan manji).

Uprkos početnom vodstvu SAD, danas u Evropi ima više operatera mobilne telefonije i više njenih korisnika. Jedan razlog je jedinstven evropski sistem mobilne telefonije, ali nije i jedini. Draga oblast u kojoj se SAD i Evropa razlikuju tiče se telefonskih brojeva. U SAD, brojevi mobilnih telefona ni po čemu se ne razlikuju od brojeva fiksnih telefona. Tako pozivalac ne može znati da li je (212) 234-5678 broj fiksnog telefona (jeftin ili besplatan razgovor) ili broj mobilnog telefona (skup razgovor). Da bi korisnike oslobodile tog straha, telefonske kompanije su odlučile da vlasnik mobilnog telefona plaća za pozive koji su mu upućeni. Iz tog razloga mnogi oklevaju da kupe mobilni telefon, strepeći od visokih računa koje im mogu napraviti drugi. U Evropi, mobilni telefoni imaju poseban pozivni broj po kome su prepoznatljivi. Zbog toga se u Evropi primenjuje uobičajeno pravilo da pozivalac plaća poziv (osim za međunarodne razgovore čiji se troškovi dele).

Treći razlog prihvatanja mobilne telefonije u Evropi jeste rasprostranjeno kori- šćenje

„pripejd“ telefona (i do 75% u nekim područjima). Takav telefon možete kupiti u prodavnici - baš kao i običan tranzistorski radio-prijemnik. S njim kupujete i razgovore u vrednosti 20 ili 50 evra, a kada ih potrošite, možete ih ponovo uplatiti pomoću skrivenog PIN koda. Na taj način, praktično svaki tinejdžer i mnoga mlađa deca u Evropi imaju (obično „pripejd“) mobilne telefone, pa njihovi roditelji uvek mogu da ih nađu, a deca pri tome ne mogu da ih iznenade velikim telefonskim računom. Ako se takav mobilni telefon koristi samo povremeno, njegovo korišćenje je skoro besplatno jer nema mesečne pretplate ili naknade za dolazne pozive.

2.6.1 Mobilna **telefonija** prve generacije: analogni prenos glasa

A sada, dosta o politici i tržišnim aspektima mobilne telefonije. Pređimo na tehnologiju i počnimo od najstarijeg sistema. Mobilna telefonija se sporadično koristila u pomorstvu i vojnim komunikacijama tokom prvih decenija 20. veka. Prvi sistem za telefoniranje iz automobila počeo je da radi 1946. u Sent Luisu. U njemu je korišćen veliki primopredajnik smešten na vrh visoke zgrade; imao je samo jedan kanal i za emitovanje i za prijem. Kada je želeo da govori, korisnik bi pritisnuo dugme i time uključio predajnik, a isključio prijemnik. Takvi, jednokanalni sistemi mobilne telefonije (engl. *push-to-talk systems*), počev od kraja pedesetih godina instalirani su u više gradova. CB-radio, taksi i policijska vozila u televizijskim programima često koriste tu tehnologiju.

Šezdesetih godina je instaliran poboljšani sistem mobilne telefonije (engl. *Improved Mobile Telephone System, IMTS)*. I u njemu je korišćen jak primopredajnik (200 W), smešten na vrh brda, ali je sistem sada imao dve frekvencije, jednu za slanje i drugu za primanje, tako da je dugme za pritiskanje postalo suvišno. Pošto se prijem i emitovanje odvijaju na dva različita kanala, korisnici nisu mogli da čuju i druge korisnike, kao stoje to slučaj kod sistema u taksi vozilima.

IMTS je podržavao 23 kanala rasuta u frekventnom opsegu između 150 i 450 MHz. Zbog malog broja kanala korisnik je često dugo čekao da dobije signal. Isto tako, zbog velike snage primopredajnika „na brdu“, susedni sistemi su morah biti međusobno udaljeni više stotina kilometara da bi se izbeglo ometanje. Kada se sve sabere, sistem je bio nepraktičan zbog niskog kapaciteta.

Napredni sistem mobilne telefonije

Sve se promenilo kada je 1982. u SAD uveden **napredni sistem mobilne telefonije** (engl. *Advanced Mobile Phone System*, *AMPS),* projektovan u Belovim laboratory ama. Uveden je i u Engleskoj, pod imenom TACS, kao i u Japanu, pod imenom MCS-L1. Iako je AMPS danas prevaziđen, razmotrićemo ga detaljnije, budući da su u sistemu D-AMPS, njegovom direktnom digitalnom nasledniku, zadržane mnoge osobine zbog kompatibilnosti sa starim sistemom.

U svim sistemima mobilne telefonije geografsko područje se deli na **ćelije** (engl. *cells),* zbog čega se mobilni telefoni ponekađa nazivaju celularni telefoni. U sistemu AMPS ćelije su obično prečnika 10 do 20 km; u digitalnim sistemima ćelije su manje. U svakoj ćeliji se koristi skup frekvencija različit od frekvencija koje koriste susedi. Osnovno načelo zbog koga sistem ćelija ima veći kapacitet od prethodnih sistema, ogleda se u formiranju srazmerno malih ćelija i višestrukom korišcenju istih frekvencija u bliskim (ali ne u susednim) ćelijama. Dok se u sistemu IMTS koji pokriva područje od 100 km svakoj

frekvenciji pridružuje samo jedan poziv, sistem AMPS na istom području može imati ćelije prečnika 10 km tako da na svakoj frekvenciji može podržati 10 do 15 poziva u međusobno udaljenim ćelijama. Na taj način se u sistemu ćelija kapacitet povećava barem za red veličine - i to sve više, što su ćelije manje. Osim toga, kad su ćelije manje, predajnici i telefoni mogu biti jeftiniji i raditi manjom snagom. Ručni mobilni telefoni imaju snagu 0,6 vati, a oni u automobilima 3 vata i to je maksimum koji dozvoljava FCC.

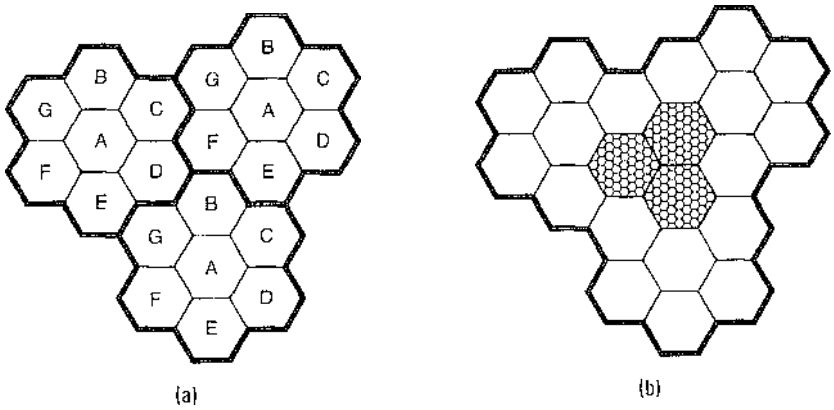
Načelo višestrukog korišćenja istih frekvencija ilustrovano je slikom 2-41(a). Ćelije u stvari nisu šestougaone, već približno kružne, ali ih je ovako lakše prikazati. Na slici 2-41(a) sve ćelije su iste veličine i razmeštene u grupe od po 7 ćelija. Svako slovo označava jedan skup frekvencija. Obratite pažnju na to da između ćelija obeleženih istim slovom postoje barem dve drugačije označene ćelije, čime se postiže dobro razdvajanje i neznatno međusobno ometanje telefona koji rade na istim frekvencijama.

U mobilnoj telefoniji glavni problem je naći uzvišeno mesto za postavljanje antene primopredajne bazne stanice (repetitora). Neki operateri su čak ušli u pregovore s Rimokatoličkom crkvom, jer ta organizacija širom sveta poseduje brojne istaknute visinske tačke, zgodne za postavljanje antena mobilne telefonije.

U područjima gde zbog sve većeg broja korisnika sistem postaje preopterećen, ćelije se dele na manje **mikroćelije** (engl. *microcells)* i snaga sistema se smanjuje u odgovarajućoj meri, čime se omogućava češće korišćenje istih frekvencija, kao na slici 2-41(b). Telefonske kompanije ponekada za popularne sportske događaje, rok-koncer- te i druge manifestacije na kojima se tokom nekoliko sati okuplja veliki broj korisnika mobilnih telefona, prave privremene mikroćelije, koristeći prenosne stubove sa satelitskim vezama. Određivanje optimalne veličine ćelije je složen zadatak (Hac, 1995).

U centru ćelije nalazi se bazna stanica s kojom su u vezi svi mobilni telefoni unutar ćelije. Baznu stanicu čine računar i primopredajnik povezan sa antenom. U malim sistemima, sve bazne stanice su povezane s **centralom sistema mobilne telefonije**

(engl. *Mobile Telephone Switching Office, MTSO* ili *Mobile Switching Center, MSC).* Veći sistemi mogu imati više centrala koje se povezuju s centralom višeg nivoa itd. Centrale igraju ulogu lokalnih telefonskih centrala i u stvari i jesu spojene s barem jednom lokalnom centralom sistema fiksne telefonije. Centrale mobilne telefonije komuniciraju s baznim stanicama, između sebe i s javnom komutiranom telefonskom mrežom, koristeći tehniku



kornutiranja paketa.

Slika 2-41. (a) Iste frekvencije se ne koriste ponovo u susednim ćelijama. (b) Kada se upotrebe male ćelije, može se uslužiti više korisnika.

Svaki mobilni telefon se u svakom trenutku logički nalazi u određenoj ćeliji, pod kontrolom bazne stanice u toj ćeliji. Kada mobilni telefon fizički napusti ćeliju, njena bazna stanica zapaža da signal slabi i svim okolnim ćelijama šalje upit o snazi signala koji primaju s tog telefona. Baza tada predaje kontrolu ćeliji koja najjače prima signal, tj. onoj u kojoj se telefon sada nalazi. Telefonu se šalje obaveštenje o njegovom „novom šefu" i, ako je razgovor u toku, od njega se traži da pređe na dragi kanal (pošto se stari kanal ne koristi u susednim ćelijama). Opisani proces **predavanja upravljanja** (engl. *handoff)* traje oko 300 ms. Nov kanal dodeljuje MTSO - mozak sistema. Bazne stanice nisu ništa drugo do radio-releji.

Predavanje upravljanja može se izvesti na dva načina. U tzv. **mekom predavanju** (engl. *soft handoff),* nova bazna stanica preuzima kontrolu nad telefonom pre nego što se prethodna baza odjavila. Na taj način se kontinuitet potpuno održava. Nezgodno je to što telefon mora istovremeno da radi na dve frekvencije (na staroj i na novoj), što ne mogu ni uređaji prve, ni uređaji druge generacije.

U **oštrom predavanju** (engl. *hard handoff),* prethodna bazna stanica prekida kontrolu nad telefonom pre nego što je nova baza preuzme. Ako nova baza ne uspe da preuzme kontrolu nad telefonom (npr. zato što nema raspoložive frekvencije), veza se naglo prekida. Naravno da se korisnici zbog toga ne vesele, ali je, nažalost, u današnjim sistemima to ponekad neizbežno.

Kanali

Sistem AMPS koristi 832 potpuna dupleksna kanala, svaki s dva kanala za prenos u suprotnim smerovima. Postoje 832 jednosmerna predajna kanala u frekventnom opsegu između 824 i 849 MHz i 832 jednosmerna prijemna kanala u opsegu između 869 i 894 MHz. Svaki jednosmerni kanal ima propusni opseg od .30 lcHz. Na taj način, kanali se u sistemu AMPS ostvaruju tehnikom modulisanja podelom frekvencija.

U području frekvencija oko 800 MHz, radio-talasi su dužine oko 40 cm i prostim se pravolinijski. Vegetacija ih apsorbuje, a tlo i zgrade ih odbijaju. Signal s mobilnog telefona može do bazne stanice da stigne direktno, ali i nešto kasnije, pošto se odbije od tla ili zgrada. To može da stvori odjek ili da izobliči signal (slabljenje zbog različitih putanja). Ponekad se može čuti i udaljen razgovor koji se na putu odbio više puta.

Osamsto trideset dva kanala sistema AMPS podeljeni su u četiri kategorije:

1. Upravljanje (od bazne stanice ka mobilnim telefonima), za održavanje sistema.
2. Pejdžing (od bazne stanice ka mobilnim telefonima), za obaveštavanje korisnika o pristiglim pozivima.
3. Pristupanje (dvosmerno) za uspostavljanje veze i dodeljivanje kanala.
4. Podaci (dvosmerni), za prenos glasa, faksimila ili podataka.

Za upravljanje je predviđen 21 kanal; oni su programski ugrađeni u PROM svakog telefona. Pošto se iste frekvencije ne mogu koristiti u susednim ćelijama, stvarni broj govornih kanala po ćeliji znatno je manji od 832, najčešće je olco 45.

Uspostavljanje veze

Svaki mobilni telefon u sistemu AMPS ima svoj 32-bitni serijski broj i desetoci- freni telefonski broj, ugrađene u PROM. Telefonski broj se sastoji od trocifrenog pozivnog broja (10 bitova) i pretplatničkog sedmocifrenog broja (24 bita). Kada se telefon uključi, on skenira fiksnu listu s dvadeset jednim upravljačkim kanalom da bi pronašao najjači signal.

Telefon tada difuzno emituje (objavljuje) svoj 32-bitni serijski broj i svoj 34-bitni telefonski broj. Kao i svi upravljački podaci u sistemu AMPS, i ovaj paket se šalje u digitalnom obliku (više puta, zajedno s kodom za ispravljanje grešaka), čak i onda kada su govorni kanali analogni.

Kada bazna stanica primi objavu da je telefon spreman za rad, ona je prosleđuje centrali koja registruje novog korisnika i obaveštava korisnikovu matičnu centralu o njegovoj trenutnoj lokaciji. Tokom normalnog rada, mobilni telefon se registruje približno svakih 15 minuta.

Kada želi da uputi poziv, korisnik uključuje mobilni telefon, unosi s tastature broj potencijalnog sagovornika i pritiska dugme SEND. Telefon tada emituje broj sago- vornika i podatke o svom identitetu preko kanala za pristupanje. Ako na njemu dođe do sukobljavanja, pokušaj emitovanja se malo kasnije ponavlja. Kada bazna stanica dobije zatev, ona o njemu obaveštava centralu. Ako je korisnik petplatnik kompanije koja održava centralu (ili neke od njenih partnera), centrala odmah traži raspoloživ kanal za uspostavljanje veze. Ukoliko ga nađe, šalje povratno njegov broj upravljačkim kanalom. Mobilni telefon se tada automatski prebacuje na dodeljeni govorni kanal i čeka da i sagovornilc uključi telefon.

Dolazni pozivi se obrađuju različito. Počnimo od toga da svi slobodni telefoni neprestano osluškuju pejdžing kanal u nameri da otkriju njima namenjen poziv. Kada se uputi poziv na broj mobilnog telefona (bilo s fiksnog, bilo s mobilnog aparata), paket se šalje centrali njegovog vlasnika da bi se utvrdila lokacija telefona. Paket se zatim šalje baznoj stanici ćelije u kojoj se telefon trenutno nalazi, koja preko pejdžing kanala upućuje opšti poziv, tipa „Cetrnaestice, jesi li tamo?“ Pozvani telefon odgovara prostim „Da“ preko kanala za pristupanje. Baza tada saopštava nešto, kao „Cetrnaestice, imaš poziv na kanalu 3“. U tom trenutku, pozvani telefon se prebacuje na kanal 3 i počinje da zvoni (ili da svira neku melodiju koju je vlasnik dobio za rođendan).

2.6.2 Mobilna **telefonija** druge generacije: digitalni prenos glasa

Prva generacija mobilnih telefona bila je analogna; druga generacija je digitalna. Kao što nije bilo globalno prihvaćenog standarda za prvu generaciju, nije ga bilo ni za drugu. Danas su u upotrebi četiri sistema: D-AMPS, GSM, CDMA i PDC. U nastavku ćemo opisati prva tri. PDC se koristi samo u Japanu i u osnovi je to sistem D-AMPS, dodatno modifikovan da bi bio kompatibilan s prethodnim japanskim analognim sistemom. Naziv **Usluge ličnih komunikacija** (engl. *Personal Communications Services, PCS),* ponekad se - u marketingu - koristi za označavanje (digitalnih) sistema drage generacije. Ono je prvobitno označavalo mobilne telefone koji rade u području 1900 MHz, ali se danas ta razlika uglavnom ne naglašava.

D-AMPS - Digitalni AMPS

Druga generacija sistema AMPS jeste potpuno **digitalan napredni sistem mobilne telefonije** *(engl. Digital Advanced Mobile Phone System, D-AMPS)*. On je opisan Međunarodnim standardom IS-54 i kasnijim standardom IS-136. Sistem D-AMPS je brižljivo projektovan za zajednički život sa sistemom AMPS, tako da telefoni prve i druge generacije istovremeno mogu da se koriste u istoj ćeliji. Sistem D-AMPS koristi iste kanale opsega 30 kHz kao i AMPS, i na istim frekvencijama, tako da analogni i digitalni kanal mogu da budu jedan do drugoga. U zavisnosti od međusobnog odnosa dve vrste telefona u ćeliji, centrala odlučuje o broju i rasporedu analognih, odnosno digitalnih kanala; tip i broj kanala menjaju se dinamički, onako kako se menja odnos i ukupan broj dve vrste telefona.

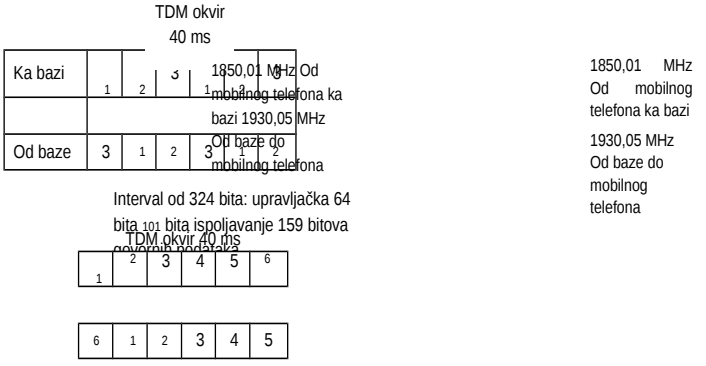
Kada je sistem D-AMPS pušten u rad, zbog očekivanog porasta saobraćaja rezervi- sano je i dodatno frekventno područje. Za saobraćaj od telefona ka baznoj stanici rezer- visano je područje između 1850 i 1910 MHz, a za saobraćaj od bazne stanice ka telefonima područje između 1930 i 1990 MHz, pri čemu su kanali išli u paru, kao u sistemu AMPS. Ove radio-talase od samo 16 cm može da prima standardna četvrttalasna antena dužine samo 4 cm, tako da su telefoni mogli biti manji. Međutim, mnogi telefoni

sistema D-AMPS mogu da koriste oba područja (850 MHz i 1900 MHz), povećavajući tako verovatnoću pronalaženja slobodnog kanala.

Govorni signal koji primi mikrofon mobilnog telefona u sistemu D-AMPS digita- lizuje se i komprimuje po modelu koji je mnogo složeniji od delta modulacije i sistema prediktivnog kodiranja o kojima smo ranije govorili. Pri komprimovanju se detaljno vodi računa o svojstvima ljudskog govornog aparata da bi se standardno PGM kodiranje (56 kb/s) svelo na 8 kb/s uz očuvanje kvaliteta glasa. Komprimovanje se vrši specijalnim kolom, vokoderom (engl. *vocoder)* (Bellamy, 2000), unutar telefona, umesto u baznoj stanici ili centrali, da bi se smanjio broj bitova koji se šalju vazdušnom vezom. U fiksnoj telefoniji takva šema ne predstavlja prednost jer smanjenje saobraćaja na lokalnoj liniji nema nikakvog uticaja na prenosni kapacitet čitavog sistema.

U mobilnoj telefoniji se digitalizovanjem i komprimovanjem u korisničkom aparatu postiže ogromna ušteda, tolika da u sistemu D-AMPS, uz pomoć multipleksiranja podelom vremena, tri korisnika mogu istovremeno da koriste isti par frekvencija. Svaki par frekvencija u sekundi može da prenese 25 okvira (trajanja po 40 ms). Svaki okvir je izdeljen u šest vremenskih intervala (po 6,67 ms), kao što se vidi sa slike 2-42(a) za par najnižih frekvencija.

(a)



(b)

Slika 2-42. (a) Kanal s tri korisnika u sistemu D-AMPS. (b) Kanal sa šest korisnika u sistemu D-AMPS.

Svaki okvir dele tri korisnika koji naizmenično koriste veze ka bazi i od nje. Tokom prvog intervala sa slike 2-42(a), na primer, prvi korisnik može da emituje ka bazi, a treći korisnik da od nje prima podatke. Svaki interval sadrži 324 bita, od kojih se 64 bita koriste za vremensko obezbeđenje kanala, sinhronizovanje i upravljanje, dok je 260 bitova na raspolaganju korisniku. Od njih se 101 bit koristi za ispravljanje grešaka nastalih na bučnoj vazdušnoj vezi, tako da na kraju ostaje samo 159 bitova za lcompri- movane govorne podatke. Uz brzinu od 50 intervala u sekundi, raspoloživa brzina pre- nosa govornih podataka iznosi manje od 8 kb/s - 1/7 standardne PCM propusne moći.

Uz bolje algoritme za komprimovanje, zauzeće opsega govornim podacima može se svesti i na 4 kb/s, kada isti okvir mogu da koriste šest korisnika, kao što je prikazano na slici 2-42(b). Sa gledišta operatera, kada šest korisnika sistema D-AMPS zbi- jete na mesto koje koristi samo jedan korisnik sistema AMPS, to predstavlja ogroman uspeh i uveliko objašnjava popularnost PCS usluga. Naravno, glas prenet brzinom

4 kb/s razlikuje se po kvalitetu od glasa prenetog brzinom 56 kb/s, ali PCS operateri i ne insistiraju na govornom kvalitetu svojih usluga. Treba takođe naglasiti da kanal propusne moći 8 kb/s prenosi podatke lošije i od prastarog modema brzine 9600 b/s.

Upravljačka struktura sistema D-AMPS prilično je složena. Ukratko, grupa od 16 okvira čini superokvir, pri čemu se u svakom superokviru ograničen broj puta pojavljuju izvesni upravljački podaci. Za upravljanje se koriste šest glavnih kanala na- menjenih: konfigurisanju sistema, klasičnom upravljanju i upravljanju u realnom vremenu, pejdžingu, odgovaranju na zahteve za pristupanje i slanju kratkih poruka. Međutim, sistem u osnovi radi kao AMPS. Kada se uključi mobilni telefon, on uspostavlja vezu s baznom stanicom da bi objavio svoje prisustvo, a zatim osluškuje upravljački kanal očekujući pozive. Pošto registruje prisustvo novog aktivnog telefona u sistemu, centrala obaveštava korisnikovu matičnu bazu o njegovoj lokaciji, tako da se pozivi mogu ispravno usmeriti.

Razlika između sistema AMPS i D-AMPS ogleda se u načinu predavanja upravljanja. U sistemu AMPS, to u potpunosti obavlja centrala, bez pomoći mobilnih uređaja. Kao što se vidi sa slike 2-42, mobilni telefon u sistemu D-AMPS trećinu vremena niti emituje, niti prima podatke. Ti neaktivni vremenski intervali koriste se za proveravanje kvaliteta linije. Kada telefon otkrije da se signal gubi, o tome obaveštava centralu koja tada može da prekine vezu, a mobilni telefon može da pokuša da pronađe jači signal neke druge baze. Kao i u sistemu AMPS, i ovde se upravljanje prenosi s jedne bazne stanice na drugu tokom oko 300 ms. Opisana tehnika naziva se **potpomognuto preuzimanje upravljanja** (engl. *Mobile Assisted HandOff, MAHO).*

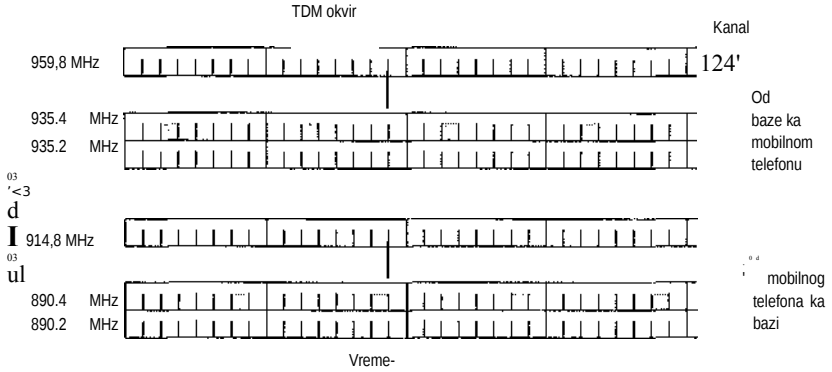
GSM - Globalni sistem mobilnih komunikacija

Sistem D-AMPS široko se koristi u SAD i (ponešto izmenjen) u Japanu. U skoro čitavom ostalom svetu koristi se **Globalni sistem mobilnih komunikacija** (engl. *Global System for Mobile Communications, GSM),* koji je u izvesnoj **meri** prodro i u SAD. Sistem GSM je na prvi pogled sličan sistemu D-AMPS jer su oba zasnovana na ćelijama. U oba sistema se koristi multipleksiranje podelom frekvencija, a svaki mobilni telefon emituje na jednoj frekvenciji, a prima na drugoj, višoj frekvenciji (ta razlika je u sistemu D-AMPS 80 MHz, a u GSM-u 55 MHz). U oba sistema se par frekvencija mul- tipleksira podelom vremena na intervale zajedničke za više korisnika. Međutim, GSM kanali su mnogo širi od kanala u sistemu D-AMPS (200 kHz, u odnosu na 30 kHz) i srazmerno su manje natrpani (8, u odnosu na 3 korisnika), tako da se u sistemu GSM ostvaruje mnogo veći protok podataka po korisniku nego u sistemu D-AMPS.

U nastavku ćemo ukratko objasniti neka od glavnih svojstava sistema GSM. Međutim, standard za GSM zauzima više od 5000 (!) stranica teksta. Veliki njegov deo tiče se inženjerskih aspekata sistema, naročito konstrukcije prijemnika za obradu signala koji se prostire različitim putanjama i sinhronizovanja rada predajnika i prijemnika. Ni o jednom, ni o drugom neće biti reči u nastavku.

Kao što se vidi na slici 2-43, svako frekventno područje ima širinu 200 kHz. Sistem GSM ima 124 para jednosmernih kanala. Svaki jednosmerni kanal širine 200 kHz podržava osam zasebnih veza multipleksiranih podelom vremena. Svakoj aktivnoj

stanici dodeljuje se jedan vremenski interval u paru kanala. U svakoj ćeliji se teorijski može podržati 992 kanala, ali mnogi od njih su zabranjeni da bi se izbegao sukobljavanje između susednih delija. Na slici 2-43, svih osam zasenčenih vremenskih intervala pripadaju istoj vezi, četiri su predviđena za jedan, a četiri za drugi smer. Emitovanje i prijem ne odvijaju se unutar istog vremenskog intervala jer GSM radio ne može istovremeno da emituje i prima poruke, a treba vremena da se prebaci s jednog režima u drugi. Kada mobilni telefon kome su dodeljene frekvencije 890,4 i 93.5,4 MHz i vremenski interval 2 želi da pošalje poruku baznoj stanici, on koristi četiri donja (na slici) zasenčena vremenska intervala (i intervale koji ih vremenski slede), smešta- juci podatke u svaki od njih sve dok svi ne budu poslati.



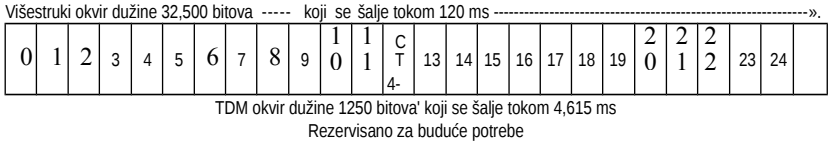
Slika 2-43. Sistem GSM koristi 124 frekventna kanala, svaki sa osam vremenskih intervala dobijenih TDM modulacijom.

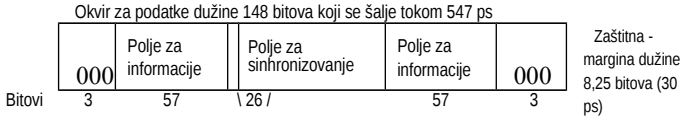
Vremenski TDM intervali, prikazani na slici 2-43, deo su složene hijerarhije okvira. Svaki TDM interval ima specifičnu strukturu, a grupe TDM intervala obrazuju višestruke okvire (engl. *multiframes),* takođe specifične strukture. Na slici 2-44 upro- šćeno je prikazana pomenuta hijerarhija okvira. Vidimo da se svaki TDM interval sastoji od 148-bitnog okvira za podatke koji zauzima kanal tokom 577 ps (uključujući i zaštitnu marginu svakog intervala od 30 ps). Svaki okvir za podatke počinje i završava se trostrukim bitom 0 zbog boljeg razgraničenja pojedinih okvira. Okvir sadrži dva polja za *informacije* (svako dužine 57

bitova), a svakom od njih pridružen je i upravljački bit koji ukazuje da li je sledeće polje za *informacije* namenjeno za govor ili za podatke. Između polja za informacije smešteno je polje za *sinhronizovanje* (uhodavanje) dužine 26 bitova, koje koristi prijemnik da bi uhvatio korak s granicama okvira koje šalje predajnik.

Prenos okvira s podacima traje samo 547 ps, ali predajnik emituje takav okvir tek svakih 4,615 ms zato što isti kanal deli sa sedam drugih predajnika. Ukupna brzina prenosa svakog kanala je 270.833 b/s, što kada se podeli na osam korisnika iznosi 33,854 kb/s - dvaput više od brzine u sistemu AMPS (324 bita, 50 puta u sekundi = 16,2 kb/s). Međutim, u oba sistema, sistemski podaci odnesu lavovski deo propusnog opsega, ostavljajući 24,7 kb/s za podatke svakog korisnika pre nego što se isprave

greške. Nakon ispravljanja grešaka, za govorne podatke preostaje 13 leb/s, što obezbeđuje znatno bolji kvalitet glasa od sistema D-AMPS (po cenu korišćenja srazmerno vec'eg propusnog opsega).





Slika 2-44. Deo hijerarhijske strukture TDM okvira.

Kao što se vidi sa slike 2-44, osam okvira s podacima čine jedan TDM okvir, a 26 TDM okvira čine višestruki okvir trajanja 120 ms. Od 26 TDM okvira unutar višestrukog okvira, interval 12 se koristi za upravljanje, a interval 26 je rezervisan za buduće potrebe, tako da preostaje samo 24 okvira za korisnički saobraćaj.

Međutim, osim višestrukog okvira sa 26 vremenskih intervala koji je prikazan na slici 2-44, koristi se i (neprikazan) višestruki okvir sa 51. intervalom. Neke od tih intervala koristi više upravljačkih kanala pomoću kojih se održava sistem. Kroz **kanal za upravljanje neusmerenim emitovanjem** (engl. *broadcast control channel)* neprestano teku podaci o identitetu bazne stanice i statusu kanala. Svi mobilni telefoni neprestano prate jačinu tog signala i tako znaju kada su ušli u drugu ćeliju.

Namenski upravljački kanal (engl. *dedicated control channel)* služi za ažuriranje podataka o lokaciji, registrovanje i uspostavljanje veze. Svaka bazna stanica održava bazu podataka o mobilnim telefonima kojima trenutno upravlja, a podaci za njeno održavanje stižu joj preko namenskog upravljačkog kanala.

Poslednjije **opšti upravljački kanal** (engl. *common control channel),* koji je logički podeljen na tri potkanala. Prvi potkanal, **kanal za objavljivanje** (engl. *paging sub- channel),* koristi bazna stanica za obaveštavanje o pristiglim pozivima. Zato ga mobilni telefoni stalno osluškuju. Drugi potkanal, **kanal za slobodan pristup** (engl. *random access channel)* omogućava korisnicima da zahtevaju vremensld interval na namenskom kontrolnom kanalu. Ako se dva zahteva sukobe, napravi se zbrka i zato se moraju kasnije ponovo uputiti. Koristeći vremenski interval na namenskom kontrolnom kanalu, telefon može da uspostavi vezu. Dodeljeni vremenski interval objavljuje se na trećem potkanalu, **kanalu za dodelu pristupa** (engl. *access grant channel).*

CDMA - kodirani višestruki pristup

Sistemi D-AMPS i GMS prilično su konvencionalni. U njima se tehnikama FDM i TDM frekventni spektar deli na kanale, a ovi na vremenske intervale. Međutim, postoji i treća varijanta, **kodirani višestruki pristup** (engl. *Code Division Multiple Access, CDMA),* koji radi sasvim drugačije. Kada je sistem CDMA prvi put predložen, industrija je reagovala na način na koji je reagovala kraljica Izabela kada se Kolumbo ponudio da doplovi do Indije jedreći u pogrešnom smeru. Međutim, štedro podržan od kompanije Qualcomm, sistem CDMA ne samo što je prihvaćen, već se i nametnuo kao najbolje tehnološko rešenje i osnova za mobilne telefonske sisteme treće generacije. U SAD se široko koristi i u mobilnim sistemima druge generacije, konkurišući ravnopravno sistemu D-AMPS. Na primer, Sprint za PCS koristi sistem CDMA, dok AT&T Wireless koristi D-AMPS. Za sistem CDMA razvijen je Međunarodni standard IS-95, pa se on ponekada tako i naziva. U upotrebi je i njegovo komercijalno ime **cdmaOne.**

CDMA radi potpuno drugačije od sistema AMPS, D-AMPS i GSM. Umesto da dodeljeno frekventno područje deli na više stotina uskih kanala, CDMA svakoj stanici dozvoljava da sve vreme emituje u ćelom frekventnom području, a njihove jednovremene emisije razdvaja drugačijim kodiranjem. Sukobljeni okviri se u sistemu CDMA ne smatraju neupotrebljivim, već se pretpostavlja da se njihovi signali linearno kombinuju.

Pre nego što objasnimo algoritam kodiranja, razmotrimo jednu analogiju: aero- dromski hol s mnogo putnika koji međusobno razgovaraju u parovima. Sistem TDM liči na situaciju u kojoj se svi putnici nalaze usred hola, ali razgovaraju tek kada na njih dođe red. Sistem FDM odgovara situaciji kada su parovi razbijeni po separeima i u svakom se istovremeno vodi drugi razgovor, nezavisno od ostalih razgovora. Sistem CDMA, pak, odgovara situaciji kad se svi putnici nalaze usred hola i svi istovremeno razgovaraju, ali svaki par na drugom jeziku. Par Francuza obraća pažnju samo na francuski jezik, zanemarujući sve drago. Slično tome, u sistemu CDMA ključno je izdvojiti samo željeni signal i istovremeno sve drago odbaciti kao „pozadinski šum". Sledi nešto uprošćen opis sistema CDMA.

U sistemu CDMA, trajanje svakog bita podeljeno je na *m* kratkih **podintervala** (engl. *chips).* Takvih podintervala obično ima 64 ili 128 po bitu, ali ćemo u pojednostavljenom primeru koji sledi smatrati daje bit podeljen na samo 8 podintervala.

Svakoj stanici se dodeljuje jedinstven m-bitni kod, tzv. **sekvenca podintervala** (engl. *chip sequence).* Kada želi da pošalje bit 1, stanica šalje svoju sekvencu podintervala. Za bit 0, ona šalje komplement sekvence podintervala. Ništa drugo nije dozvoljeno. Na taj način, zam=8, ako je sekvenca podintervala stanice 00011011, ona bit 1 šalje tako što šalje 00011011, a bit 0 šaljući 11100100.

Povećanje količine poslatih podataka sa *b* bita u sekundi na *mb* podintervala u sekundi, može se izvesti samo ako se propusni opseg poveća *m* puta, zbog čega se CDMA smata vrstom komunikacije koja radi uz proširivanje spektra (pod uslovom da način modulacije i kodiranja ostaju isti). Ako je frekventno područje širine 1 MHz na- menjeno za 100 stanica, onda uz tehniku FDM svaka dobija opseg od 10 kHz i može

da prenosi podatke brzinom 10 kb/s (uz 1 bit po hercu). U sistemu CDMA, svaka stanica koristi pun frekventni opseg od 1 MHz, tako da brzina prenosa iznosi 1 megapoď- interval u sekundi. Uz manje od 100 podintervala po bitu, efektivni propusni opseg po stanici veći je uz CDMA nego uz FDM, a i otpada problem dodeljivanja kanala.

Iz pedagoških razloga zgodnije je koristiti bipolarno označavanje (engl. *bipolar notation),* tj. predstavljanje binarne nule sa -1, a binarne jedinice sa +1. Sekvence pod-intervala ćemo prikazivati u zagradi, tako da bit 1 za stanicu A sada postaje (-1-1-1+1 +1 -1 +1 +1). Na slici 2-45(a) prikazujemo binarne sekvence podintervala za četiri stanice koje smo izabrali za primer. Na slici 2-45(b) iste sekvence prikazujemo bipolarno.

A: 0 0 0 1 1 0 1
B:001 01 1 1 0
C: 0 1 0 1 1 1 0 0
D: 0 1 0 000 1 0
(a)

A (-1 -1 -1 +1 +1 -1 +1 +1)
B (-1 -1 +1 -1 +1 +1 +1 -1)
C (-1 +1 -1 +1 +1 +1 -1 -1)
D (-1 +1 -1 -1 -1 -1 +1 -1)
(b)

Šest primera:

— 1 -	C	S_1 =(-1 +1 -1 +1 +1 +1 -1 -1)
-11-	B + C	S_2 = (-2 0 0 0+2+2 0-2)
1 0--	A + B	S_3 = (0 0-2 +2 0-2 0+2)
1 0 1 —	A + B + C	S_4 = (—1 +1 -3+3+1 -1 -1 +1)
1 1 1 1	A + B + C + D	S_5 = (-4 0-2 0 + 2 0+2-2)
110 1	A+B+Č+D	S_6 = (-2-2 0-2 0-2 +4 0)

(c)

51 C = (1+1+1+1+1+1+1+1)/8 = 1
52 c = (2 +0 +0 +0 +2 +2 +0 **+2)/8** = 1
53 C = (0 +0 +2 +2 +0 -2+0 -2)/8 = 0
54 C = (1 +1 +3+3+1 -1 +1 -1)/8 = 1
55 C= (4+0+2+0+2+0-2+2)/8 = 1
56 C= (2 —2 +0 -2 +0 -2 -4 +0)/8 = —1

(d)

Slika 2-45. (a) Binarne sekvence podintervala za četiri stanice, (b) Bipolarno označavanje sekvenci intervala, (c) Šest primera prenosa podataka, (d) Dekodiranje signala stanice C.

Svaka stanica ima sopstvenu, jedinstvenu sekvencu podintervala. Označimo simbolom S m-dimenzioni vektor za stanicu *S,* a simbolom Š njegovu negaciju. Sve sekvence podintervala predstavljaju međusobno **ortogonalne** parove, što znači da je vrednost normalizovanog skalarnog proizvoda bilo koje dve različite sekvence podintervala, S i T (pisan kao S»T) jednaka nuli. Takve sekvence mogu se generisati metodom **Volšovog koda** (engl. *Walsh codes).* Ortogonalnost sekvenci podintervala može se matematički opisati sledećim izrazom:

$$S \cdot T = -\sum_{i=i}^{1} S_i T_i = 0$$

Jednostavno rečeno, ima isto toliko istih parova koliko i različitih. Ovo svojstvo ortogonalnosti pokazaće se kasnije kao ključno. Kada je S • **T** = 0, onda je i S • **T** = 0. Normalizovani skalarni proizvod bilo koje sekvence podintervala sa samom sobom jednak je 1:

$$S \bullet S = \frac{1}{m} \sum_{i=1}^{m} S_i T_i = \frac{1}{m} \sum_{i=1}^{m} (\pm 1)^2 = 1$$

To sledi iz činjenicedaje svakiod *m* članovazbira u skalarnom proizvodu jednak 1, tako daje zbir jednak m.'Zapazite i to daje S ® Š = -1.

Tokom trajanja svakog bita, stanica može da emituje 1 šaljući svoju sekvencu pod-intervala, da emituje 0 šaljući komplement sekvence ili da ne emituje ništa. Pretpo-stavicemo da sve stanice rade sinhrono, tako da se sve sekvence podintervala emituju u istom trenutku.

Kada dve ili više stanica emituju istovremeno, njihovi bipolarni signali sabiraju se linearno. Na primer, ako u istom vremenskom podintervalu tri stanice emituju +1, a jedna stanica -1, rezultat je +2. O signalima se može razmišljati kao o naponima: tri napona od +1 V i jedan napon od -1 V daju napon od +2 V.

Na slici 2-45(c) imamo šest primera u kojima emituje jedna stanica ili više stanica istovremeno. U prvom primeru, *C* šalje bit 1, u stvari svoju sekvencu podintervala. U drugom primeru, i *B* i C šalju bit 1, tako da je zbir njihovih bipolarnih sekvenci:

(-1 -1 +1 -1 +1 +1 +1 -1) + (-1 -1 +1 -1 +1 +1 +1 -1) = (-2000+2+20 -2)

U trećem primeru, stanica A šalje bit 1, a stanica *B* bit 0. Ostale stanice miruju. U četvrtom primeru, *A* i C šalju bit 1, dok *B* šalje bit 0. U petom primera, sve četiri stanice šalju bit 1. Najzad, u poslednjem primera, *A, B i D* šalju bit 1, dok *C* šalje bit 0. Imajte na umu da se svaka od šest sekvenci 5; do *Sg* prikazanih na slici 2-45(c) emituje tokom trajanja jednog bita.

Da bi mogao da dekodira tok bitova određene stanice, prijemnik mora unapred znati njenu sekvencu podintervala. On dekodira signal tako što izračunava normalizovani skalarni proizvod primljene sekvence (linearnog zbira sekvenci svih aktivnih stanica) i sekvence podintervala stanice čiji tok bitova dekodira. Ako je primljena sekvenca S, a prijemnik pokušava da izdvoji samo emisiju stanice čija je sekvenca C, on samo izračunava normalizovani skalarni proizvod S ° C.

Da biste se uverili da ovo radi, zamislite da svaka od dve stanice, *A* i C, emituje bit 1 u isto vreme kada stanica *B* emituje bit 0. Prijemnik vidi zbir, S = A + **B** + C, i računa:

$$S \circledR C = (A + \mathbf{B} + C) \bullet C = A \circledR C + B \circledR C + C \circledR C = 0 + 0 + 1 = 1$$

Prva dva sabirka iščezavaju zato što je pažljivo izabrano da svi parovi sekvenci intervala budu ortogonalni, prema jedn. (2-4). Sada je jasno zašto uparene sekvence intervala moraju biti ortogonalne.

Opisanu situaciju možemo alternativno zamisliti i kao da je svaka sekvenca podintervala stigla u prijemnik pojedinačno. Tada bi prijemnik za svaku posebno izračunavao skalarni proizvod, a zatim sabrao rezultate. Zbog ortogonalnosti, svi skalarni proizvodi, osim proizvoda C • C, bili bi 0. Izračunavanje skalarnog proizvoda i sabiranje rezultata operacije su čiji redosled u ovom slučaju nije bitan.

Da bismo bolje razumeli postupak dekodiranja, razmotrimo ponovo šest primera sa slike 2-45(c), sada ilustrovanih na slici 2-45(d). Pretpostavimo da prijemnik želi da izdvoji bit koji šalje stanica C iz svakog od šest zbirova Sj do S_6. On će ga izračunati tako što će sabrati odgovarajuće proizvode vektora S i C sa slike 2-45(b) i potom rezultat podeliti sa 8 (pošto smo usvojili daje $m = 8$). Kao što vidimo, dobija se ispravna vrednost svakog bita. To je kao da govorite francuski, a za sve ostalo ste gluvi.

Idealan, bešuman CDMA sistem ima neograničen kapacitet (tj. može da podrži neograničen broj stanica), na isti način kao što bešumni Nikvistov kanal može da se proširuje povećavajući neprestano broj bitova u svakom uzorku. Praktični kapacitet sistema je, međutim, znatno manji, zbog fizičkih ograničenja. Prvo, pretpostavili smo da su svi podintervali vremenski sinhronizovani, što u stvarnosti nije moguće. Predajnik i prijemnik mogu se sinhronizovati tako što predajnik emituje unapred definisanu sek- vencu podintervala koja je dovoljno dugačka da bi se prijemnik za nju „zakačio“. Posle toga se svaka druga (nesinhronizovana) emisija smatra šumom. Ako takvih emisija nema previše, osnovni algoritam za dekodiranje i dalje će raditi prilično uspešno. Su- perponiranje sekvenci podintervala i šuma teorijski je temeljno obrađeno (Pickholtz i sar., 1982). Kao što biste i očekivali, što je duža sekvenca, veća je verovatnoća da se ispravno prepozna u prisustvu šuma. Veća pouzdanost se može postići korišćenjem koda za ispravljanje grešaka, koji sekvence podintervala inače nikada ne koriste.

U našem razmatranju se podrazumevalo da prijemnik prima signal iste jačine od svih stanica. CDMA se najčešće koristi za bežične sisteme s fiksnom baznom stanicom i brojnim mobilnim stanicama na različitom rastojanju od nje. Jačina signala koju prima bazna stanica zavisi od udaljenosti stanice. Pametan predlog bi bio da svaka mobilna stanica ka bazi emituje signal čija je jačina obrnuto proporcionalna jačini signala koji prima od baze. Drugim recima, mobilna stanica koja prima slab signal trebalo bi da emituje većom snagom od one koja prima jak signal. Bazna stanica može da bude ta koja će izričito naređivati mobilnim stanicama da emituju jači ili slabiji signal.

Pretpostavili smo i to da prijemnik zna koje pošiljalac. U načelu, uz dovoljan računarski kapacitet, prijemnik može istovremeno da osluškuje sve predajnike i da uporedo na svaki primenjuje algoritam za dekodiranje. To je lakše reći, nego stvarno i uraditi. CDMA ima i mnoge druge komplikacije koje smo u ovom kratkom opisu zanemarili. Pa ipak, CDMA je promućurno smišljen sistem koji nezadrživo prodire u bežične mobilne komunikacije. On normalno radi u području širine 1,25 MHz (u odnosu na širinu od 30 kHz u sistemu D-AMPS i 200 kHz u sistemu GSM), ali u njemu podržava mnogo više korisnika nego ijedan od navedena dva sistema. Propusni opseg koji nudi svakom korisniku u praksi je barem jednak onom koji nudi GSM, a često i mnogo širi.

One koji žele da se upoznaju sa detaljima sistema CDMA upućujemo na Leeja i Millera (1998). Alternativan sistem proširivanja vremena, umesto frekvencija, opisali su Crespo i saradnici (1995). Još jedan sistem nude Sari i suradnici (2000). Za razu- mevanje navedenih referenci potrebno je izvesno predznanje iz projektovanja komunikacionih sistema.

2.6.3 Mobilna telefonija treće generacije: digitalni prenos glasa i podataka

Pokušajmo da sagledamo budućnost mobilne telefonije. Industrijski razvoj je pod uticajem mnogih činilaca. Najpre, razmena podataka već je veća od govornog saobraćaja na fiksnoj mreži i raste eksponencijalno, dok govorni saobraćaj stagnira. Mnogi stručnjaci očekuju da i na mobilnim uređajima uskoro razmena podataka premaši govorni saobraćaj. Zatim telefonija, industrija zabave i računarska industrija danas rade gotovo isključivo s digitalnim podacima i njihova tehnologija se brzo ujednačava. Mnogi halapljivo očekuju lake, prenosive uređaje koji će istovremeno služiti kao telefon, CD plejer, DVD plejer, terminal e-pošte, Web portal, igrica, program za obradu teksta i štošta drugo, i pri tom biti globalno umreženi na Internet brzom bežičnom vezom. Takvim uređajima i načinima njihovog povezivanja bavi se treća generacija mobilne telefonije. Više detalja potražite kod Hubera i saradnika (2000), i Sarikaye (2000).

Organizacija ITU je još 1992. godine želela da malo konkretizuje ovaj san i napravila je strategijski plan **IMT-2000** (IMT je skraćeno od **International Mobile Telecommunications - međunarodne mobilne komunikacije).** Broj 2000 imao je trostruko značenje: (1) godinu kada je usluga trebalo da proradi, (2) frekvenciju na kojoj je usluga trebalo da radi (u MHz) i (3) propusni opseg usluge (u kHz).

Nijedan od planova nije se ostvario. Ništa nije realizovano 2000. godine. ITU je preporučila svim državama da rezervišu područje frekvencije oko 2 GHz da bi uređaji nesmetano radili bez obzira na državne granice. To frekventno područje rezervisala je samo Kina. Najzad, ustanovljeno je da brzina prenosa od 2 Mb/s trenutno nije ostvariva za korisnike koji *previše* brzo menjaju mesto (zbog teškoća brzog preuzimanja upravljanja). Realističnije je da brzinu 2 Mb/s ostvari stacionarni (kućni) korisnik (tada se može konkurisati ADSL liniji), brzinu 384 kb/s korisnik koji se kreće pešice, a 144 kb/s korisnik u automobilu. Bez obzira na to, u čitavoj **3G** oblasti (3G - treća generacija) ključa kao u kotlu. Treća generacija možda neće biti ono što se očekuje i možda će malo zakasniti, ali će se sigurno pojaviti.

Mreža IMT-2000 treba svojim korisnicima da obezbedi sledeće usluge:

1. Visokokvalitetan prenos glasa.
2. Razmenjivanje poruka (umesto e-pošte, faksa, SMS poruka, ćaskanja itd.).
3. Multimediju (slušanje muzike, gledanje videa, filmova, televizije itd.).
4. Pristup Internetu (krstarenje Webom, uključujući prijem strana sa zvukom i videom).

Dopunske usluge mogle bi biti video-konferencije, prisustvo na daljinu, grupne ig- rice i m-trgovina (plaćanje u prodavnici pomoću mobilnog telefona). Štaviše, očekuje se da sve ove usluge budu globalno raspoložive (uz automatsko povezivanje preko satelita kada nije dostupna nijedna zemaljska stanica), trenutne (uvek uključene) i garant ovano kvalitetne.

ITU je za IMT-2000 zamislila jedinstvenu tehnologiju, tako da bi proizvođači pravili jedinstven uređaj koji bi se prodavao i mogao koristiti bilo gde na planeti (slično računarima i CD plejerima, za razliku od današnjih mobilnih telefona i televizora). Jedinstvena tehnologija bi olakšala život mrežnim operaterima i ohrabrila mnoge potencijalne korisnike da se pretplate na ove usluge. Rat formata, kao između Betamaxa i VHS-a u trenutku pojave videorikordera, nije dobar za poslove.

Pojavilo se više predloga i posle izvesne „trijaže", preostala su samo dva. Ericsson je dao prvi predlog: **širokopojasni** CDMA (engl. *Wideband CDMA, W-CDMA)*. Sistem koristi direktno sekvencijalno širenje spektra o kome smo već govorili, radi s propusnim opegom 5 MHz i projektovan je za povezivanje sa GSM mrežama iako s njima nije kompatibilan. On, međutim, omogućava da korisnik istupi iz W-CDMA ćelije i uđe u GSM deliju, a da se veza pri tome ne prekine. Evropska unija je snažno podržala Ericssonov predlog i nazvala ga **Univerzalni sistem mobilnih telekomunikacija** (engl. *Universal Mobile Telecommunication System, UMTS)*.

Drugi je bio predlog Qualcomma: CDMA2000. I za taj sistem je iskorišćeno direktno sekvencijalno širenje spektra, u načelu zasnovano na standardu IS-95 i kompatibilno s njim. Propusni opseg sistema talcođe je 5 MHz, ali sistem nije predviđen za povezivanje sa GSM mrežom i ne može da prepušta veze GSM delijama (ili, kada smo vec kod toga, D-AMPS delijama). Druge razlike u odnosu na sistem W-CDMA obuhvataju drugačiju brzinu prenosa vremenskih podintervala, drugačije trajanje okvira, drugačiji spektar i drugi način sinhronizovanja.

Da su okupili inženjere Ericssona i Qualcomma i naredili im da naprave zajednički projekat, oni bi to verovatno i učinili. Na kraju krajeva, osnovni princip oba sistema je CDMA u kanalu širine 5 MHz, a niko nije insistirao na brzini prenosa. Kao i obično, problem nije bio tehničke prirode, već politički. Evropa je želela sistem koji bi se mogao povezati sa GSM mrežama; SAD su želele sistem kompatibilan sa sistemom koji se tamo vec široko koristi (IS-95). Svaka strana je takođe gurala svoju lokalnu kompaniju (Ericsson se nalazi u Švedskoj, Qualcomm u Kaliforniji). Ericsson i Qualcomm su na kraju završili na brojnim ročištima, svaki štiteći svoje CDMA patente.

Marta 1999. sudski procesi su završeni time što se Ericsson složio da kupi Qualcommovu infrastrukturu. Kompanije su se dogovorile i oko jedinstvenog 3G standarda koji je, međutim, imao mnogo nekompatibilnih opcija - papirno pokriće brojnih tehničkih razlika. Bez obzira na sve to, ipak očekujemo da se narednih godina pojave 3G uređaji i usluge.

Mnogo je napisano o 3G sistemima, uglavnom slavopojke (Collins i Smith, 2001; De Vriendt i sar., 2002; Harte i sar., 2002; Sarikaya, 2000). Međutim, neki „disidenti" smatraju daje industrija krenula pogrešnim pravcem (Garber, 2002; Goodman, 2000).

Čekajući da se slegne prašina oko 3G sistema, neki operateri su i sami oprezno krenuli u tu tehnologiju stvarajući nešto što je nazvano 2.5G, mada bi tome više odgovarala oznaka 2.1G. Jedno takvo rešenje je **sistem ubrzanog prenosa podataka za** GSM (engl. *Enhanced Data rates for GSM Evolution, EDGE),* što nije ništa drugo do GSM uz veću brzinu prenosa. Više prenetih bitova po bodu znači, nažalost, i više grešaka po bodu, tako da EDGE ima devet šema za modulisanje i ispravljanje grešaka koje se međusobno razlikuju po tome koliko je propusnog opsega rezervisano za ispravljanje grešaka uvedenih većom brzinom

prenosa.

Drugi 2.5G sistem je **opšta paketna radio usluga** (engl. *General Packet Radio Service, GPRS)* - paketna mrežna nadgradnja sistema D-AMPS ili GSM. Ona omogućava mobilnim stanicama da razmenjuju IP pakete u ćeliji koja podržava govorni sistem. Kada je sistem GPRS aktivan, određeni vremenski intervali na određenim frekvencijama rezervišu se za saobraćaj paketa. Broj i lokacije vremenskih intervala dinamički re- guliše bazna stanica, u zavisnosti od odnosa govornog i paketnog saobraćaja u ćeliji.

Raspoloživi vremenski intervali dele se u više logičkih kanala koji služe za različite svrhe. Bazna stanica raspoređuje logičke kanale po vremenskim intervalima. Jedan logički kanal služi za prenos paketa od baze do mobilnih stanica, a svaki paket nosi adresu svog odredišta. Kada želi da pošalje IP paket, mobilna stanica šalje bazi zahtev tražeći jedan ili više vremenskih intervala. Ako zahtev stigne u ispravnom stanju, bazna stanica objavljuje frekvenciju i vremenske intervale lcoje dodeljuje mobilnoj stanici radi slanja paketa. Kada paket stigne u bazu, on se prosleđuje na Internet kablovskom vezom. Pošto GPRS predstavlja samo nadgradnju postojećeg sistema za prenos govora, on je i najbolje prelazno rešenje do pojave 3G sistema.

Iako 3G mreže još uvek nisu u potpunosti razvijene, neki istraživači smatraju daje to svršena stvar i okreću se novim - 4G sistemima (Berezdivin i sar., 2002; Guo i Chaskar, 2002; Huang i Zhuang, 2002; Kellerer i sar., 2002; Misra i sar., 2002). Među pretpostavljenim osobinama 4G sistema su i veliki propusni opseg, sveprisutnost (mogućnost povezivanja bilo gde u svetu), glatko povezivanje s kablovskim (i naročito IP) mrežama, dinamička prilagodljivost resursa i frekventnog spektra, softverski radio i visokokvalitetne multimedijske usluge.

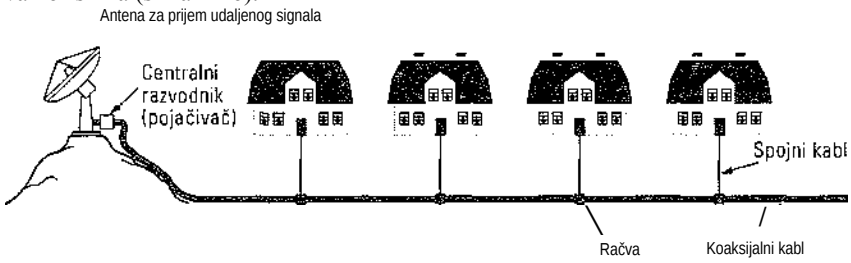
S druge strane, danas je instalirano toliko pristupnih tačaka za lokalne bežične 802.11 mreže da neki smatraju da je 3G ne samo svršena stvar, već i stvar osuđena na propast. Shodno takvom mišljenju, korisnici bežičnih mreža samo će prelaziti s jedne pristupne tačke na drugu i tako sve vreme biti na vezi. Kada bismo rekli da se danas industrija nalazi u stanju snažnog previranja, to bi bio bled opis situacije. Proverite za nekoliko godina šta će od svega toga ispasti.

2.7 KABLOVSKA TELEVIZIJA

Proučili smo prilično detaljno i fiksni i bežični sistem telefonije. Oba sistema će nesumnjivo igrati značajnu ulogu u budućim mrežama. Međutim, među fiksnim mrežama pojavio se nov igrač: kablovska televizija. Mnogi već preko te mreže koriste telefon i usluge Interneta, a operateri kablovske televizije svakog dana nude sve novije i novije usluge. U narednim odeljcima detaljnije ćemo razmotriti kablovsku televiziju kao sistem za umrežavanje i uporediti je s telefonskim sistemima s kojima smo se upravo upoznali. Više podataka o kablovskoj televiziji možete da nađete kod Laubacha i saradnika (2001), Louisa (2002), Ovadie (2001) i Smitha (2002).

2.7.1 TV sa zajedničkom antenom

Kablovska televizija je razvijena krajem četrdesetih godina kao način da se poboljša prijem TV signala za korisnike u seoskim i brdovitim područjima. Sistem se prvobitno sastojao od velike antene za prijem signala postavljene na vrh brda, pojači- vača zvanog centralni razvodnile (engl. *head end*), da ga pojača, i koaksijalnog kabla da ga sprovede do domova korisnika (slika 2-46).



Slika 2-46. Prvobitni sistem kablovske televizije.

Tako je nastala televizija sa zajedničkom antenom (engl. *Community Antenna Télévision).* Bio je to prilično amaterski posao; svako ko se pomalo razumevao u elektroniku mogao je da instalira sistem za svoje naselje (selo, grad), prikupljajući od korisnika novac da bi pokrio troškove. Kako je rastao broj korisnika, postojeći kabl je ojačavan dodatnim kablovima, a pojačivači su dodavani po potrebi. Prenos je bio jed- nosmeran, od centralnog razvodnika ka korisnicima. Sedamdesetih godina postojalo je već na hiljade takvih sistema.

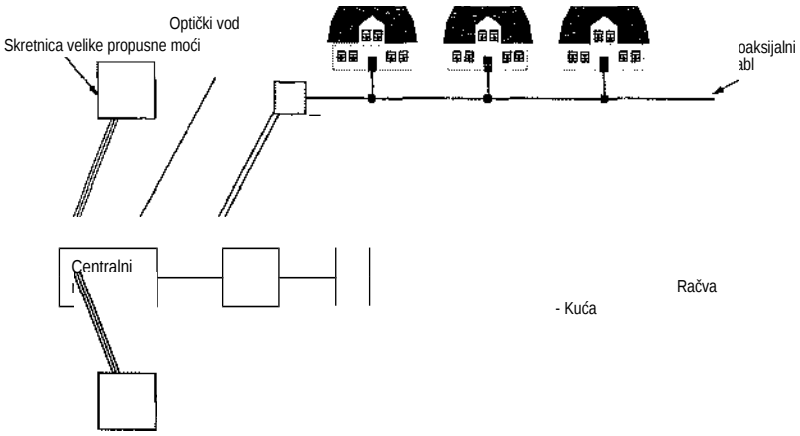
Korporacija Time je 1974. godine pokrenula nov zabavni kanal, Home Box Office, namenjen prikazivanju filmova, i distribuirala ga isključivo kablom. Zatim su se pojavili kablovski kanali koji su emitovali samo vesti ili samo sportske događaje, samo kuvanje i mnoge druge teme. Takav sled događaja izazvao je u industriji dve prome- ne. Prvo, velike korporacije su počele da kupuju postojeće kablovske sisteme i da postavljaju nove kablove da bi prikupile više mušterija. Drago, pojavila se potreba za povezivanjem više kablovskih sistema koji su se često nalazili u različitim gradovima, da bi se preko svih emitovao isti program. Kablovske kompanije su počele da postavljaju kablove između gradova da bi sve mreže povezale u jedinstven sistem. U dlaku se ponavljalo ono što se 80 godina ranije dešavalo u oblasti telefonije: povezivanje izolovanih lokalnih centrala da bi se omogućili međugradski razgovori.

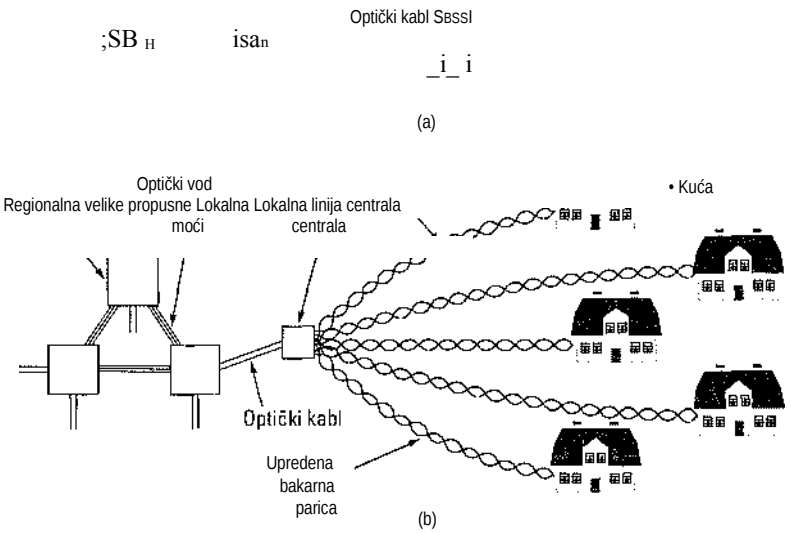
2.7.2 Kablovski Internet

Kablovski sistem je tokom godina rastao i kablovi između gradova su zamenjivani optičkim vlaknima visokog propusnog opsega, baš kao što se dešavalo i u oblasti telefonije. Sistem s međugradskim vezama izvedenim pomoću optičkih kablova i s koaksijalnim kablovima za lokalno razvođenje, nazvan je hibridni optičko-koaksijalni

sistem (engl. *Hybrid Fiber Coax, HFC).* Na spojevima koaksijalnog i optičkog kabla bili su elektrooptički pretvarači, zvani **optički čvorovi** (engl. *fiber nodes).* Pošto je propusni opseg optičkog kabla daleko veći od propusnog opsega koaksijalnog kabla, optički čvor je mogao da opslu ži brojne koaksijalne kablove. Deo savremenog HFC sistema prikazanje na slici 2-47(a).

Mnogi operateri kablovskih mreža poslednjih godina su shvatili da korisnicima mogu da ponude i pristup Internetu, a takođe i telefonske usluge. Tehničke razlike između sistema kablovske televizije i sistema telefonije odredile su šta za to mora da se uradi. Pomenimo samo to da bi svi jednosmerni pojačivači morali da se zamene dvosmernim pojačivačima.





Slika 2-47. a) Kablovska televizija, (b) Sistem fiksne telefonije.

Međutim, postoji još jedna razlika između sistema HFC sa slike 2-47(a) i telefonskog sistema sa slike 2-47(b) koju je mnogo teže otkloniti. U lokalnom korisničkom okruženju isti kabl deli mnogo domaćinstava, dok u telefonskom sistemu svaka kuca ima svoju privatnu lokalnu liniju. Kada se koristi za TV prenos, ta razlika ne igra nikakvu ulogu. Svi programi se prenose kablom i nije važno da li program gleda 10

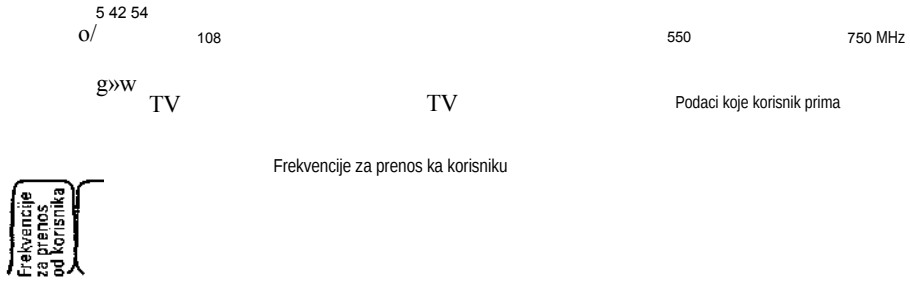
ili 10.000 korisnika. Međutim, kada se isti kabl upotrebi za Internet, tada je ključno pitanje da li ga koristi 10 ili 10.000 osoba. Ako jedan korisnik odluči da preuzme neku vrlo dugačku datoteku, propusni opseg će praktično biti uskraćen ostalim korisnicima. Što je više korisnika, biće i veće njihovo otimanje za propusni opseg. U telefonskom sistemu takvo nešto ne postoji: kada preuzimate veliku datoteku preko ADSL linije, time ne ograničavate propusni opseg svog suseda. S druge strane, propusni opseg koaksijalnog kabla mnogo je veći od propusnog opsega upredene parice.

Kablovske kompanije su problem rešile tako što su dugačak kabl izdelile na segmente i svaki segment direktno spojile sa optičkim čvorom. Propusna moć veze između centralnog koncentratom i svakog optičkog čvora praktično je neograničena, tako da se saobraćaj može neometano odvijati sve dok nema previše pretplatnika na pojedinačnim segmentima kabla. Jedan kabl danas opslužuje između 200 i 2000 domova, ali se sve više korisnika pretplaćuje i na kablovski Internet tako da će u budućnosti verovatno biti potrebno ugrađivanje dodatnih optičkih čvorova.

2.7.3 Dodeljivanje frekvencija

Kada bi potpuno prestale da emituju TV program i postojeću mrežnu infrastrukturu iskoristile isključivo za pristup Internetu, kablovske kompanije bi verovatno izazvale bes potrošača, tako da se one zasad ne usuđuju na takav korak. Osim toga, u većini gradova se strogo lcontroliše šta se nudi preko kablovske mreže, pa kablovske kompanije i kada bi htele, to ne bi mogle da urade. Upravo zato su bile prinuđene da pronađu način za „miroljubivo koegzistiranje“ Interneta i televizije na istom kablu.

Za kanale kablovske televizije u Severnoj Americi obično je rezervisano područje između 54 i 550 MHz (izuzimajući frekvencije rezervisane za FM radio, između 88 i 108 MHz). Kanali, zajedno sa zaštitnom marginom, širine su 6 MHz. U Evropi je donja frekventna granica obično 65 MHz, a kanali su široki 6-8 MHz zbog više rezolucije koju zahtevaju sistemi PAL i SECAM, ali je u svemu ostalom šema slična kao u Americi. Donji deo područja se ne koristi. Savremeni kablovski sistemi mogu da rade i iznad 550 MHz, čak i na frekvencijama većim od 750 MHz. Odlučeno je da se za prenos od korisnika koriste kanali u području između 5 i 42 MHz (nešto više u Evropi), dok bi za saobraćaj ka korisniku služile visoke frekvencije. Frekventni spektar u mreži kablovske televizije prikazan je na slici 2-48.



Slika 2-48. Raspodela frekventnih područja u tipičnom sistemu kablovske televizije koji se koristi za pristup Internetu.

Treba ukazati na to da su svi TV signali usmereni ka korisniku, zbog čega se za saobraćaj od korisnika mogu upotrebiti pojačivači koji rade isključivo u području 5-42 MHz, a za saobraćaj ka korisniku pojačivači koji rade počev od 54 MHz pa naviše, kao na slici. Tako đobijamo asimetriju između propusnih opsega za prenos podataka od korisnika i ka korisniku zato što iznad područja rezervisanog za televiziju postoji veći deo spektra, nego ispod njega. S druge strane, očekuje se da se saobraćaj uglavnom odvija ka korisniku, pa operateri kablovskih mreža nemaju razloga da brinu. Kao što smo ranije videli, telefonske kompanije obično nude asimetričnu DSL uslugu, iako nema tehničkih ograničenja koji bi ih primoravali na takvo rešenje.

Dugački koaksijalni kablovi nisu ništa bolji za prenos digitalnih signala od dugačkih lokalnih linija, pa je i kod njih potrebna analogna modulacija signala. Obično se kanal za prenos podataka ka korisniku (širine 6 ili 8 MHz) moduliše tehnikom QAM-64 ili, ako je kvalitet kabla izuzetno visok, tehnikom QAM-256. Uz širinu kanala 6 MHz i QAM-64, dobijamo 36 Mb/s. Kada se oduzmu sistemski podaci, za korisničke podatke ostane oko 27 Mb/s. Uz QAM-256, korisnicima ostane oko 39 Mb/s. U Evropi su ove vrednosti za trećinu veće.

Tehnika QAM-64 nije pogodna za prenos podataka od korisnika zbog velikog šuma koji potiče od zemaljskih mikrotalasnih uređaja, CB radija i iz drugih izvora, tako da se koristi konzervativniji sistem QPSK. Tom tehnikom (slika 2-25) postiže se 2 bita po bodu, umesto 6 ili 8 bitova po bodu koji se ostvaraju tehnikom QAM za prenos podataka ka korisniku. Zbog toga je asimetrija između prenosa podataka ka korisniku i od njega mnogo veća nego što prikazuje slika 2-48.

Nadogradnja kablovske mreže za pristupanje Internetu ne obuhvata samo pojači- vače, već i izmenu samog centralnog razvodnika - umesto pasivnog pojačivača signala, tu je sad neophodan inteligentan digitalni računarski sistem sa optičkim interfejsom velike propusne moći ka davaocu Internet usluga. On nosi i odgovarajuće novo ime: **završni sistem kablovskog modema** (engl. *Cable Modem Termination System, CMTS).* Ipak ćemo se u daljem tekstu držati starog imena.

2.7.4 Kablovski modemi

Za pristup Internetu neophodan je kablovski modem, uređaj s dva interfejsa: jedan ka računam, drugi ka kablovskoj mreži. Na počecima kablovskog Interneta, svaki operater je imao kablovski modem sopstvene izrade koji je instalirao tehničar iz kablovske kompanije. Međutim, uskoro je postalo jasno da bi postojanje otvorenog standarda stvorilo konkurentsko tržište kablovskih modema, oborilo im cenu i privuklo potencijalne korisnike ove usluge. Štaviše, ako korisnik može da u radnji kupi kablovski modem i da ga sam instalira (kao običan telefonski V.9x modem), to bi kompaniji uštedelo troškove rada na terenu.

Zbog toga su se veći operateri kablovskih mreža udružili s kompanijom CableLabs da bi stvorili odgovarajući standard i ispitali u kojoj se meri različiti proizvodi slažu s njim. **Specifikacija interfejsa za kablovski prenos podataka** (engl. *Data Over Cable* Service *Interface Specification, DOCSIS),* kako je nazvan ovaj standard, tek je počela da ispoljava svoj uticaj na proizvođače kablovskih modema. Evropljani su svoju verziju standarda nazvali **EuroDOCSIS.** Ideja postojanja standarda za kablovske modeme nije se svidela svim

operaterima jer su mnogi od njih dobro prihodovali od iznajmljivanja sopstvenih modema korisnicima koji nisu mogli da biraju. Otvoreni standard i desetine proizvođača čiji se modemi mogu kupiti u prodavnicama obustavili bi veoma unosan posao.

Što se tiče interfejsa koji spaja modem s računarom, nema velike dileme. To je obično Ethernet (ili ponekad USB) priključak brzine 10 Mb/s. U budućnosti bi čitav modem mogao biti u obliku kartice koja se priključuje u matičnu ploču računara, slično V.9x modemima.

Drugi interfejs je složeniji. Veliki deo standarda spada u oblast radio-tehnike, što nije tema ove knjige. Ovde treba pomenuti samo to da su kablovski modemi (kao ADSL modem) uvek uključeni. Oni ostvaruju vezu čim se uključe i održavaju je dok god su uključeni, jer se kablovske usluge ne naplaćuju prema trajanju veze.

Da bismo bolje shvatili kako rade, pogledajmo šta se događa kada se kablovski modem umetne i uključi. Modem skenira kanale za prenos podataka ka korisniku tražeći specijalan paket sa sistemskim parametrima koje centralni razvodnik povremeno šalje da bi usaglasio upravo priključene modeme. Kada ga pronađe, novodošli modem objavljuje svoje prisustvo ne jednom od kanala za emitovanje. Razvodnik odgovara tako što dodeljuje modem jednom kanalu za emitovanje i jednom za prijem. Ti kanali kasnije mogu da se promene ako razvodnik treba da uravnoteži opterećenje na mreži.

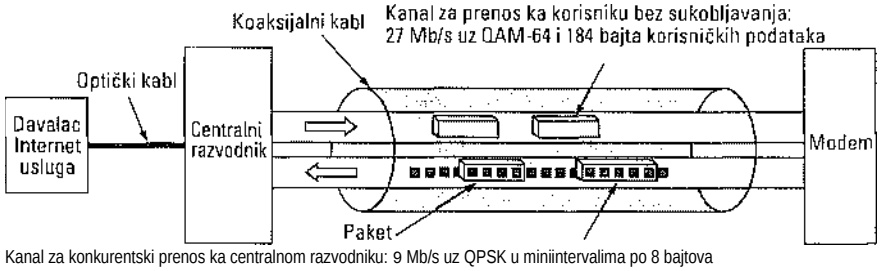
Modem zatim određuje svoju udaljenost od razvodnika tako što šalje specijalan paket i meri vreme do stizanja odgovora. To je tzv. **određivanje rastojanja** (engl. *ran- ging)*. Pomoću ovog parametra modem usklađuje način slanja podataka i ispravno se sinhronizuje. Podaci koji se šalju dele se vremenski na **miniintervale** (engl. *mimslots)*. Svaki poslati paket mora stati ujedan ili više uzastopnih miniintervala. Razvodnik periodično objavljuje početak nove serije miniintervala, ali objavu ne čuju istovremeno svi modemi jer se signal kroz kabl prostire ograničenom brzinom. Kada zna koliko je udaljen od razvodnika, svaki modem može da izračuna trenutak nastanka prvog miniintervala. Dužina miniintervala zavisi od mreže. On obično može da prihvati 8 bajtova.

Tokom inicijalizovanja, centralni razvodnik svakom modemu dodeljuje određeni miniinterval kojim modem može zahtevati propusni opseg. Po pravilu, više modema dele isti miniinterval, što dovodi do sukobljavanja. Kada računar poželi da pošalje paket, on prosleđuje paket modemu koji tada šalje zahtev za dodeljivanje odgovarajućeg broja miniintervala. Ako zahtev bude prihvaćen, centralni razvodnile šalje potvrdu modemu, naznačujući koji su miniintervali rezervisani za dati paket. Paket se tada šalje počinjući od prvog dodeljenog miniintervala. Polje u zaglavlju može se iskoristiti za zahtevanje slanja i dragih paketa.

Ako, međutim, dođe do sukobljavanja zahteva za dodeljivanje miniintervala, neće biti potvrdnog odgovora centralnog razvodnika i modem će po isteku nasumično izabranog vremenskog intervala ponoviti zahtev. Posle svakog neuspešno rešenog zahteva, nasumično izabrani vremenski interval se udvaja. (Čitaoci koji su se već upoznali s mrežama prepoznaće algoritam vremenski raspođeljenog sistema ALOHA s binarnim eksponencijalnim odustajanjem. Ethernet se ne može upotrebiti u kablovskoj mreži jer stanice ne mogu da osete medijum. Vratićemo se na ovu problematiku u 4. poglavlju.)

Kanalima za prenos podataka ka stanicama upravlja se drugačije nego kanalima za prenos ka centralnom razvodniku. Na prvom mestu, kod njih postoji samo jedan pošiljalac (centralni razvodnik), pa sukobljavanje nije moguće i nema potrebe za dodelji- vanjem miniintervala,

što je u stvari samo multipleksiranje sa statističkom podelom vremena. Zatim, saobraćaj od centralnog razvodnika obično je mnogo veći nego ka njemu, tako da se koriste paketi fiksne veličine 204 bajta. U paket ulazi i Rid-Solomonov kod za ispravljanje grešaka, kao i neki drugi sistemski podaci, tako da za korisničke podatke preostaje 184 bajta. Navedeni brojevi su izabrani zbog kompatibilnosti s digitalnom televizijom koja koristi komprimovanje MPEG-2, pa se TV signal i podaci ka korisniku formatiraju na isti način. Logička šema veza prikazana je na slici 2-49.



Slika 2-49. Detalji kanala za prenos ka centralnom razvodniku i ka korisniku tipični za Sevemu Ameriku.

Pri inicijalizovanju modema, kada modem jednom odredi rastojanje i dobije kanale za prenos ka centralnom razvodniku i od njega, kao i odgovarajuće miniintervale, on može da šalje pakete. Prvim paketom modem od davaoca Internet usluga zahteva IP adresu koja se dodeljuje dinamički protokolom DHCP, o čemu ćemo govoriti u 5. poglavlju. Modem od centralnog razvodnika zahteva i tačno vreme.

Sledeći korak se tiče bezbednosti. Pošto je kabl deljeni medijum, svako ko želi s tim da se petlja može da čita sav saobraćaj koji mimo njega prolazi. Da bi se sprečila međusobna prisluškivanja suseda (bukvalno), sav saobraćaj je u oba smera šifrovan. Deo postupka inicijalizacije obuhvata i razmenu ključa za šifrovanje. Možda vam se čini da centralni razvodnik i određeni modem nisu u stanju da tajno razmene šifre u po bela dana i naočigled hiljada korisnika koji osluškuju kabl, ali je to ipak moguće. Detaljno objašnjenje nudimo u 8. poglavlju, a za sada se zadovoljite šturim odgovorom: za razmenu se koristi Difi-Helmanov (Diffie-Hellman) algoritam.

Na kraju, modem treba da se prijavi na mrežu šaljući svoj jedinstveni identifikator bezbednim kanalom. Time se proces inicijalizacije završava. Korisnik tada može da se prijavi davaocu Internet usluga i da počne da radi.

O kablovskim modemima može se govoriti još mnogo toga. Među korisne reference spadaju: Adams i Dulchinos, 2001; Donaldson i Jones, 2001; Dutta-Roy, 2001.

2.7.5 Poređenje ADSL linije i kablovske mreže

Staje bolje - ADSL linija ili kablovska mreža? To je isto kao da pitate koji je operativni sistem bolji, koji jezik ili koja religija. Odgovor c'e zavisiti od toga kome je pitanje upućeno. Uporedimo ADSL liniju i kablovsku mrežu s nekoliko aspekata. I jedna i druga za okosnicu koriste optički kabl, ali se pri razvođenju do korisnika razlikuju: u kablovskoj mreži je koaksijalni kabl, u ADSL linijama - upredena parica. Propusna moć koaksijalnog kabla

teorijski je stotinama puta veća od propusne moći upredene parice. Međutim, pun kapacitet koaksijalnog kabla nije na raspolaganju korisnicima za razmenu podataka jer se veliki deo njegovog propusnog opsega upotrebljava za beskorisne stvari, kao što su TV programi.

U praksi je teško donositi uopštene zaključke o efektivnom kapacitetu. Davaoci ADSL usluga nude sasvim određen propusni opseg (npr. 1 Mb/s ka korisniku, 256 kb/s od korisnika) i u principu obećanje ispunjavaju sa oko 80%. Operateri kablovskih mreža ne daju nikakve izjave jer efektivni kapacitet zavisi od toga koliko je osoba trenutno aktivno na određenom korisničkom segmentu kabla. Taj kapacitet može da premaši kapacitet ADSL linije, ali i da bude manji od njega. Ono što u svemu tome nervira jeste nepredvidljivost. To što trenutno radite velikom brzinom ne znači da će tako biti i sle- dećeg minuta jer se baš tada na mrežu može uključiti najveći „gutač“ propusnog opsega u okolini.

Kada se u ADSL sistem uključuju novi korisnici, to postojeći korisnici skoro i ne primećuju jer svaki od njih ima namensku vezu. U kablovskoj mreži, sa svakim novim pretplatnikom na usluge Interneta performanse svih korisnika neizbežno postaju slabije. U takvim situacijama operater kablovske mreže jedino može da razdvaja zagušene kablove i da svaki posebno priključuje direktno za optički čvor. To, međutim, iziskuje vreme i zahteva dodatna ulaganja, pa se operateri na svaki način trude da mrežu ostave takvom kakva je.

Istaknimo uzgred da smo već proučili jedan drugi sistem koji radi s deljenim kanalom kao i kablovska mreža: sistem mobilne telefonije. Tu, takođe, grupa korisnika iste ćelije deli fiksni propusni opseg. Uobičajeno je da se opseg deli na fiksne intervale - bilo podelom frekvencija (FDM) ili podelom vremena (TDM) - jer je govorni saobraćaj prilično ujednačen. Međutim, za saobraćaj podataka takva kruta podela opsega nije efikasna jer korisnici koji razmenjuju podatke često ne rade ništa tako da se propusni opseg koji su rezervisali nepotrebno zauzima. Pa ipak, u ovom pogledu je pristup preko kablovske mreže mnogo sličniji sistemu mobilne telefonije nego fiksnom telefonskom sistemu.

Raspoloživost usluge je nešto po čemu se ADSL sistem i kablovska mreža razlikuju. Svako ima telefon, ali se ne nalaze svi korisnici dovoljno blizu lokalne centrale da bi dobili ADSL liniju. S druge strane, nisu svi priključeni na kablovsku televiziju, ali ako ste priključeni i kablovski operater nudi i pristup Internetu, možete ga dobiti. Rastojanje do optičkog čvora ili centralnog razvodnika ne predstavlja problem. Kablovska mreža je prvobitno stvorena da bude sistem za distribuiranje TV signala, pa nije zgoreg pomenuti da je zbog toga na nju povezano malo „pravnih lica“.

S obzirom da predstavlja medijum za povezivanje od tačke do tačke, ADSL linija je prirodno bezbednija od kablovske mreže. Svako ko je priključen na kablovsku mrežu može lako da čita sve pakete koji njome prolaze. Iz tog razloga, svaki kablovski operater koji drži do sebe šifrovaće sav saobraćaj u oba smera. Pa ipak, ako sused ulovi vašu poruku, makar i šifrovanu, to je manje bezbedno nego kada ne može da ulovi ništa.

Telefonski sistem je u načelu pouzdaniji od kablovske mreže. Na primer, on ima sistem za besprekidno napajanje, tako da telefoni rade i u slučaju nestanka struje. U kablovskoj mreži, pak, kada nestane struje na bilo kom pojačivaču duž kabla, svi dalji korisnici bivaju trenutno odsečeni.

Na kraju, ako se odlučite za ADSL, verovatno ćete imati veliki izbor davalaca usluga. Ponekada je i zakonom regulisano da mora postojati više davalaca. S kablovskom mrežom nije uvek tako.

Iz navedenog sledi da ADSL linija i kablovska mreža imaju više međusobnih sličnosti nego razlika. Usluge koje ova dva sistema pružaju približno su jednake, a ako se konkurencija među njima nastavi, ujednačiće im se i cene.

2.8 SAŽETAK

Fizički sloj je osnova svih mreža. Priroda svim kanalima nameće dva osnovna fizička ograničenja koja se odnose na njihovu propusnu moć. To su Nikvistovo ograničenje koje se odnosi na bešumne kanale i Šenonovo ograničenje koje se odnosi na realne kanale (sa šumom).

Medijumi za prenos podataka mogu biti materijalni (fizički) i nematerijalni (bežični). Najvažniji fizički medijumi su upredena parica, koaksijalni kabl i optički kabl. U nematerijalne medijume ubrajamo radio-talase, mikrotalase, infracrvene talase i laserske snopove koji se prostiru kroz atmosferu. Prenosni sistem koji se tek razvija jeste satelitska komunikacija, naročito sistem LEO.

Ključni element većine regionalnih mreža je telefonski sistem. Njegove glavne komponente su lokalne linije, vodovi i centrale. Lokalne linije su analogna kola od upredenih parica, pa je za prenos digitalnih podataka pomoću njih neophodan modem. ADSL linija omogućava brzine prenosa do 50 Mb/s, deleći lokalnu liniju na brojne vir- tuelne, posebno modulisane kanale. Željno očekujemo pojavu bežičnih lokalnih linija, naročito LMDS linije.

Telefonski vodovi su digitalni i mogu se multipleksirati na više načina, uključujući modulaciju podelom frekvencija (FDM), podelom vremena (RDM) i podelom talasne dužine (WDM). Kod njih je jednako zastupljeno komutiranje kola i komutiranje paketa.

Za korisnike u pokretu nije podesan fiksni telefonski sistem. Trenutno se za govorni saobraćaj masovno koriste mobilni telefoni, a uskoro će se isto tako masovno koristiti i za razmenu podataka. Prvom, analognom generacijom mobilnih telefona dominirao je sistem AMPS. Druga generacija je bila digitalna, s više sistema: D-AMPS, GSM, CDMA i drugim. Treća generacija je digitalna, zasnovna na širokopojasnom CDMA sistemu.

Alternativni sistem za pristup Internetu predstavlja mreža kablovske televizije, koja se postepeno razvila od sistema sa zajedničkom antenom do hibridnog, optičko- -koaksijalnog sistema. Ona potencijalno nudi veoma širok propusni opseg, ali on u praksi zavisi od toga

koliko je korisnika trenutno priključeno i šta oni rade.

ZADACI

1. Izračunajte koeficijente Furijeovog niza za funkciju/(r) = *t* (0 < *t* < 1).
2. Bešumni kanal širine 4 kHz uzorkuje se svake milisekunde. Kolika je maksimalna brzina prenosa podataka?
3. Televizijski kanali su širine 6 MHz. Koliko se bitova u sekundi može slati ako se koriste digitalni signali sa četiri nivoa? Pretpostavite da u kanalu nema šuma.
4. Ako se binarni signal šalje kanalom širine 3 kHz čiji je odnos signala i šuma 20 đB, kolika se maksimalna brzina prenosa može postidi?
5. Koji je potreban odnos signala i šuma da bi se nosilac TI smestio na liniju propusnog opsega 50 kHz?
6. Kakva je razlika između pasivne zvezde i aktivnog repetitora u optičkoj mreži?
7. Koliki propusni opseg odgovara širini od 0,1 pm spektra pri talasnoj dužini 1 pm?
8. Treba preneti niz slika računarskog ekr ana preko optičkog kabla. Ekran je rezolucije 480 x 640 tačaka, svakoj tački odgovara 24 bita. U sekundi se šalje 60 slika ekrana. Koliki je propusni opseg potreban i kolike će širine (pm) biti ovo talasno područje pri talasnoj dužini 1,30 pm?
9. Važi li Nikvistova teorema i za optičko vlakno ili samo za bakarnu žicu?
10. Na slici 2-6 područje na levom kraju je uže od ostalih. Zašto?
11. Radio-antene često najbolje rade kada im je prečnik jednak talasnoj dužini radio--talasa. Prečnik antene se u praksi krede između 1 i 5 metara. Koji opseg frekvencija one pokrivaju?
12. Slabljenje zbog različitih putanja najjače je kada se dva talasa razlikuju u fazi za 180 stepeni. Koliko treba da se razlikuju putanje dva talasa da bi se ostvarilo maksimalno slabljenje mikrotalasne veze frekvencije 1 GHz na rastojanju 50 km?
13. Laserski snop širine 1 mm upravljen je na detektor veličine 1 mm koji se nalazi na krovu zgrade udaljene 100 m. Koliko maksimalno ugaono skretanje (u stepenima) srne da ima laserski snop a da ne promaši detektor?
14. Šezdeset šest satelita niske orbite iz projekta Iridium podeljeno je u šest ogrlica oko Zemlje. Na visini na kojoj se nalaze, period obilaska Zemlje je 90 minuta. Koliko je srednje vreme tokom koga stacionarna zemaljska stanica može da radi sa istim satelitom?
15. Zamislite satelit koji se nalazi na visini geostacionarnih satelita, ali mu je orbitalna ravan nagnuta u odnosu na ekvatorijalnu za ugao <|). Da li je za korisnika na zemlji koji se nalazi na (j) stepeni severne geografske širine taj satelit nepokretna tačka na nebu? Ako odgovorite negativno, opišite kretanje satelita.
16. Koliko je pre 1984. godine bilo brojeva lokalnih telefonskih centrala, ako se centrala dobijala biranjem pozivnog broja područja (tri cifre) i biranjem prve tri cifre lokalnog korisničkog broja? Pozivni brojevi područja počinjali su cifrom između 2 i 9, druga cifra je bila 0 ili 1, a završavali su se bilo kojom cifrom. Prve dve cifre lokalnog broja bile su uvek između 2 i 9, a treća bilo koja.
17. Koristeći *isključivo* podatke iz knjige, izračunajte maksimalan broj telefona koji je mogao da podrži telefonski sistem SAD bez izmene strategije dodeljivanja brojeva i dodavanja nove opreme. Da li se taj broj telefonskih aparata uistinu mogao ostvariti? Umrežene računare i faksove računajte kao telefone. Pretpostavite da svaki pretplatnik koristi samo jedan aparat.

18. Jednostavan telefonski sistem sastoji se od dve lokalne centrale i jedinstvene regionalne centrale s kojom su lokalne centrale povezane potpunim dupleksnim vodom propusnog opsega 1 MHz. Po telefonu se prosečno obavi četiri razgovora tokom 8-časovnog radnog dana. Prosečno trajanje razgovora je 6 minuta. Deset procenata poziva su međugradski - prolaze kroz regionalnu centralu. Koliko najviše telefona može da podrži jedna lokalna centrala? (Pretpostavite 4 kHz po kolu.)
19. Regionalna telefonska kompanija ima 10 miliona pretplatnika. Svaki telefon je povezan sa centralom pomoću bakarne upredene parice či ja je prosečna dužina 10 km. Koliko vredi bakar u lokalnim linijama? Za izračunavanje uzmite daje prečnik svake žice 1 mm, da je gustina bakra 9 g/cm^3 i da kilogram bakra košta 3 dolara.
20. Da li je naftovod jednosmeran, poludupleksni, potpuni dupleksni sistem ili nešto četvrto?
21. Cena brzih mikroprocesora pala je na takav nivo da je sada moguće da se mikroprocesor ugradi u svaki modem. Kako to utiče na obradu grešaka u telefonskim linijama?
22. Konstelacioni dijagram modema, sličan onom na slici 2-2.5, ima tačke sa sledećim koordinatama:' (1, 1), (1, -1), (-1, 1) i (-1, -1). Koliku brzinu prenosa (b/s) može takav modem da ostvari pri 1200 boda?
23. Konstelacioni dijagram modema, nalik na onaj sa slike 2-2.5, ima tačke sa koordinatama (0, 1) i (0, 2). Da li modem koristi faznu ili amplitudnu modulaciju?
24. U konstelacionom dijagramu sve tačke leže na krugu čiji je centar u koordinatnom početku. Koja vrsta modulacije se ovde koristi?
25. Koliko frekvencija koristi potpuni dupleksni QAM-64 modem?
26. Sistem ADSL koji koristi DMT modulaciju namenjuje 3/4 raspoloživih kanala za prenos podataka ka korisniku. Na svakom kanalu se koristi QAM-64 modulacija. Koliki je ukupni kapacitet veze ka korisnicima?
27. U primeru usluge LMDS sa četiri sektora, prikazanom na slici 2-30, svaki sektor ima svoj kanal brzine 36 Mb/s. Prema teoriji stavljanja u red čekanja, ako je kanal opterećen 50%, vreme čekanja u redu biće jednako vremenu preuzimanja. Koliko ce pod ovim uslovima tr ajati preuzimanje Web strane od 5 KB? Koliko će trajati preuzimanje iste strane pomoću ADSL linije brzine 1 Mb/s? A pomoću modema brzine 56 kb/s?
28. Deset signala, od kojih je za svaki potreban propusni opseg širine 4000 Hz, multipleksiraju se na istom kanalu FDM modulacijom. Koliki minimalni propusni opseg mora imati multipleksirani kanal? Pretpostavite zaštitne margine širine 400 Hz.
29. Zašto je vreme uzorkovanja tehnikom PCM 125 ps?
30. Koliko je procentualno učešće nekorisnih podataka na nosiocu TI, tj. koji procenat od 1,544 Mb/s ne stiže do korisnika?
31. Uporedite maksimalnu brzinu prenosa podataka kroz dva bešumna kanala širine 4 kHz koji koriste
    a) analogno kodiranje (npr. QPSK) sa 2 bita po uzorku, odnosno
    b) sistem TI uz tehniku PCM.
32. Ako sistem TI izgubi korak, on pokušava da se ponovo sinhronizuje koristeći prvi bit svakog okvira. Koliko prosečno okvira treba ispitati da bi se postigla sinhroniza- cija s verovatnoćom greške 0,001?
33. U čemu je.razlika, ako uopšte postoji, između demodulacionog dela modema i dela za kodiranje kodera/dekodera? (U krajnjoj liniji, oba uređaja pretvaraju analogne signale u digitalne.)
34. Signal se prenosi u digitalnom obliku preko bešumnog kanala širine 4 kHz, po jedan uzorak svakih 125 ps. Koliko se bitova u sekundi stvarno prenosi svakom od sle- dećih

metoda kodiranja:
a) standardom CCITT brzine 2,048 Mb/s.
b) tehnikom DPCM sa 4-bitnom relativnom vrednošću signala.
c) Delta modulacijom.

35. Potpuno sinusoidalan talas amplitude A kodira se delta modulacijom, uz *x* uzoraka u sekundi. Vrednost rezultujućeg signala +1 odgovara promeni signala od +A/8, a vrednost -1 odgovara promeni signala od -A/8. Koja se najviša frekvencija može tako modulisati a da se ne akumuliraju greške?
36. Časovnici sistema SONET prave grešku veličine 1:IO^9. Posle kog vremena će odstupanje dostići širinu 1 bita? Kakve su posledice dobijenog rezultata?
37. Na slici 2-37, navedena je vrednost 148,608 Mb/s za brzinu prenosa korisničkih podataka nosiocem OC-3. Pokažite kako se ova vrednost može izvesti iz parametara SONET-ovog nosioca OC-3.
38. Da bi se prilagodio brzinama prenosa koje su manje od brzine prenosa kroz kanal STS-1, SONET ima sistem tzv. virtuelnih pritoka (VT). VT je deo korisnog sadržaja koji se može umetnuti u okvir kanala STS-1 i kombinovati s delovima drugih korisnih sadržaja u okviru za podatke. VT1.5 koristi 3 kolone, VT2 koristi 4 kolone, VT3 koristi 6 kolona, a VT6 koristi 12 kolona okvira kanala STS-1. Koja virtuelna pritoka (VT) može da prihvati
a) Uslugu DS-1 (1,544 Mb/s)?
b) Evropsku uslugu CEPT-1 (2,048 Mb/s)?
c) Uslugu DS-2 (6,312 Mb/s)?
39. Koja je suštinska razlika između komutiranja poruka i komutiranja paketa?
40. Koliki je raspoloživ korisnički propusni opseg veze ostvarene nosiocem OC-12c?
41. Svaka od tri mreže koje komutiraju pakete sadrži *n* čvorova. Prva mreža je topolo- gije zvezde sa centralnom skretnicom, druga predstavlja dvosmerni prsten, a u trećoj su svi čvorovi međusobno direktno povezani. Koje su najbolje, prosečne, odnosno najgore putanje izražene brojem skokova od čvora do čvora?
42. Uporedite kašnjenje pri slanju poruke od *x* bitova putanjom od *k* skokova u mreži s komutiranjem električnih kola i u (slabo opterećenoj) mreži s komutiranjem paketa. Vreme uspostavljanja kola je *s* sekundi, kašnjenje zbog ograničene brzine prosti- ranja signala je *d* sekundi po skoku, veličina paketa je *p* bitova, a brzina prenosa podataka je *b* b/s. Pod kojim uslovima mreža s komutiranjem paketa ima manje kašnjenje?
43. Pretpostavite da *x* bitova korisničkih podataka treba preneti putanjom od *k* skokova u mreži s komutiranjem paketa kao niz paketa od kojih svaki sadrži *p* bitova podataka i *h* bitova zaglavlja, pri čemu je $x \gg p + h$. Brzina prenosa linijama je *b* b/s, a kašnjenje zbog ograničene brzine prostiranja signala je zanemarljivo. Koja vrednost *p* minimizira ukupno kašnjenje?
44. U tipičnom sistemu mobilne telefonije sa šestougaonim ćelijama, zabranjeno je ponovo koristiti isto frekventno područje u susednoj ćeliji. Ako je na raspolaganju 840 frekvencija, koliko ih se može upotrebiti u određenoj ćeliji?
45. Stvarni raspored ćelija retko je pravilan kao na slici 2-41. Čak su i oblici pojedinih delija najčešće nepravilni. Ponudite odgovor zasto je to tako.
46. Procenite približan broj PCS mikrocelija prečnika 100 m potrebnih da pokriju područje San Franciska (120 kvadratnih kilometara).
47. Kada pokretni korisnik pređe granicu između susednih delija, ponekada se veza naglo prekine čak i ako svi predajnici i prijemnici rade savršeno. Zašto?
48. Sistem D-AMPS prenosi govor znatno lošije nego sistem GSM. Da li je to posledica

zahteva za njegovom kompatibilnošcu sa starijim sistemom AMPS, dok sistem GSM nema takvih ograničenja? Ako nije to u pitanju, šta je uzrok?

49. Izračunajte maksimalan broj korisnika koje sistem D-AMPS može da podrži u istoj deliji. Napravite istu računicu i za sistem GSM. Objasnite razliku između dobijenih rezultata.
50. Pretpostavite da *A, B* i *C* istovremeno emituju bit 0, koristeći sistem CDMA sa sekvencama podintervala prikazanim na slici 2-45(b). Koja je rezultujuća sekvenca podintervala?
51. U objašnjenju svojstva ortogonalnosti sekvenci podintervala sistema CDMA, navedeno je i sledece: ako je $S^0 T = 0$, onda je i $S \bullet T$ jedanko 0. Dokažite.
52. Razmotrite slededi način provere ortogonalnosti sekvenci podintervala sistema CDMA: odgovarajući bitovi para sekvenci mogu da se poklapaju ili da se razlikuju. Izrazite svojstvo ortogonalnosti kao poklapanje i razlikovanje odgovarajućih bitova.
53. Prijemnik u sistemu CDMA dobija sledeću sekvencu: (-1 +1 -3 +1-1-3 +1 +1). Ako su sekvence podintervala stanica onakve kao na slici 2-45(b), koje stanice su emito- vale i koje bitove je svaka od njih poslala?
54. Telefonski sistem je na korisničkom kraju oblikovan u zvezdu: sve lokalne linije iz okoline stiču se u jednoj centrali. Nasuprot tome, kablovska televizija ima jedinstven dugačak kabl koji prolazi pored svih korisnika u okolini. Pretpostavimo da ce u budućnosti TV-kabl, umesto od bakra, biti optički, brzine 10 Gb/s. Da li se pomoću njega može simulirati telefonski sistem u smislu da svako ima svoju privatnu liniju do centrale? Ako može, koliko se domova (sa po jednim telefonom) može povezati istim optičkim kablom?
55. Kablovski televizijski sistem ima 100 komercijalnih kanala i u svima se naizmenič- no emituju programi i propagandne poruke. Liči li on više na sistem TDM ili na sistem FDM?
56. Kablovska kompanija odlučuje da obezbedi i pristup Internetu za 5000 korisnika u okolini. Kompanija koristi koaksijalni kabl i takvo frekventno područje koje po kablu omogućava brzinu prenosa od 100 Mb/s ka korisnicima. Da bi privukla pretplatnike, kompanija odlučuje da garantuje najmanje 2 Mb/s ka svakom korisniku u svako doba. Opišite šta kompanija treba da uradi da bi ispunila garanciju.
57. Uz frekventna područja sa slike 2-48 i podatke navedene u propratnom tekstu, odgovorite koliko Mb/s kablovski sistem rezerviše za saobraćaj ka centralnom razvodniku, a koliko za saobraćaj ka korisnicima?
58. Kojom brzinom korisnik kablovske mreže može da prima podatke, ako u mreži nema drugog saobraćaja?
59. Multipleksiranje višestrukih tokova podataka u sistemu STS-1, zvanih pritoke (engl. *tributaries*), igra važnu ulogu u sistemu SONET. Multipleksor sa odnosom 3:1 multipleksira tri ulazne STS-1 pritoke u jedan izlazni STS-3 tok. To multipleksiranje se vrši bajt po bajt, tj. prva tri izlazna bajta predstavljaju redom prve bajtove pritoka 1, 2 i 3. Sledeća tri izlazna bajta su drugi bajtovi pritoka 1, 2 i 3 i tako dalje. Napišite program koji simulira rad multipleksora sa odnosom 3:1. Program treba da sadrži pet procesa. Glavni proces pravi četiri preostala procesa - po jedan za svaku STS-1 pritoku i jedan za multipleksor. Proces svake pritoke učitava jedan STS-1 okvir iz ulazne datoteke kao sekvencu od 810 bajtova. Svi ti procesi šalju svoje okvire (bajt po bajt) procesu multipleksora. Proces multipleksora prima te bajtove i šalje STS-3 okvir (bajt po bajt) na standardni izlaz. Komunikaciju između procesa ostvarite međuprocesnim kanalima.

3

SLOJ VEZE PODATAKA

U ovom poglavlju proučićemo principe projektovanja drugog sloja - sloja veze podataka (engl. *data link layer).* Naše proučavanje obuhvatiće algoritme za postizanje pouzdane, efikasne komunikacije između dva susedna računara na nivou sloja veze. Pod susednim računarima podrazumevamo računare međusobno povezane komunikacionim kanalom koji se teorijski ponaša kao žica (tj. kao koaksijalni kabl, telefonska linija ili bežični kanal od tačke do tačke). Suština je u tome da kanal koji se ponaša kao žica isporučuje bitove tačno onim redom kojim su poslati.

Isprva biste pomislili da je problem toliko jednostavan da zbog njega ne treba proučavati softver - računar *A* treba samo da smešta podatke na žicu, a računar *B* da ih samo prihvata. Nažalost, u komunikacionim kolima ponekada nastaju greške. Štaviše, ona podatke prenose ograničenom brzinom tako da uvek postoji izvesna vremenska razlika između trenutka slanja i trenutka prijema podataka. Navedena ograničenja pos- ledično suštinski utiču na efikasnost prenosa podataka i protokoli koji se koriste za komuniciranje moraju ih uzeti u obzir. Ti protokoli su tema ovog poglavlja.

Posle uvoda u osnovnu problematiku projektovanja sloja veze podataka, počece- mo proučavanje njegovih protokola ispitujući prirodu grešaka, njihove uzroke i načine na koji se one mogu otkriti i ispraviti. Zatim ćemo preci na niz sve složenijih protokola, od kojih svaki rešava sve više i više problema koji postoje u sloju. Izlaganje ćemo završiti ispitujući način modelovanja protokola i njihovu ispravnost, a zatim ponuditi nekoliko primera protokola sloja veze.

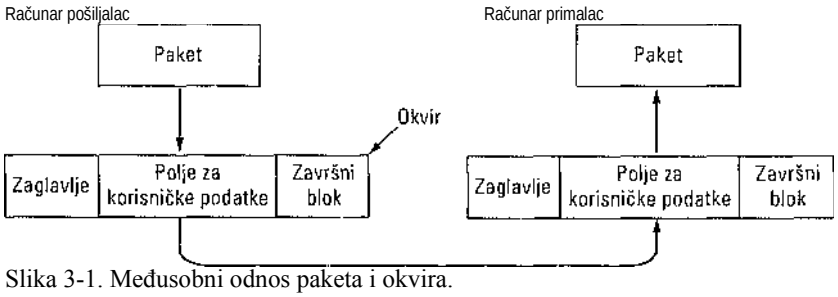
3.1 PROJEKTOVANJE SLOJA VEZE PODATAKA

Sloj veze podataka ima više specifičnih funkcija, koje se ostvaruju kroz:

1. Dobro definisan uslužni interfejs ka mrežnom sloju.
2. Obradu grešaka pri prenosu.

3. Upravljanje tokom podataka tako da računar koji prima podatke ne bude njima preplavljen.

Da bi ostvario navedene ciljeve, sloj veze podataka preuzima pakete koje dobija od mrežnog sloja i kapsulira ih u okvire (engl. *frames),* pogodne za transport. Svaki okvir sadrži zaglavlje (engl. *header),* polje za korisničke podatke (engl. *payload field)* u kome je paket, i završni blok okvira (engl. *frame trailer)* (slika 3-1). Rad sa okvirima je najvažniji posao sloja veze podataka. U narednim odeljcima ispitaćemo detaljno svaku od navedenih stavki.



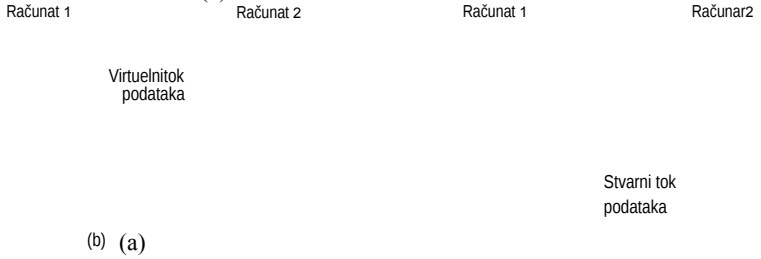
Slika 3-1. Međusobni odnos paketa i okvira.

Iako se u ovom poglavlju izričito bavimo slojem veze podataka i njegovim protokolima, mnogi principi koje ćemo proučiti, kao što su kontrola grešaka i upravljanje tokom, važe kako za transportni, tako i za druge slojeve. TJ mnogim mrežama se ove funkcije u stvari ne nalaze u sloju veze podataka, već samo u višim slojevima. Međutim, bez obzira na to gde se nalaze, ti principi su prilično jednaki, tako da nije stvarno važno na kome mestu ćemo ih proučavati. U sloju veze podataka oni se često javljaju u svom najjednostavnijem i najčistijem vidu, zbog čega su pogodni za detaljno ispitivanje.

3.1.1 Usluge koje se obezbeđuju za mrežni sloj

Sloj veze podataka treba da obezbedi usluge za mrežni sloj. Osnovna usluga je prenos podataka iz mrežnog sloja na izvorišnom računani u mrežni sloj odredišnog računara. Na izvorišnom računaru postoji deo (recimo, proces) mrežnog sloja koji isporučuje bitove sloju veze podataka da bi ih ovaj prosledio do odredišta. Posao sloja veze je da prenese te bitove do odredišnog računara da bi se oni predali njegovom mrežnom sloju, kao na slici 3-2(a). Stvarni prenos podataka odvija se prema šemi sa

slike 3-2(b), ali je lakše zamisliti da dva procesa sloja veze međusobno direktno komuniciraju pomoću odgovarajućeg protokola. Zbog toga ćemo se u čitavom ovom poglavlju služiti modelom sa slike 3-2(a).



Slika 3-2. (a) Virtuelna komunikacija, (b) Stvarna komunikacija.

Projektom se može predvideti da sloj veze podataka pruža različite usluge. Konkretne ponuđene usluge variraće od sistema do sistema, ali se obično ugrađuju barem sledeće tri usluge:

1. Prenos podataka bez uspostavljanja direktne veze i bez potvrde o njihovom prijemu.
2. Prenos podataka bez uspostavljanja direktne veze, s potvrdom o njihovom prijemu.
3. Prenos podataka sa uspostavljanjem direktne veze i s potvrdom o njihovom prijemu.

Razmotrimo detaljnije svaku od navedenih usluga.

Pri prenosu bez uspostavljanja direktne veze i bez potvrde o prijemu podataka, izvorišni računat šalje nezavisne okvire podataka odredišnom računaru ne zahtevajući od njega potvrdu o prijemu okvira. Pre prenosa se ne uspostavlja logička veza između računara, niti se po završenom prenosu takva veza raskida. Ako se neki okvir izgubi zbog postojanja smetnji na liniji, sloj veze podataka to ne registruje, niti pokušava da ispravi greške. Opisana usluga je pogodna kada je učestalost grešaka niska, tako da se ispravljanje grešaka prepušta višim slojevima. Ona je podesna i za saobraćaj koji se odvija u realnom vremenu, npr. govorni, kod koga veći problem predstavlja kašnjenje podataka, nego njihovo izobličavanje. U većini lokalnih mreža, u sloju veze koristi se usluga prenosa bez uspostavljanja direktne veze i bez potvrde o prijemu podataka.

Prenos bez uspostavljanja direktne veze, ali s potvrdom o prijemu podataka, predstavlja pouzdaniju uslugu. I dalje se ne uspostavlja logička veza između pošiljaoca i primaoca, ali se zahteva potvrda o prijemu svakog pojedinačnog okvira. Na taj način, pošiljalac zna da lije okvir stigao u ispravnom stanju. Ako utvrdi da okvir nije stigao na odredište unutar zadatog vremenskog intervala, pošiljalac ga može ponovo poslati. Opisana usluga je zgodna za prenos preko bučnih kanala, kakvi postoje naročito u bežičnim sistemima.

Treba možda naglasiti da potvrđivanje prijema okvira u sloju veze podataka nije obavezno, već samo predstavlja optimizaciju protokola. Mrežni sloj uvek može da pošalje paket i da čeka potvrdu o njegovom srećnom dolasku. Ako potvrda ne stigne u predviđenom roku, pošiljalac jednostavno može da ponovo pošalje čitavu poruku. Problem ovakve strategije jeste to što okviri obično imaju strogo ograničenu veličinu određenu hardverom, dok za pakete to ne važi. Ako se prosečan paket deli, recimo, na 10 okvira, a prosečno se gubi 20% svih paketa, tada će paketu trebati mnogo vremena da se probije. Ako se zahteva potvrda za svaki okvir i svaki se izgubljeni okvir ponovo šalje, paket stiže brže do cilja. Na pouzdanim kanalima (na primer, u optičkom vlaknu), opterećivanje kanala bezbrojnim potvrdama o prijemu okvira nije neophodno, ali na bežičnim vezama koje su po svojoj prirodi nepouzdane, takvo povećanje saobraćaja se isplati.

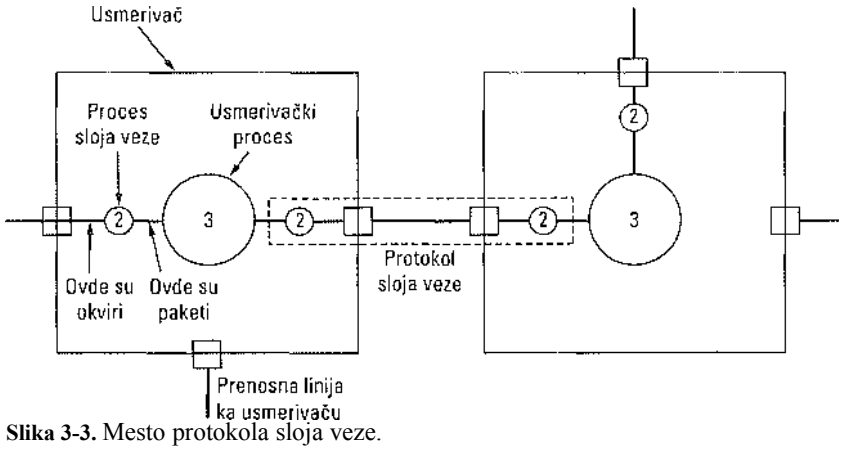
Od tri pomenute, najsloženija usluga koju nudi sloj veze podataka jeste prenos podataka sa uspostavljanjem direktne veze. Izvorišni i odredišni računar uspostavljaju vezu pre nego što se razmene podaci. Svaki poslati okvir se numerise, a sloj veze podataka garantuje daje svaki takav okvir i primljen. Štaviše on garantuje daje svaki takav okvir primljen samo jednom i da su svi okviri stigli redom kojim su poslati. Nasuprot tome, kod usluge bez uspostavljanja direktne veze lako je zamisliti da gubljenje potvrda na putu izaziva višekratno slanje (i primanje) istog paketa. Usluga sa uspostavljanjem direktne veze obezbeđuje procesima mrežnog sloja ekvivalent pouzdanog toka bitova.

Kod usluge sa uspostavljanjem direktne veze, podaci se prenose u tri jasno odelje- ne faze. U prvoj fazi se na obe strane inicijalizuju promenljive i brojači koji vode računa o primljenim okvirima. U drugoj fazi se prenosi jedan ili više okvira. U trećoj, završnoj fazi, veza se raskida, i oslobađaju se promenljive, baferi i drugi resursi upo- trebljeni za održavanje veze.

Razmotrimo tipičan primer regionalne mreže sastavljene od usmerivača povezanih iznajmljenim linijama tipa od tačke do tačke. Kada okvir stigne do usmerivača, greške se proveravaju hardverski (tehnikama o kojima ćemo malo kasnije govoriti), okvir se prosleđuje softvera sloja veze (koji može da bude ugrađen u čipu na mrežnoj kartici). Softver sloja veze proverava da lije u pitanju očekivani okvir i ako je tako, predaje paket iz polja za korisničke podatke usmerivačkom softvera (engl. *routing software)*. Usmerivački softver tada bira odgovarajuću izlaznu liniju i vraća paket softvera sloja veze, koji ga potom zaista šalje. Tok podataka kroz dva usmerivača prikazan je na slici 3-3.

Usmerivački kod često želi da posao bude odrađen kako treba, tj. s pouzdanim, sekvencijalnim vezama na svakoj liniji tipa od tačke do tačke, a ne da svaki čas ima problema sa izgubljenim paketima. Protokol sloja veze, prikazan tačkastim pravo- ugaonikom na slici 3-3, treba da nepouzdanu komunikacionu liniju tako „našminka“ da

izgleda ako ne savršeno, a ono prilično dobro. Iako smo prikazali po kopiju softvera sloja veze u svakom usmerivaču, pomenimo uzgred da u stvari ista kopija radi sa svim linijama, ali pomoću drugačijih tabela i struktura podataka za svaku od njih.



Slika 3-3. Mesto protokola sloja veze.

3.1.2 Uokvirivanje

Da bi pružio usluge mrežnom sloju, sloj veze podataka mora da iskoristi usluge koje mu nudi fizički sloj. Fizički sloj prihvata tok sirovih podataka i pokušava da ga isporuči na odredište, pri čemu ne garantuje bezgrešnu isporuku. Broj primljenih bitova može da bude manji, jednak ili veći od broja poslatih bitova, a i njihove vrednosti mogu da budu promenjene. Sloj veze podataka treba da otkrije takve greške i da ih - kada zatreba - ispravi.

Sloj veze najčešće deli tok podataka u okvire konačne veličine i izračunava kontrolni zbir (engl. *checksum)* za svaki okvir. (O algoritmima za izračunavanje kontrolnog zbira govorićerno u nastavku poglavlja.) Kada okvir stigne na odredište, kontrolni zbir se izračunava ponovo. Ako se dobijeni zbir razlikuje od zbira sadržanog u okviru, sloj veze zna daje došlo do greške i preduzima korake daje ispravi (tj. da odbaci oštećen okvir i možda da o tome povratno pošalje izveštaj).

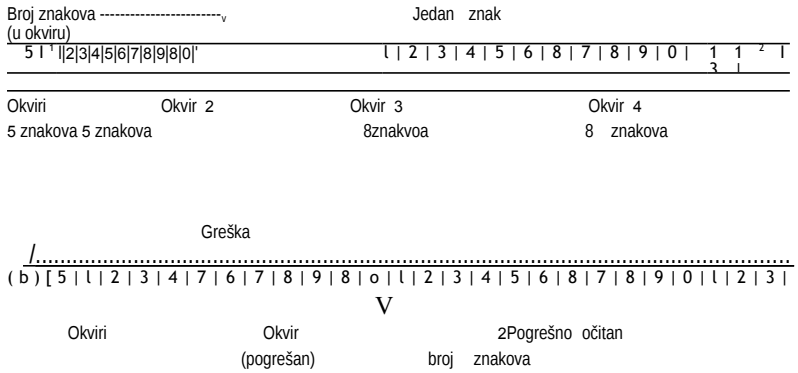
Deljenje toka podataka na okvire u stvari je teže nego što na prvi pogled izgleda. Jedan način je da se između okvira ubaci prazan vremenski interval, slično razmaku između reči u tekstu. Međutim, mreže retko garantuju sinhronizovanost događaja, pa pomenuti intervali između susednih okvira tokom transporta mogu da iščile ili, pak, da se pojave tamo gde im nije mesto.

Pošto je previše rizično računati na vremensko obeležavanje početka i kraja svakog okvira, razvijene su drugačije metode. U ovom odeljku ćemo obraditi četiri takve metode:

1. Prebrojavanje znakova.
2. Upotrebu indikatorskih bajtova uz umetanje bajtova.
3. Upotrebu početnih i završnih indikatora uz umetanje bitova.

4. Narušavanje kodiranja fizičkog sloja.

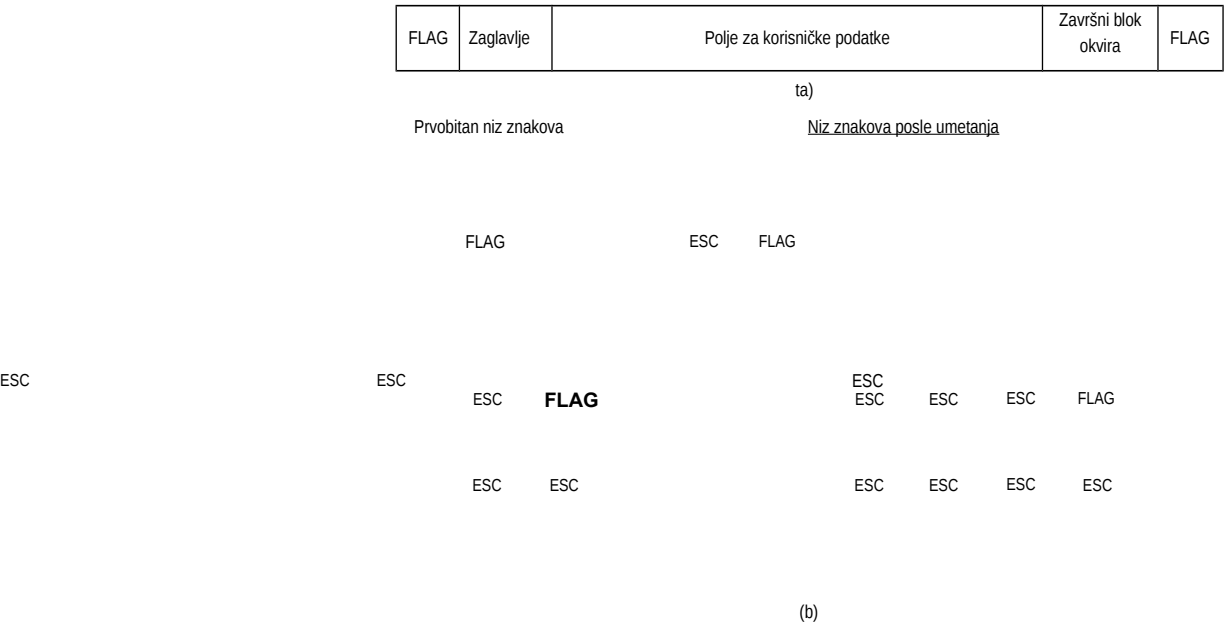
U prvoj od navedenih metoda, u zaglavlju postoji polje s brojem znakova u okviru. Kada sloj veze na odredištu pročita taj broj, on zna koliko znakova sledi iza njega i tako utvrđuje kraj okvira. Ova tehnika je prikazana na slici 3-4(a) za četiri okvira veličine 5, 5, 8 i 8 znakova.



Slika 3-4. Tok znakova, (a) Bez grešaka, (b) S jednom greškom.

U ovom algoritmu nije dobro to što broj znakova može biti pogrešno očitan zbog greške u prenosu. Na primer, ako stvarni broj znakova (5) u drugom okviru na slici 3-4(b) zbog greške postane 7, odredišni računar će izgubiti korak i neće moći da pronađe početak sledećeg okvira. Čak i ako zna daje okvir oštećen jer se njegov kontrolni zbir ne slaže, odredišni računar ipak neće moći da pronađe početak sledećeg okvira. Ni povratno slanje okvira na polazište sa zahtevom za ponovno slanje ne pomaže, pošto odredišni računar ne zna koliko znakova treba da preskoči da bi stigao do početka dela koji se ponovo šalje. Zbog svega navedenog, metoda prebrojavanja znakova više se skoro i ne koristi.

Po drugoj metodi, problem ponovnog sinhronizovanja posle greške zaobilazi se tako što se početak i kraj svakog okvira obeležavaju specijalnim bajtovima. Ranije su se početni i završni bajt razlikovali, ali u poslednje vreme većina protokola za obeležavanje kraja jednog i početka drugog okvira koristi isti **indikatorski bajt** (engl *.flag byte),* na slici 3-5(a) označen kao FLAG. Na taj način, ako odredišni računar[1] izgubi korak, on samo treba da potraži indikatorski bajt da bi pronašao kraj tekućeg okvira. Dva uzastopna indikatorska bajta označavaju kraj jednog i početak drugog okvira.



Slika 3-5. (a) Okvir ograničen indikatorskim bajtovima. (b) Četiri primera niza bajtova pre i posle umetanja bajtova.

Pri primeni ove metode nastaju ozbiljni problemi kada se prenose binarni podaci, kao što su objektni programi ili brojevi u pokretnom zarezu. Lako se može dogoditi da se sekvenca bitova indikatorskog bajta slučajno nađe u podacima. Takva situacija obično ometa ispravno očitavanje okvira. Jedan od načina kojim se ovaj problem može resiti predviđa da sloj veze pošiljaoca umetne specijalan bajt (kontrolni znak ESC) neposredno ispred svakog indikatorskog bajta koji se slučajno zadesi unutar podataka. Sloj veze primaoca uklanja taj znale pre nego što podatke preda mrežnom sloju. Opisana tehnika zove se umetanje bajtova (engl. *byte stuffing)* ili umetanje znakova (engl. *character stuffing).* Na taj način, indilcatorski bajt se može razlikovati od istog bajta unutar podataka na osnovu prisustva ili odsustva kontrolnog bajta ESC ispred njega.

Naravno, sledi pitanje: šta se događa kad se unutar podataka slučajno nađe kontrolni bajt ESC? Pa, i ispred njega se umeće kontrolni bajt ESC, pa jedan bajt ESC znači da sledeći bajt ne treba shvatiti kao indikatorski, dok dvostruki ESC znači da se bajt ESC nalazi među podacima. Na slici 3-5(b) prikazano je nekoliko primera. U svim slučajevima, niz bajtova koji se isporučuje po uklanjanju umetnutog bajta potpuno odgovara prvobitnom nizu bajtova.

Umetanje bajtova prikazano na slici 3-5 predstavlja malo uproščenu šemu umetanja koja se u protokolu PPP uglavnom primenjuje za komuniciranje kućnih računara s davaocem Internet usluga. O protokolu PPP govorićemo kasnije u ovom poglavlju.

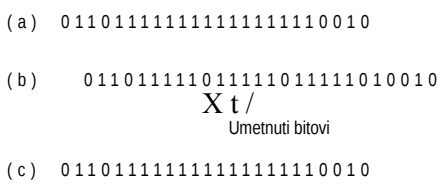
Glavni nedostatak opisane metode u okviri vanja jeste to što je čvrsto vezana za korišćenje 8-bitnih znakova. Nisu kodovi svih znakova 8-bitni; na primer, u skupu znakova

UNICODE oni su 16-bitni. Kako su se mreže razvijale, ugrađivanje dužine

koda znaka u mehanizam uokvirivanja postajalo je sve veća smetnja, tako da su morale biti smišljane nove tehnike za prenošenje znakova kodiranih proizvoljnim brojem bitova.

Nova tehnika omogućava da okviri s podacima budu različite dužine, što dopušta korišćenje znakova kodiranih proizvoljnim brojem bitova. Ona radi na sledeći način. Svaki okvir počinje i završava se specijalnom sekvencom bitova, 01111110 (u stvari, indikatorskim bajtom). Kad god sloj veze pošiljaoca u podacima naiđe na pet uzastopnih jedinica, on automatski umeće nulu u izlazni tok bitova. To umetanje bitova (engl. *bit stuffing)* analogno je umetanju bajtova, gde se ispred indikatorskog bajta u podacima umeće kontrolni bajt ESC.

Kada primalac u dolaznom toku bitova pronađe pet uzastopnih jedinica iza kojih sledi nula, on iz toka automatski izbacuje tu nulu. Umetanje (i izbacivanje) bitova potpuno je nevidljivo mrežnom sloju na oba računara, baš kao i umetanje (i izbacivanje) bajtova. Ako korisnički podaci sadrže kod indikatora, 01111110, taj indikator se prenosi kao 011111010, ali se čuva u memoriji primaoca kao 01111110. Slika 3-6 prikazuje primer umetanja bitova.



Slika 3-6. Umetanje bitova, (a) Originalni podaci, (b) Podaci tokom prenosa. (c) Podaci u memoriji primaoca posle izbacivanja bitova.

Primenom metode umetanja bitova, nedvosmisleno se utvrđuje granica između dva okvira na osnovu indikatorskog koda. Ako primalac izgubi korak, on treba samo da skenira ulazni tok i da u njemu potraži indikatorske sekvence, pošto se one mogu javiti samo na granicama okvira, ali ne i unutar podataka.

Poslednja opisana metoda uokvirivanja može se primeniti samo na mreže u kojima je kodiranje na nivou fizičkog medijuma u izvesnoj meri redundantno. Na primer, u nekim lokalnim mrežama, 1 bit podataka se kodira pomoću 2 fizička bita. Obično je bit 1 predstavljen parom visok-nizak (intenzitet signala), a bit 0 parom nizak-visok. Takav izbor znači da se na sredini svakog bita podataka menja intenzitet signala, što primaocu omogućuje da lako utvrdi granice bita. Kombinacije intenziteta: visok-vi- sok i nizak-nizak ne koriste se za podatke, ali se u nekim protokolima koriste za raz- graničavanje susednih okvira.

Na kraju razmatranja metoda uokvirivanja, pomenimo da mnogi protokoli sloja veze zbog veće sigurnosti kombinuju prebrojavanje znakova s nekom od drugih metoda. Kada okvir pristigne, koristi se polje s brojem znakova za određivanje kraja okvira. Okvir se prihvata samo ako se na njegovom kraju nalazi odgovarajući graničnik i ako se kontrolni zbir slaže. U suprotnom, u ulaznom toku se traži sledeći graničnik.

3.1.3 Kontrola grešaka

Pošto smo na odgovarajući način obeležili početak i kraj svakog okvira, nailazimo na sledeći problem: kako biti siguran da su svi okviri isporučeni mrežnom sloju primaoca i to ispravnim redom? Pretpostavimo da pošiljalac samo isporučuje okvire ne vodeći računa o

tome da li su oni stigli u ispravnom stanju. To može da bude u redu za uslugu bez uspostavljanja direktne veze i bez potvrde o prijemu, ali ne i za pouzdanu uslugu sa uspostavljanjem direktne veze.

Pouzdanost isporuke obično se obezbeđuje tako što pošiljalac dobija neku povratnu informaciju o tome šta se događa na drugom kraju linije. Najčešće protokol zahteva od primaoca da povratno šalje kontrolne okvire s potvrdom (pozitivnom ili negativnom) o pristiglim okvirima. Ako pošiljalac primi pozitivnu potvrdu o prijemu okvira, on zna daje okvir stigao na odredište u ispravnom stanju. S druge strane, negativna potvrda znači daje nešto pošlo naopako i da okvir mora da se pošalje ponovo.

Otežavajuća okolnost je i mogućnost da zbog har dverskih problema okvir potpuno nestane (na primer, zbog nagle provale šuma na liniji). U takvoj situaciji primalac neće uopšte reagovati, jer za to nema razloga. Ovo treba razumeti tako da će se protokol u kome pošiljalac šalje okvir i zatim čeka potvrdu o njegovom prijemu (pozitivnu ili negativnu), bespomoćno zauvek blokirati ako se ikada izgubi ijedan okvir zbog hardverske greške.

Takva mogućnost u sloju veze podataka predupređuje se uvođenjem tajmera. Kada pošiljalac pošalje okvir, on obično pokrene i tajmer. Tajmer je podešen na vremenski interval potreban da okvir stigne na odredište, da se tamo obradi i da potvrda o njegovom prijemu stigne natrag. U normalnim situacijama, okvir će stići na odredište, a potvrda o njegovom prijemu stići će pošiljaocu pre nego što istekne to vreme, pri čemu će se tajmer po prijemu potvrde isključiti.

Međutim, ako se okvir ili potvrda o njegovom prijemu izgube, tajmer će se posle unapred definisanog vremena automatski isključiti i ukazati pošiljaocu na moguć problem. Trivijalno rešenje je da se okvir ponovo pošalje. Međutim, kada se isti okvir šalje više puta, postoji rizik da ga i primalac primi više puta i da ga isto tako više puta prosledi mrežnom sloju. Da bi se to sprečilo, obično je neophodno da se okvirima koji se šalju dodele redni brojevi (engl. *sequence numbers*), tako da primalac može da razlikuje ponovno poslate okvire od okvira koji su stigli uobičajenim redom.

Čitava problematika rada s tajmerima i rednim brojevima okvira, koja obezbeđuje da se svaki okvir na odredištu prosleđuje mrežnom sloju samo jedanput, važan je zadatak sloja veze. U drugom delu poglavlja razradićemo tu temu prolazeći kroz sve složenije primere.

3.1.4 Upravljanje tokom podataka

Važnu stavku pri projektovanju sloja veze podataka (a i viših slojeva) predstavlja pitanje šta raditi s pošiljaocem koji sistematski šalje okvire brže nego što primalac može da ih prihvati. Takva situacija nastupa lako ako pošiljalac ima brz (ili neopterećen) računar, a primalac spor (ili opterećen) računar. Pošiljalac ne prestaje da šalje

okvire velikom brzinom sve dok primalac ne bude njima potpuno zatrpan. Čak i kad se prenos obavlja bez grešaka, u određenom trenutku primalac jednostavno neće modi da obradi sve pristigle okvire i počeđe da ih gubi. Jasno je da nešto treba preduzeti da bi se opisana situacija predupredila.

Najčešće se koriste dva pristupa. U prvom, upravljanju tokom na osnovu povratnih informacija (engl*, feedback-based flow control),* primalac šalje pošiljaocu povratnu poruku dozvoljavajući mu da šalje podatke većom brzinom ili ga barem obaveštavajući o tome kako izlazi na kraj s poslom. U drugom pristupu, upravljanju tokom zasnovanom na ograničenju brzine (engl. *rate-based flow control),* protokol ima ugrađen mehanizam koji ograničava brzinu slanja podataka, nezavistan od stanja kod primaoca. U ovom poglavlju obradićemo upravljanje tokom na osnovu povratnih informacija pošto se sistemi zasnovani na ograničenju brzine slanja nikada ne koriste u sloju veze podataka. Sisteme zasnovane na ograničenju brzine obradićemo u .5. poglavlju.

Poznate su različite vrste upravljanja tokom na osnovu povratnih informacija, ali se uglavnom sve zasnivaju na istom principu. Protokol sadrži strogo definisana pravila o tome kada pošiljalac može da pošalje sledeći okvir. Ta pravila često zabranjuju slanje okvira pre nego što primalac da podrazumevano ili izričito dopuštenje. Na primer, kada se uspostavi veza, primalac može da poruči: „Pošalji mi sada *n* okvira, ali nakon što to uradiš ne šalji okvire dalje sve dok ti to ne saopštim“. Ubrzo ćemo se detaljnije pozabaviti tim pravilima.

3.2 OTKRIVANJE I ISPRAVLJANJE GREŠAKA

Kao što smo zaključili u 2. poglavlju, telefonski sistem se sastoji iz tri dela: skretnica, vodova i lokalnih linija. Prva dva su danas u većini razvijenih zemalja gotovo potpuno digitalna. Lokalne linije su još uvek od upredene bakarne parice i tako će biti još godinama zbog visokih troškova njihove zamene. Dok do grešaka retko dolazi u digitalnom delu sistema, one su još uvek česte u lokalnim linijama. Osim toga, sve češće se koristi bežična komunikacija, a tu su greške višestruko češće nego u regionalnim optičkim kablovima. Možemo da zaključimo jedino to da ćemo se još dugo družiti s greškama u prenosu podataka. Treba da naučimo kako da se s njima izborimo.

Zbog prirode fizičkih procesa koji ih izazivaju, greške u nekim medijumima (npr. pri prenosu radio-talasima) imaju težnju da se javljaju, ne pojedinačno, već u rafalima. Takva bujica potuka ima i svoje prednosti i svoje nedostatke u odnosu na izolo- vane greške na pojedinačnim bitovima. Prednost im je to što se računarski podaci uvek šalju u blokovima bitova. Pretpostavimo daje veličina bloka 1000 bitova, a da je učestalost grešaka 0,001 po bitu. Da se greške pojavljuju nezavisno, većina blokova bi sadržala grešku. Međutim, ako se greške javljaju u rafalima, na primer, od po 100 grešaka, prosečno bi se javile u jednom ili u dva od stotinu blokova. Rafalne greške (engl. *burst errors)* imaju tu manu da se mnogo teže ispravljaju nego izolovane greške.

3.2.1 Kodovi za ispravljanje grešaka

Projektanti mreža su razvili dve osnovne strategije za obradu grešaka. Jedna je da se uz svaki blok poslatih podataka uključi i njihov višak koji bi bio dovoljan da primalac zaključi šta su bili stvarni podaci. Druga je da uključeni višak podataka bude dovoljan da primalac uspe da zaključi da se dogodila greška u prenosu (ne i koja greška) i da zahteva ponovno

slanje podataka. Prva od dve strategije koristi **kodove za ispravljanje grešaka** (engl. *error-correcting codes),* a druga **kodove za otkrivanje** grešaka (engl. *error-detecting codes).* Korišćenje kodova za ispravljanje grešaka često se naziva i **ispravljanje grešaka u hodu** (engl. *forward error correction).*

Svaka od ovih tehnika primenjuje se u posebnoj oblasti. Na visokopouzdanirn kanalima, npr. u optičkom vlaknu, jeftinije je koristiti kod za otkrivanje grešaka i samo povremeno ponovo preneti blok podataka za koji se utvrdi da je neispravan. Međutim, na kanalima s visokom učestalošću pojavljivanja grešaka, kao što su bežične veze, bolje je svakom bloku dodati višak dovoljan za rekonstruisanje stvarnih podataka, nego se oslanjati na ponovno slanje bloka koji isto tako može da sadrži grešku.

Da biste razumeli kako se greške obrađuju, moramo najpre tačno utvrditi kako greška izgleda. Okvir se obično sastoji od m bitova podataka (tj. poruke) i r redundantnih (kontrolnih) bitova. Neka je ukupna dužina okvira n (tj. $n = m + r$). Jedinica od n bitova koja sadrži i podatke i kontrolne bitove često se naziva **kodna** reč (engl. *codeword).*

Ako su zadate dve kodne reči: 10001001 i 10110001, moguće je utvrditi koliko se korespondentnih bitova u njima razlikuje. U ovom primeru, razlikuju se 3 bita. To smo utvrdili tako što smo nad kodnim recima izvršili operaciju isključive disjunkcije (isključivo ILI) i prebrajali jedinice u rezultatu:

```
10001001
10110001
00111000
```

Broj pozicija bitova u kojima se razlikuju dve kodne reči naziva se **Hamingovo rastojanje** (Hamming, 1950). Njegov smisao je u tome stoje za dve kodne reči s Hamingovim rastojanjem $d,$ potrebno jednu u dragu konvertovati đjednobitnih grešaka.

U većini slučajeva prenosa podataka, dozvoljeno je slanje svih 2^m poruka (s podacima), ali se zbog načina na koji se računaju kontrolni bitovi za to ne koristi svih 2" kodnih reči. Kada se zada algoritam za računanje kontrolnih bitova, može se napraviti potpuna lista dozvoljenih kodnih reči i u njoj pronađi dve kodne reči s minimalnim Hamingovim rastojanjem. To rastojanje je Hamingovo rastojanje celog koda.

Mogućnost koda da otkrije i ispravi greške zavisi od njegovog Hamingovog rasto- janja. Da biste detektovali cl grešaka, morate imati Hamingov kod s rastojanjem $d + 1$ jer u takvom kodu nema šanse da d jednobitnih grešaka promene važeću kodnu reč u dragu isto tako važeću kodnu reč. Kada primalac vidi da je kodna reč pogrešna, on utvrđuje daje došlo do greške u prenosu. Slično tome, da biste ispravili d grešaka, Hamingovo rastojanje mora biti $2d + 1$, jer su na taj način važeće kodne reči toliko međusobno udaljene da je, čak i uz cl promena, originalna kodna reč bliža od bilo koje druge kodne reči, pa se može nedvosmisleno utvrditi.

Kao jednostavan primer koda za otkrivanje grešaka razmotrimo kod u kome se podacima priključuje jedinstven **bit parnosti** (engl. *parity bit).* Bit parnosti se bira tako da broj jedinica u kodnoj reči bude paran (engl. *even*) ili neparan (engl. *odd).* Na primer, kada se reč 1011010 šalje uz paran broj jedinica, na kraj kodne reči se dodaje Oiona postaje 10110100. Kada se reč šalje uz neparan broj jedinica, na njen kraj se dodaje 1 i ona postaje 10110101. Kod kome je priključen jedan bit parnosti ima Hamin- govo rastojanje 2, pošto bilo koja jednobitna greška proizvodi kodnu reč pogrešne parnosti. Jedinstven bit parnosti može se upotrebiti za otkrivanje pojedinačnih grešaka.

Evo sada jednostavnog primera koda za ispravljanje grešaka koji ima samo četiri važeće kodne reči:

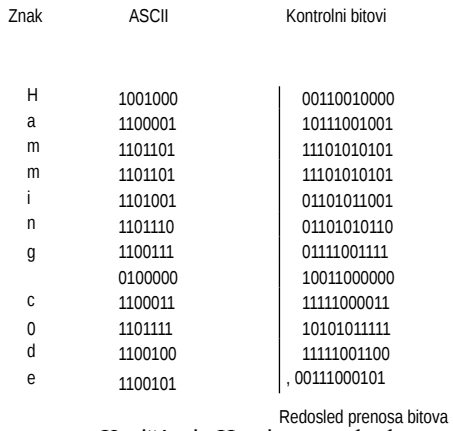
0000000000, 0000011111, 1111100000 i 1111111111

U ovom kodu Hamingovo rastojanje je 5, što znači da on može da ispravlja dvostruke greške. Kada pristigne kodna reč 0000000111, primalac zna daje to originalna kodna reč 0000011111 u kojoj su izmenjena dva bita. Ako, međutim, trostruka greška izmeni originalnu kodnu reč 0000000000 u 0000000111, greška se ne može potpuno ispraviti.

Pretpostavimo da nam treba kod sa *m* bitova podataka i *r* kontrolnih bitova koji će omogućavati ispravljanje svih pojedinačnih grešaka. Svaka od 2^m važećih poruka ima *n* nevažećih kodnih reči na rastojanju 1 od sebe. One se obrazuju sistematskim inver- tovanjem svakog od *n* bitova originalne kodne reči. Na taj način, svaka od 2^m važećih poruka „pokriva" $n + 1$ sekvenci bitova. Pošto je ukupan broj sekvenci 2", za sve dopuštene poruke potrebno je $(n + 1)2^m < 2^n$ sekvenci. Imajući u vidu daje $n = m + r$, prethodni izraz postaje $(m + r + 1) < 2^r$. Uz zadato *m* dobijamo donju granicu broja kontrolnih bitova potrebnih za ispravljanje pojedinačnih gešaka.

Navedenu teorijsku donju granicu možemo u stvari odrediti i Hamingovom metodom (1950). Bitovi kodne reči numerišu se uzastopno, počinjući od levog kraja reči. Bitovi čija pozicija predstavlja stepen od 2 (1, 2, 4, 8, 16 itd.) predstavljaju kontrolne bitove. Na ostalim pozicijama (3, 5, 6, 7, 9 itd.) nalazi se *m* bitova podataka. Svaki kontrolni bit vodi računa o tome da određeni skup bitova (zajedno s njim) ima paran (ili neparan) broj jedinica. Pomoću tog bita se mogu vršiti različiti proračuni parnosti. Da biste utvrdili koji kontrolni bitovi vode računa o bitu na poziciji *k,* napišite *k* kao zbir stepena od 2. Na primer, 11 = 1+ 2 + 8, a 29 = 1 + 4 + 8 + 16. Predmetni bit se nalazi pod budnim okom kontrolnih bitova koji se pojavljuju u zbiru (tj. bit na poziciji 11 kontrolišu bitovi 1, 2 i 8).

Kada dobije kodnu reč, primalac inicijalizuje svoj brojač na nulu. Zatim proverava svaki kontrolni bit *k (k* = 1,2, 4, 8...) u pogledu ispravne parnosti. Kada utvrdi neslaganje, primalac pomera brojač za *k.* Ako je brojač na nuli posle provere svih kontrolnih bitova (tj. kada su svi ispravni), kodna reč se prihvata kao važeća. Ako brojač ima vrednost različitu od nule, ona predstavlja poziciju neispravnog bita. Na primer, ako su pogrešni kontrolni bitovi 1, 2 i 8, invertovan je bit podataka na poziciji 11, pošto samo na njega motre kontrolni bitovi 1,2 i 8. Slika 3-7 prikazuje neke od 7-bitnih znakova iz ASCII skupa, predstavljene Hamingovim kodom kao 11-bitne reči. Imajte na umu da se podaci nalaze na pozicijama 3, 5, 6,7, 9, 10 i 11.



Slika 3-7. Korišćenje Hamingovog koda za ispravljanje rafalnih grešaka.

Hamingovi kodovi mogu da isprave samo pojedinačne greške. Međutim, prime- nom jednog trika oni se mogu naterati da ispravljaju i rafalne greške. Sekvenca od *k* uzastopnih kodnih reči poreda se u matricu, svaka reč u jedan red. U normalnim situacijama redovi se šalju jedan po jedan i to sleva nadesno. Da bi se ispravile rafalne greške, podatke treba slati u kolonama, počinjući od krajnje leve kolone matrice. Kada se pošalje *k* bitova, prelazi se na sledeću kolonu itd, kao što je prikazano na slici 3-7. Kada okvir stigne na odredište, matrica se relconstraiše iz uzastopno pristiglih kolona. Ako dođe do rafalne greške dužine *k,* ona ce izmeniti najviše 1 bit u svakoj kodnoj reči, ali Hamingov kôd to može da ispravi, pa se čitav blok podataka može obnoviti. Pomoću opisane metode, blok od *km* bitova podataka imunizuje se u odnosu na jednu rafalnu grešku najveće dužine *k* uz korišćenje *kr* kontrolnih bitova.

3.2.2 Kodovi za otkrivanje grešaka

Kodovi za ispravljanje grešaka uveliko se koriste na bežičnim vezama koje su poslovično bučne i podložne greškama u poređenju s bakarnim i optičkim kablovima. Bez koda za ispravljanje grešaka teško da bismo bežičnim putem preneli neku smislenu poruku. Učestalost grešaka pri prenosu podataka bakarnim ili optičkim kablom mnogo je manja, pa je efikasnije da se povremene greške samo otkriju i pogrešni podaci ponovo pošalju.

Razmotrimo primer jednostavnog kanala u kome je učestalost jednobitnih izolo- vanih grešaka IO'^{6} po bitu. Neka je veličina bloka 1000 bitova. Da bi se sprovelo ispravljanje grešaka u blokovima od 1000 bitova, potrebno jë 10 kontrolnih bitova; za mégabit podataka bilo bi potrebno 10.000 kontrolnih bitova. S drage strane, da bi se otkrio blok koji sadrži satno jednu jednobitnu grešku, dovoljan je 1 bit parnosti po bloku. To znači da na 1000 blokova treba poslati još samo jedan kontrolni blok (1001 bit). Ukupna količina sistemskih („ne-korisničkih“) podataka (za otkrivanje grešaka i ponovno slanje blokova) koje prema ovoj metodi treba poslati, iznosi samo 2001 bit po megabitu podataka (10.000 bitova uz primenu Hamingovog koda).

Ako se bloku priključi samo jedan bit parnosti, pa takav blok bude izobličen dugačkom

rafalnom greškom, verovatnoća da će ona biti otkrivena iznosi samo 0,5 (tj. šanse su 1:1), sto je daleko od prihvatljivog. Verovatnoća otkrivanja greške može se znatno povećati ako se svaki poslati blok posmatra kao matrica bitova sa *n* kolona i *k* redova, kao što smo ranije opisali. Bit parnosti se izračunava nezavisno za svaku kolonu i priključuje kao poslednji red matrice. Matrica se tada prenosi red po red. Kada blokovi pristignu na odredište, primalac proverava sve kontrolne bitove. Čim se neki od njih ne složi, primalac zahteva ponovno slanje čitavog bloka. Slanje istog bloka može se zahtevati sve dok se na odredištu ne slože svi bitovi parnosti.

Ovom metodom se može otkriti jedna rafalna greška dužine *n,* pošto ona menja samo jedan bit u svakoj koloni. Rafalna greška dužine *n* + 1 neće, međutim, biti otkrivena ukoliko su invertovani samo prvi i poslednji bit, dok su svi ostali bitovi ispravni. (Rafalna greška ne podrazumeva da su svi bitovi izmenjeni; dovoljno je da su izmenjeni samo prvi i poslednji bit.) Ako je blok veoma izobličen dugačkom rafalnom greškom i dejstvom više kraćih rafala, verovatnoća da će bilo koja od *n* kolona slučajno imati ispravnu parnost iznosi 0,5, tako da ukupna verovatnoća da će neispravan blok biti prihvaćen kao ispravan iznosi 2‘".

Iako opisani postupak ponekada može da bude zadovoljavajući, u praksi je raširena jedna druga metoda: **polinomski kod** (engl. *polynomial code),* poznata i kao **ciklična provera redundanse** (engl. *Cyclic Redundancy Check, CRQ.* Prema ovoj metodi, sekvenca bitova se smatra polinomom čiji su koeficijenti samo nule i jedinice. Okvir od *k* bitova smatra se listom koeficijenata polinoma od *k* članova, počev od *x*^'[1], pa do x°. To je polinom stepena *k*- 1. Najznačajniji (krajnji levi) bit jeste koeficijent člana jČ'[1]; sledeći bit je koeficijent člana itd. Na primer, sekvenca 110001 ima 6 bitova i predstavlja polinom petog stepena s koeficijentima 1, 1, 0, 0, 0, 1: $x^5 + x^4 + x°$.

Izrazi s polinomima računaju se po modulu 2, u skladu s pravilima algebarske teorije polja. Nema prenosa pri sabiranju, niti pozajmljivanja pri oduzimanju. I sabiranje i oduzimanje predstavljaju isključivu disjunkciju (XOR). Na primer:

10011011		00110011	1111000001010101
+11001010	+11001101	-10100110	-10101111
01010001	11111110	01010110	11111010

Dugačko deljenje obavlja se kao binarno, osim što se oduzimanje vrši po modulu 2, kao što je prikazano. Delitelj „ide“ u deljenik ukoliko deljenik ima isti broj bitova kao i delitelj.

Kada se primeni polinomska metoda, pošiljalac i primalac moraju prethodno da se slože oko **generatorskog polinoma** (engl. *generator polynomial), G(x).* Prvi i poslednji bit generatora moraju biti jedinice. Da bi se izračunao **kontrolni zbir** (engl. *checksum)* za okvir sa *m* bitova koji odgovara polinomu *M(x),* okvir mora biti duži od generatorskog polinoma. Namera je da se kontrolni zbir doda na kraj okvira tako da polinom koji predstavlja okvir s kontrolnim zbirom bude deljiv sa *G(x).* Kada primalac dobije okvir s priključenim kontrolnim zbirom, on pokušava da ga podeli sa *G(x).* Ako pri deljenju dobije ostatak, u prenosu je došlo do greške.

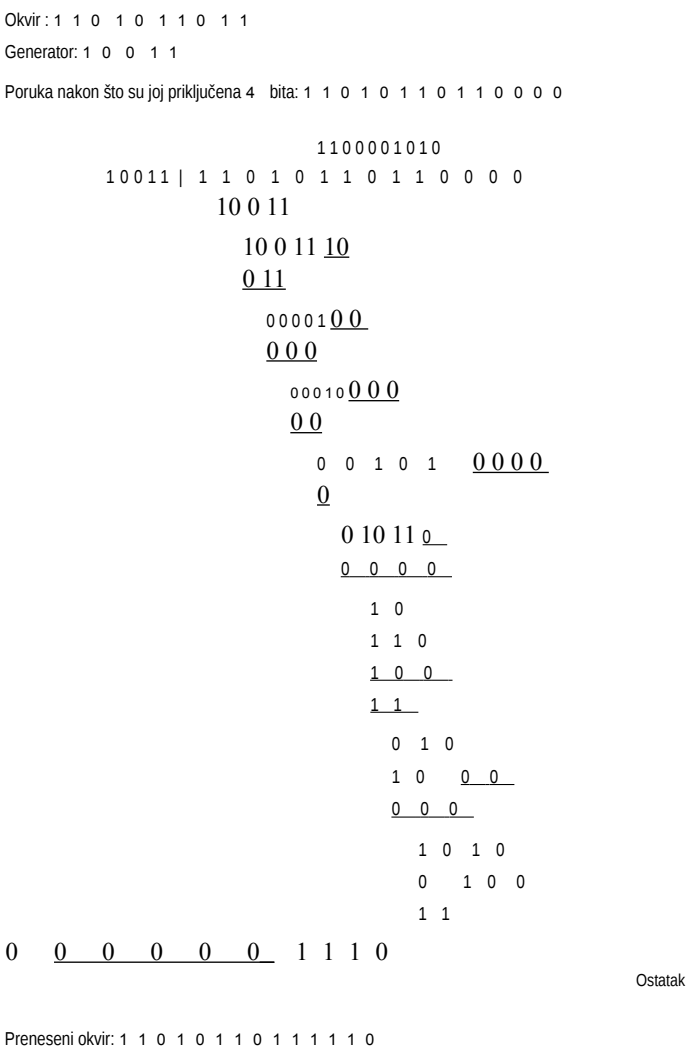
Evo kako izgleda algoritam za izračunavanje kontrolnog zbira:

1. Neka je *r* stepen polinoma *G(x)*. Dodajte *r* nula najmanje značajnom kraju okvira tako da on posle toga sadrži *m* + *r* bitova i odgovara polinomu $x^r M(x)$.

2. Podelite po modulu 2 niz bitova koji odgovara $x^r M(x)$ nizom bitova koji odgovara *G(x)*.

3. Oduzmite po modulu 2 ostatak (koji uvek ima *r* ili manje bitova) od niza bitova koji odgovaraju $x^r M(x)$. Kao rezultat čete dobiti okvir s kontrolnim zbirom koji treba poslati. Nazovimo ga polinom *T(x)*.

Slika 3-8 prikazuje proračun za okvir 1101011011, uz generator $G(x) = x^4 + x + 1$.



Slika 3-8. Izračunavanje kontrolnog zbira polinomskog koda.

Trebalo bi da vam bude jasno daje polinom *T(x)* deljiv (po modulu 2) sa *G(x).* Pri svakom deljenju, ako deljenik umanjite za ostatak, ono što ostane bice bez ostatka deljivo deliteljem. Na primer, u sistemu brojeva sa osnovom 10, ako podelite 210.278 sa 10.941, ostatak je 2399. Oduzimanjem 2399 od 210.278 dobijamo 207.879, što je bez ostatka deljivo sa 10.941.

Razmotrimo sada mogućnosti opisane metode. Kakve greške se pomoću nje mogu otkriti? Zamislite daje došlo do greške u prenosu tako da umesto niza bitova *T(x),* na odredište stigne niz *T(x) + E(x).* Svaki bit vrednosti 1 u *E(x)* odgovara bitu koji je greškom invertovan. Ako u $E(x)$ ima *k* bitova vrednosti 1, dogodilo se *k* izolovanih jednobitnih grešaka. Jedna rafalna greška prepoznaje se po početnom i krajnjem bitu vrednosti 1, između kojih je mešavina nula i jedinica, dok su svi bitovi iza krajnje jedinice 0.

Pošto primi okvir s kontrolnim zbirom, primalac ga deli polinomom G(x); drugim recima, on izračunava količnik *[T(x) + E(x)]/G(x). T(x)/G(x)* je 0, tako daje rezultat deljenja *E(x)/G(x).* Greške koje odgovaraju polinomima čiji je činilac *G(x),* biće propuštene; sve druge će biti uhvaćene.

Ako se pojavila jednobitna greška, onda je $E(x) = x^i$, gde *i* označava poziciju pogrešnog bita. Ako *G(x)* sadrži dva ili više članova, njime nikada neće moći da se po- deli *E(x),* pa će sve jednobitne greške biti otkrivene.

Ukoliko su nastale dve jednobitne greške, tada je $E(x) = x^i + x^j$, pri čemu je $i > j$. Ovo se drugačije može zapisati kao $E(x) = x^j(x^{i-j} + 1)$. Ako pretpostavimo da *G(x)* nije deljivo sax, onda je dovoljan uslov za otkrivanje svih dvostrukih grešaka to da se sa G(x) ne može podeliti x* + 1 za svako *k* do maksimalne vrednosti *i -j* (tj. do maksimalne dužine okvira). Poznati su jednostavni polinomi niskog stepena koji mogu da zaštite dugačke okvire. Na primer, $x^{15} + x^{14} + 1$ ne ide ceo broj puta *u* $x^k + 1$ ako je *k* manje od 32.768.

Ako je broj pogrešnih bitova neparan, *E(x)* sadrži neparan broj članova (npr. $x^5 + x^2 + 1$, ali ne i $x^2 + 1$). Zanimljivo je da nijedan polinom s neparnim brojem članova nema $x + 1$ kao činilac u sistemu s modulom 2. Ako x + 1 uvedemo kao činilac polinoma *G(x),* uhvatićemo sve greške s neparnim brojem invertovanih bitova.

Da biste se uverili da nijedan polinom s neparnim brojem članova nije deljiv sa x + 1, pretpostavite da *E(x)* ima neparan broj članova i da je deljiv sax + 1. Napišite *E(x)* kao proizvod činilaca $(x + 1)Q(x)$. Sada izračunajte £(1) = (1 + 1)0(1). Posto je 1 + 1=0 (po modulu 2), £(1) mora biti nula. Ako *E(x)* ima neparan broj članova, za- mena *x* vrednošću 1 uvek će kao rezultat dati jedinicu. Prema tome, nijedan polinom s neparnim brojem članova nije deljiv sa $x + 1$.

I na kraju, najvažnije od svega, polinomski kod sa r kontrolnih bitova otkriće sve rafalne greške koje nisu duže od *r.* Rafalna greška dužine *k* može se predstaviti kao $x^i(x^{k-1} + ... + 1)$, gde *i* označava udaljenost greške od desne strane okvira. Ako $G(x)$ sadrži član x°, neće imati x' kao činilac, pa ostatak nikada neće biti nula ukoliko je izraz u zagradi manjeg stepena nego polinom G(x).

Ako je dužina rafalne greške $r + 1$, ostatak deljenja sa G(x) biće nula samo ukoliko je greška identična polinomu G(x). Prema definiciji rafalne greške, njen prvi i poslednji bit moraju biti jedinice, tako da slaganje ili neslaganje zavisi od $r - 1$ bitova između njih. Ukoliko sve njihove kombinacije smatramo jednako verovatnim, verovatnoca da takav neispravan okvir bude prihvaćen kao ispravan iznosi

Može se pokazati i to da verovatnoca propuštanja rafalne greške duže od r + 1 bitova ili

više kraćih rafala, iznosi $V/$, pod pretpostavkom da su sve sekvence bitova podjednako verovatne.

Izvesni polinomi su postali međunarodni standardi. Onaj koji se koristi u mreži IEEE 802 izgleda ovako:

$$X^{32} + X^{26} + X^{23} + x^{22} + x^{16} + X^{12} + X^{11} + x^{8} + X^{7} + x^{5} + x^{4} + X^{2} + X^{1} + 1$$

Pored dobrih svojstava koje ima, on može i da otkriva sve rafalne greške do dužine 32 bita i sve rafalne greške koje proizvode neparan broj invertovanih bitova.

Iako postupak za izračunavanje kontrolnog zbira možda izgleda složeno, Petersen i Brown (1961) pokazali su da se može konstruisati jednostavan hardverski sklop s registrom za binarno pomeranje, koji izračunava i proverava kontrolni zbir. U praksi se skoro isključivo koristi takav sklop. On se koristi u svim lokalnim mrežama, a u nekim slučajevima i u vezama tipa od tačke do tačke.

Decenijama je pretpostavljano da okviri čiji kontrolni zbir treba izračunavati sadrže slučajan raspored bitova. Sve analize algoritama za izračunavanje kontrolnih zbirova rađene su uz ovu pretpostavku. Ispitivanje stvarnih podataka, međutim, pokazalo je da je ta pretpostavka potpuno pogrešna. Usled nje se pod određenim okolnostima greške propuštaju mnogo češće nego što se ranije mislilo (Partridge i sar., 1995).

3.3 OSNOVNI PROTOKOLI SLOJA VEZE PODATAKA

Načinjući temu o protokolima, počećemo opisivanjem tri protokola rastuće složenosti. Čitaoci koji traže nešto više, mogu da na Webu pronađu simulator ovih i drugih protokola (pogledajte Predgovor). Pre nego što pređemo na same protokole, korisno je da tačno objasnimo neke pretpostavke koje leže u osnovi komunikacionog modela. Najpre, pretpostavljamo da se u fizičkom sloju, sloju veze podataka i mrežnom sloju odvijaju nezavisni procesi koji međusobno komuniciraju razrnenjujući poruke. U mnogim slučajevima, procesi fizičkog sloja i sloja veze podataka izvršavaće se u procesom specijalnog mrežnog ulazno-izlaznog čipa, a kod mrežnog sloja izvršavaće se u glavnom mikroprocesora. Moguće su, međutim, i drage realizacije (na primer, tri procesa unutar istog ulazno-izlaznog čipa; ili to da se fizički sloj i sloj veze podataka predstave procedurama koje pozivaju procesi mrežnog sloja). Bilo kako bilo, kada tri sloja smatramo nezavisnim procesima, opis postaje koncepcijski jasniji, a ističe se i međusobna nezavisnost slojeva.

Draga ključna pretpostavka jeste to da računarri želi da pošalje dugačak tok podataka računam B, koristeći pouzdanu, direktno uspostavljenu vezu. Kasnije ćemo razmotriti i slučaj kada istovremeno i računar B želi da računam A pošalje podatke. Za A se pretpostavlja da ima beskonačnu zalihu podataka spremnih za slanje i da nikada ne mora da čeka na njihovo generisanje. Prema tome, kada sloj veze računara A zatraži podatke, mrežni sloj je uvek u mogućnosti da mu iziđe u susret. (Ova pretpostavka će kasnije takođe biti napuštena.)

Pretpostavljamo i to da računari rade besprekorno. Drugim recima, protokoli o kojima ćemo govoriti bave se greškama u komuniciranju, ali ne i problemima prouzrokovanim padovima sistema i ponovnim pokretanjem računara.

Što se tiče sloja veze, paket koji mu kroz interfejs stigne od mrežnog sloja sačinjavaju samo podaci koje do poslednjeg bita treba isporučiti mrežnom sloju na odredištu, Mogućnost

da mrežni sloj na odredištu deo paketa protumači kao zaglavlje ne tiče se sloja veze.

Kada sloj veze podataka prihvati paket, on ga kapsulira u okvir dodajući mu zaglavlje i završni blok koji se odnose na sloj veze (slika 3-1). Znači, okvir se sastoji od ugrađenog paketa, nešto upravljačkih podataka (u zaglavlju) i kontrolnog zbira (u završnom bloku). Okvir se prenosi do sloja veze na drugom računam. Pretpostaviće- mo da postoje odgovarajuće bibliotečke procedure za slanje (*to_physicaljayer)* i primanje okvira *(from_physical_layer)*. Transportni hardver izračunava i priključuje kontrolni zbir (kao završni blok), pa softver sloja veze o tome ne mora da brine. Za to se, na primer, može upotrebiti polinomski algoritam o kome smo ranije govorili.

Primalac na početku ne treba ništa da radi, već samo mirno da čeka da se nešto dogodi. U primerima protokola iz ovog poglavlja, označićemo čekanje sloja veze na neki događaj procedurom *waitJor_event(8ievent)*. Ta procedura vraća vrednost samo kada se nešto dogodi (na primer, kada stigne okvir). Posle toga, vrednost promenljive *event* govori o tome šta se dogodilo. Skup mogućih događaja razlikuje se od protokola do protokola, tako da ćemo ga za svaki protokol posebno definisati. Imajte na umu da se u realnim situacijama sloj veze podataka neće samo vrteti u petlji i čekati da se nešto dogodi (kao što smo uprošćeno opisali), već će paralelno raditi i drage poslove, ali će signal za prekid zaustaviti sve njegove druge aktivnosti i usmeriti ga na obradu pristiglog okvira. Ipak ćemo, jednostavnosti radi, zanemariti sve detalje paralelnih aktivnosti u sloju veze i pretpostaviti da je on sve vreme raspoloživ za obradu ovog našeg jedinog kanala.

Kada okvir stigne primaocu, hardverski se izračunava kontrolni zbir. Ako se on ne složi (tj. ako se pojavila greška u prenosu), o tome se obaveštava sloj veze podataka (*event=cksum_err*). Ako je okvir stigao neoštećen, sloj veze se obaveštava i o tome (i*event=frame_arrival*), tako da može da ga preuzme na ispitivanje procedurom *from_physical_layer.* Čim je sloj veze primaoca preuzeo neoštećeni okvir, on prove- rava upravljačke informacije u zaglavlju i ako je s njima sve u redu, prosleđuje ostatak paketa mrežnom sloju. Zaglavlje okvira nikada ne stiže u mrežni sloj.

Postoji dobar razlog što se mrežnom sloju ne sme proslediti ni deo zaglavlja okvira: tako protokoli mrežnog sloja i sloja veze ostaju potpuno nezavisni. Sve dok mrežni sloj ne zna ama baš ništa o protokolu sloja veze i formatu okvira, možete ih menjati bez potrebe da menjate softver mrežnog sloja. Neelastičan interfejs između mrežnog sloja i sloja veze podataka umnogome pojednostavljuje projektovanje softvera jer se tako mogu nezavisno razvijati komunikacioni protokoli u različitim slojevima.

Slika 3-9 prikazuje neke deklaracije (na jeziku C), koje su zajedničke za mnoge protokole o kojima ćemo govoriti. Na njoj je definisano pet struktura podataka: *boo- lean, seq_nr, packet, frame_kind iframe.* Podaci *boolean* su nabrojivog tipa i mogu imati vrednost *true i false.* Struktura *seq_nr* rezervisana je za male cele brojeve i služi za numerisanje okvira tako da ih možemo razlikovati. Niz njihovih vrednosti počinje od 0 i završava se vrednošcu *MAX_SEQ,* definisanom u svakom protokolu koji za njom ima potrebe. Struktura *packet* predstavlja jediničnu informaciju koja se raz- menjuje između mrežnog sloja i sloja veze podataka na istom računaru ili između mrežnih slojeva dva računara. U našem modelu ona uvek sadrži *MAXJPKT* bajtova, ali bi bilo realističnije daje promenljive dužine.

```
#define MAX_PKT 1024                    /* određuje veličinu paketa u bajtovima 7

typedef enum {false, true} boolean;     /* tip boolean 7
```

```
typedef unsigned int seq_nr;                    /* redni broj ili broj potvrde 7
typedef struct {unsigned char data[MAX_PKT];} packet; /* definicija paketa 7
typedef enum {data, ack, nak} frame_kind; /* definicija tipa frame_kind 7
typedef struct {                                /* okviri se prenose u ovom sloju 7
  framejđnd kind;                               /* koje vrste je    ovajokvir? 7
  seq_nr seq;                                   /* redni broj 7
  seq_nr ack;                                   /* broj potvrde    7
  packet info;                                  /* paket mrežnogsloja 7
} frame;

/* Čekaj da se nešto dogodi; vrati tip događaja u promenljivoj event. 7
void wait_for_event(event_type ‘event);
/* Uzmi iz mrežnog sloja paket za prenos kanalom. */
void from_network_layer(packet *p);
/* Podatke iz dolaznog okvira isporuči mrežnom sloju. */
void to_network_layer(packet *p);
/* Preuzmi dolazni okvir iz fizičkog sloja i kopiraj ga u r. */
void from „physical Jayerfframe *r);
/* Prosledi fizičkom sloju okvir za slanje. 7
void to_physical_layer(frame *s);
/* Pokreni sat i aktiviraj događaj timeout. */
void start_timer(seq_nr k);
/* Zaustavi sat i deaktiviraj događaj timeout. */
void stopJimer(seq_nr k);
/* Pokreni pomoćni tajmer i aktiviraj događaj ackjimeout. */
void start_ackJimer(void);
/* Zaustavi pomoćni tajmer i deaktiviraj događaj ackjimeout. */
void stop_ack_timer(void);
/* Omogući mrežnom sloju da izazove događaj network_layer_ready. */
void enablejietworkjayer(void);
/* Onemogući mrežnom sloju da izazove događaj networkJayer_ready. 7
void disablejietworkjayer(void);
/* Makro ine se ugrađuje programski: ciklično uvećava k za jedinicu. */
#define inc(k) if (k < MAX_SEQ) k = k + 1; else k = 0
```

Slika 3-9. Neke definicije za protokole koji slede. Nalaze se u datoteci *protocol.h.*

Struktura *frame* sadrži četiri polja: *kind, seq, ack* i *info,* od kojih prva tri čuvaju upravljačke podatke, a poslednje može sadržati stvarne podatke koji se prenose. Tri prva kontrolna polja nazivaju se zaglavlje okvira (engl. *frame header).*

Polje *kind* saopštava da li okvir sadrži i podatke jer neki protokoli prave razliku između okvira koji sadrže samo upravljačke informacije i okvira koji sadrže i stvarne podatke. Polja *seq* i *ack* koriste se za redne brojeve okvira i potvrda; o njima ćemo govoriti kasnije. Polje *info* okvira sadrži jedan paket podataka; u okvira koji sadrži samo upravljačke podatke polje *info* se ne koristi. Realističnije bi bilo da je polje *info* pro- menljive dužine i da ne postoji u okvira sa isključivo upravljačkim podacima.

Ponovo treba istaći razliku između okvira i paketa. Mrežni sloj pravi paket tako što uzima poruku iz transportnog sloja i dodaje joj zaglavlje mrežnog sloja. Takav paket se prosleđuje sloju veze podataka za umetanje u polje *info* odlaznog okvira. Kada okvir stigne na odredište, sloj veze izvlači paket iz okvira i prosleđuje ga mrežnom sloju. Na taj način, mrežni sloj može da radi kao da računari direktno razmenjuju pakete.

Na slici 3-9 naveden je i niz procedura. To su potprogrami iz biblioteke čiji detalji zavise od njihove realizacije; nećemo ulaziti u detalje njihovog rada. Procedura *wait_Jor_event* vrti se u petlji čekajući da se nešto dogodi, kao što smo već opisali. Sloj veze podataka koristi procedure *to_network_layer* i *from._networkJ.ayer* da bi pakete prosledio mrežnom sloju ili ih od njega prihvatio. Obratite pažnju na to da se proceduramа*from_physicaljayer* i *tojjhysicalJayer* razmenjuju okviri između sloja veze i fizičkog sloja. S drage strane, procedure *to_networkJayer* i *from_net- workJayer* služe razmenjivanju paketa između sloja veze i mrežnog sloja. Drugim recima, procedure *to_networkJayer* i *from__networkJayer* rade sa interfejsom između 2. i 3. sloja, a procedure/rom*_physicaljayer* i *to_physicaljayer* sa interfejsom između 1. i 2. sloja.

U većini protokola pretpostavljamo daje kanal nepouzdan i da povremeno gubi čitave okvire. Da bi se izborio s takvim neugodnostima, sloj veze podataka na izvorištu mora da pokrene interni tajmer ili sat kad god pošalje okvir. Ako ne dobije odgovor u unapred zadatom roku, sat se deaktivira i sloj veze podataka dobija signal za prekid.

U našim protokolima ovo se radi tako što procedura *wait_for_event* može da vrati vrednost *event=timeout.* Procedure *start jtimer* i *stopjimer* pokreću, odnosno zaustavljaju tajmer. Prekid zbog isticanja roka moguć je samo ako je tajmer uključen. Izričito je dozvoljeno da se tajmer ponovo pokrene *(start,jtimer)* dok radi; takav poziv će samo vratiti sat na nulu tako da će do prekida doći tek kada istekne pun unapred de- finisan vremenski interval (osim ako se tajmer u međuvremenu ponovo pokrene ili zaustavi).

Procedurama *start_ackjtimer* i *stopjickjimer* pokreće se, odnosno zaustavlja pomoćni tajmer koji se u izvesnim okolnostima koristi za generisanje potvrda o prijemu.

Procedure *enable_networkJayer* i *disable_networkJayer* koriste se u složenijim protokolima, gde više ne pretpostavljamo da mrežni sloj uvek ima pakete za slanje. Tek kada sloj veze podataka aktivira mrežni sloj, ovaj može da pošalje signal za prekid onda kada ima paket spreman za slanje. Takvu situaciju označavamo događajem *event=networkJayer_ready,* Kada je mrežni sloj đeaktiviran, on ne može da izazove

takav događaj. Ako pažljivo upravlja aktiviranjem i deaktiviranjem mrežnog sloja, sloj veze može da izbegne zatrpavanje paketima za koje nema dovoljno privremene memorije.

Redni brojevi okvira uvek su u zatvorenom intervalu između 0 i *MAX_SEQ,* pri čemu se *MAX_SEQ* razlikuje od jednog do drugog protokola. Često je potrebno da se niz rednih brojeva ciklično obnavlja (tj. da posle *MAX_SEQ* ponovo dođe 0). Taj posao obavlja makro *ine.* On je definisan kao makro jer se ugrađuje u kritičnu putanju. Kao što ćemo kasnije videti, performanse mreže često su ograničene brzinom obrade protokola, tako da ugrađivanje jednostavnih operacija u obliku makroa ne utiče na razumljivost koda, ali poboljšava performanse. Isto tako, pošto će vrednost *MAX_SEQ* u različitim protokolima biti različita, kada je realizujemo u obliku makroa, onda se svi protokoli bez sukobljavanja mogu ugraditi u istu binarnu datoteku. Takva mogućnost je upravo pogodna za simulator.

Deklaracije sa slike 3-9 predstavljaju sastavni deo svih protokola o kojima ćemo govoriti. Zbog ograničenosti prostora u knjizi, ali i kao zgodan priručni materijal, one su izvučene iz protokola i navedene najednom mestu; međutim, koncepcijski, one bi trebalo da su povezane s protokolima. To povezivanje se na jeziku C radi tako što se definicije smeštaju u specijalnu datoteku zaglavlja (u ovom primeru, to je datoteka *protocol.h)* i pomoću metainstrukcije #include, koja postoji u pretprocesom jezika C, uključuju u datoteke protokola.

3.3.1 Protokol za neograničen jednosmeran prenos podataka

Za prvi primer izabrali smo protokol koji ne može biti jednostavniji. Podaci se prenose samo u jednom smeru. Mrežni slojevi su uvek spremni i na izvorištu i na odredištu. Vreme obrade je zanemarljivo. Na raspolaganju su baferi neograničene veličine. Najbolje od svega, komunikacioni kanal između slojeva veze nikada ne izobličava, niti gubi okvire. Ovakav, potpuno nerealan protokol, koji smo nazvali „utopija“, prikazan je na slici 3-10.

```
/* Protokol 1 (utopija) obezbeđuje prenos podataka samo u jednom smeru - od pošiljaoca ka
   primaocu. Pri tom se pretpostavlja da komunikacioni kanal ne sadrži greške i da je prijemnik
   u stanju da beskonačno brzo obradi sve ulazne podatke. Shodno tome, pošiljalac samo
   „upumpava" podatke na liniju najbrže što može. */

typedef enum {frame_arrival} event_type;
#include „protocol.h"


  frame s;
  packet buffer;


void senderl(void)

                        /* bafer za okvir koji se šalje */ /* bafer za paket koji se šalje */

  vvhile (true) {
      from_network_layer(&buffer); /* uzmi nešto za slanje */
      s.info = buffer;                                          /* kopiraj ga u s za slanje 7
```

```
      to_physical_layer(&s);          /* pošalji ga 7
                                      /* Sve sutra i sutra i sutra,
                                         Mili to iz dana u dan sitnim
                                         korakom Do zadnjeg slova u knjizi
                                         vremena.
                                         - Magbet, V, v 7 (IJ prevodu dr Svetislava Stefanovića,
                                         Državna štamparija Kraljevine Jugoslavije,
                                         Beograd, 1939. Prim. prev.) 7

void    receiver    1
(void) {
 frame r;
 event_type event;                    /* postoji zbog procedure čekanja, ali se ovde ne koristi 7

 while (true) {                       frame_arrival 7
      wait_for_event(&event);
      from_physicalJayer(&r);         I*
      to_network_layer(&r.inf              uzmi dolazni okvir 7
      o);
 } /* jedini mogući događaj je       /* prosledi podatke mrežnom sloju 7
}
```

Slika 3-10. Protokol za neograničen jednosmeran prenos podataka.

Protokol se sastoji od dve jasno odeljene procedure: pošiljaoca i primaoca. Pošiljalac se izvršava u sloju veze izvorišnog računara, dok se primalac izvršava u sloju veze odredišnog računara. Ovde se okviri ne numerišu, niti se šalju potvrde o prijemu, tako da promenljiva *MAX_SEQ* nije potrebna. Jedini mogući događaj je *frame_arrival* (stizanje neoštećenog okvira).

Procedura pošiljalac predstavlja beskonačnu petlju while koja samo gura podatke na liniju što brže može. Telo petlje sadrži tri akcije: uzmanje paketa od (uvek spremnog) mrežnog sloja, konstruisanje okvira za slanje pomoću promenljive *s* i samo slanje okvira. U p rotokolu iz primera koristi se samo polje *info* okvira jer su ostala polja rezervisana za obradu grešaka i upravljanje tokom podataka, a grešaka i problema s tokom u našoj liniji nema.

Procedura primalac jednako je jednostavna. Na početku, ona čeka da se nešto dogodi, a to nešto može da bude samo dolazak neoštećenog okvira. Okvir najzad pristiže i procedura *wait_for_event* vraća vrednost *event=frame_arrival* (na koju se ionako ne obraća pažnja). Pozivanjem procedure *from_physical_layer,* pristigli okvir se pre- mešta iz hardverskog bafera u promenljivu r, gde mu može pristupiti kod primaoca. Na kraju se deo s podacima prosleđuje mrežnom sloju i sloj veze podataka se vraća na čekanje novog okvira, efektivno ne radeći ništa dok on ne stigne.

3.3.2 **Jednosmerni protokol „stani i čekaj“**

Sada ćemo odbaciti najmanje realističnu pretpostavku protokola 1, tj. da mrežni sloj primaoca može trenutno da obradi sve dolazne podatke (ili, što je isto, postojanje bafera neograničene veličine u sloju veze primaoca, koji može da skladišti sve pristigle

okvire do trenutka njihove obrade). I dalje, međutim, pretpostavljamo da komunikacioni kanal radi nepogrešivo, a saobraćaj još uvek ide samo u jednom smeru.

Glavni problem s kojim ovde treba da se izborimo jeste da sprečimo pošiljaoca da zatrpa primaoca podacima koje ovaj ne može dovoljno brzo da obradi. U suštini, ako je primaocu potrebno vreme A*t* da izvrši procedure *from_physical_layer* i *to_net- work_layer,* pošiljalac mora da šalje okvire prosečnom brzinom koja je manja od jednog okvira tokom vremena A*t*. Štaviše, ako pretpostavimo da se kod primaoca ne vrši hardversko privremeno skladištenje i stavljanje podataka u red čekanja, pošiljalac nikada ne sme da pošalje nov okvir dok prethodni ne preuzme procedura *from_phy- sical_layer,* da ga ne bi prebrisao.

U okolnostima kada postoje izvesna ograničenja (na primer, sinhroni prenos ili kada je sloj veze primaoca potpuno usmeren na obradu jedne ulazne linije), pošiljalac može da u protokol 1 jednostavno umetne vremensku zadršku koja će ga usporiti i tako sprečiti da primaoca ne zatrpa podacima. Međutim, češći je slučaj da'svaki sloj veze podataka mora da motri na nekoliko linija, tako da vreme između pristizanja okvira i njegove obrade može znatno da varira. Kada bi projektanti mreže mogli da predvide ponašanje primaoca u najnepovoljnijim uslovima, mogli bi pošiljaoca da nateraju da okvire šalje tako sporo da jedan okvir nikada ne prebriše dragi. Takav pristup je, međutim, krajnje konzervativan i u načelu dovodi do niskog iskorišćenja propusnog opsega, osim ako se primalac u najboljem slučaju ponaša skoro isto kao i u najgorem (tj. ako vreme reagovanja sloja veze podataka varira u uskim granicama).

Opštije rešenje ovog problema bilo bi da primalac šalje nekakve povratne informacije pošiljaocu. Pošto prosledi paket svom mrežnom sloju, primalac šalje povratno mali prazan okvir koji pošiljaocu dozvoljava da pošalje sledeći okvir s podacima. Pošiljalac, nakon što pošalje paket, prema protokolu mora da čeka dok ne stigne mali prazan okvir (tj. potvrda o pristizanju prethodnog paketa na odredište). Korišćenje povratnih informacija koje pošiljaocu omogućavuju da zna kada treba da pošalje sledeći okvir, primer je upravljanja tokom podataka koji smo ranije pomenuli.

Protokoli po kojima pošiljalac ne šalje uzastopne okvire sve dok za svaki ne dobije potvrdu o prijemu, nazivaju se protokoli tipa „stani i čekaj“ (engl. *stop-and-wait)*. Na slici 3-11 prikazanje primer takvog protokola.

```
/* Protokol 2 („stani i čekaj") takođe obezbeđuje samo jednosmeran prenos podataka od
   pošiljaoca ka primaocu. Za komunikacioni kanal se i dalje pretpostavlja da radi bez greške,
   kao u protokolu 1. Međutim, ovde primalac ima bafer ograničenog kapaciteta i podatke
   obrađuje ograničenom brzinom, pa protokol eksplicitno mora da spreči pošiljaoca da zatrpa
   primaoca podacima koje ovaj ne stiže da obradi. */

typedef enum {frame_arrival} event_type;
#includе „protocol.h"

void sender2(void)

        /* bafer za okvir koji se šalje */ frame s; packet
                                            buffer; event_type
                                            event;

                                                        /* bafer za paket koji se šalje 7
                                                    /* jedini moguć događaj je frame_arrival 7
```

```
  while (true) {
      from_networkJayer(&buffer); /* uzmi nešto za slanje 7 s.info = buffer;  /*
                kopiraj ga u s za slanje 7
      to_physicalJayer(&s);                                    /*srećan put, mali okviru 7
      wait_for_event(&event);          /*  ne nastavljaj dok ne dobiješ dozvolu 7
  }
}

 void receiver2(void)                          to_networkJayer(&r.info); to_physicalJayer(&s);
 1                                            /* baferi za okvire 7
  frame r, s;                                 /* jedini moguć događaj je frame_arrival 7
  event_type event;
  while (true) {                              /* jedina mogućnost je frame_arrival 7
      wait_for_event(&event); f               /* uzmi okvir koji je stigao 7
      rom _physical_layer(&r);                /* prosiedi podatke mrežnom sloju 7
                                              /* pošalji prazan okvir da ponovo aktiviraš pošiljaoca 7

}
```

Slika 3-11. Protokol za jednosmemi prenos tipa „stani i čekaj".

Iako se prenos podataka u ovom primeru odvija u jednom smeru - od pošiljaoca ka primaocu, okviri putuju u oba smera. Prema tome, komunikacioni kanal između dva sloja veze mora da omogući dvosmerni prenos informacija. Međutim, predmetni protokol strogo reguliše saobraćaj, tako da se on u jednom trenutku odvija samo u jednom smeru: prvo pošiljalac šalje okvir, pa primalac šalje potvrdu, zatim pošiljalac šalje drugi okvir, pa primalac šalje drugu potvrdu itd. Taj posao može da obavi polu- dupleksni fizički kanal.

Kao u protokolu 1, pošiljalac počinje tako što uzima paket iz mrežnog sloja, od njega pravi okvir i šalje ga. Ali sada, za razliku od protokola 1, pošiljalac mora da čeka potvrdu o njegovom prijemu pre nego što se vrati na početak i preuzme sledeći paket od mrežnog sloja. Sloj veze pošiljaoca ne mora čak ni da pregleda dolazni paket; to može da bude samo potvrda o prijemu.

Jedina razlika između primalaca *receiverl* i *receiver2* ogleda se u tome što posle isporučivanja paketa mrežnom sloju, *receiverž* šalje pošiljaocu potvrdu o prijemu pre nego što ponovu uđe u petlju čekanja. Pošto je pošiljaocu važno samo da primi okvir i ne zanima ga njegova sadržina, primalac u njega ne mora da smešta nikakve podatke.

3.3.3 Protokol za jednosmerno slanje podataka bučnim kanalom

Razmotrimo sada uobičajeni komunikacioni kanal u kome nastaju greške. Okviri se mogu oštetiti ili potpuno izgubiti. Međutim, pretpostavljamo da će hardver primaoca otkriti okvir oštećen u prenosu kada izračuna njegov kontrolni zbir. Ako je

okvir oštećen na takav način daje njegov kontrolni zbir ostao isti (malo verovatna situacija), ovaj protokol (a i svi drugi protokoli) propustiće takav okvir i isporučiti ga mrežnom sloju.

Na prvi pogled izgleda da bismo mogli upotrebiti neku varijantu protokola 2, na primer, da mu dodamo tajmer. Pošiljalac šalje okvir, ali primalac šalje potvrdu samo ako ga primi u ispravnom stanju. Kada oštećen okvir stigne primaocu, on ga odbacuje. Pošiljalac čeka potvrdu, a kada istekne zadati rok, ponovo šalje paket. Taj proces se može ponavljati sve dok paket ne stigne u ispravnom stanju.

Opisana šema sadrži fatalan propust. Imajte to na umu i pokušajte da otkrijete šta bi to bilo pre nego što nastavite sa čitanjem.

Da biste utvrdili šta nije u redu, setite se daje zadatak sloja veze da obezbedi bezgrešnu, transparentnu komunikaciju između procesa u mrežnom sloju. Mrežni sloj na računaru A predaje niz paketa svom sloju veze, koji mora da garantuje da će sloj veze na računani *B* isporučiti identičan niz paketa svom mrežnom sloju. Konkretno, mrežni sloj na računaru *B* nema načina da sazna da li je neki paket izgubljen ida li je neki isporučeni paket duplikat, pa sloj veze mora da garantuje isporuku paketa bez dupli- ranja, bez obzira na sve interakcije različitih grešaka u prenosu, ma kako one bile nevero vatne.

Razmotrite sledeci scenario:

1. Mrežni sloj na računam *A* predaje paket 1 svom sloju veze. Paket ispravno stiže do računara *B* i prosleđuje se njegovom mrežnom sloju. Računar *B* šalje potvrdu o tome računam A.
2. Okvir s potvrdom (engl. *acknowledgement frame)* potpuno se gubi. On nikada ne stiže do računara A. Lakše bi se disalo kada bi kanal kvario i gubio samo okvire s podacima (engl. *data frames),* a ne i upravljačke okvire (engl. *control frames),* ali ne vredi kukati, kanal ne može da ih razlikuje.
3. Rok za čekanje potvrde na računaru A ističe. Njegov sloj veze koji nije dobio potvrdu (pogrešno) pretpostavlja da je okvir s podacima izgubljen i ponovo šalje okvir s paketom 1.
4. Duplikat okvira takođe stiže do sloja veze računara *B* u potpuno ispravnom stanju i odmah se prosleđuje odgovarajućem mrežnom sloju. Ako računar A računaru *B* šalje datoteku, deo datoteke će biti dupliran (kopija datoteke koju će sastaviti računar B biće neispravna, a greška neće biti otkrivena). Drugim recima, protokol je baš „zabrljao".

Jasno je da treba naći način da primalac uspe da razlikuje okvir koji vidi prvi put od okvira koji je ponovo poslat. Trivijalno rešenje je da pošiljalac stavi redni broj u zaglavlje svakog okvira koji šalje. Tada primalac može da proverom rednog broja utvrdi da li je u pitanju nov okvir ili duplikat koji treba odbaciti.

Pošto je poželjno da zaglavlje okvira bude što manje, postavlja se pitanje: koliko je najmanje potrebno bitova za smeštanje rednog broja? Jedina dilema koju protokolom treba rešiti tiče se okvira *m* i okvira *m* + 1 koji ga neposredno sledi. Ako se okvir *m* ošteti ili izgubi, primalac neće poslati potvrdu, pa će se pošiljalac i dalje truditi da ga pošalje. Kada okvir stigne ispravno, primalac će poslati o tome potvrdu. E, tu može da dođe do nezgode. U zavisnosti od toga da li pošiljaocu stigne ispravna ili neispravna potvrda (ili uopšte ne stigne), on će poslati okvir *m* ili *m* + 1.

Događaj koji prisiljava pošiljaoca da pošalje okvir *m* + 2 jeste pristizanje potvrde o prijemu okvira *m* + 1. Ali to podrazumeva daje okvir *m* ispravno primljen, a i daje potvrda o tome ispravno stigla pošiljaocu (bez toga pošiljalac ne bi počeo da šalje okvir /7? + 1, a kamoli *m* + 2). Prema tome, greška može nastati samo između okvira i njegovog prethodnika, odnosno sledbenika, a nikako između prethodnika i sledbeni- ka određenog okvira.

Iz rečenog sledi da je za redni broj dovoljan jedan bit (redni brojevi 0 ili 1). U svakom

trenutku primalac očekuje da usledi određeni redni broj. Svaki pristigli okvir koji ne ispuni to očekivanje odbacuje se kao duplikat. Kada stigne okvir sa očekivanim rednim brojem, on se prihvata i prosleđuje mrežnom sloju. Zatim se očekivani redni broj uvećava po modulu 2 (0 postaje 1, a 1 postaje 0).

Primer protokola ovakve vrste prikazan je na slici 3-12. Protokoli kod kojih pošiljalac čeka pozitivnu potvrdu pre nego što pošalje sledeći okvir s podacima često se zovu **protokoli s potvrđivanjem i ponovnim slanjem** (engl. *Positive Acknowledgement with Retransmission, PAR)* ili **protokoli sa automatskim ponavljanjem zahteva** (engl. *Automatic Repeat reQuest, ARQ)*. Slično protokolu 2, protokol iz našeg primera takođe prenosi podatke samo u jednom smeru.

```
/* Protokol 3 (PAR) omogućava jednosmerni prenos nepouzdanim kanalom. 7

#define MAX_SEQ 1                          /* mora biti 1 za protokol 3 7

typedef enum {frame_arrival, cksum_err, timeout} event_type;
#include „protocol.h"
void sender3(void)
{
  seq_nr next_frame__to_send;              /* redni broj sledećeg okvira koji se šalje 7
  frame s;                                 /* privremena promenljiva 7
  packet buffer;                           /* bafer za pakete koji se šalju 7
  eventjype event;
  next_frame_to_send = 0;                  /* inicijalizovanje rednih brojeva odlaznih
                                              paketa 7
  from_network_layer(&buffer);             /* uzmi prvi paket  7
  while (true) {
      s.info = buffer;                     /* napravi okvir za prenošenje 7
      s.seq = nextJrame_Jo_send;           /* unesi redni broj u okvir 7
      to_physical_layer(&s);               /* pošalji okvir 7

      start_timer(s.seq);                  /* P ako okvir stigneposle zadatog vremena,
                                              isključi se 7
      wait_for_event(&event);              /* mogući događaji: frame_arrival, cksum_err,
                                              timeout 7
      if (event == frame_arrival) {
            from_physicalJayer(&s);        /* uzmi potvrdu 7
```

```
            if (s.ack == next_frame_Jo_send) {
                stop_timer(s.ack);                  /*   isključi tajmer 7
                from_network_layer(&buffer); /* uzmi sledeći okvir za slanje 7
inc(next_frame_to_send);                                    /* invertuj vrednost promenljive
                                                              next_frame_to_send 7

            }
        }
}
}

void receiver3(void)
{
seq__nr frame_expected; frame r, s; event_type event;
frame_expected = 0; while (true) {
wait_for_event(&event); if (event == frame_arrival) { from_physicalJayer(&r); if   /* mogućnosti: ftame_arrival, cksum_err 7
(r.seq == frame_expected) { to_network_layer(&r.info); inc(frame_expected);        /* stigao je ispravan okvir. 7
                                                                                   /* uzmi pristigli okvir 7
                                                                                   /* na to smo čekali. 7
                                                                                   /* prosledi podatke mrežnom sloju 7
                                                                                   /* sledeći put očekuj onaj drugi redni broj 7

                                                                                   /* zabeleži koji je okvir potvrđen 7 /* pošalji
                                                                                   potvrdu 7

}
s.ack = 1 frame_expected; to_physical_layer(&s);
        }
}
}
```

Slika 3-12. Protokol s potvrđivanjem i ponovnim slanjem.

Protokol 3 razlikuje se od svojih prethodnika po tome što i pošiljalac i primalac imaju promenljivu čija se vrednost čuva dok se sloj veze nalazi u stanju čekanja. Pošiljalac čuva redni broj sledećeg okvira koji treba da pošalje u promenljivoj *next_fmme_ to_send;* primalac čuva redni broj sledećeg očekivanog okvira u promenljivoj *frame_expected.* Svaki protokol se inicijalizuje pre nego što uđe u beskonačnu petlju.

Pošto pošalje okvir, pošiljalac uključuje tajmer. Ako on već radi, vraća ga na nulu da bi čekao pun zadati rok. Rok posle koga će se tajmer automatski isključiti tr eba izabrati tako da okvir ima vremena da stigne do primaoca, da tamo bude obrađen u najnepovolj- nijim uslovima i da se potvrda o njegovom prijemu vrati do pošiljaoca. Samo kada iste- kne rok tajmera treba pretpostaviti da su se poslati okvir ili potvrda o njegovom prijemu izgubili u putu i tada poslati duplikat. Ako je rok prekratak, pošiljalac će nepotrebno slati okvire. Iako to neće izazvati greške, performanse prenosa će se pogoršati.

Nakon što pošalje okvir i uključi tajmer, pošiljalac čeka da se dogodi nešto uzbudljivo. Za to postoje samo tri mogućnosti: da mu pristigne neoštećena potvrda o prijemu, da stigne oštećena potvrda o prijemu ili da se tajmer isključi. Ako stigne ispravna potvrda, pošiljalac uzima sledeći paket od svog mrežnog sloja i smešta ga u

bafer, brišući njime prethodni paket. On takođe uvećava redni broj. Ako mu, međutim, stigne oštećena potvrda o prijemu ili istekne rok tajmera pre nego što išta stigne, on ne dira bafer, niti menja redni broj, pa odmah može da pošalje duplikat.

Kada primalac dobije ispravan okvir, proverava njegov redni broj. Ako utvrdi daje to nov okvir, prima ga i prosleđuje mrežnom sloju, a zatim šalje potvrdu o prijemu. Duplikati i oštećeni okviri ne prosleđuju se mrežnom sloju.

3.4 PROTOKOLI KLIZNIH PROZORA

U prethodnim protokolima okvire smo prenosili samo u jednom smeru. Međutim, većinom postoji potreba da se okviri istovremeno prenose u oba smera. To se može postići korišćenjem dva paralelna komunikaciona kanala za jednosmeran saobraćaj (u suprotnim smerovima). Ako to uradimo, dobijamo dva zasebna fizička kola, svako s kanalom za slanje (podataka) i kanalom za primanje (potvrda). U oba slučaja povratni kanal se koristi skoro potpuno neefikasno. U stvari, korisnik plaća dva kola, a koristi kapacitet samo jednog.

Bolje je ako se isti kanal koristi za slanje podataka u oba smera. Uostalom, u protokolima 2 i 3 već smo slali okvire u oba smera, a povratni kanal je bio istog kapaciteta kao i direktni kanal. Po tom modelu, okvire podataka koje računar *A* šalje računaru *B* smenjuju okviri s potvrdama koje računar A takođe šalje računam *B*. Ispitujući polje *kind* u zaglavlju dolaznog okvira, primalac može utvrditi da li je u pitanju okvir s podacima ili s potvrdom.

Iako preplitanje okvira s korisničkim i upravljačkim podacima na istom kanalu predstavlja poboljšanje u odnosu na postojanje dva zasebna fizička kola, sve se to može i bolje izvesti. Umesto da pošalje zaseban upravljački okvir kada stigne okvir s podacima, primalac može da sačeka da mu njegov mrežni sloj prosledi sledeći paket podataka. Potvrda se tada priključuje odlaznom okviru s podacima (u polju *ack* u zaglavlju okvira). U stvari, potvrda „besplatno“ putuje sa sledećim okvirom podataka. Tehnika zadržavanja potvrde da bi se ona priključila sledećem odlaznom paketu s podacima poznata je kao „šlepovanje“ (engl. *piggybacking).*

Osnovna prednost šlepovanja u odnosu na slanje posebnih okvira s potvrdom o prijemu ogleda se u efikasnijem korišćenju propusnog opsega kanala. Polje *ack* u zaglavlju okvira dugačko je samo nekoliko bitova, dok se zaseban okvir sastoji od zaglavlja, potvrde i kontrolnog zbira. Osim toga, manje okvira znači i manji broj do- gađaja *frame_arrival,* što možda smanjuje potrebu za veličinom bafera kod primaoca, u zavisnosti od toga kako mu je softver organizovan. U narednom protokolu koji ćemo razmotriti, polje za šlepovanje produžava zaglavlje za samo 1 bit, a i inače nije duže od samo nekoliko bitova.

Međutim, tehnika šlepovanja unosi komplikaciju koje nije bilo kad su se slale zasebne potvrde. Koliko dugo sloj veze treba da čeka paket na koji bi nakačio potvrdu? Ako čeka duže od zadatog roka tajmera kod pošiljaoca, okvir će biti ponovo poslat i tako obesmisliti sistem potvrđivanja. Kada bi sloj veze mogao da predviđa budućnost, on bi znao kada da očekuje sledeći paket od mrežnog sloja i mogao bi da odluči da li da ga čeka ili da odmah pošalje zaseban okvir s potvrdom, u zavisnosti od toga koliki je zadati rok tajmera. Naravno, sloj veze to ne može, pa se mora prikloniti nekom *ad hoc* sistemu, na primer, da čeka određen broj milisekundi. Ukoliko nov paket pristigne brzo, potvrda se kači za njega; u suprotnom, ako paket ne stigne u tom periodu, sloj veze šalje zaseban okvir s potvrdom.

Sledeca tri dvosmerna protokola pripadaju klasi protokola kliznih prozora (engl. *sliding*

window). Među sobom se razlikuju po efikasnosti, složenosti i potrebnoj veličini bafera (o tome kasnije). Kod ovih protokola, kao kod svih protokola kliznih prozora, svaki odlazni okvir sadrži redni broj koji se krede od 0 do neke maksimalne vrednosti. Ta vrednost je obično 2" - 1, tako da redni broj tačno staje u polje širine *n* bitova. U protokolima kliznih prozora „stani i čekaj", nje jednako 1, što redne brojeve ograničava na vrednosti 0 i 1, ali se u složenijim verzijama može koristiti proizvoljno *n.*

Suština svih protokola kliznih prozora jeste to što pošiljalac u svakom trenutku čuva skup rednih brojeva okvira koje srne da pošalje. Oni predstavljaju tzv. prozor za slanje (engl. *sending window).* Slično tome, primalac održava prijemni prozor (engl. *receiving window)* koji odgovara skupu rednih brojeva okvira koje sme da primi. Prozori pošiljaoca i primaoca ne moraju da imaju iste granice (ni gornju, ni donju), niti da budu iste veličine. U nekim protokolima oni su konstantne veličine, ali u drugima mogu da se šire ili skupljaju u zavisnosti od toga kako se okviri šalju, odnosno primaju.

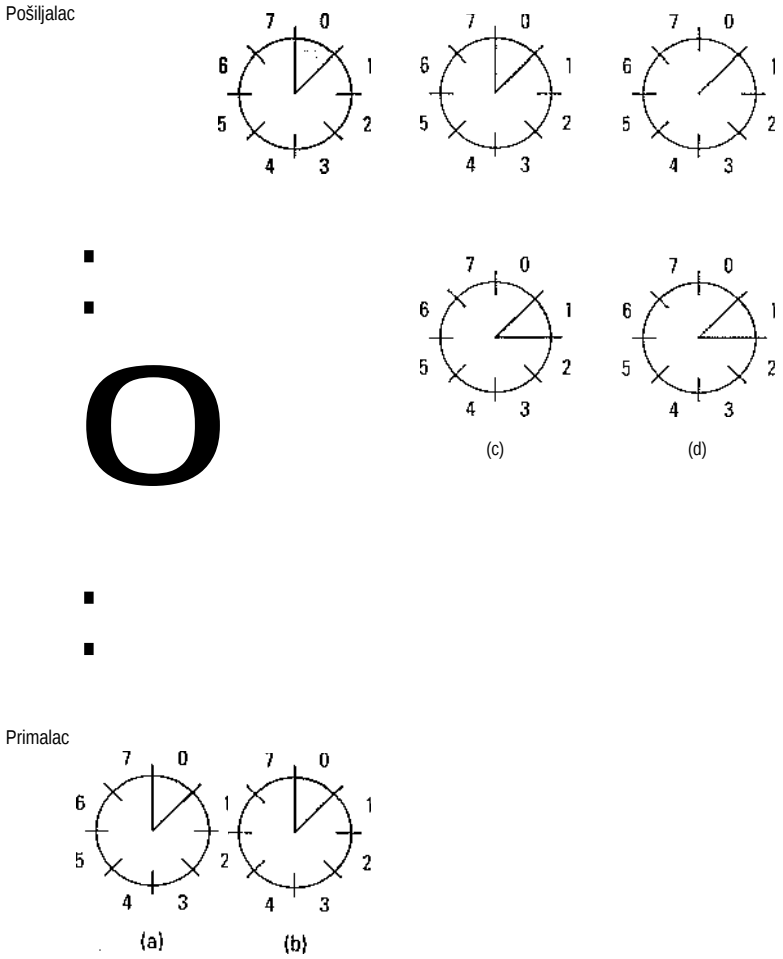
Iako ovi protokoli sloju veze daju više slobode u pogledu redosleda slanja i primanja okvira, mi se nismo potpuno odrekli zahteva da protokol mora da na odredišni mrežni sloj prosledi pakete istim redom kojim su prosleđeni sloju veze na izvorištu. Smatramo i dalje da fizički komunikacioni kanal radi kao žica, tj. da okvire isporučuje redosledom slanja.

Redni brojevi unutar pošiljaočevog prozora odgovaraju okvirima koji su poslati ili se mogu poslati, ali čiji prijem još nije potvrđen. Kad god nov paket stigne iz mrežnog sloja, dodeljuje mu se prvi najviši redni broj, a gornja granica prozora povećava se za jedan. Kada stigne potvrda, donja granica se povećava za jedan. Na taj način, prozor stalno održava listu nepotvrđenih okvira. Slika 3-13 prikazuje primer.

Pošto se okviri čiji se redni brojevi nalaze u prozora pošiljaoca mogu izgubiti ili oštetiti u prenosu, pošiljalac mora sve te okvire da čuva u memoriji da bi ih eventualno ponovo poslao. Tako, ako je maksimalna veličina prozora *n,* pošiljalac mora da ima *n* bafera za čuvanje nepotvrđenih okvira. Ukoliko prozor ikada dostigne svoju maksimalnu veličinu, sloj veze pošiljaoca mora nasilno da zatvori mrežni sloj dok se ne oslobodi neki bafer.

Prozor sloja veze primaoca sadrži redne brojeve okvira koje primalac sme.da primi. Svaki pristigli okvir koji je izvan ovog spiska, odbacuje se bez komentara. Kada stigne okvir čiji je redni broj jednak donjoj granici prozora, on se prosleđuje mrežnom sloju, generiše se potvrda i prozor rotira za jedinicu. Za razliku od prozora pošiljaoca, prozor primaoca uvek ima svoju početnu veličinu. Obratite pažnju na to da uz prozor veličine 1 sloj veze prihvata okvire redom, ali veći prozori ne moraju da rade tako. S druge strane, mrežni sloj pakete uvek dobija ispravnim redosledom, bez obzira na veličinu prozora sloja veze.

Slika 3-13 prikazuje primer prozora maksimalne veličine 1. Na samom početku, dok još nije poslat nijedan okvir, donja i gornja granica pošiljaočevog prozora se poklapaju, ali se vremenom slika menja.



Slika 3-13. Klizni prozor veličine 1, s rednim brojem veličine 3 bita. (a) Na početku, (b) Posle slanja prvog okvira, (c) Posle primanja prvog okvira, (d) Posle primanja prve potvrde.

3.4.1 **Jednobitni protokol kliznih prozora**

Pre nego što pređemo na opšti slučaj, pozabavimo se protokolom kliznih prozora maksimalne veličine 1. Takav protokol radi po principu „stani i čekaj“ jer pošiljalac šalje okvir i čeka potvrdu o njegovom prijemu pre nego što pošalje sledeći okvir.

Protokol je opisan na slici 3-14. Kao u svim protkolima, na početku se definišu neke

promenljive. *Next_frame_to_send* označava okvir koji pošiljalac pokušava da pošalje, a promenljiva *frame_expected.* odgovara okviru koji primalac očekuje. Vred- nosti obe promenljive mogu biti samo 0 ili 1.

```
/* Protokol 4 (klizni prozori) radi u oba smera. */

#define MAX_SEQ 1 /*
mora biti 1 za protokol 4 */
typedef enum {frame_arrival, cksum_err, timeout} event_type;
#include „protocol.h”
void protocol4 (void)

  seq_nr
  next_frame_to_send;
  seq_nr frame_expected;
  frame r, s; packet buffer;
  event_Jype event;

r samo 0 ili 1 */

  /* samo 0 ili 1 7 /*
  privremene promenljive 7 /*
  tekući paket se šalje 7
```

```
next_frame_to_send = 0;                    e r (& r. i nf o); inc(frame_expected); /*
frame_expected = 0;                        /* sledeći okvir u izlaznom toku 7 /*
from_networkJayer{&buffer)                 sledeći okvir koji se očekuje 7 /* uzmi
; s.info = buffer;                         paket iz mrežnog sloja 7 /* pripremi se za
s.seq =                                    slanje početnog okvira 7 /* unesi redni
next_frame_to_send; s.ack                  broj u okvir 7 /* šlepovana potvrda 7 /*
= 1 frame_expected;                        pošalji ovaj okvir 7 /* uključi tajmer 7
to_physicalJayer(&s);
start_timer(s.seq);                        /* frame_arrival, cksum_err ili timeout 7 /*
while (true) {                             okvir je stigao neoštećen, 7 /* uzmi ga 7
    wait_for_event(&event); if (event ==   /* obradi dolazni tok okvira. 7 /* prosledi paket
    frame_arrival) { f r om_physical       mrežnom sloju 7 invertuj vrednost rednog broja
    Jayer(&r); if (r.seq ==                očekivanog okvira 7
    frame_expected) { to_n et wo rkJ ay

        if (r.ack == next_frame_to_send) { /* obradi odlazni tok okvira. 7
            stop_timer(r.ack);             /*    isključi tajmer 7
            from_network_layer(&buffer); /* uzmi nov paket od mrežnog sloja 7
            inc(next_frame_to_send).; /* invertuj pošiljaočev redni broj 7
        }

    s.info = buffer;                       /* napravi okvir za slanje 7
    s.seq = next_frame_to_send;            /* unesi u njega redni broj 7
    s.ack = 1 frame_expected;              /* redni broj poslednjeg primljenog okvira 7
    to_physical Jayer(&s);                 /* pošalji okvir 7
    start_timer(s.seq);                    /* uključi tajmer 7
```

Slika 3-14. Jednobitni protokol kliznih prozora.

U normalnim okolnostima, jedan od dva sloja veze počinje protokol i šalje prvi okvir. Drugim recima, samo jedan od dva programa sloja vezetreba da sadrži pozive procedurama *to_physical_layer* i *startJimer* izvan glavne petlje. Ako oba sloja veze istovremeno započnu emitovanje, nastaje zanimljiva situacija koju ćemo kasnije razmotriti. Prvi računar uzima prvi paket od svog sloja veze, od njega pravi okvir i taj okvir šalje. Kada stigne ovaj (ili bilo koji drugi) okvir, sloj veze primaoca proverava da li je to duplikat nekog drugog okvira, baš kao u protokolu 3. Ako se pokaže daje to očekivani okvir, on se propušta mrežnom sloju, a prozor primaoca „klizi“ za jedan podelak.

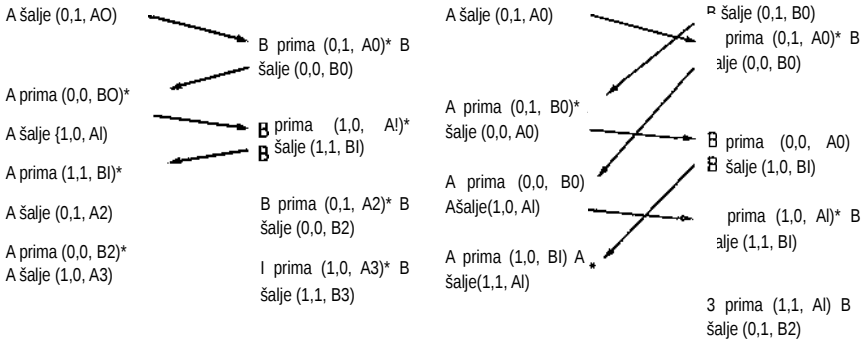
Polje *ack* sadrži redni broj poslednjeg okvira primljenog bez greške. Ako se taj broj slaže s rednim brojem okvira koji pošiljalac pokušava da pošalje, on zna da je završio posao sa okvirom koji čuva u *baferu* i može da uzme sledeći paket od svog mrežnog sloja. Ako se redni brojevi ne slože, on mora da produži sa slanjem istog okvira. Svaki put kada okvir bude primljen, drugoj strani se takođe šalje okvir.

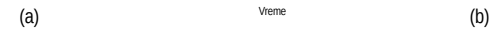
Pogledajmo sada protokol 4 da bismo utvrdili koliko je otporan na nepredviđene situacije. Pretpostavimo da računar *A* pokušava da pošalje okvir 0 računam *B,* a da računar *B* takođe pokušava da pošalje računaru *A* svoj okvir 0. Pretpostavimo da računar *A* šalje okvir računam *B,* ali da je rok tajmera računara *A* kradi nego što bi trebalo da bude. Shodno tome, tajmer računara A može više puta uzastopno da se automatski isključi i računar A *će* zato slati niz identičnih okvira *(seq* = 0 i *ack* = 1).

Kada prvi ispravan okvir stigne računara *B,* biče prihvaćen, a vrednost promenlji-*veframe_expected* postade 1. Svi naknadni (isti) okviri bide odbačeni jer *B* sada očekuje okvir s rednim brojem 1, a ne 0. Štaviše, pošto svi duplikati imaju *ack* = 1, a *B* još uvek čeka *ack* = 0, *B* nede uzimati nov paket od svog mrežnog sloja.

Nakon svakog pristiglog duplikata koji se odbacuje, računar *B* šalje računaru A okvir koji sadrži *seq* = 0 i *ack* = 0. Jedan od njih konačno stiže do A u ispravnom stanju, pa A počinje da šalje slededi paket. Nikakva kombinacija izgubljenih okvira i preranog isključenja tajmera ne može da natera protokol da isporuči duplikat mrežnom sloju, da preskoči neki paket ili da se zaglavi.

Specifična situacija nastaje kada obe strane istovremeno pošalju početni paket. Taj problem sinhronizacije prikazanje na slici 3-15. Njen deo (a) prikazuje normalan rad protokola, a deo (b) situaciju o kojoj govorimo. Ako *B* čeka prvi okvir računara A pre nego što pošalje svoj okvir, razvoj događaja sledi sliku 3-15(a) i svaki okvir se prihvata. Međutim, ako A i B istovremeno započnu komunikaciju, njihovi početni okviri *će* se mimoići na putu i slojevi veze podataka dolaze u situaciju prikazanu na slici 3-15(b). U situaciji (a) prispede svakog okvira donosi nov paket mrežnom sloju; duplikata nema. U situaciji (b) polovina okvira predstavlja duplikate, nezavisno od grešaka u prenosu. Slične situacije mogu nastati kada se tajmer isključuje prerano, čak i onda kada jedan od računara prvi započne komunikaciju. U stvari, ako se tajmer više puta prerano isključi, može se pojaviti tri i više duplikata okvira.





Slika 3-15. Dva scenarija protokola 4. (a) Normalna situacija, (b) Nenormalna situacija. Brojevi u zagradama označavaju seq, ack i redni broj paketa. Zvezdica označava trenutak kada mrežni sloj prihvata paket.

3.4.2 Protokol tipa „vrati se N“

Dosad smo prećutno pretpostavljali da putovanje okvira do primaoca i vraćanje potvrde pošiljaocu traje zanemarljivo kratko. Ponekada je, međutim, ta pretpostavka apsolutno neodrživa. U takvim situacijama dugo vreme putovanja s kraja na kraj veze može da ima ozbiljne posledice po iskorišćenje propusnog opsega. Razmotrite, na pri- mer, satelitski kanal brzine prenosa 50 kb/s čije je vreme prolaska vezom (tamo i natrag) 500 ms. Pretpostavimo da pomoću protokola 4 želite da šaljete okvire od 1000 bitova preko satelita. U trenutku $t = 0$ pošiljalac počinje da šalje prvi okvir. Nakon 20 ms, slanje okvir a je završeno. Kada prođe 270 ms, okvir još nije ceo stigao do primaoca, a potvrda o njegovom prijemu ne može stići pošiljaocu pre nego što u najboljem slučaju istekne 520 ms (bez čekanja kod primaoca i uz kratak okvir s potvrdom). To znači daje pošiljalac blokiran tokom 500 od 520 ms ili 96% vremena. Drugim recima, iskorišćava se samo 4% propusnog opsega. Jasno je daje kombinacija dugog vremena prenosa, velike propusne moći i male dužine okvira katastrofalna s gledišta efikasnosti.

Opisani problem može se smatrati posledicom strogog pravila koje nalaže da pošiljalac čeka potvrdu o prijemu paketa pre nego što pošalje sledeći paket. Ako to pravilo malo „olabavimo“, možemo okvire prenositi mnogo efikasnije. U osnovi, rešenje leži u tome da se pošiljaocu dozvoli da pre blokiranja pošalje *w* paketa umesto jednog. Uz pogodan izbor vrednosti veličine vv pošiljalac bi mogao neprekidno da šalje okvire tokom vremena potrebnog okviru za obilazak veze a da ne ispuni svoj prozor. U upravo navedenom primeru *w* bi trebalo da bude barem 26. Pošiljalac počinje tako što šalje okvir 0, kao i ranije. Nakon 520 ms, koliko mu treba da pošalje 26 okvira, upravo dobija potvrdu za nulti okvir. Nadalje, potvrde stižu svakih 20 ms, tako da pošiljalac uvek dobija dozvolu da nastavi sa slanjem upravo kada mu takva dozvola zatreba. U svakom trenutku na vezi postoji 25 ili 26 nepotvrđenih paketa. Ako to prevedemo na-jezik protokola, maksimalna veličina prozora kod pošiljaoca treba daje 26.

Pošiljalac uvek treba da ima veliki prozor ako je velika vrednost proizvoda propusnog opsega i vremena obilaska veze. Kada je propusni opseg veliki, pošiljalac će čak i uz umereno kašnjenje signala brzo popuniti prozor ukoliko nije dovoljno veliki. Ako je vremensko kašnjenje veliko (na primer, na vezi preko geostacionarnog satelita), pošiljalac će iscrpeti svoj prozor čak i uz umeren propunsi opseg. Proizvod dva navedena činioca govori o stvarnom kapacitetu bežične cevi i pošiljalac mora da bude u stanju da ga neprestano ispunjava okvirima kako bi radio maksimalno efikasno.

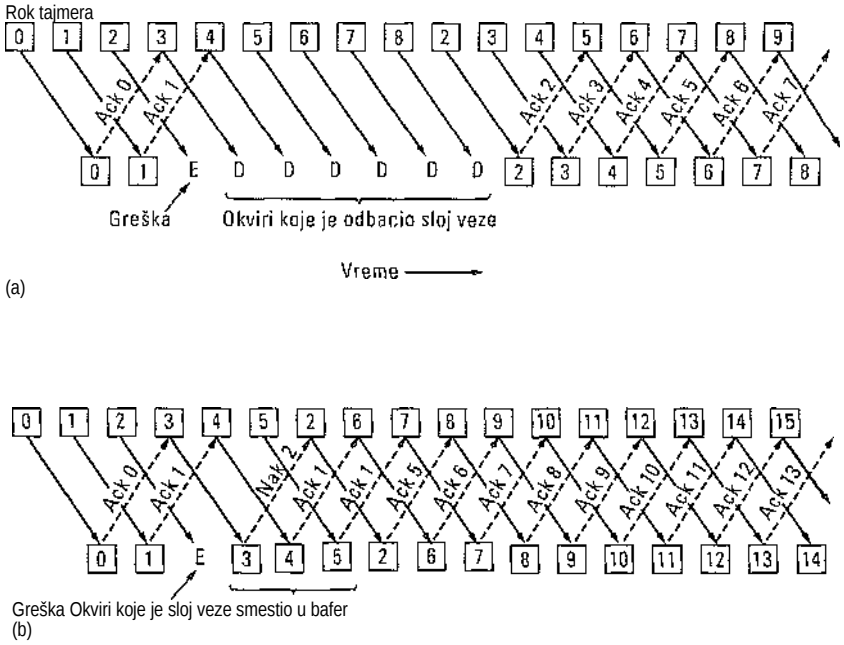
Opisana tehnika poznata je kao protočno slanje podataka (engl. *pipelining).* Ako je kapacitet kanala *b* b/s, veličina okvira / bitova, a ukupno vreme prolaska vezom (tamo i natrag) *R* s, vreme potrebno za prenos jednog okvira je *l/b* s. Nakon što se pošalje i poslednji bit okvira, on putuje do primaoca tokom *R/2* s, a zatim se potvrda vraća pošiljaocu tokom najmanje *R/2 s,* pa je ukupna zadrška R. U protokolu tipa „stani i čekaj“, linija je zauzeta tokom *l/b* s, a ništa ne radi tokom *R* s, odakle proizlazi da je

$$\text{iskorišćenje linije} = //(/ + bR)$$

Ako je $l < bR$, efikasnost će biti niža od 50%. Pošto uvek postoji vremensko kašnjenje potvrde o poslatom paketu, u načelu se može iskoristiti protočno slanje da bi se linija tokom ovog intervala koristila, ali ako je interval mali, dodatno kompliko- vanje se ne isplati.

Protočno slanje okvira nepouzdanim komunikacionim kanalom otvara neka ozbiljna pitanja. Najpre, šta se dešava ako se okvir koji se nalazi negde u sredini „rafala“ ošteti ili izgubi? Primaocu ce stidi mnogo okvira koji ga slede pre nego što pošiljalac uopšte utvrdi da nešto nije

u redu. Kada oštećen okvir stigne primaocu, očigledno je da ga treba odbaciti, ali šta primalac treba da radi sa ispravnim okvirima koji ga slede? Setite se da sloj veze primaoca mora da mrežnom sloju isporučuje pakete ispravnim redosledom. Na slici 3-16 vidimo uticaj protočnog slanja paketa na oporavljanje od grešaka. Razmotrićemo ga sada detaljnije.



Slika 3-16. Protočno slanje okvira i oporavljanje od grešaka. Uticaj greške kada je prozor primaoca (a) veličine 1 i (b) veliki, Ack - Potvrda; Nak - Nepotvrđeno.

Za obradu grešaka pri protočnom slanju podataka moguća su dva pristupa. Jedan, poznat pod nazivom „vrati se n“ (engl. *go back n),* podrazumeva da primalac, kada dobije pogrešan okvir (ili okvir uopšte ne dobije), sve sledeće okvire odbaci ne šaljući za njih potvrdu. Toj strategiji odgovara prozor primaoca veličine 1. Dragim recima, sloj veze odbija da prihvati svaki okvir različit od onog koji upravo treba da prosledi mrežnom sloju. Ako se prozor pošiljaoca ispuni pre nego što istekne rok tajmera, veza će početi da se prazni. Na kraju će se pošiljaočev tajmer isključiti i pošiljalac će ponovo redom poslati sve nepotvrđene okvire, počinjući od oštećenog ili izgubljenog okvira. Opisani pristup prilično angažuje propusni opseg ako je učestalost grešaka visoka.

Na slici 3-16(a) vidimo u radu tehniku „vrati se n“ za slučaj kada je prozor primaoca veličine 1. Okvir 0 i okvir 1 ispravno su primljeni i o tome je poslata potvrda. Okvir 2 je, međutim, oštećen ili izgubljen. Pošiljalac, nesvestan problema, nastavlja sa slanjem sve dok ne istekne rok tajmera za okvir 2. Zatim se vraća na okvir 2 i ponovo šalje okvire 2, 3, 4 itd.

Drugi pristup obradi grešaka pri protočnom slanju podataka naziva se selektivno ponavljanje (engl. *selective repeat).* Ovde se neispravan okvir odbacuje, ali se ispravni okviri koji ga slede smeštaju u bafer. Kada istekne rok tajmera pošiljaoca, pošiljalac šalje samo najstariji nepotvrđeni okvir. Ako on na odredište stigne u ispravnom stanju, primalac može da

mrežnom sloju isporuči redom i sve okvire iz bafera. Selektivno ponavljanje često se dopunjava i time što primalac pošiljaocu šalje negativnu potvrdu (NAK) kada otkrije grešku, na primer, ako se kontrolni zbir ne složi ili ako primi okvir izvan očekivanog redosleda. Signal NAK pokreće ponovno slanje i pre isteka roka tajmera i time poboljšava performanse.

Na slici 3-16(b), okviri 0 i 1 i ovde su primljeni ispravno, o čemu je poslata potvrda, a okvir 2 se izgubio. Kada okvir 3 stigne primaocu, sloj veze primećuje daje propustio jedan okvir, pa pošiljaocu šalje signal NAK za okvir 2, ali okvir 3 smešta u bafer. Kada stignu okviri 4 i 5, i njih sloj veze smešta u bafer umesto da ih prosledi mrežnom sloju. Konačno, signal NAK 2 stiže pošiljaocu i on odmah ponovo šalje okvir 2. Kada on stigne, sloj veze će imati niz okvira 2, 3, 4 i 5, pa može sve da ih prosledi mrežnom sloju ispravnim redosledom. On može i da pošalje potvrdu za sve okvire, uključujući okvir 5, kao što je prikazano na slici. Ako se okvir NAK izgubi, isteći će rok tajmera za okvir 2 i pošiljalac će ga samoinicijativno ponovo poslati (i samo njega), ali se to može dogoditi s priličnim zakašnjenjem. U stvari, signal NAK ubrzava ponovno slanje određenog okvira.

Selektivnom ponavljanju odgovara da prozor primaoca bude veći od 1. Svaki okvir u prozoru može da bude prihvaćen i smešten u bafer sve dok se prethodni okviri ne proslede mrežnom sloju. Za takav pristup možda je potrebna velika memorija sloja veze ukoliko je prozor veliki.

Od dva opisana alternativna pristupa jedan štedi propusni opseg, a drugi memoriju sloja veze. Može se upotrebiti jedan ili drugi, u zavisnosti od toga koji resurs je „tanji“. Slika 3-17 prikazuje protokol za protočno slanje podataka u kome sloj veze primaoca prihvata okvire samo utvrđenim redom; okviri koji stižu iza neispravnog ili izgubljenog okvira odbaciju se. U ovom protokolu smo po prvi put odustali od pretpostavke da mrežni sloj u svakom trenutku ima pakete za slanje. Kada mrežni sloj stvarno ima paket koji bi poslao, on izaziva događaj *network Jayerjready.* Međutim, da bi sproveo pravilo kontrole toka, po kome u bilo kom trenutku ne sme biti više od *MAXJSEQ* okvira za koje nije stigla potvrda, sloj veze mora umeti da obuzda mrežni sloj. Zbog toga on po potrebi aktivira, odnosno deaktivira mrežni sloj koristeći procedure *enable jietwork Jayer* i *disable_networkJayer.*

/* Protokol 5 (vrati se n) omogućava da vezom istovremeno putuje više okvira. Pošiljalac može da pošalje jedan za drugim do MAX_SEQ okvira ne čekajući potvrdu. Osim toga, za razliku od prethodnih protokola, više se ne pretpostavlja da mrežni sloj uvek ima paket

```
#define MAX_SEQ 7
```

spreman za slanje, već mrežni sloj - kada ga stvarno ima - izaziva događaj network_layer_ready. 7

I" treba da bude 2"n -1 7

typedef enum {frame_arrival, cksum_err, timeout, networkJayer_ready} eventjype; #include „protocol.h"

```
static boolean between(seq_nr a, seq_nr b, seq_nr c)
{
/* Vrati true ako ciklično važi a <=b < c; u suprotnom, vrati false. 7 if (((a <= b) && (b < c)) ||
((c < a) && (a <= b)) || ((b < c) && (c < a))) return(true); else return (false);
}
static void send_data(seq_nr frame_nr, seq_nr frame_expected, packet bufferj ])
{
/* Napravi i pošalji okvir podataka. 7 frame s;   /* privremena promenljiva 7

  s.info = buffer[frame_nrj;                    /* unesi paket u okvir 7
  s.seq = frame_nr;                             /* unesi redni broj u okvir 7
  s.ack = (frame_expected + MAX_SEQ) % (MAX_SEQ + 1); /* šlepuj potvrdu 7
  to_physical_layer(&s);                        /* pošalji okvir 7
  start_timer(frame_nr);                        /* uključi tajmer 7


void protocol5(void)

seq_nr next_frame_to_send;          /* MAX_SEQ > 1; koristi se za odlazni tok */
seq_nr ack_expected;                /* poslednji potvrđen okvir */
seq_nr frame_expected;              /* sledeći okvir koji se očekuje u dolaznom toku
frame r;                            */ /* privremena promenljiva */
packet buffer[MAX_SEQ +             /* baferi za odlazni tok 7
1]; seq_nr nbuffered; seq_nr        /* broj izlaznih bafera koji se trenutno koriste 7
i;                                  /* koristi se za indeksiranje niza bafera 7
event_type event;

enable_network_layer();             sledeći okvir koji se šalje 7 /* broj očekivanog
ack_expected = 0;                   dolaznog okvira 7 /* na početku nema paketa u
next_frame_to_send =                baferu 7
0; frame_expected = 0;
nbuffered = 0;
/* omogućava događaje
network_layer_ready 7 /*
sledeća očekivana
dolazna potvrda 7 /*
```

```
while (true) {

  wait_for_event(&event);                    /*   četiri mogućnosti: pogledajte goreevent_type */

  switch(event) {
    case network_layer_ready:                /*   mrežni sloj ima paket za slanje */
         /* Prihvatanje, memorisanje i slanje novog okvira. */
         from_network_layer(&buffer[next_frame_to_send]); /* uzmi nov paket */
         nbuffered = nbuffered + 1;          /*   proširi prozor pošiljaoca */
         send_data(next_frame_to_send, frame_expected, buffer);/* pošalji okvir */
         inc(next_frame_to_send);            /*   pomeri gornju granicu prozora pošiljaoca
                                             */
         break;

    case frame_arrival:              /* stigao    jeokvir s korisničkim ili upravljačkim podacima */
         from_physical_layer(&r);            /*   uzmi pristigli okvir iz fizičkog sloja */

         if (r.seq == frame_expected) {
              /* Okviri se prihvataju samo ispravnim redosledom. */
              to_network_layer(&r.info); /* prosledi paket mrežnom sloju */
              inc(frame_expected); /* pomeri donju granicu prozora primaoca */
         }

         /* Ack n podrazumeva n -1, n - 2 itd. Proveriti ovo. */ while
         (between(ack_expected, r.ack, next_frame_to_send))}
              /* Obradi šlepovanu potvrdu. */
              nbuffered = nbuffered -1; /* u baferu je jedan okvir manje */
              stop_timer(ack_expected); /* okvir je stigao ispravan; zaustavi tajmer */
              inc(ack_expected);             /*   suzi prozor pošiljaoca */
         }
         break;

    case cksum_err: break;                   /*   zanemari neispravne okvire */

    case timeout:                /*    problem; pošalji ponovo sve okvire za koje nije stigla potvrda */
         next_frame_to_send = ack_expected; /* počni s ponovnim slanjem odavde */
         for (i = 1; i <= nbuffered; i++) {
            send_data(next_frame_to_send, frame_expected, buffer);/* ponovo pošalji okvir */
            inc(next_frame_to_send); /* pripremi se da pošalješ sledeći */
         }

    }

    if (nbuffered < MAX___SEQ)
              enable_networkJayer();
    else
              disable_network_layer();
 }
}
```

Slika 3-17. Protokol kliznih prozora koji koristi pravilo „vrati se n“

Obratite pažnju na to da na vezi istovremeno može biti samo *MAX_SEQ* okvira, a ne *MAX__SEQ + l,* iako postoji *MAX_SEQ + 1* različitih rednih brojeva (0, 1, 2,..., *MAX_SEQ).*

Da biste razumeli zasto je potrebno takvo ograničenje, razmotrite slučaj kada je *MAX_SEQ* = 7.

1. Pošiljalac šalje okvire od 0 do 7.
2. Slepovana potvrda o prijemu okvira 7 na kraju stiže pošiljaocu.
3. Pošiljalac šalje sledećih osam okvira, ponovo s rednim brojevima od 0 do 7.
4. Sada ponovo dolazi šlepovana potvrda o prijemu okvira 7.

Postavlja se pitanje: da lije svih osam okvira iz drugog slanja uspešno stiglo ili su se svih osam izgubili (računajući odbacivanje pogrešnog okvira kao gubitak)? Primalac će u oba slučaja poslati potvrdu o prijemu okvira 7, a pošiljalac nema načina da razlikuje ove dve situacije. Iz tog razloga, maksimalan broj okvira koji su istovremeno na vezi mora da bude ograničen na *MAX_SEQ,*

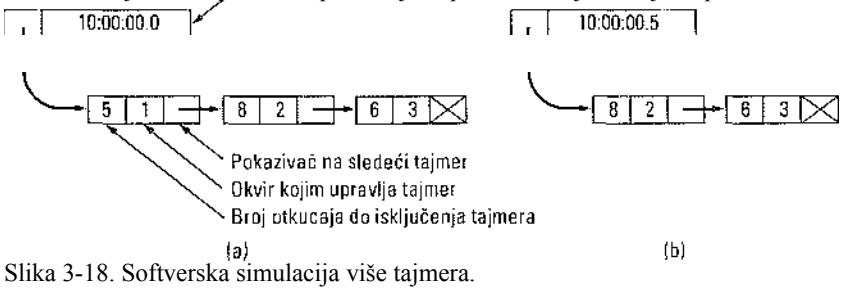
Iako se prema protokolu 5 okviri koji stignu posle nastanka greške ne smeštaju u bafer, on ne zaobilazi u potpunosti problem bafera. Pošto pošiljalac u budućnosti možda treba ponovo da pošalje sve nepotvrđene okvire, on mora da misli o njima sve dok ne bude potpuno siguran da ih je primalac ispravno dobio. Kada pristigne potvrda za okvir *n,* automatski se potvrđuje i prijem okvira *n - l , n - 2* itd. Takvo svojstvo je izuzetno važno jer se neki od prethodnih okvira s potvrdom može izgubiti ili oštetiti u putu. Kad god stigne potvrda o prijemu okvira, sloj veze proverava može li da oslobodi neki bafer. Ako uspe da oslobodi neke bafere (tj. da napravi prostor u prozoru), tada se blokiranom mrežnom sloju može dozvoliti da nastavi sa izazivanjem događaja *network_Jciyer_ready.*

U ovom protokolu pretpostavljamo da uvek postoji povratni saobraćaj okvira kojima se mogu šlepovati potvrde. Ako takvog saobraćaja nema, potvrde se ne mogu slati. U protokolu 4 takva pretpostavka nije bila neophodna, pošto se u njemu za svaki pristigli okvir povratno šalje drugi okvir, čak i kada je takav okvir upravo poslat. U sledećem protokolu rešićemo problem jednosmernog saobraćaja na elegantan način.

Pošto kod protokola 5 na vezi može istovremeno da bude više okvira, naravano da je za njih potrebno i više tajmera, po jedan za svaki okvir koji je na putu. Tajmer za svaki okvir radi nezavisno od drugih tajmera. Tajmeri se mogu lako simulirati softverski, lcorišćenjem jedinstvenog hardverskog sata koji periodično emituje prekide. Tajmeri koji su još uvek aktivni obrazuju povezanu listu, čiji svaki čvor sadrži broj otkucaja sata potrebnih da se odgovarajući tajmer automatski isključi, broj okvira kojim upravlja tajmer i pokazivač na sledeći čvor.

Primer prikazan na slici 3-18(a) predstavlja jedan od načina ugradnje tajmera. Pretpostavimo da sat otkucava svakih 100 ms. Na početku je stvarno vreme 10:00:00,0 sati; trenutno su tri tajmera aktivna; automatski treba da se isključe u 10:00:00,5, 10:00:01,3 i 10:00:01,9 sati. Svaki put kad otkuca hardverski sat, stvarno vreme se ažurira i brojač otkucaja u zaglavlju liste smanjuje svoju vrednost za jedan. Kada vrednost brojača dostigne nulu, generiše se prekid i čvor se skida s liste, kao na slici 3-18(b). Iako zbog ovakve organizacije lista treba da se skenira svaki put kada se

pozove procedura *start Jimer* ili *stopjimer,* to ne zahteva mnogo posla između dva otkucaja sata. U protokolu 2, svakoj od ovih procedura pridružuje se parametar koji ukazuje na predmetni okvir.



Slika 3-18. Softverska simulacija više tajmera.

3.4.3 Protokol sa selektivnim ponavljanjem

Protokol 5 radi dobro ako su greške retke, ali ako je linija nekvalitetna, on troši veliki deo propusnog opsega na ponovno slanje okvira. Alternativna strategija obrade grešaka omogućava primaocu da prihvati i smesti u bafer okvire koji slede iza oštećenog ili izgubljenog okvira. Po tom protokolu, okviri se ne odbacuju samo zato što je neki raniji okvir oštećen ili nestao.

Prema ovom protokolu, i pošiljalac i primalac održavaju prozore s prihvatljivim rednim brojevima okvira. Veličina pošiljaočevog prozora počinje od 0 i raste do nekog unapred definisanog maksimuma *MAX_SEQ*. Prozor primaoca, suprotno tome, ima uvek istu veličinu *MAX_SEQ*. Primalac rezerviše bafer za svaki redni broj okvira unutar prozora fiksne veličine. Svakom baferu je pridružen bit *arrived* (stigao) koji ukazuje na to da li je bafer pun ili prazan. Kad god stigne okvir, njegov redni broj pro- verava funkcija *between* (između) i utvrđuje da li se nalazi u granicama prozora. Ako je rezultat provere pozitivan i ako okvir nije već bio prihvaćen, on se prihvata sada i smešta u bafer. Ova akcija se preduzima bez obzira na to da li bafer već sadrži sledeći paket koji očekuje mrežni sloj. Naravno, okvir se mora čuvati unutar sloja veze sve dok se prethodni okviri ne isporuče mrežnom sloju ispravnim redom. Protokol sastavljen prema opisanom algoritmu prikazan je na slici 3-19.

```
r Protokol 6 (selektivno ponavljanje) prihvata okvire preko reda, ali pakete mrežnom sloju
prosleđuje redom. Svakom okviru koji je na vezi pridružuje se tajmer. Kada rok tajmera istekne,
ponovo se šalje samo taj okvir, a ne svi okviri koji su istovremeno na vezi, kao u protokolu 5. */

#define MAX_SEQ 7                                   /* treba da bude 2"n - 1 */
#define NRJ3UFS ((MAX_SEQ + 1)/2)
typedef enum {frame_arrival, cksum_err, timeout, networkJayer_ready, ackjimeout}
event_type;
#include „protocol.h"

                                          /*
boolean no_nak = true;                              još  uvek nije poslata nijedna negativna potvrda */
seq__nr oldesfjrame = MAX_SEQ + 1;                  /* početna vrednost važi samo za simulator */
```

```
static boolean between(seq_nr a, seq_nr b, seq_nr c)
{
/* Isto kao procedura between u protokolu 5, ali kraća i manje jasna. 7
return ((a <= b) && (b < c)) || ((c < a) && (a <= b)) || ((b < c) && (c <
a));
}
static void sendjrame(frame_kind fk, seq_jir frame_nr, seq__nr frame_expected, bufferj ] za
pakete)
{
/* Napravi i pošalji okvir s podacima, pozitivnom ili negativnom potvrdom. 7 frame s;
```

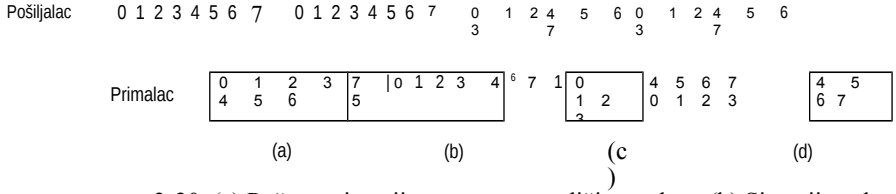
```
privremena prbmenljiva 7

 s.kind = fk;                                   /*    kind == data, ack, ili nak 7
 if (fk == data) s.info = buffer[frame_nr % NRJ3UFS];
 s.seq = frame_nr;                               /* ima smisla samo za okvire s podacima 7
 s.ack = (frame_expected + MAX_SEQ) % (MAX_SEQ + 1);
 if (fk == nak) no_nak = false;                 /*    molim, jedan signal nak po okviru 7
 to__physical_layer(&s)                         /*    pošalji okvir 7
 if (fk == data) startJimer(frame_nr % NRJ3UFS);
 stop_ackJimer();                                                                    /*
                                        nema potrebe za posebnim okvirom s potvrdom 7
}
void protocol6(void)
{
 seq_nr ack_expected;                           /*    donja granica prozora pošiljaoca 7
 seq_nr next_frame_to_send;                     /*    gornja granica prozora     pošiljaoca +    1
 7
 seq_nr frame_expected;                          /* donja granica prozora primaoca 7
 seq_nr too_far;                                                                     /*gornja
 granica prozora primaoca + 1 7
 int i;                                                          /*      pokazivač bafera 7
 frame r;                                                         /*privremena
 promenljiva 7
 packet out_buf[NR_BUFSj;                       /*    baferi za tok podataka koji se šalju7
 packet in_buf[NR_BUFS];                        /*    baferi za tok podataka koji stižu 7
 boolean arrived[NR_BUFSj;                      /* bit mapa pristiglih okvira 7
 seq_nr nbuffered;                                                                   /*koliko
 se odlaznih bafera trenutno koristi 7
 eventjype event;

 enablejnetworkJayer();                         /*  inicijalizacija 7
 ack_expected = 0;                      /*sledeća potvrda koja se očekuje u ulaznom toku */
 next_frame_to_send = 0;                        /*     redni broj sledećeg okvira koji se šalje7
 frame__expected =
 0, too Jar =
 NRJ3UFS;
 nbuffered = 0;                                                                      /*na  početku
 nema paketa u baferu                                                                 7
 for (i = 0; i < NRJ3UFS; i++) arrivedji] = false;
```

```
while (true) {
  waitJor_event(&event);                    /* pet    mogućnosti: pogledajte gore eventjype 7
  switch (event) {
    case networkJayer_ready:                /*    prihvati,      smestiumemorijui
                                            pošalji nov okvir 7
        nbuffered = nbuffered + 1;          /* proširi prozor 7
        from_networkJayer(&out_buf[nextJrameJo_send % NRJ3UFS]); /* uzmi nov
                                                                    paket */
        sendjrame(data, nextJrameJo_send, frame_expected, outjouf);/* pošalji okvir 7
        inc(nextJrameJo_send);              /*    pomeri gornju granicu prozora 7
        break;
  case frame_arrival:                       /* stigao     jeokvir     s          podacima ili
  upravljački okvir 7
      from_physical_layer(&r);              /*          uzmi     dolazni okvir iz fizičkog sloja 7
      if (r.kind == data) {

          /* Stigao je neoštećen okvir. 7 if ((r.seq != frame_expected) &&
          no_nak) send_frame(nak, 0, frame_expected, out_buf); else start_ack_timer(); if
          (between(frame_expected,r.seq,too_far) && (arriveđ[r.seq%NR_BUFS]==false)) {

              I* Okviri se mogu primati bilo kojim redom. 7 arrived[r.seq %
              NRJ3UFS] = true; /* označi bafer kao pun 7 in_buf[r.seq %
              NR_BUFS] = r.info; /* unesi podatke u bafer 7 while
              (arriveđ[frame_expected % NRJ3UFS]) {
                 /* Prosledi okvire i pomeri se u prozoru. 7
                 to_networkJayer(&in_buf[frame_expected % NR_BUFS]);
                 hojnak = true;
                 arrived[frame_expected % NR_BUFS] = false; inc(frame_expected); /*
                 pomeri donju granicu prozora primaoca 7
                 inc(too_far);                  /*pomerigornju  granicu  prozora  primaoca
                 7
                 start_ackjimer(); /* kako bi utvrdio da li je potrebna posebna potvrda 7
              }
          }
      }
      if((r.kind==nak) && between(ack__expected,(r.ack+1)%
          (MAX_SEQ+1 ),next_frame_Jo_senđ))
          send_frame(data, (r.ack+1) % (MAX_SEQ + 1), frame_expected, out_buf);

      while (between(ack_expected, r.ack, next_frame_to_send)) {
          nbuffered = nbuffered . 1;        /* obradi  šlepovanu potvrdu 7
          stop_timer(ack_expected % NR_BUFS); /* okvir je stigao netaknut 7
          inc(ack_expected);                /*pomeri  donju granicu prozora pošiljaoca 7
      }
      break; case

  cksum_err:
      if (nojiak) send_frame(nak, 0, frame_expected, out_buf); /* oštećen okvir 7
      break;
```

```
    case timeout:
        sendjrame(data, oldest_frame, frarne_expected, outjoiif); /* automatski
                                                                    smo se isključili 7
        break; case
    ack_timeout:
        send_frame(ack,0,frame_expected, out_buf); /* tajmer za pristizanje potvrde
                                                    automatski se isključio;
                                                    pošalji potvrdu 7
  }
  if (nbuffered < NRJ3UFS) enable_network_layer(); else disable_networkJayer();
}
```

Slika 3-19. Protokol kliznih prozora sa selektivnim ponavljanjem.

Prekoredno primanje okvira izaziva određene probleme kojih nema kod protokola koji okvire prihvataju samo ispravnim redosledom. Te probleme najlakše možemo objasniti najednom primeru. Pretpostavimo da imamo 3-bitni redni broj, tako da pošiljalac može da pošalje najviše sedam okvira pre nego što dobije prvu potvrdu. Na početku prozori pošiljaoca i primaoca izgledaju kao na slici 3-20(a). Pošiljalac zatim šalje okvire 0 do 6. Prozor primaoca dozvoljava prihvatanje bilo kog okvira s rednim brojem između 0 i 6. Svih sedam okvira stiže ispravno, pa primalac o tome šalje potvrde i po- mera svoj prozor da bi mogao da prihvati okvire 7, 0, 1, 2, 3, 4 ili 5, kao što prikazuje slika 3-10(b). Svih sedam bafera označavaju se kao prazni.



Slika 3-20. (a) Početna situacija s prozorom veličine sedam, (b) Situacija nakon što je sedam okvira poslato i primljeno, ali potvrde još nisu poslate, (c) Početna situacija s prozorom veličine četiri, (d) Situacija nakon što su četiri prozora poslata i primljena, ali potvrde još nisu poslate.

U tom trenutku dolazi do nepredviđene situacije - grom udara u telefonski stub i briše sve potvrde. Tajmer pošiljaoca se na kraju automatski isključuje i pošiljalac ponovo šalje okvir 0. Kad taj okvir stigne primaocu, proverava se može li da bude prihvaćen u prozor. Nažalost, okvir 0 spada u okvire koje nov prozor primaoca može da prihvati, pa će i biti prihvaćen. Primalac šalje šlepovanu potvrdu za okvir 6, pošto je primio okvire od 0 do 6.

Pošiljalac se raduje što su svi okviri koje je poslao stigli ispravno na odredište, pa pomera svoj prozor i odmah šalje okvire 7,0,1,2, 3,4 i 5. Primalac će prihvatiti okvir 7 i njegov paket odmah proslediti mrežnom sloju. Neposredno posle toga, sloj veze primaoca proverava da li već ima ispravan paket 0, utvrđuje da ga ima i prosleđuje ugrađeni paket mrežnom sloju. Dakle, mrežni sloj ne dobija pravi paket zbog nedostataka sadržanih u samom protokolu.

Suština problema je u tome da kada primalac promeni granice svog prozora, nove vrednosti rednih brojeva prebrišu stare. Zbog toga sledeći skup okvira mogu da budu duplikati (ako su sve potvrde nestale u putu) ili novi okviri (ako su sve potvrde stigle pošiljaocu). Primalac nema načina da razlikuje te dve situacije.

Rešenje problema je u tome da nove granice prozora primaoca (po prihvatanju svih okvira) ne treba da se preklapaju sa starim granicama. To se može udesiti ako se veličina prozora ograniči na najviše polovinu opsega rednih brojeva, kao na slikama 3-20(c) i 3-20(d). Na primer, ako se za redne brojeve koristi 4 bita, oni će se protezati između 0 i 16. Na vezi tada sme istovremeno da bude samo osam nepotvrđenih okvira. Na taj način, ako je primalac upravo dobio okvire 0 do 7 i pomerio granice prozora da bi prihvatio okvire 8 do 15, on nedvosmisleno može da utvrdi da li su okviri koje

posle toga dobije ponovno poslati (0 do 7) ili predstavljaju nov skup okvira (8 do 15). U načelu, veličina prozora u protokolu 6 biće (*MAX_SEQ* + l)/2. Prema tome, za
3- bitne redne brojeve, veličina prozora biće 4.

Postavlja se zanimljivo pitanje: koliko bafera treba da ima primalac? On ni pod kojim uslovima ne srne da prihvati okvir čiji je redni broj manji od donje granice prozora, niti okvir čiji je redni broj veći od gornje granice prozora. Prema tome, broj potrebnih bafera jednak je veličini prozora, a ne rasponu rednih brojeva. U navedenom primeru 4-bitnih rednih brojeva, potrebno je 8 bafera, numerisanih od 0 do 7. Kada stigne okvir *i,* on se smešta u bafer pod brojem *(i* mod 8). Obratite pažnju na to da se okvir *i* i (7 + 8) mod 8 nadmeću za isti bafer, ali se nikada ne mogu naći u istom prozoru - za to je potreban prozor veličine barem 9.

Iz istog razloga, broj potrebnih tajmera jednak je broju bafera, a ne rasponu rednih brojeva. Aktivni tajmer je efektivno povezan sa određenim baferom. Kada rok tajmera istekne, sadržaj bafera se ponovo šalje.

Kod protokola 5 podrazumevalo se da je kanal prilično opterećen. Potvrda o pristiglom okviru nije slata pošiljaocu odmah, već je šlepovana sa sledećim odlaznim okvirom podataka. Ako je saobraćaj u tom smeru bio slab, pošiljalac je mogao dugo da čeka potvrdu. Daje saobraćaj u jednom smem bio veliki, a u suprotnom nepostojeći, bilo bi poslalo samo *MAX_SEQ* paketa, a tada bi se protokol blokirao, zbog čega smo morali pretpostaviti da uvek postoji i saobraćaj u suprotnom smem.

U protokolu 6 opisani problem je rešen. Pošto pristigne jedan okvir iz sekvence, aktivira se pomoćni tajmer procedurom *start_ack_timer.* Ako se tokom aktivnosti tajmera u suprotnom smeru ne pošalje nijedan okvir s podacima, tada se šalje poseban okvir s potvrdom. Prekid izazvan automatskim isključivanjem tog tajmera predstavlja događaj *ackjimeout.* Na taj način je moguć i rad u situaciji kada se saobraćaj odvija samo u jednom smeru jer nedostatak okvira s podacima za koje bi se šlepovale potvrde više ne predstavlja problem. Postoji samo jedan pomoćni tajmer *start_ackjimer*\ ako se on pozove dok je aktivan, njegovo vreme počinje da teče od početka.

Suštinski je važno da rok automatskog isključivanja pomoćnog tajmera bude znatno kraći od roka isključivanja tajmera koji se koriste za dočekivanje potvrda. To je neophodno da bi ispravno primljen okvir mogao biti dovoljno rano potvrđen - pre nego što kod pošiljaoca istekne rok za prijem potvrde, inače će taj okvir biti ponovo poslat.

U protokolu 6 koristi se mnogo efikasnija strategija rada s greškama, nego u protokolu 5. Kad god primalac posumnja daje došlo do greške, on može pošiljaocu da pošalje negativnu potvrdu (NAK) - okvir sa zahtevom za ponovno slanje naznačenog okvira. Postoje dva slučaja koji kod primaoca mogu da pobude sumnju: okvir je stigao oštećen ili je stigao okvir koji nije očekivan (moguć gubitak okvira). Da bi izbe- gao ispostavljanje više zahteva za ponovno slanje istog izgubljenog okvira, primalac mora voditi evidenciju o tome da li je za određeni okvir poslat signal NAK. Pro- menljiva *no_nak* u protokolu 6 ima vrednost true ako još nije poslat NAK za očekivani okvir *(frame_expected*). Ako se NAK ošteti ili izgubi, nema štete, pošto će se tajmer pošiljaoca konačno automatski isključiti i izazvati ponovno slanje nedostajućeg okvira. Ako po slanju, a zatim oštećavanju ili gubitku okvira NAK, primalac dobije pogrešne okvire, *no_nak* će imati vrednost true i pomoćni tajmer će se aktivirati.
Kada mu istekne rok, primalac će poslati signal ACK da bi pošiljaoca sinhronizovao sa svojim trenutnim statusom.

U izvesnim slučajevima, vreme koje je okviru potrebno da stigne na odredište, da tamo bude obrađen i da se potvrda o njegovom prijemu vrati pošiljaocu, skoro je konstantno. Tada pošiljalac može rok svog tajmera da podesi na rok koji je nešto duži od vremenskog intervala između slanja okvira i očekivanog stizanja potvrde. Međutim, ako je ovaj interval veoma promenljiv, pošiljalac je suočen sa izborom da ga podesi na malu vrednost (rizikujuci slanje nepotrebnih okvira) ili na veliku (i tako bude dugo bez posla nakon što dođe do neke greške u prenosu).

Propusni opseg se u oba slučaja neefikasno koristi. Ako je povratni saobraćaj sporadičan, vreme pristizanja potvrde bide neujednačeno: krade kada povratni saobraćaj postoji, duže kada ga nema. Ovde može da postane problem i promenljivo trajanje obrade okvira kod primaoca. U načelu, uvekkada je standardno odstupanje intervala u kome stigne potvrda malo u odnosu na srednju vrednost tog intervala, tajmer se može podesiti tesno, a signali NAK su od male koristi. U suprotnom, tajmer treba podesiti na veću vrednost da bi se izbeglo ponovno slanje nepotrebnih okvira; tu signali NAK mogu znatno da ubrzaju ponovno slanje izgubljenih ili oštećenih okvira.

Pitanje koje se odnosi na tajmere i NAK-signale jeste i to za koji okvir se tajmer automatski isključio. U protokolu 5, to je uvek najstariji okvir - okvir *ack_expected*. U protokolu 6 to nije tako jednostavno utvrditi. Pretpostavimo da su poslati okviri 0 do 4, što znači da lista okvira koji su istovremeno na vezi sadrži sekvencu 01234, počev od najstarijeg, pa do najmlađeg okvira. Zamislite sada daje za okvir 0 rok tajmera istekao, da je poslat okvir *5* (nov), da je istekao tajmer za olcvir 1, istekao tajmer za okvir 2 i da je poslat okvir 6 (drugi nov okvir). U tom trenutku lista okvira koji su istovremeno na vezi sadrži sekvencu 3405126, od najstarijeg, pa do najmlađeg okvira. Ako sav dolazni saobraćaj (tj. okviri koji nose potvrde) zakasni, istećí će rokovi tajmera za svih sedam okvira koji su na vezi, navedenim redom.

Da primer ne bi bio još složeniji, odustali smo od objašnjavanja načina održavanja tajmera, već smo pretpostavili da se po isteku roka (nekog od tajmera) u promenljivu *oldestjrame* smešta redni broj okvira na koji se tajmer odnosi.

3.5 PROVERA RADA PROTOKOLA

Stvarni protokoli i programi u koje se ugrađuju često su prilično složeni. Zbog toga su mnoga istraživanja posvećena pronalaženju formalnih matematičkih tehnika za definisanje i proveru protokola. U narednim odeljcima razmotrićemo neke takve modele i tehnike. Iako ćemo im prići sa aspekta sloja veze podataka, oni su primenlji- vi i na druge slojeve.

3.5.1 Modeli mašine konačnih stanja

Ključni pojam koji se koristi u mnogim modelima protokola predstavlja **mašina konačnih stanja** (engl. *finite state machine).* Prema ovom modelu, svaki **računar povezan protokolom** (engl. *protocol machine),* tj. pošiljalac ili primalac, u svakom trenutku se nalazi u određenom stanju. Njegovo stanje je određeno vrednostima svih njegovih promenljivih, uključujući i vrednošću njegovog programskog brojača.

Veliki broj različitih stanja se za svrhe analize najčešće može podeliti 11 manji broj grupa. Na primer, razmatrajući primaoca u protokolu 3, od svih njegovih mogućih stanja možemo

da izdvojimo dva najvažnija: čekanje na okvir 0 i čekanje na okvir 1. Sva druga stanja mogu se smatrati prelaznim fazama ka jednom od ova dva glavna stanja. Stanja se obično biraju kao trenuci u kojima računar povezan protokolom čeka da se desi sledeći događaj [u našim primerima on izvršava proceduru *wait(event)]*. Tada je stanje računara potpuno određeno stanjem njegovih promenljivih. Broj mogućih stanja je 2", gde je *n* broj bitova potrebnih da prikazivanje svih promenljivih.

Stanje celog sistema je kombinacija svih stanja dva računara povezana protokolom i kanala. Stanje kanala određuje njegov sadržaj. Ako za primer opet uzmemo protokol 3, kanal ima četiri moguća stanja: okvir 0 putuje od pošiljaoca ka primaocu, okvir 1 putuje od pošiljaoca ka primaocu, okvir s potvrdom se vraća i kanal je prazan. Ako prihvatimo model po kome pošiljalac i primalac mogu da budu u po dva stanja, onda ceo sistem može da bude u 16 različitih stanja.

Bilo bi na mestu da malo objasnimo stanja kanala. Koncepcija „okvira u kanalu" naravno daje apstrakcija. U stvari time mislimo daje okvir verovatno primljen, ali da još nije obrađen. Okvir se u ovom smislu nalazi u kanalu sve dok računar povezan protokolom ne izvrši procedura *FromPhysicalLayer* i obradi ga.

Iz svakog stanja postoji nula ili više prelaza (engl. *transitions*) u draga stanja. Pre- laz izaziva neki događaj. Na računara povezanom protokolom, prelaz može izazvati slanje okvira, stizanje okvira, automatasko isključivanje tajmera, softverski prekid itd. Kod kanala su tipični događaji umetanje novog okvira 11a izvorištu, isporučivanje okvira na odredište i gubitak okvira zbog šuma. Kada imamo potpun opis računara povezanih protokolom i osobine kanala, možemo da nacrtamo usmereni graf na kome su stanja predstavljena čvorovima, a prelazi usmerenim lukovima.

Jedno posebno stanje označava se kao početno stanje (engl. *initial State)*. To stanje odgovara opisu sistema u trenutku pokretanja ili u nekom pogodnom bliskom trenutku posle toga. Iz početnog stanja se nizom prelaza mogu dostići neka, možda i sva druga stanja. Koristeći dobro poznate tehnike teorije grafova (npr. tranzitivno zatvaranje grafa), moguće je odrediti koja su stanja dostupna, a koja nisu. Ta tehnika se prema Linu i saradnicima (1987), naziva analiza dostupnosti (engl. *reachability analysis)*. Pomoću nje se može utvrditi ispravnost protokola.

Protokol se prema modelu mašine konačnih stanja formalno može posmatrati kao skup četiri elementa (*S*, *M, I, T),* gde

S predstavlja skup mogućih stanja procesa i kanala,

M predstavlja skup okvira koji se mogu razmenjivati kanalom,

I predstavlja skup početnih stanja procesa, a

T predstavlja skup svih prelaza između stanja.

Na početku merenja vremena svi procesi su u svojim početnim stanjima. Tada počinju da se dešavaju događaji, npr. pojavljuje se okvir za slanje ili tajmeru ističe

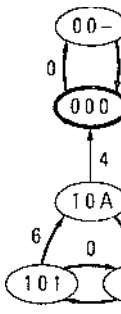
rok. Svaki događaj može da natera jedan od procesa ili kanal da preduzmu neku akciju i pređu u novo stanje. Beležeći uzastopna stanja može se napraviti graf dostupnosti i tako analizirati protokol.

Pomoću analize dostupnosti mogu se otkriti različite greške u specifikaciji protokola. Na primer, ako se određeni okvir može naći u određenom stanju, a mašina konačnih stanja ne određuje koju akciju treba preduzeti, specifikacija protokola je pogrešna (nepotpunost). Ukoliko postoji skup stanja iz kojih nema izlaza, niti napredovanja (ne mogu se dalje primati ispravni okviri), imamo drugu grešku (beskonačna petlja bez izlaza). Manje ozbiljnu grešku u specifikaciji protokola predstavlja slučaj kada se određuje način obrade događaja u stanju u kome se događaj ne može desiti (irelevantan prelaz). Mogu se otkriti i druge greške.

Primer modela mašine konačnih stanja prikazanje na slici 3-21(a). Graf odgovara protokolu 3 koji smo ranije opisali: svaki računar povezan protokolom može se naći u dva stanja, a kanal u četiri. Postoji ukupno 16 stanja sistema, od kojih nisu sva dostupna iz početnog stanja. Nedostupna stanja nisu prikazana na slici. Zbog jednostavnosti su zanemarene i greške kontrolnih zbirova.

Svako stanje je označeno slovima *SRC,* gde *S* može biti 0 ili 1, a odnosi se na okvir koji pošiljalac pokušava da pošalje; *R* takođe može biti 0 ili 1 i odnosi se na okvir koji očekuje primalac, a C može da ima vrednosti 0,1, A ili prazno (-) i odnosi se na stanje kanala. U našem primeru smo za početno stanje odabrali kombinaciju (000). Dragim recima, pošiljalac je upravo poslao okvir 0, primalac očekuje okvir 0 i okvir 0 se upravo nalazi u kanalu.

Na slici 3-21 prikazano je devet vrsta prelaza. Prelaz 0 odgovara situaciji kada kanal izgubi svoj sadržaj. Prelaz 1 odgovara situaciji kada kanal ispravno isporuči paket 0 primaocu, a primalac promeni stanje tako da očekuje okvir 1 i emituje potvrdu o prijemu. Prelaz 1 odgovara i situaciji kada primalac isporučuje paket 0 mrežnom sloju. Ostali prelazi su prikazani na slici 3-21(b). Stizanje paketa s pogrešnim kontrolnim zbirom nije prikazano jer ono ne menja stanje (u protokolu 3).



(b)

Prelaz	Aktivan	Prihvaćeni	Poslati mrežnom sloju	Prelaz je okvir	Ka
0	-	(izgubljeni okvir)			
1	R	0	A	1	-
2	S	A	1	A	Da
3	R	1	A	0	-
4	S	A	0	A	Da
5	R	0	A	A	-
B	R	1	A	0	Ne
7	S	(istek roka)		1	Ne
8	S	(istek roka)			-

Slika 3-21. (a) Dijagram stanja protokola 3. (b) Prelazi.

Tokom normalnog rada, prelazi 1, 2, 3 i 4 stalno se ponavljaju tim redom. U svakom ciklusu isporučuju se dva paketa i pošiljalac se vraća u početno stanje u kome pokušava da pošalje nov okvir s rednim brojem 0. Ako se okvir 0 izgubi u kanalu, sistem prelazi iz stanja (000) u stanje (00-). Konačno, tajmer pošiljaoca se automatski isključuje (prelaz 7) i sistem se vraća u početno stanje (000). Slučaj gubljenja potvrde je složeniji i zahteva dva prelaza (7 i 5 ili 8 i 6) za ispravljanje greške.

U protokolu s jednobitnim rednim brojevima okvira, primalac nikada ne sme da isporuči dva parna paketa bez neparnog međupaketa, i obrnuto, bez obzira na to kakav je redosled događaja. Iz grafa na slici 3-21 vidimo da se ovo pravilo može formalnije iskazati: „ne sme postojati putanja od početnog stanja s dva prelaza 1 ako između njih ne postoji prelaz 3, i obrnuto“. Sa slike se može zaključiti daje protokol u ovom smislu ispravno specificiran.

Sličan zahtev je da ni na jednoj putanji pošiljalac ne sme da dva puta menja stanje (npr. od 0 u 1 i ponovo, od 1 u 0), dok primalac ostaje u istom stanju. Kada bi takva putanja postojala, tada bi odgovarajućim sledom događaja dva okvira mogla biti nepovratno izgubljena a da primalac to ni ne primeti. Sekvenca isporučenih paketa imala bi „rupu“ veličine dva paketa koja se ne može otkriti.

Važno svojstvo protokola je i nemogućnost nastanka kružnog blokiranja. Kružna blokada (engl. *deadlock)* nastaje kada protokol više ne može da napreduje (da isporučuje pakete mrežnom sloju), bez obzira na to šta se dogodi. Jezikom grafova, kružna blokada se može opisati podskupom stanja dostupnim iz početnog stanja, koji ima dva svojstva:

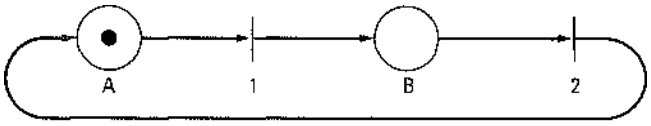
1. Iz podskupa nema prelaza.
2. U podskupu nema prelaza koji bi omogućili napredovanje protokola.

Kada se protokol kružno blokira, tu zauvek ostaje. I opet, lako je videti da u grafu protokola 3 nema takvih blokada.

3.5.2 Modeli mreže Petri

Mašina konačnih stanja nije i jedina tehnika za formalno specificiranje protokola. U ovom odeljku ćemo opisati potpuno drugačiju tehniku - mrežu Petri (engl. *Petri net),* prema Danthineu (1980). Mreža Petri ima četiri osnovna elementa: mesta, prelaze, lukove i žetone (tokene). Mesto (engl. *place*) predstavlja stanje u kome može da bude sistem (ili deo sistema). Slika 3-22 prikazuje mrežu Petri s dva mesta, *A* i *B,* prikazana krugovima. Sistem se trenutno nalazi u stanju *A,* što je označeno crnim kružićem - žetonom (engl. *token).* Prelaz (engl. *transition)* označava se kratkom vertikalnom ili horizontalnom crtom. Svaki prelaz ima nula ili više ulaznih lukova (engl. *input arcs)* koji polaze od izvorišnih mesta, i nula ili više izlaznih lukova (engl. *output arcs*) koji vode ka odredišnim mestima.

Prelaz je omogućen (engl. *enabled)* ako u svakom od njegovih izvorišnih mesta postoji barem jedan žeton. Svaki omogućeni prelaz može da se po želji aktivira (engl. *fire),* pri čemu se po jedan žeton sa svakog izvorišnog mesta premešta na po jedno odredišno mesto. Ako se brojevi ulaznih i izlaznih lukova razlikuju, žetoni neće biti sačuvani. Ako je omogućeno dva ili više prelaza, svaki od njih može da se aktivira. Prelaz koji će se aktivirati nije unapred određen, zbog čega je mreža Petri zgodna za modelovanje protokola. S drage strane, mreža Petri prikazana na slici 3-22, potpuno je deterministička i može se iskoristiti za modelovanje bilo kog dvofaznog procesa (na primer, ponašanja bebe: jede, spava, jede, spava itd.). Kao i obično, pri modelo- vanju su zanemareni nebitni detalji.



Slika 3-22. Mreža Petri s dva mesta i dva prelaza.

Na slici 3-23 prikazanje model mreže Petri koji odgovara protokolu na slici 3-12. Za razliku od modela mašine konačnih stanja, ovde stanja nisu kombinacije pojedinih elemenata, već su stanja pošiljaoca, primaoca i kanala prikazana zasebno. Prelazi 1 i 2 odgovaraju situacijama kada pošiljalac šalje okvir 0 normalno, odnosno posle automatskog isključenja tajmera. Prelazi 3 i 4 su to isto za okvir 1. Prelazi 5, 6 i 7 odgovaraju gubitku okvira 0, gubitku potvrde o prijemu, odnosno gubitku okvira 1. Prelazi 8 i 9 odgovaraju situaciji kada na odredište stigne okvir s pogrešnim rednim brojem. Prelazi 10 i 11 su stizanje sledećeg okvira ispravnim redosledom i njegovo isporuči- vanje mrežnom sloju primaoca.

Mreže Petri se mogu iskoristiti za otkrivanje mana protokola, baš kao i mašine konačnih stanja. Na primer, ako bi se prelaz 10 aktivirao dvaput uzastopce (dakle, bez međuprelaza 11), protokol ne bi radio ispravno. Pojam kružnog blokiranja u mreži Petri sličan je takvom blokiranju kod mašine konačnih stanja.

Mreže Petri mogu da se prikažu u pogodnom algebarskom obliku koji ima svoju gramatiku. Svaki prelaz predstavlja po jedno gramatičko pravilo. Svako pravilo određuje izvorišno i odredišno mesto prelaza. Pošto se na slici 3-23 nalazi 11 prelaza, odgovarajuća gramatika ima 11 pravila, označena brojevima od 1 do 11, a svako odgovara prelazu označenom istim brojem. Gramatika mreže Petri prikazane na slici

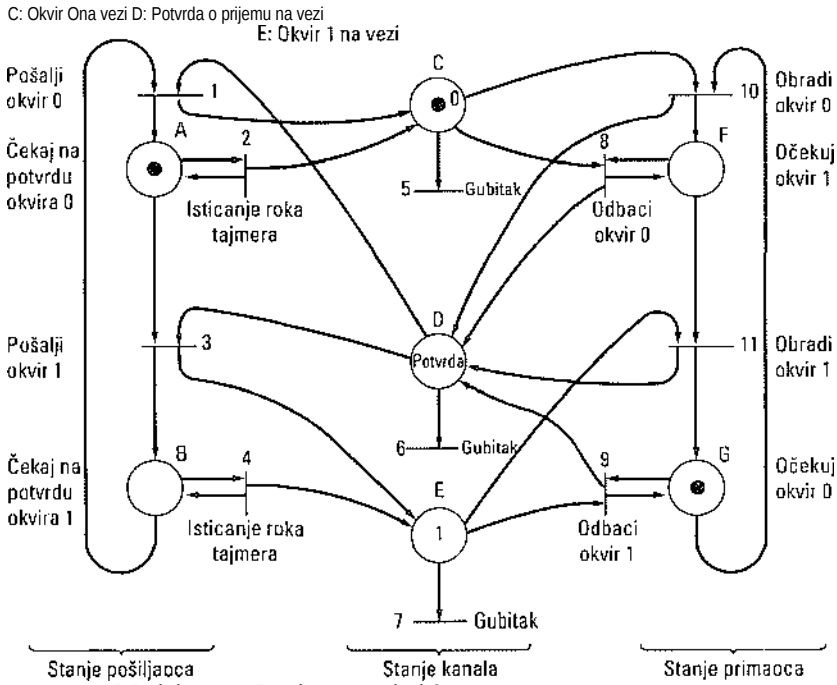
3- 23 izgleda ovako:

1: BD —>
AC 2: A —^
A 3: AD —>
BE 4: B —>
B 5: C —»
6: D —>
7: E —»
8: CF —> DF
9: EG

———

> DG

10: CG —^
DF 11: EF —»
DG



Slika 3-23. Model mreže Petri za protokol 3.

Zaista je zanimljivo to što smo jedan složen protokol uspeli da svedemo na 11 jednostavnih gramatičkih pravila s kojima računarski program može lako da radi.

Aktuelno stanje mreže Petri predstavljeno je neuređenim skupom mesta, pri čemu se svako mesto u skupu pojavljuje onoliko pita koliko ima žetona. Svako pravilo kod koga je mesto na levoj strani prisutno i u skupu, može da se aktivira, pri čemu se ta mesta uklanjaju iz aktuelnog stanja i zamenjuju odgovarajućim odredišnim mestima u novom stanju. Na slici 3-23 stanje je označeno kao *ACG* (mesta A, C i *G* imaju žetone). Prema tome, omogućena su pravila 2, 5 i 10 i svako od njih može da bude primenjeno, što sistem prevodi u novo stanje (možda sa istim oznakama kao i početno). Nasuprot tome, pravilo 3 *(AD —>*) ne može bitiprimenjenojer mesto *D* nijeoznačeno (nema žeton).

3.6 PRIMERI PROTOKOLA SLOJA VEZE

U narednim odeljcima ispitaćemo nekoliko uobičajenih protokola sloja veze. Prvi, protokol HDLC, klasičan je protokol za razmenu bitova, čije varijante već decenij ama imaju mnoge primene. Dragi, protokol PPP, povezuje kućne računare sa Internetom.

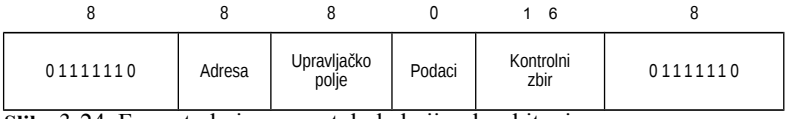
3.6.1 HDLC - protokol za upravljanje povezivanjem podataka na visokom nivou

U ovom odeljku ispitaćemo grupu bliskih protokola koji su već dugo oko nas, ali se i dalje široko koriste. Svi su izvedeni iz protokola sloja veze koji se koristio u svetu IBM-ovih centralnih računara: protokola za **sinhrono upravljanje povezivanjem podataka** (engl.

Synchronous Data Link Control, SDLC), Pošto je razvio protokol SDLC, IBM ga je podneo organizacijama ANSI i ISO s namerom da bude prihvaćen kao američki, odnosno međunarodni standard. Organizacija ANSI je protokol preradila u **naprednu proceduru za upravljanje prenosom podataka** (engl. *Advanced Data Communication Control Procedure, ADCCP),* a ISO u **upravljanje povezivanjem podataka na visokom nivou** (engl. *High-level Data Link Control, HDLC).* Organizacija CCITT gaje tada prihvatila i izmenila u **proceduru za pristupanje vezi** (engl. *Link Access Procedure, LAP),* kao deo standarda za mrežni interfejs X.25, ali ga je kasnije dodatno izmenila u **proceduru** B **za pristupanje vezi** (engl. *Link Access Procedure B, LAPB)* da bi bio kompatibilniji s novijim verzijama protokola HDLC. Postojanje tolikih standarda je odlična stvar jer vam omogućuje da birate; ako i ne pronađete nešto što vam se dopada, možete mirno da pričekate nove protokole koji će se verovatno pojaviti ubrzo.

Svi navedeni protokoli zasnivaju se na istim principima. Svi rade s bitovima i svi koriste tehniku umetanja bitova da bi jasno obeležili korisničke podatke. Iako se razlikuju samo u sitnicama, to može da bude veoma iritirajuće. Sledeći opis protokola koji rade s bitovima predstavlja opšti uvod u problematiku; detalje konkretnih protokola potražite u njihovim definicijama.

Svi protokoli koji rade s bitovima koriste struktura okvira prikazanu na slici 3-24. Polje *Adresa* je važno u vezama s više terminala, gde se koristi za identifikovanje jednog od njih. U vezama tipa od tačke do tačke ono se ponekad koristi da bi se komande jasno razgraničile od odgovora na njih.



Slika 3-24. Format okvira za protokole koji rade s bitovima.

Upravljačko polje se koristi za redne brojeve, potvrde i drago, o čemu će biti reči u nastavku.

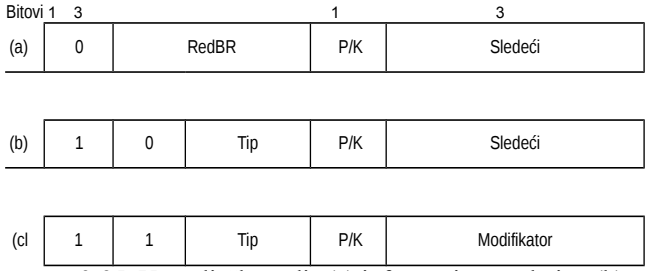
Polje *Podaci* može da sadrži bilo kakve informacije. Ono može da bude proizvoljne dužine, iako efikasnost kontrolnog zbira opada s povećanjem dužine okvira zbog veće verovatnoće pojavljivanja višestrukih rafalnih grešaka.

Polje *Kontrolni zbir* sadrži kod za cikličnu proveru redundanse tehnikom koju smo opisali u odeljku 3.2.2.

Početak i kraj okvira označeni su indikatorskom sekvencom (01111110). U vezama tipa od tačke do tačke koje trenutno ništa ne rade, stalno se emituju takve indilca- torske sekvence. Najmanji okvir sadrži tri polja ukupne dužine 32 bita, ne računajući indikatore na oba kraja.

Postoje tri vrste okvira: informacioni (engl. *information),* nadzorni (engl. *supervisory)* i nenumerisani (engl. *unnumbered).* Sadržaj *Upravljačkog polja* okvira ove tri vrste prikazanje na slici 3-25. Protokoli se služe kliznim prozorom i 3-bitnim rednim brojevima. U svakom trenutku na vezi sme da bude istovremeno do sedam nepotvrđenih okvira. Polje *RedBr* sa slike 3-25(a) sadrži redni broj okvira. Polje *Sledeći* predstavlja šlepovanu potvrdu. Ono se tako zove jer svi protokoli, umesto da kao potvrdu šalju broj poslednjeg ispravno primljenog

okvira, šlepuju broj sledeceg očekivanog okvira. U načelu se može izabrati jedno ili drugo, važno je samo da se konvencija dosledno sprovodi.



Slika 3-25. Upravljačko polje (a) informacionog okvira, (b) nadzornog okvira i (c) nenumerisanog okvira.

Bit *P/K* znači *Poziv/Kraj.* On se koristi kada računar (ili koncentrator) poziva grupu terminala. U režimu *P,* računar poziva terminal da šalje podatke. U svim okvirima koje šalje terminal, osim u poslednjem, bit *P/K* ima vrednot *P.* U poslednjem okviru on ima vrednost *K.*

U nekim protokolima, bit *P/K* se koristi za trenutno iznuđivanje nadzornog okvira od drugog računaia, umesto da se čeka normalan okvir s podacima na koji bi se šlepovale informacije o prozora. Bit *P/K* ima i izvesnu primenu u vezi s nenumerisanim okvirima.

Pomoću polja *Tip* razlikuju se vrste nadzornih okvira. Tip 0 odgovara okvira s potvrdom (čije zvanično ime je SPREMAN ZA PRIJEM, engl. *RECEIVE READY)* kojim se ukazuje na sledeći očekivani okvir. Ovaj okvir se koristi kada nema povratnog saobraćaja koji bi omogućio šlepovanje potvrde.

Tip 1 odgovara okvira s negativnom potvrdom (ODBAČEN, engl. *REJECT).* Njime se saopštava da je u prenosu otkrivena greška. U njemu polje *Sledeći* označava prvi okvir u nizu koji nije ispravno primljen (onaj koji treba ponovo poslati). Od pošiljaoca se zahteva da pošalje sve nepotvrđene okvire, počevši od okvira *Sledeći.* Takva strategija više liči na naš protokol 5, nego na protokol 6.

Tip 2 je NISAM SPREMAN ZA PRIJEM (engl. *RECEIVE NOT READY).* Njime se potvrđuju svi okviri do okvira *Sledeći,* osim njega, baš kao i okvirom SPREMAN ZA PRIJEM, ali se pošiljaocu nalaže da prestane da šalje okvire. Okvir NISAM
SPREMAN ZA PRIJEM služi da ukaže na određene privremene probleme kod primaoca, npr. na nedostatak raspoloživih bafera, i nije alternativa protokolu kliznih prozora. Kada se problemi otklone, primalac šalje okvir SPREMAN ZA PRIJEM, okvir ODBAČEN ili neki drugi upravljački okvir.

Tip 3 je SELEKTIVNO ODBACIVANJE (engl. *SELECTIVE REJECT).* Takvim okvirom se zahteva ponovno slanje samo naznačenog okvira. To u izvesnom smislu više podseca na naš protokol 6, nego na protokol 5, i zato je najkorisnije kada je veličina prozora pošiljaoca barem upola manja od opsega rednih brojeva. Tako, ako primalac želi da prekoredne okvire smesti u bafer i kasnije iskoristi, on može da izazove ponovno slanje bilo kog pojedinačnog okvira emitujući okvir SELEKTIVNO ODBACIVANJE. U protokolima HDLC i ADCCP takav okvir je predviđen, ali ga protokoli SDLC i LAPB ne poznaju - okvir tipa 3 kod njih nije definisan.

Treću klasu čine nenumerisani okviri. Takav okvir se ponekad koristi za upravljanje, ali može da nosi i podatke onda kada se oslanjamo na nepouzdanu bežičnu vezu. Protokoli koji rade s bitovima razlikuju se u pogledu ovog okvira, iako se gotovo potpuno slažu u pogledu okvira druge dve vrste. Na raspolaganju je pet bitova za označavanje tipa okvira, ali se ne koriste sve 32 mogućnosti.

U svim protokolima postoji komanda DISC (prekini vezu, engl. *DISConnect)* kojom računar objavljuje da će se isključiti (zbog preventivnog održavanja). Postoji i komanda SNRM (uspostavljanje normalnog radnog režima, engl. *Set Normal Response Mode)* kojom novouključeni računar objavljuje svoje prisustvo na vezi i pri- nuđuje sve redne brojeve okvira da se vrate na nulu. Taj radni režim je, nažalost, sve drugo osim „normalan". To je neuravnotežen (tj. asimetričan) režim u kome jedan kraj veze naređuje, dok drugi sluša. Komanda SNRM potiče iz vremena kada je pre- nos podataka podrazumevao komunikaciju između „glupog" terminala i velikog umreženog računara, koja je izvesno bila asimetrična. Da bi se rad prilagodio ravnopravnim partnerima, u protokole HDLC i LAPB uneta je dodatna komanda SABM (uspostavi uravnoteženi asinhroni režim, engl. *Set Asynchronous Balanced Mode),* koja vraća početne parametre veze i partnere proglašava jednakim. U pomenutim protokolima postoje i komande SABME i SNRME, varijante komandi SABM i SNRM, pomoću kojih se omogućava korišćenje proširenog formata okvira sa 7-bitnim rednim brojevima, umesto 3-bitnih.

Treća komanda koju ćete naći u svim protokolima jeste komanda FRMR (odbaci okvir, engl. *FRaMe Reject)* kojom se označava da je stigao okvir sa ispravnim kontrolnim zbirom i nemogućom semantikom. Primeri nemoguće semantike su nadzorni okvir tipa 3' u protokolu LAPB, okvir kraći od 32 bita, neodgovarajući upravljački okvir, potvrda o prijemu okvira koji je izvan prozora itd. FRMR okviri sadrže 24-bitno polje s podacima[1] koji opisuju šta nije u redu sa okvirom. Podaci obuhvataju upravljačko polje neispravnog okvira, parametre prozora i skup bitova koji označava različite vrste grešaka.

Upravljački okviri se mogu izgubiti ili oštetiti, baš kao i okviri s podacima, tako da je i za njih potrebna potvrda o prijemu. O tome brine poseban upravljački okvir, UA (nenumerisana potvrda, engl. *Unnumbered Acknowledgement).* Pošto samo jedan upravljački okvir može u jednom tenutku da bude nepotvrđen, uvek je jasno na koji se okvir odnosi potvrda.

Ostali upravljački okviri bave se inicijalizovanjem, pozivanjem i izveštavanjem o statusu. Postoji i upravljački okvir UI (nenumerisan informacioni, engl. *Unnumbered Information)* koji može da sadrži proizvoljne podatke. Ti podaci nisu namenjeni mrežnom sloju već sloju veze primaoca.

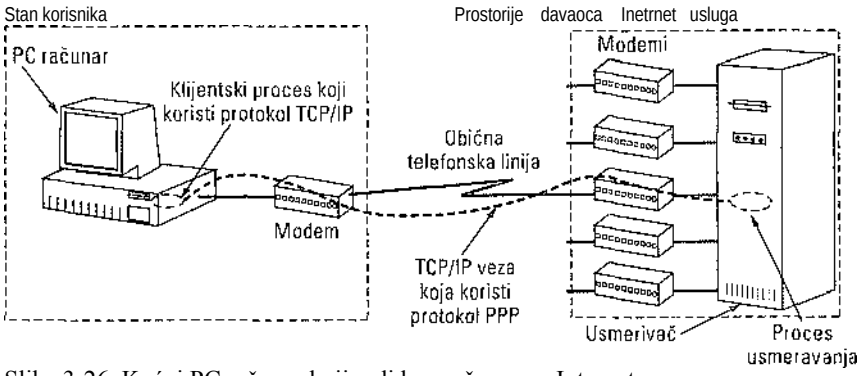
Uprkos raširenoj upotrebi, protokol HDLC je daleko od savršenstva. Kritiku njegovog rada nadi čete kod Fiorinija i saradnika (1994.).

3.6.2 Sloj veze podataka na Internetu

Internet se sastoji od pojedinačnih umreženih računara i usmerivača, međusobno povezanih komunikacionom infrastrukturom. Za povezivanje unutar pojedinačnih zgrada uveliko se koriste lokalne mreže, ali se za povezivanje na širem području većinom koriste iznajmljene linije tipa od tačke do tačke. O lokalnim mrežama ćemo govoriti u 4. poglavlju, dok ćemo se ovde zadržati na protokolima sloja veze koji rade na Internet linijama od tačke do tačke.

Komunikacija od tačke do tačke u praksi se prvenstveno koristi za dve svrhe. Prvo, u hiljadama organizacija postoji jedna ili više lokalnih mreža, svaka sa određenim brojem računara (ličnih računara, korisničkih radnih stanica, servera itd.) i usmerivača (ili mrežnih mostova, što je u pogledu funkcije isto). Usmerivači su često povezani okosnicom lokalne mreže. Sve veze ka spoljnom svetu obično se ostvaruju preko jednog ili dva usmerivača koji su iznajmljenim linijama tipa od tačke do tačke povezani sa udaljenim usmerivačima. Baš ti usmerivači i njihove iznajmljene linije obrazuju komunikacione podmreže na kojima se zasniva Internet.

Drugo, milioni pojedinačnih korisnika povezuju se na Internet pomoću modema i telefonskih linija. Obično to ide tako što se korisnikov PC računar povezuje sa usme- rivačem davaoca Internet usluga i posle toga radi kao punopravan računar na Internetu (engl. *Internet host)*. Taj postupak se ne razlikuje od korišćenja iznajmljene linije između korisnikovog PC računara i usmerivača, osim što se veza prekida kada korisnik završi sesiju. Na slici 3-26 prikazanje kućni PC računar koji se povezuje s davaocem Internet usluga. Modem je prikazan kao eksterni da bi se istakla njegova uloga, premda savremeni računari imaju interne modeme.



Slika 3-26. Kućni PC računar koji radi kao računar na Internetu.

Za stalnu iznajmljenu vezu između usmerivača, kao i za povremenu vezu između računara i usmerivača potreban je protokol sloja veze tipa od tačke do tačke radi uokvirivanja podataka, obrade grešaka i obavljanja drugih funkcija sloja veze o kojima smo govorili u ovom poglavlju. Protokol koji se za to koristi na Internetu zove se PPP. Opisaćemo ga u nastavku.

PPP - protokol od tačke do tačke

Na Internetu je protokol od tačke do tačke neophodan za mnoge svrhe, a među njima i za obavljanje saobraćaja između usmerivača, kao i između korisnika i davaoca Internet usluga. To je **protokol od tačke do tačke** (engl. *Point-to-Point Protocol, PPP),* definisan dokumentom RFC 1661 i dalje razrađen u drugim RFC dokumentima (npr. u dokumentima RFC 1662 i RFC 1663). PPP obrađuje greške, podržava više protokola, omogućava dogovaranje IP adresa prilikom povezivanja i potvrđivanje identiteta, a ima i mnoge druge osobine.

U načelu, protokol PPP obezbeđuje sledeđe:

1. Metodu uokvirivanja podataka pomoću koje se jasno razgraničava kraj jednog okvira od početka drugog. Format okvira omogućava i otkrivanje grešaka.
2. Protokol za upravljanje vezom pomoću koga se linije povezuju, proveravaju, dogovaraju komunikacione opcije i linije na kraju ponovo razvezuju. Taj protokol se zove **protokol za upravljanje vezom** (engl. *Link Control Protocol, LCP).* On podržava sinhrona i asinhrona kola, te kodiranje bitova i bajtova.
3. Dogovaranje opcija mrežnog sloja nezavisno od protokola koji se u tom sloju koristi. Izabrana je metoda u kojoj se za svaki podržani mrežni sloj koristi drugačiji **protokol za upravljanje mrežom** (engl. *Network Control Protocol, NCP).*

Da biste shvatili kako to sve zajedno radi, zamislimo tipičnu situaciju u kojoj kućni korisnik poziva davaoca Internet usluga u nameri da se privremeno poveže na Internet. Korisnikov PC računar najpre pomoću modema poziva usmerivač davaoca Internet usluga. Pošto usmerivačev modem odgovori na poziv i uspostavi fizičku vezu, PC računar šalje usmerivaču niz LCP paketa u polju za korisničke podatke jednog ili više PPP okvira. Ti paketi i odgovori na njih dogovaraju PPP parametre koji će se koristiti.

Pošto se dve strane slože oko parametara, šalje se niz NCP paketa za konfigurisanje mrežnog sloja. Najčešće PC računar želi da izvršava skup protokola TCP/IP i zato traži da mu se dodeli IP adresa. Takvih adresa nema na pretek, pa zato davalac Internet usluga obično ima blok rezervisanih adresa iz koga povezanimkorisnicima tokom trajanja sesije dinamički i privremeno dodeljuje jednu adresu. Ako davalac Internet usluga ima *n* IP adresa, istovremeno će moći da usluži *n* korisnika, ali broj njegovih korisnika može biti mnogo veći jer se oni samo povremeno uključuju. Protokol NCP prilagođen za IP dodeljuje IP adresu.

U ovom trenutku, pošto je dobio IP adresu, PC računar postaje jedan od računara na Internetu i može da šalje i prima IP pakete, kao i računari koji su stalno povezani na Internet. Kada korisnik završi rad, NCP prekida vezu u mrežnom sloju i oslobađa korišćenu IP adresu. Zatim LCP prekida vezu u sloju veze podataka. Na kraju, računat- nalaže modernu da prekine telefonsku vezu - vezu u fizičkom sloju.

Format PPP okvira izabran je tako da bude što sličniji formatu HDLC okvira jer nije bilo stvarne potrebe da se po svaku cenu izmišlja topla voda. Osnovna razlika između protokola PPP i HDLC jeste to što protokol PPP radi sa znakovima, a HDLC s bitovima. Naglasimo da se u protokolu PPP na modemskim linijama koristi umetanje bajtova, tako da svaki okvir obuhvata celobrojan broj bajtova. Nemoguće je poslati okvir dužine 30,25 bajtova, kao što se može protokolom HDLC. PPP okviri se mogu slati telefonskim linijama, ali i SONET linijama (pravim HDLC linijama koje rade s bitovima), na primer, za međusobno povezivanje usmerivača. Format PPP okvira prikazan je na slici 3-27.

Slika 3-27. Pun format PPP okvira u nenumerisanom režimu rada.

Svi PPP okviri počinju standardnim indikatorskim HDLC bajtom (01111110) koji se po

potrebi i umeće ako se njegova sekvenca pojavi u polju s korisničkim podacima. Polje Adresa uvek sadrži binarnu vrednost 11111111 koja znači da sve stanice treba da prihvate okvir. Kada se ona koristi, otpada potreba dodeljivanja adrese sloju veze.

Podrazumevana vrednost sledeceg *Upravljačkog polja* je 00000011 i ona znači da je u pitanju nenumerisan okvir. Drugim recima, PPP ne obezbeđuje podrazumevano pouzdan prenos pomoću rednih brojeva i potvrda o prijemu. U okruženju prepunom smetnji, kao što su bežične mreže, može se koristiti pouzdan prenos u režimu nume- risanja okvira. Svi detalji o tome nalaze se u dokumentu RFC 1663, ali se takav prenos u praksi retko koristi.

Pošto su vrednosti polja *Adresa* i *Upravljačkog polja* u podrazumevanoj konfiguraciji uvek iste, LCP ima mehanizam pomoću koga se dve strane mogu dogovoriti da ta polja potpuno izostave i tako uštede 2 bajta po okviru.

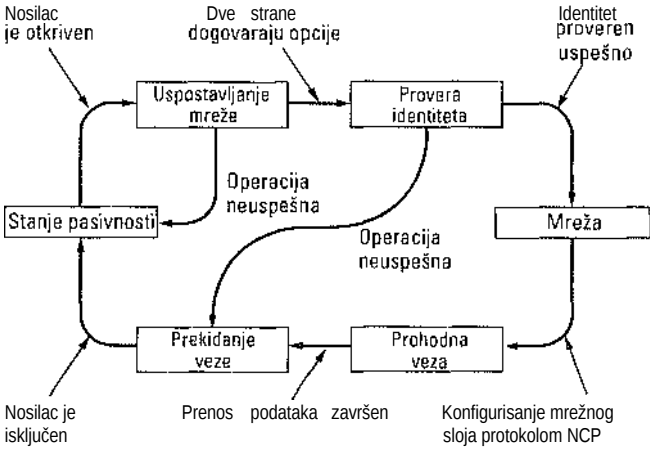
Vrednost u polju *Protokol* označava vrstu paketa u polju *Korisnički podaci.* Una- pred su definisane vrednosti za protokole LCP, NCP, IP, IPX, AppleTalk i mnoge druge protokole. Protokoli čija oznaka počinje bitom 0 predstavljaju protokole mrežnog sloja, kao što su protokoli IP, IPX, OSI CLNP, XNS. Protokoli čija oznaka počinje bitom 1 koriste se za dogovaranje drugih protokola. Među njima su LCP i poseban NCP za svaki mrežni sloj koji protokol podržava. Podrazumevana veličina polja *Protokol* je 2 bajta, ali se protokolom LCP može dogovoriti da ona bude 1 bajt.

Korisnički podaci mogu biti različite dužine, sve do dogovorenog maksimuma. Ako tokom uspostavljanja veze protokolom LCP taj maksimum nije dogovoren, koristi se podrazumevana dužina 1500 bajtova. Ako je potrebno, stvarni podaci se u polju dopunjavaju nulama do maksimalne dužine.

Posle korisničkih podataka sledi polje *Kontrolni zbir,* obično dugačko 2 bajta, ali se može dogovoriti da bude dugačko i 4 bajta.

Sve u svemu, PPP je višeprotokolarni mehanizam za uokvirivanje podataka, pogodan za korišćenje u modemima, HDLC linijama koje rade sa sekvencama bitova, SONET linijama i drugim fizičkim slojevima. On podržava okrivanje grešaka, dogovaranje opcija, komprimovanje zaglavlja i, po želji, pouzdan prenos okvira u formatu HDLC.

Pređimo sada s formata okvira PPP na način povezivanja i raskidanja linija. Na slici 3-28 prikazane su faze tog procesa. Prikazani redosled operacija važi i za modemske veze i za međusobno povezivanje usmerivača.



Slika 3-28. Uprošćen dijagram operacija pri uspostavljanju i raskidanju veze.

Protokol počinje s linijom koja je u stanju pasivnosti *(DEAD),* što znači da ne postoji nosilac fizičkog sloja, pa ni veza na nivou fizičkog sloja. Posle fizičkog povezivanja, linija je u fazi uspostavljanja veze (*ESTABLISH*). U tom trenutku započinje pregovaranje o opcijama koje, ukoliko se uspešno završi, prebacuje liniju u fazu pro- vere identiteta (*AUTHENTICATE*). Sada dve strane mogu uzajamno da provere identitet ako to žele. Kada se uđe u fazu mreže *(NETWORK),* poziva se odgovarajući NCP protokol da konfiguriše mrežni sloj. Ukoliko se to završi uspešno, veza postaje prohodna (faza *OPEN)* i podaci mogu početi da se prenose. Po završenom prenosu, linija prelazi u fazu prekidanja veze *(TERMINATE),* a odatle ponovo u stanje pasivnosti *(DEAD),* gde se napušta i fizički nosilac podataka.

LCP dogovara opcije sloja veze tokom faze njenog uspostavljanja *(ESTABLISH);* pri tome ga se opcije u stvari ne tiču, već samo mehanizam dogovaranja. On omogućuje da proces koji započinje dijalog da predlog i da druga strana taj predlog prihvati ili odbaci, delimično ili u celini. On omogućuje i to da procesi provere kvalitet linije i uvere se da je dovoljno dobra za uspostavljanje veze. Na kraju, protokol L,CP omogućava i da se veza prekine kada više nije potrebna.

U dokumentu RFC 1661 definisano je 11 LCP okvira, navedenih na slici 3-29. Četiri okvira tipa *Configure-* omogućavaju začetniku veze (I) da predloži vrednosti opcija i drugoj strani (R) da ih prihvati ili odbaci. U drugom slučaju, strana koja odgovara može da predloži nešto drugo ili da objavi da uopšte ne želi da pregovara o opcijama. Opcije o kojima se pregovara i njihove predložene vrednosti deo su LCP okvira.

Ime	Smer	Opis
Configure-request	1 —> R	Lista predloženih opcija i vrednosti
Configure-ack	1 <— R	Sve opcije se prihvataju
Configure-nak	1 <— R	Neke opcije se ne prihvataju
Configure-reject	I<-R	0 nekim opcijama se ne pregovara
Terminate-request	1 —> R	Zahtev za prekidanje veze
Terminate-ack	1 <— R	U redu, veza je prekinuta
Code-reject	1 <— R	Primljen je nepoznat zahtev
Protocol-reject	1 <- R	Zahteva se nepoznat protokol
Echo-request	1 —» R	Pošalji ovaj okvir natrag
Echo-reply	1 <- R	Šaljem taj okvir natrag
Discard-request	1 —> R	Samo odbaci ovaj okvir (za proveru)

Slika 3-29. Vrste LCP okvira.

Kodovi u okvirima tipa *Terminate-* raskidaju vezu kada više nije potrebna. Kodovi u okvirima *Code-reject* i *Protocol-reject* ukazuju na to da je draga strana dobila nešto što ne razume. To može značiti da se tokom prenosa provukla neotkrivena greška, ali najčešće znači da inicijator i draga strana izvršavaju različite verzije protokola LCP. Okviri *Echo-* služe za provera kvaliteta linije. Na kraju, okvir *Discard-request* omogućava otklanjanje grešaka. Kada se bilo koja strana suoči s teškoćama pri slanju bitova na liniju, programer može da iskoristi ovakav okvir za njeno proveravanje. Ako se okvir probije do primaoca, on će ga odbaciti da ne bi zbunjivao onoga ko proverava liniju.

Opcije koje se mogu dogovoriti obuhvataju maksimalnu dužinu polja za korisničke podatke u odgovarajućim okvirima, omogućavanje provere identiteta i izbor vrste protokola, omogućavanje praćenja kvaliteta linije tokom normalnog rada i biranje različitih opcija za komprimovanje zaglavlja.

O protokolima NCP malo se šta može uopšteno reći. Svaki od njih je prilagođen nekom mrežnom protokolu i omogućava zahteve za konfigurisanje koji su mu svojstveni. Za protokol IP, na primer, najvažnija je mogućnost dinamičkog dodeljivanja adresa.

3.7 SAŽETAK

Zadatak sloja veze podataka jeste da prosti tok bitova koji pritiče iz fizičkog sloja pretvori u niz okvira koje će moći da iskoristi mrežni sloj. Za to se koriste različite metode uokvirivanja, uključujući prebrojavanje znakova, umetanje bajtova i umetanje bitova. Protokoli sloja veze mogu da obezbede ispravljanje grešaka ponovnim slanjem oštećenih ili

izgubljenih okvira. Da bi usaglasio brzinu slanja i primanja okvira, sloj veze podataka može i da upravlja tokom. Ispravljanje grešaka i upravljanje tokom zajedno su na pogodan način integrisani u mehanizam kliznih prozora.

Protokoli kliznih prozora mogu se svrstavati prema veličini prozora pošiljaoca i prema veličini prozora primaoca. Kada je veličina oba prozora 1, to je protokol „stani i čekaj". Kada je prozor pošiljaoca veči od 1, da bi se, na primer, izbeglo čekanje pošiljaoca na liniji s velikim vremenom obilaska veze, primalac se može programirati da odbacuje sve prelcoredne okvire ili da takve okvire smešta u bafer dok na njih ne dođe red.

U ovom poglavlju smo ispitali više protokola. Protokol 1 je namenjen okruženju u kome nema grešaka i situaciji kada primalac uvek može da obradi sve pristigle okvire. U protokolu 2 se i dalje pretpostavlja bezgrešan rad, ali se uvodi i upravljanje tokom. U protokolu 3 greške se obrađuju pomoću rednih brojeva okvira i algoritma „stani i čekaj". Protokol 4 omogućava dvosmemu komunikaciju i uvodi pojam šlepovanja potvrda. U protokolu 5 koristi se protokol kliznih prozora i to protokol „vrati se n". Na kraju, u protokolu 6, koriste se selektivno ponavljanje i negativne potvrde.

Protokoli se mogu modelovati različitim tehnikama koje pomažu da se ispolji njihova efikasnost (ili neefikasnost). Najčešće su to modeli mašine konačnih stanja i mreže Petri.

U mnogim mrežama se koristi neki od protokola koji u sloju veze rade s bitovima: SDLC, HDLC, ADCCP ili LAPB. Svi oni za razgraničenje susednih okvira koriste in- dikatorski bajt, kao i tehniku umetanja bajtova da se indikatorski bajt ne bi pojavio u podacima. Svi takođe koriste klizne prozore za upravljanje tokom. Na Internetu se koristi PPP kao osnovni protokol sloja veze za linije tipa od tačke do tačke.

ZADACI

1. Paket iz jednog od gornjih slojeva izdeljen je u 10 okvira, od koji za svaki postoji 80 procenata verovatnoće da stigne neoštećen. Ako protokol sloja veze ne podržava mehanizam obrade grešaka, koliko prosečno puta treba slati poruku da bi ceo paket ispravno stigao na odredište?
2. U protokolu sloja veze koristi se sledeće kodiranje znakova:

   A: 01000111; B: 11100011; FLAG: 01111110; ESC: 11100000

   Napišite (binarnu) sekvencu bitova poslatih za okvir od četiri znaka: A B ESC INDIKATOR, kada se koristi svaka od sledećih metoda uokvirivanja:

   a) Prebrojavanje znakova.
   b) Indikatorski bajtovi uz umetanje bajtova.
   c) Početni i završni indikatorski bajtovi uz umetanje bitova.
3. U toku podataka za koji se koristi algoritam umetanja bajtova opisan u tekstu, nalazi se sledeći fragment: A B ESC C ESC INDIKATOR INDIKATOR D. Kako će on izgledati posle umetanja bajtova?
4. Jedan od vaših kolega tvrdi da je šteta da se svaki okvir završava indikatorskim bajtom, a zatim da sledeći okvir počinje istim takvim bajtom. Samo jedan indikatorski bajt mogao bi da obavi isti posao. Da li se s time slažete?
5. Niz bitova 0111101111101111110 treba da se prenese u sloju veze podataka. Kako taj niz izgleda posle umetanja bitova?
6. Kada se primeni tehnika umetanja bitova, da li je moguće da gubitak, umetanje ili

promena jednog jedinog bita izazove grešku koja se ne može otkriti kontrolnim zbirom? Ako nije, zašto? Ako jeste, objasnite kako se to događa. Da li dužina kontrolnog zbira igra ovde nekakvu ulogu?

7. Možete li da zamislite situaciju u kojoj bi protokol sa otvorenom petljom (na primer, Hamingov kod) imao prednost nad protokolima koji koriste povratne informacije o kojima smo govorili u ovom poglavlju?
8. Da bi se povećala pouzdanost iznad granice koju može da pruži samo jedan bit parnosti, u jednom kodu za otkrivanje grešaka koristi se jedan bit parnosti za proveru svih neparnih bitova, a drugi za proveru svih parnih. Koliko je Hamingovo nastojanje ovog koda?
9. Pomoću Hamingovog koda šalju se šesnaestobitne poruke. Koliko je potrebno kontrolnih bitova da bi primalac mogao da otkrije i ispravi sve jednobitne greške? Napišite sekvencu bitova koji se šalju za poruku 1101001100110101. Pretpostavite da je u Hamingovom kodu broj jedan u kodnoj reći paran.
10. Bajt binarne vrednosti 10101111 treba kodirati Hamingovim kodom uz paran broj jedinica u kodnoj reči. Kako izgleda binarna vrednost ovog bajta posle kodiranja?
11. Hamingov kod dužine 12 bitova, čija je heksadecimalna vrednost 0xE4F, stiže primaocu. Kako izgleda prvobitna vrednost poslatih podataka u heksađecimalnom obliku? Pretpostavite daje najviše 1 bit pogrešan.
12. Jedan način otkrivanja grešaka podrazumeva slanje bloka od *n* redova, sa *k* bitova u svakom redu, i dodavanje bita parnosti svakom redu i svakoj koloni. U donjem desnom uglu bloka je bit parnosti koji se odnosi na poslednji red i poslednju kolonu. Da li ova šema omogućava otkrivanje svih pojedinačnih grešaka? A, dvostrukih? Trostrukih?
13. Blok bitova sa *n* redova i *k* kolona ima horizontalne i vertikalne bitove parnosti u cilju otkrivanja grešaka. Pretpostavimo da su tokom prenosa invertovana 4 bita. Iz- vedite izraz za verovatnoću da greška prođe neprimećeno.
14. Šta je ostatak deljenja polinoma $x^7 + X^5 + 1$ generatorskim polinomom $x^3 + 1$?
15. Tok bitova 10011101 prenosi se standardnom CRC metodom opisanom u tekstu. Generatorski polinom je $X^3 + 1$. Napišite tok bitova koji se stvarno šalje. Pretpostavite da je treći bit sleva tokom prenosa invertovan. Pokažite da će primalac otkriti ovu grešku.
16. Protokoli sloja veze skoro uvek stavljaju CRC u završni blok paketa, umesto u zaglavlje. Zašto?
17. U kanalu je brzina prenosa 4 kb/s, a kašnjenje signala 20 rns. Za koji opseg veličina okvira protokol „stani i čekaj“ omogućava efikasnost od barem 50 procenata?
18. Za prenos okvira od 64 bajta pomoću protokola 5 koristi se regionalni TI kabl dužine 3000 km. Ako signal kasni tempom 6 ps/km, kolike dužine (u bitovima) treba da budu redni brojevi okvira?
19. Da li je u protokolu 3 moguće da pošiljalac ponovo aktivira (vrati na početak) već aktivan tajrner? Ako je moguće, kako se to može dogoditi? Ako nije, zašto?
20. Zamislite protokol kliznih prozora u kome se koriste redni brojevi tolike dužine da se nikada ciklično ne obnavljaju. Koji odnosi moraju postojati između četiri ivice prozora i njegove veličine koja je konstantna i jednaka kod pošiljaoca i primaoca.
21. Da procedura *between* u protokolu 5, umesto uslova $a < b < c$, proverava uslov $a < b < c$, da li bi to uticalo na ispravnost rada i efikasnost protokola? Obrazložite odgovor.
22. Kada u protokolu 6 stigne okvir s podacima, proverava se da li se njegov redni broj slaže sa očekivanim i da li je *no_nak-tmz.* Ako su oba uslova zadovoljena, šalje se okvir NAK. U suprotnom se uključuje pomoćni tajmer. Pretpostavimo daje odredba else ispuštena. Da li ce to uticati na ispravan rad protokola?
23. Pretpostavimo daje na kraju koda protokola 6 ispuštena petlja while s tri naredbe. Da li

to utiče na ispravnost rada protokola ili samo na njegovu efikasnost? Obrazložite odgovor.

24. Pretpostavimo daje odredba case za kontrolni zbir Uklonjena iz naredbe switch protokola 6. Kako će to uticati na rad protokola?
25. U protokolu 6, kod za *frame_arrival* ima deo koji se odnosi na NAK. Taj deo se poziva ako je dolazni okvir NAK i ako je zadovoljen još jedan uslov. Opišite situaciju u kojoj je suštinski važno postojanje tog dodatnog uslova.
26. Zamislite da pišete softver za sloj veze podataka linije koja se isključivo koristi za slanje podataka vama. Na drugoj strani linije koristi se protokol HDLC, sa 3-bitnim rednim brojevima i veličinom prozora od sedam okvira. Da biste povećali efikasnost, želeli biste da smeštate u bafer što je više moguće prekorednih okvira, ali ne možete da menjate softver na suprotnom kraju linije. Da li je moguće imati prozor primaoca veći od 1, a da se ipak garantuje potpuno ispravan rad protokola? Ako je moguće, koji je najveći prozor koji se bezbedno može koristiti?
27. Razmotrite rad protokola 6 na liniji brzine 1 Mb/s u kojoj se ne javljaju greške. Najveći dozvoljen okvir je 1000 bitova. Svake sekunde se generiše po jedan nov paket. Rok tajmera je 10 ms. Kada bi se uklonio tajmer specijalizovan za potvrde o prijemu okvira, dolazilo bi do nepotrebnog prekoračenja rokova. Koliko bi puta trebalo slati okvir da bi se prenela prosečna poruka?
28. U protokolu 6, *MAX_SEQ* = 2" - 1. Iako je to poželjno da bi se efikasno iskoristili bitovi zaglavlja, nismo naglašavali da je to i neophodno. Da li protokol radi ispravno za, na primer, *MAX_SEQ* = 4?
29. Kanalom propusne moći 1 Mb/s, ostvarenim pomoću geostacionarnog satelita do koga signal sa Zemlje stiže posle 270 ms, šalju se okviri veličine 1000 bitova. Potvrde o prijemu uvek se šlepuju uz okvire s podacima. Zaglavlja su vrlo kratka. Koriste se 3-bitni redni brojevi. Koliko je maksimalno moguće iskorišćenje kanala za:
    a) protokol „stani i čekaj“,
    b) protokol 5,
    c) protokol 6?
30. Izračunajte deo propusnog opsega koji se nekorisno troši (zaglavlja okvira i ponovno slanje okvira) za protokol 6 koji radi na opterećenom satelitskom kanalu brzine 50 kb/s, pri čemu se okviri s podacima sastoje od 40 bitova zaglavlja i 3960 bitova podataka. Pretpostavite da signal putuje od Zemlje do satelita 270 ms. Okviri ACK nikada se ne šalju. Okviri NAK su veličine 40 bitova. Učestalost grešaka za okvire s podacima je 1%, a za NAK okvire je zanemarljiva. Redni brojevi su 8-bitni.
31. Razmotrite bezgrešan satelitski kanal brzine 63 kb/s kojim se u jednom smeru šalju okviri s podacima dužine 512 bajtova, a u drugom veoma kratke potvrde. Koliki je maksimalan protok podataka za prozore veličine 1,7, 15 i 127? Od Zemlje do satelita signal putuje 270 ms.
32. Kabl dužine 100 km radi brzinom nosioca TI. Brzina prostiranja signala kroz kabl je 2/3 brzine prostiranja svetlosti u vakuumu. Koliko bitova može da stane u kabl?
33. Pretpostavimo da na protokol 4 želite da primenite model mašine konačnih stanja. U koliko stanja može da bude svaki od računara? U koliko stanja može da bude komunikacioni kanal? U koliko stanja može da bude ceo sistem (dva računara i kanal)? Zanemarite greške kontrolnih zbirova.
34. Napišite redosled aktiviranja mreže Petri sa slike 3-23 koji redom odgovara njenim stanjima (000), (01A), (01-), (010), (01A) na slici 3-21. Objasnite recima šta sek- venca predstavlja.

35. Nacrtajte mrežu Petri definisanu sledec'im pravilima prelaza: *AC —> B, B —> AC, CD —»E* i *E —»* *CD.* Na osnovu nje nacrtajte graf stanja dostupnih iz početnog stanja *ACD.* Koju opštepoznatu koncepciju modeluju ova pravila?
36. PPP se umnogome oslanja na protokol HDLC, u kome se umetanjem bitova sprečava zabuna kada se u podacima slučajno nađe sekvenca indikatorskog bajta. Navedite barem jedan razlog za to što se u protokolu PPP, umesto umetanja bitova, koristi umetanje bajtova.
37. Koliki se minimalan višak bitova mora upotrebiti da bi se IP paket poslao protokolom PPP? Računajte samo bitove uvedene protokolom PPP (ne i zaglavlje IP paketa).
38. U ovoj vežbi treba da ugradite mehanizam za otkrivanje grešaka pomoću standardnog algoritma CRC koji je opisan u tekstu. Napišite dva programa: program za gene- risanje (generator) i program za proveru (verifier). Generatorski program očitava sa stadnardnog ulaza *n*-bitnu poruku kao niz nula i jedinica koje predstavljaju red običnog teksta. Drugi red je /c-bitni polinom, kodiran takođe kao običan tekst. Generatorski program svoj rezultat ispisuje na standardnom izlazu kao red teksta od $n + k$ nula i jedinica, i to je poruka koju treba poslati. Zatim on ispisuje polinom, onako kako ga je očitao. Program za proveru očitava rezultate generatorskog programa i ispisuje poruku o tome da li su oni tačni ili nisu. Na kraju, napišite program za menjanje (alter) koji ce invertovati jedan bit u prvom redu u zavisnosti od prosleđe- nog argumenta (bitovi se u redu broje sleva, počinjući od jedinice), ali će sve ostalo kopirati verno. Kada upišete

    generator <file | *verifier*

    treba da dobijete odgovor daje poruka ispravna, ali kada upišete

    generator <file | alter argument | verifier treba da dobijete poruku o grešci.
39. Napišite program koji simulira ponašanje mreže Petri. Program treba da učitava pravila prelaza i listu stanja koja obuhvataju situacije kada mrežni sloj šalje ili prima nov paket. Iz početnog stanja, koje se takođe učitava, program treba da nasumično bira omogućena stanja i da ih aktivira proveravajući da li računar ikada prima 2 paketa zaredom, a da suprotna strana u međuvremenu ne emituje paket.

4

PODSLOJ ZA UPRAVLJANJE PRISTUPOM MEDIJUMIMA

Kao što smo istakli u prvom poglavlju, mreže se mogu svrstati u dve kategorije: one u kojima se koriste veze od tačke do tačke i one u kojima se koriste kanali za neusmereno (difuzno) emitovanje.

U svakoj mreži u kojoj se koristi neusmereno emitovanje, glavni problem je kako odrediti ko će koristiti kanal u situaciji kada ima više takmaca. Bolje ćete ovo razu- meti ako zamislite sastanak šestoro ljudi koji se odvija telefonskim putem, pri čemu svaki učesnik može da čuje sve druge sagovornike i da razgovara sa svakim od njih. Kada jedan od njih prestane da govori, najverovatnije će istovremeno početi da govore dva ili više drugih učesnika, što proizvodi totalnu zbrku. Na sastancima koji se odvijaju uživo, zbrka se izbegava korišćenjem spoljnih sredstava, na primer, tako što učesnici podižu ruke da bi dobili dozvolu da govore. Međutim, ako raspolažete samo jednim kanalom, mnogo je teže odrediti ko treba sledeći da emituje. Za rešenje po- menutog problema postoje mnogi protokoli i o njima govorimo u ovom poglavlju. U literaturi se kanali za neusmereno emitovanje ponekada nazivaju **kanali za višekorisnički pristup** (engl. *multiaccess channels)* ili **kanali za slobodan (slučajan) pristup** (engl. *random access channels).*

Protokoli kojima se određuje sledeći korisnik takvog kanala pripadaju podsloju sloja veze podataka, poznatom kao podsloj za **upravljanje pristupom medijumima** (engl. *Medium Access Control, MAC).* Podsloj MAC je posebno važan za lokalne mreže u kojima se za komuniciranje najčešće koristi kanal sa slobodnim pristupanjem.

Za razliku od njih, u regionalnim mrežama se koriste veze tipa od tačke do tačke, osim kada su satelitske. Pošto su kanali za slobodno pristupanje i lokalne mreže međusobno tako srodni, u ovom poglavlju ćemo razmotriti lokalne mreže u glavnim crtama, obu- hvatajući ponekada i ono što nije u direktnoj vezi sa podslojem MAC.

Podsloj MAC u tehničkom smislu predstavlja donji deo sloja veze podataka, pa bi bilo logično da je opisan u 3. poglavlju - pre bilo kojeg protokola od tačke do tačke. Ipak, protokoli koji povezuju više korisnika razumljiviji su kada se pravilno shvate protokoli koji povezuju samo dva korisnika. Samo zbog toga smo ovde neznatno odstupili od prikazivanja protokola strogim redosledom odozdo nagore.

4.1 PROBLEM DODELJIVANJA KANALA

Centralnu temu ovog poglavlja predstavlja način dodeljivanja jedinstvenog kanala za neusmereno emitovanje u situaciji kada na njega pretenduje više korisnika. Najpre ćemo u opštim crtama razmotriti statičke i dinamičke šeme dodeljivanja, a zatim objasniti više konkretnih algoritama.

4.1.1 Statičko dodeljivanje kanala u lokalnim i gradskim mrežama

Klasičan način dodeljivanja jedinstvenog kanala, kao što je telefonski vod, jednom od pretendenata na njega jeste multipleksiranje podelom frekvencija (FDM). Ako postoji *N* potencijalnih korisnika, propusni opseg se deli na iV jednakih delova (slika 2-31), po jedan za svakog korisnika. Pošto tako svaki korisnik ima svoje privatno frekventno područje, oni se međusobno ne ometaju. Kada je u pitanju mali i nepro- menljiv broj korisnika, od kojih svaki ima gust (baferovan) saobraćaj (npr. u lokalnim telefonskim centralama), FDM je

jednostavno i efikasno rešenje.

Međutim, kad je broj pošiljalaca veliki i stalno se menja ili se saobraćaj odvija u rafalima, tehnika FDM ne zadovoljava. Ako se propusni opseg izdeli na *N* područja, a u nekom trenutku želi da komunicira manje od *N* korisnika, veliki deo propusnog opsega ostaće neiskorišćen. Ako, pak, želi da komunicira više od *N* korisnika, neki od njih to neće moći, čak i u slučaju da neki od korisnika kojima su frekventna područja dodeljena ništa ne emituju niti primaju.

Čak i uz pretpostavku da bi se broj korisnika nekako mogao držati konstantnim, deljenje jedinstvenog kanala na statičke potkanale neefikasno je samo po sebi. Osnovni problem je to što se nepotrebno zauzima deo propusnog opsega rezervisan za korisnika koji ne razmenjuje podatke. On ga ne koristi, ali ga ni drugi ne mogu koristiti. Staviše, u većini računarskih sistema saobraćaj je izuzetno neredovan (odnos maksimalnog i prosečnog saobraćaja od 1000:1 sasvim je uobičajen). Zbog svega toga, većina kanala tokom najvećeg dela vremena ne radi ništa.

Jednostavan proračun izveden na osnovu teorije svrstavanja u redove čekanja (engl. *queueing theory)* jasno ukazuje na loše performanse statičkih FDM kanala. Zamislimo da u kanalu brzine *C* b/s postoji vremenska zadrška *T,* daje brzina pristizanja *X* okvira u sekundi i da dužine okvira slede gustinu verovatnoće - eksponencijalnu funkciju čija je srednja vrednost 1/p, bita po okviru.

Uz ove parametre, brzina pristizanja je *X* okvira u sekundi, a brzina usluživanja (engl. *service rate)* iznosi pC okvira u sekundi. Na osnovu teorije svrstavanja u redove čekanja može se pokazati da za pristizanje i usluživanje okvira koji se odvijaju po Poasonovom (Poisson) zakonu važi

Na primer, ako je C = 100 Mb/s, ako srednja dužina okvira 1/p iznosi 10.000 bitova, i ako brzina pristizanja okvira *X* iznosti 5000 okvira u sekundi, onda je *T* = 200 ps. Da smo zanemarili zadršku zbog svrstavanja u red čekanja i izračunali samo vre- me slanja 10.000 bitova kroz mrežu brzine 100 Mb/s, dobili bismo (netačan) rezultat od 100 ps. On bi važio samo kada na kanalu nema konkurencije.

Podelimo sada jedinstven kanal na *N* nezavisnih potlcanala, svaki s kapacitetom *C/B* b/s. Srednja brzina predavanja podataka svakom kanalu bice *X/N.* Kada *T*izradu- namo ponovo, dobij amo

$$= vh, = ^{NT} \qquad w$$

Prosečno kašnjenje koje se dobija multipleksirarvjem podelom frekvencije, *N* puta je veće od kašnjenja koje se dobija kada su svi okviri na neki čaroban način svrstani u veliki centralni red čekanja.

Sve što je rečeno za FDM, važi i za multipleksiranje podelom vremena (TDM). Svakom korisniku se statički dodeljuje jedan od *N* vremenskih intervala. Ako ga korisnik ne upotrebi, on je protraćen. Isto bi se događalo kada bismo mrežu izdelili fizički. Nastavljajući naš prethodni primer, kada bismo mrežu propusnog opsega 100 Mb/s zamenili sa 10 mreža od po 10 Mb/s i statički svaku dodelili po jednom korisniku, prosečno kašnjenje bi sa 200 ps skočilo na 2 ms.

Pošto nijedna od klasičnih metoda dodeljivanja kanala ne može uistinu da se izbori sa saobraćajem koji se odvija u rafalima, ispitacemo dinamičke metode.

4.1.2 Dinamičko dodeljivanje kanala u lokalnim i gradskim mrežama

Pre nego što pređemo na prvu od brojnih metoda dodeljivanja kanala opisaih u ovom poglavlju, treba da detaljno definišemo problem dodeljivanja. U osnovi svega leži pet osnovnih pretpostavki koje navodimo u nastavku.

1. **Model stanica.** Model sadrži /V nezavisnih **stanica** (na primer, računara, telefona ili ličnih komunikacionih uređaja) koje generišu okvire za slanje, bilo da to radi program ili korisnik. Stanice se ponekada nazivaju **terminali.** Ver- ovatnoća generisanja okvira u vremenskom intervalu A*t* iznosi *XAt,* gde je *X* konstanta (brzina pristizanja novih okvira). Kada generiše okvir, stanica se blokira sve dok okvir ne bude uspešno poslat.
2. **Pretpostavka o jedinstvenom kanalu.** Za sve komuniciranje na raspolaganju je samo jedan kanal. Stanice mogu da emituju samo preko njega i samo preko njega da primaju okvire. U hardverskom smislu sve stanice su jednake, iako protokoli nekima od njih mogu da dodele prioritete.

3. **Pretpostavka o sukobljavanju.** Ako se dva okvira istovremeno emituju, oni se vremenski preklapaju, što rezultuje izobličenim signalom. Taj događaj se naziva **sukobljavanje** (engl. *collision)*. Sve stanice mogu da otkriju sukobljavanja. Okvir koji se sukobio s drugim okvirom mora biti ponovo poslat.
Pri prenosu nema drugih grešaka osim zbog sukobljavanja okvira.
4. **Neprekidan vremenski tok.** Paket se može poslati u bilo kom trenutku. Ne postoji centralni sistemski sat koji bi vreme delio u intervale određene veličine.
.5. **Raspodeljeno vreme.** Vreme je podeljeno u intervale određene veličine. Slanje se uvele podudara s početkom intervala. Vremenski interval može sadržati 0, 1 ili više okvira, što redom odgovara praznom intervalu, uspešno poslatom okvim, odnosno sukobljavanju.
6. **Osluškivanje saobraćaja na nosiocu podataka.** Pre nego što sama upotrebi kanal, stanica može da proveri da lije slobodan. Ako ustanovi da kanal neko već koristi, stanica neće emitovati sve dok se kanal ne isprazni.
7. **Nema osluškivanja saobraćaja na nosiocu podataka.** Stanice ne provera- vaju da li je kanal prazan, već odmah emituju okvire. Tek kasnije mogu da utvrde da li je prenos obavljen uspešno.

Navedene pretpostavke treba objasniti. Prva je pretpostavka o nezavisnim stanicama koje generišu okvire za slanje ravnomernom brzinom. Podrazumeva se da na svakoj stanici postoji samo jedan program ili samo jedan korisnik - kada je stanica blokirana, okviri se uopšte ne generišu. Složeniji modeli predviđaju stanice s više programa koje mogu generisati okvire i kada je stanica blokirana, ali je analiza takvih stanica veoma složena.

Pretpostavka o jedinstvenom komunikacionom kanalu sama je srž modela. Nema drugih sredstava komunikacije. Stanice ne mogu „da podignu dva prsta“ da bi ih „nastavnik prozvao“.

Suštinska je i pretpostavka o sukobljavanju, iako ona u nekim sistemima (na pri- mer, gde se koristi širenje spektra) nije tako stroga, što dovodi do iznenađujućih rezultata. Isto tako, u nekim lokalnim mrežama (npr. mrežama tipa token ring), stanice jedna drugoj prosleđuju specijalan žeton koji im u trenutku kada ga imaju dozvoljava da emituju. U narednim odeljcima ipak ćemo se držati pretpostavke o jedinstvenom kanalu s konkurentskim stanicama i modelom sukobljavanja okvira.

Za korišćenje vremena mogu postojati dve granične pretpostavke. Vremenski tok je ili neprekidan (4a) ili izdeljen u intervale konačne dužine (4b). Jedna se koristi u jednim, a draga u dragim sistemima, tako da ćemo razmotriti obe. Za konkretan sistem može važiti samo jedna od njih.

Slično tome, stanice mogu ili da osluškuju (5a) ili da ne osluškuju saobraćaj na nosiocu podataka (5b). U lokalnim mrežama saobraćaj se uglavnom osluškuje. Međutim, to se u bežičnim mrežama ne može izvesti efikasno jer ne mogu sve stanice biti u dometu svih drugih stanica. Stanice u kablovskim mrežama u kojima se saobraćaj osluškuje mogu da utvrde da li se sukobljavaju s dragim stanicama. Otkrivanje sukobljavanja se iz tehničkih razloga retlco primenjuje u bežičnim mrežama. Imajte na umu da pod „nosiocem podataka“ ovde podrazumevamo električni signal koji se prostire kablom, što nema ničeg zajedničkog s telefonskim sistemom prenosa koji se nije promenio još od dana kada je zamenio golubove pismonoše.

4.2 PROTOKOLI ZA VIŠEKORISNIČKI PRISTUP

Postoji više algoritama za dodeljivanje zajedničkog kanala većem broju korisnika. U narednim odeljcima proueićemo one najzanimljivije, zajedno s primerima njihovog korišćenja.

4.2.1 ALOHA

Norman Abramson i njegovi saradnici s Havajskog univerziteta izumeli su sedamdesetih godina novu, elegantnu metodu za rešavanje problema pristupanja kanalu. Od tada (Abramson, 1985), njihov rad su unapredili mnogi istraživači. Iako je u Abram- sonovom sistemu ALOHA korišceno radiodifuzno emitovanje sa zemlje, osnovna ideja je primenljiva na svaki sistem u kome se nezavisni korisnici međusobno nadmeću za korišćenje jedinstvenog zajedničkog kanala.

Ovde ćemo razmotriti dve verzije sistema ALOHA: čistu i vremenski raspodeljenu. One se razlikuju u pogledu toga da li je vreme izdeljeno u intervale konačne dužine u koje moraju da se uklapaju svi okviri ili nije. Čista ALOHA ne zahteva sinhronizo- vanje globalnog vremena; za vremenski raspodeljenu varijantu to je neophodno.

Čista ALOHA

Ideja koja leži u osnovi sistema ALOHA veoma je jednostavna: dozvoliti korisnicima da emituju uvek kada imaju podatke za slanje. Naravno da će u tom slučaju biti sukobljavanja i da će se sukobljeni okviri izobličavati. Međutim, zahvaljujući tome što se pri neusmerenom emitovanju generišu povratne informacije, pošiljalac uvek može ustanoviti da li je njegov okvir uništen ako osluškuje kanal, kao i drugi korisnici. U lokalnim mrežama, povratne informacije stižu trenutno; u satelitskim mrežama, pošiljalac tek nakon 270 ms saznaje da li je njegov okvir primljen na odredištu. Ako iz nekog razloga nije moguće osluškivati tokom emitovanja, neophodne su potvrde o prijemu okvira. Kada se poslati okvir ošteti, pošiljalac će čekati potvrdu tokom slučajno odabranog vremenskog intervala i jednostavno ga poslati ponovo. Vreme čekanja mora biti nasumično odabrano inače će se isti okviri sudarati neprestano na isti način. Sistemi u kojima više korisnika dele isti kanal na način koji može dovesti do sukobljavanja, široko su poznati kao **konkurentski** (engl. *contention)* sistemi.

Na slici 4-1 skicirano je generisanje okvira u sistemu ALOHA. Svi prikazani okviri su iste veličine jer je to neophodno za postizanje maksimalnog protoka u ovom sistemu.

Kad god dva korisnika pokušaju da zauzmu kanal u istom trenutku, dolazi do su- kobljavanja i okviri se oštećuju. Čak i ako se samo prvi bit novog okvira preklopi s poslednjim bitom prethodnog okvira, oba okvira propadaju i moraju se kasnije ponovo poslati. Kontrolni zbir ne može (i ne treba) da vodi računa o tome da lije došlo do potpunog ili samo neznatnog preklapanja okvira - greška ostaje greška.

U tom smislu, zanimljivo je pitanje kolika je efikasnost ALOHA kanala. Drugim recima, koji deo poslatih okvira izbegne sukobljavanje u ovim haotičnim okolnostima? Zamislimo najpre beskonačan skup interaktivnih korisnika koji sede za svojim računalima (stanicama). Svaki korisnik se uvek nalazi u jednom od dva stanja: ili unosi podatke ili čeka odgovor. Na početku, svi korisnici unose podatke. Kada unesu red podataka, prelaze u stanje čekanja na odgovor. Stanica tada emituje okvir s redom podataka i proverava na kanalu da li je to uspešno obavljeno. Ako su podaci uspešno poslati, korisnik dobija odgovor i unosi nov red podataka. Ako odgovor ne stigne, korisnik nastavlja da čeka, a okvir se stalno ponovo šalje, sve dok ne bude uspešno poslat.

Slika 4-1. U čistom sistemu ALOHA okviri se emituju u potpuno slučajno odabranim vremenskim trenucima,

Definišimo „jedinično vreme prenosa okvira“ kao vreme potrebno da se pošalje standardni okvir fiksne dužine, tj. podelimo dužinu okvira brzinom prenosa u bitovima. Pretpostavićemo da beskonačna populacija korisnika generiše nove okvire sledeci Poasonovu distribuciju s prosečno N proizvedenih okvira tokom vremena potrebnog za prenos jednog okvira. (Pretpostavka o beskonačnoj populaciji korisnika neophodna je da se N ne bi smanjilo kada se korisnici blokiraju.) Ako je $N > 1$, korisnici generišu okvire brže nego što kanal može da ih obradi i skoro svaki okvir će se sukobiti. Protok podataka odvijače se u nekim razumnim granicama pod uslovom $0 < N < 1$.

Osim novih okvira, stanice ponovo emituju i okvire koji su se prethodno sukobili. Pretpostavimo dalje da verovatnoca k pokušanih prenosa (i starih i novih okvira) u vremenu potrebnom za prenos jednog okvira, takođe sledi Poasonovu raspodelu, s prosečnom vrednošču G. Jasno je da je $G > N$. Pri niskom opterećenju $(N \sim 0)$, biće malo sukobljavanja, pa i malo ponovljenih okvira, tako daje $G \sim N$. Pri visokom opterećenju kanala biće mnogo sukobljavanja, pa je $G > N$. Pri svakom opterećenju, protok podataka S biće jednak proizvodu opterećenja G i verovatnoće uspešnosti slanja P_Q. $S = GP_Q$, gde je P_0 verovatnoca da se okvir na putu neće sukobiti.

Okvir se neće sukobiti ako nijedan drugi okvir ne bude poslat unutar jediničnog vremena prenosa merenog od početka slanja okvira, kao što je prikazano na slici 4-2. Pod kojim uslovima će zasenčeni okvir stići neoštećen? Neka t bude vreme potrebno za slanje okvira. Ako ijedan korisnik emituje okvir u vremenskom intervalu između t_0 i $t.Q + t$, kraj tog okvira sukobiće se s početkom zasenčenog okvira. U stvari, sudbina zasenčenog okvira zapečaćena je i pre nego što se pošalje njegov prvi bit, ali pošto u čistom sistemu ALOHA stanice ne osluškuju kanal pre emitovanja, nema načina da saznaju da se na njemu već nalazi neki okvir. Slično tome, s krajem zasenčenog okvira sukobiće se i svaki okvir emitovan u vremenskom intervalu između $t_0 + 1$ i $t_0 + 21$.

Sukobljava se s početkom zasenčenog okvira

Sukobljava se s krajem zasenčenog okvira

t,+ t

t + 2t t + 3 t Vreme -

-Interval ranjivosti -

Slika 4-2. Interval ranjivosti zasenčenog okvira.

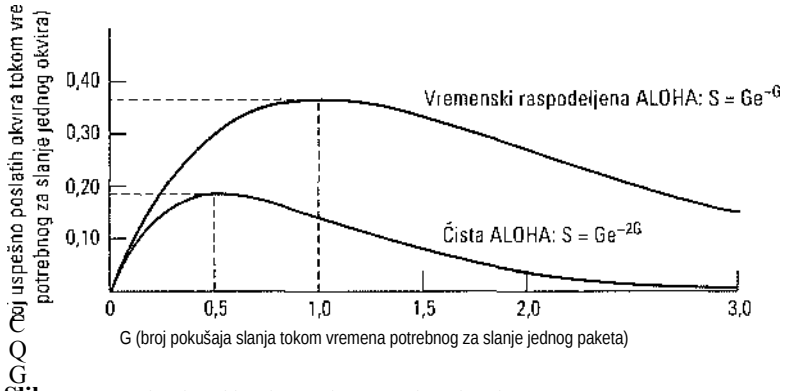
Verovatnoća da će *k* novih okvira biti generisano tokom slanja datog okvira dobija se iz Poasonove raspodele:

$$\Pr[Jt] \quad \frac{G e^{-G}}{k\backslash} \tag{4-2}$$

tako da verovatnoća generisanja 0 okvira iznosi e^{-G}. U intervalu dvaput dužem od vremena slanja okvira, generisaće se prosečno $2G$ novih okvira. Prema tome, verovatnoća da tokom intervala ranjivosti neće biti generisan nijedan nov okvir, iznosi $P_0 = e^{-2G}$. Imajući u vidu da je $S = GP_0$, dobijamo

$$\hat{S} = Ge^{-2G}$$

Odnos između pokušanog saobraćaja i stvarnog protoka podataka prikazan je na slici 4-3. Maksimalan protok podataka omogućen je pri $G = 0{,}5$, uz $S = 1/2e$, što iznosi oko 0,184. Drugim recima, maksimalno očekivano islcorišćenje kanala iznosi 18%. Takav rezultat baš ne ohrabruje, ali ako dopustimo da svako emituje kada želi, teško da možemo očekivati stoprocentan uspeh.



Slika 4-3. Stvarni saobraćaj kanalom u odnosu na pokušani za sisteme ALOHA.

Vremenski raspodeljena ALOHA

Godine 1972. Roberts je objavio metodu dupliranja kapaciteta sistema ALOHA (Roberts, 1972). On je predložio da se vreme izdeli u intervale konačne dužine i da svaki interval odgovara jednom okvira. Takav pristup podrazumeva dogovaranje korisnika oko granica vremenskih intervala. Jedan način da se to postigne bio bi da posebna stanica emituje odgovarajući signal na početku svakog intervala, slično satu.

U Robertsovoj metodi, poznatoj kao **vremenski raspodeljena** ALOHA (engl. *slotted ALOHA),* za razliku od Abramsonovog **čistog sistema** ALOHA (engl. *pure ALOHA),* računam je zabranjeno da emituje kad god se unese znak za novi red. Tada je dužan da sačeka početak sledećeg vremenskog intervala. Na taj način, kontinualna čista ALOHA pretvorena je u diskretnu. Posto je period ranjivosti tako prepolovljen, verovatnoća da neće biti dragog saobraćaja tokom slanja našeg probnog okvira iznosi $e^{\sim G}$, što daje

$$S = Ge^{\circ}$$

Kao što vidite na slici 4-3, vremenski raspodeljena ALOHA dostiže maksimum efikasnosti pri $G = 1$, uz protok podataka 5 = l/e ili oko 0,368 - dva puta više od čistog sistema ALOHA. Ako sistem radi pri $G = 1$, verovatnoća nastanka praznog vremenskog intervala je 0,368 (prema slici 4-2). Najviše što možemo da očekujemo od vremenski raspodeljenog sistema ALOHA jeste 37% praznih vremenskih intervala, 37% uspešno prenetih okvira i 16% sukobljenih okvira. Radom pri višim vrednostima G smanjuje se broj praznih intervala, ali se broj sukoba eksponencijalno povećava. Da biste se uverili kako broj sukoba brzo raste sa G, razmotrite slanje probnog okvira. Verovatnoća da će on izbeći sukobljavanje iznosti $e^{\sim G}$, a to je verovatnoća da svi ostali korisnici miruju u tom vremenskom intervalu. Verovatnoća sukobljavanja je onda 1 - $e^{\sim G}$. Verovatnoća da će okvir biti uspešno poslat posle k pokušaja (k - 1 sukoba, a zatim uspešno slanje) iznosi

$$P_k = {}_{e}\text{-C(l\_e-G)*-i}$$

Očekivan broj slanja E po jednom znaku za nov red (engl. *carriage return)* tada iznosi

$$E = J_j k P_k = \pounds\, k e^{-G}(1-e^{-G})^{k}\text{-}' = e^{G}\, k = 1 \qquad k = 1$$

Sto E eksponencijalno zavisi od G, malo povećanje opterećenja kanala može drastično da ugrozi njegove performanse.

Vremenski raspodeljena ALOHA je važna zbog nečega što nije očigledno. Protokol je razvijen sedamdesetih godina, korišćen u nekoliko eksperimentalnih sistema, a potom skoro zaboravljen. Kada je stvoren kablovski Internet, odjednom se pojavio problem dodeljivanja zajedničkog kanala jednom od mnogih pretendenata i vremenski raspodeljena ALOHA je ponovo izvučena na svetlost dana. Često se dešava da potpuno ispravni protokoli prestanu da se koriste iz čisto političkih razloga (npr. velika kompanija želi da svi rade na njen način), ali mnogo godina kasnije neka pametna glavica shvati da takav odbačeni protokol rešava problem s kojim se trenutno bakće.

Zbog toga ćemo u ovom poglavlju proučiti više elegantnih protokola koji nisu baš u širokoj upotrebi, ali se lako može desiti da budu iskorišćeni u budućim aplikacijama, pod uslovom da ih se dovoljno projektanata još uvele seća. Naravno, proučićemo i protokole koji se danas široko koriste.

4.2.2 Protokoli za višekorisnički pristup uz osluškivanje saobraćaja na nosiocu podataka

Vremenski raspodeljenim sistemom ALOHA može se postići maksimalno isko- rišćenje kanala jednako *l/e.* To nas ne iznenađuje previše, jer uz stanice koje emituju po sopstvenoj želji, ne obraćajući pažnju na druge stanice, mora da bude mnogo sukobljavanja. Međutim, u lokalnim mrežama stanica može da utvrdi šta rade druge stanice i da shodno tome podesi svoje ponašanje. U takvim mrežama iskorišćenje može biti znatno veće od l/e. U ovom odeljku pozabavićemo se protokolima koji poboljšavaju efikasnost rada kanala.

Protokoli koji predviđaju osluškivanje nosioca podataka (tj. saobraćaja na njemu) i ponašanje stanica shodno prikupljenim informacijama nazivaju se **protokoli za pristupanje uz osluškivanje saobraćaja na nosiocu podataka** (engl. *carrier sense protocols).* Predloženo je više takvih protokola od kojih su neke detaljno analizirali Kleinrock i Tobagi (1975). U nastavku ćemo pomenuti više njihovih verzija.

CSMA protokoli s trajnim i povremenim osluškivanjem kanala

Prvi protokol za pristup uz osluškivanje nosioca podataka o kome ćemo govoriti zove se **1-trajni CSMA** (engl. 1 *-persistent CSMA, l-persistent Carrier Sense Multi- ple Access).* Kada stanica ima podatke za slanje, ona prvo oslušne kanal da bi utvrdila da lije zauzet. Ako na kanalu ima saobraćaja, stanica čeka da on utihne, pa tek onda šalje svoj okvir. Kada dođe do sukobljavanja, stanica čeka tokom nasumično odabranog perioda i sve počinje od početka. Protokol nosi ime 1-trajni, zato što stanica emi- tuje s verovatnoćom 1 kada utvrdi da je kanal prazan.

Brzina prostiranja signala bitno utiče na performanse protokola. Postoji mala šansa da neposredno po početku emitovanja jedne stanice druga stanica bude spremna da šalje podatke i da osluškuje kanal. Ako signal prve stanice još nije stigao do nje, ona će smatrati daje kanal slobodan i počeće da emituje, što će izazvati sukobljavanje. Što signal duže putuje od jedne do druge stanice, opisani efekat postaje sve značajniji i, shodno tome, performanse protokola slabe.

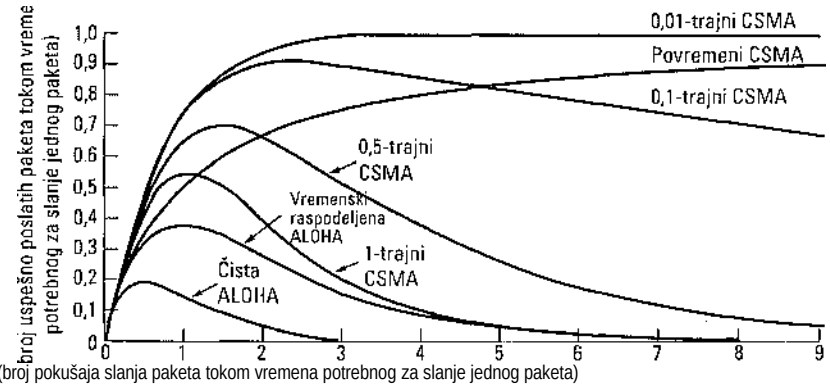
Čak i kada bi se signal prostirao beskonačnom brzinom, i tada bi bilo sukobljavanja. Ako dve stanice istovremeno budu spremne za slanje u trenutku dok emituje treća stanica, obe će uljudno čekati da se to emitovanje završi, a zatim istovremeno početi da emituju, što odmah izaziva sukobljavanje. Da nisu tako nestrpljive, bilo bi manje sukoba. No, i pored toga, ovaj protokol je mnogo bolji od čistog sistema ALOHA jer su pomenute dve stanice dovoljno pristojne da ne ometaju emitovanje treće stanice. Intuitivno možemo zaključiti da će takav pristup poboljšati performanse u odnosu na čist sistem ALOHA. Isto važi i za vremenski raspodeljen sistem ALOHA.

Dragi protokol za pristup uz osluškivanje nosioca podataka zove se **povremeni CSMA** protokol (engl. *nonpersistent CSMA).* Kod njega je učinjen svestan napor da se ograniči pohlepa radnih stanica. Tako, pre nego što počne da emituje, stanica osluškuje kanal. Ako utvrdi da na njemu nema nikoga, ona počinje da šalje podatke. Međutim, ukoliko utvrdi da neko već emituje, stanica neće neprekidno osluškivati kanal da bi ga zgrabila čim taj neko završi emisiju, već će ga ponovo oslušnuti tek nakon nasumično odabranog vremenskog intervala. Zbog toga se kanal bolje iskorišćava u odnosu na
1- trajni CSMA protokol, ali je vremenska zadrška između pojedinih okvira veća.

Poslednji je **p-trajni CSMA** protokol (engl. *p-persistent CSMA).* On se primenjuje u vremenski raspodeljenim kanalima i radi na sledeći način. Kada je stanica spremna za slanje podataka, ona osluškuje kanal. Ukoliko utvrdi da je prazan, ona emituje s ve- rovatnoćom*p.* S verovatnoćom $q = 1 - p$ ona odustaje do sledećeg vremenskog intervala. Ako je i taj interval nezauzet, ona emituje ili ponovo odustaje, s verovatnoćom *p,* odnosno *q.* Taj proces se ponavlja sve dok okvir ne bude poslat ili dok druga stanica ne počne da emituje. U ovom

dragom slučaju, stanica koja baš nema sreće reaguje kao da je došlo do sukobljavanja (povlači se, čeka nasumično odabran period vremena i sve počinje od početka). Ako stanica na početku utvrdi da je kanal zauzet, ona čeka sledeći vremenski interval da bi ponovo primenila opisani algoritam. Na slici 4-4 prikazan je izračunati stvarni protok podataka u odnosu na ponuđeni saobraćaj, za sva tri protokola, kao i za čiste, odnosno vremenski raspodeljene sisteme ALOHA.

co
c

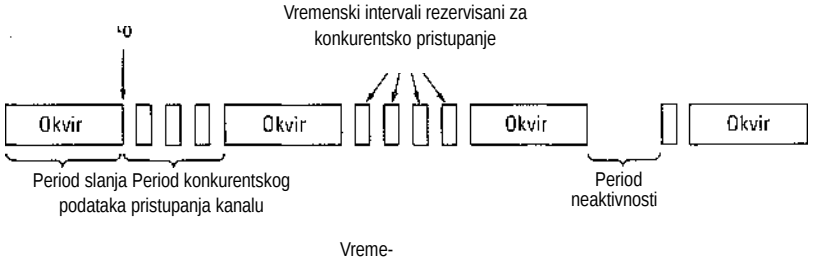


Slika 4-4. Iskorišćenje kanala u funkciji opterećenja, uz različite protokole za slobodno pristupanje.

Protokol CSMA uz otkrivanje sukoba

Trajni i povremeni CSMA protokoli predstavljaju vidljivo poboljšanje u odnosu na sistem ALOHA jer kod njih nijedna stanica ne emituje ako utvrdi da je kanal zauzet. Dodatno poboljšanje bi bilo da stanica prekine emitovanje čim utvrdi da je došlo do sukobljavanja. Dragim recima, ako dve stanice utvrde da je kanal prazan i počnu istovremeno da emituju, obe će skoro u istom trenutku otkriti sukobljavanje. Umesto da dovrše slanje svojih okvira, koji su ionako nepopravljivo izobličeni, one treba da prekinu emitovanje čim otkriju sukob. Pravovremenim prekidanjem slanja izobličenih okvira štedi se vreme i propusni opseg. Taj protokol, poznat kao **CSMA uz otkrivanje sukoba** (engl. *CSMA with Collision Detection, CSMA/CD),* široko se koristi u MAC sloju lokalnih mreža. On predstavlja i osnovu popularne lokalne Ethernet mreže, tako da ga vredi ukratko objasniti.

U protokolu CSMA/CD, kao i u mnogim drugim protokolima za lokalne mreže, koristi se koncepcija modela sa slike 4-5. U trenutku /Q stanica je dovršila slanje okvira. Sada bilo koja druga stanica koja ima okvir spreman za slanje može to i da učini. Ako dve ili više stanica reše da istovremeno počnu emitovanje, biće sukobljavanja. Sukobi se mogu otkriti ako se prati intenzitet ili širina impulsa primljenog signala i uporedi s poslatim signalom.



Slika 4-5. Protokol CSMA/CD može da bude u jednom od tri stanja: konkurentskog pristupanja, slanja podataka ili neaktivnosti.

Nakon što stanica otkrije sukob, ona prekida emitovanje, čeka tokom proizvoljno odabranog vremenskog intervala i tada pokušava da ponovo emituje, pretpostavljajući da u međuvremenu nijedna druga stanica nije započela emitovanje. Prema tome, naš model protokola CSMA/CD sastojaće se od naizmeničnih pokušaja pristupanja kanalu i perioda slanja podataka, uz povremenu neaktivnost kada sve stanice miruju (npr. nemaju podataka za slanje).

Osmotrimo sada detalje algoritma za konkurentsko pristupanje stanica kanalu. Pretpostavimo da u trenutku f_0 istovremeno počinju da emituju dve stanice. Koliko će im trebati da shvate daje došlo do sukobljavanja? Odgovor na ovo pitanje od primarnog je značaja za određivanje dužine perioda konkurentskog pristupanja i posledično, vremenske zadrške i protoka podataka. Minimalno vreme potrebno za otkrivanje sukoba jednako je vremenu putovanja od jedne stanice do drage.

Na osnovu ovakvog rezonovanja, mogli biste pomisliti da će stanica koja nakon početka emitovanja ne otkrije sukob tokom vremena potrebnog da signal prođe čitav kabl, biti sigurna da ga je ona „zauzela“, tj. da su sve drage stanice shvatile da ona emituje i daje neće ometati. Takav zaključak bio bi pogrešan. Razmotrimo šta se događa u sledećem, najnepovoljnijem slučaju. Neka je x vreme potrebno signalu da pređe put između dve međusobno najudaljenije stanice. U trenutku t_Q jedna stanica počinje da emituje. U trenutku x - e, samo malo pre nego što signal stigne u najudaljeniju stanicu, ta stanica takođe počinje da emituje. Naravno, ona gotovo odmah otkriva sukobljavanje i zaustavlja se, ali odjek izazvan sukobom stiže do prve stanice tek posle vremena 2x - e. Dragim recima, stanica u najnepovoljnijem slučaju može biti sigurna daje za sebe zauzela kanal tek ako ne otkrije sukobljavanje u vremenskom intervalu 2x. Iz tog razloga, interval konkurentskog pristupanja modelovacemo prema vremenski raspodeljenom sistemu ALOHA, sa širinom intervala 2x. U koaksijalnom kablu dužine 1 km, x iznosi približno 5 ps. Pretpostavićemo zbog jednostavnosti da svaki interval sadrži samo 1 bit. Kada jednom zauzme kanal, stanica može da emituje kojom god želi brzinom - nije ograničena na 1 bit tokom 2t sekundi.

Treba razumeti daje otkrivanje sukoba *analogan* proces. Hardver stanice mora da osluškuje kabl dok ona emituje. Ako se ono što čuje razlikuje od onoga što šalje, ona zna daje došlo do sukobljavanja. Iz ovoga sledi da signal mora biti tako kodiran da se sukobljavanje može otkriti (na primer, teško bi se otrilo sukobljavanje dva signala napona 0 V). Zbog toga se najčešće koristi specijalno kodiranje.

Treba naglasiti da stanica koja šalje podatke mora neprestano da osluškuje kanal, tražeći na njemu odjek mogućeg sukoba. Zbog toga je kanal koji radi prema protokolu CSMA/CD u stvari poludupleksni sistem. Stanica ne može istovremeno da šalje i da prima okvire jer se za traženje sudara tokom svakog slanja koristi logika primaoca.

Da bismo otklonili sve nedoumice, pomenimo i to da nijedan protokol podsloja MAC ne garantuje pouzdanu isporuku. Čak i kada nema sukoba, primalac iz raznoraznih razloga može da pogreši pri kopiranju okvira (npr. zbog nedostatka mesta u baferu ili propuštenog softverskog prekida).

4.2.3 Protokoli u kojima nema sukobljavanja

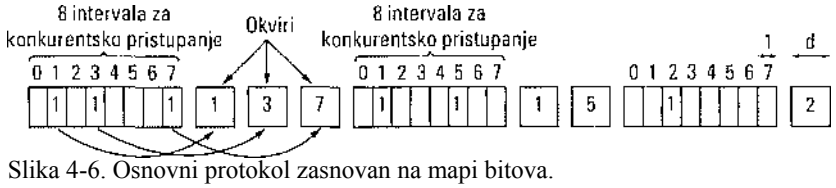
Iako kod protokola CSMA/CD više ne dolazi do sukobljavanja nakon što stanica nedvosmisleno zauzme kanal, sukobi se mogu događati tokom konkurentskog pristupanja. Ti sukobi se negativno odražavaju na performanse sistema, naročito ako je kabl dugačak (kada

je *x* veliko), a okviri uski. Isto tako, ni protokol CSMA/CD nije univerzalno primenljiv. U ovom odeljku ispitaćemo nekoliko protokola koji rešavaju pristupanje kanalu tako da uopšte ne dolazi do sukobljavanja, čak ni tokom perioda konkurencije. Oni se danas uglavnom ne koriste u većim sistemima, ali u ovoj oblasti koja se brzo menja dobro je imati u rezervi protokol izvanrednih mogućnosti.

U protokolima koje ćemo opisati smatramo da postoji tačno *N* stanica, svaka s jedinstvenom fiksnom adresom između 0 i *N*- 1. Nije važno to što će pojedine stanice u nekom periodu biti neaktivne. Smatramo takođe da je kašnjenje zbog konačne brzine prostiranja signala zanemarljivo. Osnovno pitanje i dalje je isto: koja stanica preuzima kanal po okončanju prethodne uspešne emisije? I dalje ćemo koristiti model sa slike 4-5, s njegovim ograničenim intervalima za konkurentsko pristupanje.

Protokol zasnovan na mapi bitova

U našem prvom protokolu kojim se izbegava sukobljavanje (engl. *collision-free protocol)* - **osnovnom protokolu zasnovanom na bit mapi** (engl. *basic bit-map pro- tocol),* svaki period konkurentskog pristupanja sadrži tačno *N* intervala. Ako stanica 0 ima okvir za slanje, ona tokom nultog intervala šalje bit 1. Nijednoj drugoj stanici nije dopušteno da emituje tokom tog intervala. Bez obzira na to šta radi stanica 0, stanica 1 dobija priliku da pošalje bit 1 tokom intervala 1, ali samo ako ima okvir u redu čekanja.
Stanica *j* može u načelu da objavi da ima okvir za slanje tako što će bit 1 umetnuti u interval *j*. Nakon što prođe svih *N* intervala, svaka je stanica u potpunosti obaveštena o tome koje stanice žele da emituju. U tom trenutku, one počinju da emituju numeričkim redosledom (slika 4-6).



Slika 4-6. Osnovni protokol zasnovan na mapi bitova.

Budući da se svi slažu u tome koje sledeći, nikada neće doći do sukobljavanja. Pošto i poslednja stanica koja ima spremne podatke pošalje svoj okvir, što je događaj koji lako zapažaju sve stanice, počinje nov IV-bitni period konkurentskog pristupanja. Ako stanica propusti svoj interval za pristupanje, mora da sačeka nov krug. Protokoli koji, slično opisanom protokolu, neusmereno emituju svoju nameru pre stvarnog slanja podataka, zovu se **protokoli s rezervisanjem vremena emitovanja** (engl. *reservationprotocols).*

Analizirajmo ukratko performanse ovog protokola. Pogodnosti radi, kao jedinicu za merenje vremena upotrebićemo jednobitni interval za konkurentsko pristupanje; okviri s podacima trajaće tada *d* vremenskih jedinica. U uslovima niskog opterećenja, mapa bitova će se stalno ponavljati između retkih okvira s podacima.

Razmotrimo situaciju s gledišta stanice čija je „adresa“ 0 ili 1. Kada ona bude spremna za emitovanje, aktuelni interval će se najčešće nalaziti negde u sredini mape bitova. Da bi počela da emituje, stanica će u prošeku morati da propusti *N/2* intervala kojima se završava aktuelni krug i još jedan pun krug od *N* intervala.

Izgledi stanica s višim adresama znatno su bolji. One će u načelu morati da sačekaju samo da se završi aktuelni krug (*N/2* jednobitnih intervala). Stanice s visokom adresom retlco moraju da čekaju sledeći krug. Pošto stanice s niskim adresama moraju u prošeku da čekaju

1,5*N* intervala, a stanice s visokim adresama *0,5N* intervala, srednja vrednost za sve stanice je *N* intervala. Lako možemo da izračunamo islcorišćenje kanala pri niskom opterećenju. Pošto „nekorisnih“ bitova po svakom okviru ima *N,* uz *d* bitova podataka iskorišćenje je *d/(N + d).*

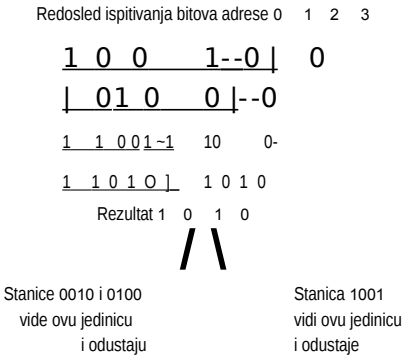
Pri visokom opterećenju, kada sve stanice neprestano imaju nešto da šalju, period konkurentskog pristupanja potpuno se raspodeljuje na *N* okvira, na svaki okvir dolazi samo jedan „nekoristan“ bit i iskorišćenje je *d/(d* +1). Kašnjenje okvira predstavlja zbir vremena koje okvir provede u redu čekanja na izvorištu i vremena *N(d* + l)/2 od trenutka kada je izbio na čelo reda.

Binarno odbrojavanje

Problem sa osnovnim protokolom zasnovanim na mapi bitova jeste njegov nekoristan teret (engl. *overhead)* od 1 bita po okviru, što počinje da pogoršava performanse na mrežama koje imaju na hiljade stanica. Umesto da dodeljujemo stanicama fiksne binarne adrese, možemo postupiti efikasnije. Kada stanica poželi da emituje, ona objavljuje svoju adresu u obliku binarnog niza, počinjući od najznačajnijeg bita. Sve adrese imaju istu dužinu. Bitovi na korespondentnim pozicijama adrese različitih stanica međusobno se podvrgavaju operaciji logičke disjunkcije (logičko ILI). Opisani protokol zvaćemo **binarno odbrojavanje** (engl. *binary countdown)',* on je iskorišćen za mrežu Datakit (Fraser, 1987). U protokolu se zanemaruje kašnjenje u prenosu, tako da sve stanice praktično trenutno vide potvrđene bitove.

Da bi se izbegli sukobi, mora se primeniti neko pravilo određivanja prvenstva: čim stanica u čijoj je adresi najznačajniji bit 0 ustanovi daje taj bit prebrisan bitom 1, ona se povlači. Na primer, ako stanice sa adresama 0010, 0100, 1001 i 1010 pokušavaju da istovremeno iziđu na kanal, u toku intervala predviđenog za prvi bit one emituju 0, 0,1, odnosno 1. Ti bitovi se podvrgavaju operaciji disjunkcije, dajući rezultat 1. Kada vide rezultat, stanice 0010 i 0100 znaju da na kanal pretenduju stanice s višim adresama i zato odustaju. Stanice 1001 i 1010 nastavljaju da konkurišu jedna drugoj.

Rezultat disjunkcije bitova na sledećoj poziciji je 0, tako da su obe stanice i dalje u igri. Treća disjunkcija daje 1, pa stanica 1001 odustaje. U ovoj rundi pobeđuje stanica 1010 jer ima najvišu adresu. Ona tada može da pošalje okvir s podacima, posle čega počinje nova runda takmičenja. Protokol je prikazan na slici 4-7. Njegovo svojstvo je da stanice s višim adresama imaju prioritet nad stanicama nižih adresa, što može da bude dobro ili loše, u zavisnosti od konteksta.



Slika 4-7. Protokol binarnog odbrojavanja. Crtica označava neaktivnost.

Efikasnost kanala koji radi po ovom protokolu iznosi $d/\{d + \log_2 A0$- Ako se, međutim,

izabere „inteligentan“ format okvira, tako da njegovo prvo polje predstavlja adresu pošiljaoca, čak se ni $\log_2 iV$ bitova ne troši uludo i iskorišćenje kanala postaje stoprocentno.

Mok i Ward (1979), opisali su varijantu binarnog odbrojavanja s paralelnim inter- fejsom umesto serijskog. Oni su predložili i korišćenje virtuelnih brojeva stanica koji rastu od 0, s tim što se uspešnoj stanici po završenom prenosu dodeljuje najniži broj

da bi se povećao prioritet stanica koje duže vreme nisu uspele da izađu na kanal. Na primer, ako stanice *C, H, D, A, G, B, F* imaju prioritete 7, 6, 5, 4, 3, 2, 1 i 0, onda uspešno emitovanje stanicu D stavlja na kraj liste, pa redosled prioriteta postaje C, *H, A, G, B, E, F, D.* Tako, *C* postaje virtuelna stanica 7, stanica A od 4 postaje 5, a *D* sa 5 pada na 0. Stanica *D* će sada moći da pristupi kanalu samo ako ga nijedna druga stanica u tom trenutku ne traži.

Binarno odbrojavanje je primer jednostavnog, elegantnog i efikasnog protokola koji čeka da ga neko ponovo otkrije. Nadamo se da če se to jednoga dana i dogoditi.

4.2.4 Protokoli sa ograničenom konkurencijom

Dosad smo razmotrili dve osnovne strategije za pristupanje kanalu u kablovskoj mreži: takmičenje stanica, kao kod protokola CSMA, i protokole kojima se izbegava sukobljavanje. Obe strategije se mogu oceniti na osnovu dva važna pokazatelja performansi: kašnjenja pri niskom opterećenju i iskorišćenja kanala pri visokom opterećenju. U uslovima lakog saobraćaja poželjnija je konkurencija stanica (tj. čista ili vremenski raspodeljena ALOHA), jer je kašnjenje manje. S povećanjem gustine saobraćaja metoda konkurencije gubi na efikasnosti jer odluka o dodeljivanju kanala postaje sve složenija, zbog čega raste promet „nekorisnih" podataka. Upravo suprotno važi za protokole kojima se sukobi izbegavaju. Pri niskom opterećenju, oni izazivaju veliko kašnjenje, ali kako opterećenje raste, povećava se i iskorišćenje kanala - suprotno od onoga što se dešava kod protokola s konkurencijom stanica.

Očigledno je da bi optimalan rezultat dala kombinacija najboljih svojstava jednih i drugih protokola. Kod takvog kombinovanog protokola, pri niskom opterećenju koristilo bi se konkurentsko pristupanje stanica zbog smanjenja kašnjenja, a pri visokom opterećenju - tehnike izbegavanja sukoba, u cilju što boljeg iskorišćenja kanala. Protokoli takve vrste, tzv. **protokoli sa ograničenom konkurencijom** (engl. *limited- -contention protocols),* zaista postoje i njima ćemo zaključiti proučavanje mreža u kojima stanice osluškuju nosilac podataka.

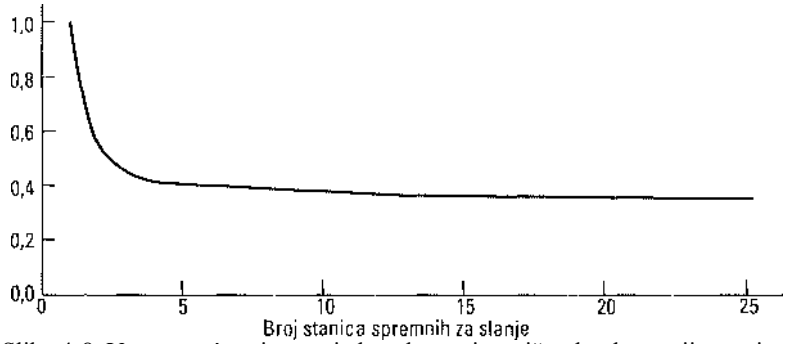
Svi konkurentski protokoli koje smo dosad proučavali jesu simetrični, tj. svaka stanica pokušava da zauzme kanal s verovatnoćom *p,* pri čemu je *p* jednako za sve stanice. Zanimljivo je da se ukupne performanse sistema ponekada mogu poboljšati ako se protokolom raznim stanicama dodeljuje različita verovatnoća pristupanja.

Pre nego što pređemo na asimetrične protokole, razmotrimo još jednom performanse simetričnih protokola. Pretpostavimo da se *k* stanica takmiči za pristupanje kanalu. Verovatnoća da bilo koja od njih pošalje okvir tokom bilo kog intervala iznosi *p.* Verovatnoća da će neka stanica uspešno zauzeti kanal tokom zadatog intervala iznosi $kp\{ 1 - p)^{k}$~ *. Da bismo izračunali optimalnu vrednost *p,* diferenciraćemo prethodni izraz u odnosu na *p,* rezultat izjednačiti s nulom i jednačinu rešiti po *p.* Kada to uradimo, nalazimo da optimalna vrednost *p* iznosi *l/k.* Zamenjujuci *p* = l//c u prethodni izraz, dobijamo

$$\text{Prfuspešnost pri optimalnom } p] = \qquad (4\text{-}4)$$

$$[-£_{\,i\text{-}rf\text{-}}\ \mathbf{i}$$

Ta verovatnoća je grafički prikazana na slici 4-8. Kada je broj stanica mali, šanse za uspeh su dobre, ali čim broj stanica pređe 5, verovatnoća pada na vrednost blisku asimptotskoj (l/c).



Slika 4-8. Verovatnoća pristupanja kanalu uz simetričnu konkurenciju stanica.

Slika 4-8 prilično jasno pokazuje da se verovatnoća da određena stanica pristupi kanalu može povećati samo smanjenjem konkurencije. Protokoli sa ograničenom konkurencijom upravo to rade. Oni najpre dele stanice u grupe (koje ne moraju biti razdvojene). Samo se članovima grupe 0 dozvoljava da konkurišu za interval 0. Ako neko od njih uspe da pristupi kanalu, može da pošalje okvir podataka. Ukoliko se interval propusti ili nastane sukobljavanje, tada članovi grupe 1 konkurišu za interval 1 itd. Svrstavanjem stanica u odgovarajuće grupe smanjuje se konkurencija za svaki interval i radni režim se pomera ka levom kraju dijagrama na slici 4-8.

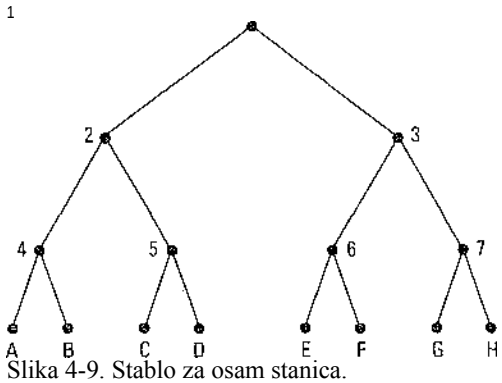
Najvažnije je da se stanice pravilno svrstaju u grupe. Pre nego što pređemo na opšti slučaj, pogledajmo neke specijalne slučajeve. Jedna krajnost je da u svakoj grupi bude po jedna stanica. Takva raspodela garantuje rad bez sukobljavanja jer najviše jedna stanica pretenduje na jedan interval. Slične protokole smo videli i ranije (npr. binarno odbrojavanje). Sledeći specijalan slučaj obuhvata dve stanice u svakoj grupi. Verovatnoća da će tokom datog intervala obe pokušati da ga zauzmu iznosi p^2; ako je p malo, ta verovatnoća je zanemarljiva. Što više stanica konkuriše za isti interval, verovatnoća sukobljavanja raste, ali se smanjuje dužina mape bitova koja svakome daje šansu. Druga krajnost je jedinstvena grupa u koju su svrstane sve stanice (npr. vremenski raspodeljena ALOHA). Iz rečenog proizlazi da nam je potreban način za dinamičko raspodeljivanje stanica u grupe: kada je opterećenje nisko, može se dozvoliti više konkurenata za jedan interval, a kada je visoko, broj stanica u grupi treba da se smanji - možda svaka grupa treba da ima samo jednu stanicu.

Prilagodljiv protokol prolaska kroz binarno stablo

Raspodeljivanje stanica po grupama može se obaviti na jedan izuzetno jednostavan način pomoću algoritma koji je Armija SAD primenila za testiranje vojnika na sifilis tokom Drugog svetskog rata (Dorfman, 1943). Od *N* vojnika je uziman uzorak krvi. Deo svakog uzorka kombinovan je u zbirni uzorak i takav zbirni uzorak testiran je na antitela. Kada je rezultat bio negativan, svi vojnici iz te grupe proglašavani su zdravim. Ako je rezultat bio pozitivan, pripremana su dva nova kombinovana uzorka, jedan od vojnika 1 do *N12* i drugi

od vojnika *NI2* + 1 do *N*. Postupale je rekurzivno ponavljan do pojedinačnog utvrđivanja inficiranih vojnika.

U računarskoj verziji ovog algoritma (Capetanakis, 1979), zgodno je stanice predstaviti listovima binarnog stabla, kao na slici 4-9. U prvom intervalu koji sledi iza uspešno iskorišćenog intervala 0, svim stanicama se dopušta da konkurišu. Ako neka od njih u tome uspe, odlično. Ukoliko dođe do sukobljavanja, tada tokom intervala 1 mogu da konkurišu samo stanice koje pripadaju čvoru 2. Ako neka od njih uspe da pristupi kanalu, interval koji sledi iza okvira rezervisan je za stanice koje pripadaju čvora 3. Ako, s drage strane, dve ili više stanica koje pripadaju čvoru 2 žele da šalju podatke, nastade sukob tokom intervala 1 i za interval 2 moći će da konkurišu stanice koje pripadaju čvora 4.



Slika 4-9. Stablo za osam stanica.

U suštini, ako se sukob dogodi tokom intervala 0, pretražuje se celo stablo - najpre po dubini - da bi se registrovale sve spremne stanice. Svaki jednobitni interval do- deljuje se po jednom čvoru stabla. Kada dođe do sukoba, pretražuje se rekurzivno levi i desni ogranak čvora. Ako je interval nezauzet ili samo jedna stanica emituje tokom njega, pretraživanje tog čvora može da se prekine jer su registrovane sve spremne stanice. (Da ima više od jedne, došlo bi do sukobljavanja.)

Kada je opterećenje sistema veliko, gotovo da se ne isplati dodeljivati interval 0 čvora 1, jer to ima smisla samo u vrlo neverovatnoj situaciji kada je samo jedna stanica spremna da pošalje okvir. Sličnim razmišljanjem moglo bi se zaključiti da uglavnom isto važi i za čvorove 2 i 3. To vodi opštijem pitanju: na kom nivou treba da počne pretraživanje stabla? Naravno, što je opterećenje veće, pretraživanje treba da počne na većoj dubini. Pretpostavićemo da svaka stanica, recimo, na osnovu posma- tranja prethodnog saobraćaja, može prilično dobro proceniti broj stanica q koje su u svakom trenutku spremne za slanje.

Označimo sada nivoe stabla na slici 4-9 počinjući od vrha: neka čvora 1 odgovara nivo 0, čvorovima 2 i 3 nivo 1 itd. Obratite pažnju na to da svaki čvor na nivou i ispod sebe ima udeo ukupnog broja stanica jednak 2^{-i}. Ako je q stanica spremnih da emituju ravnomerno raspodeljeno po čvorovima, njihov udeo ispod čvora na nivou i iznosi $T'q$. Intuitivno zaključujemo da pretraživanje treba da započne na nivou na kome je broj stanica koje konkurišu za isti interval jednak 1, tj. na nivou na kome je $2^{-i}q = 1$. Reša- vanjem ove jednačine dobijamo $i = \log_2 q$.

Otkrivena su brojna poboljšanja ovog osnovnog algoritma; detaljan opis naći ćete kod Bertsekasa i Gallagera (1992). Razmotrite, na primer, slučaj u kome su samo dve stanice *(G* i *H)* spremne da emituju. U čvora 1 doći će do sukobljavanja, pa će ispitivanjem čvora 2 biti utvrđeno da nije zauzet. Nema smisla ispitivati čvor 3, pošto u njemu garantovano postoji sukob (znamo da su spremne dve, odnosno više stanica ispod čvora 1 i da nijedna od njih nije ispod čvora 2, pa zato moraju biti ispod čvora 3). Ispitivanje čvora 3 može se preskočiti i odmah preći na čvor 6. Kada i to ispitivanje ostane bez rezultata, preskače se čvor 7 i odmah ispituje čvor *G.*

4.2.5 Protokol za višekorisnički pristup uz podelu talasne dužine

Kanalu se može pristupati i tako što se tehnikom FDM, tehnikom TDM ili njihovim kombinovanjem kanal podeli u potkanale koji se po potrebi dinamički dodeljuju. Slične tehnike se često koriste u lokalnim optičkim mrežama da bi se omogućilo istovremeno komuniciranje više korisnika na različitim talasnim dužinama (tj. frekvencijama). U ovom odeljku ćemo razmotriti jedan takav protokol (Humblet i sar., 1992).

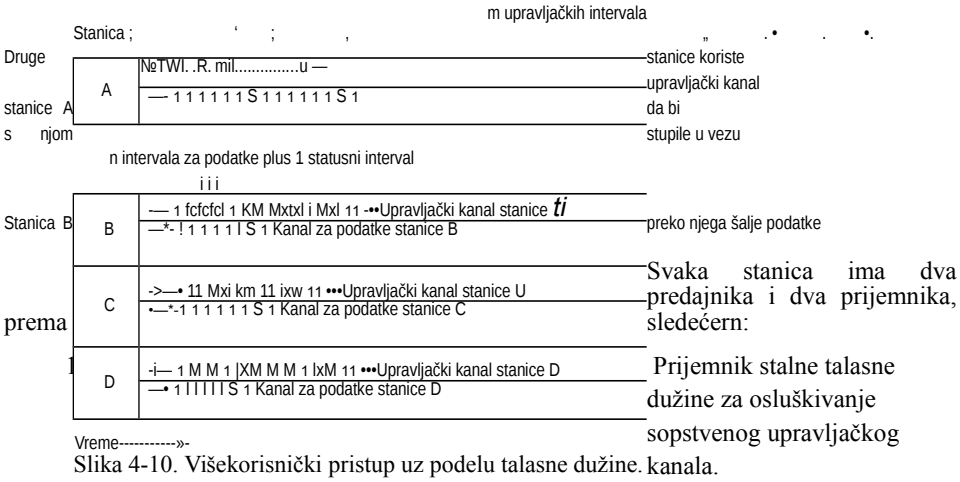
Potpuno optičku lokalnu mrežu lako je napraviti u obliku pasivne zvezde (slika 2- 10). Po dva vlakna od svake stanice zatopljena su za stakleni valjak. Jedno vlakno vodi signale ka valjku, a drago od njega ka stanici. Svetlosni signal koji emituje bilo koja stanica osvetljava ceo valjak tako ga mogu primiti sve stanice. Pasivna zvezda može da radi s više stotina stanica.

Da bi se omogućilo više istovremenih prenosa, spektar se deli na kanale (područja talasne dužine), kao što je prikazano na slici 2-31. Po protokolu za višekorisnički pristup uz podelu talasne dužine (engl. *Wavelength Division Multiple Access, WDMA),* svakoj stanici se dodeljuju dva kanala. Uskim kanalom ona prima upravljačke signale, a širokim šalje okvire s podacima.

Svaki kanal je podeljen u grupe vremenskih intervala, kao na slici 4-10. Označimo sa *m* broj intervala u upravljačkom kanalu, a sa *n* + 1 broj intervala u kanalu za podatke, pri čemu *n* intervala služe za korisničke podatke, a dodatni interval nosi podatke o statusu stanice (uglavnom o tome koji su intervali na oba kanala slobodni). Sekven- ca intervala se neprekidno ponavlja na oba kanala, pri čemu se interval 0 specijalno obeležava kako bi oni koji su se kasnije uključili mogli lako da ga prepoznaju. Svi kanali su sinhronizovani pomoću jedinstvenog globalnog sata.

Protokol podržava tri klase saobraćaja: (1) konstantne brzine prenosa sa uspostavljanjem direktne veze, kao što je nekomprimovani video saobraćaj, (2) promenljive brzine prenosa sa uspostavljanjem direktne veze, kao što je prenos datoteka i (3)

saobraćaj datagrama, kao što su UDP paketi. Kada stanica A želi da komunicira sa stanicom *B* pomoću jednog od dva protokola sa uspostavljanjem direktne veze, ona prvo mora da okvir CONNECTION REQUEST (zahtev za povezivanje) postavi u slobodan okvir na upravljačkom kanalu stanice *B.* Ako stanica *B* prihvati zahtev, njihovo komuniciranje se obavlja na kanalu za podatke stanice A.



Slika 4-10. Višekorisnički pristup uz podelu talasne dužine.

prema sledećern:

1. Prijemnik stalne talasne dužine za osluškivanje sopstvenog upravljačkog kanala.
2. Predajnik s podesivom talasnom dužinom za slanje poruka upravljačkim kanalima drugih stanica.
3. Predajnik stalne talasne dužine za slanje okvira s podacima.
4. Prijemnik s podesivom talasnom dužinom za biranje predajnika podataka (drugih stanica).

Drugim recima, svaka stanica na sopstvenorn upravljačkom kanalu osluškuje dolazne zahteve drugih stanica, ali se mora usaglasiti s talasnom dužinom njihovih predajnika da bi preuzela podatke. Talasna dužina se bira interferometrom (Fabri-Peroovim ili Mah-Zenderovim) koji prigušuje sve talasne dužine osim onih iz izabranog područja.

Pogledajmo sada kako stanica A uspostavlja komunikacioni kanal klase 2 sa stanicom *B,* da bi s njom razmenila datoteke. Najpre, stanica A podešava svoj prijemnik podataka na talasnu dužinu kanala za podatke stanice *B* i čeka statusni interval. Iz njega će saznati koji su intervali zauzeti, a koji slobodni. Na primer, na slici 4-10 vidimo da su od osam upravljačkih intervala stanice *B* slobodni intervali 0, 4 i 5. Ostali su zauzeti, što je na slici označeno krstićima.

Stanica A bira jedan od slobodnih upravljačkih intervala, npr. interval 4, i u njega umeče svoju poruku CONNECTION REQUEST. Pošto stanica *B* stalno proverava svoj upravljački kanal, ugledaće zahtev i prihvatiti ga tako što će interval 4 dodeliti stanici A. Ta dodela postaje vidljiva u statusnom intervalu kanala za podatke stanice *B,* Čim to primeti stanica A, ona zna da je uspostavila jednosmernu vezu. Da je stanica A zahtevala dvosmernu vezu, stanica *B* bi morala da sa stanicom A ponovi isti postupak.

Moguće je da u trenutku kada stanica A pokušava da prigrabi upravljački interval 4 stanice *B,* to isto želi i stanica C. Nijednoj to neće uspeti, a to će saznati posmatrajuci statusni interval upravljačkog kanala stanice *B*. Posle toga, svaki od dva konkurenta posle proizvoljnog vremenskog intervala ponovo pokušava da pristupi stanici *B*.

U ovoj fazi, svaka strana može drugoj da bez sukoba šalje kratke upravljačke poruke. Da bi poslala datoteku, stanica A šalje stanici *B* upravljačku poruku tipa „Obrati pažnju na interval 3 u mom kanalu za slanje podataka. Tamo je jedan okvir za tebe“. Kada stanica *B* dobije takvu poruku, ona podešava svoj prijemnik na talasnu dužinu kanala za slanje podataka stanice A da bi učitala pomenuti okvir. U zavisnosti od organizacije protokola u višim slojevima, stanica B, ako to želi, može da istim mehanizmom pošalje potvrdu o primljenom okviru.

Skrećemo pažnju da će nastati problem ako su stanice A i C istovremeno povezane sa stanicom *B* i obe joj odjednom nalože da motri interval 3. Stanica *B* će nasumično odabrati jedan od dva zahteva, dok će onaj drugi osujetiti.

Kada se radi uz stalnu brzinu prenosa, koristi se varijanta ovog protokola. Kada stanica A zatraži vezu, ona istovremeno šalje i pitanje: Mogu li da ti šaljem okvir svaki put kad se pojavi interval 3? Ako stanica *B* može da pristane na to (dakle, interval 3 nije prethodno rezervisala za nekog drugog), uspostavlja se veza s garantovanim propusnim opsegom. Ukoliko stanica *B* to ne može, stanica A može da podnese nov predlog, u zavisnosti od toga koji su joj intervali za slanje podataka slobodni.

Za saobraćaj klase 3 (datagramski) koristi se opet drugačija varijanta protokola. Umesto da u slobodni upravljački interval (4) unese CONNECTION REQUEST, upisaće poruku DATA FOR YOU IN SLOT 3 (podaci za tebe u intervalu 3). Ako stanica *B* nije zauzeta tokom sledećeg intervala 3 na kanalu za podatke, prenos će uspeti. U suprotnom, okvir s podacima se gubi. Kada se radi na ovaj način, nema potrebe za uspostavljanjem ikakve veze.

Moguće je zamisliti više varijanti osnovnog protokola. Na primer, umesto da svaka stanica ima sopstveni upravljački kanal, mogu sve deliti jedinstven, zajednički upravljački kanal. Svakoj stanici se u svakoj grupi dodeljuje određeni blok intervala, čime se na jednom fizičkom kanalu multipleksira više virtuelnih kanala.

Moguće je proći i samo s po jednim podesivim predajnikom, odnosno prijemnikom na svakoj stanici tako što će kanal svake starice biti izdeljen na m upravljačkih intervala i $n + 1$ intervala za podatke. Kod ove šeme je nepovoljno to što pošiljalac mora duže da čeka da bi „ulovio“ slobodan upravljački okvir i što su naknadni okviri s podacima proređeni jer se između njih umeću upravljački podaci.

Predložene su i u praksi realizovane i brojne druge varijante WDMA protokola koje se međusobno razlikuju u detaljima. U nekima postoji samo jedan upravljački kanal, dok ih u drugima ima više. U nekima se uzima u obzir kašnjenje signala, u nekima ne.
Neke eksplicitno uračunavaju vreme usaglašavanja talasne dužine, dok ga druge potpuno

zanemaruju. Varijante se razlikuju i po složenosti obrade, protoku podataka i mogućnostima primene na sisteme različite veličine. Kada se koristi veliki broj frekvencija, sistem se ponekad zove **multipleksiranje sa čestom podelom talasnih dužina** (engl. *Dense Wavelength Division Multiplexing, DWDM).* Više detalja o ovome potražite kod Boginenija i saradnilca (1993), Chena (1994), Goralskog (2001), Karta- lopoulosa (1999) i Levina i Akyildiza (1995).

4.2.6 Protokoli za bežične lokalne mreže

S povećanjem broja pokretnih računarskih i komunikacionih uređaja raste i potreba njihovog povezivanja sa spoljnim svetom. Čak su i prvi mobilni telefoni mogli da se povežu sa aparatima na fiksnoj telefonskoj mreži. Prvi prenosivi računari nisu imali tu mogućnost, ali je ubrzo modem postao njihova uobičajena komponenta. Da bi mogli da komuniciraju s drugim računalima, trebalo ih je priključiti na telefonsku utičnicu. S obzirom na potrebu za fizičkim povezivanjem s fiksnom mrežom, ti računari su možda bili prenosivi, ali ne u pravom smislu reči „pokretni".

Da bi zaista mogli da komuniciraju „u hodu", prenosivi računari su morali preći na bežičnu vezu (putem radio ili infracrvenih talasa). Na taj način, računarski zavisnici su mogli da čitaju i šalju elektronske poruke dok voze bicikl ili jedre. Sistem prenosivih računara koji međusobno komuniciraju radio putem može se shvatiti kao bežična lokalna mreža, onako kako je opisano u odeljku 1.5.4. Takve lokalne mreže se po svojstvima pomalo razlikuju od klasičnih lokalnih mreža, pa su za njih potrebni specijalni protokoli MAC podsloja. Neke od njih ćemo ispitati u ovom odeljku. Više informacija o bežičnim lokalnim mrežama potražite kod Geiera (2002) i O'Hare i Petricka (1999).

Bežična lokalna mreža se obično lconfiguriše unutar poslovne zgrade, pri čemu se bazne stanice (pristupne tačke) razrneštaju na pogodna mesta. Sve bazne stanice međusobno se povezuju bakarnim ili optičkim kablom. Ako se bazne stanice i prenosivi računari tako podese da im emisioni domet bude 3 do 4 metra, tada svaka prostorija u zgradi postaje zasebna ćelija, a sama zgrada - veliki višećelijski sistem, kao kod klasičnog sistema mobilne telefonije, o kome smo govorili u 2. poglavlju. Za razliku od sistema mobilne telefonije, ovde svaka ćelija ima samo jedan kanal koji pokriva čitav propusni opseg i sve stanice koje se u njoj nalaze. Propusni opseg se najčešće kreće u granicama između 11 i 54 Mb/s.

U opisu koji sledi, pretpostavićemo, jednostavnosti radi, da svi radio-predajnici imaju ograničen, fiksni domet. Kada se prijemnik nađe u dometu dva aktivna predajnika, signal koji primi biće u opštem slučaju izobličen i neupotrebljiv, tako da u našem opisu nećemo razmatrati sisteme tipa CDMA. Treba imati na umu da u nekim bežičnim lokalnim mrežama nisu sve stanice jedna dragoj u dometu, što donosi niz komplikacija. Osim toga, u bežičnim lokalnim mr ežama koje se nalaze u zgradama na domet pojedinih stanica veoma utiču zidovi koji ih razdvajaju.

Prvo što nam pada na um (i ne baš originalno), to je da upotrebimo protokol CSMA: osluškivaćemo emisije dragih stanica i emitovati tek onda kad se sve utišaju. Taj protokol ipak neće raditi dobro jer su problem smetnje kod prijemnika, a ne kod predajnika. To cerno bolje razumeti ako pogledamo sliku 4-11, koja prikazuje četiri bežične stanice.

U ovom trenutku nam nije važno koje od njih su bazne stanice, a koje prenosivi računali. Jačina emitovanja podešena je tako da se stanice A i /i jedna drugoj nalaze u dometu i mogu se međusobno ometati. Stanica *C* može da ometa stanice *B* i *D,* ali ne i stanicu A.

Slika 4-11. Bežična lokalna mreža, (a) Emituje stanica A. (b) Emituje stanica B,

Razmotrimo najpre šta se događa kada stanica A šalje podatke stanici *B,* kao što je prikazano na slici 4-11(a). Ako stanica C osluškuje medijum, ona neće čuti stanicu A jer se nalazi izvan njenog dometa i zato će pogrešno zaključiti da može da pošalje podatke stanici *B.* Ako stanica C počne da emituje, ometaće stanicu *B* i upropastiti okvir koji joj šalje stanica A. Kada stanica ne može da utvrdi postojanje potencijalnog konkurenta zato što je on previše udaljen, nastaje **problem skrivene stanice** (engl. *hidden station problem).*

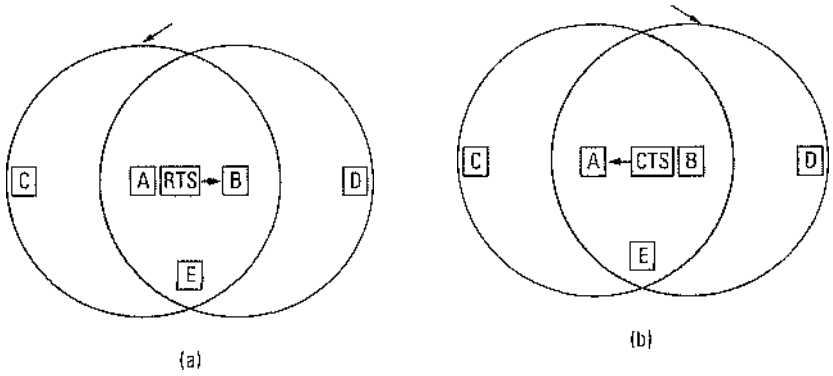
Razmotrimo sada obrnutu situaciju, kada stanica *B* šalje podatke stanici A, kao na slici 4-11(b). Ako stanica *C* osluškuje medijum, ona će čuti tu emisiju i pogrešno zaključiti da ne sme da šalje podatke stanici *D,* iako bi takva emisija ometala samo zonu između stanica *B* i C, u kojoj nema nijednog prijemnika. To je **problem izložene stanice** (engl. *exposed station problem).*

Stanica, pre nego što počne da emituje, u stvari želi tačno da zna kakva aktivnost postoji oko pretpostavljenog prijemnika, dok pomoću protokola CSMA, osluškujući medijum, može samo da utvrdi da oko nje postoji ili ne postoji aktivnost. Kada su stanice povezane žicom, signali stižu do svake od njih, tako da u čitavom sistemu u jednom trenutku može postojati samo jedno emitovanje. U sistemima zasnovanim na radio-talasima kratkog dometa, istovremeno može postojati više emisija ukoliko svaka ima drugačije odredište, a odredišta su jedno drugom izvan dometa.

Pomenuti problem se može ilustrovati i tako što ćemo zamisliti poslovnu zgradu u kojoj svaki zaposleni ima bežični prenosivi računar. Pretpostavimo da Ana želi da pošalje poruku Ivanu. Anin računar će oslušnuti lokalnu okolinu i ako ne utvrdi nikakvu aktivnost, počeće da šalje poruku. Međutim, do sukoba može da dođe u Ivanovoj kancelariji jer mu možda u istom tenutku poruku šalje treća osoba, čiji računar Ana ne može da otkrije jer je previše udaljen od nje.

Protokoli MACA i MACAW

Od protokola namenjenih bežičnim lokalnim mrežama jedan od prvih je protokol za **višekorisnički pristup uz izbegavanje sukoba** (engl. *Miltiple Access with Collision Avoidance, MACA)* (Karn, 1990). Tu prvi potez povlači pošiljalac podstičući primaoca na emitovanje kratkog okvira koji će ućutkati obližnje stanice tokom slanja narednog (velikog) okvira s podacima. Protokol MACA je prikazan na slici 4-12.



Slika 4-12. Protokol MACA. (a) Stanica A šalje okvir RTS stanici *B*.
(b) Stanica *B* odgovara stanici *A* okvirom CTS.

Razmotrimo sada kako stanica A šalje okvir stanici *B.* Stanica A počinje tako što stanici *B* šalje **zahtev za slanje** (engl. *Request to Send, RTS),* kao što je prikazano na slici 4-12(a). Taj kratak okvir (.30 bajtova) sadrži dužinu okvira s podacima koji će eventualno biti naknadno poslat. Stanica *B* odgovara **dozvolom za slanje** (engl. *Clear to Send, CTS),* kao što je prikazano na slici 4-12(b). Okvir CTS sadrži dužinu okvira s podacima (kopkanu iz okvira RTS). Pošto primi okvir CTS, stanica A počinje da šalje podatke.

Pogledajmo sada kako reaguju okolne stanice koje slušaju ovaj razgovor. Stanica koja uhvati RTS okvir izvesno je blizu stanice A i zato mora da ćuti dovoljno dugo da CTS okvir stigne do stanice A bez sukobljavanja. Stanica koja uhvati CTS okvir nalazi se blizu stanice B i mora da ćuti tokom narednog prenosa podataka, čije trajanje (dužinu) saznaje ispitujući CTS okvir.

Na slici 4-12, stanica *C* je u dometu stanice A, ali ne i u dometu stanice *B.* Prema tome, ona čuje signal RTS od stanice A, ali ne i signal CTS od stanice *B.* Sve dok se ne sukobljava sa CTS signalom, ona može da emituje tokom slanja okvira s podacima. Za razliku od nje, stanica *D* se nalazi u dometu stanice *B,* ali ne i u dometu stanice A. Ona ne čuje signal RTS, ali čuje signal CTS. Tako zna da se nalazi blizu stanice koja upravo treba da primi okvir i zato prestaje da emituje sve dok se ne završi očekivani prenos podataka. Stanica *E* čuje obe upravljačke poruke i, slično stanici *D,* mora da se primiri dok se prenos podataka ne završi.

Uprkos opisanim predostrožnostima, još uvek može da dođe do sukobljavanja. Na primer, obe stanice *(B* i *C)* mogu da stanici A istovremeno pošalju RTS okvire koji će se sukobiti i propasti. U takvoj situaciji, svaki neuspešan predajnik (onaj koji nije registrovao CTS okvir tokom očekivanog vremenskog intervala), ponovo će poslati RTS okvir posle proizvoljno izabranog intervala mirovanja. Ovde se koristi algoritam binarnog eksponencijalnog odustajanja koji ćemo obraditi kada budemo govorili o Ethernetu.

Na osnovu proučavanja u simuliranim uslovima (Bharghavan i sar., 1994), protokolu MACA su poboljšane performanse i on je dobio novo ime: MACA za bežične mreže (engl.

MACA for Wireless, MACAW). Autori su najpre zapazili da se izgubljeni

okviri ponovo šalju tek kada se njihov nedostatak primeti u transportnom sloju ukoliko se u sloju veze ne obezbedi povratno slanje potvrda. To su resili uvodeći okvir ACK posle svakog uspešno primljenog okvira s podacima. Takođe su utvrdili da je protokol CSMA ipak koristan jer može da spreči stanicu da emituje RTS okvir ukoliko neka obližnja stanica već šalje takav okvir na isto odredište, tako da su protokolu MACA omogućili i da osluškuje nosilac podataka. Istovremeno su odlučili da algoritam odustajanja, umesto za svaku stanicu, izvršavaju nezavisno za svaki tok podataka (za svaki par izvorište-odredište). Na kraju su dodali mehanizam koji stanicama omogućava da razmenjuju informacije o zagušenju i na odgovarajući način obuzdali reagovanje algoritma odustajanja na privremene probleme, te tako poboljšali performanse sistema.

43 ETHERNET

Stigli smo do samog kraja opšteg, teorijskog prikaza protokola za dodeljivanje kanala i vreme je da razmotrimo kako se ti principi primenjuju u stvarnim sistemima, naročito u lokalnim mrežama. Kao što smo naveli u odeljku 1.5.3, IEEE je standardizovao više tipova lokalnih i gradskih mreža pod oznakom IEEE 802. Kao što se vidi na slici 1-38, do danas je preživeo mali broj tih mreža. Oni koji veruju u reinkarnaciju misle da se Čarls Darvin ponovo pojavio u liku nekog člana Udruženja za standardizaciju Instituta IEEE da bi istrebio „manje sposobne“ jedinke. Od preživelih tipova mreža, najvažnije su 802.3 (Ethernet) i 802.11 (bežična lokalna mreža). Za mreže 802.15 (Bluetooth) i 802.16 (bežična gradska mreža) još se ne može doneti konačan sud, pa ga zato potražite u 5. izdanju ove knjige. Mreže 802.3 i 802.11 razlikuju se i u fizičkom sloju i u MAC podsloju, ali se susreću na istom podsloju upravljanja logičkom vezom (definisanim standardom 802.2), tako da imaju isti interfejs ka mrežnom sloju.

Ethernet smo predstavili u odeljku 1.5.3, tako da ćemo se odmah baciti na njegove tehničke detalje, protokole i skorašnji razvoj visokobrzinskog (gigabitnog) Etherneta. Pošto su Ethernet i standard IEEE 802.3 identični, osim dve male razlike koje ćemo ubrzo objasniti, mnogi ta dva izraza koriste kao sinonime, pa ćemo im se i mi priključiti. Više obaveštenja o Ethernetu možete naći kod Breyera i Rileyja (1999), Seiferta (1998) i Spurgeona (2000).

4.3.1 Kabliranje Etherneta

Pošto se ime „Ethernet“ odnosi na kabl (etar), počnimo naše razmatranje od njega. Za Ethernet se koriste četiri vrste kablova (slika 4-13).

Hronološki se prvo pojavio sistem kabliranja 10Base5, popularni debeli Ethernet (engl. *thick Ethernet).* Kabl je podsećao na žuto baštensko crevo, sa oznakama na svakih 2,5 metra na mestima gde treba ubosti račve. (U standardu 802.3 *ne insistira* se na tome da kabl bude žut, ali se to *preporučuje.)* Veze s kablom najčešće se ostvaruju pomoću ubodnih račvi (engl. *vampire taps),* čija se igla - oprezno do polovine - utisne u jezgro koaksijalnog kabla. Oznaka 10Base5 znači da sistem radi brzinom 10 Mb/s, da koristi signaliziranje u osnovnom opsegu i da može da podrži segmente dužine do 500 m. Prvi broj je brzina u Mb/s. Zatim dolazi reč „Base“ (ili, ponekad, „BASE“) koja označava da se prenos vrši u osnovnom opsegu. Postojala je i širokopojasna varijanta, 10Broad36, ali nikada nije uspela da zainteresuje

tržište, pa je napuštena. Ako je me- dijum koaksijalni kabl, na kraju dolazi njegova dužina u 100-metarskim jedinicama.

Ime	Kabl	Maksimalna dužina segmenta	Čvorova po segmentu	Prednosti
10Base5	Debeo koaksijalni	500 m	100	Prvobitni kabl; sada zastareo
10Base2	Tanak koaksijalni	185 m	30	Nisu potrebni razvodnici
10Base-T	Upredena parica	100 m	1024	Najjeftiniji sistem
10Base-F	Optički	2000 m	1024	Najbolji za međusobno povezivanje zgrada

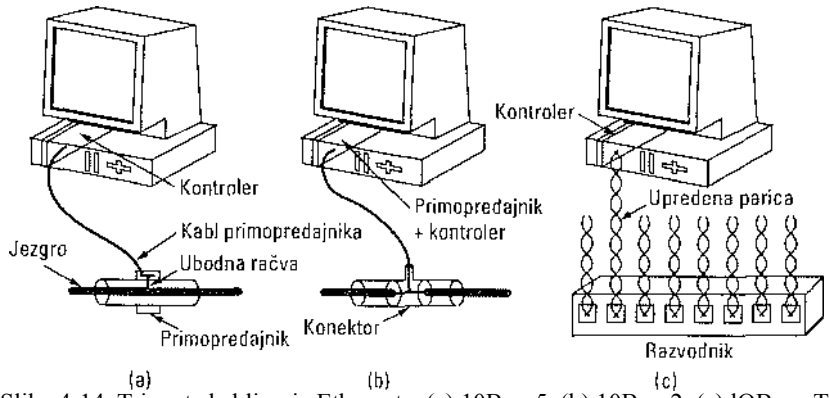
Slika 4-13. Najčešće vrste kablova za Ethernet.

Sledeći sistem kabliranja bio je **10Base2 - tanki Ethernet** (engl. *thin Ethernet*), s kablom koji se za razliku od „baštenskog creva“ lako mogao saviti. Stanice se na njega, umesto ubodnim račvama, povezuju pomoc'u standardizovanih BNC konelctora koji formiraju T-spojeve. BNC konektori se lakše koriste i pouzdaniji su. Tanki Ethernet je mnogo jeftiniji i lakše se instalira, ali segmenti kabla mogu da budu dugački najviše 185 metara i svaki može da podrži samo 30 računara.

Nalaženje prekida, „viška“ kabla, nesolidno ubodenih račvi i labavih konektora može da bude veliki problem kod oba medijuma. Zbog toga su razvijene tehnike za otkrivanje takvih grešaka. One u principu rade tako što se u kabl pusti impuls poznatog oblika i osluškuje odjek koji se javlja kada on naiđe na prepreku ili dostigne kraj kabla. Tačnim merenjem vremena između puštanja impulsa i stizanja njegovog odjeka može se lokalizovati uzrok nastanka odjeka. Tehnika se zove **reflektometrija vremenskog domena** (engl. *time domain reflectometry).*

Teškoće u vezi s nalaženjem mesta prekida u kablu uticale su na to da se za sisteme pronađe drugačija vrsta ožičenja, gde kablovi iz svih stanica vode u centralni **razvodnile** (engl. *hub)* u kome se svi međusobno električno povezuju (kao da su zalemljeni). Ti „kablovi“ su obično telefonske upredene parice, pošto je većina poslovnih zgrada njima opremljena i obično ima dosta rezervnih parica. Takav sistem se zove **10Base-T.** Razvodnici ne baferuju dolazni saobraćaj, ali ćemo u nastavku poglavlja opisati njihovu poboljšanu verziju (skretnice, engl. *switches*), koje to čine.

Tri opisane šeme ožičenja prikazane su na slici 4-14. U sistemu 10Base5, **primo-predajnik** (engl. *transceiver)* pričvršćuje se oko kabla tako da njegova račva ostvaruje pouzdan kontakt s jezgrom kabla. Primopredajnik sadrži elektronske komponente za osluškivanje nosioca i otkrivanje sukobljavanja. Kada otkrije sukob, primopredajnik emituje u kabl specijalan neregularan signal kako bi svi ostali primopredajnici nedvosmisleno razumeli daje došlo do sukobljavanja.



Slika 4-14. Tri vrste kabliranja Etherneta. (a) 10Base5. (b) 10Base2. (c) lOBase-T.

U sistemu lOBase.5, **kabl primopredajnika** (engl. *transceiver cable)* ili **spojni kabl** (engl. *drop cable)* povezuje primopredajnik sa interfejsom računara. Spojni kabl može da bude dugačak do 50 m i sadrži pet zasebno oklopljenih upredenih parica. Dve parice su za prijem i slanje podataka, dve sledece su za prijem i slanje signala, a peta, koja se ne koristi uvek, omogućava računara da napaja elektroniku primopredajnika. Neki primopredajnici dozvoljavaju da se za njih veže do osam obližnjih računara smanjujući tako broj potrebnih primopredajnika.

Kabl primopredajnika završava na kartici interfejsa u unutrašnjosti računara. Ta kartica sadrži kontroler koji razmenjuje okvire s primopredajnikom. Kontroler slaže podatke u odgovarajući format okvira, izračunava kontrolni zbir za okvire koje šalje i proverava ga u okvirima koje prima. Neki kontroleri talcođe upravljaju skupom bafe- ra za dolazne okvire, redom čekanja bafera sa okvirima za slanje, direktnom razme- nom memorije s računarima, kao i drugim poslovima u radu na mreži.

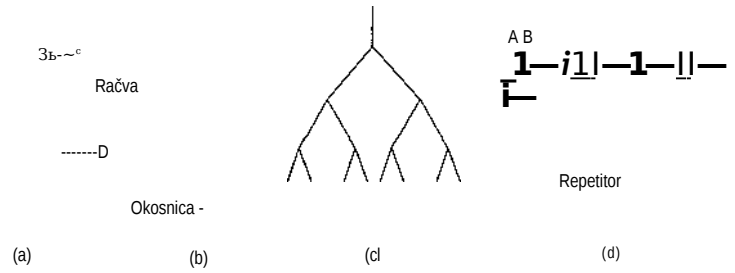
U sistemu 10Base2, spajanje s kablom izvodi se pomoću pasivnog T-spoja koji obrazuju BNC konektori. Elektronske komponente primopredajnika nalaze se na kartici kontrolera, a svaka stanica ima sopstveni primopredajnik.

U sistemu lOBase-T uopšte nema zajedničkog kabla, već samo razvodnile (kutija prepuna elektronskih komponenata) za koji su stanice vezane namenskim (dakle, ne zajedničkim) kablom. U ovoj konfiguraciji stanice se lakše dodaju i uklanjaju, a prekid kabla se lako otkriva. Nažalost, u sistemu lOBase-T, najveća dužina kabla koji vodi od razvodnika može da bude 100 m, a najviše 200 m ukoliko se upotrebe visokokvalitetne parice kategorije 5. Pa ipak, sistem lOBase-T brzo je preuzeo vodeću ulogu zahvaljujući korišćenju postojećeg ožičenja i lakoći održavanja. O bržoj verziji ovog sistema (lOOBase-T) govoriđemo u nastavku poglavlja.

Četvrti sistem kabliranja Etherneta, **lOBase-F,** zasnovan je na optičkom kablu. Ova varijanta je skupa zbog visoke cene konektora i završnih elemenata, ali je izuzetno imuna na smetnje i preporučljiva za povezivanje zgrada ili međusobno vrlo udaljenih razvodnika. Segment kabla može se pružati i čitav kilometar. Sistem je u informacio- norn smislu i bezbedniji, jer je mnogo teže prisluškivati optički kabl nego bakarni.

Na slici 4-15 prikazani su razni načini ožičenja zgrade. Na slici 4-15(a) jedan jedini kabl se provlači iz jedne prostorije u dragu, a stanice mu se priključuju u najbližim tačkama. Na slici 4-15(b), od krova do suterena zgrade prolazi vertikalni vod, od koga se preko specijalnih pojačivača (repetitora) granaju horizontalni kablovi. U nekim zgradama, horizontalni kablovi su tanki, dok je vertikalni vod (okosnica) debeo. Najčešća topologija je stablo, kao na slici 4-15(c), jer se u mrežama s dve putanje između pojedinih parova stanica javlja interferencija dva signala.

U svakoj verziji Etherneta segment kabla ima ograničenu dužinu. Da bi se ostvarile veće mreže, segmenti kablova mogu se nadovezivati pomoću repetitora (engl. *repeaters),* kao na slici 4-15(d). Repetitor je uređaj fizičkog sloja. On prima, pojačava (regeneriše) i ponovo šalje signal u oba smera. S gledišta softvera, niz segmenata povezanih repetitorima ne razlikuje se od jedinstvenog kabla (osim povećanog kašnjenja koje izazivaju repetitorij. Sistem može da sadrži više kablovskih segmenata i više repetitora, ali međusobna udaljenost primopredajnika ne sme biti veća od 2,5 km, a između dva primopredajnika može biti najviše četiri repetitora.



Slika 4-15. Topologije kabla, (a) Linearna, (b) Sa okosnicom, (c) Stabla, (d) Segmentirana.

4.3.2 Mančester kodiranje

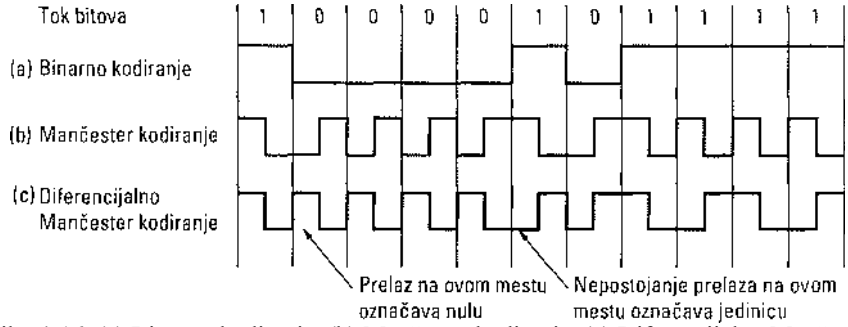
Ni u jednoj verziji Etherneta ne koristi se direktno binarno kodiranje (0 V za bit 0 i 5 V za bit 1), jer to dovodi do zabune. Ako jedna stanica pošalje niz bitova 0001000, druga to može protumačiti kao niz 10000000 ili kao niz 01000000, jer ne može da razlikuje neaktivnost pošiljaoca (0 V) i bit 0 (takođe 0 V). Problem se može rešiti ako se bit 1 označi sa +1 V, a bit 0 sa -1 V, ali će i dalje biti teškoća ako primalac uzorkuje signal nešto drugačijom frekvencijom od one kojom gaje pošiljalac generisao. Različit radni takt može da naruši sinhronizaciju u blizini granice bita, naročito posle dugačkog niza samih nula ili jedinica.

Očigledno je neophodan mehanizam pomoću koga će primaoci pouzdano moći da utvrde početak, kraj ili sredinu svakog bita bez spoljnog sata. Dva takva pristupa su Mančester kodiranje (engl. *Manchester encoding)* i diferencijalno Mančester kodiranje (engl.

differential Manchester encoding). Mančester kodiranjem, period svakog bita deli se na dva jednaka intervala. Bit 1 se šalje tako što se tokom prvog intervala napon drži na visokom nivou, a tokom drugog na niskom. Bit 0 se šalje upravo obrnuto: prvo nizale napon, zatim visok. Uz takav postupale, u sredini perioda svakog bita nastaje

prelaz koji primaocu omogućava da se lako sinhronizuje s pošiljaocem. Za Mančester kodiranje, međutim, potreban je dvostruko veći propusni opseg (dvostruko viša frekvencija), nego za direktno kodiranje, jer je impuls dvaput kraći. Na primer, da bi se podaci slali brzinom 10 Mb/s, signal treba da se menja 20 miliona puta u sekundi. Mančester kodiranje je prikazano na slici 4-16(b).

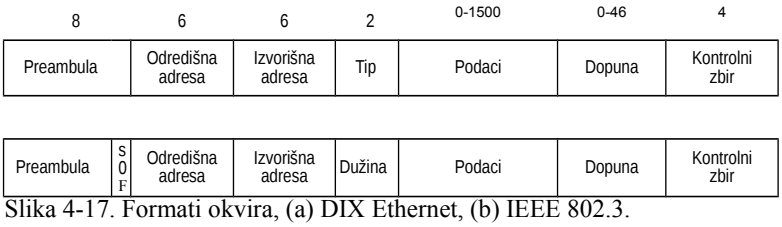
Diferencijalno Mančester kodiranje, prikazano na slici 4-16(c), varijanta je osnovnog Mančester kodiranja. U njemu, bit 1 je naznačen nepostojanjem prelaza na početku intervala. Nasuprot tome, bit 0 je naznačen prelazom na početku intervala. U oba slučaja postoji i prelaz u sredini. Za diferencijalno kodiranje potrebna je složenija oprema, ali je ono i manje podložno smetnjama. Zbog svoje jednostavnosti, Mančester kodiranje se koristi u svim Ethernet mrežama. Viši napon signala iznosi +0,85 V, a niži -0,8.5 V (u odsustvu signala napon je tačno 0 V). Diferencijalno Mančester kodiranje ne koristi se na Ethernetu, ali se koristi u drugim lokalnim mrežama (npr. u mreži 802.5 tipa token ring).



Slika 4-16. (a) Binarno kodiranje, (b) Mančester kodiranje, (c) Diferencijalno Mančester kodiranje.

4.3.3 Protokol MAC podsloja za Ethernet

Na slici 4-17(a) prikazana je originalna struktura DIX okvira (DIX = DEC, Intel, Xerox). Svaki okvir počinje *Preambulom* dužine 8 bajtova, koja je uvek predstavljena nizom bitova 10101010. Kada se taj niz kodira tehnikom Mančester, dobija se tokom 6,4 jxs pravougaoni talas frekvencije 10 MHz, koji primaocu omogućava da se sinhronizuje s pošiljaocem. Pošiljalac i primalac moraju ostati sinhronizovani tokom preno- sa ostatka okvira, pri čemu se granice bitova označavaju Mančester kodiranjem.



Slika 4-17. Formati okvira, (a) DIX Ethernet, (b) IEEE 802.3.

Okvir sadrži adrese izvorišta i odredišta. Standardom su dozvoljene 2-bajtne i 6-bajtne

adrese, ali se u parametrima koji definišu standard osnovnog opsega od 10 MHz koriste samo 6-bajtne. Najznačajniji bit odredišne adrese je 0 za obične adrese, a 1 za grupne adrese. Grupna adresa omogućava da više stanica prima poruke prelco jedne adrese. Kada se okvir pošalje na grupnu adresu, sve stanice je primaju. Slanje poruka grupi stanica naziva se **višesmerno emitovanje** (engl. *multicast).* Adresa koja se sastoji od samih jedinica rezervisana je za **neusmereno (difuzno) emitovanje** (engl. *broadcast).* Okvir koji u odredišnoj adresi ima same jedinice primaju sve stanice na mreži. Razlika između višesmernog i neusmerenog emitovanja dovoljno je važna daje ponovo istaknemo. Višesmerno emitovan okvir stiže odabranoj grupi stanica na Ethernetu; neusmereno poslat okvir stiže svim stanicama na Ethemetu. Višesmerno emitovanje je selektivnije, ali zahteva rad s grupama. Neusmereno emitovanje je grublje, ali ne zahteva rad s grupama.

Zanimljivo je i to što se sledeci manje značajan bit (46.) koristi za razlikovanje lokalnih i globalnih adresa. Lokalne adrese dodeljuje administrator i one nemaju značaja izvan lokalne mreže. Globalne adrese, međutim, dodeljuje samo IEEE da bi svaka stanica na svetu imala svoju jedinstvenu adresu. Na raspolaganju je 48 - 2 = 46 bitova, što omogućava oko 7×10^{13} različitih globalnih adresa. Namera je bila da se svakoj stanici omogući da pozove drugu stanicu navodeći samo njen jedinstven 48-bitni broj. Posle toga, mrežni sloj mora nekako da se snađe da locira odredište.

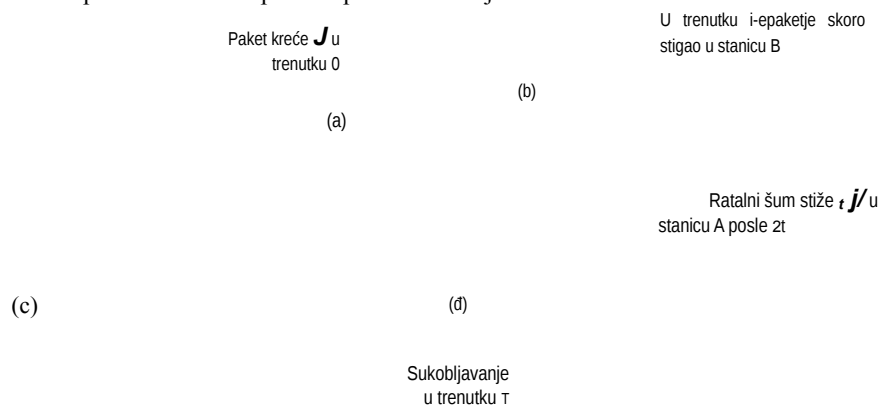
Sledeće polje je *Tip* koje primaocu saopštava šta da radi sa okvirom. Na istom računaru može istovremeno da se izvršava više protokola mrežnog sloja, pa jezgro operativnog sistema mora da zna kome od njih da prosledi okvir koji je stigao preko Etherneta. Poljem *Tip* specificira se proces kome treba predati okvir.

Zatim dolaze *Podaci,* polje dužine do 1500 bajtova. Ta, pomalo proizvoljna granica, izabrana je zbog toga što primopredajnik mora imati dovoljno radne memorije da prihvati čitav okvir, a u trenutku uspostavljanja DIX standarda (1978) RAM memorija je bila prilično skupa. Veće polje s podacima zahtevalo bi i više RAM-a, znači skuplji primopredajnik.

Osim što postoji maksimalna dužina okvira, postoji i njegova minimalna dužina. Iako je polje s podacima dužine 0 bita ponekada korisno, ono izaziva probleme. Kada primopredajnik otkrije sukob, on naglo prekida slanje aktuelnog okvira, što znači da kablom stalno putuju zalutali bitovi i delovi okvira. Da bi se ispravan okvir lakše razlikovao od smeća, Ethernet zahteva da ispravan okvir od odredišne adrese do kontrolnog zbira (uključujući obe stavke) bude dugačak najmanje 64 bita. Ako je polje s podacima kraće od 46 bitova, okvir se u polju *Dopuna* (engl. *pad)* dovodi do minimalne dužine.

Drugi, važniji razlog za propisivanje minimalne dužine okvira jeste sprečavanje stanice da završi slanje kratkog okvira pre nego što njegov prvi bit stigne na drugi kraj kabla, gde se može sukobiti s nekim drugim okvirom. Opisani problem je prikazan na slici 4-18. U trenutku 0, stanica A koja se nalazi na jednom kraju mreže, počinje da šalje okvir. Neka je vreme potrebno da okvir stigne na drugi kraj mreže jednako *%.* Upravo pre nego što okvir dostigne drugi kraj mreže (tj. u trenutku *%* - e), najudaljenija stanica *B* počinje da emituje. Kada stanica *B* utvrdi da prima više snage nego što

emituje, zna da je došlo do sukobljavanja, pa prestaje da emituje i generiše 48-bitni rafalni šum da bi na sukob upozorila i ostale stanice. Drugim recima, ona zagušuje nosilac podataka kako bi i pošiljalac pouzdano znao da je došlo do sukobljavanja. Približno u trenutku 2T pošiljalac čuje rafalni šum i takođe prekida emitovanje. Zatim čeka proizvoljni period vremena i ponovo pokušava slanje.



Slika 4-18. Otkrivanje sukoba može potrajati i 2X.

Ako stanica počne da šalje veoma kratak okvir, može se pojmiti da *će* - ukoliko dođe do sukoba - stanica završiti s njegovim emitovanjem pre nego što joj (posle *2x)* stigne rafalni šum koji označava sukobljavanje. Stanica će tada pogrešno zaključiti da je uspešno poslala okvir. Da se ovo ne bi događalo, vreme slanja svih okvira mora biti veče od 2t, tako da emitovanje još uvek bude u toku u trenutku kada pošiljaocu stigne rafalni šum. Za lokalne mreže brzine 10 Mb/s, maksimalne dužine 2.500 m i s najviše četiri repetitora (prema specifikaciji 802.3), utvrđeno je da je vreme obilaska mreže (uključujući i vreme zadržavanja u četiri repetitora) u najgorem slučaju oko 50 ps. Prema tome, najkraći dozvoljeni okvir mora se emitovati tokom ovog vremena. Pri brzini 10 Mb/s, jedan bit traje 100 ns, tako da najkraći okvir koji ne pravi probleme mora da sadrži 500 bitova. Zbog sigurnosti, taj broj je zaokružen na 512 bitova, što čini okruglo 64 bajta. Okviri kraći od 64 bajta dopunjavaju se do 64 bajta u polju *Dopuna.*

S porastom brzine mreže, minimalna dužina okvira mora da raste ili da se odgovarajuće smanjuje dužina kabla. Za lokalnu mrežu dužine 2500 m koja radi brzinom 1 Gb/s, minimalna dužina okvira bila bi 6400 bajtova. Mogao bi se napraviti i kompromis, da minimalna dužina okvira bude 640 bajtova, ali da maksimalno rastojanje između stanica bude 250 m. Takva ograničenja postaju sve bolnija kako napredujemo ka mrežama brzine više gigabita u sekundi.

Poslednje polje Ethernet okvira je *Kontrolni zbir.* To je u stvari 32-bitni ključ za heširanje podataka. Ako neki bitovi podataka budu pogrešno primljeni (zbog smetnji na kablu), kontrolni zbir će skoro sigurno biti pogrešan i greška će biti otkrivena. Kontrolni zbir se podvrgava cikličnoj proveri redundanse (CRC), o kojoj smo govorili u 3. poglavlju. Ona ne ispravlja greške, već ih samo otkriva.

Kada je IEEE standardizovao Ethernet, u DIX format su unete dve izmene koje su prikazane na slici 4.17(b). Prvom izmenom *Preambula* je skraćena na sedam bajtova, a osmi bajt je iskorišćen kao graničnik *Početka okvira,* da bi se standard usaglasio sa standardima 802.4 i 802.5. Dragom izmenom, polje *Tip* je preimenovano u polje *Dužina.* Naravno, tako primalac ne bi znao šta da radi sa okvirom da problem nije pre- vaziđen unošenjem kratkog zaglavlja s potrebnim informacijama u polje s podacima. Format polja s podacima razmotrićemo kasnije u ovom poglavlju, kada budemo stigli do upravljanja logičkom vezom.

Nažalost, do trenutka objavljivanja standarda 802.3 u upotrebu je ušlo toliko hardvera i softvera za DIX Ethernet da proizvođači i korisnici nisu baš sa oduševljenjem prihvatili preimenovanje polja *Tip* u polje *Dužina.* Godine 1997, IEEE je pružio raku pomirenja i prihvatio oba formata. To se, na sreću, moglo učiniti jer su sva polja *Tip* koja su korišćena do 1997, imala vrednost veću od 1500. Prema tome, svaka vrednost manja ili jednaka 1500 mogla se tumačiti kao *Dužina,* a vrednost veća od 1500 kao *Tip.* Tako je IEEE uspeo da nametne jedinstven standard, a korisnike koji su nastavili po starom istovremeno oslobodio griže savesti.

4.3.4 Algoritam binarnog **eksponencijalnog** odustajanja

Razmotrimo sada način na koji se posle sukobljavanja bira proizvoljan period čekanja do ponovnog emitovanja. Koristićemo model sa slike 4-5. Posle nastanka sukoba, vreme se deli u intervale konačne dužine, jednake vremenu obilaska mreže u najnepovoljnijem slučaju (2x). Da bi se prilagodila najdužoj putanji koju dozvoljava Ethernet, dužina intervala je podešena na trajanje 512 bitova, tj. na 51,2 ps.

Posle prvog sukobljavanja, svaka stanica propušta 0 ili 1 vremenski interval i počinje ponovo da emituje. Ako se stanice sukobe, pa zatim obe slučajno izaberu isti broj, ponovo će doći do sukobljavanja. Posle drugog sukoba, svaka stanica na slučajan način bira broj 0, 1, 2 ili 3 i čeka toliko vremenskih intervala pre ponovnog emitovanja. Ako dođe i do trećeg sukobljavanja (verovatnoća za to je 0,2.5), tada se broj propuštenih vremenskih intervala pre ponovnog emitovanja bira na slučajan način iz intervala 0 do 2^3 - 1.

U opštem slučaju, posle *i* sukoba bira se slučajan broj iz intervala 0 do 2' - 1, i toliko se vremenskih intervala propušta pre ponovnog pokušaja emitovanja. Međutim, nakon deset ponovljenih sukoba, interval za biranje slučajnih brojeva ostaje zamrznut na vrednosti 1023. Posle 16 uzastopnih sukoba, kontroler prekida igra i izveštava računar o neuspehu. Dalji oporavak sistema zavisi od toga kako su organizovani viši slojevi.

Opisani algoritam binarnog eksponencijalnog odustajanja (engl. *binary exponential backoff)* automatski se prilagođava broju stanica koje istovremeno žele da emi- tuju. Kada bi interval iz koga se biraju slučajni brojevi za sve sukobe bio 1023, verovatnoća ponovnog sukobljavanja dve stanice bila bi zanemarljiva, ali bi se posle sukoba propuštalo više stotina vremenskih intervala, što bi dovelo do znatnog kašnjenja u prenosu podataka. S druge strane, kada bi svaka stanica posle sukobljavanja uvele odmah ponovo emitovala ili propuštala samo jedan vremenski interval, onda bi se pretpostavljenih 100 stanica koje pokušavaju da istovremeno emituju stalno ponovo sukobljavale, sve do trenutka kada bi (slučajno) njih 99 izabralo jedinicu, a ona preostala stanica nulu, na šta bi se moglo čekati godinama. Kada se interval iz koga se na slučajan način biraju brojevi širi eksponencijalno s brojem uzastopnih

sukoba, time se uvodi samo neznatno kašnjenje u slučaju sukobljavanja nekoliko stanica, ali se pri sukobljavanju velikog broja stanica problem razrešava u razumnom periodu. Broj 1023 kao gornja granica intervala slučajnih brojeva održava algoritam jednostavnim.

Dosad smo naučili da protokol CSMA/CD ne predviđa potvrđivanje okvira. Pošto nepostojanje sukoba nije isključiva garancija da su bitovi stigli neizmenjeni, primalac bi u pouzdanom prenosu morao da proveri kontrolni zbir, pa ako nađe daje tačan, da izvorišnom računani pošalje potvrdu o prijemu. Što se tiče protokola, u normalnim situacijama potvrda predstavlja samo još jedan okvir koji konkuriše za svoje mesto na kanalu zajedno s drugim okvirima. Međutim, jednostavna izmena algoritma za konkurentsko pristupanje kanalu omogućila bi przo potvrđivanje prijema okvira (Tokoro iTamaru, 1977). Trebalo bi samo za odredišnu stanicu rezervisati vremenski interval za konkurentsko pristupanje koji sledi neposredno iza uspešno poslatog okvira. Nažalost, standard ne predviđa takvu mogućnost.

4.3.5 Performanse Etherneta

Ispitajmo sada performanse Etherneta u uslovima gustog i konstantnog saobraćaja, tj. kada je *k* stanica uvek spremno da emituje. Stroga analiza algoritma binarnog eksponencijalnog odustajanja vrlo je složena. Zato ćemo slediti uprošćenje Metcalfea i Boggsa (1976) i pretpostaviti za svaki interval konstantnu verovatnoću ponovnog slanja. Ako svaka stanica tokom vremenskog intervala predviđenog za konkurentsko pristupanje emituje s verovatnoćom *p,* onda je verovatnoća *A* da će neka od njih zauzeti kanal u tom intervalu jednaka

$$A = k p \{1 - p)^{k \sim x}$$

A ima najveću vrednost pri *p = l/k,* a teži vrednosti l/e kada *k* teži beskonačnosti. Verovatnoća da blok predviđen za konkurentsko pristupanje sadrži tačno ./ intervala iznosi *A(* 1 - *A)i~~* *, tako daje prosečan broj intervala po bloku:

oo

$$7 = 0$$

Pošto svaki interval traje *2x,* prosečan period konkurencije je *w = 2x/A.* Ako pretpostavimo optimalno *p,* prosečan broj intervala konkurencije nikada nije veći od e, tako da vv može biti najviše *2xe ~ 5,4x.*

Ukoliko je za slanje prosečnog okvira potrebno *P* sekundi, u situaciji kada mnoge stanice imaju okvire za slanje, efikasnost kanala je

$$\text{Efikasnost kanala} = \frac{P}{P + 2x/A} \qquad (4\text{-}6)$$

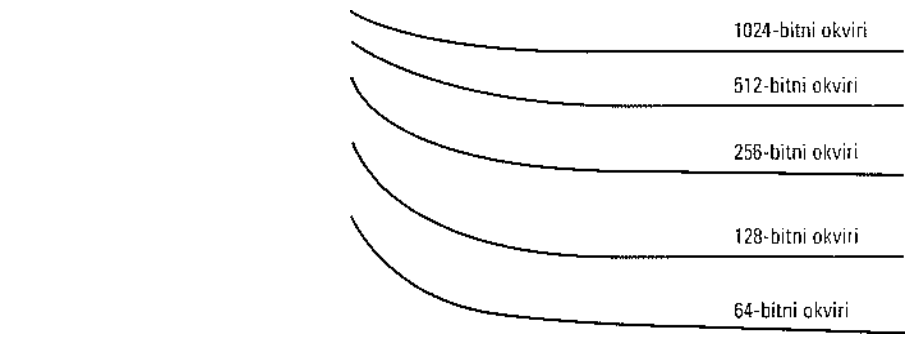
Ovde vidimo kako maksimalna dužina kabla između bilo koje dve stanice utiče na performanse, zbog čega se i teži da se topologija sa slike 4-15(a) pevaziđe. Što je duži kabl, duži je i period konkurencije. Zbog toga standard za Ethernet ograničava dužinu kabla.

Poučno je jednačinu (4-6) prikazati u funkciji dužine okvira *F,* propusne moći mreže *B,* dužine kabla *L* i brzine prostiranja signala *c* za optimalan slučaj *e* konkurentskih intervala po okviru. Uz *P = F/B,* jednačina (4-6) postaje

$$\text{Efikasnost kanala} = \frac{1}{1 + 2BLe/cF} \qquad (4\text{-}7)$$

Kada je drugi član imenioca veliki, efikasnost mreže postaje niska. Konkretno, povećanje propusnog opsega mreže ili rastojanja (proizvod *BL)* smanjuje efikasnost pri konstantnoj veličini okvira. Nažalost, u mrežni hardver se ulaže mnogo upravo u cilju povećanja ovog proizvoda. Korisnici žele veliki propusni opseg na dugačkom kablu (na primer, u optičkim gradskim mrežama), odakle se može zaključiti da Ethernet ugrađen na opisani način možda nije najbolji sistem za takve primene. Upoznacemo se i sa drugim načinima ugradnje Etherneta kada u nastavku poglavlja stignemo do komutiranog Etherneta.

Na slici 4-19, prikazana je efikasnost kanala u funkciji broja spremnih stanica za 2x = 51,2 ps i brzinu prenosa podataka lOMb/s, izračunata prema jednačini (4-7). Uz dužinu intervala od 64 bajta, nije čudno da je neefikasan i prenos okvira dužine 64 bajta. S druge strane, sa okvirima dužine 1024 bajta i asimptotskom vrednošću *e* broja 64-bajtnih intervala po periodu konkurencije, period konkurencije je dugačak 174 bajta, a efikasnost je 0,85.



Broj stanica koje pokušavaju da emituju

Slika 4-19. Efikasnost Etherneta pri brzini prenosa 10 Mb/s uz intervale od 512 bajtova.

Da bismo odredili prosečan broj stanica spremnih za emitovanje u uslovima gustog saobraćaja, možemo da iskoristimo sledeće (grubo) zapažanje. Svaki okvir „zatvara“ kanal tokom jednog perioda konkurencije i vremena potrebnog za prenos jednog okvira, u ukupnom tr ajanju $P + w$ sekundi. Broj okvira u sekundi je, prema tome, $1/(P + w)$.

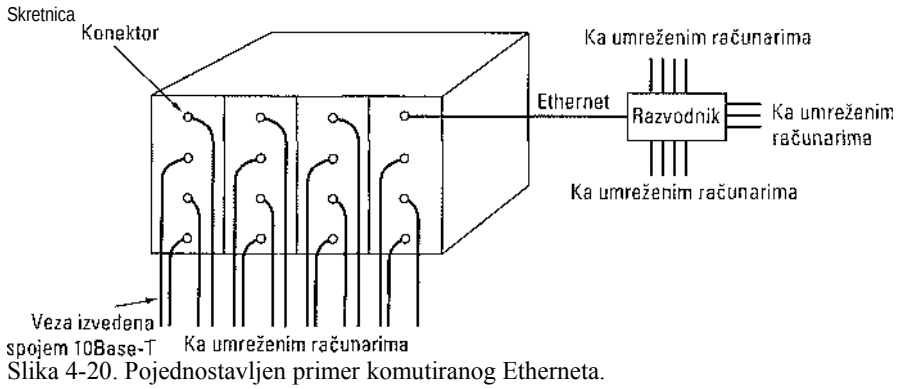
Ako svaka stanica generiše okvire srednjom brzinom À. okvira u sekundi, a sistem je u stanju *k,* onda je ukupna brzina slanja svih neblokiranih stanica *kX* okvira u sekundi. Pošto u stanju ravnoteže brzine slanja i primanja moraju biti jednake, možemo da izjednačimo ta dva izraza i da odatle izračunamo *k.* (Obratite pažnju na to da je *w* funkcija *k.)* Složeniju analizu možete da nađete kod Bertsekasa i Gallagera (1992).

Možda treba pomenuti da se performanse Etherneta (kao i drugih mreža) često teorijski analiziraju. Skoro u svim analizama se za gustinu saobraćaja pretpostavlja Poa- sonova raspodela. Kada su istraživači počeli da prikupljaju podatke iz prakse, pokazalo se da se Poasonova raspodela retko može primeniti, već daje saobraćaj „sebi sličan" (engl. *self-similar).* Za mrežni saobraćaj se kaže da je sebi sličan ako za svaki njegov deo postoji manji deo koji mu je sličan.) (Paxon i Floyd, 1994; Willinger i sar., 1995). To znači da nam izračunavanje srednje vrednosti saobraćaja tokom dužeg perioda neće olakšati analizu. Srednji minutni broj okvira u toku jednog sata varira isto koliko i srednji sekundni broj okvira tokom jednog minuta. Možemo zaključiti da se modelovanjem saobraćaja na mreži uglavnom ne dobija slika stvarnog stanja i da takve modele uvek treba uzeti „s trunkom (bolje s tonom) soli".

4,3.6 Komutirani Ethernet

Sto više stanica uključujemo u Ethernet, saobraća.) postaje sve gušći. Na kraju će se mreža zasititi. Da bi se to prevazišlo, može se primeniti veća brzina prenosa: ume- sto 10 Mb/s, recimo, 100 Mb/s. Međutim, kada uzmemo u obzir tempo razvoja multimedije, lako se mogu zasititi čak i gigabitne mreže.

Na sreću, postoji i drugi način borbe s povećanim saobraćajem: komutirani Ethernet (slika 4-20). Jezgro tog sistema je skretnica (engl. *switch*) koja na brzoj osnovnoj ploči (engl. *backplane)* najčešće ima mesta za 4 do 32 linijske kartice (engl. *plug-in Une card),* od kojih svaka sadrži jedan do osam konektora. Svaki konektor povezuje jedan umreženi računar preko upredene parice i spoja 10Base-T.



Slika 4-20. Pojednostavljen primer komutiranog Etherneta.

Kada stanica zaželi da pošalje Ethernet okvir, ona skretnici pošalje standardan okvir. Linijska kartica kojoj stigne takav okvir može da proveri da li je on namenjen računani priključenom na istu karticu. Ako je tako, okvir se kopira na njega. Ako nije, okvir se preko

brze osnovne ploče šalje kartici na koju je priključen odredišni računar. Osnovna ploča obrađuje mnogo Gb/s, koristeći za to poseban protokol koji je vlasništvo proizvrođača (engl. *proprietcvj protocol)*.

Šta se dešava ako dva računara priključena na istu linijsku karticu istovremeno pošalju svoje okvire? To zavisi od konstrukcije kartice. Ako su svi priključci na kartici međusobno povezani tako da obrazuju lokalnu „mini mrežu“, onda će sukobi na njoj biti otkrivani i obrađivani kao i u bilo kojoj CSMA/CD mreži, a sukobljeni okvir će se ponovo slati uz primenu algoritma binarnog eksponencijalnog odustajanja. Takva kartica u jednom trenutku može da obradi samo jedan zahtev za prenos podataka, ali više njih u skretnici mogu istovremeno da rade različite poslove. Svaka kartica obrazuje svoj domen sukobljavanja (engl. *collision domain),* nezavisno od ostalih kartica. Uz samo jednu stanicu po domenu sukobljavanja, sukobljavanje postaje nemoguće i performanse se poboljšavaju.

Kod kartica druge vrste, svakom ulaznom priključku dodeljuje se bafer tako da se dolazni okviri redom smeštaju u RAM memoriju na kartici. Konstrukcija omogućava istovremeno primanje (i slanje) okvira preko svih ulaznih priključaka - paralelan, potpun dupleksni režim - nešto što nije moguće postići protokolom CSMA/CD na jedinstvenom kanalu. Kada okvir stigne u celini, kartica proverava da li je namenjen priključku na istoj ili na nekoj dragoj kartici. U prvom slučaju, može ga odmah uputiti na odredište, a u drugom - samo preko brze osnovne ploče. Uz opisanu konstrukciju, svaki priključak predstavlja zaseban domen sukobljavanja, tako da do sukobljavanja ne dolazi. Ukupan protok podataka često je za red veličine veći od protoka kroz sistem 10Base5 koji ima jedinstven domen sukobljavanja.

Pošto skretnica na svakom ulaznom priključku očekuje samo standardni Ethernet okvir, neki priključci se mogu koristiti kao koncentratori. Priključak na slici 4-20 koji se nalazi u gornjem desnom uglu skretnice nije direktno povezan s računarom, već s razvodnikom koji i sam sadrži 12 priključaka. Okviri koji stižu u razvodnile konkurišu jedan drugome na uobičajen način - sudarajući se i odustajući. Sa okvirima koji se uspešno probiju do skretnice tamo se postupa kao i s drugim pristiglim okvirima: oni se preko brze osnovne ploče upućuju na odgovarajuću izlaznu liniju. Razvodnici su jeftiniji od skretnica, ali kako cene skretnica padaju, razvodnici se sve više napuštaju. Pa ipak, i dalje postoje razvodnici nasleđeni iz starijih sistema.

4.3.7 Brzi Ethernet

U početku je 10 Mb/s izgledalo „brzo kao munja“, baš kao što su se i modemi od 1200 b/s prsili nad akustičkim modemima brzine 300 b/s. Međutim, svaka novost za tri dana. Nije jasno da li je u pitanju još jedna manifestacija Parkinsonovog zakona („Posao teži da se rastegne na vreme raspoloživo za njegovo obavljanje“), tek izgleda da podaci uvek teže da potpuno ispune propusni opseg koji je na raspolaganju za njihovo prenošenje. Da bi povećale brzinu prenosa, različite industrijske grupacije predložile su dve nove vrste prstenastih lokalnih mreža zasnovanih na optičkim kablovima. Jedna je nazvana **interfejs za podatke distribuirane optičkim kablom**
(engl. *Fiber Distributeri Data Inteiface, FDDI),* a druga jednostavno **optički kanal** (engl. *Fibre Channel)*. (Da, baš „fibre“, a ne „fiber“, jer je urednik dokumenta bio Englez.) I, da skratimo priču, iako su oba sistema korišćena za mrežne okosnice, nijedan se nije probio do lokalne mreže. Rad sa stanicama u oba slučaja bio je previše složen, pa je zahtevao složene i

skupe čipove. Pouka: Majstore, ne komplikuj!

U takvoj situaciji, IEEE je 1992. ponovo sazvao komitet za mrežu 802.3 i zadao mu da pronađe rešenje za bržu lokalnu mrežu. Jedan od predloga bio je da se zadrži postojeća mreža 802.3, samo da se ubrza. Drugi su predlagali da se mreža 802.3 iz ko- rena izmeni, da joj se doda mnogo novih svojstava (saobraćaj u realnom vremenu, di- gitalizovani glas...), ali da se iz markentiških razloga zadrži staro ime. Posle mnogo trvenja, odlučeno je da se mreža 802.3 u osnovi ne menja, samo da se ubrza. Predla- gači drugog rešenja uradili su ono što se u navedenim okolnostima i moglo očekivati od industrijalaca - istupili su, osnovali sopstveni komitet i standardizovali sopstvenu lokalnu mrežu (kao 802.12). Ona je jadno završila.

Komitet za mrežu 802.3 odlučio se da samo ubrza Ethernet prvenstveno zbog:

1. Potrebe za kompatibilnošću s postojećim lokalnim Ethernet mrežama.
2. Straha da bi nov protokol mogao doneti nepredviđene probleme.
3. Zelje da posao obavi pre nego što se promeni tehnologija.

Posao je dovršen brzo (po merilima komiteta za standardizovanje), a IEEE je rezultat, mrežu **8Q2.3u,** zvanično ustoličio juna 1995. S formalnog stanovišta, mreža 802.3u nije predstavljena kao nov standard, već kao dopuna postojećeg standarda 802.3 (da bi se istakla kompatibilnost s postojećim standardom). Pošto je svi zovu brzi Ethernet *(engl. fast Ethernet),* i mi ćemo se u nastavku tome pridružiti.

Osnovna zamisao pri projektovanju brzog Etherneta bila je prilično jednostavna: zadržati postojeće formate okvira, interfejse i proceduralna pravila, ali skratiti trajanje jednog bita sa 100 ns na 10 ns. Tehnički je bilo izvodljivo da se kopira sistem 10Base-5 ili 10Base-2, ada se sukobi još uvek pravovremeno otkrivaju ako se dužina kabla desetostruko smanji. Međutim, prednosti ožičenjapo sistemu lOBase-T bile su tako velike daje projekat brzog Etherneta zasnovan isključivo na njemu. Na taj način, u svim sistemima brzog Etherneta postoje razvodnici i skretnice; spojni kablovi s više ubodnih račvi ili BNC konektora nisu dozvoljeni.

Pa ipak, o nečemu se moralo i odlučivati, najpre o podržanoj vrsti žice. Jedna moguća varijanta bila je upredena parica 3. kategori je, za koju je glavni argument bio to da je skoro svaka kancelarija na Zapadu opremljena s barem četiri upredene parice barem 3. kategorije koje vode do telefonskog razvodnog ormana u dometu od 100 m. Ponekada postoje i dva takva kabla. Pema tome, kada bi se podržala upredena parica
3. kategorije, mogao bi se dovesti brzi Ethernet do računara bez potrebe za novim ožičavanjem zgrade, što bi bila ogromna prednost za mnoge organizacije.

Glavni nedostatak upredene parice 3. kategorije jeste to što ona ne može da prenosi signale od 200 megaboda (100 Mb/s uz Mančester kodiranje) na daljinu od 100 m, što je najveće rastojanje između razvodnika i računara specificirano za sistem lOBase-T (slika 4-13). Nasuprot tome, upredena parica 5. kategorije to može lako, a optičkim vlaknom signal se može preneti i na mnogo veću udaljenost. Na kraju je odlučeno da se dozvole sve tri mogućnosti (slika 4-21), s tim da se i dalje radi na parici 3. kategorije da bi se postigao potreban kapacitet prenosa.

Ime	Kabl	Maksimalna dužina segmenta	Prednosti
100Base-T4	Upredena parica	100 m	Koristi se neoklopljena parica 3. kategorije
100Base-TX	Upredena parica	100 m	Potpuni dupleks pri 100 Mb/s (neoklopljena parica 5. kategorije)
100Base-FX	Optičko vlakno	2000 m	Potpuni dupleks pri 100 Mb/s; velika rastojanja

Slika 4-21. Originalno kabliranje brzog Etherneta.

Sistem s neoklopljenom upredenom paricom 3. kategorije, pod imenom lOOBase- T4, radi uz brzinu signala 25 MHz, što je samo 25 procenata brže od 20 MHz standardnog Etherneta (setite se Mančester kodiranja, prikazanog na slici 4-16, za koje su potrebna dva otkucaja sata za svaki od 10 miliona bitova koji se šalju u sekundi). Sistem 100Base-T4 može da ostvari taj propusni opseg sa samo četiri upredene parice. Pošto standardna telefonska mreža već decenijama ima četiri upredene parice po kablu, povezivanje poslovnih prostorija sistemom 100Base-T4 obično ide bez problema. Naravno, to znači da treba da odustanete od telefona u kancelariji, ali za uzvrat dobijate bržu elektronsku poštu.

Od četiri upredene parice, jedna uvele vodi ka razvodniku, druga uvele vodi od njega, a preostale dve mogu se po potrebi iskoristiti za tekući prenos podataka. Potreban propusni opseg ostvaruje se tako što se ne koristi Mančester kodiranje jer uz suvremene satove i kratka rastojanja ono više nije potrebno. Osim toga, emituje se signal s tri naponska nivoa, tako da tokom svakog otkucaja sata intenzitet signala može da odgovara nuli, jedinici ili dvojki. Uz tri upredene parice u smeru emitovanja i trojno signaliziranje, može se preneti svaki od 27 mogućih simbola, što omogućava slanje 4 bita uz izvestan višak. Prenošenjem 4 bita tokom svakog od 25 miliona impulsa u sekundi postiže se neophodnih 100 Mb/s. Osim toga, na četvrtoj parici uvele postoji povratni kanal brzine 33,3 Mb/s. Ovaj sistem, poznat kao 8B/6T (8 bitova preslikano u 6 trojnih signala), nije baš elegantan, ali radi u postojećem ožičenju.

Za ožićenje neoklopljenom upredenom paricom 5. kategorije jednostavniji je sistem 100Base-TXjer te parice mogu da rade pri brzini sistemskog sata 125 MHz. Koriste se samo dve parice po stanici - ka razvodniku i od njega. Ne koristi se direktno binarno kodiranje, već sistem 4B/5B koji je preuzet od protokola FDDI i kompatibilan je s njim. Svaka grupa od pet uzastopnih otkucaja sata, uz korišćenje dve naponske vrednosti, omogućava 32 kombinacije. Šesnaest takvih kombinacija koristi se za prenošenje 4-bitnih grupa 0000, 0001, 0010,..., 1111. Neke od preostalih šesnaest kombinacija koriste se za upravljanje, na primer, za obeležavanje granica okvira.

Upotrebljene kombinacije tako su izabrane da se ostvaraju jasni prelazi potrebni za sinhronizovanje. 100Base-TX je potpun dupleksni sistem; stanice istovremeno mogu i da emituju i da primaju podatke brzinom 100 Mb/s. Često se sistemi 100Base-TX i 100Base-T4 nazivaju zajedničkim imenom **lOOBase-T.**

U sistemu **100Base-FX,** poslednjem koji razmatramo, koriste se dva snopa više-režimskih (multimodnih) optičkih vlakana, po jedno za svaki smer, tako da je i to potpun dupleksni sistem brzine 100 Mb/s u svakom smeru. Pored toga, razdaljina između stanice i razvodnika može da bude i 2 km.

Kao odgovor na zahteve korisnika, komitet za mrežu 802 je 1997. godine standardizovao nov način kabliranja 100Base-T2 koji omogućava da brzi Ethernet radi pomoću dve postojeće parice 3. kategorije. Međutim, zbog složene šeme kodiranja, sistemu je neophodan poseban procesor digitalnog signala koji mu znatno podiže cenu. Sistem do danas nije često korišćen, kako zbog složenosti i cene, tako i zbog činjenice da su mnoge poslovne zgrade već uvele ožičenje paricom .5. kategorije.

Kao što se vidi na slici 4-20, za sistem lOOBase-T postoje dve vrste spojnih uređaja: razvodnici i skretnice. Kod razvodnika, sve ulazne linije su logički povezane (ili barem linije koje dolaze do iste kartice), tako da obrazuju jedinstven domen sukobljavanja. Primenjuju se sva standardna pravila, uključujući i algoritam binarnog ekspo- nencijalnog odustajanja, tako da sistem prividno radi kao i standardni Ethernet. Međutim, u jednom trenutku samo jedna stanica može da šalje podatke, tako da raz- vodnik radi u poludupleksnom režimu.

Kada se upotrebi skretnica, svaki dolazni okvir privremeno se smešta u bafer na kartici, a zatim, po potrebi, preko brze osnovne ploče prosleđuje odredišnoj kartici. Postoje ploča skrivena u unutrašnjosti skretnice, nije bilo potrebe da se ona standar- dizuje. Ako sudimo na osnovu dosadašnjeg iskustva, proizvođači će se verovatno posvetiti razvijanju ploča sa što bržim kolima da bi povećali protok podataka kroz sistem. Kablovi sistema 100Base-FX predugačld su zakorišćenje normalnog algoritma za otkrivanje sukoba na Ethernetu, pa se njihovi segmenti moraju povezivati preko skretnica, pri čemu svaki segment predstavlja zaseban domen sukobljavanja. U sistemu 100Base-FX nisu dozvoljeni razvodnici.

Pomenimo na kraju da sve skretnice mogu da rade sa stanicama brzine i 10 i 100 Mb/s, što olakšava nadogradnju mreže. Za svaku novougrađenu stanicu brzine 100 Mb/s treba samo u skretnicu umetnuti novu linijsku utičnu karticu. U stvari, standard predviđa mogućnost da dve stanice dogovore optimalnu brzinu (10 ili 100 Mb/s) i režim prenosa (potpuni ili poluduplelcs). Većina brzih Ethernet mreža tu mogućnost koristi da bi automatski podesile svoje parametre.

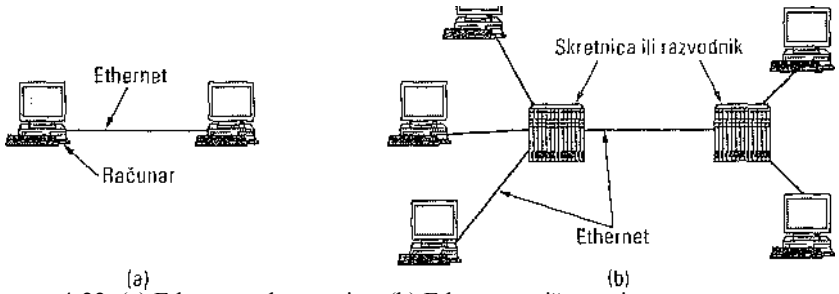
4.3.8 Gigabitni Ethernet

Verovatno je standard za brzi Ethernet još mirisao „na štampu“, kada je komitet za mrežu 802 započeo rad na standardizovanju još brže Ethernet **mreže** (1995). Ona je ubrzo dobila nadimak **gigabitni Ethernet** (engl. *gigabit Ethernet*), a zvanično ime 802.3z dao joj **je** 1998. godine IEEE. Takvo ime navodi na pomisao da gigabitni
Ethernet treba da bude završetak ovog tipa mreža, osim ako neko ne izmisli i slovo koje dolazi posle z. U odeljku koji sledi razmotrićemo glavne osobine gigabitnog Etherneta. Detaljnija objašnjenja potražite kod Seiferta (1998).

Autori mreže 802.3z imali su isti cilj kao i autori mreže 802.3u: povećati brzinu Etherneta 10 puta, a da on ipak ostane kompatibilan sa svim postojećim prethodnim verzijama. Konkretno, trebalo je da gigabitni Ethernet obezbedi uslugu datagrama bez potvrđivanja prijema (uz jednosmemo i višesmerno emitovanje), da koristi postojeću 48-bitnu šemu adresiranja, kao i format okvira, uključujući njegovu minimalnu i njegovu maksimalnu veličinu. Kada je bio dovršen, standard je ispunio sve zadate ciljeve.

Sve konfigurcije gigabitnog Etherneta, umesto spojnog kabla s više priključaka - kako to predviđa klasični Ethernet (engl. *classic Ethernet),* brzine 10 Mb/s - koriste veze od tačke do tačke. U svojoj najjednostavnijoj konfiguraciji, prikazanoj na slici

4- 22(a), gigabitni Ethernet se sastoji od samo dva, međusobno direktno povezana računara. U opštijem slučaju, postoji razvodnik za koji je vezano više računara, a mogu postojati i dodatni razvodnici, odnosno skretnice, kao na slici 4-22(b). U obe konfiguracije, međutim, svaki pojedinačni Ethernet kabl spaja samo dva uređaja.



Slika 4-22. (a) Ethernet s dve stanice, (b) Ethernet s više stanica.

Gigabitni Ethernet može da radi u dva režima: poludupleksnom (engl. *half-duplex mode)* i potpunom dupleksnom režimu (engl *.full-duplex mode).* Podrazumevani režim je potpuni dupleks koji istovremeno omogućava saobraćaj u oba smera. On se koristi kada postoji centralna skretnica povezana s računarima (ili i s drugim skretnicama) na periferiji. U takvoj konfiguraciji, saobraćaj iz svih linija privremeno se smešta u bafer, tako da skretnica može da šalje okvire kad god to poželi. Pošiljalac ne mora da osluškuje da bi utvrdio da lije kanal „prazan“ jer je sukobljavanje nemoguće. Na liniji koja povezuje računar sa skretnicom, računar je jedini mogući pošiljalac ka skretnici i takvo slanje uspeva čak i onda kada skretnica u istom trenutku šalje okvir računara jer linija radi kao potpuni dupleks. Pošto je sukobljavanje nemoguće, ne koristi se protokol CSMA/CD, pa dužina kabla ne zavisi od vremena potrebnog da ra- falni šum pod najnepovoljnijim uslovima stigne do pošiljaoca, već isključivo od jačine signala. Skretnice autonomno menjaju i usuglašavaju brzine prenosa. Podržano je automatsko podešavanje parametara sistema, kao kod brzog Etherneta.

Drugi, poludupleksni režim rada, koristi se kada računari nisu povezani skretnicom, već razvodnikom. Razvodnik ne smešta dolazne okvire u bafer, već u svojoj unutrašnjosti povezuje sve linije oponašajući tako spojni kabl s više priključaka koji se koristi u klasičnim Ethernet mrežama. Sukobljavanje je u ovom režimu moguće, tako da je neophodno koristiti standardni protokol CSMA/CD. Pošto se najmanji (64-bajtni) okvir sada može slati 100 puta brže, maksimalna udaljenost je 100 puta man ja (25 m), nego kod klasičnog Etherneta da bi se očuvao uslov prema kome pošiljalac i u najne- povoljnijem slučaju treba da još uvek šalje okvir u trenutku kada mu (i ako mu) stigne rafalni šum. Pošiljalac 64-bajtnog okvira koji emituje brzinom 1 Gb/s u kabl dužine 2500 m, završio bi slanje pre nego što bi rafalni šum prevalio i deseti deo puta u jednom smeru, a kamoli tamo i natrag.

Komitet za mrežu 802.3z zaključio je da je rastojanje od samo *25 m* između sused- nih uređaja neprihvatljivo i zato je u standard uneo dve dopune. Prva se odnosi na proširenje nosioca podataka (engl. *carrier extension)',* ona nalaže hardvera da sopst- venim resursima dopuni normalan okvir do .512 bajtova. Pošto dopunu generiše hardver pošiljaoca, a uklanja je hardver primaoca, softver o tome nema pojma - znači ne treba ga menjati. Naravno, kada

se za prenošenje 46 bajtova korisničkih podataka (koristan teret 64-bajtnog okvira) utroši 512 bajtova propusnog opsega, iskorišćenje veze je 9%.

Druga dodata osobina, tzv. bujica okvira *(engl. frame bursting),* omogućava pošiljaocu da u jednom prenosu pošalje ulančanu sekvencu više okvira. Ako je pojedinačna bujica kraća od .512 bajtova, ponovo se dopunjava hardverski. U situacijama kada mnogi okviri čekaju slanje, opisani postupak je veoma efikasan i poželjniji od proširenja nosioca podataka. Uvedene osobine proširuju radijus mreže na 200 m, što je verovatno dovoljno za većinu poslovnih prostorija.

Među nama, tešlco je zamisliti organizaciju koja bi se trudila da nabavi i instalira kartice za gigabitni Ethernet kako bi postigla visoke performanse, a zatim povezala računare razvodnikom i tako simulirala klasični Ethernet sa svim njegovim sukobljavanjima. Iako su razvodnici nešto jeftiniji od skretnica, mrežne kartice za gigabitni Ethernet i dalje su prilično skupe. Prema tome, kalkulacija da se kupi jeftin razvodnik i time oslabe performanse sistema, nema nikakvog stvarnog osnova. Kompatibilnost s prethodnim verzijama i dalje je Sveto pismo računarske industrije, pa je komitet za mrežu 802.3z odlučio daje i ovde ostvari.

Gigabitni Ethernet podržava i bakarni i optički kabl (slika 4-23). Signaliziranje brzinom bliskom 1 Gb/s znači da svetlosni izvor treba da se upali i ugasi za manje od 1 ns. Svetlosne diode (LED) ne mogu da rade tako brzo, pa se zato koriste laseri. Dozvoljene su dve talasne dužine: 0,85 pm (manja) i 1,3 pm (veća). Laseri koji emitu- ju svetlost talasne dužine 0,85 pm koštaju manje, ali ne rade s jednorežimskim (rnono- modnim) vlaknima.

Za optičko vlakno su dozvoljena tri prečnika: 10, .50 i 62,5 pm. Prvi je za jedno-režimski, a druga dva za višerežimski rad. Nisu, međutim, dozvoljene sve kombinacije, a maksimalna udaljenost između uređaja zavisi od konkretne kombinacije. Vrednosti na slici 4-23 odgovaraju optimalnoj situaciji. Konkretno, rastojanje od 5000 m može se postići samo laserom talasne dužine 1,3 pm kroz vlakno debljine 10 pm u jednorežimslcom radu, ali je to najbolje rešenje za povezivanje više zgrada na širem području i očekuje se da bude prihvaćeno, uprkos tome što je i najskuplje.

Ime	Kabl	Maksimalna dužina segmenta	Prednosti
1000Base-SX	Optički	550 m	Višerežimsko vlakno (50; 62,5 pm)
1000Base-LX	Optički	5000 m	Jednorežimsko (10 pm) ili višerežimsko (50; 62,5 pm)
1000Base-CX	2 oklopljene parice	25 m	Oklopljena upredena parica
1000Base-T	4 neoklopljene parice	100 m	Standardna neoklopljena parica 5. kategorije

Slika 4-23. Kabliranje gigabitnog Etherneta.

U sistemu 1000Base-CX koriste se kratki oklopljeni bakarni kablovi. Problem je u tome što taj sistem treba da se takmiči s prethodnim sistemom koji radi sa optičkim kablom visokih performansi, kao i sa sledećim, u kome se koriste jeftine neoklopljene parice. Mali su izgledi da uopšte uđe u upotrebu.

Poslednji sistem obuhvata snop od 4 neoklopljene parice 5. kategorije koje rade tim ski. Pošto u postojećim instalacijama već ima mnogo takvih žica, to će verovatno biti način

realizacije gigabitnog Etherneta za one koji nemaju sredstava za razbacivanje.

U gigabitnom Ethernetu sa optičkim kablovima, koriste se nova pravila kodiranja. Mančester kodiranje pri brzini prenosa 1 Gb/s može se postići samo uz signal od 2 Gboda, što je teško, a u pogledu iskorišćenja propusnog opsega i rasipno. Zbog toga je za optičke kanale uveden nov način kodiranja (8B/10B). Svaki 8-bitni bajt u optičkom vlaknu kodira se sa 10 bitova - otuda i naziv 8B/10B. Pošto za svaki ulazni bajt postoji 1024 moguće kodne reči, imamo izvesnu slobodu da između njih izaberemo one koje su dozvoljene. Izbor se vrši prema sledećim pravilima:

1. Nijedna kodna reč ne sme da ima više od četiri identična bita u nizu.
2. Nijedna kodna reč ne sme da ima više od 6 jedinica, niti više od 6 nula.

Navedena pravila su uvedena da bi se u toku bitova održali jasni prelazi pomoću kojih primalac održava korak s pošiljaocem, a i da bi se broj jedinica i nula na vlaknu što više uravnotežio. Osim toga, mnogim ulaznim bajtovima dodeljene su po dve moguće kodne reči. Kada uređaj (program) za kodiranje treba da bira jednu od njih, uvek će izabrati reč koja pomaže izjednačavanju broja nula i jedinica u dotadašnjem pre- nosu. Naglasak koji je stavljen na uravnotežavanje nula i jedinica proizašao je iz potrebe da se jednosmerna komponenta signala održi na što nižem nivou kako bi kroz transformatore prošla neizmenjena. Premda računarski stručnjaci nisu baš oduševljeni time što im svojstva transformatora diktiraju način kodiranja, od stvarnosti se po- nekada ne može pobeći.

Gigabitni Ethernet tipa lOOOBase-T koristi drugačiji način kodiranja, s obzirom da je „ubacivanje podataka u bakarnu žicu“ svake nanosekunde previše složen zadatak. U sistemu lOOOBase-T koriste se četiri upredene parice 5. kategorije koje omogućavaju istovremeni prenos četiri simbola. Svaki simbol se kodira pomoću jednog od pet naponskih nivoa: 00,01,10,11 ili specijalnog nivoa za svrhe upravljanja. Na taj način se može preneti 2 bita podataka po parici ili 8 bitova podataka po jednom taktu sistemskog sata. Sistemski sat je frekvencije 125 MHz, što omogućava prenos brzinom 1 Gb/ s. Ovde se umesto četiri, koristi pet naponskih nivoa da bi se neke kombinacije mogle iskoristiti za uokvirivanje i upravljanje.

Gigabit u sekundi je prilično velika brzina. Ako je primalac nečim trenutno zauzet, pa samo u toku 1 ms ne isprazni ulazni bafer na jednoj liniji, za taj „tren“ će se u njemu nakupiti do 1953 okvira. Isto tako, kada računar s gigabitnog Etherneta isporučuje podatke računam na klasičnom Ethernetu, vrlo lako može da dođe do preli- vanja bafera. Zbog ove dve neugodne mogućnosti, gigabitni Ethernet predviđa i kontrolu toka (kao u brzom Ethernetu, iako on radi drugačije).

Kontrola toka se sastoji u tome što jedna strana dragoj pošalje specijalan upravljački okvir s nalogom da privremeno prekine emitovanje. To je normalan Ethernet okvir koji sadrži tip 0x8808. Prva dva bajta polja s podacima predstavljaju komandu, dok ostali bajtovi sadrže eventualne parametre. Za kontrolu toka koristi se komanda PAUSE, a parametrima se naznačava dužina pauze kao umnožak minimalnog trajanja okvira. U gi- gabitnom Ethernetu ta vremenska jedinica je 512 ns, što omogućava pauze do 33,6 ms.

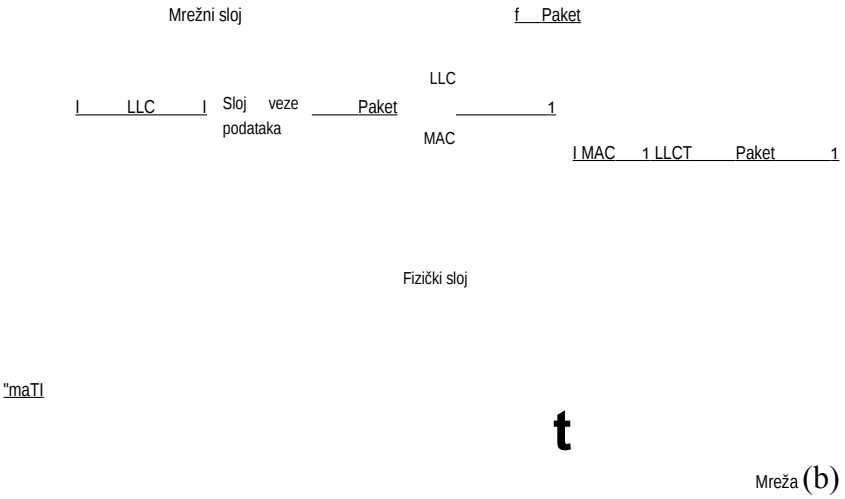
Čim je standarđizovan gigabitni Ethernet, komitet za mrežu 802 počeo je da se dosađuje i odmah je tražio nov posao. IEEE im je tada naložio da razrade standard za Ethernet brzine 10 Gb/s. Posle bezuspešne potrage za slovom koje bi sledilo z, pre- orijentisali su se na dvoslovne nastavke. Bacili su se na posao i IEEE je 2002. godine potvrdio nov standard 802.3ae. Koliko ćemo čekati na 100-gigabitni Ethernet?

4.3.9 **IEEE 802.2:** upravljanje **logičkom** vezom

Možda je vreme da predahnemo i uporedimo ono što smo naučili u ovom poglavlju sa onim što smo naučili ranije. U 3. poglavlju smo videli da dva računara mogu da pouzdano komuniciraju preko nepouzdane linije koristeći razne protokole sloja veze. Ti protokoli su obezbeđivali kontrolu grešaka (pomoću potvrda o prijemu okvira) i kontrolu toka (pomoću kliznih prozora).

Nasuprot tome, u ovom poglavlju nismo ni pomenuli pouzdanu komunikaciju. Najbolje što Ethernet i dragi 802 protokoli mogu da ponude jeste usluga datagrama. Međutim, takva usluga je ponekada sasvim dovoljna. Na primer, kada se prenose IP paketi, garancija se niti traži, niti očekuje. IP palcet se može direktno smestiti u polje s podacima mreže 802 i poslati na odredište. Ako se usput izgubi, nije važno.

Pa ipak, postoje i sistemi gde je poželjno imati protokol sloja veze koji ispravlja greške i upravlja tokom. IEEE je definisao takav protokol koji se izvršava iznad protokola za Ethernet i drugih protokola mreže 802. Taj protokol za upravljanje logičkom vezom (engl. *Logical Link Control, LLC)* skriva razlike između različitih vrsta mreža 802 obezbeđujući jedinstven format okvira i jedinstven interfejs ka mrežnom sloju. Format, interfejs i sam protokol u bliskoj su vezi s protokolom HDL.C koji smo opisali u 3. poglavlju. Protokol LLC obrazuje gornju polovinu sloja veze podataka, dok je donja polovina MAC podsloj (slika 4-24).



Slika 4-24. (a) Mesto LLC podsloja. (b) Formati protokola.

Protokol LLC obično se koristi na sledeći način. Mrežni sloj pošiljaoca prosleđuje paket pođsloju LLC koristeći njegove osnovne usluge za pristupanje. Paketu se u LLC pođsloju dodaje LLC zaglavlje koje sadrži redni broj i broj za potvrđivanje. Ta struktura se umeće u polje za korisničke podatke normalnog okvira mreže 802 i šalje. Kod primaoca se postupale odvija obrnuto.

LLC obezbeđuje tri vrste usluga: nepouzdanu uslugu datagrama, uslugu datagra- ma s potvrđivanjem prijema i pouzdanu uslugu sa uspostavljanjem direktne veze, LLC zaglavlje sadrži tri polja: odredišnu pristupnu tačku, izvorišnu pristupnu tačku i kontrolno polje. Pristupne tačke ukazuju na proces koji je generisao okvir i na proces kome treba da bude isporučen, zamenjujući time polje *Tip* u okviru DIX formata. Kontrolno polje sadrži redni broj i broj za potvrđivanje, u stilu protokola HDLC (slika 3- 24), ali ne i identično s njim. Ta polja se prvenstveno upotrebljavaju onda kada je potrebna pouzdana veza u sloju veze podataka, pri čemu se koriste protokoli slični protokolima iz 3. poglavlja. Za svrhe Interneta dovoljno je da se paketi šalju na najbolji mogući način i da se od LLC podsloja ne traži potvrda o njihovom prijemu.

4.3.10 Retrospektiva **Etherneta**

Ethernet je s nama već 20 godina i nema ozbiljnijeg konkurenta, tako da ćemo se s njim verovatno još družiti. Malo je mikroprocesora, operativnih sistema ili programskih jezika koji su izdržali toliko dugo. U čemu je zapravo stvar? Staje to toliko dobro kod Etherneta?

Osnovni uzrok dugovečnosti Etherneta verovatno je njegova jednostavnost i elastičnost. U praksi se jednostavnost ogleda u pouzdanosti, jeftinoći i lakoći održavanja. Čim su ubođne račve zamenjene BNC priključcima, kvarovi mreže su se sasvim proredili. Ljudi oklevaju da zamene nešto što sve vreme radi savršeno, naročito zato što znaju da gomila stvari u računarskoj industriji radi jadno, pa su mnoge „nadgradnje“ lošije od prethodnih rešenja.

Jednostavno znači i jeftino. Za tanki Ethernet i ožičenje upredenom paricom ne treba mnogo sredstava. Mrežne kartice takođe nisu skupe. Značajne investicije su samo razvodnici

i skretnice, ali u trenutku kada su se pojavili, Ethernet je već imao lep radni staž.

Ethernet se lako održava. Nema softvera koji treba instalirati (osim upravljačkih programa), nema ni tabela s parametrima konfiguracije koje treba održavati (pa ni mogućnosti da se pri tome pogreši). Nove računare je potrebno samo (fizički) priključiti na mrežu.

Značajno je i to da Ethernet s drugim mrežama sarađuje koristeći protokol TCP/IP, lcoji dominira tom oblašću. I IP je protokol bez uspostavljanja direktne veze, što se savršeno slaže s načinom rada Etherneta. IP se mnogo manje slaže sa ATM mrežama jer one rade sa uspostavljanjem direktne veze, što im definitivno umanjuje šanse.

Na kraju, Ethernet je dokazao da može da se razvija u pravcima koji su od najvećeg značaja. Brzine su porasle više redova veličine, uvedeni su razvodnici i skretnice, ali pri svemu tome nije bilo potrebno menjati softver. Kada prodavač pokaže prstom na neku veliku instalaciju i kaže: „Evo mreže za vas. Treba samo da zamenite sav hardver i ponovo napišete softver“, i sam zna da će to biti problem. Kada su bile uvođene mreže tipa FDDI, Fibre Channel i ATM, sve su bile brže od Etherneta, ali su bile s njim nekompatibilne, složenije i teže za održavanje. Na kraju ih je Ethernet dostigao u pogledu brzine, time im izbio iz rake i poslednji adut i one su jedna za drugom tiho nestale, osim sistema ATM koji se duboko ušančio u jezgru telefonskog sistema.

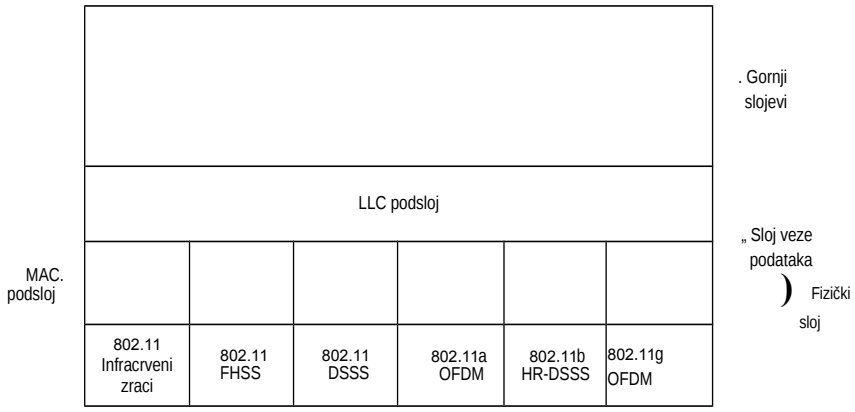
4.4 BEŽIČNE LOKALNE MREŽE

Iako se Ethernet danas koristi u najvećoj meri, preti mu konkurencija. Bežične lokalne mreže (engl. *wireless LANs)* postaju sve popularnije i njima se oprema sve više poslovnih zgrada, aerodroma i drugih javnih objekata. Bežične lokalne mreže mogu da rade u jednoj od dve konfiguracije (slika 1-35): s baznom stanicom i bez nje. Shodno tome, standard za bežične lokalne mreže (802.11) uvažava obe konfiguracije, kao Što ćete se ubrzo i sami uveriti.

U odeljku 1.5.4 izneli smo osnovne podatke o mreži 802.11. Sada ćemo se malo više pozabaviti njenom tehnologijom. U narednim odeljcima pretrešćemo njen skup protokola, tehnike radio prenosa u fizičkom sloju, protokol MAC podsloja, strukturu okvira i usluge. Detaljnije informacije o mreži 802.11 možete naći kod Crowa i sa- radnika (1997), Geiera (2002), Heegarda i saradnika (2001), Kappa (2002), O'Hare i Petricka (1999) i Severancea (1999). Ako želite da se obavestite na samom izvoru, pogledajte standard 802.11.

4.4.1 Skup protokola mreže **802.11**

Protokoli koji se koriste u svim mrežama tipa 802, uključujući i Ethernet, strukturno su slični. Delimičan pregled skupa protokola 802.11 prikazanje na slici 4-25. Fizički sloj prilično dobro odgovara fizičkom sloju prema modelu OSI, ali sloj veze podataka u svim protokolima 802 razbijen je na dva ili više podslojeva. U mreži
802.11, MAC podsloj upravlja načinom dodele kanala, odnosno određuje redosled emitovanja. Iznad njega je LLC podsloj koji miri različite varijante mreže 802 i mrežnom sloju uvele prosleđuje podatke istog formata. Taj podsloj smo obradili u vezi saEthernetom, pa je izlišno da o njemu dalje govorimo.



Slika 4-25. Delimičan skup protokola mreže 802.**11**.

Godine 1997, standardom 802.11 propisane su tri tehnike prenosa u fizičkom sloju. Prenos infracrvenim zračenjem uglavnom se zasniva na tehnologiji koja se već koristi za daljinsko upravljanje TV prijemnikom. Za druge dve tehnike (FHSS i DSSS) koriste se radio-talasi kratkog dometa i to u području za koje nije potrebno tražiti odobrenje (ISM područje na 2,4 GHz). Uređaji za daljinsko otvaranje vrata takođe rade u ovom području, tako da vaš prenosivi računar može da se sukobi s vratima vaše garaže. Ovo područje koriste i fiksni bežični telefoni, kao i mikrotalasne rerne. Brzina prenosa je u svim slučajevima 1 ili 2 Mb/s, a domet je dovoljno mali da se uređaji uglavnom međusobno ne ometaju. Godine 1999. uvedene su dve nove tehnike da bi se povećao propusni opseg: uz tehniku OFDM on može da bude i 54 Mb/s, a uz HR- -DSSS do 11 Mb/s. Godine 2001. uvedena je i druga OFDM modulacija, u drugom frekventnom području. Sada ćemo ih sve ukratko opisati. Njihov opis bi formalno trebalo da se nađe u 2. poglavlju, ali pošto su ove tehnike u tesnoj vezi sa svim lokalnim mrežama, kao i sa MAC podslojem, smestili smo ga ovde.

4.4.2 Fizički sloj mreže 802.11

Pomoću svake od pet dozvoljenih metoda prenosa može se poslati MAC okvir od jedne stanice drugoj. Tehnike se, međutim, između sebe razlikuju po tehnologiji i brzini prenosa. Detaljno opisivanje tehnologije prevazilazi okvire ove knjige, ali neće smetati da o svakoj kažemo nekoliko reči i upoznamo vas s terminologijom kako bi čitaoci koji žele više informacija lakše mogli da ih pronađu na Internetu i u drugim izvorima.

Za prenos u infracrvenom području koristi se difuzan (dakle, neusmeren) snop zrakova talasne dužine 0,85 ili 0,95 pm. Propisane su dve brzine prenosa: 1 Mb/s i 2 Mb/s. Pri brzini 1 Mb/s, koristi se takav sistem kodiranja (Grejev kod, engl. *Gray code)* kod koga se grupe od po 4 bita pretvaraju u 16-bitne kodne reči sastavljene od 15 nula i jedne jedinice. Takvim kodiranjem se postiže da malo odstupanje od sinhroni- zacije proizvodi samo jednobitnu grešku. Pri brzini 2 Mb/s, kodiranju se grupe od po

2 bita u 4-bitne kodne reči, koje talcođe sadrže samo po jednu jedinicu, naime: 0001, 0010, 0100 ili 1000. Infracrveno zračenje ne prolazi kroz zidove, tako da su ćelije koje se nalaze u zasebnim prostorijama dobro izolovane jedne od drugih. Pa ipak, zbog malog propusnog opsega (i činjenice da sunčeva svetlost stvara smetnje pri komunikaciji pomoću infracrvenih zrakova), ovaj način prenosa nije stekao veliku popularnost.

Kod **skokovitog frekventnog širenja spektra** (engl. *Frequency Hopping Spread Spectrum, FHSS)* koristi se 79 kanala širine 1 MHz, koji se redaju od donje granice ISM područja na 2,4 GHz. Redosledom skakanja s jedne na drugu frekvenciju upravlja generator pseudoslučajnih brojeva. Dokle god sve stanice koriste istu klicu pseudoslučajnog niza brojeva koje proizvodi generator i dokle god su vremenski sin- hronizovane, one će istovremeno prelaziti s jedne frekvencije na drugu. **Vreme boravka** (engl. *dwell time*) na pojedinačnim frekvencijama može se podešavati, ali mora biti krađe od 400 ms. Slučajan način biranja frekvencija koji se koristi pri pri- meni ove tehnike omogućava ravnomerno (ili ravnopravno) korišcenje spektra u ovom inače neregulisanom ISM području. On u izvesnoj meri i obezbeđuje vezu jer niko ne može da prisluškuje prenos ako ne zna redosled menjanja frekvencija i vreme boravka. Na dužim rastojanjima može doći do slabljenja signala zbog različitih putanja, ali FHSS ima načina da se s tim izbori. Tehnika FHSS takođe je srazmerno neosetljiva na radio smetnje, zbog čega je omiljena za uspostavljanje veze između zgrada. Glavna mana joj je mali propusni opseg.

Treća tehnika modulacije, direktno **sekvencijalno širenje** spektra (engl. *Direct Sequence Spread Spectrum, DSSS),* talcođe je ograničena na brzine 1 ili 2 Mb/s. Sistem koji se koristi ima izvesnih sličnosti sa sistemom CDMA o kome smo govorili u ođeljku 2.6.2, ali mu nije identičan. Svaki bit se prenosi uli podintervala kao **Barkerova sekvenca** (engl. *Barker sequence)*. Koristi se modulacija faznim pomakom pri 1 Mbodu i prenosi 1 bit po bodu kada se radi brzinom 1 Mb/s, a 2 bita po bodu pri brzini 2 Mb/s. FCC je godinama zahtevala da svi bežični uređaji u SAD koji rade u ISM području koriste širenje spektra, ali je maja 2002. to pravilo napušteno jer su se pojavile nove tehnologije.

Prva od lokalnih bežičnih mreža velike brzine, **802.11a,** koristi multipleksiranje **sa ortogonalnom podelom frekvencija** *(Orthogonal Frequency Division Multiplexing, OFDM)* za prenos brzinama do 54 Mb/s u širem ISM području na 5 GHz. Kao što i naziv tehnike ukazuje, koriste se 52 različite frekvencije - njih 48 služi za prenošenje podataka, a 4 za sinhronizovanje, što podseca na ADSL. Pošto se prenos podataka istovremeno odvija na više frekvencija, tehnika se svrstava u metode širenja spektra, ali se razlikuje od tehnika CDMA i FHSS. Cepanje jedinstvenog područja na više uskih područja za prenos signala, donosi izvesne suštinske prednosti, uključujući veću otpornost na smetnje i mogućnost korišćenja nesusednih uskih područja. Kodiranje je složeno i zasnovano na modulaciji faznim pomakom do brzina 18 Mb/s i na sistemu QAM iznad toga. Pri brzini prenosa .54 Mb/s, grupe od 216 bitova podataka kodiraju se u 288-bitne simbole. Deo privlačnosti sistema OFDM leži u njegovoj kompatibilnosti sa evropskim sistemom HiperLAN/2 (Doufexi i sar., 2002). Tehnika dobro iskorišćava frekventni spektar (u smislu broja bitova prenesenih po hercu) i otporna je na slabljenje signala zbog različitih putanja.

Sledećom tehnikom, **visokobrzinskim direktnim sekvencijalnim širenjem spektra** (engl. *High Rate Direct Sequence Spread Spectrum, HR-DSSS*), generiše se 11 miliona podintervala u sekundi da bi se u području na 2,4 GHz postigla brzina pre- nosa 11 Mb/s.

Standard je nazvan **802.11b,** ali nije izveden iz standarda 802.11a. U stvari, on je prvi nastao i prvi se pojavio na tržištu. Dozvoljene brzine prenosa podataka po standardu 802.11b iznose: 1,2,5,5 i 11 Mb/s. Prenos dvema manjim brzinama ostvaruje se uz 1 Mbod (sa 1, odnosno sa 2 bita po bodu), a koristi se modulacija faznim pomakom (zbog kompatibilnosti sa sistemom DSSS). Prenos dvema većim brzinama odvija se uz 1,375 Mbodova (4, odnosno 8 bitova po bodu) i kodiranje po **Volš- -Hadamardu** (engl. *Walsh-Hadamard).* Brzina prenosa podataka može se tokom rada dinamički podešavati da bi se postigao optimum u uslovima aktuelnog saobraćaja i šuma. Radna brzina prenosa sistema 802.11 b u praksi je skoro uvek 11 Mb/s. Iako je mreža 802.11b sporija od mreže 802.11a, njen domet je oko 7 puta veći, što je u mnogim situacijama odlučujuće.

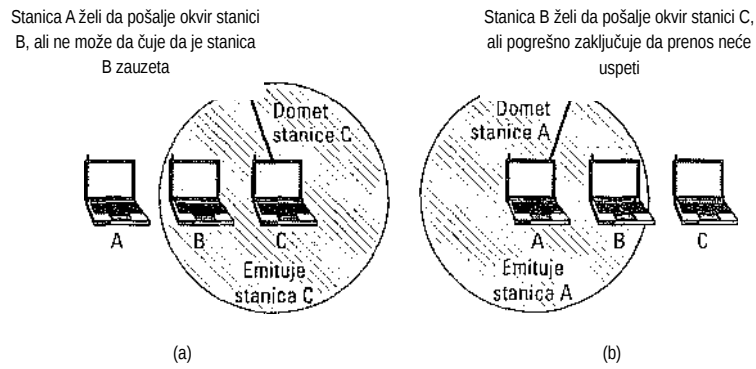
IEEE je posle mnogo politiziranja o tome čiju će patentiranu tehnologiju upotre- biti, novembra 2001. prihvatio poboljšanu verziju standarda 802.11b, mrežu **802.11g.** U njoj se koristi OFDM modulacija iz standarda 802.11a, ali ona radi u uskom ISM području na 2,4 GHz uporedo s mrežom 802.1 lb. Brzine koje se teorijski u njoj mogu postići dosežu 54 Mb/s. Još uvek nije jasno da li se one mogu ostvariti i u praksi. Ipak, jasno je daje komitet za mrežu 802.11 definisao tri različite visokobrzinske bežične lokalne mreže: 802.11a, 802.11b i 802.1 lg (da i ne pominjemo tri sporije bežične mreže). Možemo se s pravom upitati da li tako treba da radi komitet za standardizo- vanje. Možda je tri njihov srećan broj.

4.4.3 Protokol MAC podsloja mreže 802.11

Vratimo se sada sa elektrotehnike ponovo računarskim naukama. Protokoli MAC podsloja za mrežu 802.11 i Ethernet potpuno su različiti jer je bežični sistem po prirodi mnogo složeniji od kablovskog. Na Ethernetu stanica samo čeka da saobraćaj utihne, a onda počinje da emituje. Ako ne primi povratni rafalni šum tokom prva 64 bajta prenosa, to znači daje okvir gotovo sigurno ispravno isporučen. U bežičnom sistemu takav scenario je neostvarljiv.

Počnimo od problema skrivene stanice koji smo ranije opisali, a sada ponovo ilu- strovali slikom 4-26(a). Pošto nisu sve stanice jedna drugoj u dometu, prenos koji se odvija u jednom delu ćelije možda neće biti opažen u njenom drugom delu. U prikazanom primeru, stanica C šalje podatke stanici *B.* Ako stanica A osluškuje kanal, ona neće ništa čuti i pogrešno će zaključiti da može da pošalje podatke stanici *B.*

Postoji i problem suprotan opisanom, problem izložene stanice, ilustrovan slikom 4-26(b). Ovde stanica *B* želi da pošalje podatke stanici *C* i zato osluškuje kanal. Kada začuje emisiju, pogrešno zaključuje da ne srne da šalje podatke stanici C, dok u stvari to A možda šalje podatke stanici *D* (koja nije prikazana). Osim toga, stanice većinom rade u poludupleksnom režimu, što znači da ne mogu istovremeno na istoj frekvenciji da emituju i da osluškuju rafalni šum. Zbog svega toga se u mreži 802.11 ne koristi protokol CSMA/CD, kao kod Etherneta.



Slika 4-26. (a) Problem skrivene stanice, (b) Problem izložene stanice.

U cilju razrešavanja opisanih problema, u mreži 802.11 podržana su dva režima rada. U prvom, gde se koristi distribuirana koordinativna funkcija (engl. *Distributed Coordination Function*, *DCF),* nema centralizovanog upravljanja (što je slično Ethernetu). U drugom, gde se koristi jedinstvena koordinativna funkcija (engl. *Point Coordination Function, PCF),* bazna stanica upravlja svime što se događa u ćeliji. Sve realizacije moraju da obuhvate DCF, dok je režim PCF neobavezan. Ispitajmo ih, jedan za drugim.

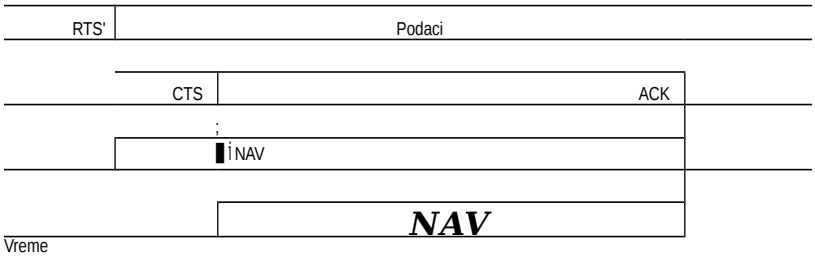
Kada se primeni DCF režim, mreža 802.11 radi uz protokol CSMA/CA (CSMA uz izbegavanje sukoba, engl. *CSMA with Collision Avoidance).* Prema tom protokolu, osluškuje se i fizički i virtuelni kanal. Protokol CSMA/CA podržava dva načina rada. Po prvom, kada stanica želi da emituje, ona najpre osluškuje kanal. Ukoliko na njemu nema saobraćaja, stanica počinje da emituje. Ona tokom emitovanja ne osluškuje kanal, već emituje čitav okvir koji može da propadne zbog sukobljavanja kod primaoca. Ako je kanal zauzet, pošiljalac se povlači dok se kanal ne oslobodi, a zatim počinje da emituje. Ako dođe do sukobljavanja, sukobljene strane posle vremenskog intervala izabranog algoritmom binarnog eksponencijalnog odustajanja (Ethernet), ponovo pokušavaju da emituju.

Drugi način rada zasniva se na protokolu MACAW uz osluškivanje virtuelnog kanala (slika 4-27). U prikazanom primeru, stanica A želi da pošalje okvir stanici *B.* Stanica C je u dometu stanice A (verovatno i u dometu stanice *B,* ali to nije bitno). Stanica *D* je u dometu stanice *B,* ali ne i u dometu stanice A.

Protokol počinje time što stanica A poželi da pošalje podatke stanici *B.* Ona svoju želju saopštava tako što stanici *B* šalje RTS okvir tražeći dozvolu da joj pošalje podatke. Kada stanica *B* primi zahtev, ona može da ga prihvati, u kom slučaju odgovara stanici A okvirom CTS. Po prijemu CTS okvira, stanica A šalje okvir s podacima i aktivira ACK tajmer. Kada primi okvir s podacima u ispravnom stanju, stanica *B* odgovara ACK okvirom, čime se razmena završava. Ako se ACK tajmer automatski isključi pre nego što stigne potvrda o prijemu (ACK), ceo protokol se ponavlja.

Pogledajmo kako opisanu razmenu vide stanice *C* i *D.* Stanica Cje u dometu stanice A, pa može da primi njen RTS okvir. Ako ga primi, zna da će neko ubrzo slati i podatke, pa se zbog opšteg dobra uzdržava od emitovanja sve dok se ta razmena ne završi. Na osnovu informacija iz RTS zahteva, ona može da odredi trajanje razmene, uključujući i vreine potrebno za stizanje potvrde o prijemu, tako da sebi stavlja zabranu korišćenja virtuelnog kanala pomoću

vektora za **dodelu** mreže (engl. *Network Allocation Vector, NAV),* prikazanog na slici 4-27. Stanica *D* ne čuje okvir RTS, ali čuje okvir CTS, pa i ona postavlja svoj NAV. Imajte na umu da se signali NAV ne emituju - to su samo interni podsetnici da stanica treba da se tokom izvesnog vremena utiša.

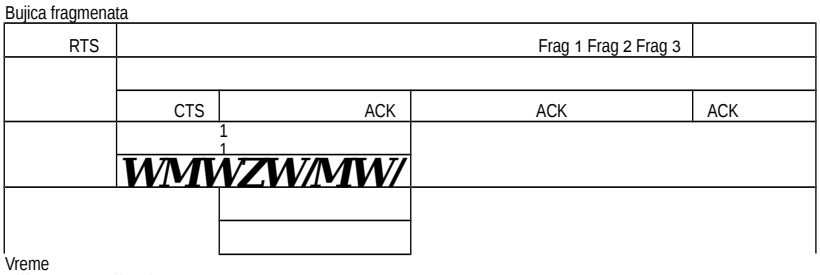


Slika 4-27. Osluškivanje virtuelnog kanala u protokolu **CSMA/CA.**

Za razliku od kablovskih mreža, bežične mreže su bučne i nepouzdane, što je u znatnoj meri uzrokovano radom mikrotalasnih pećnica koje koriste isto neregulisano ISM područje. Zbog toga se s dužinom okvira smanjuje verovatnoća da on na odredište stigne neoštećen. Ako je *p* verovatnoća da bilo koji pojedinačan bit bude pogrešan, onda verovatnoća isporuke »-bitnog okvira u potpuno ispravnom stanju iznosi (1 - *p)*^{n}. Na primer, kada je $p = IO^{'4}$, verovatnoća primanja Ethernet okvira (12.144 bitova) u potpuno ispravnom stanju iznosi manje od 30%. Ako je $p = 10^{"5}$, od 9 okvira jedan će biti oštećen. Čak i kada je $p = 10^{"6}$, biće oštećeno više od 1% okvira, tj. oko desetu sekundi, a još i više ako se ne koriste okviri maksimalne dužine. Sve u svemu, ako je okvir predugačak, male su šanse da stigne neoštećen i verovatno mora da bude ponovo poslat.

Da bi se izišlo na kraj s bučnim kanalima, u mreži 802.11 dozvoljava se deljenje okvira u manje fragmente, svaki sa svojim kontrolnim zbirom. Fragmenti se pojedinačno obeležavaju i potvrđuju pomoću protokola „stani i čekaj“ (pošiljalac ne sme da pošalje fragment *k* + 1, sve dok ne dobije potvrdu o prijemu fragmenta *k).* Kada se posle razmene okvira RTS i CTS kanal konačno rezerviše za prenos, može se slati više fragmenata u nizu (slika 4-28); to je bujica fragmenata (engl *.fragment burst).*

Fragmentacijom se povećava protok podataka jer se ponovo šalju samo oštećeni fragmenti, a ne celi okviri. Veličina fragmenta nije propisana standardom; to je parametar svake pojedinačne ćelije koji određuje odgovarajuća bazna stanica. Mehanizam NAV signala uspeva da ućutka stanice do sledeće potvrde o prijemu, ali se za neometano slanje bujice fragmenata koristi drugi mehanizam (koji opisujemo u nastavku).



Slika 4-28. Bujica fragmenata.

Sve što smo naveli važi za DCF režim mreže 802.11. U njemu nema centralizova- nog upravljanja i stanice se nadmeću za kanal, kao na Ethernetu. Drugi režim je PCF, u kome bazna stanica poziva računare iz ćelije da šalju okvire. Pošto ovde redosle- dom emitovanja u potpunosti upravlja bazna stanica, ne postoji mogućnost sukobljavanja. Standard propisuje mehanizam pozivanja, ali ne i učestalost, odnosno redosled pozivanja, pa ni to da sve stanice moraju da budu uslužene na isti način.

U načelu, bazna stanica 10 do 100 puta u sekundi difuzno emituje signalni okvir (engl. *beacon frame).* Taj okvir sadrži sistemske parametre, kao što su redosled skakanja s jedne na drugu frekvenciju i vreme boravka za FHSS, parametre za sinhroni- zaciju itd. On sadrži i poziv novim stanicama da se prijave za uslugu pozivanja. Kada se stanica prijavi za pozivanje određenom učestalošću, time joj se garantuje odgovarajući deo propusnog opsega, što njoj omogućava da garantuje kvalitet slanja okvira.

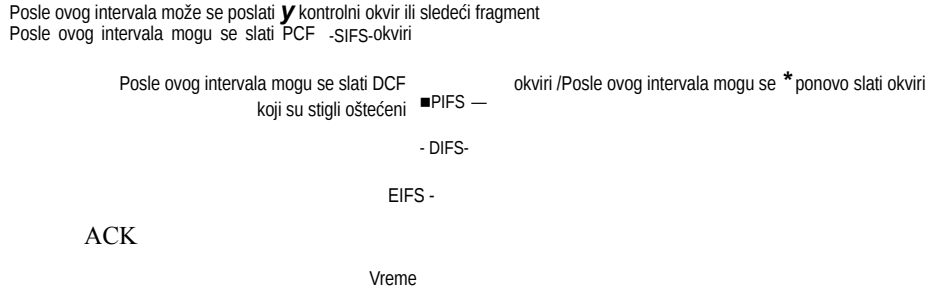
Vek baterija uvek predstavlja problem kod pokretnih bežičnih uređaja, tako da se u mreži 802.11 posvećuje pažnja napajanju. Konkretno, bazna stanica može da „uspava“ pokretnu stanicu sve dok joj sama ne uputi izričit poziv ili je ne probudi korisnik. Time.što može da uspava pokretnu stanicu, bazna stanica preuzima odgovornost da sačuva svaki okvir koji tokom mirovanja pokretne stanice stigne na njenu adresu. Okvire će joj isporučiti kada se probudi.

Mehanizmi PCF i DCF mogu da postoje u istoj ćeliji. Teško je zamisliti u prvi mah da istovremeno može da postoji centralizovano i distribuirano upravljanje, ali se u mreži 802.11 to postiže tačnim definisanjem vremenskog intervala između uzastopnih okvira. Pošto neko pošalje okvir, mora da prođe određeno „mrtvo“ vreme pre nego što bilo ko pošalje sledeći. Definišu se četiri različita intervala, svaki za određenu svrhu (slika 4-29).

Najkraći interval nosi odgovarajuće ime: **kratak razmak između okvira** (engl. *Short InterFrame Spacing, SIFS).* On omogućava dijalog između stanica, tj. slanje CTS odgovora na RTS okvir, slanje potvrde o prijemu fragmenta ili celog okvira i slanje bujice fragmenata bez potrebe da se ponovo šalje RTS zahtev.

Uvek postoji samo jedna stanica koja je ovlašćena da odgovori posle intervala SIFS. Ako ona propusti tu šansu, a prođe i PCF **razmak između okvira** (engl. *PCF InterFrame Spacing, P1FS),* bazna stanica može da pošalje signalni okvir ili okvir s

pozivom za slanje. Taj mehanizam omogućava stanici koja šalje okvir ili niz fragmenata s podacima da neometano dovrši slanje, ali dozvoljava i baznoj stanici da ugrabi kanal kada primeti daje pošiljalac završio posao i da se ne takmiči za pristup s dragim spremnim korisnicima.



Slika 4-29. Razmaci između uzastopnih okvira u mreži 802,11.

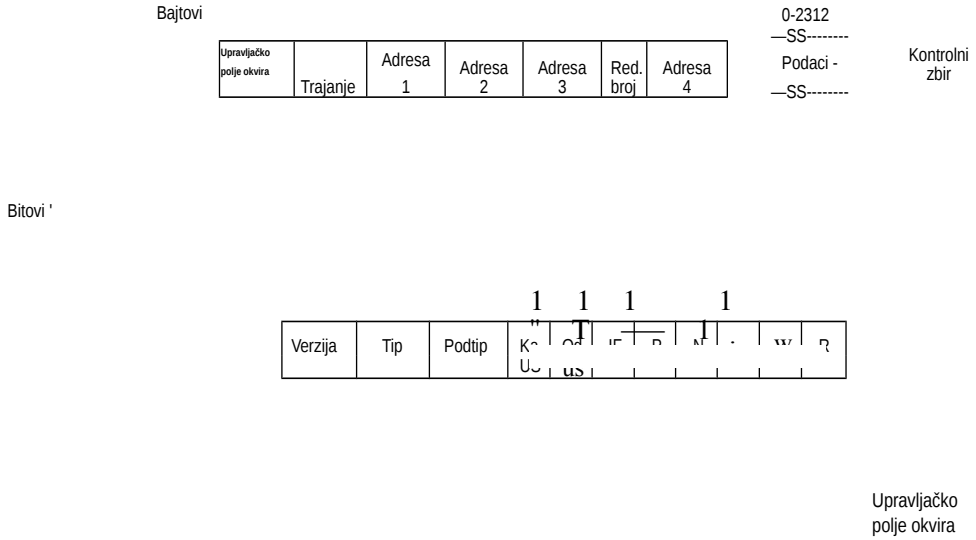
Ako bazna stanica nema šta da einituje, a prođe DCF **razmak između okvira** (engl. *DCFInterFrame Spacing, DIFS),* svaka stanica inože pokušati da zauzme kanal da bi poslala nov okvir. Tu se primenjuju uobičajena takmičarska pravila, a za slučaj sukobljavanja može se predvideti algoritam binarnog eksponencijalnog odustajanja.

Poslednji, **produženi razmak između okvira** (engl. *Extended InterFrame Spacing, EIFS),* koristi za izveštavanje samo ona stanica koja je primila oštećen ili nepoznat okvir. Tom događaju je dat najniži prioritet jer primalac koji ne zna o čemu se radi treba strpljivo da sačeka završetak tekućeg dijaloga između dve stanice.

4.4.4 Struktura okvira u mreži 802.11

Standard 802.11 definiše tri klase okvira za bežični prenos: okvire za podatke, kontrolne okvire i upravljačke okvire. Zaglavlje okvira svake klase sadrži više polja koja se koriste u MAC podsloju. Osim toga, postoje i zaglavlja koja se koriste u fizičkom sloju, ali ona se uglavnom tiču načina modulacije, pa ih ovde nećemo razmatrati.

Format okvira s podacima prikazan je na slici 4-30. Najpre dolazi *Upravljačko polje okvira,* koje sadrži 11 potpolja. Prvo od njih je *Verzija* protokola i ono omogućava da se u istoj ćeliji istovremeno izvršavaju dve verzije protokola. Zatim dolaze potpolja *Tip* (podaci, kontrola ili upravljanje) i *Podtip* (na primer, RTS ili CTS). Bitovima *Ka DS* i *Od DS* naznačava se da li se okvir upućuje *ka* međućelijskom dis- tribucionom sistemu (npr. Ethernetu) ili *od* njega. Bitom *JF* naznačuje se da sledi još fragmenata. Bit *P* označava ponovno slanje ranijeg okvira. Bit A^apajanje) koristi bazna stanica da bi uspavala, odnosno probudila primaoca. Bitom *JO* naznačuje se da pošiljalac ima još okvira za primaoca. Bit *W* znači daje telo okvira šifrovano algoritmom **WEP (privatnost kao u kablovskoj mreži,** engl. *Wired Equivalent Privacy).* Na kraju, bit *R* saopštava primaocu da sekvencu okvira u kojima ovaj bit ima vrednost 1 mora da obradi držeći se redosleda.



Slika 4-30. Okvir podataka u mreži 802.11

Drugo polje, *Trajanje,* ukazuje na to koliko će okvir i potvrda za njega zauzimati kanal. To polje postoji i u kontrolnim okvirima; na osnovu njega druge stanice određuju dužinu svog NAV intervala. U zaglavlju okvira su četiri adrese, sve u standardnom formatu IEEE 802. Jasno je da su tu adrese pošiljaoca i primaoca, ali šta predstavljaju druge dve adrese? Setite se da okvir može da uđe u ćeliju i da iz nje izađe preko bazne stanice. Druge dve adrese koriste bazne stanice izvorišta i odredišta za saobraćaj između ćelija.

Polje *Redni broj* omogućava numerisanje fragmenata. Od 16 raspoloživih bitova, njih 12 identifikuju okvire, a 4 bita označavaju fragmente. Polje *Podaci* sadrži do 2312 bajtova korisničkih podataka, a zatim sledi uobičajeni *Kontrolni zbir.*

Upravljački okviri imaju format sličan opisanom, nedostaje im samo adresa jedne bazne stanice jer je njihovo korišćenje ograničeno na ćeliju. Još kraći su kontrolni okviri: oni imaju samo jednu do dve adrese, nemaju *Podatke* i *Redni broj.* Najvažniji podaci se nalaze u potpolju *Podtip* (obično RTS, CTS ili ACK).

4.4.5 Usluge

Prema standardu 802.11 svaka bežična lokalna mreža mora da obezbedi devet usluga koje su svrstane u dve kategorije: pet distribucionih i četiri za potrebe radnih stanica. Distribucione usluge upravljaju pripadnošću stanica ćelijama i komunikacijom između stanica u različitim ćelijama. One druge bave se isključivo aktivnostima unutar ćelije.

Pomoću pet distribucionih usluga bazne stanice prate svoje pokretne srodnike dok prelaze iz ćelije u ćeliju i „predaju“ ih jedna dragoj. To su sledeće usluge:

1. Usluga povezivanja (Association). Pomoću ove usluge pokretne stanice se povezuju s baznim stanicama. Ona se obično aktivira kada pokretna stanica dođe u domet bazne

stanice. Stanica se tada predstavlja i otkriva svoje mogućnosti, saopštavajući brzine prenosa za koje je sposobna, potrebu za PCF uslugama (tj. za pozivanjem) i zahteve za upravljanje napajanjem. Bazna stanica može da prihvati ili da odbije pokretnu stanicu. Ako pokretna stanica bude prihvaćena, moraće da potvrdi svoj identitet.

2. **Usluga razvezivanja** (Disassociation). Vezu može da raslcine ili bazna ili pokretna stanica. Pokretna stanica treba da iskoristi ovu uslugu pre nego što se isključi ili napusti ćeliju, ali je može aktivirati i bazna stanica pre nego što se povuče radi održavanja sistema.
3. **Usluga ponovnog povezivanja** (Reassociation). Pokretna stanica pomoću ove usluge može da promeni svoju podrazumevanu baznu stanicu. Usluga je potrebna pokretnim stanicama koje menjaju ćelije. Ako se izvrši ispravno, tokom preuzimanja upravljanja neće doći do gubljenja podataka. (Međutim, mreža 802.11, kao i Ethernet, radi samo najbolje što može, a ne bez greške.)
4. **Usluga distribuiranja** (Distribution). Okviri koje prima bazna stanica usmeravaju se ovom uslugom. Ako je odredište u okviru iste ćelije, okviri se mogu poslati bežičnim putem. U suprotnom, moraju se uputiti kablom.
5. **Usluga integrisanja** (Integration). Ako okvir treba poslati kroz neku drugačiju mrežu koja podržava drugi format adrese ili okvira, onda ova usluga prevodi format mreže 802.11 u format odredišne mreže.

Ostale četiri usluge tiču se aktivnosti unutar ćelije. One se mogu koristiti posle povezivanja.

1. **Usluga provere identiteta** (Authentication). Pošto neovlašćene stanice bežičnim putem mogu lako slati i primati poruke, stanica mora da se podvrgne proveri identiteta pre nego što dobije dozvolu da šalje podatke. Nakon što se pokretna stanica pridruži odgovarajućoj baznoj stanici (bude prihvaćena u njenu ćeliju), bazna stanica joj šalje specijalan probni okvir da bi pro- verila da li pokretna stanica zna tajni ključ (lozinku) koji joj je dodeljen. Pokretna stanica to dokazuje tako što ključem šifruje probni okvir i vraća ga baznoj stanici. Ako to uradi ispravno, stanica se u potpunosti prihvata u ćeliju. U standardu kakav je sada, bazna stanica ne mora da se identifikuje pokretnoj stanici, ali je ispravka ovog propusta u toku.
2. **Brisanje identiteta** (Deauthentication). Kada stanica koja se prethodno uspešno identifikovala želi da napusti ćeliju, bazna stanica briše njen identitet iz svog internog spiska. Posle toga, pokretna stanica ne može više da koristi mrežu dok se ponovo ne identifikuje i ne dobije odobrenje za pristup.
3. **Privatnost** (Privacy). Da bi se održala privatnost podataka koji se šalju bežičnom lokalnom mrežom, oni se moraju šifrovati. Uslugom privatnosti obez- beđuje se njihovo šifrovanje i dešifrovanje. Šifrovanje se obavlja algoritmom RC4 koji je napisao Ronald Rivest s Masačusetskog tehničkog instituta.
4. **Isporuka podataka** (Data delivery). Na kraju krajeva, sve se vrti oko isporuke podataka, pa je prirodno da standard 802.11 obezbeđuje i način za slanje i primanje podataka. Postoje mreža 802.11 modelovana prema Ethernetu, a Ethernet ne garantuje sasvim pouzdano slanje podataka, ni isporuka podataka mrežom 802.11 nije 100% pouzdana. Viši slojevi treba da vode računa o otkrivanju i ispravljanju grešaka.

Ćelija mreže 802.11 ima nekoliko parametara koji se mogu proveriti i - u nekim slučajevima - podesiti. Oni se odnose na šifrovanje, rad tajmera, brzinu prenosa podataka, učestalost signaliziranja itd.

Bežične lokalne mreže po standardu 802.11 počele su da se postavljaju po poslovnim zgradama, aerodromima, hotelima, restoranima i univerzitetima širom sveta. Očekuje se njihovo brzo širenje. Neka iskustva sa ovom mrežom stečena na Univerzitetu Karnegi-Melon (Pitsburg, Pensilvanija) možete naći kod Hillsa (2001).

4.5 ŠIROKOPOJASNI BEŽIČNI PRENOS

Previše smo se zadržali „unutra". Zato izađimo i pogledajmo šta se događa napolju. Ispostavlja se da tamo sve vri, a nešto od toga odnosi se i na tzv. poslednji kilometar. Uporedo s prelaskom državnih telefonskih kompanija u privatne ruke, novim vlasnicima je u mnogim zemljama dozvoljeno da nude lokalne usluge prenosa glasa i brzog povezivanja na Internet. Izvesno je da se takve usluge mnogo traže. Problem je, međutim, u tome što je razvlačenje optičkog ili koaksijalnog kabla do miliona domova i poslovnih prostorija, pa i upredene parice 5. kategorije, izuzetno skupo. Šta u takvoj situaciji investitor treba da radi?

Pravi odgovor je širokopojasna bežična veza. Postavljanje velike antene na brdu izvan grada i instaliranje korisničkih antena po krovovima mnogo je lakše i jeftinije od kopanja kanala i razvlačenja kablova. Na taj način, konkurentske telefonske kompanije nalaze veliki interes u obezbeđivanju visokobrzinske usluge bežičnog prenosa glasa, Interneta, videa na zahtev itd. Kao što smo videli na slici 2-30, usluga LMDS stvorena je upravo za ove svrhe. Međutim, donedavno je svaki vlasnik telefonske kompanije imao svoj sopstveni sistem. Takav nedostatak standarda znači da se potreban hardver i softver nisu mogli masovno proizvoditi, zbog čega su cene bile visoke, a zainteresovanost za uslugu mala.

Mnogi privrednici su shvatili da standard za širokopojasni bežični prenos predstavlja onu ključnu „nedostajuću kariku", pa je IEEE zamoljen da oformi komitet od predstavnika glavnih kompanija i univerziteta, i da standardizuje ovu oblast. Naredni raspoloživ broj u sistemu mreža 802 bio je **802.16,** tako da je standard dobio tu oznaku. Posao je započet jula 1999, a standard je usvojen aprila 2002. Standard se zvanič- no zove „Interfejs za fiksne pristupne širokopojasne bežične sisteme" (engl. Air Interface for Fixed Broadband Wireless Access Systems). Mnogi ga, međutim, zovu **bežična gradska mreža** (engl. *wireless MAN)* ili **bežična lokalna linija** (engl. *wireless local loop),* Koristićemo sve navedene nazive u istom značenju.

Slično drugim standardima serije 802, i mreža 802.16 je pod velikim uticajem OSI modela, što obuhvata (pod)slojeve, terminologiju, osnovne uslužne operacije i drugo. Nažalost, kao i OSI model, i ona je prilično složena. U narednim odeljcima dademo kratak prikaz glavnih karakteristika mreže 802.16, izvesno nepotpun i bez mnogih detalja. Dopunska obaveštenja o širokopojasnom bežičnom prenosu uopšte, potražite kod Bolcskeia i saradnika (2001) i Webba (2001), a direktno o mreži 802.16 kod Eklunda i saradnika (2002).

4.5.1 Poređenje mreža 802.11 i 802.16

Možda se sada pitate čemu nov standard. Zašto i nadalje ne bismo koristili mrežu 802.11? Za to ima dobrih razloga, a prvi je to što se mrežama 802.11 i 802.16 rešavaju različiti problemi. Pre nego što pređemo na samu tehnologiju mreže 802.16, možda treba da objasnimo zašto je uopšte bilo potrebno stvarati nov standard.

Okruženja u kojima rade mreže 802.11 i 802.16 na neki način su slična, naročito zato što

obe mreže treba da obezbede bežičnu komunikaciju velikog propusnog opsega. One se, međutim, i razlikuju s više aspekata. Najpre, mreža 802.16 opslužuje zgrade, a zgrade (najčešc'e) nisu pokretne, tj. ne sele se često iz jedne ćelije u drugu. Veći deo mreže 802.11, s druge strane, usmeren je upravo na polcretnost korisnika i taj deo nema ničeg zajedničkog s mrežom 802.16. Zatim, u zgradama može da bude više računara, što je situacija potpuno drugačija od one kada krajnji korisnik ima samo jedan prenosivi računar.

Pošto su vlasnici zgrada u odnosu na vlasnike prenosivih računara voljni da potroše više novca za hardver, radio uređaji su im kvalitetniji. To znači da se u mreži 802.16 može koristiti potpun dupleksni režim - nešto što se u mreži 802.11 izbegava zbog povećanih troškova.

Pošto se mreža 802.16 prostire preko dela gradskog područja, razdaljine se mere kilometrima, pa snaga signala koju od računara prima bazna stanica može široko da varira. To variranje utiče na odnos signala i šuma, pa je neophodno koristiti više tehnika modulacije signala. Isto tako, otvoreno komuniciranje gradskim područjem po- drazumeva obavezu zaštite privatnosti poruka.

Osim toga, očekuje se da u svakoj ćeliji mreže 802.16 bude mnogo više korisnika i da ti korisnici zauzmu mnogo veći propusni opseg, nego u tipičnoj ćeliji mreže 802.11. Retko ćete naći kompaniju koja će u istoj prostoriji okupiti 50 saradnika s pre- nosivim računarima samo da bi utvrdila da će tih 50 saradnika zasititi mrežu 802.11 gledajući istovremeno 50 različitih filmova na svojim računarima. Zbog takvih situacija, potreban je veći propusni opseg nego što može da obezbedi ISM područje, pa je mreža 802.16 prisiljena da radi na mnogo višim frekvencijama: između 10 i 66 GHz - u jedinoj oblasti koja još uvek nije zauzeta.

Međutim, ti milimetarski talasi imaju drugačija svojstva od dužih talasa iz ISM područja, zbog čega je fizički sloj potrebno projektovati sasvim drugačije. Milimetarske talase snažno apsorbuje voda (naročito kišne kapi, ali u izvesnoj meri i sneg, grad, čak i gusta magla). Zbog toga je ispravljanje grešaka u prenosu ovde mnogo važnije nego u prenosu kroz enterijer. Milimetarski talasi mogu se usmeriti u snop (mreža 802.11 zrači difuzno), tako da razmišljanja vezana za različite putanje signala u mreži 802.11 ovde nemaju praktičnog značaja.

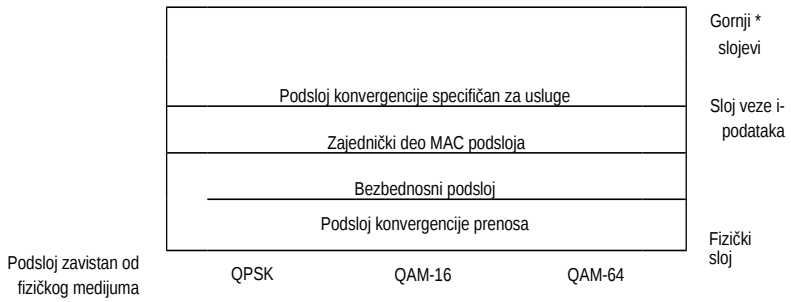
Razlika postoji i u kvalitetu usluga. Iako mreža 802.11 u izvesnoj meri podržava saobraćaj u realnom vremenu (u režimu PCF), ona u stvari nije projektovana za tele- foniju i veliki promet multimedijskog sadržaja. Nasuprot tome, od mreže 802.16 očekuje se da potpuno podrži ove primene jer je namenjena kako privatnim, tako i poslovnim korisnicima.

Ukratko, mreža 802.11 predstavlja Ethernet namenjen pokretnim korisnicma, dok mreža 802.16 treba da bude bežična, ali stacionarna kablovska televizija. Pomenute razlike su tolike, da se i standardi za ove dve mreže veoma razlikuju jer pokušavaju da optimizuju različite stvari.

Vredi izneti i sasvim kratko poređenje sa sistemom mobilne telefonije. U njoj pokretne stanice male snage za prenos govora koriste uzak opseg mikrotalasa srednje talasne dužine. Niko na GSM mobilnom telefonu (još uvele) ne gleda dvočasovni film u visokoj rezoluciji. Čak ni sistem UMTS nema šanse da to izmeni. Ukratko, gradske bežične mreže postavljaju mnogo više zahteva od sistema mobilne telefonije, pa su za njih neophodni sasvim drugačiji sistemi. Zanimljivo pitanje je da li ce se mreža
802.16 u budućnosti koristiti i za pokretne uređaje. Ona za njih nije optimizovana, ali takva mogućnost ipak postoji. Trenutno je usmerena na fiksne bežične veze.

4.5.2 Skup protokola mreže 802.16

Na slici 4-31 prikazan je skup protokola mreže 802.16. Njegova opšta struktura odgovara mrežama serije 802, osim što ovde ima više podslojeva. Krajnji donji podsloj upravlja prenosom koji se ostvaruje klasičnim uskopojasnim radio-uređajima uz korišćenje uobičajenih tehnika modulisanja. Iznad fizičkog prenosnog sloja nalazi se podsloj konvergencije, koji od sloja veze podataka skriva različite tehnologije. U stvari, i mreža 802.11 ima sličan mehanizam kome, međutim, u standardu nije dato posebno ime u stilu modela OSI.



Slika 4-31. Skup protokola mreže 802.16.

Iako ih nismo prikazali na slici, u toku je rad na dva nova protokola fizičkog sloja. Standard 802.16a treba da podrži OFDM u području frekvencija između 2 i 11 GHz, a standard 802.16b rad u ISM području na 5 GHz. Oba protokola predstavljaju pokušaj približavanja mreži 802.11.

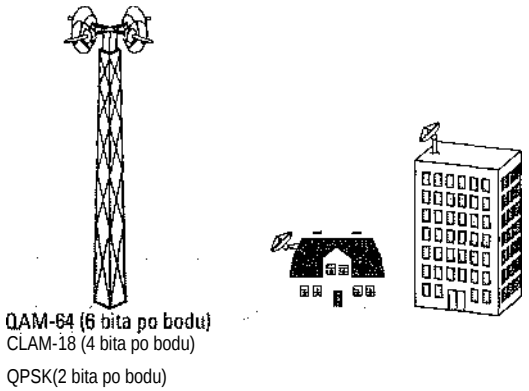
Sloj veze podataka deli se na tri podsloja. Najniži je zadužen za privatnost i bez- bednost, što je mnogo važnije za javni spoljni saobraćaj nego za privatni razgovor u enterijeru. U njemu se razmenjuju šifre i šifruju, odnosno dešifruju poruke.

Srednji podsloj je zajednički deo MAC podsloja. U njemu su smešteni glavni protokoli, npr. protokol za rad s kanalima. Model predviđa da bazna stanica upravlja celim sistemom. Ona može vrlo efikasno da raspoređuje kanale za saobraćaj ka korisnicima, a glavni je „dispečer“ i za kanale kojima joj korisnici upućuju poruke. Za razliku od drugih mreža serije 802, ovaj MAC podsloj je potpuno usmeren na uspostavljanje direktne veze jer mu je glavni cilj da obezbedi kvalitetne telefonske i multimedijske usluge.

Podsloj konvergencije specifičan za usluge zamenjuje podsloj za upravljanje logičkom vezom u drugim 802 protokolima. On treba da obezbedi interfejs ka mrežnom sloju. Situacija je dodatno iskomplikovana time što mreža 802.16 treba da potpuno neprimetno integriše protokole za prenos datagrama (npr. PPP, IP i Ethernet) i ATM. Problem je u tome što protokoli za prenos paketa ne uspostavljaju direktnu vezu, dok je ATM uspostavlja. To znači da svaka ATM veza treba da se preslika u neku vezu mreže 802.16, što je u principu jasno. Ostaje nejasno to u *koju* će se vezu mreže
802.16 preslikati dolazni IP paket, što u stvari rešava ovaj podsloj.

4.5.3 Fizički sloj mreže 802.16

Kao što smo već istakli, za širokopojasni bežični prenos potreban je veliki opseg, a on se jedino može naći u području između 10 i 66 GHz. Milimetarski talasi iz tog područja imaju zanimljivo svojstvo koje nemaju duži talasi: oni se ne prostiru u svim pravcima kao zvuk, već pravolinijski - kao svetlost. Zbog toga bazna stanica mora da ima više antena usmerenih u različite pravce, kao na slici 4-32. Svaki sektor okolnog terena koji pokriva jedna antena ima svoje korisnike koji su srazmerno nezavisni od korisnika u susednim sektorima - nešto što ne postoji u mobilnoj radiotelefoniji gde baza emituje u svim pravcima.

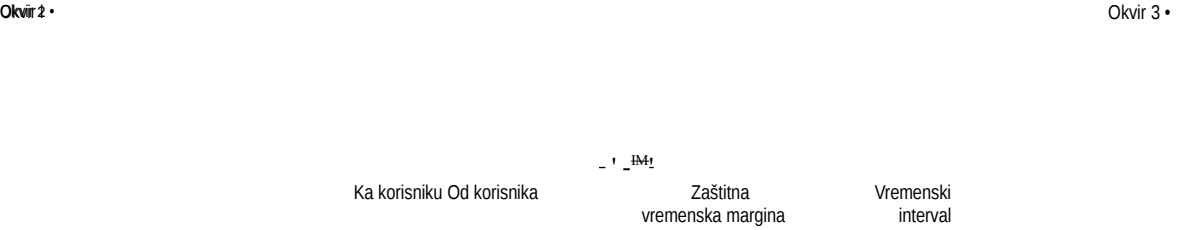


Slika 4-32. Područje mreže 802,16.

Pošto jačina signala u milimetarskom području naglo opada s rastojanjem od bazne stanice, zajedno s njom opada i odnos signala i šuma. Zbog toga se u mreži 802.16 koriste tri tehnike modulacije, zavisno od udaljenosti korisnika od bazne stanice.

U neposrednom okruženju koristi se sistem QAM-64 sa 6 bitova po bodu, na srednjim udaljenostima sistem QAM-16 sa 4 bita po bodu, a modulacija za najudaljenije korisnike je po sistemu QPSK sa 2 bita po bodu. Na primer, ako se iskoristi deo spektra od tipično 25 MHz, uz QAM-64 dobićemo 150 Mb/s, uz QAM-16 100 Mb/s, a uz QPSK - 50 Mb/s. Dragim recima, što je pretplatnik, dalje od bazne stanice, sporiji je prenos podataka (slično sistemu ADSL sa slike 2-27). Konstelacioni dijagrami za tri pomenute tehnike modulacije prikazani su na slici 2-25.

Kada su dobili zadatak da razviju širokopojasni sistem prenosa, projektanti mreže 802.16 morali su se dobro potruditi da uz navedena fizička ograničenja što bolje iskoriste raspoloživi spektar frekvencija. Pri tome nisu bili oduševljeni načinom na koji rade sistemi GSM i DAMPS, tj. korišc'enje dva frekventna područja jednake širine za prenos podataka ka korisniku i od njega. Pri prenosu govora, saobraćaj je verovatno približno jednak u oba smera, ali kada je u pitan ju Internet, saobraćaj ka korisniku najčešće je mnogo veći od saobraćaja ka baznoj stanici. Zbog toga se u mreži 802.16 predviđa mnogo elastičniji način dodeljivanja propusnog opsega pomoću dve tehnike: **dvosmernog prenosa pođelom frekvencije** (engl. *Frequency Division Duplexing, FDD)* i **dvosmernog prenosa podelom vremena** (engl. *Time Division Duplexing, TDD).* Draga od dve tehnike prikazana je na slici 4-33. Tu bazna stanica periodično emituje okvire podeljene u vremenske intervale. Prvi intervali su predviđeni za saobraćaj ka korisniku, zatim dolazi zaštitna margina - vremenski interval u kome stanica menja smer prenosa, a onda dolaze intervali predviđeni za saobraćaj ka baznoj stanici. Broj vremenskih intervala rezervisanih za jedan ili drugi smer može se menjati dinamički, u zavisnosti od konkretnog saobraćaja.



Slika 4-33. Okviri i vremenski intervali **u** dvosmemom prenosu podelom vremena.

Saobraćaj ka korisniku preslikava u vremenske intervale bazna stanica zato što je ona isključivo odgovorna za taj smer. Složeniji saobraćaj od korisnika (ka baznoj stanici) zavisi od zahtevanog kvaliteta usluge. Ubrzo ćemo se ponovo vratiti na dodelji- vanje vremenskih intervala kada budemo opisivali MAC podsloj.

Fizički sloj ima i zanimljivu sposobnost da u jedinstvenom fizičkom prenosu podataka više MAC okvira pakuje jedan uz dragi. Na taj način se bolje islcorišćava spektar frekvencija jer se smanjuje broj preambula i zaglavlja okvira fizičkog sloja.

Treba naglasiti da se u fizičkom sloju koristi Hamingov kod za ispravljanje grešaka u hodu. U skoro svim drugim mrežama, odtrivanje grešaka se prepušta kontrolnom zbiru i tada se zahteva ponovno slanje neispravnog okvira. Međutim, u širokopojasnom prenosu podataka na većem području očekuje se toliko grešaka da se one ispravljaju ne samo u višim slojevima pomoću kontrolnog zbira, već i u fizičkom sloju.

Zbog višestrukog ispravljanja grešaka kanal izgleda bolje nego u stvarnosti (u istom smislu kao što CD-ROM izgleda veoma pouzdano) ali samo zato što je više od polovine bitova namenjeno ispravljanju grešaka u fizičkom sloju.

4.5.4 Protokol MAC podsloja mreže 802.16

Sloj veze podataka podeljen je u tri podsloja (slika 4-31). Pošto se kriptografijom nećemo baviti sve do 8. poglavlja, teško je u ovom trenutku objasniti kako radi pod- sloj za bezbednost. Biće dovoljno ako kažemo da se nepovredivost svih prenetih podataka postiže njihovim šifrovanjem. Pri tome, u svakom okviru se šifruju samo korisnički podaci, a ne i zaglavlje. To znači da onaj ko prisluškuje može da utvrdi lco s kim razgovara, ali ne može da otkrije sadržaj konverzacije.

Ako već nešto znate o kriptografiji, sledeći pasus će vam ukratko objasniti rad podsloja za bezbednost. Ako o kriptografiji ne znate baš ništa, sledeći pasus neće na vas ostaviti utišale (ali ćete mu se možda ponovo vratiti kada pročitate 8. poglavlje).

Pri uspostavljanju veze između pretplatnika i bazne stanice, dve strane međusobno proveravaju identitete šifrujući poruke javnim ključem pomoću algoritma RSA i služeći se sertifikatima X.509. Na korisničke podatke se primenjuje simetričan ključ, bilo ulančavanjem blok-šifara po sistemu DES ili trostrukim DES sistemom uz dva ključa. Uskoro će najverovatnije biti dodat napredniji sistem šifrovanja AES (Rijnda- el). Integritet poruka proverava se algoritmom SHA-1.1 nije bilo tako strašno, zar ne?

Obratimo sada pažnju na zajednički deo MAC podsloja. MAC okviri zauzimaju određeni (ceo) broj vremenskih intervala fizičkog sloja. Svaki okvir sadrži više pod- olcvira, a prva dva su mape za saobraćaj ka korisniku i od njega. U mapama je naznačeno šta se u kom vremenskom intervalu nalazi i koji su intervali slobodni. Mapa za saobraćaj ka korisniku sadrži i razne sistemske parametre za obaveštavanje korisnika koji su se tek uključili.

Kanal za prenos ka korisniku radi prilično jednostavno. Bazna stanica odlučuje šta će da stavi u koji podokvir. Kanal za prenos ka baznoj stanici složeniji je jer se za njega nezavisno nadmeće više korisnika koji ne znaju jedan za drugog. Njegovo dode- ljivanje usko je vezano za kvalitet usluge, koji se razvrstava u četiri klase:

1. Usluga s konstantnom brzinom prenosa.
2. Usluga s promenljivom brzinom prenosa u realnom vremenu.
3. Usluga s promenljivom brzinom prenosa koja se ne izvršava u realnom vremenu.
4. Najbolja moguća usluga.

Sve usluge u mreži 802.16 izvršavaju se sa uspostavljanjem direktne veze, a vezi se pridružuje jedna od navedenih klasa tek kada se veza uspostavi. To je potpuno drugačije nego kod mreže 802.11 ili Etherneta, gde se u MAC podsloju ne uspostavlja veza.

Usluga s konstantnom brzinom namenjena je prenosu nekomprimovanog govora, slično prenosu kanalom TI. Ona treba da pošalje određenu količinu podataka služeći se vremenskim intervalima koji se unapred dodeljuju vezi ovog tipa. Kada se na taj način obezbedi potreban propusni opseg, vremenski intervali su automatski raspoloživi - nije potrebno zahtevati ih pojedinačno.

Prenos promenljivom brzinom u realnom vremenu koristi se za komprimovan multimedijski sadržaj i druge interaktivne, ali nezahtevne aplikacije, jer se u takvim slučajevima potreba za propusnim opsegom neprestano menja. Usluga se realizuje tako što bazna stanica periodično poziva korisnika da bi saznala koliki mu je propusni opseg potreban sledeceg trenutka.

Prenos promenljivom brzinom koji se ne izvršava u realnom vremenu rezervisan je za dugotrajno slanje, npr. za slanje dugačkih datoteka. Tokom ove usluge bazna stanica često poziva korisnika, ali ne u nekim strogo propisanim intervalima. Pretplatnik na uslugu slanja konstantnom brzinom može u jednom od tih okvira podesiti odgovarajući bit i tako zahtevati poziv za slanje dodatnih podataka (promenljivom brzinom).

Ako korisnička stanica ne odgovori na poziv *k* puta uzastopce, bazna stanica će je skinuti s liste stanica lcoje se pojedinačno pozivaju i svrstati u grupu za višesmerno pozivanje. Kada takva grupa dobije poziv, svaka stanica iz nje može da odgovori - sve ravnopravno konkurišu za uslugu. Na taj način, dragoceni individualni pozivi ne troše se na stanice koje retko šalju i/ili primaju podatke.

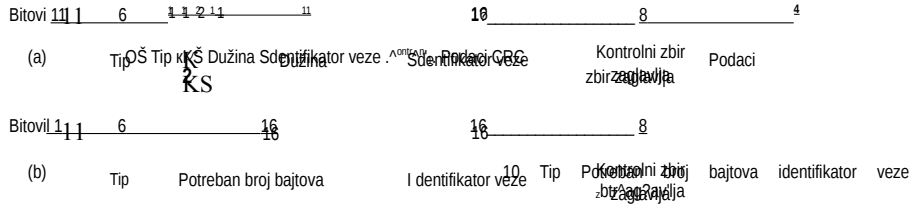
Poslednja, najbolja moguća usluga koristi se za sve ostalo. Pozivi se ne šalju i za propusni opseg se ravnopravno takmiče svi pretplatnici na ovu uslugu. Oni svoje zah- teve smeštaju u za to predviđene vremenske intervale u mapi saobraćaja ka baznoj stanici. Ako zahtev bude prihvaćen, bazna stanica će to naznačiti u sledećoj mapi saobraćaja ka korisnicima. U suprotnom, korisnici kojima zahtev nije prihvaćen moraće kasnije ponovo da ga pošalju. Da bi se proredilo sukobljavanje, koristi se Ethernet algoritam binarnog eksponencijalnog odustajanja.

Standard predviđa dva načina dodeljivanja propusnog opsega: po stanici i po vezi. U prvom slučaju, korisnička stanica prikuplja potrebe svih korisnika u zgradi i ispostavlja kolektivan zahtev. Kada dobije traženi propusni opseg, ona ga po svom nahođenju deli drugim korisnicima. U drugom slučaju, bazna stanica direktno upravlja svakom vezom.

4.5.5 Struktura okvira u mreži 802.16

Svi MAC okviri počinju zaglavljem iste strukture, a iza njega mogu da slede korisnički podaci i kontrolni zbir (CRC), kao što je prikazano na slici 4-34. Korisnički podaci se ne očekuju, na primer, u sistemskim okvirima pomoću kojih se zahteva vremenski interval. Ne treba da vas iznenadi ni to što kontrolni zbir takođe nije obavezan: greške se ispravljaju u fizičkom sloju, a na prenos u realnom vremenu niko još nije pokušao da primeni ponovno slanje okvira. Ako nema ponovnog slanja, šta će nam kontrolni zbir?

Sledi kratko objašnjenje polja iz zaglavlja sa slike 4-34(a). Bit *Š* naznačuje da li su korisnički podaci šifrovani. Polje *Tip* određuje vrstu okvira, a najčešće to da li je primenjeno pakovanje ili fragmentiranje. Bit *UKZ* označava da li postoji ukupan kontrolni zbir (za ceo okvir). U polju *KŠ* naznačava se eventualni ključ za šifrovanje. *Dužina* je dužina okvira uključujući zaglavlje, a *Identifikatorom veze* okvir se pridružuje odgovarajućoj vezi. Na kraju dolazi *Kontrolni zbir zaglavlja* izračunat pomoću polinoma $x^8 + x^2 + x + 1$.



Slika 4-34. (a) Opšta struktura okvira, (b) Okvir sa zahtevom za dodelu propusnog opsega

Na slici 4-34(b) prikazano je drugačije zaglavlje, za okvir koji zahteva propusni opseg. Ono počinje bitom 1 umesto bitom 0 i ima strukturu sličnu normalnom zaglavlju, osim što drugi i treći bit obrazuju 16-bitni broj kojim se saopštava propusni opseg potreban za prenos određenog broja bajtova. Okviri sa zahtevom za dodelu propusnog opsega ne sadrže korisničke podatke, niti ukupan kontrolni zbir.

Svašta bi se još moglo reći o mreži 802.16, ali ne na ovom mestu. Ako želite dopunska objašnjenja, potražite ih u tekstu samog standarda.

4.6 BLUETOOTH

Godine 1994, kompanija Ericsson poželela je da bez kablova poveže svoje mobilne telefone s drugim uređajima (npr. sa LDP računarima). Tada je sa četiri drage kompanije (IBM, Intel, Nokia, Toshiba) formirala konzorcijum - Specijalnu interesnu grupu (SIG) za standardizovanje veze između računara i komunikacionih uređaja, ostvarene radio-talasima kratkog dometa i male snage. Projekat je dobio ime **Bluetooth** po Ha- raldu Blaatandu II (940-981), vikinškom kralju koji je ujedinio (tj. pokorio) Dansku i Norvešku, talcođe „bez kablova“. (Blaatand bukvalno znači „plavi zub“, engl. Bluetooth, a logotip Bluetootha su inicijali HB, ispisani nordijskim ranama. Prirn. prev.)

Prvobitna zamisao je podrazumevala samo uklanjanje kablova između uređaja, ali se ideja ubrzo otela kontroli i proširila na područje bežičnih lokalnih mreža. Iako je to proširilo mogućnost primene standarda, Bluetooth je kao glavnog konkurenta dobio mrežu 802.11. Još gore, ta dva sistema su se električno ometala. Treba pomenuti i to da je pre nekoliko godina Hevvlett-Packard razvio bežičnu mrežu za povezivanje periferijskih komponenata računara, zasnovanu na talasima iz infracrvenog područja, ali ona nikada nije bila šire prihvaćena u praksi.

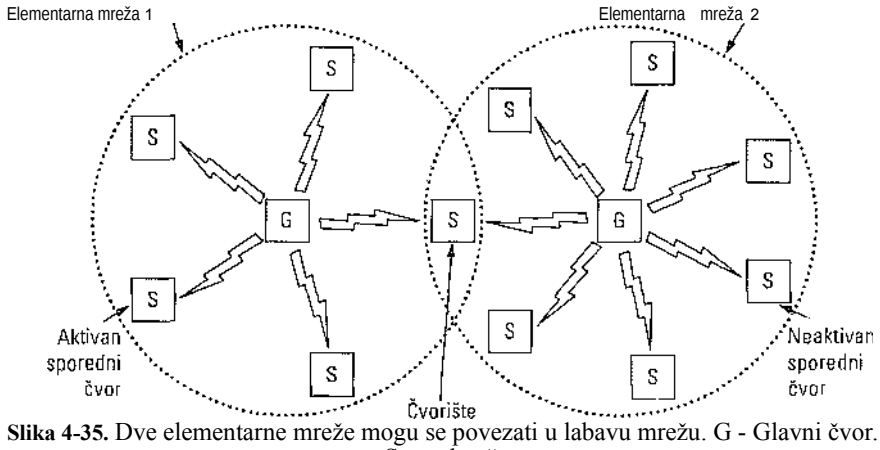
Grupu SIG sve to nije obeshrabrilo i ona je jula 1999. na 1500 strana podnela specifikaciju prve verzije (V1.0) Bluetootha. Ubrzo posle toga, razmatrajući lične bežične mreže 802.15, IEEE je prihvatio dokument Bluetooth kao osnovu i počeo da ga razrađuje. Iako standardizovanje nečega za šta već postoji detaljna specifikacija i što ne zahteva usuglašavanje nekompatibilnih realizacija može izgledati čudno, istorija pokazuje da nezavisna institucija, kao što je IEEE, pri razradi standarda često prornoviše i upotrebu određene tehnologije. Jasnoće radi, naglasimo da Bluetooth predstavlja specifikaciju za čitav sistem, od fizičkog sloja do sloja aplikacija. IEEE komitet za

mrežu 802.15 standardizovao je samo fizički sloj i sloj veze podataka; ostali protokoli iz skupa nalaze se van njegovog delokmga.

IEEE je 2002. godine prihvatio prvi PAN standard, mrežu 802.15.1, ali specijalna interesna grupa za Bluetooth i dalje radi na njenom poboljšanju. Iako su SIG i IEEE standardizovali dve različite verzije Bluetootha, nadamo se da će one u bliskoj budućnosti postati jedinstven standard.

4.6.1 Arhitektura Bluetootha

Počnimo opis Bluetootha kratkim pregledom onoga što sistem sadrži i šta treba da radi. Osnovna gradivna jedinica sistema Bluetooth je elementarna mreža (engl. *pi- conet),* koja se sastoji od glavnog čvora i najviše sedam aktivnih sporednih čvorova unutar kruga od 10 metara. U istoj (većoj) prostoriji može biti više elementarnih mreža, čak povezanih preko čvorišta, kao na slici 4-35. Povezan skup elementarnih mreža zove se labava mreža (engl. *scatternef).*



Slika 4-35. Dve elementarne mreže mogu se povezati u labavu mrežu. G - Glavni čvor. S - Sporedan čvor.

Osim najviše sedam aktivnih sporednih čvorova u svakoj elementarnoj mreži, u čitavoj mreži može da bude i najviše 255 takvih neaktivnih (engl. *parked)* čvorova. To su uređaji koje je glavni čvor „uspavao“ da bi im poštedeo baterije. Kada je uspavan, uređaj reaguje samo na aktivirajući signal glavnog čvora. Postoje još dva stepena budnosti uređaja („hold“ i „sniff“), ali nas oni ovde ne zanimaju.

Razlog za uvođenje hijerarhije uređaja (glavni/sporedni) jeste to što su projektanti želeli da cenu ugradnje kompletnih Bluetooth čipova spuste ispod .5 dolara. Zbog toga su sporedni (podređeni) uređaji prilično „glupi“ i rade uglavnom samo ono što im glavni čvor naredi. Elementarna mreža je, u suštini, centralizovan TDM sistem, u kome glavni čvor upravlja sistemskim satom i određuje uređaj koji može da komunicira u određenom vremenskom intervalu. Sva komunikacija se odvija između glavnog čvora i sporednih čvorova; direktna komunikacija između sporednih čvorova nije moguća.

4.6.2 **Primene** sistema Bluetooth

Pomoću mrežnih protokola uglavnom se samo uspostavljaju kanali između sugovornika, a aplikacijama se ostavlja da ih iskoriste na najbolji mogući način. Na primer, specifikacijom mreže 802.11 ne određuje se da li će korisnici svoje prenosive računare upotrebiti za e-poštu, za pregledanje Weba ili za nešto treće. Nasuprot tome, specifikacijom Bluetooth Vl.l podržava se 13 konkretnih primena i za svaku obez- beđuje zaseban skup protokola. Takav pristup, nažalost, čini sistem mnogo složenijim, pa ćemo odustati od njegovog detaljnog opisivanja. Pomenutih 13 aplikacija, tzv. profili, prikazani su na slici 4-36. Kada ih i samo letimično pregledamo, biće nam jasnije čemu je namenjen Bluetooth.

Ime	**Opis**
Generic access	Procedure za upravljanje vezom
Service discovery	Protokol za otkrivanje ponuđenih usluga
Serial port	Zamena za serijski kabl
Generic object exchange	Definiše klijentsko-serverski odnos za premeštanje objekta
LAN access	Protokol za vezu između pokretnog računara i fiksne lokalne mreže
Dial-up networking	Omogućava prenosivom računaru da uputi poziv pomoću mobilnog telefona
Fax	Omogućava faks-mašini đa komunicira preko mobilnog telefona
Cordless telephony	Povezuje slušalicu s fiksnom bazom bežičnog telefona
Intercom	Digitalni voki-toki
Headset	Omogućava komuniciranje glasom slobodnih ruku
Object push	Obezbeđuje način za razmenu jednostavnih objekata
File transfer	Obezbeđuje opštiji način razmene datoteka
Synchronization	Omogućuje LDA računaru da se sinhronizuje s drugim računarom.

Slika 4-36. Profili Bluetootha.

Profil *Generic access* nije aplikacija, već pre osnova na kojoj se prave aplikacije. Glavni mu je zadatak da obezbedi način za uspostavljanje i održavanje bezbedne veze (kanala) između glavnog i sporednih čvorova. Ni profil *Service discovery* nije ništa konleretniji; pomoću njega uređaj otkriva usluge koje nude dragi uređaji. Svi uređaji povezani sistemom Bluetooth treba da imaju ova dva profila; ostali su opcioni.

Profil *Serial port* obezbeđuje protokol za prenos podataka koji koristi većina ostalih profila. On emulira kabl za serijski prenos podataka i naročito je koristan za starije aplikacije koje takav kabl očekuju.

Profil *Generic object exchange* definiše klijentsko-serverski odnos pri preme- štanju podataka. Klijenti pokreću operacije, ali sporedan čvor može da bude u ulozi i klijenta i servera. Kao i profil serijskog priključka, i ovaj profil je alatka koju koriste drugi profili.

Naredna tri profila tiču se rada u mreži. Profil *LAN access* omogućava Bluetooth uređaju da se poveže s fiksnom mrežom. Taj profil je direktan konkurent mreži
802.11. Profil *Dial-up networking* bio je prvobitan motiv čitavog projekta. Taj profil

omogućava prenosivom računani da se bežičnim putem poveže s mobilnim telefonom u koji je ugrađen modem. Profil *Fax* ima sličnu funkciju - omogućava bežičnoj faks-mašini da prima i šalje faksove preko mobilnog telefona bežičnim putem.

Sledeća tri profila tiču se telefonije. Profil *Cordless telephony* omogućava povezivanje slušalice bežičnog telefona s njegovom fiksnom bazom. Trenutno, fiksni bežični telefoni većinom ne mogu da se koriste kao mobilni, ali će se oni - u budućnosti - možda izjednačiti. Profil *Intercom* omogućava da dva telefona međusobno komuniciraju u stilu voki-toki uređaja. Na kraju, profil *Headset* omogućava komunikaciju između kompleta za glavu i njegove bazne stanice da biste, na primer, mogli da razgovarate telefonom dok upravljate automobilom.

Preostala tri profila rezervisana su u potpunosti za razmenjivanje objekata između dva bežična uređaja. To mogu da budu posetnice, slike ili datoteke. Profil *Synchronization* posebno je namenjen prenosu podataka u LDA ili prenosivi računar kada se polazi na teren, i preuzimanje podataka iz njih po povratku s terena.

Da li je bilo zaista potrebno da se navedene oblasti primene opišu tako detaljno i da se za svaku smisli zaseban skup protokola? Najverovatnije nije, ali treba imati u vidu da su delove standarda razrađivale različite grupe stručnjaka od kojih se svaka usredsređivala samo na pojedinačan problem i napravila profil samo za njega. Razmišljajte o tome kao o praktičnoj realizaciji Konvejevog (Conway) zakona. (U april- skom broju časopisa *Datamation* iz 1988. godine, Melvin Konvej je primetio da će grupa od *n* stručnjaka kojoj je zadato da napravi programski prevodilac, napraviti prevodilac sa *n* prolaza; drugim recima, struktura softvera odražava strukturu grupe koja ga je napravila.) Verovatno bi bilo moguće, umesto 13 protokola, zadržati samo 2 - jedan za razmenu datoteka i dragi za komuniciranje u realnom vremenu.

4.6.3 Skup Bluetooth protokola

Standard za sistem Bluetooth sadrži mnogo protokola razvrstanih po slojevima. Struktura slojeva ne prati modele OSI, TCP/IP, 802, niti ijedan drugi poznati model. Međutim, IEEE se trudi da Bluetooth bolje prilagodi modelu mreže 802. Osnovna arhitektura Bluetooth protokola kakvu je predložio komitet za mrežu 802, prikazana je na slici 4-37.

Na dnu je fizički radio-sloj koji prilično dobro odgovara fizičkom sloju u modelima OSI i 802. U njemu se obavljaju radio-prenos i modulacija. Glavna pitanja u vezi sa ovim slojem tiču se sniženja cene uređaja za njegovo realizovanje da bi se mogli masovno proizvoditi.

Dragi, osnovni sloj, pomalo liči na poznati MAC podsloj, ali obuhvata i elemente fizičkog sloja. On određuje kako glavni čvor upravlja vremenskim intervalima i kako se vremenski intervali grupišu u okvire.

Slika 4-37. Arhitektura Bluetooth protokola u verziji 802.15.

Zatim dolazi sloj s grupom više-manje srodnih protokola. Sistem za rad s vezom uspostavlja logičke kanale između uređaja, brine o napajanju, proveri identiteta i kvalitetu usluge. Protokol za LLC adaptaciju (poznat i kao L2CAP) služi da od viših slojeva zakloni detalje prenosa. On je analogan standardnom LLC podsloju mreže 802, ali se formalno od njega razlikuje. Tu su i protokoli za prenos zvuka (Audio), odnosno za upravljanje (Control), kojima aplikacije mogu direktno da pristupe - ne prolazeći kroz protokol L2CAP.

Sledeći viši je posrednički sloj koji sadrži nekoliko različitih protokola. Protokol za upravljanje logičkom vezom (LLC) iz mreže 802 na to mesto je postavio IEEE zbog kompatibilnosti s dragim mrežama tipa 802. Protokoli RFcomm, Telephony i Setvice Discovery predstavljaju originalne Bluetooth protokole. RFcomm (engl. radio/requen- cy communication) emulira standardni serijski priključak PC računara, preko koga se, pored ostalih uređaja, s računarom povezuju tastatura, miš i modem. Namenjen je upravo priključivanju starijih uređaja. Protokol Telephony koriste tri profila koji rade sa zvukom u realnom vremenu. Taj protokol služi i za pozivanje i za prekidanje veze. Na kraju, pomoću protokola Service Discovery pronalaze se usluge na mreži.

Aplikacije i profili smešteni su u najvišem sloju. Oni svoj posao obavljaju koristeći protokole iz nižih slojeva. Svaka aplikacija ima svoj namenski podskup protokola. Pojedinačni uređaji, kao što je audio komplet za glavu, ne sadrže nikakve druge protokole osim onih koji su njima potrebni.

U narednim odeljcima ispitaćemo tri najniža sloja iz skupa Bluetooth protokola jer oni grubo odgovaraju fizičkom sloju i MAC podsloju.

4.6.4 **Radio-sloj** sistema Bluetooth

Glavni čvor razmenjuje bitove sa sporednim čvorovima kroz radio-sloj. To je sistem male snage, dometa do 10 metara, koji radi u ISM području na 2,4 GHz. Područje je iz- deljeno u 79 kanala, svaki širine 1 MHz. Koristi se modulacija s frekventnim pome- ranjem, što uz 1 bit po hercu daje ukupnu brzinu prenosa 1 Mb/s, ali se veliki deo ovoga potroši na sistemske podatke. Da bi se kanali ravnopravno dodeljivali, koristi se skoko- vito frekventno širenje spektra uz 1600 skokova u sekundi i vreme boravka 625 |is. Svi čvorovi u elementarnoj mreži istovremeno prelaze na sledeću frekvenciju koju diktira glavni čvor.

Pošto obe mreže (Bluetooth i 802.11) rade sa istih 79 kanala u frekventnom ISM području

na 2,4 GHz, one se međusobno ometaju. Buduc'i da Bluetooth brže menja frekvenciju nego mreža 802.11, verovatnije je da će Bluetooth uređaj upropastiti pre- nos koji se odvija mrežom 802.11, nego obrnuto. Mreže 802.11 i 802.1.5 su IEEE standardi i zato ta organizacija neprestano pokušava da nađe rešenje problema, ali to ne obećava mnogo jer oba sistema koriste ISM područje iz istog razloga: ono nije podložno licenciranju. U sistemu 802.11a koristi se drugo ISM područje (na 5 GHz), ali taj sistem ima mnogo manji domet od sistema 802.11b (zbog prirode radio-talasa), tako da ni mreža 802.11 a ne predstavlja univerzalno rešenje. Neke kompanije su reši- le problem tako što su potpuno zabranile korišćenje Bluetootha. Tržišno rešenje bi bilo da mreža koja je jača u političkom i ekonomskom smislu natera onu slabiju da svoje standarde izmeni i tako prekine ometanje. Neka razmišljanja na ovu temu možete da nađete kod Landsforda i saradnika (2001).

4.6.5 Osnovni sloj sistema Bluetooth

Osnovni sloj u Bluetoothu po koncepciji je najbliži MAC podsloju. On tok sirovih bitova pretvara u okvire i definiše neke ključne formate. U najjednostavnijem slučaju, glavni čvor svake elementarne mreže definiše niz vremenskih intervala od po 625 ps, pri čemu su parni intervali rezervisani za emitovanje glavnog čvora, a neparni za emi- tovanje sporednih. To predstavlja klasično multipleksiranje podelom vremena pola- pola. Okvir može da zauzme 1, 3 ili 5 vremenskih intervala.

Učestalost menjanja frekvencije omogućava pri svakom skoku stabilizovanje bežičnih kola tokom 2.50-260 ps. Kola se mogu i brže stabilizovati, ali uz veće troškove. Od okvira koji staje u jedan vremenski interval (625 bitova), posle stabilizo- vanja ostaje 366 iskoristivih bitova; 126 bitova otpada na pristupni kod i zaglavlje, tako da za podatke ostaje 240 bitova. Kada se uzastopno nadoveže pet vremenskih intervala, za njih je potreban samo jedan (i to nešto kraći) period stabilizovanja, tako da od 5 x 625 = 3125 bitova iz pet vremenskih intervala, osnovnom sloju ostaje 2781 bit. Prema tome, duži okviri su efikasniji od okvira koji staju u jedan vremenski interval.

Između glavnog i sporednog čvora svaki okvir se prenosi logičkim kanalom, zvanim **veza** (engl. *link)*. Postoje dva načina povezivanja. Prvi je **asinhrono povezivanje bez uspostavljanja direktne veze** (engl. *Asynchronous Connection-Less, ACL),* koje se koristi za povremeni saobraćaj komutiranjem paketa. Ti podaci dolaze iz sloja L2CAP pošiljaoca i isporučuju se sloju L2CAP primaoca. ACL razmena paketa odvija se na principu najbolje moguće usluge, pri čemu se ništa ne garantuje. Okviri se mogu izgubiti i možda će morati da se šalju ponovo. Sporedni čvor može imati samo jednu takvu vezu s glavnim čvorom.

Drugi način je **sinhrono povezivanje sa uspostavljanjem direktne veze** (engl. *Synchronous Connection Oriented, SCO),* namenjeno razmeni podataka u realnom vremenu, npr. telefonskom saobraćaju. Za svaki smer na takvom kanalu dodeljuje se tačno definisan vremenski interval. Zahvaljujući tome što je vreme u SCO vezama kritičan činilac, okviri se nikada ne šalju ponovo. Da bi se povećala pouzdanost veze, umesto toga se može primeniti ispravljanje grešaka u hodu. Sporedan čvor može da bude povezan sa svojim glavnim čvorom pomoću najviše tri takve veze. Svaka SCO veza može da podrži jedan impulsno-kodno modulirani govorni kanal brzine 64.000 b/s.

4.6.6 Sloj L2CAP sistema Bluetooth

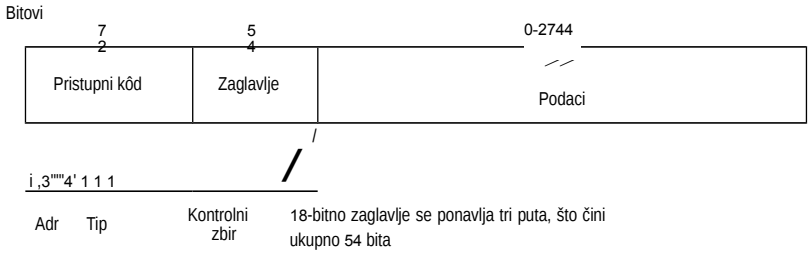
Sloj L2CAP ima tri glavne funkcije. Najpre, on od viših slojeva prihvata pakete veličine do 64 KB i raspoređuje ih u okvire za slanje. Na drugom kraju, od okvira se ponovo sklapaju paketi.

Zatim, on multipleksira i demultipleksira pakete iz više izvora. Kada ponovo sklopi paket, sloj L2CAP određuje protokol iz višeg sloja kome će ga predati; to, na primer, može biti protokol Telephony ili protokol RFcomm.

Potom, L2CAP obezbeđuje zahtevani kvalitet usluga u trenutku uspostavljanja veze i tokom normalnog rada. Isto tako, on pri uspostavljanju veze dogovara i maksimalnu veličinu polja s korisničkim podacima da uređaj koji može da razmenjuje velike pakete ne bi istisnuo uređaj koji može da radi samo s manjim paketima. Ta predostrožnost je neophodna jer ne mogu svi uređaji da obrade paket maksimalne veličine 64 KB.

4.6.7 Struktura Bluetooth okvira

Postoje okviri različitih formata, a najvažniji je prikazan na slici 4-38. On počinje pristupnim kodom koji najčešće identifikuje glavni čvor tako da sporedni čvorovi koji se nalaze u dometu dva glavna čvora mogu da utvrde koji saobraćaj im je namenjen. Zatim dolazi zaglavlje veličine 54 bita koje sadrži tipična polja MAC podsloja. Iza njega je polje za korisničke podatke, veličine najviše 2744 bita (za prenos u pet vremenskih intervala). Kada se prenos vrši u jednom vremenskom intervalu, format je isti, samo što je polje za korisničke podatke dugačko 240 bitova.



Slika 4-38. Struktura najčešćeg Bluetooth okvira.

Objasnimo ukratko struktura zaglavlja. Polje *Adresa* određuje jedan od osam aktivnih uređaja kome je paket namenjen. U polju *Tip* je vrsta okvira (ACL, SCO, pozivni ili prazan okvir), vrsta postupka za ispravljanje grešaka u polju s podacima, i broj vremenskih intervala koje zauzima okvir. Bitom *T {Tok)* sporedni čvor objavljuje da mu je

bafer pun i da više ne može da prima podatke. To je rudimentaran način upravljanja tokom podataka. Bit *P* (*Potvrda*) služi za šlepovanje potvrde o stizanju okvira. Bitom *RB (Redni broj)* numerišu se okviri i tako utvrđuje koji okvir treba ponovo poslati. Protokol je tipa „stani i čekaj", pa je za numerisanje okvira dovoljan jedan bit. Na kraju je 8-bitni *Kontrolni zbir* zaglavlja. Sva navedena polja (ukupno 18 bitova) ponavljaju se istim redosledom tri puta, čime se formira 54-bitno zaglavlje okvira sa slike 4-38. Primalac jednostavnim kolom proverava sve tri kopije istog bita. Ako su iste, bit se prihvata. Ukoliko se jedna od njih razlikuje, prihvataju se druge dve. Na taj način, troši se 54 bita da bi se poslalo 10 bitova zaglavlja. Pouzdano slanje podataka u bučnom okruženju, kada se koriste jeftini i u računarskom smislu jednostavni uređaji male snage (2,5 mW), može se postidi samo ako se oni šalju više puta.

Polje s podacima u ACL okviru može da bude u više formata. U tom pogledu, jednostavniji su SCO okviri: njihovo polje s podacima uvek zauzima 240 bitova. Defi- nisane su tri varijante, sa 80, 160 i 240 stvarnih podataka, dok se ostatak koristi za ispravljanje grešaka. U najpouzdanijoj verziji (sa 80 bitova korisničkih podataka), sadržaj polja se ponavlja tri puta uzastopce, kao u zaglavlju.

Pošto sporedni čvor može da koristi samo neparne vremenske intervale, njemu pripada 800 intervala u sekundi, baš kao i glavnom čvoru. Uz 80 bitova stvarnih korisničkih podataka po okviru, kapacitet kanala i u jednom i u drugom smeru iznosi 64.000 b/s, što je upravo dovoljno za jedan potpun dupleksni PCM govorni kanal (zbog čega je i izabrano da se frekvencija menja 1600 puta u sekundi). Navedeni brojevi znače da potpun dupleksni govorni kanal brzine 64.000 b/s u svakom smeru u lcome se koristi najpouzdaniji format okvira, potpuno zasićuje elementarnu mrežu uprkos tome što je njen osnovni propusni opseg 1 Mb/s. U najnepouzdanijoj varijanti (240 bitova po vremenskom intervalu, bez ponavljanja podataka na ovom nivou), istovremeno se mogu podržati tri potpuna duplelcsna govorna kanala, zbog čega se sporednom čvoru dozvoljavaju najviše tri SCO veze.

O Bluetoothu se može još dugo pričati, ali ne ovde. Detaljnija obaveštenja potražite kod Bhagwata (2001), Bisdikiana (2001), Braya i Sturmana (2002), Haartsena (2000), Johanssona i saradnika (2001), Millera i Bisdikiana (2001) i Sairama i sa- radnika (2002).

4.7 KOMUTIRANJE U SLOJU VEZE

U mnogim organizacijama postoji više lokalnih mreža koje je poželjno povezati. Lokalne mreže se mogu međusobno povezivati tzv. **mrežnim mostovima** (engl. *brid- ges)* koji rade u sloju veze podataka. Mostovi pregledaju adrese veza u tom sloju i vrše potrebno usmeravanje. Posto ne zagledaju u polje s podacima okvira koji usmeravaju, oni mogu prosleđivati i pakete IPv4 (koji se danas koriste na Internetu), IPv6 (lcoji će se na Internetu koristiti u budućnosti), pakete AppleTalk, ATM, OSI i bilo koje druge pakete. Za razliku od toga, **usmerivači** (engl. *routers)* pregledaju adrese u paketima i

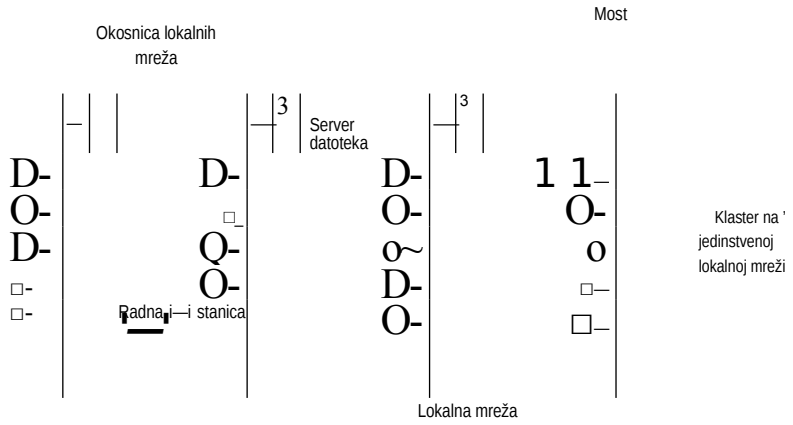
putanje zasnovane na njima. Iako se čini daje time razlika između mostova i usmeri- vača jasno postavljena, neke su novotarije (npr. komutirani Ethernet) uspele da naprave zbrku, ali o tome kasnije. U narednim odeljcima pozabavic'emo se mostovima i skretnicama, naročito u smislu povezivanja različitih lokalnih mreža serije 802. Iscrpnu raspravu o mostovima, skretnicama i srodnim temama naći ćete kod Perlmana (2000).

Pre nego što zađemo u tehnologiju rada mostova, osvrnimo se na situacije u kojima se mostovi najčešće koriste. Navešćemo šest razloga zbog kojih organizacija može da se odluči za više lokalnih mreža.

Prvo, mnoge univerzitetske katedre i delovi preduzeca imaju sopstvene lokalne mreže, najčešće da bi povezali sopstvene računare, radne stanice i servere. Pošto različite katedre (delovi preduzeca) imaju različite interne potrebe, i lokalne mreže im se razlikuju. Pre ili posle, međutim, nastaje potreba za međusobnim povezivanjem lokalnih meža, a tada su neophodni mostovi. U navedenom primeru nastaje više vrsta lokalnih mreža zahvaljujući autonomiji njihovih vlasnika.

Drugo, organizacija se može prostirati na više zgrada, međusobno znatno udaljenih. Ponekada je jeftinije da se lokalne mreže u pojedinim zgradama povežu mostovima ili laserskim vezama, nego da sve zgrade budu povezane kablom u jedinstvenu mrežu.

Treće, možda je neophodno da se ono što logički izgleda kao jedinstvena lokalna mreža izdeli u zasebne lokalne mreže da bi se struktura mreže prilagodila saobraćaju. Na mnogim univerzitetima, na primer, postoje hiljade računara koje koriste studenti i zaposleni. Datoteke se obično drže na serverima odakle ih korisnici mogu preuzeti na svoje računare. Već dimenzije ovakvog sistema isključuju mogućnost da se sve stanice priključe na istu lokalnu mrežu - potreban propusni opseg bio bi prevelik. Zbog toga se koristi više lokalnih mreža povezanih mostovima, kao na slici 4-39. Svaka lokalna mreža sadrži klaster radnih stanica sa sopstvenim serverom datoteka, tako da se najveći deo saobraćaja odvija unutar lokalnih mreža, ne opterećujući okosnicu.



Slika 4-39. Više lokalnih mreža povezanih okosnicom mogu da podrže vec'i saobraćaj od jedinstvene lokalne mreže.

Iako lokalne mreže obično prikazujemo na klasičan način - kao da su povezane spojnim kablovima (slika 4-39), naglašavamo da se one danas najčešće realizuju pomoću razvodnika ili specijalnih skretnica. Međutim, dugačak spojni kabl s više priključenih računara i razvodnik s njegovim računarima funkcionalno su identični. U oba slučaja, svi računari pripadaju istom domenu sukobljavanja i svi šalju okvire služeći se protokolom CSMA/CD. Kao što smo naučili, komutirane lokalne mreže su drugačije i na njih ćemo se ubrzo vratiti.

Četvrto, jedinstvena lokalna mreža bi u određenim slučajevima odgovarala gustini saobraćaja, ali je rastojanje između najudaljenijih umreženih računara preveliko (tj. veće od 2,5 km za Ethernet). Čak i kada postavljanje kablova ne bi predstavljalo problem, mreža ne bi radila zbog prevelikog vremena obilaska mreže. Jedino rešenje je da se takva mreža razbije u više delova i delovi povežu mostovima. Pomoću mostova se može povećati ukupno fizičko rastojanje između najudaljenijih računara, a da mreža ipak radi.

Peto, postavlja se pitanje pouzdanosti rada. Defektan čvor na jedinstvenoj mreži koji neprestano emituje smeće može da upropasti mrežu, pa se zato na kritičnim me- stima mogu postaviti mostovi koji, kao vrata za slučaj požara u zgradama, sprečavaju da podivljali čvor uništi ceo sistem. Za razliku od repetitora, koji kopira sve što mu stigne, most se može programirati u pogledu onoga šta prosleđuje.

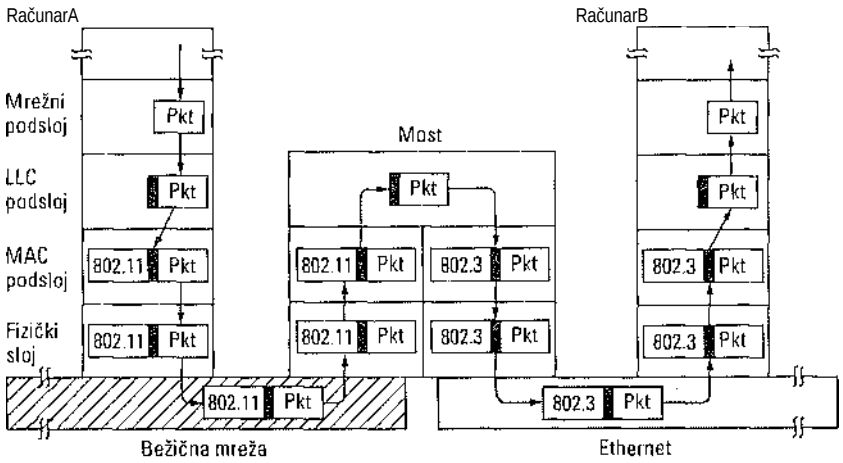
Šesto i poslednje, mostovi mogu da doprinesu opštoj bezbednosti organizacije. In- terfejsi u lokalnim mrežama uglavnom rade u promiskuitetnom režimu (engl. *pro- miscuous mode),* u kome se računam predaju *svi* okviri, a ne samo oni koji su na njega adresirani. Špijuni i radoznalci obožavaju ovaj režim. Kada postavite mostove na više mesta i ne prosleđujete osetljiv sadržaj, administrator sistema može da izoluje delove mreže tako da saobraćaj iz njih ne može da procuri i padne u pogrešne rake.

Mostovi bi u idealnom slučaju trebalo da rade potpuno neprimetno, što znači da biste računar mogli da premestite na drugi segment kabla a da ne menjate ni hardver, ni softver, ni konfiguracione tabele. Isto tako, trebalo bi da računari iz jednog segmenta budu u stanju da komuniciraju s računarima iz bilo kog drugog segmenta, bez obzira na razlike u njihovim lokalnim mrežama i lokalnim mrežama koje se nalaze između njih. To se ponekada postiže, ali ne uvek.

4.7.1 Mostovi između mreža 802.x i 802.y

Pošto smo objasnili zašto su mostovi potrebni, objasnimo i kako rade. Slika 4-40 pri- kazuje način rada jednostavnog mosta koji spaja dva priključka. Računar *A* na bežičnoj mreži 802.11 treba da pošalje paket fiksnom računara *B* na Ethemetu (802.3) s kojim je bežična mreža povezana. Paket se spušta u podsloj LLC gde dobija L.LC zaglavlje (na slici označeno kao crno). Paket zatim prelazi u MAC podsloj, gde dobija i zaglavlje formata 802.11 (takođe kao završni blok koji nije prikazan). Tako proširen paket emituje se u etar, gde ga hvata bazna stanica i shvata da treba da ga prosledi na fiksni Ethernet. Kada paket stigne do mosta između mreža 802.11 i 802.3, on počinje da se od fizičkog sloja probija naviše. Pri tome mu se u MAC podsloju mosta uklanja zaglavlje formata
802.11. Paket (koji sad sadrži samo LLC zaglavlje) prosleđuje se podsloju LLC mosta. U našem primeru, paket je namenjen mreži 802.3, pa zato opet silazi kroz slojeve mosta (ovoga

puta na strani mreže 802.3) i isporučuje se Ethernetu. Most koji povezuje *k* različitih lokalnih mreža treba da za svaku ima zaseban MAC podsloj i fizički sloj.

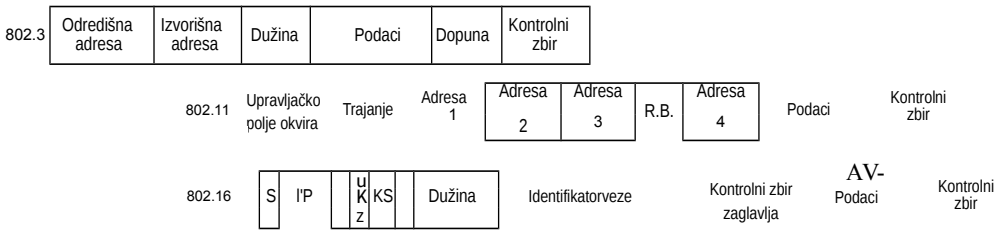


Slika 4-40. Način rada mosta između lokalnih mreža 802.11 i 802.3.

U ovom trenutku nam se čini da je premeštanje okvira iz jedne lokalne mreže u drugu jednostavan posao. To, međutim, nije tako. U ovom odeljku skrenućemo vam pažnju na teškoće s kojima će se sresti svako ko pokuša da napravi most između različitih lokalnih mreža tipa 802 (ili između različitih gradskih mreža). Najviše ćemo govoriti o mrežama 802.3, 802.11 i 802.16, ali postoje i druge mreže - svaka sa svojim problemima.

Počnimo od toga da svaka lokalna mreža koristi drugačiji format okvira (pogledajte sliku 4-41). Dok za razlilčite formate u Ethernetu i mrežama token bus, odnosno token ring možemo da okrivimo istoriju ili uobraženost velikih korporacija, te razlike su u našem slučaju u izvesnoj meri opravdane. Na primer, polje *Trajanje* u formatu 802.11 dugujemo protokolu MACAW - ono nema smisla u Ethernet mreži. Iz toga sledi da uz kopiranje okvira s jedne lokalne mreže na drugu obavezno ide i njihovo ponovno formatiranje, a to troši procesorsko vreme, zahteva ponovno izračunavanje kontrolnog zbira i omogućava nastanak grešaka u samom mostu.

Dragi problem je to što povezane lokalne mreže ne moraju da rade istom brzinom. Kada uputite dugačak niz međusobno priljubljenih okvira s brže na sporiju mrežu, most neće moći da ih obradi brzinom kojom pristižu. Na primer, ako s gigabitnog Etherneta šaljete bitove najvećom brzinom u lokalnu mrežu 802.11b (brzine 11 Mb/s), most će morati da ih baferuje nadajući se da neće prepunih memoriju. Mostovi koji povezuju tri i više lokalnih mreža uleću u isti problem ako više lokalnih mreža istovremeno pokuša da pošalje podatke istoj lokalnoj mreži, čak i onda kada sve rade istom brzinom.



Slika 4-41. Formati okvira u IEEE mrežama serije 802. Crtež nije u razmeri.

Treći, možda najvažniji problem, ogleda se u tome sto se u različitim mrežama tipa 802 dozvoljava različita maksimalna dužina okvira. Problem nastaje kada dugačak okvir treba uputiti lokalnoj mreži koja ne može da ga prihvati. Rasparčavanje okvira u ovom sloju ne dolazi u obzir jer svi protokoli pretpostavljaju da okviri ili stižu ili ne stižu, i nijedan ne predviđa ponovno sklapanje okvira iz delova. Time ne tvrdimo da se takvi protokoli ne bi mogli napraviti jer smo ih već videli na delu. Želimo samo da istaknemo da takvih protokola nema u sloju veze podataka, tako da mostovi ne smeju ni da pipnu korisničke podatke u okviru. Za navedeni problem nema načelnog rešenja - predugački okviri se moraju odbaciti. Toliko o transparentnosti rada.

I obezbeđenje podataka skopčano je s problemima. Mreže 802.11 i 802.16 podržavaju šifrovanje u sloju veze, ali ne i Ethernet. To znači da će razne usluge za šifrovanje koje postoje u bežičnim mrežama postati beskorisne kada podaci stignu na Ethernet. Gore je to što se podaci koji se na bežičnim stanicama šifruju, u sloju veze neće moći dešifrovati kad stignu na Ethernet. S druge strane, ako bežična stanica ne koristi šifrovanje, njene poruke ne mogu ostati privatne. Ovako ili onako, problem uvek postoji.

Problem bi se mogao resiti kada bi se šifrovanje obavljalo u višim slojevima, ali bi tada stanica mreže 802.11 morala da zna da li razgovara s drugom stanicom na istoj mreži (i da koristi šifrovanje u sloju veze) ili ne (i da ne koristi šifrovanje u sloju veze). Ako stanica treba da bira od slučaja do slučaja, prenos podataka više nije transparentan.

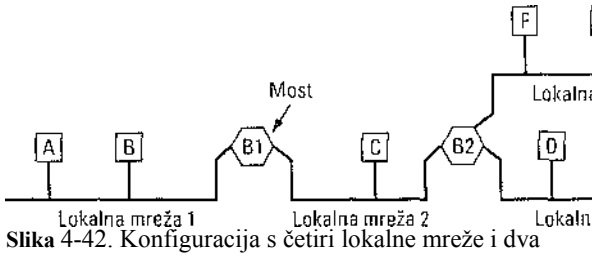
Poslednji problem tiče se kvaliteta usluge. Mreže 802.11 i 802.16 ostvaruju ga na različite načine: prva koristeći PCF režim, a druga konstantnom brzinom prenosa podataka. Na Ethernetu ne postoji koncept kvaliteta usluge, tako da će ta karakteristika saobraćaja na njemu nestati.

4.7.2 Međusobno povezivanje lokalnih mreža

U prethodnom odeljku bavili smo se povezivanjem dve različite lokalne mreže tipa 802 pomoću mosta. Međutim, u velikim organizacijama s više lokalnih mreža nastaju problemi i pri njihovom povezivanju, čak i kada su to sve Ethernet mreže. U idealnom slučaju, trebalo bi samo kupiti mostove prema IEEE standardu, uključiti konektore u mostove i sve bi trebalo odmah savršeno da radi. Ne bi trebalo da menjate hardver ili softver, da podešavate adresnu skretnicu ili da preuzimate tabele za usmeravanje,

odnosno parametre - ništa od svega toga. Samo treba da utaknete kablove i da mimo odete. Osim toga, mostovi ne bi trebalo ni na koji način da poremete rad lokalnih mreža. Drugim recima, mostovi bi morali da budu potpuno nevidljivi (i hardveru i softveru). Možda ne verujete, ali to je ipak moguće. Pogledajmo detaljnije kako se to može postići.

U svom najjednostavnijem obliku, nevidljivi most radi u promiskuitetnom režimu, prihvatajući svaki okvir koji se prenosi mrežama za koje je vezan. Razmotrimo, primera radi, konfiguraciju prikazanu na slici 4-42. Most B1 je spojen s lokalnim mrežama 1 i 2, a most B2 s mrežama 2, 3 i 4. Okvir namenjen stanici A koji mostu B1 stigne s mreže 1, može se odmah odbaciti jer se već nalazi na odredišnoj mreži, ali se okviri koji stižu s mreže 1, a namenjeni su stanicama C ili *F,* moraju proslediti.



Slika 4-42. Konfiguracija s četiri lokalne mreže i dva mosta.

Kad stigne okvir, most mora odlučiti da li da ga odbaci ili prosledi, a u drugom slučaju i gde da ga prosledi. On odluku donosi tražeći odredišnu adresu u velikoj tabeli ključeva koju ima. Tabela može da sadrži sva moguća odredišta zajedno s pripadajućim linijama (lokalnim mrežama). Na priiner, tabela mosta B2 sadržala bi stanicu A kao deo mreže 2, jer most B2 treba da zna samo mrežu u koju će uputiti okvir namenjen stanici A. Njega se ne tiče da li će okvir kasnije biti ponovo usmeravan.

Kada se mostovi prvi put priključe, njihove tabele ključeva su prazne. Nijedan most ne zna gde je bilo koje odredište, pa zato svi izvršavaju algoritam plavljenja: svaki pristigli okvir čije se odredište ne zna šalje se u sve priključene mreže, osim u mrežu iz koje je stigao. Vremenom, kao što ćemo pokazati u nastavku, svi mostovi nauče sva odredišta. Čim odredište postane poznato, okviri namenjeni njemu šalju se samo u ispravnu mrežu.

Nevidljivi mostovi (engl. *transparent bridges*) primenjuju algoritam učenja na iskustvu (engl. *backward learning*). Kao što smo videli, mostovi rade u promiskuitetnom režimu, tako da do njih stiže svaki okvir s priključenih mreža. Na osnovu izvorišnih adresa oni saznaju koja je stanica dostupna preko koje mreže. Na primer, ako mostu B1 sa slike 4-42 stigne okvir- od stanice *C* na mreži 2, on zna da se stanici C može pristupiti preko mreže 2, pa u svoju tabelu ključeva beleži da za okvire upućene stanici C treba da koristi mrežu 2. Svaki sledeći okvir koji dolazi s mreže 1, a upućen je stanici *C,* biće prosleđen, ali će okvir upućen stanici *C,* koji dolazi s *mreže* 2, biti odbačen.

Uključivanjem i isključivanjem stanica i mostova, ili njihovim premeštanjem, menja se topologija mreže. Da bi se moglo raditi s mrežama promenljive topologije, kad god se u tabelu unese izvorišna adresa zabeleži se i vreme kada je okvir stigao. To vreme se ažurira uvek kada sa istog izvorišta stigne nov okvir. Na taj način, podatak zabeležen uz svaku adresu predstavlja vreme kada je s nje stigao poslednji okvir.

Specijalan proces periodično pregleda tabelu i iz nje izbacuje sve odrednice starije od

nekoliko minuta. Na taj način, ako računar isključimo iz lokalne mreže, premesti- mo ga na drugo mesto u zgradi i ponovo tamo u nju uključimo, on će posle nekoliko minuta raditi normalno, bez ikakve dodatne intervencije. Ako se računar tokom nekoliko minuta ne čuje, algoritam predviđa i to da se svaka poruka upućena na njegovu adresu šalje u sve mreže sve dok on sam ne pošalje okvir.

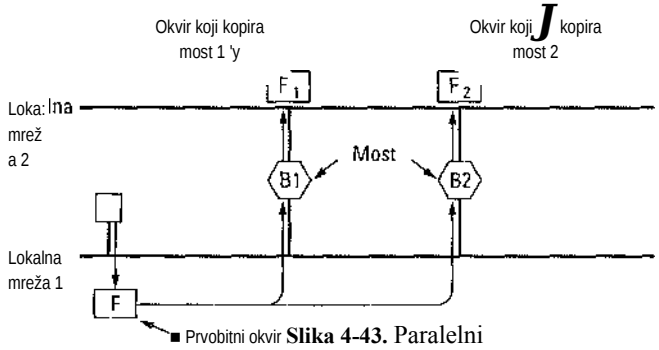
Postupak usmeravanja dolaznog okvira zavisi od lokalne mreže s koje potiče i lokalne mreže na koju je upućen:

1. Ako su odredišna i izvorišna mreža identične, okvir se odbacuje.
2. Ako odredišna i izvorišna mreža nisu identične, okvir se prosleđuje.
3. Ako je odredišna mreža nepoznata, okvir se šalje svim mrežama.

Navedena pravila obavezno se primenjuju na svaki pristigli okvir. Za pregledanje tabele i ažuriranje odrednica koriste se specijalni VLSI čipovi koji taj posao obavljaju za nekoliko mikrosekundi.

4.7.3 Mostovi u razgranatom stablu

Neke lokacije, radi veće pouzdanosti, koriste dva i više paralelnih mostova između mreža, kao na slici 4-43. Takav raspored, međutim, donosi dodatne probleme jer stvara petlje u topologiji.



Slika 4-43. Paralelni nevidljivi mostovi.

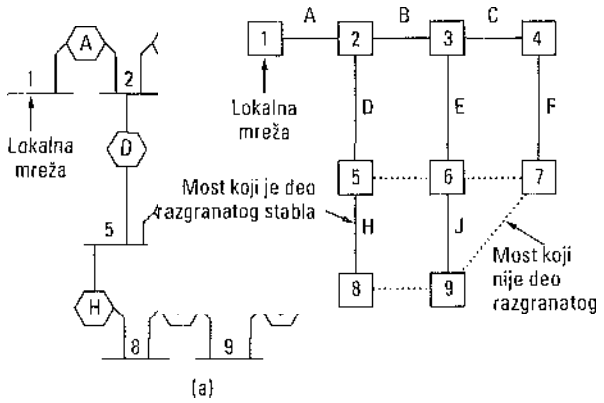
Možemo ih uočiti ako posmatramo kako se u mreži sa slike 4-43 obrađuje okvir *F* s nepoznatim odredištem. Svaki most, primenjujući uobičajena pravila obrade nepoznate odredišne adrese, koristi plavljenje, što u ovom primeru znači da kopira okvir na mrežu 2. Ubrzo zatim, i most 1 primećuje F_2, okvir s nepoznatim odredištem, koji

kopira na mrežu 1, generišući okvir *Fj* (nije prikazan). Slično tome, inost 2 kopira *F*\ na mrežu 1 generišući okvir F_4 (takođe nije prikazan). Most 1 sada prosleđuje F_4, a most 2 kopira *F*3. Taj ciklus se večno ponavlja.

Rešenje problema je u međusobnom komuniciranju mostova i prekrivanju postojeće topologije razgranatim stablom koje dopire do svake lokalne *mreže.* U stvari, pri tome se neke potencijalne veze između lokalnih mreža zanemaruju da bi se ostvarila pretpostavljena topologija u kojoj nema petlji. Na primer, na slici 4-44(a) vidimo devet lokalnih mreža povezanih s deset mostova. Ta konfiguracija se može predstaviti grafom čije čvorove čine lokalne mreže. Luk povezuje dve lokalne mreže koje su povezane mostom. Graf se može svesti na razgranato stablo ispuštajući lukove koji su na slici 4-44(b) označeni isprekidano. U razgranatom stablu postoji samo jedna putanja od bilo koje lokalne mreže do bilo koje druge lokalne mreže. Kada se mostovi dogovore 0 razgranatom stablu, dalje prosleđivanje okvira između mreža sledi njegovu konfiguraciju. Pošto postoje jedinstvene putanje iz svakog izvorišta do svakog odredišta, petlje nisu moguće.

Da bi se izgradilo razgranato stablo, mostovi prvo moraju da se slože koji će most biti njegov koren. Svaki od njih difuzno emituje svoj serijski broj za koji proizvođač garantuje daje jedinstven u svetu i most s najnižim serijskim brojem postaje koren stabla. Zatim se konstruišu najkraće putanje od korena do svakog mosta i lokalne mreže. To se zove razgranato stablo. Ako most ili lokalna mreža otkažu, konstruiše se novo stablo.

Opisanim algoritmom dobij amo jedinstvene putanje od svake lokalne mreže do korena stabla i odatle do svake drage lokalne mreže. Iako razgranato stablo pokriva sve lokalne mreže, u njemu ne moraju da postoje svi mostovi (da bi se izbegle petlje). Algoritam nastavlja da se izvršava i posle uspostavljanja razgranatog stabla - tokom normalnog rada - da bi automatski otkrivao promene topologije i ažurirao stablo. Ra- dia Perlman je autorka distribuiranog algoritma koji se koristi za konstruisanje razgranatog stabla (Perlman, 2000). Algoritam je postao IEEE standard 802. ID.

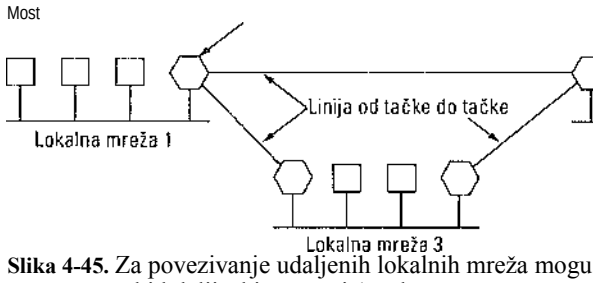


Slika 4-44. (a) Međusobno povezane lokalne mreže, (b) Razgranato stablo koje prekriva lokalne mreže. Isprekidane linije nisu deo stabla.

4.7.4 Daljinski mostovi

Mostovi obično služe za povezivanje dve ili više udaljenih lokalnih mreža. Na pri- mer, kompanija može da ima pogone (filijale) u više gradova, svaki sa svojom sopst- venom lokalnom mrežom. Bilo bi idealno kada bi te mreže bile međusobno povezane u jednu veliku lokalnu mrežu.

Proklamovani cilj se može postidi ako se na svaku lokalnu mrežu postavi most i parovi mostova povežu linijama od tačke do tačke (npr. linijama iznajmljenim od telefonske kompanije). Takav jednostavan sistem s tri lokalne mreže prikazanje na slici 4- 45. U njemu se primenjuju uobičajeni algoritmi za usmeravanje. To čete najlakše uočiti ako linije od tačke do tačke posmatrate kao lokalne mreže u kojima nema raču- nara. Tako ćemo dobiti normalan sistem sa šest mreža, povezan mostovima. Ništa što smo do sada naučili ne obavezuje lokalnu mrežu da sadrži računare.



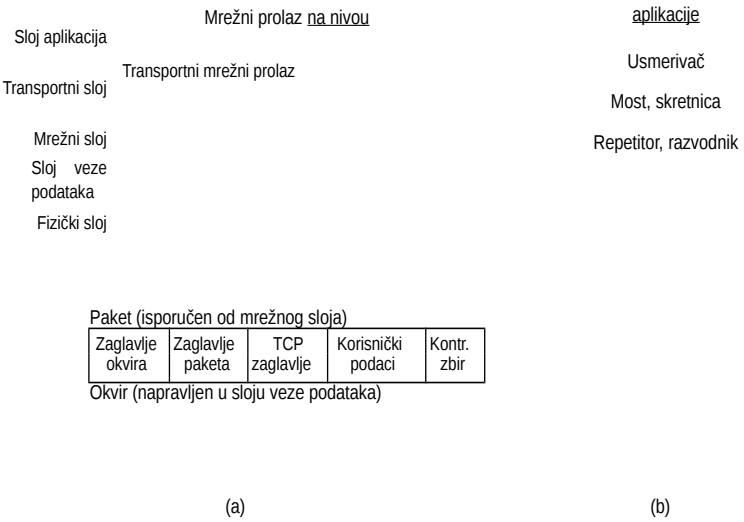
Slika 4-45. Za povezivanje udaljenih lokalnih mreža mogu se upotrebid daljinski mostovi (engl. *remote bridges).*

Na linijama od tačke do tačke mogu se koristiti različiti protokoli. Jedna mogućnost je da se izabere neki standardni protokol sloja veze tipa od tačke do tačke, kao što je PPP, koji će kompletne MAC okvire smeštati u polje s korisničkim podacima. Ta strategija radi najbolje ako su sve lokalne mreže identične, a ostaje jedino pitanje kako isporučiti okvir pravoj lokalnoj mreži. Druga mogućnost je da se na izvorišnom mostu iz okvira uklone zaglavlje i završni blok MAC sloja i da se ono što ostane smesti u polje s korisničkim podacima protokola od tačke do tačke. Okviru se na odredišnom mostu ponovo dodaju zaglavlje i završni blok MAC sloja. Nezgodno je to što kontrolni zbir koji stigne odredišnom računani nije kontrolni zbir koji je izračunao izvorišni računar, tako da se greške izazvane obradom u mostu ne mogu otkriti.

4.7.5 Repetitori, razvodnici, mostovi, skretnice, usmerivači, mrežni prolazi

Do sada smo u ovoj knjizi upoznali različite načine prenošenja okvira i paketa s jednog segmenta kabla na drugi. Pominjali smo repetitore, mostove, skretnice, raz- vodnike, usmerivače i mrežne prolaze. Svi se oni široko koriste, ali se ipak razlikuju ne samo u pojedinostima, već i u važnijim osobinama. Pošto ih ima tako mnogo, nije loše da ih sve uporedimo najednom mestu i istaknemo njihove sličnosti i razlike.

Navedeni uređaji, na prvom mestu, rade u različitim slojevima, kao što prikazuje slika 4-46(a). Sloj u kome rade je važan jer različiti uređaji za usmeravanje koriste različite podatke. Situacija je najčešće takva da korisnik generiše podatke koje treba poslati udaljenom računaru. Podaci se predaju transportnom sloju koji im dodaje, npr. TCP zaglavlje, a zatim sve zajedno prosleđuje mrežnom sloju. Mrežni sloj im dodaje svoje zaglavlje formirajući paket mrežnog sloja, npr. IP paket. IP paket je na slici 4-46(b) prikazan sivom bojom. Paket tad odlazi u sloj veze podataka koji mu dodaje svoje zaglavlje i kontrolni zbir (CRC), i rezultujuci okvir predaje fizičkom sloju za slanje, npr. preko lokalne mreže.



Slika 4-46. (a) Uređaji po slojevima, (b) Okviri, paketi, zaglavlja.

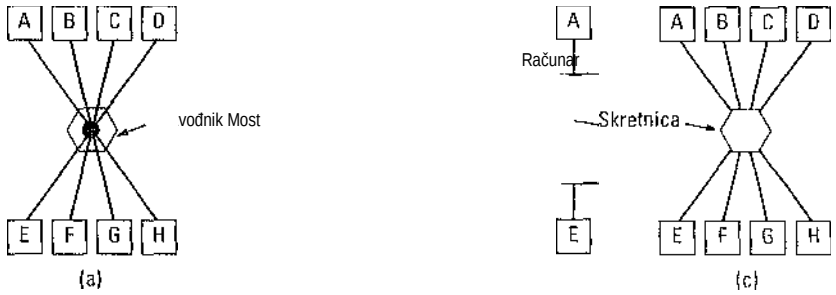
Vratimo se sada uređajima za usmeravanje i pogledajmo njihov postupak s paketima i okvirima. Na samom dnu, u fizičkom sloju, nalazimo repetitore. To su analogni uređaji koji povezuju dva segmenta kabla. Signal s jednog segmenta pojačava se i prenosi na drugi segment. Repetitori ne razlikuju okvire, pakete ili zaglavlja. Njih zanima samo električni potencijal. Klasični Ethernet, na primer, podržava četiri repeti- tora da bi se dužina kabla mogla povećati sa 500 na 2500 metara.

Dolazimo do razvodnika. U razvodniku se električno spaja više ulaznih linija. Okviri koji stižu jednom od linija šalju se u sve druge linije. Ako dva okvira stignu istovremeno, sukobiće se baš kao u koaksijalnom kablu. Drugim recima, čitav razvodnik je jedinstven domen sukobljavanja. Sve linije koje se stiču u razvodniku moraju da rade istom brzinom. Razvodnici se razlikuju od repetitora po tome što (obično) ne pojačavaju ulazne signale i sadrže više linijskih kartica, od kojih svaka ima više ulaza, ali te razlike nisu prevelike. Slično repetitorima, ni razvodnici ne pro- veravaju adrese formata 802, niti ih na bilo koji način koriste. Razvodnik je prikazan na slici 4-47(a).

Pređimo sada u sloj veze podataka, gde se nalaze mostovi i skretnice. S mostovima smo se upravo površno upoznali. Most povezuje dve ili više mreža, kao na slici 4-47(b). Kada stigne okvir, softver u mostu izvlači odredišnu adresu iz zaglavlja okvira i upo- reduje je sa svojom tabelom da bi utvrdio gde da uputi okvir. Za Ethernet je ta adresa 48-bitna (slika 4-17). Slično

razvodniku, savremeni most ima linijske kartice, obično sa po četiri ili osam ulaza određenog tipa. Linijska kartica za Ethernet ne može da obradi, recimo, okvir formata token ring jer ne zna gde se u zaglavlju okvira nalazi odredišna

adresa. Međutim, most može da sadrži linijske kartice za različite vrste mreža i različite brzine prenosa. Za razliku od razvodnika, kada se upotrebi most, svaka linija postaje zaseban domen sukobljavanja.



Slika 4-47. (a) Razvođnik. (b) Most. (c) Skretnica.

Skretnice liče na mostove jer i one usmeravaju okvire na osnovu adresa u njihovom zaglavlju. Mnogi smatraju da su to u suštini identični uređaji. Glavna razlika između njih je u tome što se skretnica obično koristi za povezivanje pojedinačnih računara, kao na slici 4-47(c). Da je na mestu skretnice na slici 4-47(c) most, pa računar *A* pošalje okvir računani *B,* most bi takav okvir automatski odbacio. Skretnica bi, međutim, aktivno uputila okvir računani *B,* jer je to jedina putanja između A i *B.* Pošto se svaki priključak skretnice obično vezuje za samo jedan računar, skretnica mora imati prostora za mnogo više linijskih kartica nego most koji je namenjen za povezivanje čitavih lokalnih mreža. Svaka linijska kartica ima bafer za okvire koji stižu na njene priključke. Pošto svaki priključak predstavlja zaseban domen sukobljavanja, u skretnici je sukobljavanje nemoguće. Međutim, ako okviri pristižu brže nego što se mogu slati, baferi se mogu prepuniti i skretnica će početi da gubi okvire.

Da bi predupredile takvu situaciju, savremene skretnice počinju da prosleđuju okvir čim iz njegovog zaglavlja prime polje sa odredišnom adresom - pre nego što pristigne ostatak okvira (naravno, ako je slobodna izlazna linija). Te, tzv. skretnice za komuti- ranje ne koriste tehniku „čuvaj i prosledi“. One se ponekada zovu **prolazne skretnice** (engl. *cut-through switch*). Takve skretnice najčešće rade potpuno hardverski, dok mostovi obično sadrže mikroprocesor koji softverski izvršava algoritam „čuvaj i prosledi“. Međutim, pošto savremeni mostovi i skretnice sadrže specijalna integrisana kola za komutiranje, razlike između njih više su marketinške, nego tehničke prirode.

Do sada smo upoznali repetitore i razvodnike, koji su prilično slični, kao i mostove i skretnice, koji su takođe međusobno veoma slični. Sada prelazimo na usmerivače, koji su nešto sasvim drago. Kada okvir stigne u usmerivač, uklanjaju se njegovo zaglavlje i završni blok, i paket se iz polja za korisničke podatke (zasenčeno na slici 4-46) predaje usmerivačkom softvera. Softver bira izlaznu liniju na osnovu zaglavlja paketa. Umesto 48-bitne adrese formata 802, zaglavlje jednog IP paketa sadrži 32-bitnu (IPv4) ili 128-bitnu (IPv6) adresu. Usmerivački softver nikada ne dolazi u priliku da vidi adrese iz okvira, čak i ne zna da lije paket došao iz lokalne mreže ili preko linije od tačke do tačke. Usmerivače i usmeravanje proučićemo detaljno u 5. poglavlju.

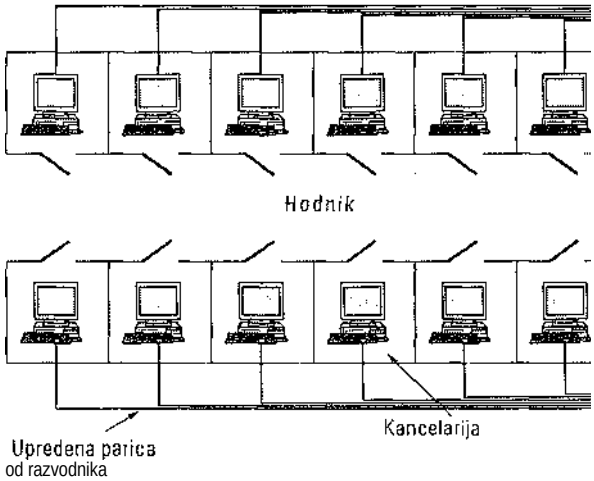
U sledecem višem sloju nalazimo transportne mrežne prolaze (engl. *transport gateways)*. Oni spajaju dva računara koji koriste različite vrste transportnih protokola sa uspostavljanjem direktne veze. Pretpostavimo, na primer, da računar s protokolom TCP/IP želi da komunicira s računarom koji koristi transportni protokol ATM. Transportni mrežni prolaz tada može da kopira pakete s jedne veze na drugu i da im usput menja format, ukoliko je to potrebno.

Na kraju, mrežni prolaz koji radi na nivou aplikacije razume format i sadržinu podataka, tako da prevodi pomke iz jednog formata u dragi. Na primer, mrežni prolaz za e-poštu može da prevodi Internet poruke u SMS poroke za mobilne telefone.

4.7.6 Virtuelne lokalne mreže

U doba prve pojave lokalnih rneža, debeli žuti kablovi vijugali su kroz kablovode mnogih poslovnih zgrada. Na njih je priključivan svaki računar pored koga su prošli. Često je više kablova spajano okosnicom (slika 4-39) ili centralnim razvodnikom. Nije se vodilo računa o tome koji je računar na kojoj mreži. Svi zaposleni iz bliskih prostorija povezivani su u istu lokalnu mrežu, ma čime da su se bavili. Fizički raspored je trijumfovao nad logikom.

Kada su se devedesetih godina pojavili sistem 10Base-T i razvodnici, sve se izme- nilo. Zgrade su ponovo ožičavane (uz znatne troškove) da bi se oslobodile žutih „baštenskih creva“ i snabdele upredenim paricama koje su vodile od svake kancelarije do centralnih razvodnih ormana u dnu svakog hodnika ili do računskog centra firme (slika 4-48). U situacijama kada je izvođač imao trenutke nadahnuća, uvođene su parice 5. kategorije. U ostalim, prizemnijim situacijama, iskorišćene su postojeće telefonske parice (3. kategorije), da bi samo posle nekoliko godina - kada se pojavio brzi Ethernet - ponovo bile zamenjene.



Slika 4-48. Zgrada s centralizovanim ožičenjem, razvodnicima i skretnicama.

Uz Ethernet, komutiran pomoću razvodnika (a kasnije i pomoću skretnica), lokalna mreža se često mogla projektovati logičnije. Ako je kompanija želela *k* lokalnih mreža, kupovala je *k* razvodnika. Promišljenim priključivanjem kablova u pojedine razvodnike mogle su se ostvariti lokalne mreže korisnika sa srodnim poslovima. Naravno, ako dve osobe iz istog organizacionog odeljenja rade u različitim zgradama, najverovatnije će se priključiti na

različite razvodnike i tako biti u različitim lokalnim mrežama. Pa ipak, i uz takve izuzetke postignuta je mnogo bolja situacija od one kada je pripadnost lokalnoj mreži bila dirigovana samo fizičkom lokacijom korisnika.

Kakve veze ima koje na kojoj mreži? Na kraju krajeva, lokalne mreže su u skoro svim organizacijama međusobno povezane. Pa ipak, odgovor glasi: to često ima veze. Administratori mreža iz mnogih razloga više vole da svrstavaju korisnike u lokalne mreže na osnovu organizacione strukture, nego na osnovu njihove fizičke lokacije. Na prvom mestu, to je pitanje bezbednosti. Poznato je da se svaki mrežni interfejs može prebaciti u promiskuitetni režim rada, kada kopira sve što nailazi kablom. Zaposleni u mnogim ođeljenjima (istraživačkom, patentnom, obračunskom) ne bi želeli da interne informacije iscure izvan odeljenja. U sličnim situacijama ima smisla da se svi pripadnici takvog odeljenja stave na istu lokalnu mrežu iz koje će se onemogućiti slanje podataka. Uprava nikada ne želi da čuje da je takvo nešto teško ostvariti, osim ako se sve osoblje takvog odeljenja smesti u susedne prostorije u koje je onemogućen pristup uljezima.

Zatim dolazi pitanje opterećenosti mreže. U nekim lokalnim mrežama saobraćaj je gušći nego u drugima, pa ih je ponekad poželjno odvojiti od ostatka mreže. Na primer, ako se u istraživačkom odeljenju izvode opsežni eksperimenti koji znaju da izmaknu kontroli i tada zasite lokalnu mrežu, oni iz obračunskog odeljenja neće baš oduševljeno prihvatiti da istraživačima pozajme deo svog propusnog opsega.

Potom dolazi pitanje neusmerenog emitovanja. Ono je podržano u većini lokalnih mreža i mnogi protokoli viših slojeva intenzivno ga koriste. Na primer, ako korisnik želi da pošalje paket na IP adresu x, koju će MAC adresu staviti u okvir? To ćemo detaljno proučiti u .5. poglavlju, a zasad prihvatite da će on difuzno emitovati okvir s pitanjem: „Ko je vlasnik IP adrese *xT*\ a zatim čekati odgovor. Postoji još mnogo primera korišćenja neusmerenog emitovanja. Kako raste broj međusobno povezanih lokalnih mreža, neusmereni saobraćaj koji stiže do svakog računara povećava se približno linearno s brojem umreženih računara.

Situacija bliska prethodnoj nastaje i kada neispravna mrežna kartica počne da na sve strane emituje beskonačan tok okvira. Rezultat je **neusmerena** bujica (engl. *broadcast stotm)* koja potpuno plavi kapacitet lokalne mreže i bezumno tera sve racu- nare na povezanim lokalnim mrežama da neprekidno obrađuju i odbacuju primljene neusmerene okvire.

Na prvi pogled se čini da se neusmerena bujica može ograničiti na uže područje ako se lokalne mreže razdvoje mostovima ili skretnicama, ali ako se želi transparentan (ne-primetan) rad takvih uređaja (tj. da se računar može bez ikakvih posledica premeštati iz jedne lokalne mreže u dragu), mostovi moraju da propuštaju neusmerene okvire.

Pošto smo upoznali neke razloge zbog kojih kompanije žele da pojedine njihove lokalne mreže budu ograničeno dostupne, vratimo se na problem razdvajanja logičke i fizičke topologije mreže. Pretpostavimo daje korisnik premešten iz jednog u drugo odeljenje iste organizacije, a da pri tome nije napustio svoju kancelariju, ili daje samo promenio kancelariju ostajući na istom radnom mestu. Premeštanje korisnika iz jedne lokalne mreže u drugu kada su mreže spojene razvodnicima, znači da administrator mreže treba da u razvodnom ormanu premesti priključak korisnikovog računara iz jednog razvodnika u drugi.

U mnogim organizacijama stalno dolazi do organizacionih pramena, što znači da će administrator mreže svaki čas čupati i ponovo spajati kablove. Isto tako, izmena se u nekim situacijama neće moći izvesti zbog prevelike udaljenosti korisnikovog računara od odgovarajućeg razvodnika (npr. korisnik se premestio 11 drugu zgradu).

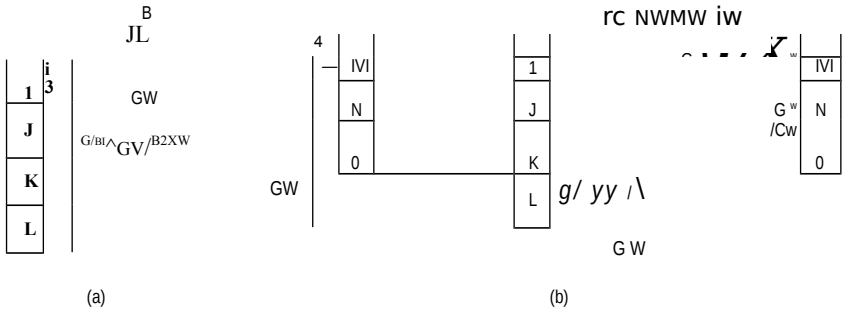
Kao odgovor na korisničke zahteve u pogledu veće fleksibilnosti, projektanti mreža su

počeli da traže način da čitave zgrade ponovo ožiče softverskim putem. Rezultat njihovih napora bila je virtuelna lokalna mreža (engl. *Virtual LAN, VLAN)* koju je čak standardizovao komitet za mrežu 802. Ona se sada primenjuje u mnogim organizacijama, pa ćemo je zato razmotriti. Dodatne informacije o virtuelnim lokalnim mrežama potražite kod Breyera i Rileya (1999) i Seiferta (2000).

Virtuelne lokalne mreže zasnivaju se na specijalno projektovanim skretnicama koje „reaguju" na takvu mrežu, iako na periferiji mogu imati i razvodnike (slika 4-48). Kada treba da uspostavi sistem zasnovan na virtuelnim lokalnim mrežama, administrator mreže najpre odlučuje o broju VLAN mreža, o računalima koji će biti na svakoj VLAN mreži, kao i o nazivima VLAN mreža. Virtuelne lokalne mreže često (neformalno) dobijaju imena boja, pošto se na taj način lako razlikuju na planovima: članovi crvene mreže su crveni, zelene mreže - zeleni itd. Na istom planu se tada mogu istovremeno videti i fizički i logički raspored mreže.

Primera radi, razmotrimo četiri lokalne mreže sa slike 4-49(a), na kojoj osam računara pripada „sivoj" virtuelnoj lokalnoj mreži G (engl. *grey),* a sedam „beloj" virtuel- noj lokalnoj mreži W (engl. *white).* Četiri fizičke lokalne mreže povezane su pomoću dva mosta, *BI* i *B2.* Ako se koristi centralizovano ožičenje upredenom paricom, mogu se pojaviti i četiri razvodnika (nisu prikazani); spojni kabl i razvodnik su u logičkom smislu ista stvar, a s kablovima je slika jednostavnija. Isto tako, izraz „most" danas se uglavnom koristi kada na istom priključku ima više računara (kao na slici), a inače su „most" i „skretnica" logički identični. Slika 4-49(b) prikazuje iste računare i iste virtuelne mreže povezane skretnicama kod kojih na svaki priključak dolazi samo jedan računar.

Da bi virtuelne lokalne mreže radile ispravno, u mostovima ili skretnicama moraju se uspostaviti odgovarajuće konfiguracione tabele. Tabele sadrže podatke 0 tome kojoj se virtuelnoj mreži može pristupiti preko kog priključka (linije). Kada stigne okvir, recimo, sa sive virtuelne mreže, on se mora proslediti svim priključcima označenim sa G. To važi za običan (to jest usmeren) saobraćaj, kao i za višesmeran, to jest neusmeren saobraćaj.



Slika 4-49. (a) Četiri fizičke lokalne mreže organizovane u dve virtuelne lokalne mreže, sivu (G) i belu (W), pomoću dva mosta (BI i B2). (b) Istih 15 računara organizovanih u dve virtuelne lokalne mreže pomoću dve skretnice (SI i S2).

Obratite pažnju na to da se isti priključak može označiti s više VLAN boja. To se najbolje vidi na slici 4-49(a). Pretpostavimo da računar *A* pošalje okvir neusmereno. Most *BI* prima okvir, utvrđuje daje stigao sa sive virtuelne mreže i zato ga šalje svim G priključcima (osim onom s koga je stigao). Pošto most *BI* ima samo dva druga priključka i oba su označena sa G, on šalje okvir i jednom i drugom priključku.

Na mostu *B2* situacija je drugačija. Ovde most zna da nema sivih računara na lokalnoj mreži 4, pa tamo i ne upućuje okvir, već samo u mrežu 2. Kada se korisnik mreže 4 premesti u drugo odeljenje, tj. pređe na sivu mrežu, moraju se ažurirati tabele mosta *B2* i priključak W označiti sa GW. Ako „posivi“ računar *F,* tada se priključak GW za mrežu 2 mora označiti sa G.

Zamislimo sada da svi računari s mreža 2 i 4 prelaze na sivu virtuelnu mrežu. Tada će priključci mosta *B2* za mreže 2 i 4 dobiti oznaku G, ali će i priključak mosta *BI* ka mostu *B2* takođe izmeniti oznaku iz GW u G pošto beli okviri koji mostu *BI* stižu s mreža 1 i 3 ne moraju više da se prosleđuju mostu *B2.* Isto važi za sliku 4-49(b), samo su svi priključci koji povezuju računare sa skretnicama jednobojni jer jedan računar može biti samo u jednoj virtuelnoj mreži.

U primeru smo prećutno prihvatili da mostovi i skretnice nekako znaju koje je boje dolazni okvir. Zaista, kako je oni prepoznaju? Za to postoje tri metode:

1. VLAN boja se dodeljuje svakom priključku.
2. VLAN boj se dodeljuje svakoj MAC adresi.
3. VLAN boja se dodeljuje svakom protokolu sloja 3 ili svakoj IP adresi.

Po prvoj metodi, svaki priključak se označava jednom VLAN bojom. Međutim, ova metoda radi samo ako svi računari na istom priključku pripadaju istoj virtuelnoj mreži. Na slici 4-49(a), to važi za priključak mosta *BI* ka mreži 3, ali ne i za njegov priključak ka mreži 1.

Prema drugoj metodi, most ili skretnica imaju tabelu sa spiskom 48-bitnih MAC adresa svakog računara s kojim su povezani i virtuelnih mreža na kojima se nalaze.

Tada je moguće na istoj fizičkoj lokalnoj mreži imati računare iz više virtuelnih mreža, kao u lokalnoj meži 1 sa slike 4-49(a). Kada okvir stigne, most ili skretnica treba samo da uporede njegovu MAC adresu sa svojom tabelom da bi utvrdili s Icoje vir- tuelne mreže dolazi.

Trećom metodom most ili skretnica ispituju polje okvira u kome su korisnički podaci da bi, na primer, sve IP računare svrstali u jednu virtuelnu mrežu, a sve Ap- pleTalk računare u drugu. U prvoj od dve pomenute mreže, prema IP adresi se mogu i identifilcovati pojedini računari. Ova strategija je najkorisnija kada su mnogi računali prenosivi i stalno menjaju priključnu stanicu. Svaka priključna stanica ima i svoju MAC adresu, ali to što je znamo ne znači da znamo i virtuelnu mrežu prenosivog računara koji se u njoj trenutno nalazi.

Metoda, međutim, narušava osnovno pravilo rada u mreži: nezavisnost slojeva. Sloj veze podataka ne treba da gura svoj nos u polje s korisničkim podacima. On ne treba da zna šta se tamo nalazi, pogotovo ne bi trebalo da išta zaključuje na osnovu nađenog sadržaja. Kada se primeni opisana metoda, tada izmena protokola sloja 3 (npr. prelazak sa IPv4 na IPv6) dovodi do otkazivanja skretnica. Nažalost, danas se na tržištu nalaze takve skretnice.

Nema, naravno, ničeg lošeg u tome što se usmeravanje zasniva na IP adresama - skoro celo 5. poglavlje obrađuje tu temu - ali mešati slojeve znači prizivati nevolju. Proizvođač skretnica bi odbacio ovaj argument rekavši da njegove skretnice razume- ju oba protokola: i IPv4 i IPv6, te je, dakle, sve u redu. Ali šta će se dogoditi kada se pojavi IPv7? Proizvođač bi verovatno odgovorio: „Pa, mogao bi već jednom da kupiš i novu skretnicu“.

IEEE standard 802.1Q

Nakon malo više razmišljanja o ovoj problematici, utvrđujemo da je važnije znati VLAN samog okvira, nego VLAN računara s koga potiče. Kada bi se VLAN mreža mogla označiti u zaglavlju okvira, otpala bi potreba za zavirivanjem u polje s korisničkim podacima. Za nove mreže, kao što su 802.11 i 802.16, ništa nije lakše nego dodati u zaglavlje i VLAN polje. U stvari, *Identifikator veze* u mreži 802.16 po smislu je blizak identifikatoru VLAN mreže. Ali, šta raditi sa Ethemetom koji preovladava među lokalnim mrežama i koji nema rezervnih polja za smeštanje VLAN identifikatora?

IEEE komitet za mrežu 802 pozabavio se ovim problemom 1995. godine. Posle mnogo rasprave, sproveo je ono što je do tada smatrano nezamislivim: izmenio je zaglavlje Ethernet okvira. Nov format, objavljen 1998, pojavio se kao IEEE standard 802.1Q. Taj format sadrži VLAN oznaku koju ćemo ubrzo objasniti. Izmena tako široko prihvaćenog standarda ne može da prođe bezbolno, pa nam se odmah nameće nekoliko pitanja:

1. Treba li da bacimo stotine miliona postojećih Ethernet kartica?
2. Ako to ne moramo, ko će generisati nova polja?
3. Šta će se dogoditi sa okvirima koji su već maksimalne veličine?

Naravno, IEEE komitet za mrežu 802 i sam je bio dovoljno svestan pomenutih problema, pa je predložio nekoliko rešenja.

Pri tome je najhitnije shvatiti da se VLAN poljima ne služe računali, već ih stvarno koriste samo mostovi i skretnice. Tako, na slici 4-49 nije stvarno važno da VLAN polja postoje na linijama koje vode ka krajnjim korisnicima sve dok se nalaze na linijama između mostova ili skretnica. Da bi se služili sistemom VLAN mreža, mostovi ili skretnice moraju da ih prepoznaju, ali takav zahtev već postoji. Sada samo dodatno uvodimo i to da moraju da prepoznaju i standard 802.1Q, što noviji uređaji i čine.

Da li treba baciti postojeće Ethernet kartice? Definitivan odgovor je: ne. Setite se da

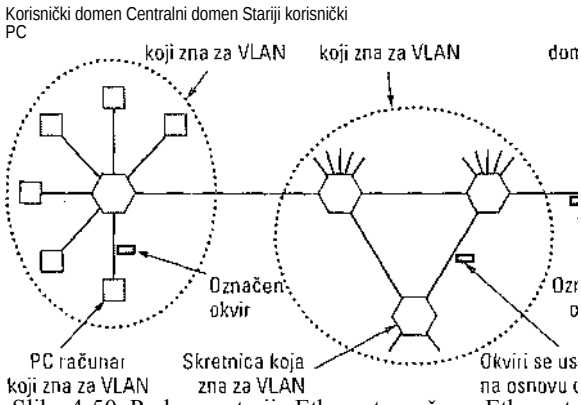
komitet za mrežu 802.3 nije zahtevao od korisnika čak ni da polje *Tip* promene u polje *Dužina*. I sami možete zamisliti reakciju korisnika na preporuku da bace postojeće Ethernet kartice. Međutim, s pojavom novih Ethernet kartica raste i nada da će one biti usaglašene sa standardom 802.1Q i ispravno popunjavati VLAN polja okvira.

Elem, ako začetnik veze ne generiše VLAN polja, ko to treba da uradi? Odgovor je jednostavan: polje će biti dodato okviru na prvom mostu ili skretnici koji znaju šta je to VLAN polje, a poslednji takav most ili skretnica ukloniće ga s njega. Međutim, kako će most ili skretnica znati koji okvir pripada kom VLAN-u? Pa evo, prvi most ili skretnica može priključku da dodeli VLAN broj, da pogleda MAC adresu ili (ne daj bože) da ispita korisničke podatke. Dogod sve Ethernet kartice poštuju standard 802.1Q, nalazimo se tamo odakle smo pošli. Stvarno se uzdamo u to da će sve kartice za gigabitni Ethernet od samog početka poštovati taj standard i da će tokom vremena, kako korisnici budu unapređivali svoje sisteme do gigabitnog Etherneta, standard 802.1Q biti automatski uveden. Što se tiče okvira dužih od 1518 bajtova, naglasimo da je standard 802.1Q već pomerio tu granicu na 1522 bajta.

U prelaznom periodu, mnoge instalacije će istovremeno sadržati starije sklopove (najčešće klasični ili brzi Ethernet) koji ne znaju za virtuelne mreže i nove (najčešće gigabitni Ethernet) koji znaju za njih. Ta situacija je ilustrovana slikom 4-50 gde za- senčeni uređaji ne znaju za virtuelne mreže, dok nezasenčeni uređaji znaju za njih. Jednostavnosti radi, pretpostavili smo da sve skretnice prepoznaju virtuelne mreže. Ako to nije slučaj, prva stanica koja zna za virtuelne mreže može okvirima da doda oznake zasnovane na MAC ili IP adresama.

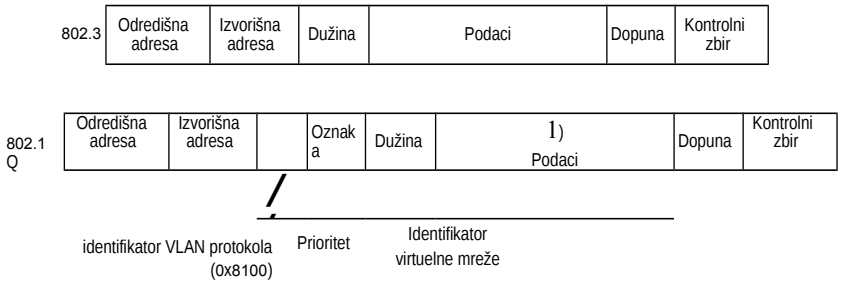
Na ovoj slici, Ethernet kartice koje znaju za virtuelne mreže direktno generišu označene okvire (tj. okvire prema standardu 802.1Q) koji se dalje usmeravaju na osnovu tih oznaka. Da bi ispravno usmeravale okvire, skretnice - baš kao i ranije - moraju da znaju kojoj se virtuelnoj mreži može pristupiti preko kog priključka. To što okvir pripada sivoj virtuelnoj mreži ne znači ništa ako skretnica ne zna priključak preko koga su povezani računari iz sive mreže. Zato skretnica ima tabelu uređenu prema virtuelnim mrežama, sa uporednim podacima o odgovarajućim priključcima i informacijom da li su u pitanju računari koji znaju za VLAN ili ne.

Kada stariji PC računar pošalje okvir skretnici koja zna za VLAN, skretnica formira nov okvir označen na osnovu svog poznavanja pošiljaočeve virtuelne mreže (priključka, MAC adrese ili IP adrese). Nadalje više nije važno da li je pošiljaočev računar zastareo. Slično tome, skretnica koja želi da isporuči označen okvir starijem računaru mora da ga prebaci u stari format pre nego što ga isporuči.



Slika 4-50. Prelaz sa starije Ethernet mreže na Ethernet koji zna za virtuelne mreže.
Zasenčeni uređaji znaju za VLAN mreže. Nezasenčeni ih ne poznaju.

Pogledajmo sada kako izgleda format okvira po standardu 802.1Q (slika 4-51). Jedinu razliku u odnosu na stari format čine nekoliko 2-bajtnih polja. Prvo je *Identifi- kator VLAN protokola* i uvek ima vrednost 0x8100. Postoje taj broj veći od 1500, sve Ethernet kartice ga tumače kao tip, a ne kao dužinu. Pitanje kako bi kartice starog tipa protumačile ovo polje, ostače otvoreno jer se takav okvir njima i ne šalje.



Slika 4-51. Stari (802.3) i novi (802,1Q) format Ethernet okvira.

Drugo 2-bajtno polje sadrži tri potpolja. Glavno je *Identifikator virtuelne mreže,* koje zauzima 12 najmanje značajnih bitova. Oko toga je cela priča - kojoj virtuelnoj mreži pripada okvir? Trobitno polje *Prioritet* nema nikakve veze s virtuelnim mrežama. Međutim, kada se jednom u deset godina odlučite da promenite Ethernet zaglavlje i pri tome angažujete mnogo ljudi, zašto da tu priliku ne iskoristite i za drage stvari? Pomoću tog polja mogu se razlikovati gust i redak saobraćaj u realnom vremenu i saobraćaj koji nije vezan za vreme, i tako obezbediti bolji kvalitet usluga u Ethernetu. On je potreban za govorni saobraćaj koji se odvija Ethernetom (iako, istini za volju, IP već četvrt veka ima slično polje koje niko nikada nije iskoristio).

Poslednji bit, *IKF (Indikator kanoničkog formata),* trebalo je nazvati *IKS (Indikator korporacijskog samoljublja*). Prvobitno je bio namenjen da označi formate MAC adresa (little-endian ili big-endian), ali se ta upotreba izgubila u brojnim raspravama.
Njegovo prisustvo sada znači da korisnički podaci sadrže „duboko zamrznut“ okvir formata

802.5 koji se - dok putuje Ethernetom - nada da će na odredištu pronaći dra gu lokalnu 802.5 mrežu. Ceo taj aranžman, naravno, nema nikakve veze s virtuelnim lokalnim mrežama. Međutim politika komiteta za standardizaciju prilično liči na politički marketing koji stalno gledamo: ako ti glasaš za moj bit, i ja ću glasati za tvoj.

Kao što smo već naveli, kada označen okvir stigne na skretnicu koja zna za VLAN, skretnica na osnovu Indikatora virtuelne mreže iz okvira u svojoj tabeli pronalazi priključak na koji ga treba poslati. Ali, otkud ta tabela? Ako nju treba praviti ručno, onda smo baš na nuli: ručno konfigurisanje mostova. Sva pogodnost korišćenja nevidljivih mostova leži u principu „uključi i koristi“ bez ikakvog ručnog podešavanja. Ne bi valjalo izneveriti taj princip. Na sreću, mostovi koji znaju za virtuelne mreže mogu sami da se konfigurišu na osnovu oznaka u okvirima koji kroz njih prolaze. Ako okvir označen kao VLAN 4 stigne na priključak 3, to znači da bi se računar na priključku 3 mogao nalaziti na virtuelnoj mreži 4. Standard 802.1Q objašnjava kako se dinamički prave tabele za usmeravanje, oslanjajući se uglavnom na odgovarajuće delove algoritma koji je razvila Radia Perlman, obuhvaćenog standardom 802.1D.

Pre nego što završimo temu usmeravanja u virtuelnim lokalnim mrežama, treba još nešto naglastiti. Mnogi iz sveta Interneta i Etherneta fanatični su pobornici rada u mreži bez uspostavljanja direktne veze i suprotstavljaju se svemu što iole miriše na povezivanje u sloju veze podataka ili u mrežnom sloju. Ipak, virtuelne lokalne mreže uvode nešto što iznenađujuće liči na povezivanje. Da bi se virtuelne lokalne mreže mogle koristiti kako je predviđeno, svaki okvir mora da nosi specijalan nov identifi- kator koji se koristi kao indeks u tabeli unutar skretnice za pronalaženje odredišta okvira. Upravo se tako nešto događa i u mrežama sa uspostavljanjem direktne veze. U mrežama u kojima se veza ne uspostavlja direktno, usmeravanje se vrši preko odredišne adrese, a ne pomoću nekakvog identifikatora veze. U 5. poglavlju ćemo još raspravljati o ovom ekscentričnom „konekcionizmu“.

4.8 SAŽETAK

U nekim mrežama postoji samo jedan kanal kroz koji se odvija sva komunikacija. U njima je glavni problem dodeljivanje kanala jednoj od konkurentskih stanica koje .žele da ga koriste. Za to postoje brojni algoritmi, a najvažnije metode su prikazane na slici 4-52.

Kanal se najjednostavnije dodeljuje pomoću tehnika FDM i TDM. One su efikasne kada je broj stanica mali i konstantan, a saobraćaj ujednačen. Obe se u takvim uslo- vima. široko koriste, na primer, za dodeljivanje propusnog opsega telefonskih vodova.

Kada je broj stanica veliki i promenljiv ili se saobraćaj odvija u rafalima, FDM i TDM predstavljaju loš izbor. Kao alternativa je predložen protokol ALOHA, u čistom vidu ili vremenski raspodeljen. Sistem ALOHA, njegove brojne varijante i modifikacije, često su analizirani i korišćeni u realnim sistemima.

Metoda	Opis
FDM	Namenjuje frekventno područje svakoj stanici
WDM	Dinamička FDM tehnika za optičko vlakno
TDM	Namenjuje vremenski interval za svaku stanicu
Čista ALOHA	Nesinhronizovan prenos u svakom trenutku
Vremenski raspodeljenaALOHA	Nasumično raspoređen prenos u dobro definisanim vremenskim intervalima
1-trajni CSMA	Standardni višekorisnički pristup uz osluškivanje nosioca podataka
Povremeni CSMA	Odustajanje tokom nasumično odabranog intervala vremena ako je kanal zauzet
P-trajni CSMA	CSMA, ali uz verovatnoću emitovanja p
CSMA/CD	CSMA, ali uz odustajanje ako dođe do sukoba
Zasnovana na mapi bitova	Ciklično dodeljivanje na osnovu mape bitova
Binarno odbrojavanje	Sledeća je stanica s najvišim brojem koja je spremna za emitovanje
Prolaz kroz binarno stablo	Proređivanje sukobljavanja sistemom selekcije
MACA, MACAW	Protokoli za bežične lokalne mreže
Ethernet	CSMA/CD uz binarno eksponencijalno odustajanje
FHSS	Skokovito frekventno širenje spektra
DSSS	Direktno sekvencijalno širenje spektra
CSMA/CA	Višekorisnički pristup uz osluškivanje nosioca podataka i izbegavanje sukobljavanja

Slika 4-52. Metode i sistemi za dodeljivanje zajedničkog kanala.

U situaciji kada se kanal može osluškivati, stanice mogu da odustanu od emito- vanja ako čuju da neko drugi već emituje. Ta tehnika osluškivanja nosioca podataka izrodila je različite protokole za lokalne i gradske mreže.

Klasa protokola koji potpuno isključuju sukobljavanje ili ga barem svode na znatno manju meni, dobro je poznata. Binarno odbrojavanje u potpunosti sprečava sukobljavanje. Protokol prolaska kroz binarno stablo smanjuje broj sukoba razvrstavajući stanice u dve razdvojene grupe: one kojima je dozvoljeno da emituju i one kojima to nije dozvoljeno. Stanice se na taj način uzastopno razvrstavaju sve dok ne ostane samo jedna stanica kojoj se dozvoljava da emituje.

U bežičnim mrežama takođe postoje svojevrsni problemi i rešenja za njih. Najveći problem stvaraju skrivene stanice, pa se ne može primeniti protokol CSMA. Jedan način rešavanja, pomoću protokola MACA i MACAW, usmeren je na stimulisanje prenosa oko odredišta kako bi protokol CSMA bolje radio. Koristi se i skolcovito frekventno i direktno sekvencijalno širenje spektra. IEEE standard 802.11 kombinuje protokole CSMA i MACAW u protokol C'SMA/CA.

Ethernet je najčešći standard za lokalne mreže. U njemu se za dodeljivanje kanala koristi protokol CSMA/CD. U starijim njegovim verzijama vijugao je kabl od jednog računara do drugog, ali se sada za povezivanje računara koriste upredene parice i raz- vodnici ili skretnice. Brzina prenosa se povećala od 10 Mb/s na 1 Gb/s i još raste.

Postale su uobičajene i bežične mreže, a najčešća je mreža 802.11. Njen fizički sloj je predviđen za pet vrsta prenosa, uključujući prenos infracrvenim talasima, različite tehnike širenja spektra i višekanalni sistem FDM. Ona može da radi s baznom stanicom u svakoj ćeliji, ali i bez nje. Protokol je varijanta protokola MACAW, uz oslu- škivanje virtuelnog nosioca.

Pojavljuju se i gradske bežične mreže. To su širokopojasni sistemi koji pomoću ra- diotalasa premošćuju poslednji kilometar telefonskih veza. U njima se koriste klasične uskopojasne tehnike modulacije. Tu je važan kvalitet usluge, a standard za mrežu 802.16 razvrstava ga u četiri klase: jednu - s konstantnom brzinom prenosa, dve - s promenljivom brzinom prenosa i jednu - najbolju moguću uslugu.

Sistem Bluetooth takođe radi bežično, ali je više namenjen radnom okruženju - za povezivanje brojnih periferijskih uređaja s računarom. On može da poveže i periferijske uređaje, npr. faks mašine, s mobilnim telefonima. Slično mreži 802.11, i Bluetooth koristi skokovito frekventno širenje spektra u ISM području. Zbog očekivanog nivoa šuma u mnogim okruženjima i potrebe za interaktivnim radom u realnom vremenu, u protokole su ugrađene složene tehnike ispravljanja grešaka u hodu.

Kada postoji mnogo lokalnih mreža, uvek se na kraju kao problem nametne njihovo međusobno povezivanje. Za to se koriste mostovi i skretnice. PnP mostovi rade u strukturi razgranatog stabla. Nov pristup povezivanju lokalnih mreža predstavljaju virtuelne lokalne mreže (VLAN); one razdvajaju logičku topologiju mreže od njene fizičke topologije. Razvijen je nov format (802.1Q) Ethernet okvira da bi organizacije lakše prihvatile virtuelne mreže.

ZADACI

1. Za zadatak upotrebite formulu iz ovog poglavlja, ali je prvo izvedite. Okviri za slanje stižu u slučajnim vremenskim intervalima na kanal brzine 100 Mb/s. Ako je kanal zauzet kad okvir stigne, on se smešta u red čekanja. Dužina okvira je raspoređena eksponencijalno, a srednja vrednost dužine je 10.000 bitova po okviru. Za svaku od sledećih brzina stizanja izračunajte kašnjenje okvira prosečne dužine, uzimajući u obzir vreme čekanja u redu i vreme prenosa.
   a) 90 okvira u sekundi.
   b) 900 okvira u sekundi.
   c) 9000 okvira u sekundi.
2. Grupa od *N* stanica deli kanal brzine 56 kb/s u čistom sistemu ALOHA. Svaka stanica emituje okvir od 1000 bitova prosečno jednom u 100 sekundi, čak i ako prethodni okvir još nije poslat (npr. ako stanice mogu da odlazne okvire smeštaju u bafer). Kolika je najveća vrednost *N1*
3. Uporedite kašnjenje u čistom i vremenski raspođeljenom sistemu ALOHA. Gde je kašnjenje manje? Obrazložite odgovor.
4. Deset hiljada stanica za rezervisanje avionskih karata konkurišu za jedan vremenski raspođeljen kanal sistema ALOHA. Prosečna stanica šalje 18 zahteva na sat. Vremenski interval je 125 |is. Koliko približno iznosi ukupno opterećenje kanala?
5. Velika populacija korisnika sistema ALOHA generiše u sekundi 50 zahteva za slanje kako novih, tako i ponovljenih okvira. Vreme je podeljeno u intervale od po 40 ms.
   a) Kolike su šanse za uspeh iz prvog pokušaja?
   b) Kolika je verovatnoća uspeha posle tačno *k* sukoba?

c) Koliki je očekivan broj pokušaja potrebnih za uspešno slanje okvira?

6. Ispitivanjem vremenski raspodeljenog kanala sistema ALOHA s beskonačno mnogo korisnika utvrđuje se da 10 procenata vremenskih intervala prolazi neiskorišćeno.
   a) Koliko je opterećenje kanala G?
   b) Koliki je protok podataka kroz kanal?
   c) Da li je kanal nedovoljno iskorišćen ili preopterećen?
7. U vremenski raspoređljenom sistemu ALOHA s beskonačno mnogo korisnika, pro- sečan broj intervala koje stanica propušta posle sukoba da bi ponovo pokušala da emituje iznosi 4. Prikažite grafički zavisnost kašnjenja od protoka podataka za ovaj sistem.
8. Koliko stanica *s* treba da čeka u najnepovoljnijem slučaju da bi počela da šalje okvir preko lokalne mreže u kojoj se koristi
   a) osnovni protokol zasnovan na mapi bitova?
   b) protokol Moka i Warda s permutovanjem brojeva virtuelnih stanica?
9. U lokalnoj mreži se koristi binarno odbrojavanje u verziji po Moku i Warđu. U odre- đenom trenutku, deset stanica ima virtuelne brojeve 8,2,4,5,1,7,3, 6,9 i 0. Sledeće tri stanice koje imaju okvire spremne za slanje jesu stanice 4, 3 i 9. Kako izgleda re- dosled brojeva virtuelnih stanica pošto tri navedene stanice pošalju svoje okvire?
10. Šesnaest stanica s brojevima od 1 do 16 nadmeću se za pristup zajedničkom kanalu koristeći prilagodljiv protokol prolaska kroz binarno stablo. Ako sve stanice čije adrese predstavljaju proste brojeve istovremeno budu spremne za slanje, koliko će jednobitnih intervala biti potrebno da se razreši sukobljavanje?
11. Skup 2" stanica koristi prilagodljiv protokol prolaska kroz binarno stablo za dodelji- vanje pristupa zajedničkom kablu. Dve od njih su u određenom trenutku spremne za slanje. Koliki je najmanji, najveći i prosečan broj intervala potreban za prolazak kroz stablo ako je 2" » 1 ?
12. U bežičnim lokalnim mrežama koje smo proučavali, umesto protokola CSMA/CD koriste se protokoli kao što je MACA. Pod kojim bi se uslovima, ako je to uopšte moguće, u njima mogao koristiti protokol CSMA/CD?
13. Koja zajednička svojstva imaju protokoli WDMA i GSM za pristupanje kanalu? Protokol GSM potražite u 2. poglavlju.
14. Šest stanica, *A* do *F,* međusobno komuniciraju pomoću protokola MACA. Da li su istovremeno moguća dva prenosa? Obrazložite odgovor.
15. U sedmospratnoj poslovnoj zgradi na svakom spratu ima 15 susednih kancelarija. U svakoj kancelariji, na prednjem zidu, nalazi se utičnica za terminal, tako da utičnice u vertikalnoj ravni obrazuju pravougaonu mrežu s horizontalnim i vertikalnim rast- ojanjem od 4 m između utičnica. Uz pretpostavku daje moguće pravolinijski spojiti bilo koje dve utičnice (horizontalno, vertikalno ili di jagonalno), koliko je potrebno metara kabla da bi se spojile sve utičnice ako se koristi
   a) konfiguracija zvezde s jedinstvenim usmerivačem u centru?
   b) mreža 802.3?
16. Kolika je brzina u bodovima standardnog Etherneta od 10 Mb/s?
17. Skicirajte Mančester kodiranje za tok bitova: 0001110101.
18. Skicirajte diferencijalno Mančester kodiranje za tok bitova iz prethodnog zadatka. Pretpostavite da je na početku na liniji nizak nivo signala.
19. U lokalnoj mreži dužine 1 km, brzine 10 Mb/s, koja radi s protokolom CSMA/CD (nije u pitanju mreža 802.3), brzina prostiranja signala iznosi 200 m/ps. U sistemu nisu dozvoljeni repetitori. Okviri s podacima dugački su 256 bitova, uključujući 32 bita zaglavlja, kontrolnog zbira i drugih sistemskih podataka. Prvi jednobitni interval posle uspešnog prenosarezervisan je za primaoca da pristupi kanalu i pošalje 32- -bitni okvir s

potvrdom. Kolika je efektivna brzina prenosa podataka, bez sistemskih bitova, pretpostavljajući da nema sukobljavanja.

20. Dve stanice povezane protokolom CSMA/CD pokušavaju da jedna drugoj pošalju dugačke datoteke (u više okvira). Posle svakog poslatog okvira one se nadmeću za kanal koristeći algoritam binarnog eksponencijalnog odustajanja. Kolika je verovat- noća da će se nadmetanje završiti u rundi k i koliki je prosečan broj rundi po jednom konkurentskom periodu?
21. Zamislite mrežu brzine 1 Gb/s na kablu od 1 km, koja radi uz protokol CSMA/CD i nema repetitore. Brzina prostiranja signala kablom iznosi 200.000 km/s. Kolika je najmanja veličina okvira?
22. IP paket koji treba preneti Ethernetom dugačak je 60 bajtova, zajedno sa svim zagla- vljima. Ako se ne koristi LLC, da li je Ethernet okvir potrebno dopuniti i ako jeste, s koliko bajtova?
23. Ethernet okviri moraju imati dužinu najmanje 64 bajta da bi pošiljalac još uvek slao podatke u trenutku kada eventualno dođe do sukobljavanja na drugom kraju kabla. Okvir u brzom Ethernetu ima istu minimalnu dužinu od 64 bajta, ali brzi Ethernet može da prenosi bitove deset puta brže. Kako je bilo moguće zadržati istu minimalnu dužinu okvira?
24. Za maksimalnu veličinu Ethernet okvira u nekim knjigama se, umesto 1500, navodi vrednost od 1518 bajtova. Da lije to pogrešno? Obrazložite odgovor.
25. Specifikacija 1000Base-SX nalaže da sistemski sat treba da radi na frekvenciji 1250 MHz, premda se od gigabitnog Etherneta očekuje da prenosi podatke brzinom najviše 1 Gb/s. Da li propisana veća brzina služi kao osiguranje rada sistema? Ako nije to u pitanju, šta je?
26. Koliko okvira u sekundi može da obradi gigabitni Ethernet? Razmislite dobro i uzmite u obzir sve odgovarajuće situacije. *Pomoć:* ključna je činjenica daje to *gigabitni* Ethernet.
27. Navedite dve mreže u kojima se okviri mogu pakovati tesno, jedan uz drugi. Zašto je to dobro?
28. Na slici 4-27 prikazane su četiri stanice: A, B, C i D. Šta mislite, koja je od poslednje dve stanice bliža stanici A i zašto?
29. Pretpostavite da se lokalnom mrežom 802.1b, brzine 11 Mb/s, prenose 64-bitni okviri, jedan uz drugi, rađio-kanalom u kome je učestalost grešaka 10^{-7}. Koliko se pro- sečno ošteti okvira u sekundi?
30. Mreža 802.16 ima kanal širine 20 MHz. Koliko se bitova u sekundi može slati pret- platničkoj stanici?
31. IEEE standard 802.16 podržava četiri klase usluga. Koja klasa usluga predstavlja najbolji izbor za slanje nekomprimovanog videa?
32. Ponudite dva razloga zbog kojih bi se u mreži koristio kod za ispravljanje grešaka, umesto da se greške samo otkrivaju i oštećeni okviri ponovo šalju.
33. Sa slike 4-35 zaključujemo da se Bluetooth uređaj može istovremeno nalaziti u dve elementarne mreže. Postoji li razlog da isti uređaj ne bude istovremeno glavni čvor u obe elementarne mreže?
34. Slika 4-25 prikazuje više protokola fizičkog sloja. Koji je od njih najsličniji protokolu fizičkog sloja sistema Bluetooth? Šta je najveća razlika između njih?
35. Bluetooth podržava dve vrste povezivanja glavnog i sporednog čvora. Kakve su to veze i čemu služi svaka od njih?
36. Signalni okviri u varijanti skokovitog frekventnog širenja spektra koja se koristi u mreži 802.11 sadrže i vreme boravka. Mislite li da analogni signalni okviri u sistemu Bluetooth takođe sadrže vreme boravka? Obrazložite odgovor.
37. Razmotrite međusobno povezane lokalne mreže sa slike 4-44. Pretpostavite da su

računari *a* i *b* na mreži 1, da je *c* na mreži 2, a *d* na mreži 8. Tabele ključeva u svim mostovima na početku su prazne i koristi se razgranato stablo sa slike 4-44(b). Pokažite kako se menjaju tabele ključeva različitih mostova posle svakog od navedenih događaja koji slede jedan za drugim:

a) *a* šalje ka *b*.
b) *c* šalje ka *a*.
c) *d* šalje ka *c*.
d) *d* prelazi na mrežu 6.
e) *d* šalje ka *a*.

38. Posledica korišćenja razgranatog stabla za prosleđivanje okvira u proširenoj lokalnoj mreži jeste i to da neki mostovi ne učestvuju u prosleđivanju. Identifikujte tri takva mosta na slici 4-44. Postoji li neki razlog da mostovi ipak ostanu u mreži, iako ne učestvuju u prosleđivanju okvira?
39. Zamislite da skretnica ima linijsku karticu sa četiri ulazne linije. Često se dešava da okvir koji je došao jednom od ovih linija iziđe drugom linijom sa iste kartice. Šta u takvoj situaciji može da preduzme projektant skretnice?
40. Skretnica koja je projektovana za brzi Ethernet ima osnovnu ploču brzine 10 Gb/s. Koliko okvira u sekundi može ploča da obradi u najnepovoljnijem slučaju?
41. Razmotrite mrežu sa slike 4-49(a). Ako računar *J* odjednom „pobeli“, treba li me- njati oznake? Ako treba, šta treba menjati?
42. Opišite ukratko razliku između skretnica koje „čuvaju i prosleđuju“ podatke i prolaznih skretnica.
43. Skretnice koje rade tehnikom „čuvaj i prosledi“ imaju prednost nad prolaznim skretnicama u pogledu rada sa oštećenim okvirima. Objasnite u čemu je njihova prednost.
44. Da bi virtuelne lokalne mreže radile, mostovi i skretnice moraju imati konfiguracio- ne tabele. Sta ako se u virtuelnim mrežama na slici 4-49(a), umesto spojnih kablova, upotrebe razvodnici? Da li i razvodnici moraju imati konfiguracione tabele? Zašto moraju, odnosno zašto ne moraju?
45. Skretnica koja se nalazi u starijem korisničkom domenu na desnom kraju slike 4-50 zna za VLAN. Da li bi se umesto nje na ovom mestu mogla upotrebiti skretnica starog tipa? Ako bi mogla, kako bi radila? Ako ne bi mogla, zašto?
46. Napišite program koji simulira ponašanje protokola CSMA/CD na Ethernetu kada tokom prenošenja okvira ima *N* stanica spremnih za emitovanje. Program treba da ispisuje vreme kada svaka stanica uspešno počne da šalje okvir. Pretpostavite da sistemski sat kuca u ritmu vremenskih intervala (svake 51,2 mikrosekunde) i da otkrivanje sukoba i slanje rafalnog šuma oduzima jedan vremenski interval. Svi okviri imaju maksimalno dozvoljenu dužinu.

5

MFieZM! SLOJ

Zadatak mrežnog sloja je da pakete od izvorišta sprovede celim putem do odredišta, što znači da treba da ih provuče kroz brojne usputne usmerivače. Taj zadatak jasno odudara od funkcije sloja veze - jednostavnog premeštanja paketa s jednog kraja žice na dragi. Mrežni sloj je najniži sloj koji se bavi prenosom od jednog do drugog kraja mreže.

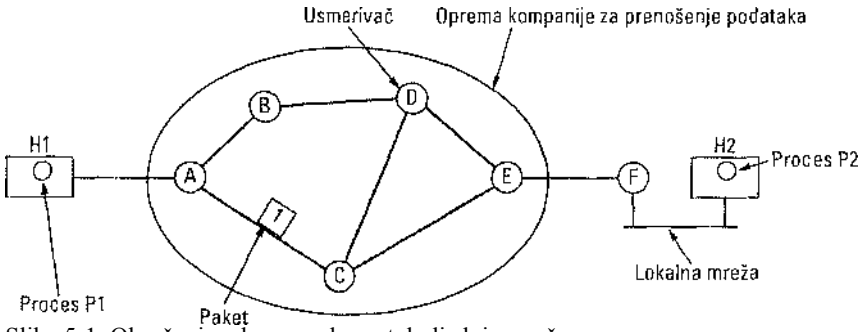
Da bi ostvario postavljeni zadatak, mrežni sloj mora da poznaje topologiju komunikacione podmreže (tj. skupa svih usmerivača) i da kroz nju bira odgovarajuće putanje. On takođe mora da vodi računa o tome da neke linije i usmerivače ne preoptereti, a da drugi ne rade ništa. I na kraju, kada se izvorište i odredište nalaze na različitim mrežama, pojavljuju se nepredviđeni problemi koje mrežni sloj mora da rešava na licu mesta. U ovom poglavlju proučićemo pomenutu problematiku i ilustro- vati je, najčešće pomoću Interneta i protokola njegovog mrežnog sloja - protokola IP, iako ćemo se doticati i bežičnih mreža.

5.1 **PROJEKTOVANJE** MREŽNOG SLOJA

U narednim odeljcima bavićemo se pitanjima na koja moraju da odgovore projektanti mrežnog sloja. Ta pitanja se tiču usluge koju mrežni sloj obezbeđuje transportnom sloju, kao i interne organizacije podmreže.

5.1.1 Komutiranje paketa tehnikom „čuvaj i prosledi“

Pre detaljnog objašnjavanja mrežnog sloja, podsetićemo se konteksta u kome rade njegovi protokoli (slika 5-1). Glavne komponente sistema čini oprema prenosioca podataka (engl. *carrier),* tj. kompanije za prenošenje podataka (usmerivači, povezani preno- snim linijama), prikazana unutar zasenčenog područja, kao i korisnička oprema izvan njega. Računar *Hl* direktno je vezan za jedan od usmerivača prenosioca podataka (A), pomoću iznajmljene linije. Nasuprot tome, računar *H2* se nalazi na lokalnoj mreži sa usmerivačem F, koje poseduje i kojima upravlja korisnik. Taj usmerivač je sa opremom kompanije za prenošenje podataka takođe povezan iznajmljenom linijom. Usmerivač F smo prikazali izvan zasenčenog područja jer ne pripada kompaniji za prenošenje podataka, ali se u pogledu konstrukcije, softvera i protokola on najverovatnije ne razlikuje od njenih drugih usmerivača. Može se raspravljati o tome da li on pripada podmreži, ali ćemo u ovom poglavlju smatrati da su korisnikovi usmerivači deo podmreže jer izvršavaju iste algoritme kao i usmerivači prenosioca podataka, a to je ono što nas sada zanima.



Slika 5-1. Okruženje u kome rade protokoli sloja mreže.

Oprema se koristi na sledeći način. Računar koji ima paket za slanje upućuje ga najbližem

usmerivaču na sopstvenoj lokalnoj mreži ili preko linije od tačke do tačke usmerivaču prenosioca podataka. Paket se tamo čuva dok ne stigne u potpunosti da bi mogao da mu se proveri kontrolni zbir. Zatim se on na isti način šalje sledećem usmerivaču na putanji, sve dok ne stigne na odredište, gde se isporučuje. To je poznati mehanizam „čuvaj i prosledi“, o kome smo govorili u prethodnim poglavljima.

5.1.2 Usluge koje se obezbeđuju transportnom sloju

Mrežni sloj obezbeđuje usluge transportnom sloju na interfejsu između njih. Važno pitanje je koje usluge mrežni sloj obezbeđuje transportnom sloju. Usluge mrežnog sloja projektovane su imajući na umu sledeće:

1. Usluge treba da su nezavisne od tehnologije usmerivača.
2. Transportni sloj ne sme da zna ništa o broju, vrstama i topologiji usmerivača.
3. Mrežne adrese koje se stavljaju na raspolaganje transportnom sloju moraju biti uniformno označene, čak i u lokalnim, odnosno regionalnim mrežama.

Uz navedene ciljeve, projektantima mrežnog sloja ostavljeno je dosta slobode za detaljno specificiranje usluga koje se nude transportnom sloju. Ta sloboda se, doduše, često pretvori u ogorčenu borbu dva zavađena tabora oko toga treba li mrežni sloj da obezbeđuje usluge sa uspostavljanjem ili bez uspostavljanja direktne veze.

Jedan tabor (koji predstavlja zajednica korisnika Interneta) brani gledište da usme- rivači treba samo da usmeravaju pakete i ništa više. Prema njihovom mišljenju (zasnovanom na tridesetogodišnjem radu u realnoj mreži stvarnih računara), podmreža je po prirodi nepouzdana, kako god da je projektovana. Prema tome, računari treba da prihvate činjenicu daje mreža nepouzdana i da sami sprovode kontrolu grešaka (tj. njihovo otkrivanje i ispravljanje) i upravljanje tokom.

Ovo gledište brzo vodi zaključku da mrežne usluge treba da budu bez uspostavljanja direktne veze, sa osnovnim operacijama SEND PACKET, RECEIVE PAC- KET i možda još s nekom dodatnom operacijom. Posebno treba izbegavati uređivanje redosleda paketa i upravljanje tokom jer će to ionako raditi računari, pa će se malo dobiti ako se to uradi još jednom. Osim toga, svaki paket mora da nosi punu odredišnu adresu, pošto se svaki paket šalje nezavisno od svojih eventualnih prethodnika.

Drugi tabor (koji čine telefonske kompanije) tvrdi da podmreža treba da obezbedi pouzdanu uslugu sa uspostavljanjem direktne veze. Oni smatraju da stogodišnje uspešno iskustvo s globalnim telefonskim sistemom predstavlja neoboriv putokaz za budućnost. Po njihovom mišljenju, odlučujući činilac je kvalitet usluga, a bez veza u podmreži kvalitet se teško može postići, naročito u saobraćaju koji se odvija u realnom vremenu (govor, video).

Sukob između dva tabora najbolje se može shvatiti ako uporedimo Internet i ATM. Internet nudi uslugu mrežnog sloja bez uspostavljanja direktne veze, a ATM mreže nude uslugu mrežnog sloja sa uspostavljanjem direktne veze. Treba, međutim, naglasiti da se sa sve većom potražnjom za garantovanim kvalitetom usluge, Internet razvija u tom pravcu. On, kao što ćemo kasnije videti, počinje da dobija svojstva karakteristična za uslugu sa uspostavljanjem direktne veze. U stvari, nešto od toga smo već videli dok smo u 4. poglavlju proučavali virtuelne lokalne mreže.

5.1.3 Realizacija usluge bez uspostavljanja direktne veze

Pošto smo ukazali na dve klase usluga koje mrežni sloj može da ponudi korisnicima,

pogledajmo kako ovaj sloj radi iznutra. Moguće gaje organizovati na dva načina, u zavisnosti od vrste usluge koju nudi. Ako se nudi usluga bez uspostavljanja direktne veze, paketi se pojedinačno ubacuju u podmrežu i usmeravaju nezavisno jedan od dragog. Ništa se ne podešava unapred. U ovom kontekstu, paketi se često zovu **datagrami** (engl. *datagrams)* (po analogiji s telegramima), a podmreža se zove **datagramska podmreža** (engl. *datagram subnet)*. Ako se koristi usluga sa uspostavljanjem direktne veze, pre nego što se pošalje ijedan paket mora se uspostaviti putanja od izvorišnog do odredišnog usmerivača. Takva veza se zove **virtuelno kolo** (engl. *Virtual circuit, VC),* po analogiji s fizičkim kolima koja se uspostavljaju u telefonskom sistemu, a podmreža nosi ime **podmreža s virtuelnim kolima** (engl. *virtual-circuit subnet)*. U ovom odelj- ku proučićemo datagramske podmreže, a u sledećem podmreže s virtuelnim kolima.

Pogledajmo kako radi datagramska podmreža. Pretpostavimo da proces *P1* na slici .5-2 ima dugačku poruku za proces *P2.* On predaje poruku transportnom sloju sa uputstvom da je ovaj isporuči procesu *P2* na računaru *H2.* Kod transportnog sloja izvršava se na računam *Hl,* najčešće unutar njegovog operativnog sistema. On na početak poruke dodaje zaglavlje transportnog sloja i takvu je predaje mrežnom sloju, verovatno samo još jednoj proceduri unutar operativnog sistema.

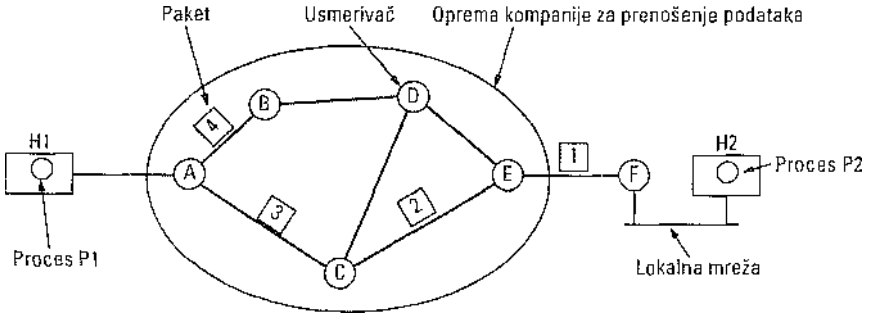


Tabela usmerivača A Tabe,a

na početku ka merivača E

A	-	A	-
B	B	B	B
C	C	C	c
D	B	D	B
E	c	E	B
F	c	F	B

Odredište Linija

Tabe|a

A	A
B	A
C	-
D	D
E	E
F	E

A	i c
B	1 D
C	! C
D	1 D
E	I
F	; F

Pretpostavimo da je poruka četiri puta duža od maksimalne veličine paketa, tako da

Slika 5-2. Usmeravanje u datagramskoj podmreži.

mrežni sloj mora da je podeli u četiri paketa: 1, 2, 3 i 4, i da svaki redom pošalje usmerivaču *A* pomoću nekog protokola od tačke do tačke, npr. protokola PPP. Od tog trenutka posao preuzima prenosilac podataka. Svaki usmerivač ima internu tabelu sa smerovima za svako moguće odredište. Svaka odrednica tabele sadrži odredište i izlaznu liniju namenjenu za to odredište. Mogu se koristiti samo direktno povezane linije. Na primer, na slici 5-2 usmerivač *A* ima samo dve izlazne linije - ka *B* i ka C - tako da svaki pristigli paket mora da se uputi jednom od ta dva usmerivača, iako krajnje odredište predstavlja neki dragi usmerivač. Početno stanje tabele usmerivača A označeno je na slici sa „na početku".

Kada stignu u usmerivač *A,* paketi 1, 2 i 3 u njemu se kratko zadržavaju (da bi im se proverio kontrolni zbir). Zatim se svaki, prema tabeli usmerivača *A,* prosleđuje usmerivaču *C.* Paket 1 se tada upućuje do *E,* pa do *F,* Kada stigne u usmerivač *F,* on se kapsulira u okvir sloja veze i lokalnom mrežom šalje računaru *H2,* Paketi 2 i 3 slede istu putanju.

Međutim, s paketom 4 događa se nešto drugo. Kada stigne u usmerivač A, on se šalje usmerivaču *B,* iako je i on namenjen usmerivaču *F. Iz* nekog razloga, usmerivač *A* je odlučio da paket 4 pošalje drugačijim putem nego prva tri paketa. Možda je na putanji *ACE* naišao na saobraćajnu gužvu i ažurirao svoju tabelu za usmeravanje, što je na slici označeno sa „kasnije". Algoritam koji radi s tabelama i donosi odluke o usmeravanju paketa naziva se algoritam za usmeravanje (engl. *routing algorithm).* Takvi algoritmi predstavljaju jednu od glavnih tema ovog poglavlja.

5.1.4 Realizacija usluge sa uspostavljanjem direktne veze

Za uslugu sa uspostavljanjem direktne veze potrebna nanije podmreža s virtuelnim kolima. Pogledajmo kako ona radi. Virtuelna kola su stvorena da se ne bi za svaki paket izmišljala nova putanja, kao na slici 5-2. Umesto toga, kada se veza uspostavi, putanja između izvorišta i odredišta postaje njen parametar koji se upisuje u tabele usmerivača. Putanja se koristi za sav saobraćaj preko te veze, baš kao u sistemu tele- fonije. Kada se veza raskine, raskida se i virtuelno kolo. Kod usluga koje rade sa uspostavljanjem direktne veze, svaki paket nosi identifikator virtuelnog kola kome pripada.

Razmotrimo primer na slici 5-3. Ovde je računar *Hl* uspostavio vezu 1 s računalom *H2.* Ona je zapisana kao prva odrednica u tabeli svakog usmerivača na putanji. U prvom redu tabele usmerivača A piše da paket sa identifikatorom veze 1 koji stigne od računara *Hl* treba uputiti usmerivaču *C* i dati mu identifikator 1. Prva odrednica tabele usmerivača C, slično tome, upućuje paket usmerivaču *E,* takođe sa identifikatorom veze 1.

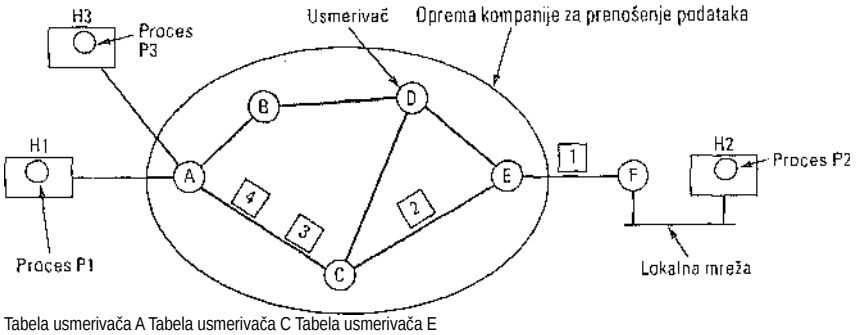


Tabela usmerivača A Tabela usmerivača C Tabela usmerivača E

A ; 1

H3! 1 A ; 2 E ; 2

Ka usmerivaču Iz usmerivača c ; 1

Slika 5-3. Usmeravanje u podmreži s virtuelnim kolima.

A sada pogledajmo šta se događa ako računar *H3* takođe želi da uspostavi vezu s računarom *H2*. On bira identifikator veze 1 (jer on inicira vezu, a to je njegova jedina veza) i nalaže podmreži da uspostavi virtuelno kolo. Tako nastaje drugi red u tabelama usmerivača. Ovde imamo sukob: *A* lako može da razlikuje pakete veze 1 koji dolaze od računara *Hl* i pakete veze 1 koji dolaze od računara *H3*, ali usmerivač *C* to ne može. Zbog toga A dodeljuje drugačiji identifikator paketima koji putuju dragom vezom. Da bi mogli da razreše ovakve sukobe, usmerivači moraju biti sposobni da iz- mene identifikator veze u paketima koje šalju. U određenim slučajevima, to se naziva komutiranje paketa na osnovu oznaka (engl. *label switching).*

5.1.5 Poređenje podmreža s virtuelnim kolima i đatagramskih podmreža

I virtuelna kola i datagrami imaju svoje pobornike i svoje protivnike. Pokušacemo da na istom mestu iznesemo argumente obe strane. Glavne razlike su navedene na slici 5-4, premda bi čistunci verovatno pronašli suprotan primer za svaku stavku.

Stavka	Datagramska podmreža	Podmreža s virtuelnim kolima
Uspostavljanje kola	Nepotrebno	Neophodno
Adresiranje	Svaki paket sadrži punu izvorišnu i odredišnu adresu	Svaki paket sadrži kratak broj virtuelnog kola
Statusni podaci	Usmerivači ne čuvaju podatke o stanju veze	Neophodan je prostor u tabelama usmerivača za svako virtuelno kolo
Usmeravanje	Svaki paket se usmerava nezavisno	Putanja se bira pri uspostavljanju virtuelnog kola i njom se šalju svi paketi
Posledice otkazivanja usmerivača	Nema ih, osim paketa izgubljenih tokom havarije	Prekidaju se sva virtuelna kola koja idu preko pokvarenog usmerivača
Garantovanje kvaliteta usluge	Teško	Lako, ako se unapred može rezervisati dovoljno resursa za svako virtuelno kolo
Kontrola zagušenja	Teška	Jednostavna, ako se unapred može rezervisati dovoljno resursa za svako virtuelno kolo

Slika 5-4. Poređenje podmreža s virtuelnim kolima i datagramima.

Unutar podmreže čine se razni kompromisi da bi se ostvarila virtuelna kola, odnosno prenos datagrama. Jedan se tiče memorije usmerivača i njegovog propusnog opsega. Virtuelna kola dozvoljavaju da paketi nose brojeve kola umesto potpunih adresa odredišta. Kada su paketi uglavnom kratki, puna odredišna adresa u svakom od njih zauzima znatan prostor i tako nepotrebno troši propusni opseg. Međutim, interno korišćenje virtuelnih kola placa se veličinom memorije potrebne za tabele usmerivača. Jedno ili drugo rešenje može da ispadne jeftinije, u zavisnosti od cene komunikacionih kola, odnosno memorije usmerivača.

Drugi kompromis se odnosi na trajanje uspostavljanja veze i trajanje razrešavanja adresa. Kada upotrebite virtuelno kolo, potrebno je određeno vreme da se ono uspostavi, što troši

resurse. Međutim, posle toga se paketi usmeravaju lako: usmerivač na osnovu broja kola i svoje tabele odmah zna kuda da uputi paket. U datagramskoj mreži potrebna je složenija procedura pretraživanja da bi se pronašla odgovarajuća odrednica za prosleđivanje paketa.

Još jedan problem je količina memorije potrebne za tabelu usmerivača. U datagramskoj podmreži, tabela mora da ima odrednicu za svako moguće odredište, dok je u podmreži s virtuelnim kolima potrebna samo odrednica za svako virtuelno kolo. Ta prednost je, međutim, pomalo varljiva, pošto se moraju usmeravati i paketi za uspostavljanje veze, a oni koriste pune adrese odredišta, kao i datagrami.

Virtuelna kola imaju izvesnu prednost jer garantuju kvalitet usluge i uspešno izbe- gavaju zagušenja mreže budući da se resursi (npr. baferi, propusni opseg i procesor- sko vreme) mogu rezervisati unapred - u trenutku uspostavljanja veze. Kada krenu paketi s podacima, na raspolaganju su neophodan propusni opseg i kapacitet usmerivača. U datagramskoj podmreži zagušenje se teže može izbeći.

Kada se radi o obradi transakcija (npr. proveravanju kreditnih kartica pri kupovini u prodavnicama), zadrška pri uspostavljanju i raskidanju virtuelnih kola lako može biti razlog da se ona ne primene. Ako se očekuje da većina saobraćaja bude takve vrste, onda nema smisla koristiti virtuelna kola unutar podmreže. S druge strane, trajna virtuelna kola, koja se ručno uspostavljaju i održavaju mesecima ili godinama, mogu da pomognu u takvim situacijama.

Virtuelna kola su i ranjiva. Ako usmerivač otkaže i izgubi podatke iz memorije, sva virtuelna kola koja prolaze kroz njega moraju se raskinuti, čak i ako se usmerivač povrati samo posle jedne ili dve sekunde. Nasuprot tome, ako usmerivač otkaže u datagramskoj mreži, time će biti pogođeni samo korisnici čiji se paketi u tom trenutku nalaze u usmerivaču, čak možda ni oni ako je već stigla potvrda za pakete. Prekid komunikacionog voda može da bude fatalan za virtuelna kola koja ga koriste, ali se u datagramskoj podmreži taj gubitak Iako nadoknađuje. Usmerivači u datagramskoj podmreži mogu i da uravnotežavaju saobraćaj jer se uzastopni paketi do istog odredišta mogu slati različitim putanjama.

5.2 ALGORITMI ZA USMERAVANJE

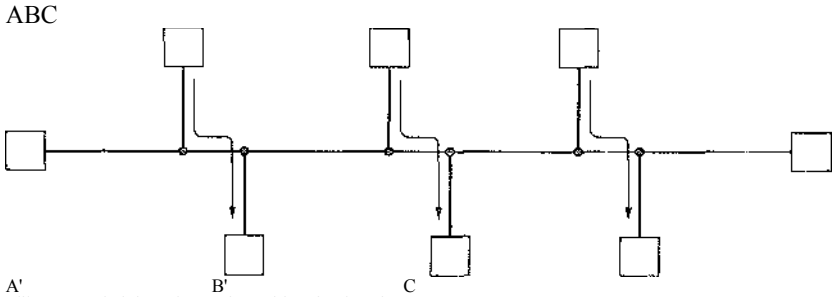
Glavni zadatak mrežnog sloja je da pakete usmeri sa izvorišnog na odredišni računan U većini podmreža, paketi taj put moraju da pređu u više skokova. Jedini značajniji izuzetak su mreže sa difuznim emitovanjem, ali čak i tu ima usmeravanja ako se izvorište i odredište ne nalaze na istoj mreži. Algoritmi za biranje putanja i strukture podataka kojima se oni služe predstavljaju jedan od glavnih zadataka pri projektova- nju mrežnog sloja.

Algoritam za usmeravanje (engl. *routing algorithm*), deo je softvera mrežnog sloja koji odlučuje na koju će izlaznu liniju uputiti pristigli paket. Ukoliko se u podmreži interno koriste datagrami, pomenuta odluka mora se donositi za svaki paket
koji stigne pošto se u međuvremenu može promeniti optimalna putanja. Ako se u podmreži interno koriste virtuelna kola, odluka o usmeravanju donosi se samo kada se uspostavlja novo virtuelno kolo. Posle toga, paketi samo slede utvrđenu putanju. Ta vrsta usmeravanja ponekad se naziva **usmeravanje** za sesiju (engl. *session muting),* pošto ista putanja važi tokom čitave korisničke sesije (npr. sesije prijavljivanja na terminalu ili sesije prenosa datoteka).

Treba podvući razliku između usmeravanja - donošenja odluke o putanji, i pro- sleđivanja - aktuelnog rada s paketom posle njegovog dolaska. Usmerivač se može zamisliti kao uređaj u kome se izvršavaju dva procesa. Čim paket stigne, prihvata ga prvi proces i u tabelama za usmeravanje traži za njega odgovarajuću izlaznu liniju. To je **prosleđivanje** (engl. *forwarding).* Drugi proces, u kome glavnu ulogu ima algoritam za usmeravanje, bavi se popunjavanjem i ažuriranjem tabela za usmeravanje.

Algoritam za usmeravanje treba da ima određena svojstva, bez obzira na to da li se putanja bira nezavisno za svaki paket ili samo pri uspostavljanju nove veze. Ta svojstva su, redom: tačnost, jednostavnost, robusnost, stabilnost, pravičnost i optimal- nost. Tačnost i jednostavnost po sebi su jasna svojstva, ali o robusnosti treba nešto reći. Kada se veća mreža pusti u rad, obično se očekuje da ona godinama radi bez sistemskih prekida. Tokom tog perioda dolaziće do hardverskih i softverskih kvarova: računati, usmerivači i linije stalno će otkazivati, a topologija mreže više ce se puta promeniti. Urnesto da se obustavljaju svi poslovi, ugase svi računati, i mreža ponovo pokrene svaki put kad otkaže neki usmerivač, algoritam za usmeravanje mora da u hodu izlazi na kraj s promenama topologije i saobraćaja u mreži.

I stabilnost je važna osobina algoritma za usmeravanje. Postoje algoritmi koji nikada ne dostižu stanje ravnoteže, bez obzira koliko dugo rade. Nasuprot tome, stabilan algoritam srazmerno brzo dostiže ravnotežu i zatim je ne napušta. Pravičnost i optimalnost sigurno su poželjni - niko razuman se tome ne bi protivio - ali se ispostavlja da se ova dva svojstva često sukobljavaju. Na slici 5-5 dat je jednostavan primer sukobljavanja. Pretpostavimo daje saobraćaj između stanica A i A', stanica *B* i *B* i stanica C i *C* dovoljno intenzivan da su horizontalne veze zasićene. Da bi se ukupan protok podataka maksimirao, saobraćaj između stanica *X* i *X'* treba potpuno obustaviti. Nažalost, stanice *X* i *X'* ne vide situaciju na isti način. Očigledno treba postići kompromis između globalne efikasnosti i pravičnosti prema pojedinačnim vezama.



Slika 5-5. Sukob između pravičnosti i optimalnosti.

Pre nego što pokušamo da potražimo kompromisno rešenje, moramo jasno utvrditi šta je to što želimo da optimizujemo. To može da bude minimizovanje srednjeg kašnjenja paketa, ali i maksimiranje ukupnog protoka kroz mrežu. Čak se i ova dva za- hteva sukobljavaju, jer približavanje radnog režima gornjoj granici projektovanog kapaciteta obavezno produžava boravak paketa u redovima čekanja. U mnogim mrežama se, kao kompromisno rešenje, minimizira broj skokova paketa, pošto to smanjuje njihovo ukupno kašnjenje, a štedi se i propusni opseg, zbog čega se povećava protok kroz mrežu.

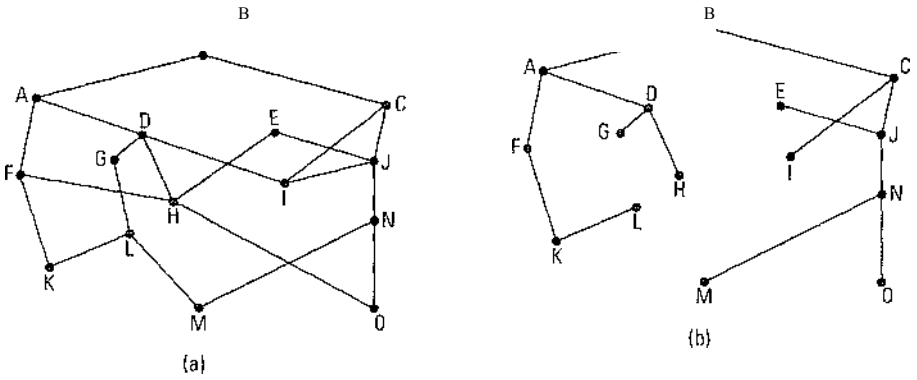
Algoritme za usmeravanje možemo svrstati u dve glavne klase: neprilagodljive i prilagodljive. **Neprilagodljivi algoritmi** (engl. *noncidaptive algorithms*) ne zasnivaju svoje odluke o usmeravanju na metenju ili procenjivanju tekućeg saobraćaja u mreži, niti na njenoj aktuelnoj topologiji, već se putanja između *I* i *J* (za svako / i svako *J)* izračunava unapred, na radnom stolu, i učitava u usmerivače pri pokretanju mreže. Taj postupak se ponekad naziva statičko usmeravanje (engl. *static routing).*

Za razliku od toga, prilagodljivi algoritmi (engl. *adaptive algorithms*) menjaju svoje odluke o usmeravanju u zavisnosti od promena u topologiji mreže, a često i od trenutnog saobraćaja kroz nju. Prilagodljivi algoritmi se razlikuju po tome odakle do- bijaju informacije (npr. iz lokalnih izvora, od okolnih usmerivača ili od svih usmeri- vača u mreži), koliko često menjaju putanje (npr. svakih A*T* sekundi, svaki put kada se promeni opterećenje mreže ili kada se promeni njena topologija), i kakva metrika se koristi za optimizovanje (npr. razdaljina, broj skokova, ili procenjeno vreme pre- nosa). U narednim odeljcima opisaćemo više algoritama za usmeravanje, kako statičkih, tako i dinamičkih.

5.2.1 Princip **optimalnosti**

Pre nego što pređemo na konkretne algoritme, možda treba naglasiti daje moguće doneti opšti stav o optimalnoj putanji, bez obzira na topologiju mreže i saobraćaj u njoj. Taj stav je poznat kao princip optimalnosti (engl. *optimality principle).* On glasi: ako se usmerivač *J* nalazi na optimalnoj putanji između usmerivača *I* i usmerivača *K,* tada se optimalna putanja između *J* i *K* nalazi na istoj putanji. Da biste stav razu- rneli, označite sa r_x putanju između *I* i /, a ostatak putanje sa r_2. Kada bi između *J* i *K* postojala putanja bolja od r_2, ona bi se mogla nadovezati na /•] da bi se poboljšala putanja između *I* i *K,* što protivreči pretpostavci daje putanja *rp-o* optimalna.

Iz principa optimalnosti direktno proizlazi da skup putanja iz svih izvora ka zadatom odredištu obrazuje stablo ukorenjeno na odredištu. To je **stablo optimalnih putanja** (engl. *sink tree);* prikazano je na slici 5-6, gde se rastojanje meri brojem skokova. Obratite pažnju na to da ne mora postojati samo jedno stablo optimalnih putanja; mogu postojati i druga stabla sa istom dužinom putanja. Osnovni zadatak svih algoritama za usmeravanje jeste da otkrivaju stabla optimalnih putanja i da ih pri- menjuju na sve usmerivače.



Slika 5-6. (a) Podmreža. (b) Stablo optimalnih putanja za usmerivač#.

Posto je stablo optimalnih putanja zaista stablo, ono ne sadrži petlje, pa se svaki paket može isporučiti posle konačnog, ograničenog broja skokova. U praksi to izgleda malo drugačije. Veze i usmerivači otkazuju i ponovo se oporavljaju tokom rada, pa različiti usmerivači mogu imati različitu predstavu o aktuelnoj topologiji mreže. Vesto smo zao- bišli i pitanje treba li svaki usmerivač pojedinačno da prikupi informacije na osnovu kojih izračunava stablo optimalnih putanja ili se ti podaci prikupljaju na neki drugi način. Na to pitanje ćemo se ubrzo vratiti. Pa ipak, princip optimalnosti i stablo optimalnih putanja predstavljaju kriterijum za poređenje svih algoritama za usmeravanje.

5.2.2 Usmeravanje najkraćom putanjom

Počnimo proučavanje praktičnih algoritama od tehnike čije se mnoge varijante široko koriste zato što je jednostavna i shvatljiva. Prvo se napravi graf podmreže u kome svaki usmerivač predstavlja jedan čvor, a svaka komunikaciona linija (zvana i veza) jedan luk. Algoritam bira putanje između para zadatih usmerivača tako što na grafu pronalazi najkraći put između njih.

Koncept **najkraće putanje** (engl. *shortest path)* treba kratko objasniti. Jedan način merenja dužine putanje jeste brojanje skokova. Uz tu metriku, putanje *ABC* i *ABE* na slici 5-7 jednake su dužine. Dragi način je merenje geografskog rastojanja u kilometrima, kada je putanja *ABC* očigledno duža od putanje ri/JA (pod pretpostavkom da je na slici mreža predstavljena u razmeri).

Međutim, mogući su i mnogi drugi načini merenja razdaljine. Na primer, može se za svaki luk odrediti prosečno kašnjenje probnog paketa izazvano čekanjem u redu i samim prenosom. Kada se ceo graf tako označi, onda nije najkraća ona putanja koja ima najmanje lukova ili kilometara, već putanja kojom paket najbrže prolazi.

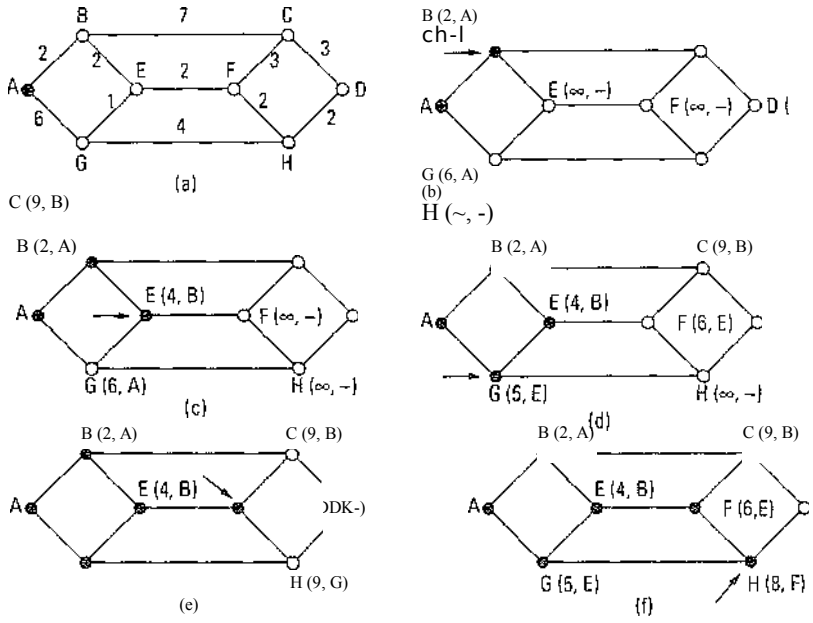
Kašnjenje u pojedinim lukovima u opštem slučaju može se izračunavati u funkciji rastojanja, propusnog opsega, prosečnog saobraćaja, cene komuniciranja, srednje dužine redova čekanja, registrovanog kašnjenja i drugih činilaca. Ta funkcija se može prilagoditi da uzme u obzir samo neke od pobrojanih činilaca ili neku njihovu kombinaciju, tako da se

pomoću algoritma za usmeravanje može izračunati „najkraća putanja“ baš za tu situaciju.

Poznato je više algoritama za izračunavanje najkraće putanje između dva čvora grafa. Opisaćemo algoritam koji dugujemo Dijkstri (1959). Svaki čvor se (u zagradi) označi rastojanjem od izvorišnog čvora duž najbolje poznate putanje. Na početku nije poznata nijedna putanja, tako da je označeno rastojanje svakog čvora beskonačno. Kako se algoritam izvršava, pronalaze se nove (sve bolje) putanje i oznake se shodno tome menjaju. Oznaka može da bude privremena i trajna. Na početku su sve oznake privremene. Međutim, kada se jednom utvrdi daje putanja od izvorišta do odredišta najkraća moguća, ona postaje trajna i više se ne menja.

Da biste razumeli kako radi algoritam označavanja, pogledajte ponderisani neu- smeren graf na slici 5-7(a), gde pojedine „težine“ lukova predstavljaju, na primer, ra- stojanja. Želimo da pronađemo najkraći put od A do *D*. Počinjemo tako što čvor *A* proglašavamo trajnim i obeležavamo ga crnim kružićem. Zatim redom ispitujemo svaki čvor u neposrednoj okolini (radnog) čvora A i označavamo ih rastojanjem od njega. Kad god čvoru pridružimo rastojanje, naznačujemo i čvor od koga je mereno, tako da kasnije možemo da rekonstruišemo konačnu putanju. Pošto ispitamo čvorove bliske čvoru A, ispitujemo i sve privremeno označene čvorove u ćelom grafu, i onaj s najmanjom oznakom pretvaramo u trajan čvor, kao na slici 5-7(b). On tada postaje nov radni čvor.

Sada počinjemo od *B* i ispitujemo sve njemu bliske čvorove. Ako je zbir oznake čvora *B* i rastojanja između *B* i ispitivanog čvora manji od oznake ispitivanog čvora, pronašli smo kraću putanju, pa tom čvoru menjamo oznaku.



Slika 5-7. Prvih pet koraka izračunavanja najkraće putanje od ,4 do ***D.*** Radni čvor je označen strelicom.

Pošto ispitamo sve čvorove bliske radnom čvoru i ako je moguće izmenimo njihove oznake, pretražujemo ceo graf da bismo našli čvor s najmanjom vrednošđu privremene oznake. Taj čvor pretvaramo u trajan čvor i on postaje radni čvor za sledeći korak. Slika 5-7 prikazuje prvih pet koraka algoritma.

Da biste videli kako algoritam radi, pogledajte sliku 5-7(c). U toj fazi smo čvor *E* upravo učinili trajnim. Pretpostavimo da, na primer, postoji putanja *AXYZE* koja je kraća od putanje *ABE*. Ima dve mogućnosti: ili je čvor Z već postao trajan ili nije. Ako je već postao trajan, tada je čvor *E već* ispitan (u koraku iza onog kada je Z postao trajan), tako da pri ispitivanju nismo mogli zaobići putanju *AXYZE,* što znači da ona nije kraća.

Sada razmotrimo slučaj kada Z još uvek nosi privremenu oznaku. Oznaka čvora Z može biti jednaka oznaci čvora *E* ili biti veća od nje, kada *AXYZE* ne može biti krađe od *ABE*, ili je, pak, ona manja od oznake čvora *E,* što znači da će od dva čvora čvor Z prvo postati trajan i da će tek on ispitati čvor *E.*

Ovaj algoritam je prikazan na slici .5-8. Globalne promenljive *n* i *dist* opisuju graf i inicijalizuju se pre nego što se pozove procedura *shortestjpath.* Jedina razlika između programa i opisanog algoritma jeste u tome što se na slici 5-8 najkraća putanja izračunava počev od završnog čvora *t,* umesto od izvorišnog čvora *s.* Pošto je u neus- merenom grafu najkraća putanja između f i *s* isto što i najkraća putanja između i' i *t,* nije važno odakle počinjemo izračunavanje (osim ako ima više najkraćih putanja, kada obrtanjem smera izračunavanja možemo da otkrijemo drugačije putanje). U programu se primenjuje izračunavanje od kraja ka početku zato što svaki čvor nosi oznaku prethodnog, a ne narednog čvora. Kada se konačna putanja kopira u izlaznu promenljivu *path,* ona se izvrće; ako izvrnemo i redosled pretraživanja grafa, dva efekta se poništavaju i u rezultatu dobij amo putanju baš kao što treba.

5.2.3 Plavljenje

Drugi algoritam statičkog usmeravanja jeste **plavljenje** *(engl. ftooding);* svaki dolazni paket šalje se na sve izlazne linije, osim na onu s koje je došao. Plavljenjem se očito generiše ogroman, u stvari, beskonačan broj duplikata, osim ako se određenim merama proces ne priguši. Takva mera je brojač skokova u zaglavlju svakog paketa; njegova vrednost se pri svakom skoku smanjuje za jedinicu, a kada dostigne nulu, paket se odbacuje. Bilo bi idealno da se brojač skokova inicijalizuje vrednošđu koja odgovara rastojanju između izvorišta i odredišta. Ako pošiljalac ne zna to rastojanje, u najgorem slučaju može da inicijalizuje brojač prema prečniku čitave pođmreže.

Poplava paketa se može ograničiti i tako što se povede računa o poslatim paketima. Tu izvorišni usmerivač pridružuje redni broj svakom paketu koji primi od raču- nara za koje je vezan. Svaki dalji usmerivač tada treba da vodi listu za svaki izvorišni usmerivač i da u nju beleži pakete koji su već stigli s tog izvora. Kada ponovo stigne paket koji je na listi, on se dalje ne emituje mehanizmom plavljenja.

Da liste ne bi neograničeno rasle, svakoj se pridružuje brojač *k,* čija vrednost znači da su kroz usmerivač prošli svi paketi do rednog broja *k.* Kada stigne paket rednog broja manjeg od *k,* on se odbacuje kao duplikat. Lista paketa s rednim brojevima manjim od *k* nije potrebna, zato što ih brojač *k* sve sumira.

Varijanta plavljenja je tzv. selektivno plavljenje (engl. *selective flooding).* Ovde

usmerivači ne šalju svaki dolazni paket na svaku liniju, već samo na one linije koje se pružaju u približno ispravnom smeru. Paket upućen na istok skoro da nema smisla slati na „zapadnu“ liniju, osim ako je topologija mreže baš neobična, a usmerivač tačno zna šta radi.

```
#define MAXJMODES 1024                          INFINITY; p->label = tentative;
#define INFINITY 1000000000                  /* maksimalan broj čvorova 7
int n, dist[MAX_NODES][MAX_NODES];           /* broj veći od svake najveće putanje 7
                                             /* distjijO] je rastojanje od i do j 7
void shortest__path(int s, int t, int pathfl)
{struct state {
      int predecessor;                       /* putanja na kojoj se radi 7 /*
      int length;                            prethodni čvor 7
       enum {permanent, tentative} label;    /* rastojanje od izvorišnog čvora do ovog
} state[MAX_NODES];                          7 /* stanje oznake 7

int i, k, min, struct state *p;

for (p = &state[0; p < &state[n]; p++) { p-
      >predecessor = 1; p->length =
                                             /* Inicijalizovanje stanja 7

state[t],length = 0;state[tj.label = permanent;
k = t;                                       /*   k je početni radni čvor 7
do {                                         /*   Ima li boljeg puta od k dovde? 7
      for (i =  0;i < n; i++)                /*   ovaj graf ima n čvorova 7
            if (dist[k][i] != 0 && stateji].label == tentative) { if
            (statejkj.length + dist[k][i] < state[i].length)
            { state[i],predecessor = k; stateji].length =
            statejk].length + dist[k][i];



      /* Pronađi privremeno označen čvor s najmanjom oznakom. 7
      k = 0; min = INFINITY; for (i = 0; i < n; i++)
            if (statejij.label == tentative && stateji].length < min) {
                  min = stateji].length; k = i;

      statejk]. label = permanent;
  while (k != s);

/* Kopiraj putanju u izlazni niz. 7
i = 0; k = s;
do }path[i++] = k; k = statejk].predecessor;} while (k >= 0);
}
```

Slika 5-8. Dijkstrin algoritam izračunavanja najkraćeg puta kroz graf.

Plavljenje je obično neprikladan način usmeravanja, ali ima situacija gde se koristi. Na primer, u ratnim uslovima, gde svakog trenutka može biti uništen veliki broj usmerivača, poželjno je temeljito plaviti mrežu. Plavljenje može biti korisno i kada istovremeno treba ažurirati veliki broj distribuiranih baza podataka. U bežičnim mrežama, sve poruke koje šalje stanica mogu primiti sve dinge stanice u njenom dometu, što je u stvari plavljenje koje iskorišćavaju neki algoritmi. Plavljenje se može iskoristiti i kao standard za upoređivanje drugih algoritama za usmeravnje. Plavlje- njem se uvek pronalazi i najkraća putanja (s

najmanjim kašnjenjem) jer se paralelno šalju paketi svim mogućim putanjama. Svi drugi algoritmi usmeravanja mogu da pro- izvedu samo isto ili veće kašnjenje (ako zanemarimo zadršku koju generiše sam proces plavljenja).

5.2.4 Usmeravanje zasnovano na vektoru razdaljine

U savremenim računarskim mrežama obično se umesto opisanih statičkih koriste algoritmi za dinamičko usmeravanje jer statički algoritmi ne uzimaju u obzir aktuelni saobraćaj na mreži. Najpopularnija su dva dinamička algoritma za usmeravanje koji su zasnovani: na vektoru razdaljine i na stanju veze. U narednom odeljku bavićemo se prvim od njih, a posle toga i drugim.

Algoritmi za **usmeravanje na osnovu vektora razdaljine (engl.** *distance vector routing*) rade tako što usmerivač održava tabelu (tj. vektor) s najkraćim poznatim ra- stojanjima do svakog odredišta, i linijama preko kojih se do odredišta može stići. Usmerivač ažurira te tabele razmenjujući informacije sa susedima.

Algoritam usmeravanja zasnovan na vektoru razdaljine nosi i drugačija imena, najčešće **Belman-Fordov** algoritam za distribuirano usmeravanje i **Ford-Fulkersonov** algoritam, prema istraživačima koji su ih napravili (Bellman, 1957, Ford i Fulkerson, 1962). To je bio prvobitni algoritam za usmeravanje u ARPANET-u, a korišćen je i na Internetu pod imenom RIP.

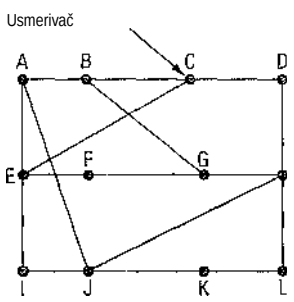
Pri usmeravanju zasnovanom na vektoru razdaljine, svaki usmerivač održava in- deksiranu tabelu svih usmerivača u podmreži. Odrednica za svaki usmerivač sadrži: najpovoljniju izlaznu liniju i procenjeno vreme ili rastojanje za stizanje do njega. Dužina puta se može meriti brojem skokova, kašnjenjem u milisekundama, ukupnim brojem paketa u redu čekanja na putanji itd.

Za usmerivač se pretpostavlja da „zna“ rastojanje do svakog od svojih suseda. Ako se rastojanje meri skokovima, onda je ono do svakog suseda samo jedan skok. Uko liko se meri dužina reda čekanja, usmerivač jednostavno ispita svaki red. Ako se meri kašnjenje, usmerivač ga može direktno izmeriti šaljući specijalan paket ECFIO kome primalac samo „zalepi“ vremensku oznaku i odmah ga pošalje nazad.

Pretpostavimo, primera radi, da se kao mera rastojanja koristi kašnjenje paketa i da usmerivač zna kašnjenje do svakog od svojih suseda. Svakih T sekundi svaki usmerivač šalje svakom svom susedu listu svojih procenjenih kašnjenja do njih. On takođe dobija sličnu listu od svakog svog suseda. Zamislite da je jedna takva tabela upravo stigla od suseda X, pri čemu je X_I njegova procena vremena potrebnog za stizanje do

nekog usmerivača *i*. Ako usmerivač zna daje paketu do *X* potrebno *m* milisekundi, on zna i to da paket može da stigne do usmerivača *i* preko *X* za $X_t + m$ milisekundi. Vršeći slične proračune za svakog suseda, usmerivač utvrđuje koja je procena najbolja, koristi je i upisuje u svoju novu tabelu za usmeravanje. Imajte na umu da se u ovakvim proračunima ne koristi stara tabela.

Opisani proces ažuriranja prikazan je na slici 5-9. Njen deo (a) prikazuje pod- mrežu. Prve četiri kolone u delu (b) prikazuju vektore kašnjenja koje je usmerivač *J* primio od svojih suseda. A obaveštava da kašnjenje od njega do *B* iznosi 12 ms, do *C* - *25* ms, do *D* - 40 ms itd. Pretpostavimo da je usmerivač *J* izmerio ili procenio kašnjenje do svojih suseda: A, I, H i K i da su ta kašnjenja redom: 8, 10, 12 i 6 ms.



Vektori koje J primi od svoja četiri suseda

Do	A	I	H	K
A	0	24	**20**	21
B	12	36	31	28
C	25	18	19	36
D	40	27		24
E	14		30	**22**
F	23	**20**	19	40
G	18	3		31
H	17	**20**		19
1	21		14	**22**
J	9			10
K	24	**22**	**22**	
L	29	33		
	Kašnjenj između J i A je	između J i I je 10	između J i H je 12	između J i K je **6**

Novo procenjeno kašnjenje od J | Linija

8	A
20	A
28	I
20	H
17	I
30	I
18	H
12	H
10	I
0	-
6	K
15	K

Nova tabela za usmeravanje kod usmerivača J

(b) (a)

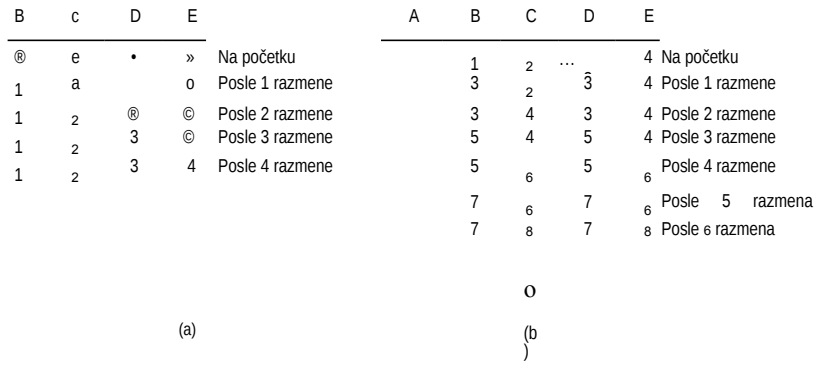
Slika 5-9. (a) Podmreža. (b) Podaci dobijeni od usmerivača A, I, H, K i nova tabela za usmeravanje kod usmerivača J,

Razmotrite kako *J* izračunava svoju novu putanju do usmerivača G. On zna da do A može da stigne za 8 ms, a usmerivač A tvrdi da G može da dosegne za 18 ms, pa *J* zna da može da računa sa zadrškom od 26 ms ako paket namenjen usmerivaču *G* pro- sledi preko A. On na sličan način izračunava kašnjenje do *G* preko *I, H* i *K:* 41 (31 + 10), 18 (6 + 12), odnosno 37 (31 + 6) ms. Najmanja od ovih vrednosti je 18, pa tu vrednost smešta u svoju tabelu kao dužinu puta do usmerivača G, upisujući i to da pakete za G treba slati preko *H.* Isto izračunavanje se ponavlja za sva ostala odredišta i tako se popunjava tabela predstavljena poslednjom desnom kolonom na slici.

Problem približavanja beskonačnosti

Usmeravanje zasnovano na vektoru razdaljine teorijski je dobro zamišljeno, ali u praksi pokazuje ozbiljan nedostatak: iako algoritam uvek na kraju daje optimalno rešenje, to dugo traje. Konkretno, on brzo reaguje na povoljne informacije, ali kad dobije nepovoljne, okleva da ih upiše. Razmotrite usmerivač čija je procena najkraćeg rastojanja do odredišta *X* velika. Ako pri sledećoj razmeni informacija njegov susedA najednom izvesti o kratkom kašnjenju do *X,* usmerivač će se odmah prebaciti na tu liniju i saobraćaj ka *X* upućivati preko A. On prihvata dobra vest samo na osnovu jedne informacije.

Da biste se uverili kako se brzo šire dobre vesti, razmotrite (linearnu) podmrežu s pet čvorova, prikazanu na slici 5-10, gde se rastojanje meri brojem skokova. Pretpostavimo da je na početku usmerivač A isključen i da ostali usmerivači to znaju. Dragim recima, svi su oni u svoje tabele zabeležili beskonačno kašnjenje do usmerivača A.



Slika 5-10. Problem približavanja beskonačnosti.

Kada se A uključi, ostali usmerivači će to saznati nakon što razmene vektore. Pretpostavićemo zbog jednostavnosti da negde otkucava džinovski sat koji usklađuje istovremenu razmenu vektora svih usmerivača. U prvoj razmeni, *B* saznaje da kašnjenje od njegovog levog suseda do A iznosi 0 (skokova). *B* zapisuje u tabelu da se A nalazi na jedan skok ulevo od njega. Svi ostali usmerivači još uvek misle da je A u kvaru. Odrednice za A u tabelama usmerivača u tom trenutku odgovaraju drugom redu na slici 5-10(a). U sledećoj razmeni, C saznaje da je *B* udaljen za jedan skok od A, pa ažurira svoju tabelu vrednošću 2, ali dobre vesti još nisu stigle do C i *D,* Odavde je jasno da se dobre vesti šire brzinom od jednog skoka po jednoj razmeni informacija. U podmreži čija najduža putanja obuhvata *N* skokova, svi će posle *N* razmena saznati da su se neki usmerivač ili linija vratili u život.

Razmotrimo sada situaciju na slici 5-10(b), na kojoj u početku sve linije i svi usmerivači rade. Usmerivači *B, C, DiE* udaljeni su od usmerivača A za 1, 2, 3, odnosno 4 skoka. Najednom, usmerivač A „puca“ ili se prekida linija između A i *B,* što usmeri- vaču *B* znači isto.

U prvoj razmeni paketa, *B* ne dobija ništa od usmerivača A. Na sreću, C mu saopšta- va: ne brini; ja znam putanju do *A* dužine 2 skoka. *B* ne zna da putanja od usmerivača C do usmerivača A vodi kroz njega; on misli da C možda ima desetinu linija s putanjama do A i da su sve dužine 2 skoka. Zbog toga, *B* misli da do A može da stigne preko C, putanjom dužine 3 skoka. Usmerivači *D* i *E* ne ažuriraju svoje odrednice za A u prvoj razmeni.

U drugoj razmeni, C primećuje da svaki njegov sused saopštava da ima putanju do A dužine 3 skoka. On nasumično bira jednog od njih i beleži svoju novu putanju do A u 4 skoka - treći red na slici 5-10(b). U narednim razmenama situacija se razvija onako kako je prikazano u preostalim redovima na slici 5-10(b).

Sa slike bi trebalo da bude jasno zašto se loše vesti sporo šire; vrednost razdaljine do zadatog odredišta na susednim usmerivačima razlikuje se najviše za jedinicu. Vremenom se svi usmerivači približavaju beskonačnosti, ali broj razmena koji je potreban da se ona dostigne zavisi od brojčane vrednosti kojom je predstavljena. Zbog toga je mudar potez da se beskonačnost definiše vredn'ošću koja je za 1 skok veća od najduže putanje. Ako se putanje mere vremenom prenosa paketa, onda „najduža“ putanja nije strogo definisana, pa se za beskonačnost mora predvideti znatno viša vrednost da se zbog paketa koji od nekog usmerivača stižu sporo taj usmerivač ne bi proglasio „pokvarenim“. Neće vas iznenaditi stoje opisani problem nazvan problem približavanja beskonačnosti (engl. *count-to-infinityproblem)*. Bilo je nekoliko pokušaja da se on resi (jedan je opisan u RFC dokumentu 1058), ali se u praksi baš nisu dobro pokazali. Kada usmerivač *X* saopšti usmerivaču *Y* da ima putanju do nekog odredišta, problem je u tome što usmerivač *Y* nema načina da sazna nalazi li se i sam na njoj.

5.2.5 Usmeravanje zasnovano na stanju veze

Usmeravanje zasnovano na vektoru razdaljine korišćeno je u ARPANET-u do 1979, kada je zamenjeno usmeravanjem na osnovu stanja veze. Tu smenu su podstak- la dva problema. Prvo, pošto je za merenje „rastojanja“ korišcena dužina reda čekanja, pri biranju linija nije uziman u obzir njihov propusni opseg. Na početku su sve linije radile brzinom od 56 kb/s, tako da se taj problem nije postavljao, ali se on oštro ispoljio kada je brzina u nekim linijama povećana na 230 kb/s, a u nekim drugim čak i na 1,544 Mb/s. Naravno, metrika se mogla podesiti tako da uzme u obzir propusni opseg linija, ali je postojao i drugi problem: algoritam je sporo dostizao ravnotežu (problem približavanja beskonačnosti). Zbog svega toga, način usmeravanja zame- njen je potpuno novim algoritmom, poznatim kao usmeravanje zasnovano na stanju veze (engl. *link State routing)*. Sada se koriste njegove različite varijante.

Logika usmeravanja na osnovu stanja veze jednostavna je i može se sažeti u pet koraka. Svaki usmerivač treba:

1. Da otkrije svoje susede i sazna njihove mrežne adrese.
2. Da izrneri vremensko rastojanje ili troškove slanja do svakog svog suseda.
3. Da napravi paket sa svim podacima koje do datog trenutka ima.

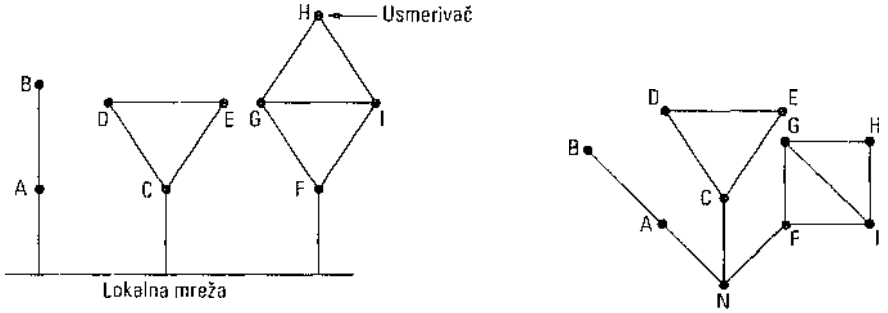
4. Da taj paket pošalje svim usmerivačima.
5. Da izračuna najkraću putanju do svakog drugog usmerivača.

Na taj način, eksperimentalno utvrđeni topologija mreže i vremensko kašnjenje do svakog odredišta distribuiraju se svakom usmerivaču. U nastavku ćemo detaljno obraditi svaki navedeni korak.

Upoznavanje suseda

Kada se usmerivač pokrene, prvi zadatak mu je da se obavesti o susedima. Da bi to saznao, na svaku liniju od tačke do tačke šalje specijalan pozdravni paket HELLO. Kada usmerivač dobije takav paket, treba da se predstavi. Imena moraju biti globalno jedinstvena kako bi usmerivač koji kasnije dobije informaciju da su tri usmerivača povezana sa usmerivačem F, mogao biti siguran da sva tri usmerivača misle na isto F.

Kada se dva ili više usmerivača nalaze u lokalnoj mreži, situacija je nešto složenija. Na slici 5-11 (a) prikazana je lokalna mreža s kojom su direktno povezana tri usmerivača: A, C i F. Kao što vidite sa slike, svaki od ovih usmerivača dodatno je povezan s jednim ili više drugih usmerivača.



(a) (b)

Slika 5-11. (a) Devet usmerivača i lokalna nireža. (b) Graf koji odgovara mreži (a).

Lokalna mreža se na grafu može i sama predstaviti čvorom, kao na slici 5-ll(b). Tu smo uveli nov, zamišljeni čvor *N,* za koji su vezani čvorovi A, *C* i F. Činjenica da se od A do C može stići preko lokalne mreže, ovde je predstavljena putanjom *ANC.*

Merenje troškova slanja linijom

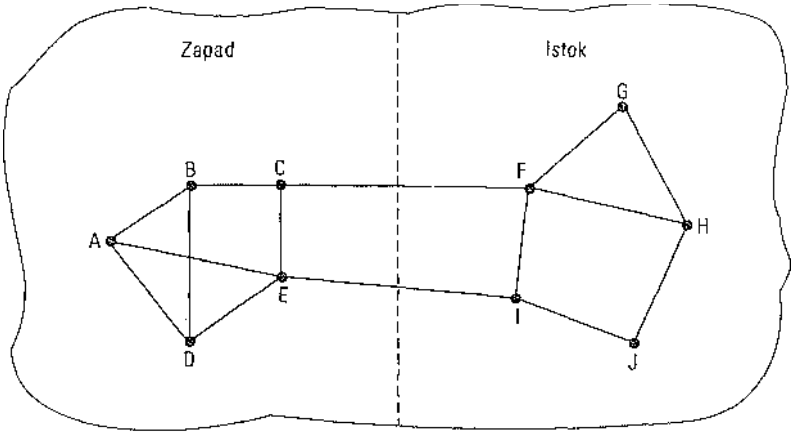
Algoritam zasnovan na stanju veze obavezuje svaki usmerivač da zna ili barem da može razumno da proceni vreme koje paket provede na putu do svakog svog suseda. To vreme se neposredno može odrediti ako se na liniju pošalje paket ECHO koji druga strana mora vratiti čim ga primi. Kada izmeri vremenski interval između slanja i ponovnog primanja istog paketa, i podeli ga sa dva, pošiljalac dobija razumnu pro- cenu vremena potrebnog paketu da stigne do tog odredišta. Rezultat je tačniji ako se opisani postupak ponovi više puta i izvuče

prošek. Naravno, ovde se pretpostavlja da put paketa u oba smera traje isto, što ne mora uvek da bude slučaj.

Postavlja se zanimljivo pitanje da li pri merenju putanje treba uzeti u obzir i opterećenje. Da bi se opterećenje uzelo u obzir, tajmer za merenje vremena obilaska putanje treba aktivirati kada se paket ECHO smesti u red čekanja. Ako opterećenje zanemarujemo, tajmer treba aktivirati kada paket ECHO izbije na čelo reda čekanja.

Postoje argumenti i za jednu i za drugu opciju. Ako se u vreme putovanja paketa uključi i opterećenje, to znači da bi usmerivač mogao da bira između linije koja je sve vreme opterećena i linije koja to nije, pa bi neopterećenu liniju smatrao kraćom putanjom. Takav izbor bi dao bolje performanse.

Nažalost, postoji i argument protiv uključivanja opterećenja u merenje. Razmotrite podmrežu na slici 5-12, čija su dva dela, Istok i Zapad, spojeni pomoću dve linije, *CF* i *EI.*

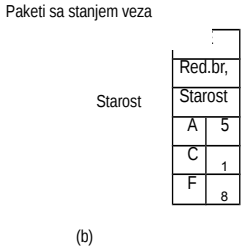


Slika 5-12. Podmreža čija su delovi (Istok i Zapad) spojeni pomoću dve linije.

Pretpostavimo da se glavnina saobraćaja između Istoka i Zapada odvija linijom *CF,* zbog čega je ova linija teško opterećena, a vremenska zadrška u njoj velika. Kada u merenje uključimo boravak u redu čekanja, linija *EI* dobija prednost. Pošto se instaliraju nove tabele za usmeravanje, veći deo saobraćaja između Istoka i Zapada pre- mestiće se na liniju *EI* i preopteretiti je. Pri sledećem ažuriranju, prednost će ponovo imati linija *CF,* sada manje opterećena. I tako će odrednice u tabelama za usmeravanje stalno oscilovati između dve vrednosti, što može da izmakne kontroli i da dovede do mnogih potencijalnih problema. Ako se opterećenje pojedinih linija zanemari i u obzir uzme samo njihov propusni opseg, opisana situacija ne može da se javi. Alternativno, opterećenje se može ravnomerno rasporediti na dve linije, ali takvo rešenje ne bi u potpunosti koristilo prednost traženja najkraće putanje. Bez obzira na to, da bi se sprečilo oscilovanje tabela za usmeravanje, možda je korisno da se opterećenje raspodeli na više linija, pri čemu se svakom linijom odvija neki unapred poznat udeo ukupnog saobraćaja.

Pravljenje paketa sa stanjem veze

Kada prikupi podatke potrebne za razmenu, svaki usmerivač treba da napravi paket koji ih sadrži. Pošiljalac se na početku paketa identifikuje, a zatim dolaze redni broj i starost (što ćemo kasnije objasniti), pa lista suseda. Uz susede se navodi kašnjenje do svakog od njih. Na slici 5-13(a) prikazana je podmreža s kašnjenjima označenim uz svaku liniju, dok su na slici 5-13(b) prikazani paketi sa stanjem veze za svih šest usmerivača.



Slika 5-13. (a) Podmreža. (b) Paketi sa stanjem veza za ovu podmrežu.

Lako je napraviti paket sa stanjem veza - teže je odrediti kada ga treba praviti. Jedna mogućnost je da se takvi paketi prave periodično, tj. u redovnim vremenskim razmacima. Druga je da se prave kada se dogodi nešto neuobičajeno, npr. kada linija ili suseđ otkažu ili se oporavljaju od otkaza, ili primetno promene svoja svojstva.

Distribuiranje paketa sa stanjem veze

Najproblematičniji deo algoritma je pouzdano distribuiranje paketa sa stanjem veze. Dok se paketi distribuiraju i instaliraju, usmerivači koji su ih prvi primili već će promeniti svoje putanje. Zbog toga, u određenom trenutku, razni usmerivači mogu da koriste različite verzije topologije, što može dovesti do pojave nedoslednosti, petlji, nedostupnih računara i sličnih problema.

Najpre ćemo objasniti osnovni algoritam za distribuiranje, a kasnije ćemo ga malo usavršiti. Prema osnovnoj zamisli, paketi sa stanjem veze distribuiraju se mehanizmom plavljenja. Da bi se poplava držala pod kontrolom, svaki paket sadrži redni broj koji se uvećava za svaki sledeći poslati paket. Usmerivači evidentiraju sve parove: izvorišni usmerivač-redni broj, koje vide. Kada stigne nov paket sa stanjem veze, on se poredi s listom već pristiglih paketa. Ako je zaista nov, on se prosleđuje na sve linije, osim na onu s koje je stigao. Ukoliko je u pitanju duplikat, paket se odbacuje. Ako ikada stigne paket s rednim

brojem koji je manji od najvećeg već viđenog rednog broja, on se odbacuje kao zastareo, pošto usmerivač ima svežije podatke.

Opisani algoritam ima mana, ali se one mogu ublažiti. Prvo, kada bi se redni brojevi ciklično ponavljali, nastala bi zabuna. Tu je rešenje da se koriste 32-bitni redni brojevi. Uz jedan paket sa stanjem veze u sekundi, niz takvih rednih brojeva iscrpao bi se tek posle 137 godina, pa ih za praktične svrhe možemo smatrati jedinstvenim.

Drugo, kada usmerivač otkaže, on gubi i evidenciju rednih brojeva. Ako ponovo počne da ih koristi od nule, njegov sledeći paket biće odbačen kao duplikat.

Treće, kada se redni broj ošteti, paumesto broja 4 stigne redni broj 65.540 (1-bitna greška), paketi s rednim brojem od 5 do 65.540 biće odbacivani kao zastareli, pošto usmerivači pogrešno zaključuju daje aktuelni redni broj 65.540.

Svi navedeni problemi mogu se resiti ako se iza rednog broja uključi i njegova „starost" - brojač čija će se vrednost svake sekunde smanjivati za jedan. Kada starost paketa dostigne nulu, odbacuju se informacije od usmerivača koji je poslao paket. Nov paket stiže normalno svakih, recimo, 10 sekundi, tako će informacije biti odbačene zbog starosti samo kada usmerivač ne radi (ili kada se izgubi šest uzastopnih paketa, što je malo verovatno). Osim toga, i svaki usmerivač smanjuje vrednost polja *Starost* tokom početnog plavljenja da se nijedan paket ne bi izgubio i beskonačno lutao mrežom (paket čija je starost nula, odbacuje se).

Robusnost ovog algoritma može se povećati izvesnim podešavanjima. Kada paket sa stanjem veze stigne u usmerivač koji treba da ga emituje mehanizmom plavljenja, on se ne stavlja odmah u red čekanja za slanje, već se za izvesno vreme srnešta u „čekaonicu". Ako drugi paket sa stanjem veze koji potiče sa istog izvorišta stigne pre nego što se pošalje prvi paket, porede se njihovi redni brojevi. Ako su isti, duplikat se odbacuje. Ako se razlikuju, odbacuje se stariji. Da bi se predupredile greške na linijama između usmerivača, traži se potvrda za svaki paket sa stanjem veze. Kada se linija isprazni, čekaonica se ciklično proverava da bi se odabrao paket ili potvrda za slanje.

Struktura podataka koju koristi usmerivač *B* za podmrežu sa slike 5-13(a) prikazana je na slici 5-14. Tu svaki red odgovara nedavno primljenom, još uvek nepotpuno obrađenom paketu sa stanjem veze. U tabelu se beleži gde je paket nastao, zatim njegov redni broj, njegova starost i podaci. Osim toga, tu su i indikatori za slanje i potvrdu za svaku od tri linije usmerivača *B* (redom za linije A, C i *F).* Indikator za slanje u nekoj od linija znači da se paket mora poslati tom linijom. Indikator za potvrdu znači da preko te linije mora biti potvrđen.

Na slici 5-14, paket sa stanjem veze koji šalje A stiže direktno, tako da se mora poslati usmerivačima *C* i *F,* i potvrditi usmerivaču A, upravo kao što pokazuju indi- katorski bitovi. Slično tome, paket od *F* treba proslediti usmerivačima A i C. i potvrditi usmerivaču *F.*

Indikatori za slanje		Indikatori za potvrdu			Izvorište	Redni broj	Starost	A	C	F	A	C	F	Podaci
A		21	60			0	1	1	1	0	0			
F		21	60			1	1	0	0	0	1			
E		21	59			0	1	0	1	0	1			
C		20	60			1	0	1	0	1	0			
D		21	59			1	0	0	0	1	1			

Slika 5-14. Čekaonica - bafer paketa kod usmerivača *B* sa slike 5-13.

Međutim, situacija je drugačija s trećim paketom koji dolazi od *E.* On je stigao dva puta, prvi put putanjom *EAB,* a drugi put putanjom *EFB.* Zbog toga, njega treba poslati samo usmerivaču C, ali ga treba potvrditi usmerivačima A i *F,* kao što pokazuju indikatorski bitovi.

Ako stigne duplikat dok se original još nalazi u baferu, indikatorski bitovi se moraju menjati. Na primer, ukoliko kopija stanja usmerivača C stigne od usmerivača *F* pre nego što se prosledi četvrta odrednica tabele, šest indikatorskih bitova treba pro- meniti u 100011 i time naznačiti da paket treba potvrditi usmerivaču *F,* ali mu ga ne treba slati.

Izračunavanje novih putanja

Kada usmerivač prikupi potpun skup paketa sa stanjem veza, on može da kon- struiše graf celokupne podmreže jer je podacima predstavljena svaka veza. Svaka veza je u stvari predstavljena dvaput, po jednom za svaki smer. Te dve vrednosti mogu se koristiti zasebno ili se može računati s njihovim prošekom.

Sada se Dijkstrin algoritam može upotrebiti lokalno da bi se konstruisale najkraće putanje do svih mogućih odredišta. Rezultati koje on pruži mogu se instalirati u tabele za usmeravanje, a zatim nastaviti s normalnim radom.

Za podmrežu sa *n* usmerivača, od kojih svaki ima *k* suseda, memorija neophodna za skladištenje ulaznih podataka proporcionalna je *kn.* To može da predstavlja problem u velikim podmrežama. Može da bude problem i trajanje izračunavanja. Pa ipak, u mnogim praktičnim situacijama, usmeravanje na osnovu stanja veze radi dobro.

Međutim, problemi s hardverom i softverom mogu da naprave zbrku sa ovim algoritmom (kao i s drugim algoritmima). Na primer, ako usmerivač tvrdi da ima liniju koju u stvari nema ili zaboravi na liniju koju ima, graf podmreže biće pogrešan. Ako usmerivač ne prosledi pakete ili ih oštеti dok ih prosleđuje, nastaje problem. Najzad, ako mu se prepuni memorija ili pogrešno izračuna usmeravanje, takođe će nastati problemi. Kako mreža narasta i počinje da obuhvata desetine i stotine hiljada čvorova, verovat- noća da će neki od njih povremeno otkazati prestaje da bude zanemarljiva. Rešenje leži u predviđanju mera za ograničavanje štete kada se neizbežno dogodi. Perlmann (1988) detaljno opisuje pomenute probleme i rešenja za njihovo ublaživanje.

Usmeravanje zasnovano na stanju veze široko se koristi u današnjim mrežama, pa treba nešto reći o protokolima koji ga koriste. Protokol OSPF, koji se često koristi na Internetu, radi sa algoritmom usmeravanja zasnovanim na stanju veze. Opisaćemo ga u odeljku .5.6.4.

Drugi takav protokol je protokol za vezu između međusistema (engl. *Intermediate System-Intermediate System, IS-IS),* koji je projektovan za saradnju s mrežnim protokolom DECnet, ali ga je kasnije prihvatila organizacija ISO za saradnju s njenim protokolom CLNP - mrežnim protokolom koji radi bez uspostavljanja direktne veze. Otada je više puta modifikovan da bi mogao da sarađuje i s drugim protokolima, naročito s protokolom IP. Protokol IS-IS koristi se u nekim okosnicama Interneta (uključujući i okosnicu mreže NSFNET) i u nekim digitalnim sistemima mobilne te- lefonije, npr. uz protokol CDPD. Novell NetWare koristi skraćenu varijantu protokola IS-IS (NLSP) za usmeravanje IPX paketa.

Protokol IS-IS u načelu distribuira sliku topologije usmerivača, iz koje se izračunavaju

najkraće putanje. U svojim informacijama o stanju veze svaki usmerivač objavljuje adrese mrežnog sloja kojima može da direktno pristupi. To mogu da budu IP adrese, IPX, AppleTallc adrese ili bilo koje druge. IS-IS čak može da istovremeno podrži više protokola mrežnog sloja.

Mnoge inovacije projektovane za IS-IS usvojene su i za protokol OSPF (koji je projektovan više godina posle njega). Među njima su postupale ažuriranja plavljenjem koji se sam stabilizuje, koncepcija namenskog usmerivača u lokalnoj mreži, i postupale izračunavanja i podržavanja razdvajanja putanja i višestruke metrike. Zbog toga su protokoli IS-IS i OSPF veoma slični. Najhitnija razlika između njih je u tome što je IS-IS kodiran na način da lako i prirodno istovremeno prenosi informacije o više protokola mrežnog sloja, što protokol OSPF ne može. Ta prednost naročito dolazi do izražaja u okruženjima u kojima se izvršava više protokola.

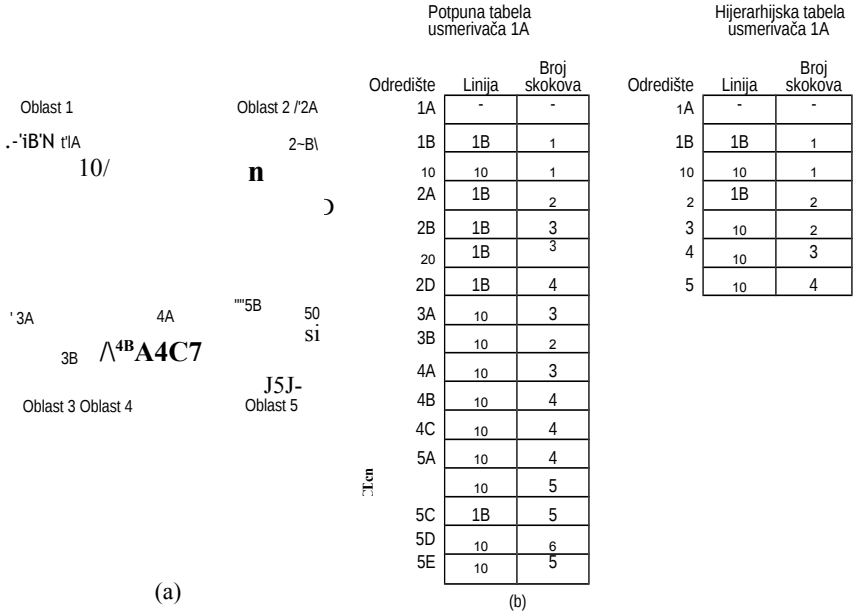
5.2.6 **Hijerarhijsko usmeravanje**

Kako raste mreža, tako rastu i tabele u usmerivačima. Zbog toga se intenzivnije angažuje memorija usmerivača, ali se isto tako više troši i procesorsko vreme za obradu tabela, a potreban je i veći propusni opseg za razmenu informacija o stanju veze. Mreža u određenom trenutku može toliko narasti da neće svaki usmerivač moći da vodi evidenciju o svim dragim usmerivačima, pa se usmeravanje mora organizovati hijerarhijski, kao u telefonskoj mreži.

Kada se primeni hijerarhijsko usmeravanje, usmerivači se dele u oblasti (engl. *re- gions*) i svaki usmerivač tad zna sve o usmeravanju u njegovoj oblasti, ali ništa o internoj strukturi drugih oblasti. Kada se povezuju različite mreže, prirodno je da svaka od njih postane posebna oblast kako usmerivači iz jedne mreže ne bi morali da „uče“ struktura svih dragih mreža.

Ako je mreža zaista velika, možda za usmeravanje neće biti dovoljna dvostepena hijerarhija; možda će biti potrebno da se oblasti grupišu u skupine, ove u zone, a zone u grupe itd, sve dok nam ne ponestane imena za pojedine stupnjeve hijerarhije. Kao primer višestepene hijerarhije razmotrimo usmeravanje paketa iz Berklija u Kalifor- niji u Malindi u Keniji. Usmerivač na Berkliju znaće detaljno topologiju mreže u Ka- liforniji, ali će sav saobraćaj za „inostranstvo“ upućivati usrnerivaču u Los Anđelesu. Taj usmerivač može da usmerava saobraćaj unutar SAD, ali pakete narnenjene drugim zemljama šalje u Njujorlc. Njujorški usmerivač je programiran tako da sav saobraćaj uputi onom usrnerivaču u drugoj državi koji je odgovoran za međunarodni saobraćaj, a to je u našem primeru, recimo, usmerivač u Najrobiju. Na kraju, paket pronalazi svoj put niz hijerarhijsko stablo, sve dok ne stigne u Malindi.

Slika 5-15 kvantitativno prikazuje usmeravanje u dvostepenoj hijerarhiji s pet oblasti. Potpuna tabela usmerivača 1A ima 17 odrednica, kao na slici 5-15(b). Kada je usmeravanje hijerarhijsko, kao na slici 5-15(c), postoje odrednice za sve lokalne usme- rivače, kao i ranije, ali su sve ostale oblasti skupljene u jednom usrnerivaču, tako da sav saobraćaj za oblast 2 ide linijom *IB-2A,* ali saobraćaj za drage oblasti ide linijom *1C-3B.* Hijerarhijskim usmeravanjem broj odrednica u tabeli smanjio se sa 17 na 7. Kako raste odnos broja oblasti i broja usmerivača po jednoj oblasti, sve veće su i uštede u tabelama.



Slika 5-15. Hijerarhijsko usmeravanje.

Ušteda prostora u tabelama, nažalost, nije besplatna. Za to postoji cena, a cena je veća dužina puta. Na primer, najbolja putanja između usmerivača *lA* i *5C* ide preko oblasti 2, ali u hijerarhijskom usmeravanju sav saobraćaj za oblast 5 ide preko oblasti 3, pošto je to povoljnije

za većinu odredišta u oblasti 5.

Kada jedinstvena mreža postane veoma velika, postavlja se pitanje broja stupnjeva hijerarhije. Razmotrite, na primer, podmrežu sa 720 usmerivača. Bez hijerarhijskog uređivanja, svaki usmerivač bi imao tabelu sa 720 odrednica. Ukoliko se podmreža podeli na 24 oblasti sa po 30 usmerivača, svaki usmerivač će imati tabelu sa 30 lokalnih odrednica i 23 udaljene odrednice, što ukupno čini 53 odrednice. Ako se izabere trostepena hijerarhija sa 8 skupina od po 9 oblasti, a svaka oblast sadrži 10 usmerivača, tabela svakog usmerivača imaće 10 lokalnih odrednica, 8 oblasnih odrednica unutar sopstvene skupine i 7 odrednica za udaljene skupine, što ukupno iznosi 25 odrednica. Utvrđeno je (Kamoun i Kleinrock, 1979), da optimalan broj nivoa podmreže sa A" usmerivača iznosi In/V, kada tabele usmerivača sadrže *e* ln*N* odrednica. Autori su takođe pokazali da je povećanje dužine putanja izazvano hijerarhijskim usmerava- njem u granicama prihvatljivog.

5.2.7 **Realizovanje neusmerenog emitovanja**

U nekim situacijama, računar treba da pošalje poruke mnogim, ako ne i svim dragim računarima. Tako se, na primer, distribuiraju meteorološki izveštaji, promene stanja na berzi, a i radio programi tako rade najbolje, pri čemu se zainteresovanim

korisnicima omogućava da po želji i čitaju vesti. Istovremeno slanje paketa na sva odredišta naziva se **neusmereno (difuzno) emitovanje** (engl. *broadcasting*); postoji više načina da se ono realizuje.

Jedan metod, koji podmreži ne postavlja nikakve dodatne zahteve, sastoji se u tome da pošiljalac po jedan paket pošalje na svako odredište. Taj metod ne samo da opterećuje propusni opseg, već i zahteva da pošiljalac zna adrese svih odredišta. On će u praksi ponekad biti i jedina mogućnost, ali je ipak najmanje poželjan.

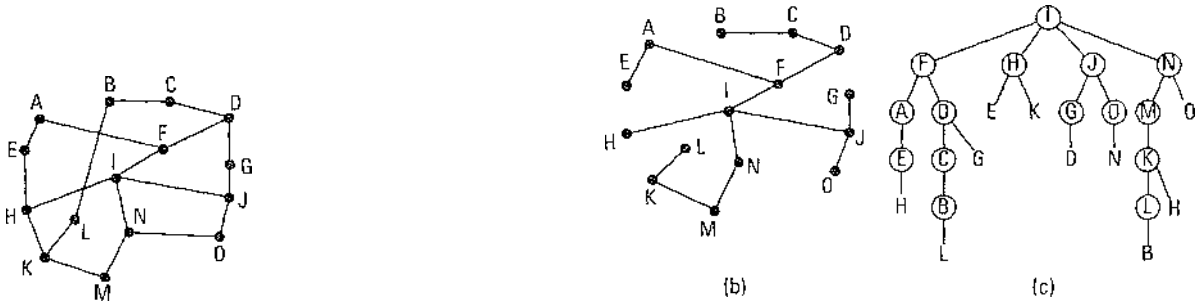
Druga očigledna mogućnost je plavljenje. Iako ono nije prvenstveno namenjeno komunikaciji od tačke do tačke, treba ga ozbiljno izeti u obzir kada se razmatra difu- zno emitovanje, naročito ako se ne može primeniti nijedna od metoda koje opisujemo u nastavku. Mana plavljenja pri difuznom emitovanju ista je kao u komunikaciji od tačke do tačke: generiše se previše paketa i zauzima veliki propusni opseg.

Treći metod je **usmeravanje na više odredišta** (engl. *multidestination routing*). Kada se on koristi, svaki paket sadrži listu odredišta ili bit mapu koja ukazuje na željena odredišta. Kada paket stigne u usmerivač, ovaj proverava odredišta i utvrđuje sve potrebne izlazne linije. (Potrebno je nameniti liniju ako je barem za jedno odredište to najbolja putanja.) Usmerivač generiše novu kopiju paketa za svaku odabranu izlaznu liniju i u svaku kopiju upisuje odredišne adrese koje se dosežu samo preko te linije. Tako se skup odredišnih adresa raspodeljuje po izlaznim linijama. Posle određenog broja skokova, svaki paket će nositi adresu samo jednog odredišta i biće tretiran kao običan paket. Usmeravanje na više odredišta slično je usmeravanju posebno adresiranih paketa, osim što u situaciji kada više paketa moraju da idu istom putanjom, jedan od njih plaća ceh za sve ostale.

U četvrtom algoritmu za neusmereno emitovanje direktno se koristi stablo optimalnih putanja usmerivača koji započinje emitovanje - ili i bilo koje drugo pogodno stablo optimalnih putanja. **Razgranato stablo** (engl. *spanning tree*) predstavlja podskup podmreže koji obuhvata sve usmerivače, ali ne sadrži petlje. Ako svaki usmerivač zna koja od njegovih linija pripada razgranatom stablu, on svaki neusmereno emitovan dolazni paket može da kopira na sve linije razgranatog stabla, osim na onu s koje je došao. Postupale izuzetno štedi propusni opseg, generišući apsolutno najmanji broj neophodnih paketa. Nezgodno je samo to što svaki usmerivač mora da zna za neko razgranato stablo. Takvi podaci ponekada postoje (kao pri usmeravanju na osnovu stanja veze), ali nekada i ne postoje (kao pri usmeravanju na osnovu vektora razdaljine).

Na kraju, pokušajmo da realizujemo neusmereno emitovanje oslanjajući se na prethodni algoritam, u situaciji kada usmerivači ne znaju ništa o razgranatim stablima. Ideja, tzv. **prosleđivanje paketa ispitivanjem izvorišta** (engl. *reverse path forwarding),* izuzetno je jednostavna kada je jednom shvatite. Kada neusmereno emitovan paket stigne u usmerivač, usmerivač proverava da li je paket stigao linijom koja se obično koristi za slanje paketa *izvorištu* neusmerenog emitovanja. Ako je to tačno, postoji *velika* šansa daje neusrnereni paket slučajno pronašao najbolju putanju i daje do usmerivača stigla njegova prva kopija. Zaključujući tako, usmerivač prosleđuje kopije paketa na sve linije osim na onu kojom je paket stigao. Ako je, međutim, paket stigao linijom koju usmerivač ne smatra najboljom putanjom do izvorišta, paket se odbacuje pod sumnjom da je duplikat.

Primer prosleđivanja paketa uz ispitivanje izvorišta prikazan je na slici 5-16. Deo (a) prikazuje podmrežu, deo (b) prikazuje stablo optimalnih putanja za usmerivač *I* iste podmreže, a deo (c) prikazuje kako radi prosleđivanje ispitivanjem izvorišta. U prvom skoku, *I* šalje pakete usmerivačima *F, H, J* i *N,* kao što pokazuje drugi red stabla. Svaki paket stiže do *I* najboljom putanjom (pretpostavljajući da najbolja putanja pripada stablu optimalnih putanja) i zato je označen kružićem oko slova. U drugom skoku generiše se osam paketa, po dva na svakom usmerivaču koji je primio paket u prvom skoku. Ispada da svaki od osam paketa stiže na dotada nepoznat usmerivač, pa zato pet od njih slučajno stižu najboljim putanjama. Od šest paketa generi- sanih u trećem skoku, samo tri stižu najboljim putanjama (usmerivačima *C, E* i *K);* ostali su duplikati. Posle pet skokova i 24 paketa, difuzno emitovanje se prekida, pri čemu su četiri skoka i 14 paketa tačno sledili stablo optimalnih putanja.



Slika 5-16. Prosleđivanje paketa ispitivanjem izvorišta, (a) Podmreža, (b) Stablo optimalnih putanja, (c) Stablo izgrađeno ispitivanjem izvorišta.

Osnovna prednost usmeravanja ispitivanjem izvorišta jeste to što je prilično efikasno i lako se realizuje. Za njegovu primenu usmerivači ne moraju da znaju za razgranata stabla, niti paketi moraju da sa sobom vuku nekoristan teret (listu odredišta, odnosno bit mapu), kao pri usmeravanju na više odredišta. Nije potreban specijalan mehanizam za prekidanje procesa, kao kod plavljenja (brojač skokova u svakom paketu i prethodno poznavanje opsega mreže, ili lista već viđenih paketa na svakom izvorištu).

5.2.8 Višesmerno usmeravanje

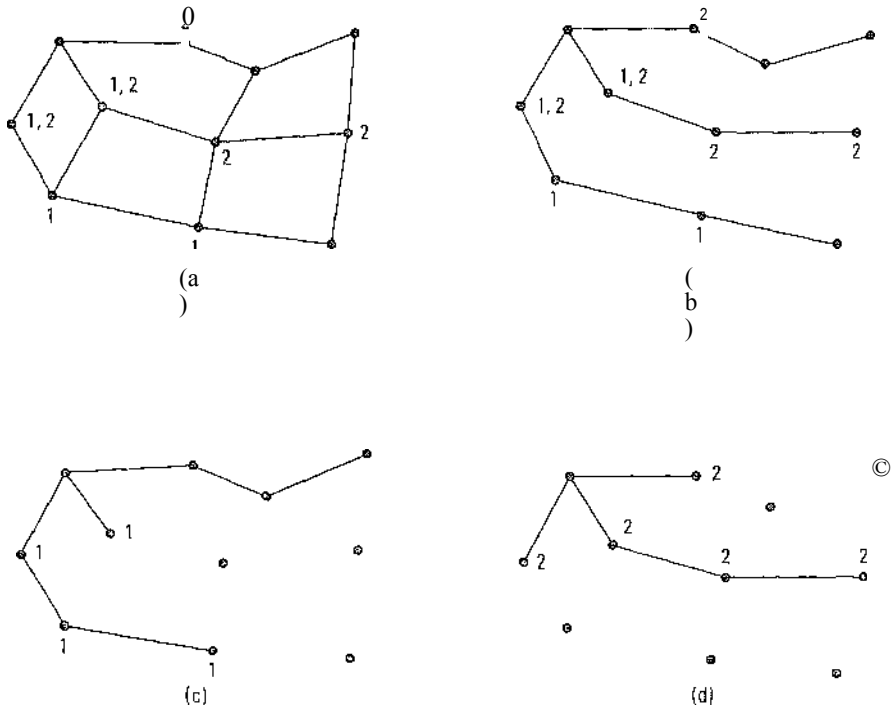
Za neke primene neophodan je zajednički, grupni rad više međusobno udaljenih procesa, kao pri realizovanju sistema distribuirane baze podataka. U takvim situacijama, često je neophodno da jedan proces pošalje poruku svim članovima grupe. Ako je grupa mala, on može svakom članu da pošalje poruku mehanizmom od tačke do tačke, ali kada je grupa velika, to se ne isplati. Ponekada se za to može koristiti i neus- mereno emitovanje, ali upotrebiti ga kada treba informisati 1000 računara u mreži od milion čvorova izuzetno je neefikasno jer većina primalaca neće biti zainteresovana za poruku (ili još gore, biće zainteresovani, a ne bi trebalo daje dobiju). Zato nam treba način da šaljemo poruke dobro definisanoj gmpi koja je u apsolutnom smislu ve- lika, ali ipak mala u odnosu na mrežu.

Slanje poruke takvoj grupi naziva se višesmerno emitovanje (engl. *multicasting),* a odgovarajući algoritam - algoritam za višesmerno usmeravanje (engl. *multicast routing).* U ovom odeljku razmotrićemo jednu njegovu varijantu; dodatna objašnjenja potražite kod Chua i suradnika (2000), Coste i saradnika (2001), Kasere i suradnika (2000), Madruge i Garcia-Luna-Acevesa (2001), kao i kod Zhanga i Ryua (2001).

Višesmerno emitovanje zahteva rad s grupom. Mora postojati mehanizam za pravljenje i raspuštanje grupa, kao i mehanizam kojim se procesi u njih smeštaju, odnosno iz njih uklanjaju. Kako se to ostvaruje, ne tiče se algoritma za usmeravanje. Njega više zanima to da lije proces, koji je pristupio određenoj grupi, o tome obavestio svoj računar, jer je važno da svi usmerivači znaju koji računar pripada kojoj grupi. Računali će o tome sami obavestiti svoje usmerivače ili će usmerivači morati povremeno da ih anketiraju. Ovako ili onako, usmerivači na kraju saznaju koji od njihovih raču- nara spada u koju grupu. O tome usmerivači obaveštavaju svoje susede, tako da se vest širi kroz podmrežu.

Da bi mogao da šalje pakete u više smerova, svaki usmerivač konstruiše razgra- nato stablo koje obuhvata sve druge usmerivače. Na primer, na slici 5-17(a) imamo dve grupe, 1 i 2. Kao što vidite sa slike, neki usmerivači su povezani s računalima koji pripadaju jednoj ili obema grupama. Razgranato stablo za usmerivač na levom kraju podmreže prikazano je na slici 5-17(b).

Kada proces pošalje „višesmerni“ paket svim članovima grupe, prvi usmerivač ispita svoje razgranato stablo i s njega „okreše“ sve grane (linije) koje ne vode ka nekom članu grupe. U ovom primeru, slika 5-17(c) prikazuje tako okresano stablo za grupu 1, a slika 5-17(d) za grupu 2. Višesmerni paketi usmeravaju se samo duž preostalih ogranaka proređenog razgranatog stabla.



Slika 5-17. (a) Mreža, (b) Razgranato stablo za krajnji levi usmerivač. (c) Višesmerno stablo za grupu 1. (d) Višesmerno stablo za grupu 2.

Stablo se može prorediti na razne načine. Najjednostavnije je ako se izvršava algoritam usmeravanja na osnovu stanja veze, tako da svaki usmerivač zna kompletnu to- pologiju, uključujući i to koji računar spada u koju grupu. Tada se iz razgranatog stabla, napredujući svakom putanjom od njenog kraja prema korenu stabla, mogu uklanjati usmerivači koji ne spadaju u odgovarajuću grupu.

Kada se paketi usmeravaju na osnovu vektora razdaljine, stablo se proređuje drugačije. Osnovni algoritam je usmeravanje ispitivanjem izvorišta. Međutim, kad god usmerivač koji nema računare zainteresovane za određenu grupu i nema veza s drugim usmerivačima, primi višesmernu poruku za tu grupu, on odgovara porukom PRUNE („okreši“) i time saopštava pošiljaocu da mu više ne šalje poruke za tu grupu. Kad usmerivač čiji računari nisu članovi grupe primi takvu poruku preko svih linija, i on može da odgovori porukom PRUNE. Na taj način, podmreža se rekurzivno proređuje.

Algoritam ima moguću manu da se teško prenosi na veće mreže. Pretpostavimo da mreža ima *n* grupa, prosečno od po *m* članova. Za svaku grupu treba pamtiti *m* pro- ređenih razgranatih stabala, ukupno *mn* stabala. Kada ima mnogo velikih grupa, za pamćenje svih njihovih stabala potrebna je velika memorija.

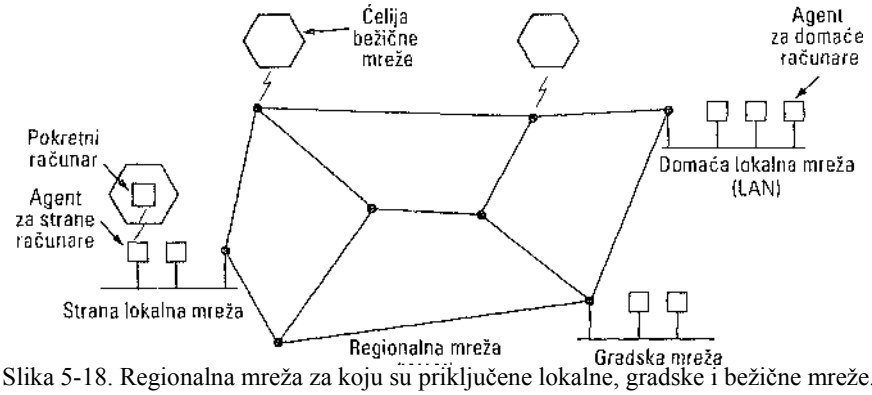
Alternativa je stablo zasnovano na korenu (engl. *core-based tree)* (Ballardie i sar., 1993). Ovde se konstruiše razgranato stablo, jedinstveno za grupu, čiji se koren (jezgro) nalazi približno u središtu grupe. Poruka namenjena svim članovima grupe šalje se korenu koji je zatim prosleđuje svakom članu grupe duž razgranatog stabla. Iako takvo stablo ne mora da bude uvek optimalno, smanjenje memorije neophodne za skladištenje podataka o stablima u odnosu *m:l* opravdava konstrukciju.

5.2.9 Usmeravanje za pokretne računare

Milioni ljudi imaju danas prenosive računare koje nose svuda sa sobom, a želeli bi da uvek mogu da čitaju svoju elektronsku poštu ili da pristupe svojim poslovnim datotekama ma gde se nalazili. Ti pokretni korisnici uzrok su nove komplikacije: da bi im se paket uručio, mreža mora najpre da zna gde se nalaze. Uključivanje pokretnih računara u mrežu još je sasvim mlada disciplina, ali ćemo u ovom odeljku skicirati uočene probleme i za njih ponuditi moguća rešenja.

Model sveta koji najčešće koriste projektanti mreža prikazanje na slici 5-18. Tu imamo regionalnu mrežu koja sadrži usmerivače i računare. Na nju su priključene lokalne mreže, gradske mreže i ćelije bežične mreže, kakve smo proučavali u 2. poglavlju.

Za računare koji nikada ne inenjaju mesto kaže se da su stacionarni (engl. *stationary hosts)*. Oni su s mrežom povezani pomoću bakarne žice ili optičkog kabla. Nasuprot njima, postoje još dve drugačije grupe računara. Migrirajući računari (engl. *migratory hosts)* u osnovi su stacionarni računari koji se povremeno premeštaju s jednog mesta na drugo, ali koriste mrežu samo kada su s njom fizički povezani. Lutajući (engl. *roaming)* računari rade doslovce u hodu i žele da su stalno na vezi. Izrazom pokretni računari (engl. *mobile hosts)* obuhvatićemo obe poslednje kategorije, tj. sve računare i srodne uređaje koji su daleko od kuće, ali žele da su s njom u vezi.



Slika 5-18. Regionalna mreža za koju su priključene lokalne, gradske i bežične mreže.

Pretpostavlja se da svi računari imaju svoje matične lokacije (engl. *home locati- on)* koja se nikada ne menjaju. Računari imaju i stalnu kucnu adresu preko koje se može doci do njihove matične lokacije, analogno načinu na koji telefonski broj 381-11-222-333 označava našu zemlju (381) i grad Beograd (11). Usmeravanje u sistemima s pokretnim računarima treba da omogući slanje paketa pokretnim računalima preko njihove matične adrese i da obezbedi njihovu efikasnu isporuku ma gde se računari nalazili. Prvi problem je, naravno, pronaći računate.

U modelu na slici 5-18, svet je podeljen (geografski) na male jedinice. Nazovimo ih područjima, pri čemu je područje obično lokalna mreža ili ćelija bežične mreže. Svako područje ima jednog ili više agenata za strane računare (*engl.forei.gn agents)* - procesa koji brinu o svim pokretnim računarima koji zađu u područje. Osim toga, svako područje ima i agenta za domaće računare (engl. *home agent),* koji brine o računarima čija je matična lokacija u području, ali se trenutno nalaze izvan njega.

Kad nov računar stupi u područje tako što se poveže na njega (npr. uključi se u lokalnu mrežu) ili samo zaluta u ćeliju, on mora da se registruje kod agenta za strance. Postupale registrovanja najčešće teče ovako:

1. Svi agenti za strance periodično u svim pravcima emituju paket kojim objavljuju svoje postojanje i svoju adresu. Pokretni računar koji je upravo stigao može da sačeka jednu od ovih poruka, ali ako ona ne stigne dovoljno brzo, on može da difuzno emituje paket s pitanjem da li u okolini postoje agenti za strance.
2. Pokretni računar se registruje kod agenta za strance, daje mu svoju matičnu adresu, tekuću adresu sloja veze podataka i izvesne podatke koji se tiču bez- bednosti.
3. Agent za strance stupa u vezu s agentom pokretnog računara koji je zadužen za domaće računare i kaže: jedan od tvojih računara je kod nas. Ta poruka sadrži i mrežnu adresu agenta za strance, kao i bezbednosne informacije koje udaljenog agenta za domaće računare treba da uvere da je njegov računar stvarno tamo gde se tvrdi da jeste.
4. Agent za domaće računare ispituje bezbednosne informacije koje sadrže vremensku oznaku kao dokaz da je poruka generisana u prethodnih nekoliko sekundi. Ukoliko je zadovoljan, on poručuje agentu za strance da nastavi s postupkom.

.5. Kada agent za strance dobije potvrdu od agenta za domaće računare, on u tabele unosi novu odrednicu i obaveštava pokretni računar da je registrovan.

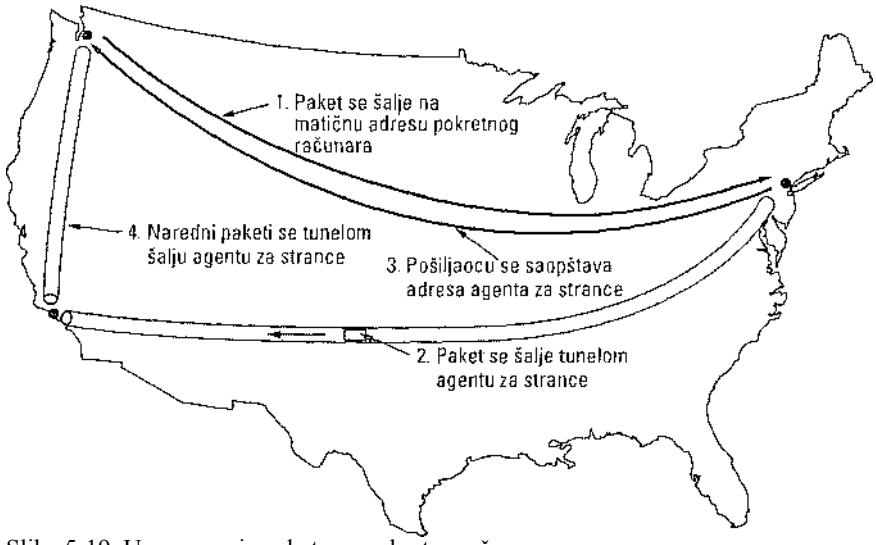
Idealno bi bilo da svaki računar koji napušta određeno područje to i objavi da bi bio izbrisan iz registra, ali se mnogi korisnici jednostavno isključe čim završe posao zbog koga su došli.

Kada se paket pošalje pokretnom računani, on se upućuje u njegovu matičnu lokalnu mrežu

jer je to ono što sledi iz adrese paketa, kao stoje prikazano korakom 1 na slici 5-19. Tu pošiljalac iz Sijetla na severozapadu želi da pošalje paket računaru koji se nalazi na drugoj strani SAD, u Njujorku. Pakete koji se pokretnom računaru šalju na matičnu adresu njegove lokalne mreže u Njujorku tamo presreće agent za domaće računare. On traži novu (privremenu) adresu pokretnog računara i pronalazi adresu agenta za strance u Los Anđelesu u čijoj je ona nadležnosti.

Agent za domaće računare tada radi dve stvari. Najpre, on kapsulira paket u polje za korisničke podatke većeg paketa i takav paket šalje agentu za strance (korak 2 na slici 5-19). To se zove upotreba tunela (engl. *tunneling)',* kasnije ćemo se time više pozabaviti. Pošto dobije kapsulirani paket, agent za strance vadi prvobitni paket iz polja za korisničke podatke i šalje ga pokretnom računaru kao okvir veze podataka.

Drugo, agent za domaće računare saopštava pošiljaocu da naredne pakete za pokretnog korisnika šalje kapsulirane u polje za korisničke podatke paketa eksplicitno adresiranih na agenta za strance, umesto da ih, kao do tada, šalje na matičnu adresu pokretnog računara (korak 3). Naredni paketi se sada mogu usmeravati direktno računaru preko agenta za strance (korak 4), potpuno zaobilazeći njegovu matičnu lokaciju.



Slika 5-19. Usmeravanje paketa za pokretne računare.

Razni predloženi sistemi razlikuju se međusobno na više načina. Prvo, pitanje je koji deo protokola treba da obavljaju usmerivači, a koji deo računari, a kod računara - i koji njihov sloj. Drugo, u nekoliko sistema, usmerivači duž puta beleže mapirane adrese tako da mogu da presretnu i preusmere pakete pre nego što i stignu do matične lokacije. Treće, u nekim sistemima se svakom posetiocu daje jedinstvena privremena adresa; u drugim sistemima, ta adresa odgovara agentu koji upravlja saobraćajem svih posetilaca.

Četvrto, sistemi se razlikuju po tome kako stvarno postižu da se paketi upućeni na jednu adresu isporuče na drugu. Jedna mogućnost je da se paketu promeni odredišna adresa i da se tako izmenjen paket ponovo pošalje. Paket se, alternativno, zajedno s matičnom adresom može kapsulirati u polje za korisničke podatke drugog paketa i poslati na privremenu adresu. Sistemi se razlikuju i sa aspekta bezbednosti. Kada računar ili usmerivač dobiju poruku tipa „Svu poštu za Milicu od sada šalji meni“, oni u načelu mogu da postave par pitanja o tome s kim

razgovaraju i da li je taj predlog razuman. Osobine više protokola za usmeravanje saobraćaja na pokretne računare opisali su i uporeedili Hac i Guo (2000), Perkins (1998a), Snoeren i Balakrishnan
(2000) , Solomon (1998), te Wang i Chen (2001).

5.2.10 Usmeravanje u ad hoc mrežama

Dosad smo videli kako se saobraćaj usmerava kada su računari pokretni, a usmerivači fiksni. Još ekstremniji je slučaj kada su i usmerivači pokretni. Takve potrebe imaju:

1. Vojna vozila na bojnom polju na kome nema nikakve infrastukture.
2. Flote brodova na moru.
3. Osoblje koje otklanja posledice zemljotresa koji je razorio infrastrukturu.
4. Skupovi ljudi s prenosivim računarima u području bez mreže 802.11.

U svim navedenim, a i drugim sličnim slučajevima, svaki čvor ima i usmerivač i računar, obično u istom uređaju. Mreže čvorova koji se slučajno nađu jedan do drugog zovu se ad hoc mreže (engl. *ad hoc networks)* ili mobilne ad hoc mreže (engl. *Mobile Ad hoc NETworks, MANE'Ts).* Razmotrimo ih ukratko. Više informacija o njima možete naći kod Perkinsa (2001).

Ad hoc mreže i kablovske mreže razlikuju se po tome što uobičajena pravila o fiksnim topologijama, fiksnim i poznatim susedima, fiksnim odnosima između IP adresa i lokacija, i štošta drugo što važi za kablovske mreže možete da bacite kroz prozor. Usmerivači se mogu pojavljivati i nestajali ili pojavljivati na novim mestima tokom trajanja jednog bita. U kablovskim mrežama, ako usmerivač zna pravu putanju do nekog odredišta, ona je uvek ispravna (osim u slučaju nekog otkazivanja u sistemu). U ad hoc mreži, topologija sve vreme može da se menja, tako da se i poželjnost određene putanje, čak i njena ispravnost, mogu nekontrolisano menjati, bez ikakvog upozorenja. Skoro da nema potrebe reći da takve okolnosti čine usmeravanje u ad hoc mrežama potpuno drugačijim od usmeravanja u kablovskim mrežama.

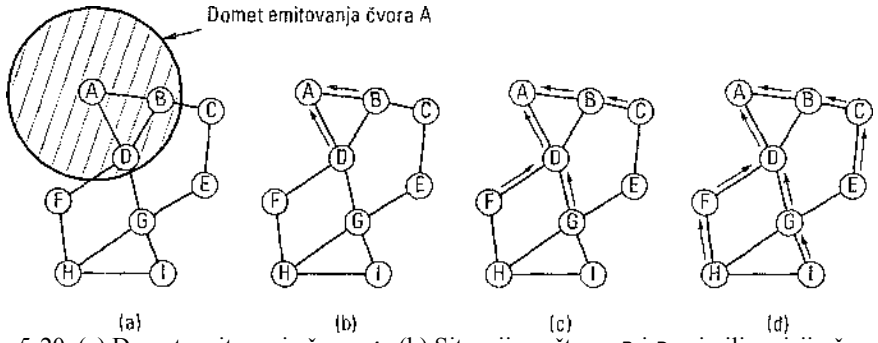
Predloženo je više algoritama za usmeravanje u ad hoc mrežama. Jedan od pri- vlačnijih je **usmeravanje na zahtev zasnovano na vektoru razdaljine** (engl. *Ad hoc On-demand Distance Vector, AODV),* koji su razvili Perkins i Royer (1999). On je daleki rođak Belman-Fordovog algoritma zasnovanog na vektoru razdaljine, ali je prilagođen za rad u okruženju pokretnih računara i uzima u obzir ograničen propusni opseg i vek trajanja njihovih baterija. Neobično je i to što algoritam radi „na zahtev“, tj. određuje putanju do nekog odredišta samo ako neko poželi da tamo pošalje paket. Pogledajmo šta to znači.

Otkrivanje putanje

Ad hoc mreža može da se u bilo kom trenutku opiše grafom čvorova (usmerivači + računali). Dva čvora su spojena (između njih na grafu postoji luk) ako direktno međusobno komuniciraju radiom. Pošto jedan od njih može da ima jači predajnik, postoji mogućnost da A bude u vezi sa *B,* ali da *B* ne bude u vezi sa A. Međutim, zbog jednostavnosti, pretpostavićemo da su sve veze simetrične. Treba naglasiti da iz činjenice što su dva čvora jedan drugom u dometu ne proizlazi da su i povezani. Između njih mogu postojati zgrade, uzvišenja i druge prepreke koje blokiraju komunikaciju.

Algoritam čemo najbolje objasniti pomoću ad hoc mreže sa slike 5-20, u kojoj proces u čvoru A želi da pošalje paket čvoru /. Algoritam AODV u svakom čvoru održava tabelu sređenu po

odredištima, dajući informacije o njima, uključujući i to kojim suse- dima treba slati pakete da bi stigli do odredišta. Pretpostavimo da A pretražuje svoju tabelu u kojoj ne nalazi odrednicu za 7. On sada mora da otkrije put do *I.* Ovo svojstvo otkrivanja putanje samo kada je potrebna odgovorno je za ime algoritma („na zahtev“).



Slika 5-20. (a) Domet emitovanja čvora *A*. (b) Situacija pošto su *B* i *D* primili emisiju čvora A. (c) Situacija pošto su *C, F\ G* primili emisiju čvora *A*. (d) Situacija pošto su *E, H* i *I* primili emisiju čvoraA. Zasenčeni čvorovi su novi prijemnici. Strelice označavaju moguće povratne putanje.

Da bi locirao 7, čvor A konstruiše specijalan paket ROUTE REQUEST (zahtev za putanju) i emituje ga difuzno. Paket stiže do čvorova *B* i *D,* kao što se vidi na slici 5-20(a). U stvari, to što su *B* i *D* na grafu povezani sa A znači da oni mogu da prime poziv upućen od njega. Čvor *F,* na primer, nije povezan lukom saA jer ne može da primi njegov radio signal. Na taj način, čvor *F* nije povezan sa čvorom A.

Format paketa ROUTE REQUEST prikazan je na slici 5-21. U njemu su izvorišna i odredišna adresa, tj. najčešće njihove IP adrese, koje identifikuju ko koga traži. Paket sadrži i *Identifikator zahteva* - lokalni brojač koji nezavisno održava svaki čvor, čija vrednost raste za jedinicu uvek kada čvor emituje zahtev ROUTE REQUEST. Uzeta zajedno, polja *Izvorišna adresa* i *Identifikator zahteva* jedinstveno identifikuju paket ROUTE REQUEST kako bi drugi čvorovi mogli da odbace eventualno primljene duplikate.

Izvorišna adresa	Identifikator zahteva	Odredišna adresa	Izvorišni redni broj	Odredišni redni broj	Broj skokova

Slika 5-21. Format paketa ROUTE REQUEST.

Osim *Identifikatora zahteva,* svaki čvor održava i drugi brojač čija se vrednost povećava za jedan svaki put kada se pošalje ROUTE REQUEST (ili se odgovori na nečiji ROUTE REQUEST). Njegova funkcija je pomalo slična satu i koristi se za razlikovanje novih i starih putanja. Četvrto polje na slici 5-21 predstavlja redni broj paketa na čvoru A; peto polje je najsvežija vrednost rednog broja paketa čvora / koju je zabeležio čvor A (ona je 0 ako je čvor A nikada nije registrovao). Namena ovih polja ubrzo će postati jasna. Poslednje polje, *Broj skokova,* vodi računa o tome koliko je skokova paket napravio. Ono se u početku postavlja na nulu.

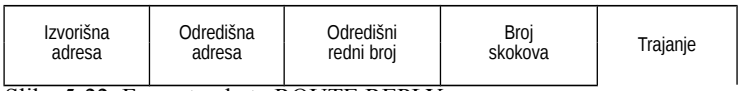
Kada paket ROUTE REQUEST stigne u čvor (u ovom slučaju, u čvorove *B* i *D),* on se

obrađuje kroz sledeće faze.

1. U tabeli sa istorijom lokalnih događaja traži se par: *Izvorišna adresa, Identifikator zahteva,* da bi se utvrdilo nije li možda takav zahtev ranije već stigao i bio obrađen. Ako je u pitanju duplikat zahteva, on se odbacuje i obrada se prekida. Ako je zahtev nov, u tabelu se beleže *Izvorišna adresa* i *Identifikator zahteva,* i obrada se nastavlja.
2. Primalac traži odredište u svojoj tabeli za usmeravanje. Ako je poznata sveža putanja do odredišta, pošiljaocu se vraća paket ROUTE REPLY (odgovor s putanjom) sa objašnjenjem kako da do njega stigne (odgovor je u načelu: Idi preko mene). Putanja je sveža ako je *Odredišni redni broj* u tabeli za usmeravanje veći ili jednak *Odredišnom rednom broju* iz paketa ROUTE REQUEST. Ako je manji, to znači da je putanja koju ima primalac starija od putanje koju ima pošiljalac, pa se prelazi na korak 3.
3. Pošto primalac ne zna svežu putanju do odredišta, on uvećava vrednost polja *Broj skokova* i ponovo difuzno emituje paket ROUTE REQUEST. On takođe prepisuje podatke iz paketa i čuva ih kao novu odrednicu u tabeli za usmeravanje ispitivanjem izvorišta. Pomoću tih podataka konstruisaće novu povratnu putanju kojom će kasnije poslati odgovor izvorištu. Povratne putanje su na slici 5-20 prikazane strelicama. Za odrednicu nove povratne putanje istovremeno se aktivira i tajmer. Ako se on automatski isključi, odrednica se briše.

Ni *B* ni *D* ne znaju gde se nalazi 7, tako da svaki od ta dva čvora upisuje odrednicu za povratnu putanju do *A* (kao što je prikazano strelicama na slici 5-20), i difuzno emituje paket u kome je vrednost polja *Broj skokova* uvećana za 1. Paketi od usmeri- vača *B* stižu usmerivačima *C* i *D.* Usmerivač C unosi za njega odrednicu u tabeli povratnih putanja i ponovo difuzno šalje paket. Nasuprot tome, usmerivač *D* ga odbacuje kao duplikat. Slično tome, *B* odbacuje neusmerene pakete od usmerivača *D.* Međutim, te pakete prihvataju *F* i *G* i skladište ih, kao na slici 5-2Q(c). Pošto neusmerena emisija stigne do usmerivača 7J, 77 i 7, paket ROUTE REQUEST konačno stiže na mesto gde znaju za 7, a to je sam usmerivač 7, kao sto je prikazano na slici 5-20(d). Iako smo neus- mereno emitovanje ovde prikazali u tri uzastopna koraka, imajte na umu da difuzno emitovanje čvorova ni na koji način nije koordinirano.

Kao odgovor na pristigli zahtev, 7 pravi paket ROUTE REPLY (slika 5-22). *Izvorima adresa, Odredišna adresa* i *Broj skokova* kopiraju se iz samog zahteva, ali se za *Odredišni redni broj* uzima vrednost iz memorije usmerivača. Polje *Broj skokova* postavlja se na nulu. Polje *Trajanje* određuje rok važnosti putanje. Ovaj paket se šalje samo čvoru od koga je stigao ROUTE REQUEST, sto je u ovom slučaju čvor *G.* Paket dalje sledi povratnu putanju do *D* i konačno, do *A.* Vrednost polja *Broj skokova* u svakom čvoru se uvećava za jedinicu, tako da čvor može da vidi koliko je udaljen od odredišta (7).



Slika 5-22. Format paketa ROUTE REPLY.

Na svom putu ka izvorištu, paket se pregleda na svakom usputnom čvoru. Podaci iz njega unose se u lokalnu tabelu za usmeravanje kao odrednica za putanju do 7 ukoliko je ispunjen barem jedan od sledećih uslova:

1. Nije poznata nijedna putanja do usmerivača 7.
2. Redni broj za 7 u paketu ROUTE REPLY veći je od vrednosti u tabeli za usmeravanje.

3. Redni brojevi su jednaki, ali je nova putanja kraća.

Na taj način, zahvaljujući potrazi koju sprovodi čvor *A,* svi čvorovi na povratnoj putanji besplatno dobijaju informaciju kako se stiže do 7. Čvorovi koji su dobili originalni paket ROLJTE REQUEST, ali se ne nalaze na povratnoj putanji (u primeru su to čvorovi *B, C,E,F* i 77), odbacuju odrednicu iz tabele povratnih putanja kada se odgovarajući tajmer automatski isključi.

U velikim mrežama, algoritam generiše mnoge difuzne emisije, čak i za bliska odredišta. Broj emisija se može smanjiti tako što pošiljalac inicijalizuje polje *Životni vek* IP paketa vrednošću koja odgovara očekivanom opsegu mreže, pri čemu ta vrednost opada za jedan pri svakom skoku. Kada ona dostigne nulu, paket se odbacuje.

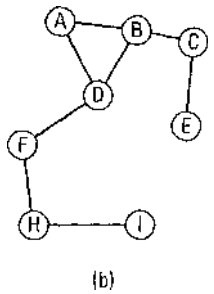
Postupak otkrivanja putanje tada se modifikuje na sledeći način. Da bi locirao odredište, pošiljalac neusmereno emituje paket ROUTE REQUEST s parametrom *Životni vek* podešenim na 1. Ako ne dobije odgovor unutar razumnog vremena, on šalje drugi paket, čiji je *Životni vek* 2. U narednim pokušajima, tom parametra daje vrednosti 3,4, 5 itd. Na taj način, prvo se pretražuje najbliža okolina, a zatim se krug sve više širi.

Održavanje putanje

Pošto se čvorovi mogu premeštati i isključivati, topologija se stalno menja. Na primer, na slici 5-20, ako se *G* isključi, *A* neće shvatiti da putanja koju koristi do / (*ADGl)* više ne važi. Algoritam treba da predupredi ovakve probleme. Svaki čvor periodično svima emituje poruku *Hello.* Očekuje se da svaki njegov sused odgovori na nju. Ako odgovor ne stigne, pošiljalac zna da se sused udaljio i da više nije s njim u vezi. Slično tome, ako pokuša da takvom susedu pošalje poruku, znače da mu on više nije na raspolaganju.

Pomoću takvih informacija odbacuju se putanje koje više ne važe. Svaki čvor *N,* za svako odredište, vodi evidenciju suseda koji su mu poslali paket za to odredište u toku poslednjih _*T* sekundi. To su aktivni susedi (engl. *active neighbors)* čvora *N* za određeno odredište. Čvor A to čini tako što tabelu za usmeravanje uređuje prema odredištima, a uz njih zapisuje izlazni čvor za njegovo dostizanje, broj skokova do njega, najsvežiji odredišni redni broj i listu svojih suseda za to odredište. U topologiji mreže iz našeg primera, čvor D bi mogao imati tabelu za usmeravanje kao na slici 5-23(a).

Sledeći Odredište	skok	Razdaljina	Aktivni susedi	Ostala polja
A	A	1	F, G	
B	B	1	F, G	
C	B	2	F	
E	G	2		
F	F	1	A, B	
G	G	1	A, B	
H	F	2	A, B	
1	G	2	A, B	



(b)

Slika 5-23. (a) Tabela za usmeravanje čvora `D` pre nego što se čvor G isključio, (b) Graf pošto se čvor `G` isključio.

Kada bilo koji sused čvora *N* iziđe iz dometa, *N* iz svoje tabele briše odredišta za koja su putanje išle preko tog suseda. Istovremeno obaveštava preostale aktivne su- sede da se do tog odredišta ne može više stici preko čvora *N,* pa ga i oni brišu iz svojih tabela. Aktivni susedi to saopštavaju svojim aktivnim susedima itd, sve dok iz svih tabela ne budu izbrisane odrednice koje zavise od nedostajućeg čvora.

Razmotrimo ponovo naš primer, ali pretpostavimo sada da se čvor *G* odjednom isključio. Tako izmenjena topologija prikazana je na slici 5-2.3(b). Kada *D* otkrije da čvora *G* više nema, on pregleda svoju tabelu za usmeravanje i pronalazi da se čvor *G* koristio na putanjama do čvorova *E,GiI.* Unija aktivnih suseda za ova odredišta jeste skup {*A, B*}. Drugim recima, *A* i *B* zavise od *G* u pogledu nekih svojih putanja, pa moraju biti obavešteni da te putanje više ne važe. Čvor *D* im stoga šalje pakete pomoću kojih će oni ažurirati svoje tabele za usmeravanje. Čvor *D* i iz svoje tabele briše odrednice koje se odnose na odredišta *E, G i I.*

Možda u našem opisu to nije dovoljno naglašeno, ali protokol AODV odudara od Belman-Fordovog algoritma zbog toga što čvorovi ne emituju periodično čitave svoje tabele za usmeravanje. Time se znatno štede propusni opseg i vek baterija.

Protokolom AODV može se vršiti i neusmereno i višesmerno usmeravanje. Detalje o tome potražite kod Perkinsa i Royera (2001). Ad hoc usmeravanje je trenutno vruća tema i o njoj ima dosta objavljenih radova. Neke od njih možete naći u spisku referenci na kraju knjige: Chen i saradnici (2002), Hu i Johnson (2001), Li i saradnici
(2001) , Raju i Garcia-Luna-Aceves (2001), Ramanathan i Redi (2002), Royer i Toh (1999), Spohn i Garcia-Luna-Aceves (2001), Tseng i saradnici (2001); Zadeh i saradnici (2002).

5.2.11 Pretraživanje čvorova u mrežama ravnopravnih računara

Mreže ravnopravnih računara srazmerno su nova pojava; tu veliki broj osoba, najčešće povezanih na Internet pomoću kablovskih priključaka, dolazi u međusobnu vezu deleći resurse. Prva masovna primena tehnologije ravnopravnih računara bila je zaobilaženje zakona: 50 miliona korisnika Napstera međusobno je razmenjivalo zaštićene muzičke numere bez odobrenja autora, sve dok nakon velike rasprave Napster nije zatvoren odlukom suda. Bez obzira na to, tehnologija ravnopravnih računara ima mnoge zanimljive, sasvim legalne primene. Ona ima nešto zajedničko i s problemima usmeravanja, iako oni nisu baš isto što i problemi s kojima smo se dosad susretali. Pa ipak, tehnologija ravnopravnih računara zavređuje da joj posvetimo malo vremena.

Sistemi ravnopravnih računara su zanimljivi zato što su potpuno distribuirani. Svi čvorovi su simetrični i ne postoji ništa što bi ličilo na centralno upravljanje ili hijerarhiju. U tipičnom sistemu ravnopravnih računara svaki korisnik ima neke informacije koje mogu zanimati ostale korisnike. To može da bude besplatan softver, muzika (u javnom vlasništvu), fotografije i slično. Ako ima mnogo korisnika, oni ne znaju jedan za drugog i ne znaju gde da pronađu ono što traže. Jedno rešenje može da bude velika centralna baza podataka, ali to može da bude neizvodljivo iz određenih razloga (npr. niko ne želi da baza bude locirana kod njega i da on mora da je održava). Tako se problem svodi na pitanje: kako da korisnik pronađe čvor koji sadrži ono što on traži ako ne postoji centralizovana baza podataka, čak ni centralizovan indeks?

Pođimo od toga da svaki korisnik ima određen sadržaj: muzičke numere, fotografije, programe, datoteke itd, koje bi dragi korisnici takođe voleli da imaju. Svakoj stavci je kao ime pridružen ASCII tekst. Potencijalni korisnik zna samo ime stavke, a želi da sazna da li jedna ili više osoba imaju njenu kopiju i ako je imaju, koje su njihove IP adrese.

Razmotrimo, primera radi, distribuiranu genealošku bazu podataka (rodoslov). Svaki genealog ima u elektronskom obliku određene zapise o svojim precima i rođacima, možda i s fotografijama, audio zapisima, čak i s video sekvencama. Više osoba mogu da imaju istog pradedu, tako da za istog pretka mogu postojati zapisi u različitim čvorovima. Ime zapisa je ime

dotičnog pretka u nekom definisanom obliku. U jednom trenutku, određeni genealog otkriva u arhivi pradedin testament kojim prade- da zaveštava zlatni džepni sat svom nećaku. Genealog tako saznaje necakovo ime i želi da zna ima li neki drugi genealog podatke o njemu. Kako ćemo bez centralne baze podataka saznati da li neko uopšte ima takve zapise, a kamoli ko je to?

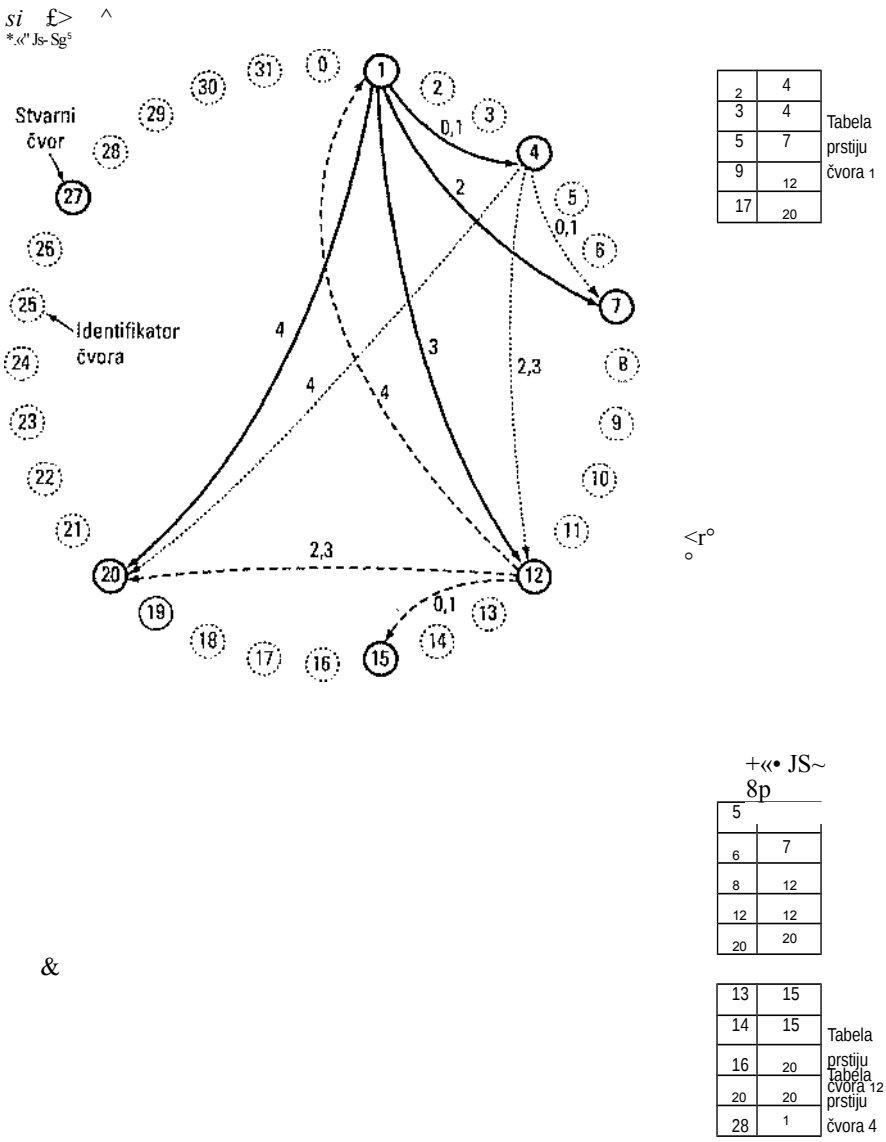
Za rešavanje opisanog problema predlagani su brojni algoritmi, a mi ćemo opisati Chord (Dabek i sar., 2001a; Stoica i sar., 2001). Sledi jednostavno objašnjenje njegovog rada. Sistem Chord obuhvata *n* učesnika, od kojih svaki može da ima uskladištene zapise i voljan je da skladišti bitove i delove indeksa koje će upotrebljavati drugi korisnici. Svaki korisnički čvor ima svoju IP adresu koja se može heširati u m-bitni broj pomoću odgovarajuće *hash* funkcije. Za generisanje *hash* funkcije Chord koristi algoritam SHA-1. Taj algoritam se koristi u kriptografiji i na njega ćemo se vratiti u 8. poglavlju. Zasad prihvatite daje *hash* funkcija koja niz bajtova promenljive dužine pretvara u 160- -bitni broj s nasumičnim rasporedom bitova. Na taj način, svaku IP adresu možemo da prevedemo u 160-bitni broj zvan identifikator čvora (engl. *node identifier)*.

Koncepcijski je svih 2^{160} identifilcatora čvorova raspoređeno rastućim redosledom po periferiji velikog kruga. Neki od njih odgovaraju čvorovima učesnika, ali većina to nije. Na slici 5-24(a) prikazujemo krug identifilcatora čvorova za $m = 5$ (za trenutak zanemarite lukove unutar kruga). U našem primeru, čvorovi sa identifikatorima 1, 4, 7, 12, 15, 20 i 27 odgovaraju stvarnim čvorovima i zato su zasenčeni; ostali ne postoje.

Definišimo sada funkciju *sledbenik(k)* kao identifikator prvog stvarnog čvora koji sledi *k* kada se po krugu krećemo u smeru kazaljke. Na primer, *sledbenik(6)* - 7, *sled- benik(S)* = 12, a *sledbenik(22*) = 27.

Imena zapisa (muzičkih numera, predaka itd.) talcođe se heširaju *hash* funkcijom (tj. algoritmom SHA-1) u 160-bitni broj zvan ključ (engl. *key).* Tako, da biste *ime* (ASCII ime zapisa) preveli u njegov ključ, upotrebićete izraz *ključ = hash(ime).* To izračunavanje se obavlja lokalnim pozivanjem procedure *hash.* Ako osoba koja ima genealoški zapis *ime* želi da ga svima ponudi, ona prvo pravi dvokomponentni podatak - dublet ili tapl (engl. *tuple),* u ovom slučaju to je dublet *(ime, moja_IP_adresa)* - i poziva funkciju *sledbenik(hash(ime))* da uskladišti dublet. Ako za ovo ime postoji više zapisa (u različitim čvorovima), svi njihovi dubleti biće uskladišteni u istom čvoru. Na ovaj način, indeks se nasumično distribuira po čvorovima. U cilju predu-pređivanja gešaka, može se koristiti *p* različitih funkcija za heširanje u *p* čvorova, ali na ovom mestu nećemo dalje zalaziti u detalje.

Ako neki korisnik kasnije želi da potraži *ime,* on ga hešira da bi dobio *ključ,* a zatim koristi funkciju *sledbenik(ključ)* da bi pronašao IP adresu čvora koji čuva njegove indeksne dublete. Prvi korale je lak; drugi je teži. Da bi omogućio pronalaženje IP adrese čvora koji odgovara zadatom ključu, svaki čvor mora da održava izvesne administrativne strukture podataka. Jedna od njih je IP adresa čvora sledbenika duž kruga identifikatora čvorova. Na primer, na slici 5-24, sledbenik čvora 4 je čvor 7, a njegov sledbenik je čvor 12.



Slika 5-24. (a) Skup 32 identifikatora čvorova razmešten po krugu. Zasenčeni čvorovi odgovaraju stvarnim računarima. Lukovi su prsti koji se pružaju od čvorova 1, 4 i 12. Oznake na lukovima su indeksi iz tabele, (b) Primeri tabela prstiju.

Pretraživanje može sada da se odvija na sledeći način. Inicijalni čvor šalje sledbe- niku paket sa svojom IP adresom i ključem koji traži. Paket se prosleđuje duž prstena sve dok ne locira sledbenika identifikatora čvora koji se traži. Taj čvor proverava da li ima podatke koji odgovaraju ključu i ako ih ima, vrača ih direktno inicijalnom čvoru čiju IP adresu ima.

U prvoj optimizaciji opisanog postupka, svaki čvor bi mogao da čuva IP adrese kako svog sledbenika, tako i svog prethodnika, pa bi se upiti mogli slati i u smeru kazaljke i u suprotnom

smeru, u zavisnosti od toga šta se smatra kraćim putem. Na pri- mer, čvor 7 na slici 5-24 mogao bi da šalje upit za čvor 10 u smeru kazaljke, ali za čvor 3 u suprotnom smeru.

Čak i kada postoji izbor smera kretanja, linearno pretraživanje svih čvorova veoma je neefikasno u velikim sistemima ravnopravnih računara pošto prosečan broj čvorova po upitu iznosi *n/2*. Da bi se pretraživanje znatno ubrzalo, svaki čvor održava ono što se u žargonu algoritma Chord naziva tabela prstiju (engl. *finger table)*. Tabela prstiju ima *m* odrednica, indeksiranih od 0 do *m* - 1, a svaka ukazuje na dragi stvaran čvor. Svaka odrednica ima dva polja: *početak* i IP adresu čvora čiji je identi-

fikator *sledbenik(početak),* kao stoje na slici 5-24(b) prikazano za tri čvora. Vrednosti polja za odrednicu *i* u čvora *k* glase:

početak -k + *2′* (po modulu 2^m)
IP adresa čvora sledbenik(početak[i*])*

Obratite pažnju na to da svaki čvor skladišti IP adrese srazmerno malog broja Čvorova i da inu je većina tih čvorova prilično blizu (mereno brojem identifikatora).

Pomoću tabele prstiju, traženje *ključa* inicirano u čvora *k* teče ovako. Ako se *ključ* nađe između *k* i *sleđbenika(k),* tada podatke o *ključu* ima *sledbenik(k)* i pretraživanje se završava. U suprotnom, u tabeli prstiju traži se odrednica čije je polje *početak* najbliže prethodniku *ključa.* Zahtev se tada upućuje direktno na IP adresu iz te odrednice tabele prstiju, s molbom da odgovarajući čvor nastavi traženje. Pošto je on bliži *ključu*, ali je još uvek daleko od njega, postoje dobre šanse da će on odgovoriti na upit posle malog broja naknadnih pretraga. U stvari, pošto svaka pretraga polovi preostalo rastojanje do cilja, može se pokazati daje prosečan broj potrebnih pretraga jednak $\log_2 n$.

Kao prvi primer, razmotrite pretragu kada je *ključ* = 3, a započinje je čvor 1. Pošto čvor 1 zna da *ključ* leži između njega i njegovog sledbenika, čvora 4, željeni čvor je 4 i pretraživanje se završava vraćanjem IP adrese čvora 4.

U dragom primera razmotrite pretragu iniciranu čvorom 1, kada je *ključ* = 14. Pošto 14 ne leži između 1 i 4, gleda se u tabelu prstiju. Najbliži prethodnik čvoru 14 je čvor 9, tako da se zahtev prosleđuje na IP adresu odrednice čvora 9, tj. na adresu čvora 12. Čvor 12 vidi da 14 pada između njega i njegovog sledbenika (15), pa vraća IP adresu čvora 15.

I u trećem primera čvor 1 inicira pretragu, a *ključ* = 16. Upit se ponovo šalje čvoru 12, ali ovog puta čvor 12 ne može sam da nađe pravi odgovor. On traži najbliži čvor koji prethodi čvora 16 i nalazi čvor 14, a taj čvor konačno saopštava IP adresu čvora 15. Upit se zatim šalje tamo. Čvor 15 primećuje da čvor 16 leži između njega i njegovog sledbenika (20), pa pozivaocu vraća IP adresu čvora 20, koja putuje natrag do čvora 1.

Pošto se čvorovi sve vreme priključuju i isključuju, Chord mora imati načina da takve događaje obradi. Pretpostavićemo da je na početku rada sistem dovoljno mali, tako da čvorovi mogu direktno da razmene informacije i izgrade prvi krug i tabele prstiju. Posle toga je potrebna neka automatska procedura, kao što sledi u nastavku. Kada nov čvor r poželi da se pridruži, on mora da stupi u vezu s nekim postojećim čvorom i da ga zamoli da u njegovo ime pronađe IP adresu čvora *sledbenik(r).* Novi čvor tada pita *sledbenika(r)* za IP adresu njegovog prethodnika. Posle toga novi čvor moli oba postojeća čvora da u krug između sebe umetnu čvor r. Na primer, ako bi čvor 24 na slici 5-24 želeo da se pridruži, on će pitati bilo koji postojeći čvor za čvor *sled-benik(*24), a to je čvor 27. Zatim će pitati čvor 27 za njegovog prethodnika, a to je čvor 20. Pošto su oba čvora saznala da on postoji, čvor 20 će kao sledbenika koristiti čvor 24, a čvor 27 će ga koristi kao prethodnika. Osim toga, čvor 27 predaje ključeve za opseg 21-24 čvora 24 jer taj opseg sada pripada njemu. U tom trenutku, čvor 24 je potpuno prihvaćen u društvo.

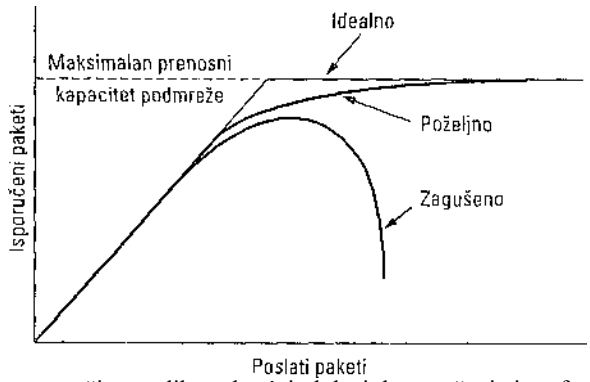
Međutim, sada su mnoge tabele prstiju pogrešne. Da bi se ispravile, svaki Čvor u pozadini periodično proverava svaki prst pozivajući uzastopno funkciju *sledbenik.* Kada jedan od ovih upita pogodi nov čvor, ažurira se odrednica za odgovarajući prst.

Kada se čvor iz mreže udalji na pristojan način, on predaje ključeve svom sledbe- niku i informiše prethodnika o svom odlasku, tako da prethodnik može da se poveže s njegovim bivšim sledbenikom. Međutim, kada čvor naglo otkaže, njegov prethodnik gubi važećeg sleđbenika. Da bi se izbegao problem ovakve vrste, svaki čvor vodi evidenciju ne samo o direktnom sledbeniku, već i o *s* direktnih sleđbenika, kako bi u slučaju otkaza mogao preskočiti do *s* - 1 uzastopnih čvorova i ponovo povezati krug.

Algoritam Chord je korišcen za konstruisanje distribuiranog sistema datoteka (Da- bek i sar., 2001b) i za druge primene, i njegovo istraživanje se nastavlja. Drugačiji sistem ravnopravnih računara, sistem Pastry i njegove primene, opisali su Rowstron i Druschel (2001a, 2001b). Treći takav sistem, Freenet, razmatrali su Clarke i saradnici
(2002) . Četvrti sistem ovoga tipa opisali su Ratnasamy i saradnici (2001).

53 ALGORITMI ZA UPRAVLJANJE ZAGUŠENJEM

Kada se u podmreži ili njenom delu istovremeno nađe previše paketa, performanse opadaju. Ta situacija se naziva zagušenje (engl. *congestion).* Slika 5-25 prikazuje ovaj simptom. Kada broj paketa koje računari „pumpaju“ u podmrežu ne prelazi njen prenosni kapacitet, oni se svi isporučuju (osim onih koji se gube zbog grešaka u pre- nosu) i broj isporučenih paketa proporcionalan je broju poslatih paketa. Međutim, kad saobraćaj počne previše da raste, usmerivači ne mogu da održe korak i počinju da gube pakete. S porastom saobraćaja stanje se pogoršava i - pri nekom veoma velikom saobraćaju ~ performanse potpuno degradiraju, a paketi se skoro i ne isporučuju.



Slika 5-25. Kada je na mreži prevelik saobraćaj, dolazi do zagušenja i performanse naglo slabe.

Do zagušenja može doci iz više razloga. Ako se istovremeno pojave tokovi paketa s tri ili četiri ulazne linije i svi zahtevaju istu izlaznu liniju, red čekanja će trenutno narasti. Ako nema dovoljno memorije za sve, paketi će početi da se gube. Povećanje memorije može da pomogne u izvesnoj meti, ali je Nagle (1987) utvrdio da usmeri- vači s beskonačno mnogo memorije ne razrešavaju zagušenja, već ih samo pogoršavaju, jer paketu koji predugo stoji u redu čekanja ističe tajmer, pa se, umesto da stigne original, stalno ponovo šalju duplikati. Duplikati se poslušno šalju sledećem usmeri- vaču, što samo povećava gužvu ka odredištu.

Zagušenje mogu da izazovu i spori mikroprocesori. Ako procesor usmerivača sporo obavlja

svoje administrativne poslove (svrstavanje u red čekanja, ažuriranje tabela itd.), red čekanja će rasti iako sama linija može da prenese tu količinu paketa. Slično tome, i linije malog propusnog opsega mogu da dovedu do zagušenja. Pojačavanje linija bez pojačavanja procesora i obrnuto pomaže u izvesnoj meri, ali često i samo pre- rnešta usko grlo ka odredištu. Stvarni problem nastaje zbog neusuglašenosti svih delova sistema i on se ne može resiti dok se svi elementi mreže ne uravnoteže.

Treba jasno povući razliku između kontrole toka (engl.*flow control)* i kontrole zagušenja (engl. *congestion control)* jer je u pitanju finesa. Kontrola zagušenja treba da obezbedi da podmreža prenese sve ponuđene pakete. Ona je globalna, jer obuhvata ponašanje svih računara, usmerivača, obrade „čuvaj i prosledi“ u svakom usmerivaču i sve druge činioce koji utiču na smanjenje prenosnog kapaciteta podmreže.

Za razliku od toga, kontrola toka se odnosi na saobraćaj od tačke do tačke između određenog pošiljaoca i određenog primaoca. Njen zadatak je da spreči brzog pošiljaoca da podacima zatrpa sporijeg primaoca. Kontrola toka obično podrazumeva neke povratne informacije na osnovu kojih pošiljalac saznaje kako primalac izlazi na kraj s paketima.

Da biste jasnije uočili razliku između ova dva mehanizma, razmotrimo mrežu izvedenu optičkim kablom, kapaciteta 1000 Gb/s, na kojoj superračunar pokušava da pošalje datoteku PC računam brzinom 1 Gb/s. Iako ovde neće doći do zagušenja (mreža nema problema s prenosom), neophodno je naterati superračunar da pravi pauze kako bi PC računam omogućio da predahne.

Kao drugu krajnost, razmotrimo mrežu koja radi na principu „čuvaj i prosledi“, ima linije brzine IMb/s i 1000 velikih računara, od kojih jedna polovina pokušava da drugoj polovini šalje datoteke brzinom 100 kb/s. Ovde neće brzi pošiljaoci preplaviti sporije primaoce paketima, već je problem u tome što se nudi veći saobraćaj nego što mreža može da obradi.

Kontrola toka i kontrola zagušenja često se mešaju zato što neki algoritmi za kontrolu zagušenja - kad mreža dospe u kritičnu situaciju - šalju zahteve različitim pošiljaocima da uspore svoje emitovanje. Na taj način, pošiljalac može da primi poruku „uspori“ iz dva razloga: kada primalac ne može da dovoljno brzo obradi pakete ili kada sama mreža ne može da ih prenese. Na ovo ćemo se vratiti kasnije.

Proučavanje kontrole zagušenja počećemo razmatranjem opšteg modela. Zatim ćemo opisati načelne pristupe za sprečavanje zagušenja, a potom se pozabaviti dinamičkim algoritmima za razrešavanje već nastalog zagušenja.

5.3.1 Opšti principi kontrole zagušenja

Mnogi problemi u složenim sistemima, kao što su računarske mreže, mogu se po- smatrati s gledišta teorije upravljanja. Takav pristup omogućava da se rešenja problema traže pre nego što dođe do zagušenja ili tek onda kad zagušenje nastane. Prva grupa rešenja usmerava se na dobro projektovanje mreže kako do zagušenja ne bi ni došlo. Kada se sistem pusti u rad, na njemu se više ne interveniše.

Sprečavanje zagušenja na prvi način obuhvata donošenje različitih odluka: kada da se prihvati nov saobraćaj, kada i koji paketi da se izbace, kako da se na različitim mestima u mreži rasporede paketi za slanje i slično. Zajedničko im je to što se odluke donose bez obzira na trenutno stanje saobraćaja u mreži.

Nasuprot tome, druga grupa rešenja zasniva se na korišćenju povratnih informacija. Uz taj pristup, zagušenje se kontroliše u tri glavna koraka:

1. Stalno nadziranje sistema u cilju otkrivanja vremena nastanka i mesta zagušenja.
2. Prosleđivanje informacija do mesta na kojima se može preduzeti akcija.
3. Podešavanje rada sistema da se problem otkloni.

Zagušenje podmreže može se meriti na razne načine, a najvažniji su: na osnovu procenta svih odbačenih paketa zbog nedostatka prostora u baferu, na osnovu proseč- ne dužine redova čekanja, na osnovu broja paketa koji se moraju ponovo slati jer im se tajmer isključio, na osnovu prosečnog kašnjenja paketa i na osnovu standardnog odstupanja kašnjenja paketa. U svim slučajevima, povećanje brojčane vrednosti parametra nagoveštava zagušenje.

Drugi korak u rešenjima zasnovanim na korišćenju povratnih informacija jeste prenošenje informacija o zagušenju sa mesta gde je zagušenje otkriveno na mesto gde se nešto u pogledu toga može preduzeti. Odmah se dosećamo da usmerivač koji je otkrio zagušenje treba da pošalje paket pošiljaocu ili pošiljaocima da bi im najavio problem. Naravno, on to treba da uradi u najnepoželjnijem trenutku - kada je mreža već zagušena.

Međutim, postoje i drage mogućnosti. Na primer, u svakom paketu se može rezer- visati jedan bit ili polje koje treba da popuni usmerivač čim gustina saobraćaja kod njega pređe određenu granicu. Kada usmerivač tako utvrdi da saobraćaj teži da se zaguši, on popunjava to polje u svim paketima koje prosleđuje kako bi upozorio susede na novonastalu situaciju.

Drugi načinje da svi računari ili usmerivači povremeno šalju probne pakete ispitujući moguće zagušenje. Te informacije se zatim mogu iskoristiti da se saobraćaj usmeri oko problematičnih područja. Neke radio-stanice imaju helikoptere koji kruže nad gradskim područjem i izveštavaju o zagušenjima na putevima, tako da njihovi slušaoci (u automobilima) na vreme mogu da ih izbegnu.

U svim sistemima zasnovanim na korišćenju povratnih informacija pretpostavlja se da će računari na osnovu njih preduzeti mere za smanjenje zagušenja. Da bi sistem radio ispravno, mora se vremenski pažljivo uskladiti. Ako svaki put kada mu uzastopno stignu dva paketa usmerivač drekne: STOJ, a svaki put kada tokom 20 mikrosekundi ništa ne radi, vikne: NAPRED, sistem će početi da divlje osciluje i nikada se neće uravnotežiti. S druge strane, ako usmerivač premišlja 30 minuta pre nego što išta kaže, mehanizam kontrole zagušenja rađice tako sporo da od njega nece biti nikakve praktične koristi. Potrebno je, dakle, naći pravu meru, ali to nije baš jednostavno.

Postoji mnogo algoritama za kontrolu zagušenja. Da bi ih sistematizovali na shvatljiv način, Yang i Reddy (1995), napravili su njihovu taksonomiju. Prvo su sve algoritme podelili u dve, već opisane grupe: one koji sprečavaju zagušenje („preventivne“) i one koji razrešavaju nastalo zagušenje („kurativne“). Preventivne algoritme dalje su podelili na one koji se izvršavaju na izvorištu i one koji se izvršavaju na odredištu. Kurativne algoritme su takođe podelili na dve potkategorije: na algoritme sa eksplicitnim i implicitnim upozoravanjem. Kod eksplicitnih, paket se s mesta zagušenja direktno šalje izvorištu. Kod implicitnih, sam izvor donosi zaključale o zagušenju na osnovu lokalnih posmatranja, npr. vremena potrebnog za stizanje potvrde o prijemu paketa.

Zagušenje znači daje (trenutno) opterećenje veće od onoga što resursi (u delu sistema) mogu da obrade. Odmah nam se nameću dva rešenja: proširiti resurse ili smanjiti opterećenje. Na primer, u podmrežu se mogu uključiti modemske telefonske linije da bi se privremeno povećala propusna moć između određenih tačaka. U satelitskim sistemima, povećanje snage predajnika često povećava i propusni opseg. Raspo- deljivanje saobraćaja duž većeg broja manje optimalnih putanja, umesto da se slepo držimo samo optimalne, takođe može da poveća propusni opseg. Najzad, kada dođe do ozbiljnog zagušenja, mogu se uključiti i usmerivači koji inače služe samo za čuvanje rezervnih kopija (da bi sistem bio otporan na otkaze).

Međutim, kapacitet se ne može uvek povećati ili sistem već radi blizu maksimalnog opterećenja. Tada je jedino rešenje da se smanji saobraćaj. Za to postoji više načina: odbijanje usluga određenim korisnicima, sniženje kvaliteta usluga nekim ili svim korisnicima i prisiljavanje korisnika da svoje zahteve postavljaju na predvidljiviji način.

Neke od ovih metoda, koje ćemo ukratko opisati, mogu se najbolje primeniti na virtuelna kola. Podmreže koje interno koriste virtuelna kola ugrađuju takve algoritme u svoj mrežni sloj. Bez obzira na to, oni se u datagramskim mrežama mogu ponekada koristiti u vezama transportnog sloja. U ovom poglavlju opisaćemo njihovu upotrebu u mrežnom sloju, a kontrolu zagušenja pomoću protokola transportnog sloja ostavi- ćemo za sledeće poglavlje.

5.3.2 Pravila sprečavanja zagušenja

Počnimo naše proučavanje metoda kontrole zagušenja razmatranjem sistema za njegovo sprečavanje. Ti sistemi su projektovani da unapred smanje mogućnost zagušenja, a ne da reaguju tek kada do njega dođe. Cilj se postiže primenom odgovarajućih pravila ponašanja na različitim nivoima. Na slici 5-26 vidimo više takvih pravila koja se primenjuju u sloju veze podataka, mrežnom i transportnom sloju (Jain, 1990).

Počnimo razmatranje od sloja veze podataka naviše. Pravila o ponovnom slanju paketa tiču se roka tajmera kod pošiljaoca i onoga šta on šalje kada se tajmer isključi.

Sloj	Pravila o
Transportni	• Ponovnom slanju paketa • Čuvanju prekorednih paketa • Potvrđivanju paketa • Kontroli toka • Isticanju roka tajmera
Mrežni	■ Virtuelnim kolima/datagramima unutar podmreže • Svrstavanju paketa u redove čekanja i njihovoj obradi • Odbacivanju paketa • Algoritmu za usmeravanje • Načinu korišćenja „životnog veka" paketa
Veze podataka	• Ponovnom slanju paketa • Čuvanju prekorednih paketa • Potvrđivanju paketa • Kontroli toka

Slika 5-26. Pravila koja utiču na zagušenje.

Nervozan pošiljalac čiji se tajmer brzo isključuje i koji ponovo šalje već poslate, a još neprimljene pakete koristeći algoritam „vrati se n“, više opterećuje sistem nego smireniji pošiljalac koji koristi selektivno ponavljanje. Slično razmišljanje može se pri- meniti i na čuvanje prekorednih paketa (engl. *out-of-order packets)*. Ako primalac sistematski odbacuje pakete koji stižu preko reda, oni će biti ponovo poslati, opterećujući dodatno mrežu. U pogledu kontrole zagušenja, selektivno ponavljanje ima jasnu prednost nad algoritmom „vrati se n“.

Način na koji se potvrđuju paketi takođe utiče na zagušenje. Ako se za svaki paket odmah šalje potvrda, paketi s potvrdama dodatno opterećuju mrežu. Međutim, ukoliko se potvrde šlepuju uz povratne pakete, tajmeri će se učestalije isključivati i biće više ponovo poslatih paketa. Sema kojom se tok podataka strogo kontroliše (npr. mali prozor) smanjuje brzinu slanja podataka i na taj način odlaže zagušenje.

Izbor između virtuelnih lcola i datagrama u mrežnom sloju utiče na zagušenje jer mnogi algoritmi za kontrolu zagušenja rade samo s podmrežama zasnovanim na vir- tuelnim kolima. Redovi čekanja utiču na zagušenje na različit način ako usmerivači imaju po jedan red čekanja na svakoj ulaznoj liniji, na svakoj izlaznoj liniji ili na obe linije. Važan je i redosled kojim se

paketi obrađuju (npr. ciklično ili na osnovu prioriteta). Pravilima odbacivanja paketa predviđa se koji će se paket odbaciti kad ponestane prostora u baferu. Mudro smišljena pravila otkloniće zagušenje, a ona olako primenjena samo će ga pogoršati.

Dobar algoritam za usmeravanje može da spreči zagušenje tako što će saobraćaj uputiti duž svih linija, dok loš algoritam upućuje suviše paketa duž linija koje su već zagušene. Najzad, životni vek paketa predstavlja vreme posle koga se paket odbacuje. Ako je on prevelik, izgubljeni paketi mogu dugo da zagušuju mrežu, ali ukoliko je premali, tajmeri se mogu automatski isključivati pre nego što paketi stignu do odredišta, što povlači njihovo ponovno slanje.

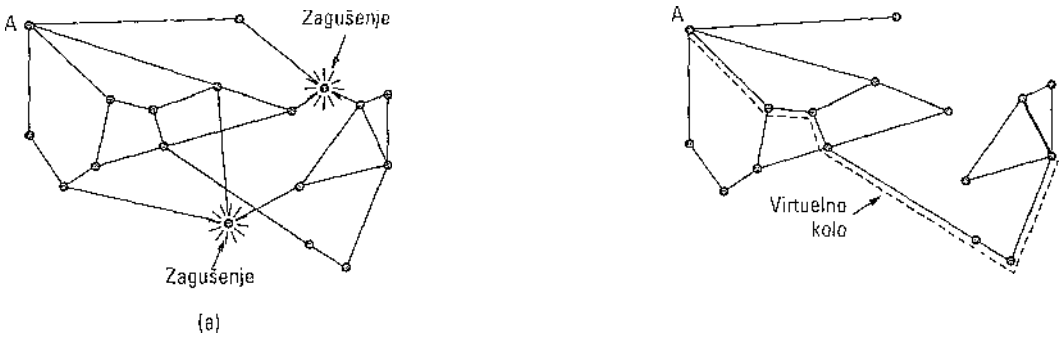
U transportnom sloju odvijaju se slične stvari kao u sloju veze podataka, a dodatna komplikacija je to što se rok isključivanja tajmera teže određuje jer je vreme prenosa paketa preko čitave mreže manje predvidljivo od vremena prenosa između susednih usmerivača. Ukoliko se tajmer prerano isključuje, nepotrebno će se ponovo slati paketi. Ako se, pak, isključuje prekasno, zagušenje će se doduše smanjiti, ali će se usporiti komunikacija kad god se neki paket izgubi.

5.3.3 Kontrola zagušenja u podmrežama s virtuelnim kolima

Metode kontrole zagušenja koje smo dosad opisali u osnovi služe da na prvom mestu spreče nastanak zagušenja. U ovom odeljku ćemo opisati neke pristupe za dinamičku kontrolu zagušenja u podmrežama s virtuelnim kolima. U naredna dva odeljka upoznaćemo se i s metodama koje se mogu primeniti na bilo kakvu podmrežu.

Tehnika koja se široko koristi za obuzdavanje već nastalog zagušenja jeste kontrola pristupa (engl. *admission control).* Zamisao je u osnovi jednostavna: kada se najavi zagušenje, ne uspostavljaju se nova virtuelna kola dok se situacija ne smiri. Tako propadaju svi pokušaji da se uspostave nove veze u transportnom sloju. Ako pustite još korisnika u mrežu, stvar može samo da se pogorša. Premda je ovaj pristup grab, jednostavan je i lak za primenu. U telefonskom sistemu, kada je centrala preopterećena, ona takođe kontroliše pristup i ne daje signal (tj. onemogućava biranje broja).

Alternativno se može dozvoliti uspostavljanje novih virtuelnih kola, samo se ona moraju pažljivo voditi oko problematičnog područja. Razmotrite, na primer, pod- mrežu sa slike 5-27(a) u kojoj su dva usmerivača zagušena.



Slika 5-27. (a) Zagušena podmreža. (b) Ista podmreža iz koje su uklonjeni zagušeni usmerivači. Prikazano je i virtuelno kolo od A do *B*

Pretpostavimo da računar povezan sa usmerivačem *A* želi da uspostavi vezu s računarom povezanim sa usmerivačem *B.* U normalnoj situaciji, ta veza bi prolazila kroz jedan od zagušenih usmerivača. Zbog toga ćemo iz mreže ukloniti zagušene delove, kao na slici 5-27(b). Isprekidana linija prikazuje moguću trasu virtuelnog kola kojom se zaobilazi zagušenje.

U podmreži s virtuelnim kolima može se primeniti i strategija dogovaranja između računara i podmreže pri uspostavljanju virtuelnog kola. Dogovor obično obuhvata obim i oblik saobraćaja, kvalitet usluge i druge parametre. Da bi ispunila dogovoreno, podmreža će duž uspostavljenog virtuelnog kola obično rezervisati odgovarajuće resurse koji se odnose na prostor tabela i bafera u usmerivačima i na propusni opseg linija. Na taj način, malo je verovatno da dođe do zagušenja pri uspostavljanju novog virtuelnog kola jer su svi neophodni resursi garantovani.

Opisano rezervisanje resursa može se sprovoditi rutinski ili samo onda kada je podmreža zagušena. Kada se primenjuje u normalnom radu, nezgodno je to što se resursi više troše. Ako šest virtuelnih kola koja rezervišu propusni opseg od po 1 Mb/s prolaze kroz istu fizičku liniju propusne moći 6 Mb/s, ta linija se mora označiti kao popunjena, iako će se retko desiti da svih šest kola istovremeno prenose podatke najvećom brzinom. Prema tome, sprečavanje zagušenja u normalnom radu plaća se manjim iskorišćenjem propusnog opsega.

5.3.4 Kontrola zagušenja u datagramskim podmrežama

Pogledajmo šta se u pogledu kontrole zagušenja može uraditi u datagramskim pod- mrežama (što je primenljivo i na podmreže s virtuelnim kolima). Svaki usmerivač lako može da prati opterećenost svojih izlaznih linija i drugih resursa. Na primer, on svakoj liniji može da pridruži realnu promenljivu *u,* čija vrednost (između 0,0 i 1,0) odražava njeno skorašnje opterećenje. Da bi promenljiva *u* što približnije odražavala stvarno stanje, periodično se snima trenutno iskorišćenje linije/(0,0 ili 1,0) i *u* ažurira prema sledećem:

$M_{novo} = aM_{staro} + (1 - a)f$ gde konstanta *a* određuje brzinu kojom usmerivač zaboravlja prethodno stanje.

Kada vrednost *u* pređe zadati prag, izlazne linije se stavljaju u stanje pripravnosti. Tada se za svaki pristigli paket proverava da li je namenjen izlaznoj liniji koja je u stanju pripravnosti. Ako jeste, preduzima se jedna od mogućih akcija, o kojima ćemo govoriti u nastavku.

Bit upozorenja

U staroj arhitekturi DECNET mreža, upozoravanje je ostvarivano postavljanjem specijalnog bita u zaglavlje paketa. Tako radi i štafetni prenos okvira (engl. *frame relay).* Kada paket stigne na odredište, transportni proces kopira bit u sledeću potvrdu koju šalje izvorištu. Izvorišni računar tada usporava emitovanje.

Dok god je u stanju pripravnosti, usmerivač stalno postavlja bit upozorenja, tako da izvorišni računar neprestano dobija upozoravajuće potvrde. Izvorišni računar prati broj upozoravajućih potvrda koje pristižu i prema njima usklađuje emitovanje. Izvorište usporava emitovanje sve dok mu stižu upozoravajuće potvrde. Kada se one sasvim prorede, izvorišni računar poveća brzinu emitovanja. Obratite pažnju na to da svaki usmerivač na putanji može da postavi bit upozorenja, tako da se saobraćaj stvarno ubrzava teka kada nijedan usmerivač nije u škripcu.

Prigušni paketi

Opisani algoritam je možda previše pristojan - on izokola saopštava izvorištu da uspori emitovanje. Zašto mu to ne bi saopštio direktno? U pristupu o kome sada govorimo, usmerivač šalje izvorištu tzv. **prigušni paket** (engl. *chokepacket*) sa adresom odredišta koju je našao u pristiglom paketu. Originalni paket se posebno označava (bit u zaglavlju postavlja mu se na jedan) da na svom putu do odredišta ne bi ponovo izazvao slanje prigušnih paketa, i šalje se na uobičajen način.

Kada dobije prigušni paket, od izvorišnog računara se zahteva da emitovanje ka određenom odredištu smanji za *X* procenata. Pošto je ka tom odredištu već poslao još paketa koji će ponovo izazvati povratno slanje prigušnih paketa, računar treba da zanemari prigušne pakete koji se odnose na to odredište tokom određenog vremenskog intervala. Kada taj interval istekne, računar tokom sledećeg vremenskog intervala čeka da li će stići još prigušnih paketa. Ako takav paket stigne, znači daje linija još zagušena, pa računar dodatno usporava emitovanje i ponovo počinje da zanemaruje prigušne pakete. Ukoliko tokom perioda čekanja ne stigne prigušni paket, računar može ponovo da ubrza emitovanje. Povratne informacije koje se ovim protokolom implicitno šalju, mogu da pomognu u sprečavanju zagušenja, pa ipak ne ograničavaju protok podataka sve dok nešto zaista ne otkaže.

Računari mogu da ograniče svoj saobraćaj menjanjem vrednosti nekih parametara, na prirner, veličine prozora. Prvi prigušni paket najčešće smanjuje brzinu prenosa na 50% prvobitne vrednosti, drugi na 25% itd. Nakon otklonjenog zagušenja, brzina prenosa se povećava u manjim koracima da se podmreža ne bi odmah ponovo zagušila.

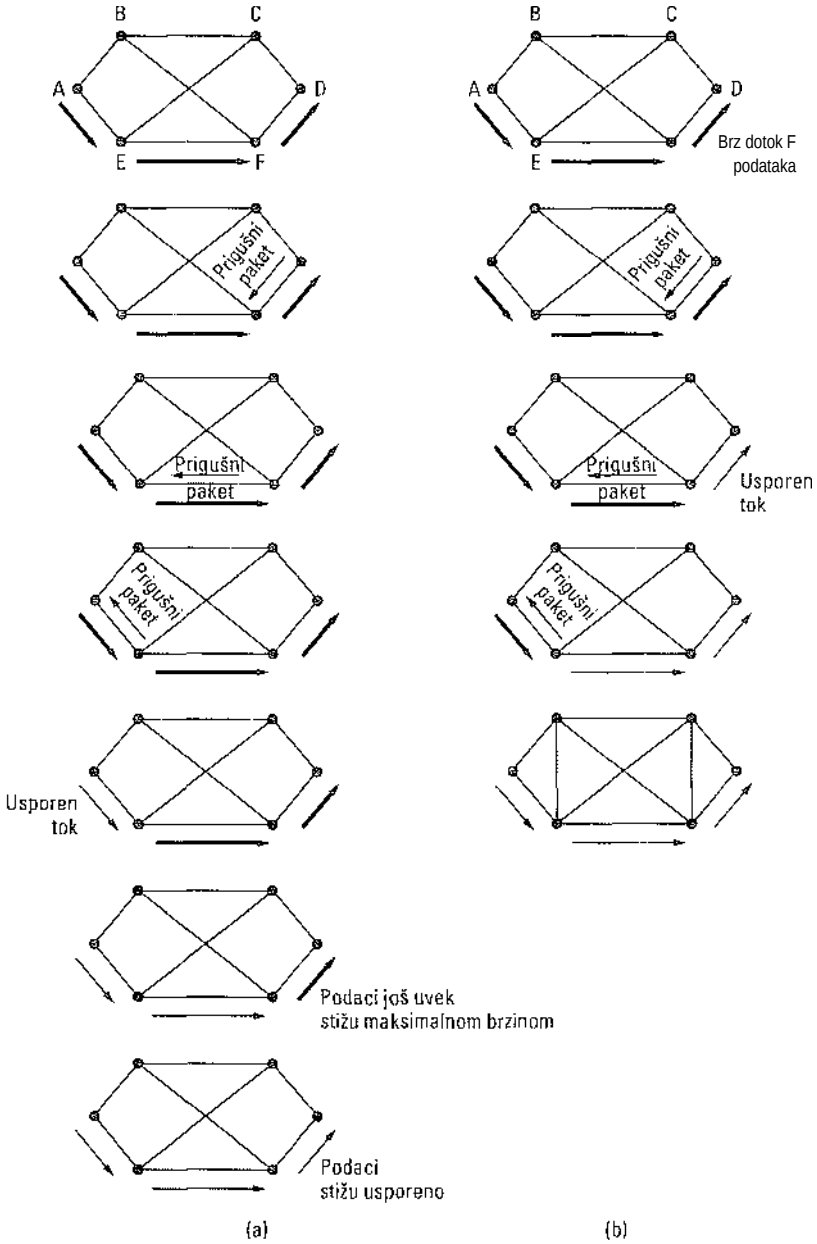
Predloženo je više varijanti opisanog algoritma za kontrolu zagušenja. Po jednoj, usmerivači mogu imati tri stepena pripravnosti, svaki ozbiljniji od prethodnog. U zavisnosti od toga u kome se stanju usmerivač nalazi, paketi mogu sadržati blago upozorenje, strogo upozorenje i ultimativan zahtev.

U drugoj varijanti, kao argument za upozoravanje ne koristi se opterećenost linije, već dužina redova čekanja ili opterećenost bafera. Naravno, i ovde se može upotrebiti eksponencijalno ponderisanje, kao kod promenljive *u*.

Prigušni paketi za svaki skok

Pri velikim brzinama prenosa ili pri prenosu na velika rastojanja, slanje prigušnog paketa izvorištu ne radi dobro jer je reakcija previše spora. Zamislite, na primer, da računar u San Francisku (usmerivač A na slici 5-28) šalje podatke računara u Njujor- ku (usmerivač *D* na slici 5-28) brzinom 155 Mb/s. Ako njujorškom računara počne da preliva bafer, prigušni paket će tek posle 30 ms uspeti da uspori računar u San Francisku. Putovanje prigušnog paketa prikazano je u dragom, trećem i četvrtom koraku na slici 5-28(a). U tih 30 ms biće poslato novih 4,6 megabita. Čak i ako računar u San Francisku po prijemu prigušnog paketa potpuno obustavi emitovanje, 4,6 megabita podataka nastaviće svoj put i na odredištu stvoriti probleme. Tek će u sedmom dija-gramu na slici 5-28(a) njujorški usmerivač primetiti da je prenos podataka usporen.

Pozitivan efekat slanja prigušnog paketa odmah bi se osetio ako bi prigušni paket na svom putu ka izvorištu „prigušio“ i svaki usmerivač na koji skoči, kao na slici 5-28(b). Ovde, čim paket stigne do *F,* od *F* se traži da smanji tok podataka ka *D.* Da



Slika 5-28. (a) Prigušni paket koji deluje samo na izvorište, (b) Prigušni paket koji deluje na svaki usmerivač na putu od odredišta do izvorišta.

bi ispunio zahtev, *F* će morati da optereti svoje bafere jer paketi od izvorišta i dalje stižu punom brzinom, ali će time omogućiti usmerivaču *D* da za trenutak predahne. U sledećem skoku prigušni paket stiže do *E* i naređuje mu da uspori slanje ka *F.* To još više opterećuje bafere usmerivača *E,* ali omogućuje usmerivaču *F* da predahne. Najzad, prigušni paket stiže do *A* i emitovanje paketa se stvarno usporava.

Ukupan efekat ove šeme u kojoj prigušni paket đeluje na svaki usmerivač duž putanje jeste otklanjanje zagušenja po cenu intenzivnijeg korišćenja bafera uz tok. Na taj način, zagušenje se može suzbiti u samom začetku, a da se pri tome ne izgubi nijedan paket. Algoritam su detaljno razmotrili i njegove rezultate đobijene simulaci- jom prikazali Mishra i Kanakia (1992).

5.3.5 Odbacivanje paketa

Kada nijedna od opisanih metoda ne otkloni zagušenje, usmerivači mogu da pri- mene svoje „oružje „ : odbacivanje paketa. Odbacivanje paketa (engl. *load shed- ding)* predstavlja ono što usmerivači rade kada su zasuti paketima koje ne mogu da obrade. Izraz (bukvalno: odbaciti, otpasti) potiče iz oblasti proizvodnje električne energije, gde se iz mreže u špicu potrošnje isključuju pojedina područja potrošača da ne bi došlo do kolapsa čitavog sistema.

Usmerivač koji je zatrpan paketima može ih odbacivati nasumice, ali se on obično drži izvesnog sistema. Koje će pakete odbaciti može da zavisi od aplikacija koje se izvršavaju. Kada se radi o prenosu datoteka, stari paket je vredniji od novog jer će odbacivanje paketa 6, uz zadržavanje paketa 7, 8, 9 i 10 napraviti prekid kod primaoca i možda izazvati ponovno slanje svih paketa između šestog i desetog (ukoliko primalac rutinski odbacuje prekoredne pakete). U datoteci od 12 paketa, odbacivanje paketa 6 može izazvati ponovno slanje svih paketa između šestog i dvanaestog, dok će odbacivanje paketa 10 izazvati ponovno slanje samo paketa između desetog i dvanaestog. Nasuprot tome, kod multimedije je nov paket važniji od starog. Prethodno pravilo (staro je bolje od novog) često se zove vino, a potonje (novo je bolje od starog) - mleko.

Za iole inteligentnije rešenje bilo bi potrebno da se ostvari nekakva saradnja s pošiljaocima. Nisu svi paketi jednako vređni - u mnogim aplikacijama, neki paketi su važniji od drugih. Na primer, određeni algoritmi za komprimovanje videa periodično šalju čitavu statičnu sliku, a potom sukcesivno samo razlike u odnosu na nju. U ovom slučaju, bolje je odbaciti paket s nekom od razlika, nego paket koji predstavlja deo statične slike. Ili, pale, razmotrimo slanje dokumenta koji se sastoji od običnog teksta i slika. Gubitak reda piksela sa slike mnogo je manje dramatičan od gubitka reda teksta.

Da bi se mogla primeniti pravila inteligentnog odbacivanja paketa, same aplikacije moraju svoje pakete da prema važnosti svrstaju u određene klase prioriteta i tako ih označe. Ako tako urade, usmerivač će u slučaju potrebe najpre odbaciti pakete najniže klase, zatim sledeće više klase itd. Naravno, ako niko ne naredi drugačije, najverovat- nije će tada svi paketi nositi oznaku VRLO VAŽAN PAKET - NE ODBACUJ GA NI POD KOJIM USLOVIMA.

Pravilo prioriteta može zaživeti ako se prioritet na određeni način oporezuje, tj. ako slanje paketa niskog prioriteta bude jeftinije od slanja paketa visokog prioriteta. Alternativno, pošiljaocima se može ponuditi da pakete visokog prioriteta šalju samo kada je saobraćaj na mreži redak; s povećanjem gustine saobraćaja i ovi paketi počinju da se odbacuju, pa se pošiljalac prinuđuje da prestane da ih šalje.

Još jedna mogućnost je da se računalima dozvoli da prekorače granicu dogovorenu pri uspostavljanju virtuelnog kola (npr. korišcenje šireg propusnog opsega), ali uz uslov da se dodatni saobraćaj tretira kao saobraćaj niskog prioriteta. Takva strategija u stvari i nije loša jer se uz nju aktiviraju neiskorišćeni resursi i računarima se omogućuje da ih koriste sve dok se neko drugi za njih ne zainteresuje, ali bez ikakvog polaganja prava na njih kada dođe do gužve.

Rano otkrivanje zagušenja

Dobro je poznato da je mnogo bolje sprečiti zagušenje u začetku, nego raščišćavati gužvu kad se sve već zaglavi. Ovo zapažanje dovelo je do ideje da paketi počnu da se odbacuju i pre nego što se popune svi baferi. To je zadatak popularnog algoritma za **rano otkrivanje zagušenja** (engl. *Random Early Detection, RED*) (Floyd i Jackob- son, 1993). U nekim transportnim protokolima (uključujući i protokol TCP), gubljenje paketa znači da izvorišni računar treba da uspori. Pošto je TCP projektovan za kablovske mreže, a one su vrlo pouzdane, paketi se uglavnom ne gube zbog grešaka u prenosu, već zbog preopterećenja bafera. Ta činjenica se može iskoristiti za smanjenje zagušenja.

Kada su usmerivači podešeni da odbacuju pakete pre nego što situacija postane beznadežna (zbog toga je ono „rano“ u imenu algoritma), podrazumeva se da postoji vreme za reagovanje pre nego što bude prekasno. Da bi utvrdili kada treba da počnu da odbacuju pakete, usmerivači stalno prate prosečnu dužinu svojih redova čekanja. Kada srednja dužina reda čekanja na nekoj liniji pređe kritičnu granicu, ta linija se smatra zagušenom i preduzima se odgovarajuća akcija.

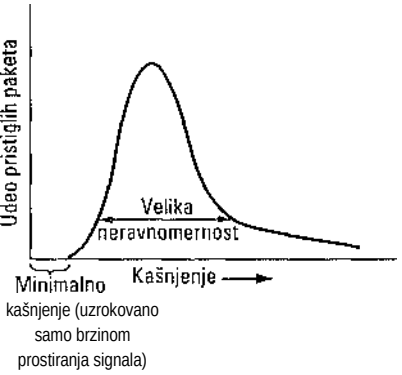
Pošto usmerivač verovatno ne može da odredi koje je izvorište glavni krivac za zagušenje, nasumično odbacivanje paketa iz zagušenog reda čekanja isto je tako dobro rešenje kao i svako drugo.

Kako da usmerivač obavesti izvorište o problemu? Jedan, već opisani način jeste da mu pošalje prigušni paket. Ne valja to što treba slati paket kroz mrežu koja je već zagušena. Drugi način je da samo odbaci paket i da nikoga o tome ne obaveštava; izvorište će već zapaziti da mu potvrda nije stigla, pa će preduzeti neke mere. Pošto zna da paketi u načelu nestaju zbog zagušenja i zato što se odbacuju, usporiće emito- vanje. Takva podrazumevana povratna sprega radi samo ako izvorište, kada primeti da se paketi gube, uspori slanje. U bežičnim mrežama, gde se većina paketa gubi zbog nepouzdane veze, opisani pristup se ne može primeniti.

5.3.6 Kontrola neravnomernosti pristizanja paketa

Za oblasti primene kao što su audio i video prenos u realnom vremenu, nije previše važno da li paketi pri isporuci kasne 20 ms ili 30 ms, sve dok je to kašnjenje konstantno. Varijacija (tj. standardna devijacija) vremena stizanja paketa naziva se **neravno- mernost** (engl *.jitter).* Velika neravnomernost pri stizanju paketa, npr. ako neki paketi stižu posle 20 ms, a drugi posle 30 ms, narušava kvalitet zvuka ili slike. Neravnomernost je prikazana na slici 5-29. Nasuprot tome, dogovor da će 99% paketa biti isporučeno s kašnjenjem između 24,5 ms i 25,5 ms mogao bi se prihvatiti.

Dogovoreni interval kašnjenja, naravno, mora da bude realno izvodljiv. Moraju se uzeti u obzir brzina prostiranja signala, minimalno zadržavanje u usmerivačima i možda ostaviti mala rezerva za nepredviđene okolnosti.



Slika 5-29. (a) Visok stepen neravnomemosti stizanja paketa, (b) Nizak stepen neravnomemosti.

Neravnomernost pristizanja paketa može se ograničiti izračunavanjem očekivanog vremena prolaska pri svakom skoku na putanji. Kada paket stigne u usmerivač, ovaj upoređuje njegovo stvarno vreme sa očekivanim. Ta informacija se smešta u paket i ažurira pri svakom skoku. Ako paket stigne pre očekivanog vremena, on se zadržava onoliko koliko je potrebno da se vrati u svoj vremenski raspored. Ukoliko paket stigne sa zakašnjenjem, usmerivač se trudi da ga što pre ekspeduje.

U stvari, pri određivanju paketa koji će sledeći biti poslat na određenu izlaznu liniju, algoritam uvek može da izabere paket koji najviše kasni u odnosu na svoj raspored stizanja. Na taj način, brži paketi se usporavaju, a sporiji ubrzavaju, što smanjuje neravnomernost njihovog stizanja na odredište.

U nekim oblastima primene, kao što je video na zahtev, neravnomernost stizanja paketa može se eliminisati ako se paketi bafemju kod primaoca i za prikazivanje isporučuju iz bafera, a ne s mreže u realnom vremenu. Međutim, u drugim oblastima, naročito tamo gde je

neophodna interakcija između korisnika u realnom vremenu, kao što su Internet telefonija i video konferencije, kašnjenje prouzrokovano baferovanjem nije prihvatljivo.

Kontrola zagušenja je područje koje se aktivno istražuje. Dokle se na tom polju stiglo, saznajte od Gevrosa i saradnika (2001).

5.4 KVALITET USLUGA

Tehnike koje smo razmatrali u prethodnim odeljcima namenjene su smanjenju zagušenja na mreži i poboljšanju njenih performansi. Međutim, s porastom multimedijskog saobraćaja u mrežama, te ad hoc mere često nisu dovoljne. Za garantovanje

kvaliteta usluge potrebni su ozbiljni napori na poboljšanju postojećih mreža i protokola. U narednim odeljcima nastavicemo da se bavimo performansama mreže, ali uglavnom sa aspekta obezbeđivanja kvaliteta usluga koji odgovara potrebama različitih oblasti primene. Treba ipak na samom početku naglasiti da su mnoge ideje o kojima ćemo govoriti još uvek nedovoljno definisane i da se mogu izmeniti.

5.4.1 Zahtevi

Niz paketa koji od izvorišta putuje ka odredištu naziva se tok (engl. *flow)*. U mrežama u kojima se veza direktno uspostavlja, svi paketi koji pripadaju toku slede istu putanju; u mrežama koje rade bez uspostavljanja direktne veze oni mogu slediti različite putanje. Potrebe svakog toka mogu se opisati pomoću četiri osnovna parametra, a to su: pouzdanost, kašnjenje, neravnomernost stizanja paketa i propusni opseg. Oni zajedno određuju kvalitet usluge (engl. *Quality of Service, QoS)* koju tok zahteva. Na slici 5-30 navedene su uobičajene oblasti primene, zajedno sa ozbiljnošću njihovih zahteva.

Oblast primene	Pouzdanost	Kašnjenje	Neravnomernost	Propusni opseg
E-pošta	Velika	Malo	Mala	Mali
Prenos datoteka	Velika	Malo	Mala	Srednji
Pristup Webu	Velika	Srednje	Mala	Srednji
Daljinsko prijavljivanje	Velika	Srednje	Srednja	Mali
Audio na zahtev	Mala	Malo	Velika	Srednji
Video na zahtev	Mala	Malo	Velika	Veliki
Telefonija	Mala	Veliko	Velika	Mali
Video konferencije	Mala	Veliko	Velika	Veliki

Slika 5-30. Ozbiljnost zahteva za kvalitetom usluge.

Prve četiri oblasti primene zahtevaju veliku pouzdanost - ne sme se pogrešno isporučiti nijedan bit. Cilj se obično postiže tako što svaki paket nosi kontrolni zbir koji se proverava na odredištu. Ako se paket ošteti u transportu, za njega se ne šalje potvrda i očekuje se da on odmah bude ponovo poslat. Takva strategija obezbeđuje veliku pouzdanost prenosa. Poslednje četiri oblasti primene (audio/video) mogu da tolerišu greške, pa se za njih kontrolni zbir niti izračunava, niti proverava.

Za prenos datoteka, uključujući e-poštu i video, kašnjenje nije važno. Ako svi paketi ravnomerno kasne nekoliko sekundi, nije strašno. Na kašnjenje su osetljivije interaktivne radnje, kao što su pretraživanje Weba i daljinsko prijavljivanje. Aktivnosti koje se obavljaju u realnom vremenu: telefonija i video konferencije, imaju stroge za- hteve u pogledu kašnjenja. Ako sve reči izgovorene u telefonskom razgovoru kasne tačno 2 sekunde, takva veza će korisnicima biti neprihvatljiva. S druge strane, kada se muzičke numere i video sekvence reprodukuju dok se preuzimaju sa servera, ne postavljaju se visoki zahtevi u pogledu kašnjenja.

Za prve tri oblasti primene nije važno ako paketi stižu u neredovnim vremenskim intervalima. Daljinsko prijavljivanje je na to osetljivije, pošto će se slova na ekranu po- javljivati rafalno ako paketi stižu veoma neravnomerno. Video, i naročito audio sadržaji, izuzetno su osetljivi 11a neravnomemost. Ukoliko korisnik gleda film preko mreže i sve slike kasne po 2 sekunde, nema štete. Ali ako slike nasumično kasne između 1 i 2 sekunde,

rezultat će biti katastrofalan. Pri prenosu zvuka, jasno se zapaža i neravno- mernost od nekoliko milisekundi.

Najzad, oblasti primene se razlikuju i po svojim potrebama za propusnim opsegom, pri čemu je za e-poštu i daljinsko prijavljivanje potreban mali opseg, dok svi oblici videa zahtevaju veliki propusni opseg.

U ATM mrežama, tokovi se svrstavaju u četiri široke kategorije, u zavisnosti od kvaliteta usluga koji zahtevaju:

1. Konstantna brzina prenosa (npr. telefonija).
2. Promenljiva brzina prenosa u realnom vremenu (npr. komprimovane video konferencije).
3. Promenljiva brzina prenosa koji se ne odvija u realnom vremenu (npr. gledanje filma na Internetu).
4. Raspoloživa brzina prenosa (npr. prenos datoteka).

Navedene kategorije su korisne i za druge svrhe i za druge mreže. Konstantna brzina prenosa je pokušaj da se obezbeđivanjem konstantnog propusnog opsega i konstantnog kašnjenja simulira prenos kablom. Promenljivom brzinom prenosi se komprimovani video u kome su neke slike više komprimovane od drugih. Na taj način, slanje slike prepune detalja znači slanje mnogo bitova, dok snimak belog zida može da se kompri- muje u par bitova. Raspoloživa brzina prenosa namenjana je aplikacijama koje nisu osetljive na kašnjenje i neravnomemost, kao što je e-pošta.

5.4.2 Tehnike za postizanje dobrog kvaliteta usluga

Pošto smo se delimično upoznali sa zahtevima u pogledu kvaliteta usluga, pogledajmo kako se oni mogu zadovoljiti. Najpre odbacimo iluziju o čarobnom štapiću: nijedna pojedinačna tehnika ne obezbeđuje efikasan i pouzdan kvalitet usluga na optimalan način. Umesto toga, za obezbeđivanje kvaliteta usluga razvijeno je više tehnika, a u praksi se on često postiže njihovim kombinovanjem. U nastavku ćemo opisati neke od tih tehnika.

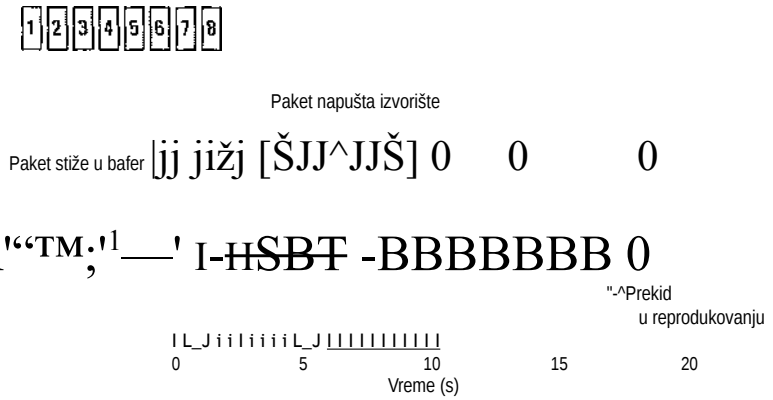
Obezbeđivanje viška resursa

Najjednostavnije je ako obezbedite takav kapacitet usmerivača, bafera i propusnog opsega, da paketi samo proleću mrežom. Međutim, takvo rešenje je skupo. Pa ipak, budući da projektanti stalno koriguju svoje mišljenje o dovoljnom kapacitetu, možda će takva tehnika jednom i ući u praksu. Telefonski sistem, u izvesnom smislu, obezbeđuje višak kapaciteta. Retko ćete kad podići slušalicu a da odmah ne začujete signal. Tu jednostavno ima toliko kapaciteta da se potrebama gotovo uvele može izići u susret.

Privremeno skladištenje

Tok podataka se kod primaoca može najpre smeštati u bafer, pa tek onda isporučivati. Privremenim skladištenjem (engl. *buffering)* paketa ne remete se pouzdanost i propusni opseg, kašnjenje se povećava, ali se otklanja neravnomernost. Za audio i video na zahtev, glavni problem je neravnomernost stizanja paketa, tako da opisana tehnika povećava kvalitet usluge.

Na slici 5-29 videli smo razliku između visokog i niskog stepena neravnomernosti. Na slici 5-31 vidimo tok paketa koji se isporučuju veoma neravnomerno. Server šalje paket 1 u trenutku $t = 0$ s, a on stiže klijentu u trenutku $t = 1$ s. Paket 2 kasni više i treba mu 2 s da stigne. Paketi koji stižu klijentu, redom se smeštaju u bafer.



Slika 5-31. Ujednačavanje izlaznog toka smeštanjem paketa u bafer.

U trenutku $t = 10$ s počinje reprodukovanje. U međuvremenu su paketi od prvog do šestog bili smešteni u bafer, tako da se iz njega mogu izvlačiti u pravilnim vremenskim intervalima, što obezbeđuje glatko reprodukovanje. Nažalost, paket 8 je toliko zakasnio da nije bio tu kada je na njega došao red - nastala je neugodna pauza u reprodukovanju zvuka ili slike. Problem bi se mogao izbeći kada bi se početak reprodukovanja još odložio, ali je za to potreban i veći bafer. Na komercijalnim Web lokacijama koje nude audio i video sadržaje za reprodukovanje tokom preuzimanja, koriste se plejeri koji podatke drže u baferu oko 10 sekundi pre nego što počnu da ih reprodukuju.

Ujednačavanje saobraćaja

U navedenom primeru, izvorišni računar šalje pakete u jednakim vremenskim intervalima, ali se u drugim situacijama oni mogu slati neravnomerno, što može da izazove zagušenje na mreži. Neravnomernost slanja je karakteristična za servere koji istovremeno šalju više tokova podataka, a omogućavaju i druge akcije, kao što su brzo premotavanje na kraj ili početak sekvence, proveru identiteta korisnika itd. Isto tako, opisani pristup (baferovanje) nije uvek moguće sprovesti, npr. u video konferencijama. Međutim, ako bi se nešto moglo učiniti na tome da server (ili računari, uopšte) emituje u pravilnim vremenskim intervalima, kvalitet usluge bi u načelu bio bolji. Opisaćemo sada tehniku **ujednačavanja saobraćaja** (engl *traffic shaping),* kojom se izglađuje saobraćaj na strani servera.

Ujednačavanje saobraćaja podrazumeva regulisanje prosečne *brzine* prenosa (i iznenadnih provala) podataka. Nasuprot tome, protokoli kliznih prozora o kojima smo ranije govorili ograničavaju količinu podataka koji se šalju u jednom pokušaju, a ne njihovu brzinu slanja. Tokom uspostavljanja veze, korisnik i podmreža (tj. mušterija i operater, tj. vlasnik podmreže) dogovaraju se o obliku saobraćaja u tom kolu, što se ponekad naziva **dogovaranje nivoa usluge** (engl. *service level agreement)*. Sve dok korisnik ispunjava svoju obavezu i šalje pakete onako kako je dogovoreno, operater obećava da će ih isporučivati na vreme. Ujednačavanjem saobraćaja smanjuju se zagušenja i operater može da ispuni svoje obećanje. Takvi dogovori su manje važni za prenos datoteka, ali su izuzetno važni za prenos podataka u realnom vremenu, npr, na audio i video vezama koje postavljaju oštre zahteve u pogledu kvaliteta usluge.

Sve u svemu, kada se primenjuje ujednačavanje saobraćaja, korisnik kaže operateru: Moj saobraćaj će izgledati približno ovako; možeš li da ga tako i preneseš? Ako se operater složi, ostaje pitanje kako će znati da se korisnik drži dogovora i šta može da preduzme ako utvrdi da ga se ne drži. Operater tad preduzima nešto što se u žargonu zove **policijski nadzor saobraćaja** (engl. *traffic policing)*. Ugovaranje oblika saobraćaja i njegovo kasnije nadziranje lakši su u podmrežama s virtuelnim kolima nego u datagramskim podmrežama. Međutim, čak se i u datagramskim podmrežama iste ideje mogu prirneniti u vezama transportnog sloja.

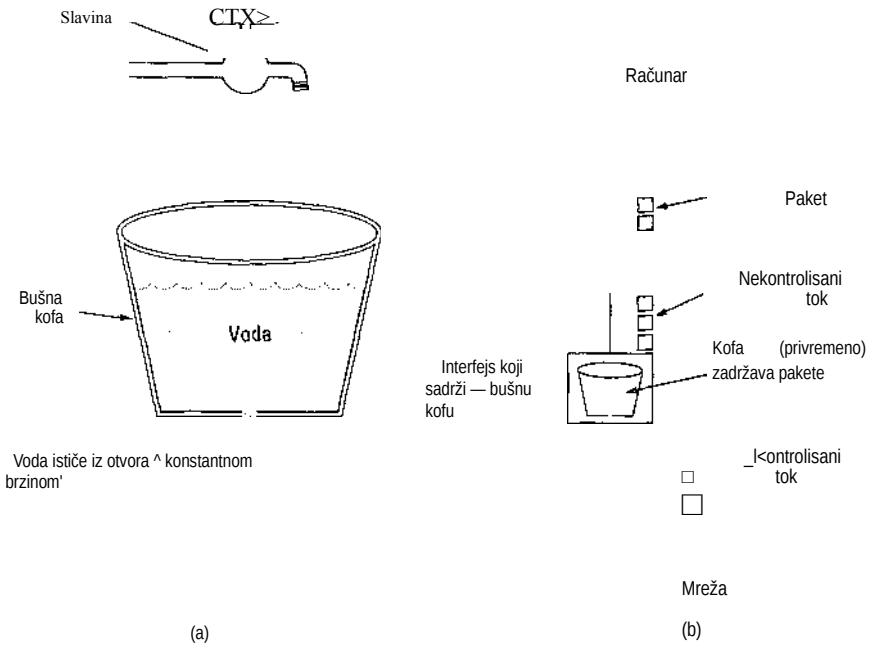
Algoritam bušne kofe

Zamislite kofu s malim otvorom na dnu, kao na slici 5-32(a). Bez obzira na brzinu kojom voda utiče u kofu, ona iz nje otiče konstantnom brzinom p kada u kofi ima imalo vode, a uopšte ne ističe kada je kofa prazna. Takođe, kada se kofa napuni, višak će početi da se preliva preko ivice (i neće proći kroz otvor na dnu kofe).

Sličan princip je primenljiv i na pakete, kao na slici 5-32(b). U osnovi, svaki raču- nar se s mrežom povezuje preko interfejsa koji sadrži analogiju bušne kofe - interni red čekanja konačne dužine. Ako paket stigne u red čekanja kada je ovaj pun, paket se odbacuje. Drugim recima, ako jedan ili više procesa pokušaju da pošalju paket kada se u redu čekanja već nalazi maksimalan broj paketa, novi paket se bez izuzetka odbacuje. Takav postupak može se hardverski ugraditi u interfejs ili ga može simuli- rati operativni sistem. Predložio gaje Turner (1986) i nazvao **algoritam bušne kofe** (engl. *leaky bucket algorithm)*. U stvari, on nije ništa drugo do sistem svrstavanja u red čekanja s konstantnom brzinom usluživanja, primenjen na pojedinačan server.

Računaru se dozvoljava da na svaki otkucaj sistemskog sata u mrežu pušta po jedan paket. Ponavljamo, to se može postići hardverski (na mrežnoj kartici) ili akcijom operativnog sistema računara. Taj mehanizam pretvara neravnomeran dotok podataka iz korisničkih procesa unutar računara u ravnomeran tok podataka koji se šalje u mrežu, izglađujući sve „špiceve“ i uveliko smanjujući mogućnost zagušenja.

Kada su svi paketi iste veličine (npr. ATM ćelije), algoritam se može prirneniti kao što je opisano. Međutim, kada su paketi različite veličine, često je mudrije da se po jednom otkucaju sata dozvoli slanje konstantnog broja bajtova, nego jednog paketa. Tako, kada se dozvoli 1024 bajta po otkucaju, može se poslati jedan paket od 1024 bajta, dva paketa po 512 bajtova, četiri paketa po 256 bajtova itd. Kada se vrednost brojača bajtova previše smanji, paket mora sačekati sledeći otkucaj sata.



Slika 5-32. (a) Bušna kofa s vodom, (b) „Bušna kofa“ s paketima.

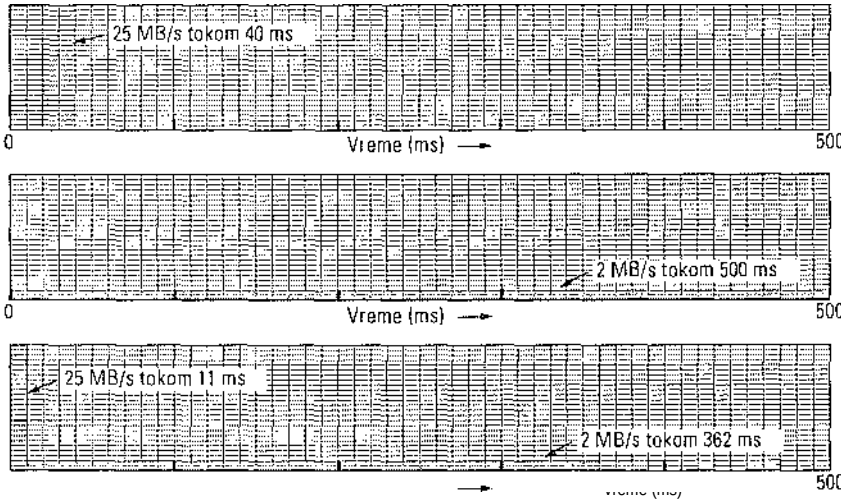
Algoritam bušne lcofe se lako ugrađuje. Bušna kofa predstavlja red čekanja konačne veličine. Kada paket stigne, ako ima mesta u redu, on mu se priključuje; ukoliko je red pun, paket se odbacuje. Pri svakom otkucaju sistemskog sata, iz reda se u mrežu šalje jedan paket (osim ako u redu čekanja nema nijednog paketa).

Algoritam koji umesto s paketima radi s bajtovima primenjuje se na sličan način. Pri svakom otkucaju sistemskog sata brojač se postavlja na vrednost *n.* Ako prvi paket u redu čekanja ima manje bajtova od tekuće vrednosti brojača, on se šalje a brojač se smanjuje za broj njegovih bajtova. Istovremeno se mogu poslati i dodatni paketi, sve dok je vrednost brojača dovoljno visoka. Kada vrednost brojača padne ispod dužine sledećeg paketa u redu čekanja, njegovo slanje se odlaže do sledećeg otkucaja sata kada se brojač ponovo postavlja na početnu vrednost.

Primer bušne kofe imate kod računara koji podatke šalje brzinom 25 miliona bajtova u sekundi (200 Mb/s) u mrežu koja radi istom brzinom. Usmerivači, međutim, mogu tu brzinu da prihvate samo u kratkim vremenskim intervalima (dok im se ne popune baferi). Oni kontinualno najbolje rade pri brzinama koje ne prelaze 2 miliona bajtova u sekundi. Pretpostavimo sada da podaci svake sekunde stižu u obliku rafala od milion bajtova koji traje 40 ms. Da bismo prosečnu brzinu smanjili na 3 MB/s, mogli bismo upotrebiti bušnu kofu kapaciteta C = 1 MB i brzine isticanja p= 2 MB/s. To znači da se rafali veličine do 1 MB mogu obraditi bez gubitaka i raspodeliti na vremenski interval od 500 ms, ma kako brzo stizali.

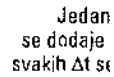
Na slici 5-33(a) vidimo rafalni dotok podataka u bušnu kofu; brzina stizanja je 25 MB/s, a rafal traje 40 ms. Na slici 5-33(b) prikazan je izlazni tok koji se „cedi“ konstantnom brzinom 2 MB/s, tokom 500 ms.

Algoritam kofe sa žetonima



Slika 5-33. (a) Ulazni tok u bušnu kofu, (b) Izlazni tok iz bušne kofe. Izlazni tok iz kofe sa žetonima kapacitet a (c) 250 KB, (d) 500 KB i (e) 750 KB. (0 Izlazni tok iz kofe sa žetonima kapacitet a 500 KB koji puni bušnu kofu izlazne brzine 10 MB/s.

Algoritam bušne kofe kruto nameće izlaznom toku zadatu prosečnu brzinu, bez obzira na stepen neravnomernosti saobraćaja. Za mnoge oblasti primene korisnije je da se izlazni tok malo ubrza kada naiđe rafal podataka, pa je potreban fleksibilniji algoritam koji, ako je to moguće, nikada ne gubi podatke. Takav je **algoritam „kofe sa žetonima“** (engl. *token bucket algorithm).* Njegova „lcofa“ sadrži žetone koje gene- riše sistemski sat - po jedan svakih *At* sekundi. Na slici 5-34(a) vidimo kofu s tri žetona i pet paketa koji čekaju na slanje. Da bi paket bio poslat, on mora da uhvati i uništi jedan žeton. Na slici 5-34(b) vidimo da su tri od pet paketa to uspela, ali preostala dva paketa moraju da čekaju generisanje novih žetona.



(a) (b)

Slika 5-34. Algoritam kofe sa žetonima, (a) Pre akcije, (b) Posle akcije.

Algoritam kofe sa žetonima ujednačava saobraćaj na drugačiji način od algoritma bušne kofe. Algoritam bušne kofe ne dozvoljava računarima koji trenutno nemaju ništa za slanje da rezervišu pravo na kasnije rafalno slanje podataka. Algoritam kofe sa žetonima to dozvoljava, s rafalima čija dužina iznosi najviše koliko i kapacitet kofe *n.* To znači da se rafali od najviše *n* paketa mogu slati odjednom, što će malo poremetiti izlazni tok, ali je to istovremeno i brza reakcija na ulazni rafalni tok.

Druga razlika između dva algoritma jeste to što se kod kofe sa žetonima paketi nikada ne gube - kada se kofa prepuni, odbacuju se samo žetoni. Prepunjena bušna kofa odbacivaće pakete.

I u algoritam kofe sa žetonima može se uneti mala izmena: da žeton ne predstavlja pravo na slanje jednog paketa, već pravo na slanje *k* bajtova. Tada se paket može poslati samo ako u kofi ima dovoljno žetona za sve njegove bajtove. Višak žetona čuva se za sledeći ciklus.

Algoritmi bušne kofe i kofe sa žetonima mogu se iskoristiti i za umirivanje saobraćaja između usmerivača, a ne samo saobraćaja koji potiče od računara, kao u prethodnim primerima. Međutim, tu postoji barem jedna jasna razlika: kofa sa žetonima koja upravlja tokom iz računara može taj tok da zaustavi ako tako nalažu pravila. S druge strane, ne dozvoliti usmerivaču da se isprazni, u situaciji kada mu s druge strane i dalje pristižu podaci, može da izazove gubljenje podataka.

Realizacija osnovnog algoritma kofe sa žetonima svodi se na definisanje promenljive koja broji žetone. Brojač povećava svoju vrednost za jedinicu svakih *At* sekundi, a smanjuje je za jedinicu kada se pošalje paket. Kada vrednost brojača dostigne nulu, ne može se poslati nijedan paket. U varijanti s brojanjem bajtova, vrednost brojača se svakih *At* sekundi

povećava za *k* bajtova, a smanjuje u zavisnosti od dužine poslatog paketa.

Kofa sa žetonima u stvari uobličava neravnomerno poslate podatke u rafale određene maksimalne dužine. Pogledajte, na primer, sliku 5-33(c). Tu imamo kofu sa žetonima kapaciteta 250 KB. Žetoni u nju stižu tempom koji dozvoljava izlaznu brzinu podataka 2 MB/s. Ako pretpostavimo da je kofa puna žetona kada stigne rafal podataka dužine 1 MB, ona jedan njihov deo može poslati brzinom 25 MB/s tokom oko 11 ms, a potom brzinu slanja mora smanjiti na 2 MB/s dok ne pošalje ostatak rafala.

Izračunavanje vremena tokom koga se deo rafala šalje maksimalnom brzinom ne svodi se na jednostavno deljenje 1 MB sa 25 MB/s jer tokom njegovog slanja u kofu stižu novi žetoni. Ako je trajanje rafala *S* sekundi, kapacitet kofe sa žetonima C baj- tova, brzina stizanja žetona p bajtova u sekundi, a maksimalna brzina slanja iz kofe *M* bajtova u sekundi, izlazni rafal sadržace najviše *C* + p.S' bajtova. Takođe znamo da broj bajtova u delu rafala dužine *S* koji se šalje maksimalnom brzinom iznosi *MS.* Odatle imamo

C + pS = *MS*

Kada tu jednačinu rešimo po 5, dobij amo 5 = *C/(M* - p). Vrednosti u našem pri- meru su C = 250 KB, M = 25 MB/s i p = 2 MB/s, pa za trajanje rafala dobijamo oko 11 ms. Slike 5-33(d) i 5-33(e) prikazuju kofu sa žetonima kapaciteta 500 KB, odnosno 750 KB.

Propuštanje snažnih rafala može da bude problem algoritma kofe sa žetonima, bez obzira na to što se trajanje (dužina) rafala može regulisati izborom vrednosti za p i *M*. Često je poželjno da se smanji maksimalna brzina prenosa, ali ne u toj meri kao s bu- šnom kofom.

Saobraćaj se na takav način može umiriti ako se bušna kofa postavi iza kofe sa žetonima. Brzina „curenja“ iz bušne kofe treba daje veća od brzine „curenja“ iz kofe sa žetonima, ali manja od maksimalne brzine mreže. Na slici 5-33(f) prikazanje izlazni tok koji se dobija kada se iza kofe sa žetonima kapaciteta 500 KB postavi bušna kofa brzine „curenja“ 10 MB/s.

Policijsko nadziranje ove šeme pomalo je komplikovano, jer mreža u stvari treba da simulira algoritam i da onemogući slanje većeg broja paketa ili bajtova nego što je dozvoljeno. Bez obzira na to, opisane alatke ujednačavaju mrežni saobraćaj u takvom stepenu da se može odgovoriti na zahteve za kvalitetom usluga.

Rezervisanje resursa

Mogućnost ujednačavanja ponuđenog saobraćaja prvi je korak ka obezbeđivanju kvalitetne usluge. Tu mogućnost ćemo efikasno iskoristiti samo ako se svi paketi usmere istom putanjom. Ukoliko ih nasumično raspemo po svim usmerivačima u mreži, teško da možemo išta garantovati. Zbog toga se između izvorišta i odredišta mora uspostaviti nešto slično virtuelnom kolu i svi paketi koji pripadaju toku slati tom putanjom.

Kada se putanja uspostavi, onda se duž nje mogu rezervisati resursi koji će obe- zbediti potreban kapacitet. Potencijalno se mogu rezervisati tri vrste resursa:

1. Propusni opseg.
2. Prostor u baferima.
3. Vreme mikroprocesora.

Najjasniji je zahtev za propusnim opsegom. Ako je za tok potreban propusni opseg 1 Mb/s, a izlazna linija ima kapacitet 2 Mb/s, onda neće uspeti pokušaj da se istovremeno tri takva toka pošalju kroz nju. Rezervisanje propusnog opsega, dakle, znači da ne treba

„prebukirati“ izlaznu liniju.

Prostor u baferima takođe je resurs za kojim je uvek velika potražnja. Kada paket stigne kablom, obično ga sam hardver smešta na mrežnu karticu. Usmerivački softver ga tada kopira u bafer u radnoj memoriji i bafer smešta u red čekanja za slanje određenom izlaznom linijom. Ako trenutno nije slobodan nijedan bafer, paket se mora odbaciti. Kada želimo da pružimo kvalitetnu uslugu, neke bafere možemo rezervisati za određeni tok podataka, tako da on ne mora da se za njih nadmeće s dragim tokovima. Za takav tok uvek će postojati slobodan bafer, sve do određenog maksimuma.

I procesorsko vreme može da bude traženi resurs. Mikroprocesoru treba vremena da obradi svaki paket, tako da usmerivač može obraditi samo određen broj paketa u sekundi. Kada povedemo računa o tome da mikroprocesor nikada ne bude previše opterećen, tada će svaki paket moći da se obradi na vreme.

Na prvi pogled može izgledati da usmerivač kome za obradu jednog paketa treba 1 ps, može da obradi milion paketa u sekundi. Ta računica nije tačna zato što uvek postoje periodi mirovanja zbog statističkih fluktuacija opterećenja. Kad bi mikroprocesoru za obavljanje posla bio potreban svaki njegov ciklus, već bi propuštanje samo nekoliko ciklusa zbog zastoja u prilivu podataka izazvalo zaostatak koji nikako ne bi mogao da nadoknadi.

Međutim, i pri opterećenju koje je nešto manje od teorijskog kapaciteta mogu da porastu redovi čekanja i kašnjenje. Razmotrite situaciju kada paketi stižu u neredovnimintervalima, prosečnom brzinom *X* paketa usekundi. Procesorsko vreme potrebno za obradusvakog paketa takođe še menja na slučajan načini njegovabrzina obrade u prošeku iznosi p paketa u sekundi. Pod uslovom da i vreme stizanja paketa i trajanje njihove obrade slede Poasonovu raspodelu, na osnovu teorije svrstavanja u redove čekanja može se dokazati daje srednje kašnjenje (7j paketa jednako

$$T = \frac{1}{p}\times\frac{1}{1 - A/p} = \frac{1}{p}\times\frac{1}{1 - p}$$

gde p = X/p predstavlja iskorišćenje mikroprocesora. Prvi činilac (1/p) predstavlja trajanje usluge u odsustvu konkurencije. Dragi činilac je usporenje zbog nadmetanja s drugim tokovima. Na primer, ako je *X* = 950.000 paketa/s, ap = 1.000.000 paketa/s, tada je p = 0,95, a paketi će u prošeku, umesto 1 ps, kasniti 20 ps. U njega ulazi vreme čekanja u redu i trajanje usluge, što postaje očigledno kada je opterećenje vrlo malo (A,/p « 0). Ako duž putanje postoji, recimo, 30 usmerivača, kašnjenje bi bilo 600 ps samo zbog stajanja paketa u redovima čekanja.

Kontrola pristupa

Stigli smo do tačke kada dolazni tok podataka iz nekog izvora može da bude na odgovarajući način ujednačen i usmeren po mogućnosti duž jedinstvene putanje na kojoj se mogu unapred rezervisati resursi usmerivača. Kada se takav tok ponudi usmerivaču, on može da ga prihvati ili da ga odbije, zavisno od svojih kapaciteta i trenutnih obaveza koje ima prema drugim tokovima.

Odluka o prihvatanju ili odbijanju ponuđenog toka ne može se doneti jednostavnim poređenjem zahtevanih resursa (propusnog opsega, prostora u baferima i proce- sorslcog

vremena) s raspoloživim resursima usmerivača. Najpre, iako neke aplikacije znaju koliki im propusni opseg treba, malo ih je koje išta mogu da kažu o baferima i procesorskom vremenu, pa je u najmanju ruku tok potrebno definisati na neki drugi način. Zatim, neke aplikacije lako tolerišu neispunjenje obećanja, dok druge to ne mogu. Na kraju, neke aplikacije su voljne i da se cenlcaju oko parametara toka. Na pri- mer, program za gledanje filmova koji normalno radi brzinom 30 kadrova u sekundi, možda će se vratiti na brzinu 25 kadrova u sekundi ako ne može da dogovori dovoljan propusni opseg. Slično tome, mogu se podešavati i druga svojstva, npr. broj piksela po kadru, propusni opseg za zvuk itd.

Pošto mnoge strane mogu učestvovati u dogovaranju (pošiljalac, primalac i svi usmerivači na putanji koja ih povezuje), tok se mora tačno opisati pomoću parametara koji su podložni dogovaranju. Skup takvih parametara naziva se specifikacija toka (engl,*jlow specification)*. Najčešće, pošiljalac (npr. video server) pravi i šalje specifikaciju toka predlažući parametre koje bi želeo da koristi. Dok specifikacija napreduje duž putanje, pregleda je svaki usmerivač i po potrebi menja. Izmenama se tok može samo usporavati, nikako ubrzavati (npr. brzina prenosa podataka može biti samo manja od predložene, nikako veća). Kada specifikacija stigne na odredište, mogu se definitivno utvrditi svi parametri.

Primer onoga što može biti u specifikaciji vidite na slici 5-35, koja se zasniva na RFC dokumentima 2210 i 2211. Navodi se pet parametara, a prvi (*Brzina generisanja žetona)* odnosi se na broj bajtova koji se u sekundi stavlja u kofu. To je maksimalna podržana brzina kojom pošiljalac može da emituje, dobijena kao prošek u dužem vremenskom intervalu.

Parametar	**Jedinica**
Brzina generisanja žetona	Bajtovi/s
Kapacitet kofe sa žetonima	Bajtovi
Maksimalna brzina slanja podataka	Bajtovi/s
Minimalna veličina paketa	Bajtovi
Maksimalna veličina paketa	Bajtovi

Slika 5-35. Primer specifikacije toka.

Drugi parametar je kapacitet kofe u bajtovima. Ako, na primer *Brzina generisanja žetona* iznosi 1 Mb/s, a *Kapacitet kofe sa žetonima* - 500 KB, kofa se može neprekidno puniti tokom 4 sekunde, pre nego što se prelije (u odsustvu bilo kakvog prenosa). Svi žetoni generisani posle toga nestaju.

Treći parametar, *Maksimalna brzina slanja podataka,* predstavlja najveću brzinu koju pošiljalac ne sme da pređe čak ni u kratkim vremenskim intervalima.

Poslednja dva parametra definišu najmanju i najveću veličinu paketa, zajedno sa zaglavljima transportnog i mrežnog sloja (npr. TCP i IP zaglavljima). Minimalna veličina je važna jer obrada svakog paketa traje isto, bez obzira na njegovu dužinu. Usmerivač može biti pripremljen da u sekundi obradi 10.000 paketa od po 1 KB, ali ne i 100.000 paketa od po 50 bajtova, iako je u dragom slučaju brzina prenosa podataka manja. Maksimalna veličina paketa je važna zbog internih ograničenja mreže koja se ne smeju prekoračiti. Na primer, ako deo putanje vodi preko Ethemeta, paket ne srne biti veći od 1500 bajtova, bez obzira na to šta ostatak mreže može da obradi.

Zanimljivo je kako usmerivač pretvara specifikaciju toka u skup rezervacija odgovarajućih resursa. To preslikavanje zavisi od realizacije i nije standardizovano. Pretpostavimo da usmerivač može da obradi 100.000 paketa u sekundi. Ako mu se ponudi tok brzine 1 MB/s s najmanjim i najvećim paketom od po 512 bajtova, usmerivač može da sračuna da će iz tog toka dobijati 2048 paketa u sekundi. U tom slučaju, on za njega mora da rezerviše 2% procesorskog vremena, možda i više ako želi da izbegne dugačke redove čekanja. Ukoliko se usmerivač drži pravila da nikada ne dodeljuje više od 50% procesorskog vremena (što podrazumeva dvostruko kašnjenje), a već je 49% zauzeto, tada se ponuđeni tok mora odbiti. Sličan proračun vrši se i za druge resurse.

Sto je strožija specifikacija toka, usmerivači će je pre prihvatiti. Ako se specifikacijom zahteva da *Brzina generisanja žetona* bude 5 MB/s, ali da veličina paketa može da varira između 50 i 1500 bajtova, tada će brzina prenosa paketa varirati između 3500 paketa u sekundi i 105.000 paketa u sekundi. Usmerivač se može uspaničiti i odbiti takav tok, dok bi s minimalnom veličinom paketa od 1000 bajtova, pri 5 MB/s, takav tok mogao prihvatiti.

Proporcionalno usmeravanje

Algoritmi za usmeravanje većinom pokušavaju da pronađu najbolju putanju do svakog odredišta, a zatim njom šalju sav saobraćaj za to odredište. Za obezbeđenje višeg kvaliteta usluga predložen je drugačiji pristup, prema kome se saobraćaj za svako odredište deli u više putanja. Pošto usmerivači u nečelu nemaju potpun uvid u saobraćaj na čitavoj mreži, usmeravanje duž više putanja može se izvršiti jedino na osnovu lokalno dostupnih informacija. Najjednostavnije je da se saobraćaj podeli jednako na sve izlazne linije ili, pak, proporcionalno kapacitetu svake od njih. Međutim, postoje i složeniji algoritmi (Nelakuditi i Zhang, 2002).

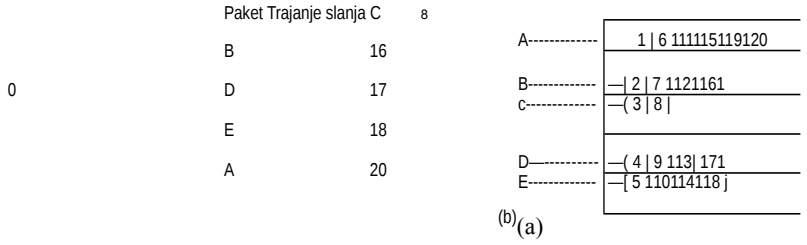
Raspoređivanje paketa

Ako usmerivač radi s više tokova, postoji opasnost da jedan tok prigrabi previše njegovog kapaciteta i ostavi drage tokove na cedilu. Obrada paketa redosledom njihovog stizanja znači da agresivan pošiljalac može da zauzme veći deo kapaciteta usmerivača koji se nađe na putanji njegovog paketa, snižavajući tako kvalitet usluga za druge tokove. Da bi se takvi incidenti predupredili, smišljeno je više algoritama za raspoređivanje paketa (Bhatti i Crowcroft, 2000).

Jedan od prvih bio je algoritam za **ravnopravnu obradu redova čekanja** (engl. *fair queueing)* (Nagle, 1987). Suština je u tome da usmerivači svakoj izlaznoj liniji pridruže zasebne redove čekanja za svaki tok. Kada se linija oslobodi, usmerivač

ciklično proverava redove čekanja uzimajući iz svakog samo prvi paket. Na taj način, kada se *n* računara nadmeće za istu izlaznu liniju, svaki računar dobija šansu da unutar *n* poslatih paketa pošalje jedan svoj paket. Računar može i da brže emituje pakete, ali to ne utiče na ovaj odnos.

Algoritam ipak nije bio bez mane: davao je veći propusni opseg računarima koji su slali veće pakete. Demers i saradnici (1990) predložili su poboljšanje u kome se paketi iz redova čekanja šalju prema veličini. Algoritam ciklično, više puta, proverava iste redove čekanja dok ne utvrdi veličinu prvog paketa u svakom redu čekanja, a zatim pakete sortira za slanje, od najmanjeg do najvećeg. Algoritam je prikazan na slici 5-36.



Slika 5-36. (a) Usmerivač s pet paketa u redovima čekanja za liniju *O*. (b) Trajanje slanja svakog od pet paketa.

Na slici 5-36(a) vidimo pakete dužine 2 do 6 bajtova. Pri prvom otkucaju (virtuel- nog) sata šalje se prvi bajt paketa s linije *A*. Zatim se šalje prvi bajt paketa s linije *B* itd. Tako će u potpunosti biti poslat najpre paket na liniji C, posle 8 otkucaja sata. Re- dosled slanja paketa prikazanje na slici 5-36(b). Ako u međuvremenu ne stignu novi paketi, postojeći paketi će biti poslati prikazanim redosledom, od C do A.

Mana ovog algoritma je to što svim računarima daje isti prioritet. U mnogim situacijama poželjnije je da se video serverima obezbedi veći propusni opseg, nego serve- rima datoteka, tako da se njima može dozvoliti slanje dva ili više bajtova po jednom otkucaju sata. Tako modifikovan algoritam zove se algoritam za ponderisanu ravnopravnu obradu redova čekanja (engl. *weighted fair queueing*); on je u širokoj upotrebi. Ponekada se propusni opseg ravnomemo deli na sve izlazne tokove. Efikasnu ugradnju algoritma razradili su Shreedhar i Varghese (1995). Sve više se, međutim, paketi kroz usmerivače i skretnice prosleđuju hardverskim putem (Elhanany i sar., 2001).

5.4.3 Integrisane usluge

Grupa IETF je između 1995. i 1997. godine uložila dosta truda da razvije arhitekture pogodne za prenos multimedijskih sadržaja koji se reprodukuju u realnom vremenu. Rezultat tog posla prikazanje u više od dvadeset RFC dokumenata, a najvažniji su 2205-2210. Ti, takozvani algoritmi zasnovani na toku podataka (engl. *flow-based algorithms*) ili integrisane usluge (engl. *integrated services*) namenjeni su aplikacijama koje rade kako s jednosmernim, tako i s visesmernim slanjem podataka. Primer

za prve je (jedan) korisnik koji gleda video sekvencu dok je preuzima s lokacije koja objavljuje vesti. Primer za drage je skup digitalnih TV stanica koje svoje programe emituju kao tokove IP paketa ka mnogim primaocima na različitim mestima. U nastavku ćemo govoriti o višesmernom emitovanju jer je jednosmerno emitovanje samo njegov specijalan slučaj.

U mnogim aplikacijama koje rade s višesmernim slanjem, članstvo u grupama može se dinamički menjati. Na primer, više korisnika učestvuje u video konferenciji, a zatim, kada im postane dosadno, isključuju se jedan po jedan, neko da bi gledao 157.3. nastavak srceparajuce serije, a neko fudbal. U takvim okolnostima teško je re- zervisati propusni opseg, pošto svaki pošiljalac stalno mora da vodi evidenciju o tome ko se priključuje, a ko isključuje iz njegove konferencije. U sistemu namenjenom za prenos TV programa milionima petplatnika, to i nije moguće.

Protokol za rezervisanje resursa

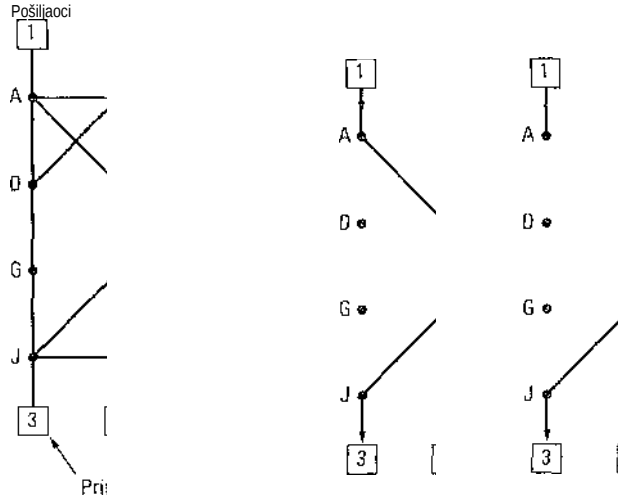
Glavni protokol koji je grupa IETF osmislila za arhitekturu integrisanih usluga jeste **protokol za rezervisanje resursa** (engl. *Resource reSerVation Protocol, RSVP).* Opisan je u RFC dokumentu 2205 i dragim RFC dokumentima. Protokol se koristi isključivo za rezervisanje, dok dragi protokoli brinu o slanju podataka. Protokol RSVP omogućava većem broju pošiljalaca da šalju podatke različitim grupama primalaca, dozvoljava pojedinačnim primaocima da slobodno menjaju kanale, optimizuje kori- šćenje propusnog opsega i istovremeno otklanja zagušenja.

Protokol u svom najjednostavnijem vidu koristi visesmerno emitovanje kroz raz- granato stablo, kao sto smo ranije opisali. Svakoj grupi se dodeljuje grupna adresa. Kada šalje pakete grupi, pošiljalac u njih stavlja adresu grupe. Standardni algoritam za višesmerno usmeravanje tada gradi razgranato stablo koje pokriva sve članove grupe. Algoritam za usmeravanje nije deo algoritma RSVP. Jedinu razliku u odnosu na normalno višesmerno emitovanje predstavlja to što se ka grupi periodično šalju dodatne informacije pomoću kojih se usmerivačima duž putanje nalaže da u svojoj memoriji održavaju određene strukture podataka.

Razmotrite kao primer mrežu prikazanu na slici 5-37(a). Računati 1 i 2 su više- smerni emiteri, a računati 3, 4 i 5 - višesmerni prijemnici. Ovde su pošiljaoci i primaoci razdvojeni, ali se u načelu ta dva skupa mogu i preklapati. Višesmerna stabla za računare 1 i 2 prikazana su na slikama 5-37(b) i 5-37(c).

Da bi imao bolji prijem i izbegao zagušenje, svaki primalac iz grupe može da pošiljaocu pošalje poruku sa zahtevom za rezervisanje. Ta poruka se prosleđuje pošiljaocu algoritmom za ispitivanje izvorišta, o kome smo ranije govorili. Pri svakom skoku, odgovarajući usmerivač zapaža zahtev i rezerviše neophodan propusni opseg. Ako nema toliki slobodan opseg, vraća poruku o neuspešnom rezervisanju. Dok poruka stigne do izvorišta, usput će na svim usmerivačima duž razgranatog stabla biti rezervisan potreban propusni opseg.

Primer takvog rezervisanja prikazan je na slici 5-38(a). Tu je računar 3 zahtevao kanal do računara 1. Kada se kanal uspostavi, paketi od računara 1 do računara 3 mogu da teku bez bojazni od zagušenja. Pogledajte šta se događa kada računar 3 potom rezerviše kanal ka drugom pošiljaocu, računara 2, tako da korisnik istovremeno



(a) (b) (c)

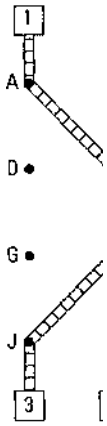
Slika 5-37. (a) Mreža, (b) Višesmerno razgranalo slabio za računar 1. (c) Višesmerno razgranalo stablo za računar 2.

može da gleda dva televizijska programa. Rezervisana je i druga putanja, kao što prikazuje slika 5-38(b). Obratite pažnju na to da su od računara 3 do usrnerivača *E* potrebna dva zasebna kanala, pošto se prenose dva nezavisna toka podataka.

Na kraju, na slici 5-38(c), računar 5 odlučuje da i on gleda program koji emituje računar 1, pa zato traži rezervaciju. Najpre se namenski propusni opseg rezerviše sve do usrnerivača *H,* Međutim, taj usmerivač primećuje da već ima dotok podataka od računara 1, tako da propusni opseg ne mora da rezerviše i drugi put. Obratite pažnju na to da su računari 3 i 5 mogli zahtevati različit propusni opseg (npr. računar 3 vezan je za crno-beli televizor, pa mu ne trebaju bitovi koji definišu boju), pa rezervisani kapacitet mora na prvom mestu da zadovolji potrebe pohlepnijeg primaoca.

Kada traži rezervaciju, primalac može (ali ne mora) da navede jedno ili više izvorišta od kojih želi da prima podatke. On može i naglasiti da li je taj njegov izbor konačan tokom trajanja rezervacije ili zadržava pravo da kasnije promeni izvorište. Usmerivači takve informacije koriste da bi optimalno rasporedili propusni opseg. Konkretno, dva primaoca se smeštaju na istu putanju samo ako oba naglase da kasnije neće menjati izvorište.

Ovakva strategija se primenjuje u potpuno dinamičkom okruženju jer se rezervisani propusni opseg ne vezuje za izbor konkretnog izvorišta. Nakon što primalac rezerviše propusni opseg, on može da promeni izvorište i da zadrži deo rezervisane putanje koji ga spaja s novim izvorištem. Ako računar 2 emituje više video tokova, računar 3, na primer, može da prelazi s jednog toka na drugi ne menjajuci ništa u rezervaciji: usmerivaču je svejedno koji program gleda korisnik.



Slika 5-38. (a) Računar 3 zahteva kanal do računara **1,** (b) Računar 3 zatim zahteva drugi kanal, do računara 2. (c) Računar 5 zahteva kanal do računara 1.

5.4.4 Diferencirane usluge

Algoritmi zasnovani na toku podataka mogu da ponude dobar kvalitet usluga jednom ili većem broju tokova zato što su u stanju da duž putanje rezervišu sve potrebne resurse. Međutim, ni oni nisu bez mana. Za svaki tok kojim oni upravljaju potrebno je prethodno podešavanje (uspostavljanje rezervisane putanje), što ne zvuči dobro ako se radi o hiljadama ili milionima tokova. Isto tako, oni u usmerivačima za svaki tok održavaju određeno stanje, koje se nepovratno gubi u slučaju otkazivanja usme- rivača. Najzad, u kodu usmerivača prave se suštinske izmene koje se prilikom uspostavljanja toka na složen način razmenjuju između usmerivača. Zbog svega toga, do danas postoji malo realizacija algoritma RSVP ili bilo čega sličnog tome.

Imajući rečeno u vidu, grupa IETF je razvila i jednostavniji pristup za obezbeđenje kvaliteta usluga, pristup koji se najvećim delom može realizovati lokalno na svakom

usmerivaču, bez potrebe da se prethodno podešava i bez uključivanja cele putanje u proces. To su takozvane **usluge s više klasa kvaliteta** (engl. *class-based quality of service)*. Grupa IETF je standardizovala i odgovarajuću arhitekturu - **diferencirane usluge** (engl *differentiated services, DS),* koja je opisana u RFC dokumentima 2474, 2475 i brojnim drugim dokumentima. Obradićemo je u nastavku.

Diferencirane usluge može da ponudi skup usmerivača koji obrazuju administrativni domen (npr. davalac Internet usluga ili telefonska kompanija). Administracija definiše skup klasa usluga sa odgovarajućim pravilima prosleđivanja. Ako se korisnik pretplati na diferencirane usluge, njegovi paketi mogu sadržati polje *Tip usluge,* pri čemu se nekim klasama obezbeđuje bolja usluga (npr. prvoklasna). Za saobraćaj unutar određene klase može se zahtevati da bude određenog oblika, npr. sličan saobraćaju iz bušne kofe s nekom definisanom brzinom „curenja". Operater koji ima poslovnog duha može da zaračunava dodatnu cenu za svaki paket koji se pošalje prvom klasom ili može da dozvoli slanje *N* prvoklasnih paketa za fiksnu mesečnu doplatu. Imajte u vidu da ova šema ne zahteva prethodno podešavanje, rezervisanje resursa, niti

vremenski zahtevno dogovaranje s kraja na kraj veze za svaki tok, kao što je slučaj kod integrisanih usluga. Zbog toga se diferencirane usluge lako realizuju.

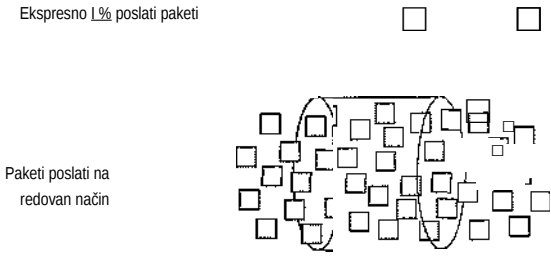
Usluge s više klasa kvaliteta pojavljuju se i u drugim oblastima. Na primer, kompanije za isporučivanje pošte često nude dostavu pošiljaka unutar 24 sata, unutar dva, odnosno unutar tri dana. Avio-kompanije nude prvu klasu, poslovnu klasu i klasu za ostalu „raju". Transkontinentalni vozovi često nude više klasa usluga. Čak i pariška podzemna železnica ima dve klase usluga. Za pakete podataka, klase se između ostalog (ali verovatno ne u pogledu većih Ethernet paketa) mogu razlikovati po kašnjenju, (ne)ravnomernosti toka i verovatnoći da budu odbačeni u slučaju zagušenja.

Da bismo razliku između kvaliteta usluga zasnovanog na toku i onog zasnovanog na klasama jasnije razumeli, razmotrimo primer Internet telefonije. U šemi zasnovanoj na toku, svaki telefonski poziv dobija zasebne resurse i garancije. U šemi zasnovanoj na klasama, svi telefonski pozivi zajedno dobijaju resurse rezervisane za đatu klasu telefonije. Te im resurse ne može oteti ni klasa prenosa datoteka, niti bilo koja druga klasa, ali nijedan telefonski poziv ne može da rezerviše resurse samo za sebe.

Ekspresno prosleđivanje

Svaki operater može da bira klase usluga koje će da ponudi, ali pakete između podmreža često upućuju različiti operateri, pa IETF radi na klasama usluga koje neće zavisiti od mreže. Najjednostavnija klasa je ekspresno prosleđivanje (engl. *expedited forwarding),* pa počnimo od nje. Ona je opisana u RFC dokumentu 3246.

Ekspresno prosleđivanje je vrlo jednostavno. Mogu da postoje dve vrste usluga: redovne (engl. *regular)* i ekspresne (engl. *expedited).* Očekuje se da najveći deo saobraćaja bude redovan, ali njegov mali deo čine i ekspresno poslati paketi. Takvi paketi treba da prođu mrežom kao da na njoj nema drugih paketa. Simboličan prikaz takvog „dvocevnog" sistema dat je na slici 5-39. Imajte naumu da ipak postoji samo jedna fizička linija. Dve logičke cevi prikazane na slici predstavljaju način rezervisanja propusnog opsega, a ne dragu fizičku liniju.



Slika 5-39. Ekspresno poslatim paketima izgleda kao da na mreži nema saobraćaja.

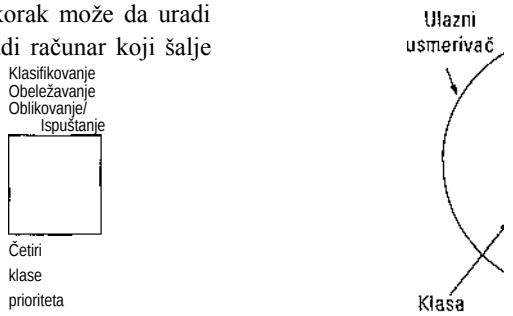
Jedan od načina realizacije opisane strategije jeste programiranje po dva reda čekanja za svaku izlaznu liniju usmerivača: jedan za redovan i jedan za ekspresni saobraćaj. Kada paket stigne u usmerivač, on se svrstava u odgovarajući red čekanja. Paketi se raspoređuju na način sličan ponderisanoj ravnopravnoj obradi redova čekanja. Na primer, ako ekspresni saobraćaj čini 10% ukupnog saobraćaja, a na redovan

otpada 90%, 20% propusnog opsega moglo bi se dodeliti ekspresnom, a ostatak redovnom saobraćaju. Time ekspresni saobraćaj dobija dvaput veći propusni opseg od potrebnog da bi se minimizovalo kašnjenje paketa. Takvo dodeljivanje može se postići slanjem jednog ekspresnog paketa na svaka četiri redovna (pod uslovom da su veličine paketa u obe klase raspodeljene slično). Na taj način, pretpostavlja se da će ekspresni paketi pred sobom videti neopterećenu mrežu, čak i kada je u njoj gust saobraćaj.

Garantovano prosleđivanje

Nešto složenija šema za obezbeđivanje klasa usluga zove se garantovano prosleđivanje (engl. *assured forwarding)*. Opisana je u RFC dokumentu 2597. Ona predviđa četiri klase usluga, svaku sa svojim resursima. Osim toga, u njoj se definišu tri nivoa verovatnoće za odbacivanje paketa koji naiđu na zagušenje: nizak, srednji i visok. Uzeti zajedno, ovi parametri definišu 12 klasa usluga.

Slika 5-40 prikazuje jedan način obrade paketa kada se prosleđuju garantovano. Prvi korak je da se paketi svrstaju u jednu od četiri klase prioriteta. Taj korak može da uradi pošiljalac (kao na slici) ili ulazni (prvi) usmerivač. Bolje je da to uradi računar koji šalje



pakete jer bolje zna koji paket pripada kom toku.

Paketi

čekanja

Slika 5-40. Moguća realizacija toka podataka pri garantovanom prosleđivanju.

Drugi korak je obeležavanje paketa prema klasama. Za to je potrebno neko polje u zaglavlju. Srećom u IP zaglavlju paketa postoji 8-bitno polje *Tip usluge*, o kome ćemo ubrzo govoriti. RFC dokumentom 2597 nalaže se da se šest bitova iz ovog polja iskoriste za označavanje klase, a dva ostave za postojeće i eventualne buduće klase.

Paketi zatim u trećem koraku prolaze kroz filtar za oblikovanje/odbacivanje koji neke od

njih usporava ili odbacuje da bi na prihvatljiv način ujednačio četiri toka, koristeći, na primer, algoritam bušne kofe ili kofe sa žetonima. Ako ima previše paketa, neki od njih mogu ovde biti odbačeni prema pravilima odbacivanja. Mogući su i složeniji sistemi koji uzimaju u obzir metriku ili povratne informacije.

U prikazanom primeru sva tri koraka se izvode kod pošiljaoca, tako da se izlazni tok upućuje usmerivaču preko koga je pošiljalac povezan s mrežom. Treba naglasiti da opisani postupak može da sprovede i specijalan mrežni softver, čak i operativni sistem, da bi se izbeglo menjanje postojećih aplikacija.

5.4.5 Komutiranje paketa na osnovu oznaka i MPLS

Dok je grupa IETF razrađivala integrisane i diferencirane usluge, proizvođači usmerivača su razvijali bolje metode prosleđivanja. Svoje napore su usmerili ka tome da ispred svakog paketa dodaju oznaku (engl. *label),* na osnovu koje bi se paketi usmeravali, umesto pomoću odredišne adrese. Kada se oznake u obliku indeksa unesu u tabele za usmeravanje, pronalaženje prave izlazne linije svodi se na jednostavno pretraživanje tabele. Usmeravanje se pomoću ove tehnike može obaviti veoma brzo, a duž putanje se mogu rezervisati svi neophodni resursi.

Naravno, označavanje tokova na ovaj način opasno se približava konceptu virtu- elnih kola. U mreži X.25, ATM mreži, mreži sa štafetnim prenosom okvira i u drugim mrežama koje koriste podmrežu virtuelnih kola, takođe se u svaki paket stavlja oznaka (npr. identifikator virtuelnog kola), zatim se ta oznaka traži u tabeli i paket usme- rava na osnovu odrednice iz nje. Uprkos tome što mnogi korisnici Interneta nikako ne mogu da prihvate mrežni rad sa uspostavljanjem direktne veze, izgleda da se taj koncept stalno vrača, ovoga puta sa ciljem da obezbedi brže usmeravanje i bolji kvalitet usluge. Međutim, postoje suštinske razlike u pogledu toga kako Internet konstruiše putanju i načina na koji to radi mreža sa uspostavljanjem direktne veze, tako da se pri- menjuje tehnika koja izvesno nije klasično komutiranje električnih kola.

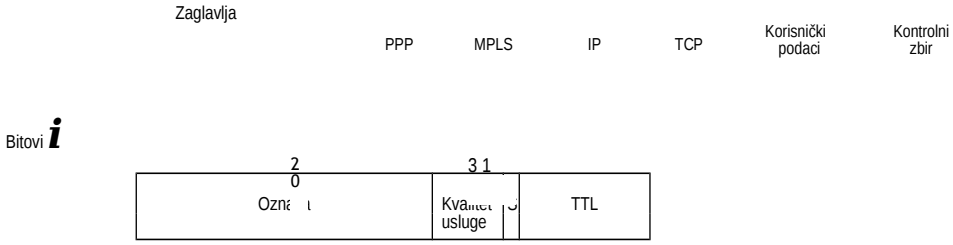
„Nov“ način komutiranja nosi različita komercijalna (engleska) imena, kao što su *label switching* ili *tag switching,* a i jedno i drugo znači komutiranje na osnovu oznaka. Najzad je IETF počeo da standardizuje postupak, pod imenom višeprotokolarno komutiranje oznaka (engl. *Multiprotocol Label Switching, MPLS).* Nadalje ćemo ga zvati kratko MPLS. Postupale je, između ostalog, opisan i u RFC dokumentu 3031.

Pomenimo uzgred da neki prave razliku između *usmeravanja* (engl. *routing*) i *komutiranja* (engl. *switching).* Usmeravanje je postupak traženja odredišne adrese u tabeli da bi se paket mogao poslati na nju. Nasuprot tome, kod komutiranja se kao indeks za pretraživanje tabele koristi oznaka uzeta iz paketa. Navedene definicije, međutim, daleko su od toga da budu opšte prihvaćene.

Prvo pitanje je gde smestiti oznaku. Pošto IP paketi nisu pravljeni za virtuelna kola, IP zaglavlje nema polje u koje bi se mogao smestiti broj virtuelnog kola. Zbog toga se ispred IP zaglavlja moralo dodati novo MPLS zaglavlje paketa. Na liniji između dva usmerivača, uz PPP kao protokol za uolcvirivanje, format okvira, zajedno sa PPP, MPLS, IP i TCP zaglavljem izgleda kao na slici 5-41. MPLS tako u izvesnom smislu predstavlja sloj 2,5.

Opšte MPLS zaglavlje ima četiri polja, od kojih je najznačajnije *Oznaka* - ono sadrži indeks. *Kvalitet usluge* ukazuje na klasu usluge. Polje *S* tiče se višestrukih oznaka u hijerarhijskim mrežama (ubrzo ćemo ga objasniti). Polje *TTL* označava *Životni vek* paketa. Ako njegova vrednost dostigne nulu, paket se odbacuje. Ta osobina onemogućava beskonačno kruženje paketa u slučaju nestabilnog usmeravanja.

Pošto MPLS zaglavlja nisu deo paketa mrežnog sloja, niti okvira sloja veze, MPLS je u velikoj meri nezavistan od oba sloja. To između ostalog znači da se mogu napraviti MPLS skretnice koje prosleđuju i IP pakete i ATM ćelije, zavisno od toga šta se pojavi. Zbog toga se MPLS komutiranje naziva „višeprotolcolarnim“.



Slika 5-41. Prenos TCP segmenta pomoću protokola IP, MPLS i PPP.

Kada paket (ili ćelija) obrađeni protokolom MPLS stignu u usmerivač koji zna za MPLS, on koristi oznaku paketa kao indeks za pretraživanje svoje tabele i određivanje linije na koju će paket uputiti, kao i za utvrđivanje nove oznake koju će mu dati. Ta zamena oznaka koristi se u podmrežama zasnovanim na virtuelnim kolima zato što oznake imaju samo lokalni značaj, pa dva različita usmerivača mogu trećem usmeri- vaču slati pakete istih oznaka iz dva različita toka, za prenos preko iste izlazne linije. Da bi ih druga strana mogla razlikovati, njihove oznake se moraju ponovo dodeljivati (preslikavati) pri svakom skoku. Rad tog mehanizma videli smo na slici 5-3. MPLS koristi istu tehniku.

U odnosu na klasična virtuelna kola, postoji razlika u pogledu nivoa grupisanja. Naravno da se može postići da svaki tok koristi sopstveni skup oznaka kroz celu podmrežu. Međutim, češće se viđa da usmerivači grupišu tokove koji se završavaju u određenom usmerivaču ili određenoj lokalnoj mreži i za njih koriste jedinstvenu oznaku. Tokovi, grupisani pod zajedničkom oznakom pripadaju tzv. **klasi ekvivalentnog prosleđivanja** (engl. *Forwarding Equivalence Class, FEC).* Ta klasa obuhvata ne samo odredište paketa, već i njihovu klasu usluga (kada su one diferencirane), pošto se svi njihovi paketi u smislu prosleđivanja tretiraju na isti način.

Pri klasičnom usmeravanju kroz virtuelna kola, nije moguće više različitih putanja s različitim završnim tačkama podvesti pod identifikator istog virtuelnog kola, jer se one ne bi mogle razlikovati na odredištu. MPLS paketi, međutim, osim oznake, zadržavaju i adresu odredišta, tako da se na kraju MPLS putanje oznaka može ukloniti, a paket svoj put produžiti na uobičajen način, koristeći odredišnu adresu mrežnog sloja.

Izgled tabele za usmeravanje prilično se razlikuje kod protokola MPLS i sistema zasnovanih na virtuelnim kolima. U klasičnim mrežama s virtuelnim kolima, kada korisnik poželi da uspostavi vezu, u mrežu se šalje paket za podešavanje (engl. *setup packet)* koji pravi putanju i u tabele unosi odgovarajuće odrednice. MPLS ne radi na taj način jer ne postoji faza podešavanja svake pojedinačne veze (to bi se sukobilo s previše mnogo postojećeg softvera za Internet).

Umesto toga, odrednice u tabelama prave se na jedan od sledeća dva načina. U **MPLS komutiranju vođenom podacima** (engl. *data-driven MPLS),* kada paket stigne u prvi usmerivač, on stupa u vezu sa usmerivačem na koji paket treba da ode i traži od njega da generiše oznaku za tok. Postupale se primenjuje rekurzivno i suštinski predstavlja uspostavljanje virtuelnog kola na zahtev.

Protokoli koji raspodeljuju tokove vrlo pažljivo izbegavaju petlje. U njima se često koristi tehnika tzv. obojenih niti (engl. *colored threads)*. Provlačenje FEC klase kroz podmrežu može se uporediti s provlačenjem obojenog konca. Kada usmerivač ugleda boju koju već ima, zna da postoji opasnost od stvaranja petlje i preduzima odgovarajuće mere. Podacima vođen protokol MPLS najčešće se koristi u mrežama u kojima se unutrašnji transport odvija u režimu asinkronog prenosa, ATM (kao što je veći deo sistema fiksne telefonije).

Na mrežama koje nisu zasnovane na ATM-u, prednost ima autonomno MPLS **komutiranje** (engl. *control-driven MPLS)*. Jedna od mnogih njegovih varijanti radi na sledeći način. Kada se usmerivač uključi, on traži putanje za koje je sam konačno odredište (npr. proverava koji se računari nalaze u njegovoj lokalnoj mreži). On tada za njih pravi jednu ili više klasa ekvivalentnog prosleđivanja (FEC), svakoj dodeljuje oznaku, i oznake prosleđuje svojim susedima. Oni oznake smeštaju u svoje tabele za usmeravanje i nove oznake šalju svojim susedima, sve dok svi usrnerivači ne usvoje označene putanje. Istovremeno se mogu rezervisati i resursi koji treba da obezbede odgovarajući kvalitet usluga.

MPLS može istovremeno da radi na više nivoa. Na najvišem nivou, svaki se nosilac može posmatrati kao svojevrstan metausmerivač, pri čemu se ostvaraje putanja od izvorišta, kroz metausmerivače, do odredišta. Tu putanju može da koristi MPLS. Međutim, MPLS se može koristiti nazavisno i unutar svake pojedinačne mreže, što predstavlja dragi nivo označavanja. U stvari, paket može da sa sobom nosi skup oznaka. Bit *S* na slici 5-41 omogućava usmerivaču koji uklanja oznaku da sazna ima li još preostalih oznaka. Njegova vrednost za osnovnu oznaku je 1, a za sve ostale oznake je 0. Ta mogućnost se u praksi najčešće koristi za ostvarivanje virtuelnih privatnih mreža i rekurzivnih tunela.

Iako je osnovna ideja višeprotokolarnog komutiranja oznaka prilično jednostavna, njeno realizovanje je izuzetno složeno. Postoje mnoge varijante i optimizacije u koje dalje nećemo zalaziti. Ako vas to ipak zanima, pogledajte sledeću literatura: Davie i Rekhter (2000), Lin i saradnici (2002), Pepelnjak i Guichard (2001) i Wang (2001).

5.5 KOMBINOVANJE RAZLIČITIH MREŽA

Do sada smo prećutno smatrali da postoji jedinstvena homogena mreža u kojoj svaki računar u svakom sloju koristi isti protokol. Nažalost, takva pretpostavka je beznadežno naivna. Postoje mnoge različite lokalne, gradske i regionalne mreže. U svakom sloju koristi se više različitih protokola. U narednim odeljcima detaljno ćemo razmotriti probleme koji nastaju kada se dve ili više mreža povežu u **kombinovanu mrežu** (engl. *internet)*.

Gotovo da nema slaganja po pitanju da lije obilje različitih vrsta mreža koje danas postoje privremenog karaktera i da li će ono ubrzo nestati kada svi shvate kako savršeno radi [upišite svoju omiljenu mrežu], ili je to obilje neizbežna i trajna pojava na koju moramo da se naviknemo. Različite mreže neizostavno traže i različite protokole.

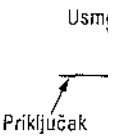
Mi verujemo da će oko nas uvek biti različitih mreža (pa i različitih protokola), iz sledečih razloga. Na prvom mestu, već je instalirano mnogo različitih mreža. Skoro svi PC računari koriste protokol TCP/IP. Mnoge velike kompanije koriste centralne računare, gde se koristi IBM-ov standard SNA. Priličan broj telefonskih kompanija radi sa ATM mrežama. U nekim lokalnim mrežama PC računara i dalje se koriste No - vellov protokol NCP/IPX ili AppleTalk. I na kraju, bežične mreže su novo područje koje se ubrzano razvija, s brojnim sopstvenim protokolima. Takav trend će se nastaviti i u budućnosti, gde će značajnu ulogu igrati nasleđeni sistemi, nove tehnologije i činjenica da mnogi proizvođači oklevaju da

svojim mušterijama omoguće lak prelazak na paletu proizvoda drugog proizvođača.

Zatim, kako računari i mreže postaju svakim danom jeftiniji, tako se i odlučivanje u organizacijama spušta na sve niži nivo. Mnoge kompanije neguju pravilo da nabavke npr. iznad milion dolara može da odobri samo Centralni upravni odbor, da su za nabavke između sto hiljada i milion dolara odgovorne Uprave pojedinih sektora, a da nabavke ispod sto hiljada dolara može da odobri i Upravnik odeljenja ne tražeći sa- glasnost viših struktura. Takva politika lako može dovesti do situacije da Inženjersko odeljenje sasvim legalno instalira radne stanice pod Unixom i s protokolom TCP/IP, a Marketing - Macintosh računare s protokolom AppleTalk.

I najzad, u različitim mrežama (npr. u ATM ili bežičnim mrežama) koriste se veoma različite tehnologije, pa ne bi trebalo da nas iznenadi ako se za nov hardver odmah izmisli i nov softver. Na primer, današnji prosečan porodični stan u razvijenom svetu, izgleda kao što je prosečna kancelarija izgledala pre desetak godina: prepun je računara koji međusobno ne komuniciraju. U budućnosti će možda biti normalno da telefon, televizor i drugi uređaji budu zajedno umreženi tako da se svima može upravljati iz daljine. Ta nova tehnologija nesumnjivo će doneti nove vrste mreža i nove protokole.

Na slici 5-42 prikazanje primer mogućeg povezivanja više mreža. Tu vidimo korporacijsku mrežu čiji su različiti delovi povezani regionalnom ATM mrežom. Na jednoj lokaciji se koristi optička FDDI okosnica za međusobno povezivanje jedne Ethernet mreže, jedne bežične lokalne mreže tipa 802.11, i SNA mreže centralnog računara korporacije.



Slika 5-42. Skup međusobno povezanih mreža.

Cilj povezivanja svih ovih mreža jeste da se omogući međusobno komuniciranje korisnika u ćelom povezanom sistemu, i da se svakom korisniku omogući pristup podacima smeštenim u bilo kojoj od povezanih mreža. Da bi se taj cilj ostvario, moraju se slati paketi iz jedne mreže u drugu. Pošto se mreže često međusobno značajno razlikuju, prebacivanje paketa iz mreže u mrežu ne ide uvele lako, u šta ćete se ubrzo i sami uveriti.

5.5.1 Razlike između mreža

Mreže se mogu razlikovati po mnogo čemu. Neke razlike - na primer, tehnika modulacije i format okvira - zadiru u fizički sloj i sloj veze podataka, i one nas ovde neće zanimati. Umesto toga, na slici 5-43 navodimo neke od razlika koje se mogu javiti u mrežnom sloju, i

zbog kojih je međumrežni rad teži od rada u jedinstvenoj mreži.

Kada paketi poslati sa izvorišta u jednoj mreži moraju da prođu kroz više drugih mreža da bi stigli do odredišne mreže (koja se i sama može razlikovati od izvorišne mreže), mnogo problema može nastati na interfejsima između mreža. Počnimo s tim da se paketi koji dolaze iz mreže kod koje se veza direktno uspostavlja moraju preurediti kada treba da prođu kroz mrežu u kojoj se veza ne uspostavlja direktno, što pošiljalac ne očekuje, a primalac nije spreman da preuzme na sebe. Paketi često treba da se prevode iz jednog protokola u drugi, a to je teško izvodljivo ako se funkcionalnost zahtevana u jednom protokolu ne može izraziti u drugom. Treba prevoditi i adrese, za šta je možda potreban sistem kataloga. Pri prolasku kroz mrežu koja ne podržava višesmerno slanje, za svaki paket poslat na više adresa moraju se generisati posebni paketi - po jedan za svako odredište.

Najveće teškoće može da stvori maksimalna veličina paketa koja je u različitim mrežama različito definisana. Kako ćete paket od 8000 bajtova poslati kroz mrežu u kojoj je najveći paket 1500 bajtova? I kvalitet usluga postaje problematičan kada paket sa ograničenjima koje postavlja isporuka u realnom vremenu treba da prosledite kroz mrežu koja za takvu isporuku ne daje nikakve garancije.

Stavka	**Neke razlike**
Ponuđena usluga	Sa uspostavljanjem direktne veze ili bez nje
Protokoli	IP, IPX, SNA, ATM, MPLS, AppleTalk itd.
Adresiranje	Prosto (802) ili hijerarhijsko (IP)
Višesmerno emitovanje	Postoji/ne postoji (važi i za neusmereno, tj. đifuzno emitovanje)
Veličina paketa	Svaka mreža ima sopstveni maksimum
Kvalitet usluge	Postoji/ne postoji; mnoge različite klasifikacije
Obrada grešaka	Pouzdana isporuka, isporuka redom i bez reda
Kontrola toka	Klizni prozori, kontrola brzine emitovanja, nešto treće ili ništa
Kontrola zagušenja	Bušna kofa, kofa sa žetonima, RED, prigušni paketi itd.
Bezbednost	Pravila privatnosti, šifrovanje itd.
Parametri	Različiti rokovi tajmera, specifikacije tokova itd.
Naplaćivanje	Prema trajanju veze, po paketu, po bajtu ili se ne obračunava

Slika 5-43. Neke od mnogih razlika između mreža.

Kontrola grešaka, toka i zagušenja često se različito sprovodi u različitim mrežama. Ako i izvorište i odredište očekuju da paketi budu isporučeni poslatim redosle- dom i bez grešaka, a neka mreža na putanji baca pakete čim nanjuši zagušenje, mnoge aplikacije to neće modi da prihvate. Isto tako, ako se paketima dozvoli da jedno vreme bespomoćno lutaju, a onda se odjednom pojave da bi bili isporučeni, nastade problemi ukoliko se na to unapred ne računa i ne postoji gotov način da se takvo ponašanje obradi. Probleme mogu da izazovu i različiti bezbednosni mehanizmi, vrednosti parametara, pravila naplaćivanja, čak i državni propisi o privatnosti.

5.5.2 Načini međusobnog povezivanja mreža

Mreže se međusobno mogu povezivati različitim uređajima, kao što smo videli u 4. poglavlju. Podsetimo se ukratko. U fizičkom sloju, mreže se mogu povezivati repetitorima ili razvodnicima koji samo prebacuju bitove s jedne na drugu identičnu mrežu. To su uglavnom analogni uređaji koji ne znaju ništa o digitalnim protokolima (oni samo regenerišu signale).

Stepenicu više - u sloju veze podataka - nalazimo mostove i skretnice. Oni mogu da prihvataju okvire, da ispituju MAC adrese i da prosleđuju okvire različitim mrežama obavljajući rudimentarno prevođenje između protokola, npr. iz Etherneta u FDDI ili802.il.'

U mrežnom sloju imamo usmerivače koji mogu da povežu dve mreže. Ako dve mreže imaju različite mrežne slojeve, usmerivač će možda moći da usaglasi formate paketa, iako se prevođenje paketa sve rede može sresti. Usmerivač koji razume više protokola naziva se **višeprotokolarni usmerivač** (engl. *multiprotocol router).*

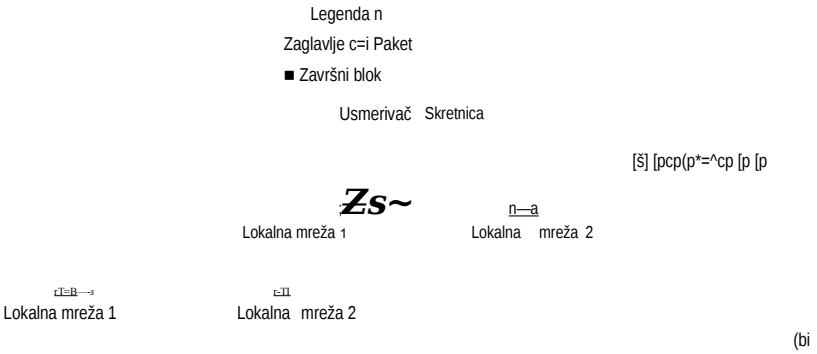
Na sledećem nivou nalazimo mrežne prolaze koji mogu da ostvare interfejs između dva priključka u transportnom sloju. Na primer, transportni mrežni prolaz omogućuje tok paketa između TCP mreže i SNA mreže koja ima drugačiji transportni protokol, tako što praktično slepljuje TCP i SNA priključke.

Najzad, u sloju aplikacija, mrežni prolazi za aplikacije usaglašavaju semantiku poruka. Na primer, mrežni prolazi za elektronsku poštu između Interneta (RFC 822) i sistema X.400 moraju da leksički urede poroke i da promene različita polja u njihovom zaglavlju.

U ovom poglavlju ćemo se ograničiti na međumrežni rad u mrežnom sloju. Da biste razumeli po čemu se on razlikuje od komutiranja u sloju veze podataka, pogledajte sliku 5-44. Na slici 5-44(a), izvorišni računar *S* želi da pošalje paket odredišnom računam *D*. Računari se nalaze na različitim lokalnim (Ethernet) mrežama, povezanim skretnicom. Računar *S* kapsulira paket u okvir i šalje ga. Okvir stiže u skretnicu koja zagleda njegovu MAC adresu i utvrđuje da ga treba poslati u lokalnu mrežu 2. Skretnica samo skida okvir s lokalne mreže 1 i postavlja ga u lokalnu mrežu 2.

Razmotrimo sada dve lokalne mreže povezane preko dva usmerivača. Usmerivači su međusobno povezani linijom od tačke do tačke, verovatno iznajmljenom i dugačkom hiljade kilometara. Ovde usmerivač uzima okvir i vadi paket iz njegovog polja za korisničke podatke. Usmerivač pronalazi adresu u paketu (npr. IP adresu) i traži je u svojoj tabeli za usmeravanje. Na osnovu te adrese on odlučuje da paket pošalje

udaljenom usmerivaču, možda kapsuliran u drugačiji okvir, u zavisnosti od protokola kojim radi linija. Paket se na dragom kraju smešta u polje za podatke Ethernet okvira i postavlja na lokalnu mrežu 2.



Slika 5-44. (a) Dve lokalne mreže povezane skretnicom, (b) Dve lokalne mreže povezane usmerivačima.

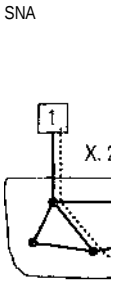
Osnovnu razliku između situacije sa skretnicom (ili mostom) i situacije sa usme- rivačem čini to što skretnica (most) šalje čitav okvir na osnovu njegove MAC adrese, a usmerivač vadi paket iz okvira i na osnovu adrese iz njega odlučuje kuda da ga pošalje. Skretnice ne moraju da razumeju protokol mrežnog sloja koji se koristi za komutiranje paketa. Usmerivači to moraju.

5.5.3 Nado vezana virtuelna kola

Moguća su dva stila povezivanja mreža: nadovezivanje (engl. *concatenation)* podmreža virtuelnih kola sa uspostavljanjem direktne veze, i datagramsko umrežavanje. Razmotricemo oba stila, ali najpre da na nešto upozorimo. U prošlosti je većina (javnih) mreža radila sa uspostavljanjem direktne veze (a još uvek tako rade štafetni prenos okvira, mreže SNA, 802.16 i ATM mreže). Kada se pojavio Internet, u modu su ušli datagrami. Ipak, pogrešno bi bilo misliti da će datagrami ostati zauvek. U računarstvu jedino možete biti sigurni da će se sve stalno menjati. Kako se širi prenos multimedije, izgleda da će se mreže sa direktnim vezama vratiti na ovaj ili onaj način jer je lakše garantovati kvalitet usluge kada veza postoji. Upravo zato, u nastavku ćemo određen prostor posvetiti radu u mrežama sa uspostavljanjem direktne veze.

U modelu nadovezanih virtuelnih kola, prikazanom na slici 5-45, veza s računarom na udaljenoj mreži uspostavlja se na približno klasičan način. Podmreža shvata daje odredište udaljeno i pravi virtuelno kolo ka usmerivaču koji je najbliži odredišnoj mreži. Zatim konstraiše virtuelno kolo od usmerivača do spoljnog mrežnog prolaza (engl. *gateway)* - višeprotokolamog usmerivača. Taj usmerivač beleži postojanje vir- tuelnog kola u svoje tabele i nastavlja da pravi drago virtuelno kolo do usmerivača na sledećoj podmreži. Opisani postupak se nastavlja sve do odredišnog računara.

Kada paketi krenu duž putanje, svaki mrežni prolaz ih sprovodi i istovremeno im po potrebi menja format i broj virtuelnog kola. Svi paketi s podacima moraju da prođu isti niz mrežnih prolaza, pa mreža nikada ne menja redosled paketa koji čine tok.



Princip ovog pristupa je uspostavljanje sekvence virtuelnih kola, počev od izvorišta, preko jednog ili više mrežnih prolaza, do odredišta. Svaki mrežni prolaz u svojim tabelama beleži koja virtuelna kola prolaze kroz njega, kuda ih treba usmeriti i pod kojim brojem.

Sistem radi najbolje kada sve mreže imaju približno ista svojstva. Na primer, ako sve garantuju pouzdanu isporuku paketa mrežnog sloja, tok od izvorišta do odredišta takođe će biti pouzdan, osim ako dođe do neke havarije na putanji. Isto tako, ako nijedna mreža ne garantuje pouzdanu isporuku, tada je ne garantuje ni nadovezivanje virtuelnih kola. S druge strane, ukoliko je izvorišni računar na mreži koja garantuje pouzdanu isporuku, ali neka od mreža na putanji može da gubi pakete, tada nadovezivanje u osnovi menja prirodu usluge.

Nadovezana virtuelna kola česta su i u transportnom sloju. Konkretno, može se napraviti kanal koji koristi protokol SNA i završava se u mrežnom prolazu, a zatim od tog do sledećeg mrežnog prolaza koristiti protokol TCP. Na taj način, može se izgraditi virtuelno kolo od jednog do drugog kraja koje prolazi kroz različite mreže i radi uz različite protokole.

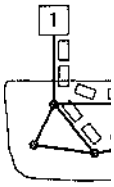
5.5.4 Međumrežni rad bez uspostavljanja direktne veze

Alternativni model međumrežnog rada je datagramski model (slika 5-46). Ovde mrežni sloj pomaže transportnom sloju samo tako što mu prosleđuje datagrame. U mrežnom sloju virtuelna kola se i ne pominju, a kamoli njihovo nadovezivanje. Prema modelu nije obavezno da svi paketi koji pripadaju istoj vezi prođu istu sekvencu mrežnih prolaza. Na slici 5-46 prikazano je da paketi od računara 1 do računara 2 putuju kroz različite mreže. Odluka o usmeravanju donosi se nezavisno za svald paket, često u zavisnosti od trenutnog saobraćaja na mreži. Uz takvu strategiju može se koristiti više putanja i na taj način ostvariti veći propusni opseg nego prema modelu na- dovezanih virtuelnih kola. S druge strane, ništa ne garantuje da će paketi na odredište stići redosledom kojim su poslati, ako uopšte stignu.

Model na slici 5-46 nije tako jednostavan kao što izgleda. Najpre, ako svaka mreža u mrežnom sloju ima drugačiji protokol, paket ne može da pređe s jedne mreže na drugu. Možda zamišljate da višeprotokolarni usmerivači bezgrešno prevode jedan format u drugi, ali ako predmetni formati nisu bliski i sa istim tipovima polja, takvo prevođenje će uvek biti nepotpuno i često neuspešno. Upravo zbog toga, prevođenje formata retko se primenjuje.

Dragi, ozbiljniji problem, tiče se adresiranja. Zamislite jednostavnu situaciju: raču- nar na Internetu pokušava da pošalje IP paket računani na pripojenoj SNA mreži. IP adresa se razlikuje od SNA adrese. Za prevođenje bi bilo potrebno preslikavanje između IP i SNA adresa u oba smera. Štaviše, u dve mreže se razlikuje i ono što može da ima adresu. U IP

protokolu, adrese imaju računari (zapravo, mrežne kartice). U protokolu SNA, adrese osim računara mogu da imaju i drugi hardverski uređaji. U najboljem slučaju, neko bi trebalo da održava bazu podataka u kojoj se sve preslikava jedno u drugo, ali bi ona bila nepresušan izvor problema.



Drugi pristup je da se napravi univerzalni paket za „međumrežni“ prenos koji će prepoznavati svi usmerivači. Takav je, u stvari, IP paket - napravljen da prolazi kroz mnoge mreže. Naravno, može se desiti da IPv4 (aktuelni protokol za Internet) potisne sve drage formate, da IPv6 (budući protokol za Internet) nikada ne zaživi i da niko više ne izmisli ništa novo, ali nas istorija uči drugačije. Teško je postići saglasnost oko jedinstvenog formata dok kompanije smatraju da je format ono čime mogu da utiču na svoju zaradu.

Sumirajmo ukratko dva opisana načina međumrežnog rada. Nadovezana virtuelna kola nude iste prednosti kao i virtuelna kola u jedinstvenoj podmreži: baferi se mogu unapred rezervisati, može se garantovati isporuka paketa redosledom kojim su poslati, mogu se koristiti kratka zaglavlja i mogu se izbeći teškoće sa zakasnelim duplikatima paketa.

Ona imaju i svoje mane: potreban prostor u tabelama usmerivača za svaku tekuću vezu, nemogućnost alternativnog usmeravanja u slučaju zagušenja, i osetljivost na otkazivanje usmerivača duž putanje. Osim toga, teško je - ako je uopšte i moguće - ugraditi ovakvu šemu ukoliko je jedna od mreža nepouzdana datagramska mreža.

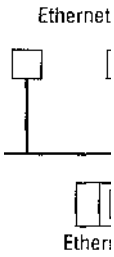
Datagramski međumrežni rad odvija se na sličan način kao u jedinstvenoj data- gramskoj podmreži: postoji veća mogućnost zagušenja, ali i šire mogućnosti da mu se rad prilagodi, robusnost u pogledu otkazivanja usmerivača i duža zaglavlja paketa. Mogu se primeniti različiti prilagodljivi algoritmi za usmeravanje, kao i u jedinstvenoj datagramskoj mreži.

Osnovna prednost međumrežnog rada s datagramima jeste to što se datagrami mogu primeniti na podmreže koje interno ne koriste virtuelna kola. Mnoge lokalne mreže, pokretne mreže (npr. vazduhoplovne i pomorske flote), čak i neke regionalne mreže, spadaju u ovu kategoriju. Kada kombinovana mreža obuhvata jednu od takvih mreža, nastaju ozbiljni problemi ako se u međumrežnom radu oslonimo na virtuelna kola.

5.5.5 Upotreba tunela

Traženje opšteg rešenja za povezivanje dve različite mreže u jedinstvenu, kombi- novanu mrežu izuzetno je teško. Međutim, postoji uobičajen specijalan slučaj koji je jednostavniji. To je slučaj u kome su i izvorište i odredište na mrežama istog tipa, dok se između njih nalazi drugačija mreža. Zamislite, na primer, neku međunarodnu banku s filijalama u Parizu i Londonu, obe sa lokalnim Ethernet mrežama i protokolom TCP/IP, dok se između njih pruža

regionalna mreža (npr. ATM mreža), koja ne radi po protokolu IP (slika 5-47).



Slika 5-47. Slanje paketa tunelom od Pariza do Londona.

Rešenje opisanog problema postiže se tzv. upotrebom tunela (engl. *tunneling)*. Da bi računaru 2 poslao IP paket, računar 1 konstruiše paket koji sadrži IP adresu računara 2, unosi ga u Ethernet okvir adresiran na pariški višeprotokolarni usmerivač
1 postavlja ga na Ethernet. Kada višeprotokolarni usmerivač dobije okvir, on iz okvira vadi IP paket, stavlja ga u polje za korisničke podatke paketa mrežnog sloja regionalne mreže koji adresira na regionalnu adresu londonskog višeprotokolarnog usmerivača. Kada paket tamo stigne, londonski usmerivač vadi IP paket i šalje ga računaru
2 unutar Ethernet okvira.

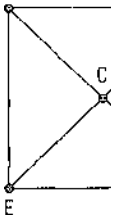
Regionalnu mrežu možemo smatrati dugačkim tunelom koji spaja dva višeprotoko- lama usmerivača. IP paket samo putuje od jednog do drugog njegovog kraja baš kao turista - ne brinući o prevozu koji obavlja regionalna mreža. O tome ne brinu ni računari u lokalnim mrežama. Jedino višeprotokolami usmerivači moraju da razumeju i IP i regionalne pakete. U stvari, sav prenos od središta jednog višeprotokolarnog usmerivača do središta dragog usmerivača odvija se kao daje u pitanju prenos serijskom linijom.

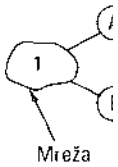
Efekat tunela postade jasniji ako upotrebimo jednu analogiju. Zamislite osobu koja vozi kola u smeru od Pariza do Londona. Dok je u Francuskoj, automobil se kreće sopstvenim pogonom, ali kad stigne do Lamanša, tovari se na specijalan voz i prevozi u Englesku kroz Chunnel (igra reči koja kombinuje kanal i tunel, engl. *channel* + *tun- ne!.* Prim. prev.) jer se automobilima ne dozvoljava da voze kroz njega. Kola se, u stvari, prevoze kao običan teret (slika 5-48). U Engleskoj, automobil se skida s voza i nastavlja put sopstvenim pogonom. Sprovođenje paketa tunelima kroz drugačije mreže radi na isti način.

Slika 5-48. Prenošenje automobila tunelom između Francuske i Engleske.

5.5.6 Usmeravanje kroz kombinovanu mrežu

Usmeravanje kroz kombinovanu mrežu obavlja se na sličan način kao i usmeravanje kroz jedinstvenu mrežu, ali uz dodatne komplikacije. Razmotrite, na primer, kombinovanu mrežu na slici 5-49(a) koja je sastavljena od pet mreža povezanih sa šest (možda, višeprotokolarnih) usmerivača. Pravljenje grafa za ovu situaciju dodatno komplikuje činjenica da svaki usmerivač može direktno da pristupi (tj. pošalje paket) svakom drugom usmerivaču koji je povezan za mrežu s kojom je on spojen. Na primer, usmerivač *B* na slici 5-49(a) može da direktno pristupi usmerivačima A i C preko mreže 2, a takođe i usmerivaču *D* preko mreže 3. To vodi grafu prikazanom na slici 5-49(b).





Slika 5-49. (a) Kombinovana mreža, (b) Graf kombinovane mreže.

Pošto se konstruiše graf, na skup višeprotokolarnih usmerivača može se primeniti neki poznat algoritam za usmeravanje, npr. algoritam zasnovan na vektoru razdaljine ili na stanju veze. Time dobijamo algoritam za usmeravanje na dva nivoa: unutar svake mreže koristi se **unutrašnji protokol za mrežni prolaz** (engl. *interior gateway protocol),* a između mreža **spoljni protokol za mrežni prolaz** (engl. *exterior gateway protocol*). („Mrežni prolaz" je stari termin za „usmerivač".) U stvari, pošto su mreže nezavisne, u svakoj može da se koristi drugi algoritam. Pošto svaka mreža unutar kombinovane mreže radi nezavisno od drugih, ona se često naziva **autonoman sistem** (engl. *Autonomous System*, *AS).*

Tipičan međumrežni paket kreče iz svoje lokalne mreže adresiran (u zaglavlju MAC sloja) na lokalni višeprotokolarni usmerivač. Kada stigne tamo, mrežni sloj pregleda svoje tabele za usmeravanje i odlučuje o višeprotokolarnom usmerivaču na koji će ga uputiti. Ako se taj usmerivač može doseći pomoću protokola mreže iz koje je paket potekao, paket se tamo šalje direktno. U suprotnom, koristi se tunel, a paket se kapsulira u format protokola koji se koristi na konkretnoj mreži. Opisani postupale se ponavlja sve dok paket ne stigne u odredišnu mrežu.

Usmeravanje kroz kombinovanu mrežu razlikuje se i po tome što kombinovana mreža može da prelazi državne granice. Tu odmah stupaju na scenu različiti propisi, kao što je švedski zakon o zaštiti privatnosti koji strogo ograničava „izvoženje" ličnih podataka Šveđana izvan Švedske. Drugi primer je kanadski zakon koji propisuje da saobraćaj podataka začet u Kanadi i okončan u Kanadi ne srne da napusti kanadsku teritoriju. To znači da saobraćaj iz Vindzora (Ontario) za Vankuver ne sme da se us- meri preko obližnjeg Detroita (SAD), iako je ta putanja brža i jeftinija.

Dragu razliku između unutrašnjeg i spoljnog usmeravanja čine troškovi. Unutar jedinstvene mreže obično se primenjuje jedan algoritam obračunavanja troškova. Međutim, različite mreže imaju i različite vlasnike, pa se cene putanja i zbog toga mogu razlikovati. Slično tome, kvalitet usluga može da se razlikuje od jedne mreže do druge, pa i to može da bude razlog da se izabere (ili ne izabere) određena putanja.

5.5.7 Fragmentiranje

Svaka mreža na odeđeni način ograničava veličinu paketa. Za to postoji više razloga, a neki su:

1. Hardver (npr. veličina Ethernet okvira).
2. Operativni sistem (npr. svi baferi su veličine 512 bajtova).
3. Protokoli (npr. broj bitova u polju *Dužina).*
4. Usaglašavanje s nekim nacionalnim ili međunarodnim standardom.
5. Zelja da se do određene mere smanji ponovno slanje paketa uzrokovano greškama u prenosu.
6. Zelja da se spreči predugo zauzeće kanala jednim paketom.

Iz pobrojanih razloga, projektanti mreža ne mogu da po želji biraju dužinu paketa. Maksimalan koristan teret paketa kreće se od 48 bajtova (ATM ćelije) do 65.515 bajtova (IP paketi), premda je koristan teret u višim slojevima često veći.

Očigledan problem nastaje kada veliki paket treba da prođe mrežom koja dozvoljava znatno manju veličinu paketa. Najbolje je preduzeti sve da do nečega takvog uopšte i ne dođe. Drugim recima, kombinovana mreža treba da koristi algoritam za usmeravanje koji neče slati pakete kroz mreže koje ne mogu da ih obrade. Međutim, takvo rešenje u stvari i nije rešenje. Šta raditi ako je izvorišni paket prevelik i za odredišnu mrežu? Algoritam za

usmeravanje ne može da zanemari odredište.

U osnovi, jedino rešenje je da mrežni prolazi izdele pakete na manje fragmente i da svaki fragment pošalju kao zaseban međumrežni paket. Međutim, kao što svako zna, mnogo je lakše kinesku vazu „usitniti“, nego je od fragmenata ponovo sastaviti. (Fizičari su ovom efektu čak dali ime: drugi zakon termodinamike.) I mreže koje rade s komutiranjem paketa imaju problema da ih ponovo sastave od fragmenata.

Slika 5-50. (a) Nevidljivo fragmentiranje. (b) Vidljivo fragmentiranje.

Postoje dve suprotne strategije za ponovno sastavljanje paketa iz fragmenata. Prema prvoj, fragmentiranje koje obavlja mreža sa ograničenom veličinom paketa nevidljivo je za sve naredne mreže kroz koje paket mora da prođe na putu do odredišta. Ona je prikazana na slici 5-50(a). Mreža koja previše ograničava veličinu paketa ima mrežne prolaze (najverovatnije, specijalizovane usmerivače) pomoću kojih se povezuje s drugim mrežama. Kada na mrežni prolaz stigne prevelik paket, mrežni prolaz ga deli u fragmente. Svaki fragment se adresira na isti izlazni mrežni prolaz, gde se paket ponovo sastavlja. Na taj način, rad mreže koja ograničava veličinu paketa ostao je nevidljiv. Naredne mreže ni ne znaju daje paket prethodno bio fragmentiran. ATM mreže, na primer, imaju specijalan hardver koji neprimetno fragmentira pakete u ćelije i od ćelija ponovo sklapa pakete. U ATM svetu, fragmentiranje se zove seg- mentiranje; osnovna ideja je ista, ali se detalji razlikuju.

Nevidljivo fragmentiranje je jednostavno, ali nije bez mana. Najpre, izlazni mrežni prolaz mora da zna kada je primio sve fragmente jednog paketa, pa se mora pred- videti polje s brojačem fragmenata ili identifilcator „kraj paketa“. Zatim, svi paketi moraju da iziđu kroz isti mrežni prolaz. Ako fragmentima ne dozvolite da biraju

najbolju putanju, performanse će u izvesnoj meri oslabiti. Poslednji problem je dodatno opterećenje izazvano stalnim fragmentiranjem i sklapanjem paketa pri prolasku kroz niz mreža koje ograničavaju veličinu paketa. ATM mreže zahtevaju nevidljivo fragmentiranje.

Prema drugoj strategiji, paketi se ne sklapaju od fragmenata na usputnim mrežnim prolazima. Kada se paket fragmentira, sa svakim fragmentom se radi kao sa zasebnim prvobitnim paketom. Svi fragmenti se šalju kroz jedan ili više izlaznih mrežnih prolaza, kao na slici 5-50(b). Tek odredišni računar ponovo sklapa pakete. IP radi na ovaj način.

I vidljivo fragmentiranje ima svojih mana. Na primer, za njega je potrebno da *svaki* računar zna da sastavi pakete iz fragmenata. Pri fragmentiranju većih paketa svaki fragment dobija zaglavlje, tako da to predstavlja dodatno opterećenje. Dok pri nevidljivom fragmentiranju to zaglavlje nestaje čim paket napusti „nezgodnu“ mrežu, ovde ono ostaje do kraja puta. Prednost vidljivog fragmentiranja je, međutim, to što pojedini fragmenti mogu birati najbolje putanje (izlazne mrežne prolaze), pa se performanse poboljšavaju. Naravno, ako se koriste nadovezana virtuelna kola, navedena prednost se ne ispoljava.

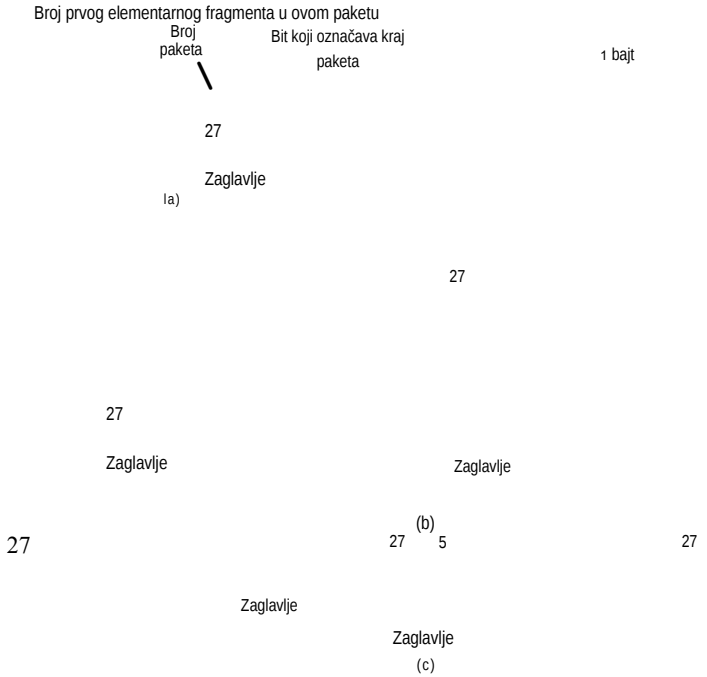
Kada se paket izdeli u fragmente, fragmenti se moraju numerisati tako da se iz njih može rekonstruisati prvobitni tok podataka. Numerisanje se može izvesti pomoću binarnog stabla. Ako se mora izdeliti paket 0, njegovi delovi ce biti 0.0, 0.1, 0.2 itd. Ukoliko se kasnije i ovi fragmenti moraju dalje deliti, njihovi delovi će biti 0.0.0,
0. 0.1, 0.0.2,..., 0.1.0, 0.1.1, 0.1.2 itd. Ukoliko se u zaglavlju rezerviše dovoljno mesta za najnepovoljniji slučaj i nigde ne prave duplikati, ova šema je dovoljna za rekon- struisanje paketa na odredištu, bez obzira na redosled kojim stižu fragmenti.

Međutim, ako jedna jedina mreža u nizu gubi ili odbacuje pakete, pakete je neophodno ponovo slati s jednog kraja na drugi, što se nepovoljno odražava na numerisanje. Pretpostavimo da je paket od 1024 bita na početku izdeljen u četiri jednaka fragmenta: 0.0, 0.1, 0.2 i 0.3. Fragment 0.1 se gubi, ali ostali stižu na odredište. Taj- meru na izvorištu ističe rok i on ponovo šalje paket. Ovog puta ga pogađa Marfijev zakon i izabrana putanja vodi kroz mrežu sa ograničenjem veličine paketa na .512 bitova, tako da se paket deli u dva fragmenta. Kada nov fragment 0.1 stigne na odredište, primalac ce pomisliti da sada ima sva četiri dela paketa i paket će sklopiti - neispravno.

Sasvim drugačiji (i bolji) sistem numerisanja koristi se u međumrežnom protokolu za definisanje elementarnog fragmenta, dovoljno malog da može da prođe kroz svaku mrežu. Kada se paket deli na fragmente, svi delovi su veličine elementarnog fragmenta osim poslednjeg, koji može da bude kraci. Paket koji se prenosi kroz kombinovanu mrežu može da zbog efikasnosti bude sastavljen od više fragmenata. Zaglavlje takvog paketa mora da sadrži broj originalnog paketa i broj (prvog) elementrnog fragmenta u paketu. Kao i obično, teba da postoji i bit koji će ukazivati da je poslednji elementarni fragment unutar paketa koji putuje kombinovanom mrežom takođe poslednji fragment originalnog paketa.

Ovakav postupak zahteva dva polja s rednim brojem u zaglavlju paketa koji putuje kombinovanom mrežom: broj originalnog paketa i broj njegovog fragmenta. Postoji jasan kompromis između veličine elementarnog fragmenta i broja bitova upotreblje- nih za redni broj fragmenta. Pošto se veličina elementarnog fragmenta bira tako da

zadovolji svaku mrežu, naknadno deljenje međumrežnog paketa od više fragmenata ne stvara teškoće. Donja granica veličine elementarnog fragmenta iznosi jedan bit ili jedan bajt, pri čemu broj fragmenta predstavlja priraštaj (bitova ili jedan bajt) u odnosu na redosled u originalnom paketu (slika 5-51).



Slika 5-51. Fragmentiranje kada elementarna veličina podataka iznosi **1** bajt. (a) Originalni paket sa 10 bajtova podataka, (b) Fragmenti posle prolaska kroz mrežu s maksimalnom veličinom paketa od 8 bajtova podataka i zaglavljem, (c) Fragmenti posle prolaska kroz mrežni prolaz koji u paketu dozvoljava 5 bajtova podataka.

U nekim međumiežnim protokolima, ova ideja se razvija do kraja i čitav prenos virtuelnim kolom posmatra se kao jedan džinovski paket, tako da svaki fragment sadrži apsolutni redni broj prvog bajta podataka u fragmentu.

5.6 MREŽNI SLOJ NA INTERNETU

Pre nego što zađemo u detalje mrežnog sloja Interneta, vredi da opišemo principe zbog kojih je napravljen u prošlosti, i zbog kojih je i danas tako uspešan. Izgleda da se danas preko njih previše lako prelazi. Ti principi su navedeni i obrazloženi u RFC dokumentu 1958 koji bi trebalo svako da pročita (a naročito programeri zaduženi za protokole - njih bi posle čitanja trebalo ispitati da bi se videlo šta su zapamtili). Navedeni RFC dokument široko se oslanja na ideje Clarlca (1988) i Saltzera i saradnika (1984). U nastavku ćemo navesti 10 principa koje smatramo ključnim (počinjući s najvažnijima).

1. To što **pravite** mora da radi. Ne zatvarajte fasciklu projekta ili standarda sve dok više puta niste eksperimentalno utvrdili da prototipovi mogu da međusobno komuniciraju. Programeri često prvo napisu standard na 1000 stranica, dobiju za

njega zeleno svetio, a zatim otkriju suštinske greške zbog kojih standard ne radi. Posle toga, oni napišu njegovu verziju 1.1. Tako ne treba raditi.

2. **Neka sve bude što jednostavnije.** Kada ste u nedoumici, izaberite najjednostavnije rešenje. Viljem Okam je još u 14. veku formulisao taj princip (Oka- mova oštrica). Rečeno savremenim jezikom: izbegavajte nepotrebne ukrase. Ukoliko nešto nije apsolutno neophodno, ne uvodite ga, naročito ako se isti rezultat može postidi kombinovanjem drugih mogućnosti.
3. **Ponudite jasan izbor.** Ako se isto može uraditi na više načina, izaberite samo jedan način. Više puteva do istog cilja samo stvaraju probleme. U standardima se često pojavljuje više opcija, režima ili parametara samo zato što nekoliko zainteresovanih strana smatraju daje njihovo rešenje najbolje. Programeri treba da se odlučno odupru takvom iskušenju. Jednostavno recite: ne.
4. **Insistirajte na modularnosti.** Ovaj princip direktno vodi konceptu skupa međusobno nezavisnih protokola. Na taj način, kada je neophodno promeniti jedan modul ili sloj, ostali ostaju netaknuti.
5. **Očekujte heterogenost.** Svaka velika mreža obuhvata različit hardver, uređaje za prenos podataka i aplikacije. Zbog toga projekat mreže mora da bude jednostavan, dovoljno uopšten i prilagodljiv.
6. **Izbegavajte statične opcije i parametre.** Ako parametre (npr. maksimalnu veličinu paketa) ne možete da izbegnete, najbolje je da ne definišete fiksne vrednosti, već da pošiljaocu i primaocu prepustite da se o tome dogovore.
7. **Neka projekat bude samo dovoljno dobar - ne mora da bude savršen.** Cesto programeri naprave dobar projekat mreže, ali se ispostavi da u njemu probleme pravi neki specijalan slučaj. Umesto da zbog toga menjaju projekat, programeri bi trebalo da ga realizuju, a da rešavanje tog specijalnog slučaja prepuste onima koji imaju specijalne zahteve.
8. **Poštujte pravila kada šaljete pakete, a gledajte kroz prste kada ih primate.** Šaljite samo pakete koji striktno odgovaraju standardu, ali očekujte da vam stignu nestandardni paketi i budite spremni da ih obradite.
9. **Mislite na moguću veličinu mreže.** Ako sistem treba efikasno da opsluži mi- lione računara i milijarde korisnika, nezamislive su centralizovane baze podataka i opterećenje se mora što ravnomemije raspodeliti na raspoložive resurse.
10. **Mislite na performanse i cenu.** Ako mreža ima loše performanse ili pravi izuzetno visoke troškove, niko je neće koristiti.

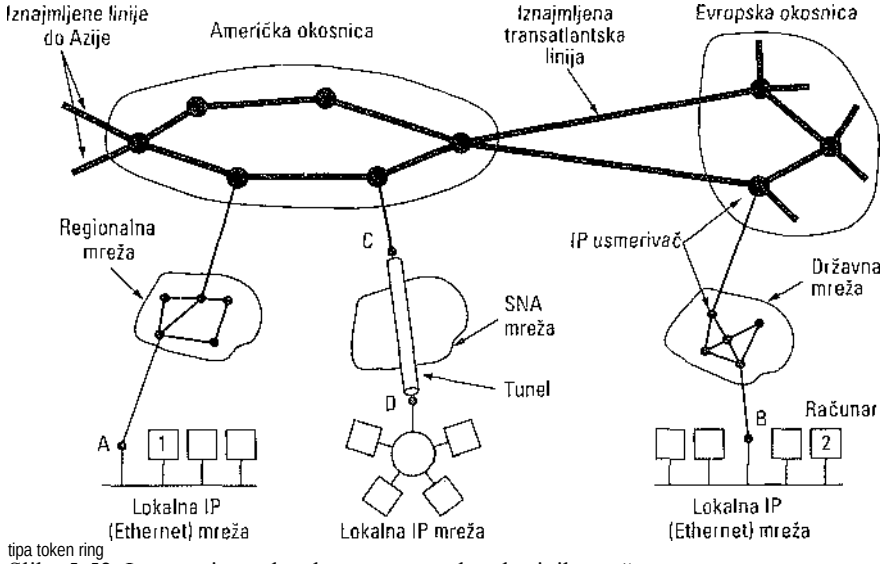
Ostavimo sada opšte principe i pređimo na detalje mrežnog sloja Interneta. Internet se u mrežnom sloju može posmatrati kao skup međusobno povezanih podmreža ili **autonomnih sistema** (engl. *Autonomous System, AS).* Kombinovana mreža nema određenu strukturu, osim većeg broja velikih okosnica koje se sastoje od linija velike propusne moći i brzih usmerivača. Na okosnice su priključene regionalne mreže (srednjeg nivoa), a za njih se vezuju lokalne mreže univerziteta, kompanija i davalaca Internet usluga. Ova pseudohijerarhijska organizacija šematski je prikazana na slici 5-52.

Čitav Internet na okupu drži protokol mrežnog sloja, tzv. **protokol IP** (engl. *Internet Protocol, IP).* Za razliku od većine starijih mrežnih protokola, protokol IP je od početka projektovan za međumrežni rad. O mrežnom sloju teba da razmišljate na sledeci način. Njegov zadatak je da na najbolji način (dakle, ne garantovano) obezbe- di prenos datagrama od izvorišta do odredišta, bez obzira na to da li se računari nalaze na istoj mreži ili se i druge mreže nalaze između njih.

Na Internetu se komunicira na sledeci način. Transportni sloj preuzima tokove podataka i

deli ih u datagrame. Datagrami teorijski mogu biti veličine 64 KB, ali im u praksi veličina ne prelazi 1500 bajtova (tako da staju u Ethernet okvir). Svaki data- gram se prenosi Internetom i možda usput deli na manje fragmente. Kada svi delovi datagrama konačno stignu na odredišni računar, mrežni sloj od njih sklapa originalni datagram. Taj datagram se zatim predaje transportnom sloju koji ga umeće u ulazni tok procesa za prihvatanje podataka kod primaoca. Kao što vidite sa slike 5-52, paket koji svoj put počinje na računam 1 mora da prođe šest mreža da bi stigao do računara

2. U praksi često mora da prođe mnogo više od šest mreža.



Slika 5-52. Internet je međusobno povezan skup brojnih mreža.

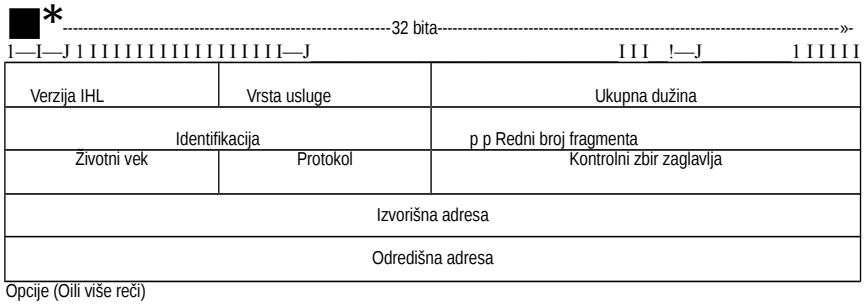
5.6.1 Protokol IP

Proučavanje mrežnog sloja Interneta najpogodnije ćemo početi analizom formata samih IP datagrama. IP datagram sadrži zaglavlje i tekstualni deo. Zaglavlje ima fiksni deo dužine 20 bajtova i opcioni deo promenljive dužine. Format zaglavlja prikazan je na slici 5-53. Ono se prenosi redosledom „big-endian“ - sleva udesno, pri čemu prvo ide najznačajniji bit polja *Verzija.* (Sistem SPARC je „big-endian“; Pentium se čita s drugog kraja, on je „little-endian“.) Na „little-endian“ računarima potrebno je softversko prevođenje i pri slanju i pri primanju.

U polju *Verzija* označena je verzija protokola kome pripada datagam. Kada verziju uključite u datagrame, neće morati cela mreža da odjednom pređe na novu verziju protokola - zamena se može obavljati godinama. Danas je aktuelan prelazak s protokola IPv4 na protokol IPv6, a on traje već više godina i nema znakova da će uskoro biti dovršen (Durand, 2001; Wiljakka, 2002; Waddington i Chang, 2002). Neki čak misle da se on nikada neće dovršiti (Weiser, 2001). Što se tiče brojeva verzija, pome- nimo uzgred daje postojao protokol za prenos u realnom vremenu - IPv5, koji nikada nije bio šire prihvaćen.

Pošto dužina zaglavlja nije fiksna, postoji u njemu polje *DIZ* (engl. *IHL, Internet header length)* u kome se beleži dužina zaglavlja u 32-bitnim recima. Najmanja vred- nost je 5, što

se koristi kada nema opcija. Najveća vrednost ovog 4-bitnog polja je 15, što ograničava zaglavlje na 60 bajtova, a polje *Opcije* na 40 bajtova. Za neke opcije, npr. za beleženje putanje paketa, 40 bajtova je premalo, pa takva opcija ne služi ničemu.



Slika 5-53. Zaglavlje IPv4 paketa (Internet protokol).

Vrsta usluge je jedno od retkih polja koje je tokom godina (neznatno) promenilo svoju ulogu. Ono je od početka bilo namenjeno razgraničavan ju različitih klasa usluga, a tu ulogu ima i sada. U tom pogledu, moguće su različite kombinacije pouzdanosti i brzine isporuke. Za digitalizovan govor, brza isporuka ima prednost nad tačnom isporukom. Za prenos datoteka, prenos bez grešaka je važniji od brzog prenosa.

Ovo 6-bitno polje prvobitno je sadržavalo (sleva udesno) 3-bitno potpolje *Prioritet* i tri indikatora *D, T* i *R.* Prioritet je označavan vednostima od 0 (normalan) do 7 (mrežni upravljački paket). Tri indikatorska bita su računaru omogućavala da naznači o čemu najviše brine (o kašnjenju - engl. Delay, protoku podataka - engl. Throughput ili o pouzdanosti prenosa - engl. Reliability). Ta polja su teorijski omogućavala usmerivačima da biraju između, na primer, satelitske veze s visokim protokom podataka i velikim kašnjenjem, i iznajmljene linije s niskim protokom podataka i malim kašnjenjem. U praksi, međutim, usmerivači često potpuno zanemaruju polje *Vrsta usluge.*

Na kraju je IETF popustio i neznatno izmenio polje da bi ga prilagodio diferenciranim uslugama. Šest bitova je iskorišćeno za već opisano svrstavanje paketa prema klasi usluga. To su četiri klase prioriteta za svrstavanje u redove čekanja, tri mogućnosti za odbacivanje paketa i nasleđene klase.

Polje *Ukupna dužina* obuhvata sve što se nalazi u datagramu - i zaglavlje i podatke. Maksimalna dužina je 65.535 bajtova. Ta gornja granica je zasad prihvatljiva, ali u budućim gigabitnim mrežama može se pojaviti potreba za većim datagramima.

Iz polja *Identifikacija* odredišni računar utvrđuje kom datagramu pripada pristigli fragment. Svi fragmenti istog datagrama imaju istu *Identifikaciju.*

Posle ovog polja sledi jedan neiskorišćen bit, a zatim dva jednobitna polja. *NF* znači *Ne* Fragmentiraj. To je naredba usmerivačima da ne fragmentiraju datagram jer odredište ne može da od njih ponovo sklopi datagram. Na primer, kada se računar podiže, njegov ROM može zahtevati da mu se memorijska slika pošalje u obliku jedinstvenog datagrama. Kada datagram označi bitom *NF,* pošiljalac zna da će on stići u jednom komadu, pa makar morao da odstupi od optimalne putanje da bi izbegao mreže koje ograničavaju veličinu paketa. Od svih računara se zahteva da prihvate fragmente veličine 576 bajtova ili manje fragmente.

JF znači Još Fragmenata. Svi fragmenti datagrama, osim poslednjeg, imaju postavljen taj bit. On je potreban da bi se znalo kada je datagram kompletiran.

Redni broj fragmenta pokazuje gde spada fragment unutar datagrama. Svi fragmenti datagrama, osim poslednjeg, moraju biti umnošci od 8 bajtova - veličine elementarnog fragmenta. Pošto je polje dužine 13 bitova, dozvoljeno je najviše 8192 fragmenta po datagramu, što daje datagram maksimalne dužine 65.536 bajtova, za bajt više no što dozvoljava polje *Ukupna dužina.*

Životni vek je brojač koji ograničava trajanje paketa na mreži. Predviđen je da vre- me meri sekundama, pa je najveći životni vek paketa 255 sekundi. Brojač mora smanjivati vrednost za jedinicu pri svakom skoku, a trebalo bi daje smanjuje i ako se duže zadrži u redu čekanja usmerivača. U praksi, međutim, on samo broji skokove. Kada njegova vrednost dostigne nulu, paket se odbacuje, a izvorišnom računani šalje se paket upozorenja. Ovo polje onemogućava datagrame da večno lutaju mrežom, što bi se moglo dogoditi ako se poremete tabele usmerivača.

Kada mrežni sloj sklopi potpun datagram, on treba da zna šta s njim da radi. Polje *Protokol* naznačava proces kome paket treba predati. Taj proces može biti protokol TCP, ali i protokol UDP ili drugi procesi. Protokoli se na Internetu globalno označavaju brojevima. Ranije su se brojevi protokola i drugi brojevi koji se dodeljuju nalazili u RFC dokumentu 1700, ali se sada nalaze u mrežnoj bazi podataka, na adresi www.iana.org.

Kontrolnim zbirom zaglavlja proverava se, prirodno, samo zaglavlje. Takav kontrolni zbir se koristi za provem grešaka izazvanih neispravnim memorijskim recima usmerivača. Algoritam radi tako što se aritmetikom nepotpunih komplemenata sabi- raju sve 16-bitne polureči onako kako pristižu, a zatim se od rezultata odbije nepotpuni komplement. Za svrhe ovog algoritma, pretpostavlja se da je *Kontrolni zbir zaglavlja* po stizanju jednak nuli (tj. ako nije došlo do greške.“). Opisani algoritam je robusniji od običnog sabiranja. Imajte na umu da se *Kontrolni zbir zaglavlja* mora ponovo izračunavati pri svakom skoku, pošto se barem jedno polje uvek menja (*Životni vek),* ali postoje načini da se to izračunavanje ubrza.

Izvorišna adresa i *Odredišna adresa* ukazuju na broj mreže i broj računara. Adresama na Internetu posvetićemo sledeći odeljak. Polje *Opcije* prihvata informacije koje nose novije verzije protokola (za koje se nije znalo u trenutku projektovanja formata), omogućava eksperimentatorima da isprobaju nove ideje i obezbeđuje mesto za informacije koje se retko koriste. Same opcije mogu biti različite dužine. Svaka počinje 1-bajtnim kodom koji identifikuje opciju. Kod nekih opcija zatim sledi 1-bajtno polje s dužinom opcije, a zatim kod svih sledi jedan ili više bajtova podataka. Polje *Opcije* dopunjava se nulama do umnoška od četiri bajta. Prvobitno je definisano pet opcija (slika 5-54), ali ih je kasnije dodato još. Potpunu ažuriranu listu opcija naći ćete na Internet adresi www.iana.org/assignments/ip-parameters.

Opcija	Opis
Bezbednost (Security)	Označava stepen tajnosti datagrama
Strogo usmeravanje sa izvora (Strict source routing)	Daje se potpuna putanja za prosleđivanje paketa
Približno usmeravanje sa izvora (Loose source routing)	Daje se lista obaveznih usmerivača
Beleženje putanje (Record route)	Svaki usmerivač treba da fragmentu doda svoju IP adresu
Vremenska oznaka (Timestamp)	Svaki usmerivač treba da fragmentu doda svoju IP adresu i vreme prolaska

Slika 5-54. Neke IP opcije.

Bezbednost naznačava tajnost informacija. Ovo polje teorijski može da upotrebi vojska da bi iz putanje isključila države koje smatra nepodobnim. U praksi ga, međutim, svi usmerivači zanemaruju, tako da služi samo špijunima, kao indikator „vredne robe“.

Strogo usmeravanje sa izvora daje potpunu putanju od izvorišta do odredišta kao sekvencu IP adresa. Datagram mora da sledi tu putanju. Opcija najviše koristi administratorima sistema da pošalju upozoravajuće pakete kada se poremete tabele za usmeravanje ili onda kada vrše vremenska merenja.

Opcijom *Približno usmeravanje sa izvora* zahteva se da paket prođe listu usmeri- vača navedenim redom, ali se ne zabranjuje prolazak i kroz drage usmerivače. Njom se najčešće zadaje samo nekoliko usmerivača da bi se forsirala određena putanja. Na pri- mer, da biste prisilili paket da iz Londona za Sidnej krene na zapad umesto na istok, možete navesti usmerivače u Njujorku, Los Anđelesu i Honoluluu. Opcija je najkorisnija kada politički ili ekonomski razlozi diktiraju (ne)prolazak kroz određene države.

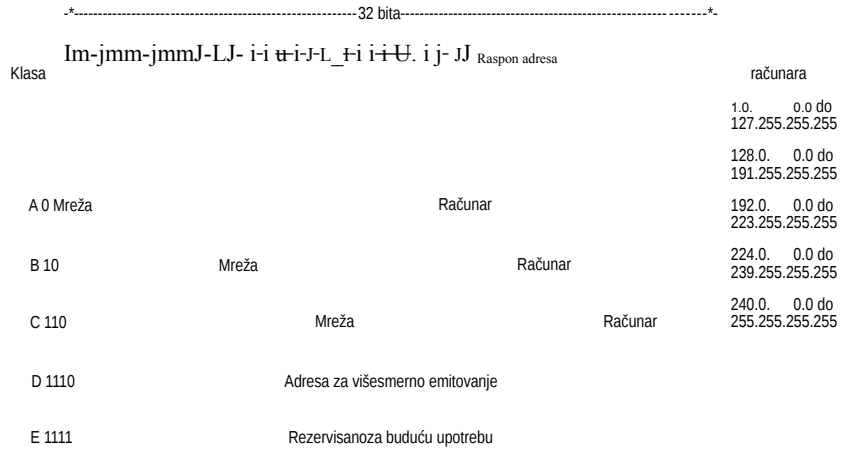
Opcija *Beleženje putanje* nalaže usmerivačima duž putanje da u polje *Opcije* dodaju svoju IP adresu. To administratorima sistema omogućava da povratnim putem pronađu greške u algoritmima za usmeravanje („Zašto paketi iz Hjustona za Dalas prvo idu u Tokio?“). Kada je ARPANET prvi put pušten u rad, nijedan paket nije prolazio kroz više od devet usmerivača, tako da je raspoloživih 40 bajtova bilo i više nego dovoljno. Kao što smo već naglasili, sada je to premalo.

Beleženje putanje liči na *Beleženje putanje,* osim što pored svoje 32-bitne IP adrese svaki usmerivač beleži i 32-bitnu vremensku oznaku. I ova opcija uglavnom služi za otkrivanje grešaka u algoritmima za usmeravanje.

5.6.2 IP adrese

Svaki računar i svaki usmerivač na Internetu imaju svoju IP adresu koja obuhvata broj njihove mreže i broj računara. Ta kombinacija je jedinstvena: dva računara na Internetu u načelu ne mogu imati istu IP adresu. Dužina svih IP adresa je 32 bita i one se koriste u poljima *Izvorišna adresa* i *Odredišna adresa* IP paketa. Treba naglasiti da se IP adresa u stvari ne odnosi na računat, već na mrežni interfejs, pa ako se računar nalazi u dve mreže, mora imati dve IP adrese. Međutim, u praksi je to retko, računari su uglavnom na jednoj mreži i imaju jednu IP adresu.

Već više decenija IP adrese se dele u pet kategorija prikazanih na slici 5-55. Takva podela je dobila naziv **klasno ađresiranje** (engl. *classful addressing).* Ona se više ne koristi, ali se u literaturi još uvek navodi. O zameni klasnog adresiranja drugim sistemom govoraćemo malo kasnije.

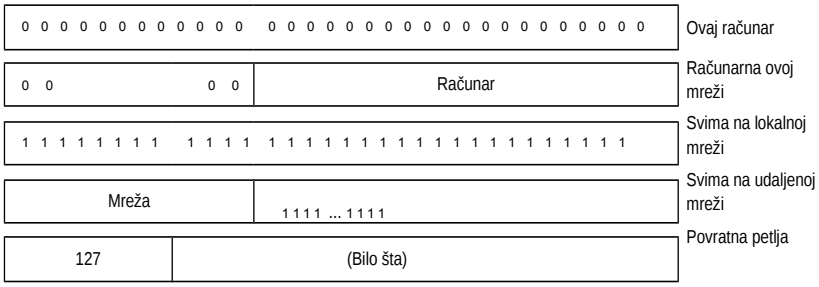


Slika 5-55. Formati IP adresa.

Formati klasa A, B, C i D omogućavaju 128 mreža sa po 16 miliona računara, 16.384 mreže sa po 64K računara i 2 miliona mreža (npr. lokalnih) sa 256 računara (iako su neke od njih specijalne). Podržano je i višesmerno emitovanje - upućivanje datagrama mnogim računalima. Adrese koje počinju na 1111 rezervisane su za buduću upotrebu. Na Internet je sada povezano preko 500.000 mreža, a taj broj raste svake godine. Da bi se izbegli sukobi, brojeve mreža dodeljuje neprofitna **Korporacija za** đo- **deljivanje** imena i **brojeva na Internetu** (engl. *Internet Corporation for Asigned Names and Numbers, ICANN).* Organizacija ICANN je zauzvrat poverila delove adresnog prostora različitim regionalnim vlastima, koje IP adrese dalje dele davaocima Internet usluga i drugim kompanijama.

Mrežne adrese, 32-bitni brojevi, obično se pišu decimalnom notacijom s tačkom (engl. *dotted decimal notatiori).* Tom notacijom, svaki od 4 bajta piše se decimalno, od 0 do 255. Na primer, 32-bitna heksadecimalna adresa C0290614 piše se kao 192.41.6.20. Najniža IP adresa je 0.0.0.0, a najviša 255.255.255.255.

Vrednosti 0 i -1 (sve jedinice) imaju specijalno značenje, što se vidi sa slike 5-56. Vrednost -1 koristi se kao adresa za neusmereno emitovanje svim računarima na naznačenoj mreži.



Slika 5-56. Specijalne IP adrese.

Adresu 0.0.0.0 koriste računari u fazi uključivanja. IP adresa s mrežnim brojem 0 označava tekuću mrežu. Takve adrese omogućavaju računarima da adresiraju sop- stvenu mrežu iako joj ne znaju broj (ali moraju znati njenu klasu jer od toga zavisi koliko će nula da uključe u adresu). Adresa koja se sastoji od samih jedinica omogućava neusmereno (difuzno) emitovanje u lokalnoj mreži. Adrese koje u mrežnom delu imaju ispravan broj mreže, a ostatak su same jedinice, omogućavaju slanje i difuzno distribuiranje paketa u naznačenoj udaljenoj mreži bilo gde na Internetu (premda mnogi administratori mreža isključuju ovu opciju). I na kraju, sve adrese tipa *Yll.xx.yy,zz* re- zervisane su za testiranje povratnom petljom. Paketi koji se pošalju na takvu adresu ne izlaze na žicu; oni se obrađuju lokalno kao paketi koji su došli spolja. To omogućava da se paketi pošalju lokalnoj mreži, a da pošiljalac ne mora da zna njen broj.

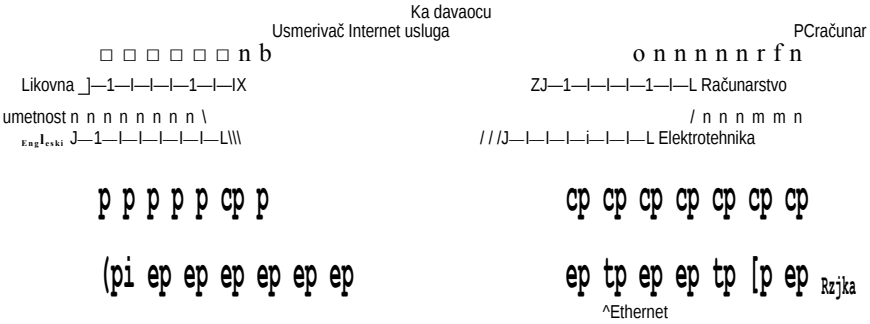
Podmreže

Kao što smo objasnili, svi računari na istoj mreži moraju imati isti mrežni broj. To pravilo IP adresiranja može da stvori probleme tokom rasta mreža. Razmotrite primer univerziteta koji je počeo od jedne mreže klase B, smeštene na Katedri za računarstvo unutar Ethernet mreže. Posle godinu dana, Katedra za elektrotehniku želi da iziđe na Internet, pa nabavlja repetitor da bi Ethernet s Katedre za računarstvo „rasteglila“ do svoje zgrade. Tokom vremena, i druge katedre nabavljaju računare, pa se brzo dostiže granica od četiri repetitora u jednoj Ethernet mreži. Sada je očigledno potrebna drugačija organizacija.

Nije lako dobiti još mrežnih adresa jer su adrese tražena roba, a univerzitet već ima dovoljno adresa za svojih 60.000 računara. Problem je u tome što se klase adresa A, B ili C odnose na jedinstvenu mrežu, a ne na skup lokalnih mreža. S vremenom sve više organizacija dolazi u ovu nezavidnu situaciju, pa se zato u sistem adresiranja unose male izmene.

Rešenje je nađeno u tome da se mreža interno podeli na više delova, ali da za spolj- ni

svet ostane celovita. Tipična današnja univerzitetska mreža mogla bi u tom smislu izgledati kao na slici 5-57, pri čemu je glavni usmerivač povezan s davaocem Internet usluga ili s regionalnom mrežom, a od njega se protežu linije ka brojnim lokalnim mrežama. Svaka lokalna mreža je s glavnim usmerivačem povezana preko sopstve- nog usmerivača (možda se lokalni usmerivač nalazi na okosnici lokalne mreže, ali to ovde nije od značaja).



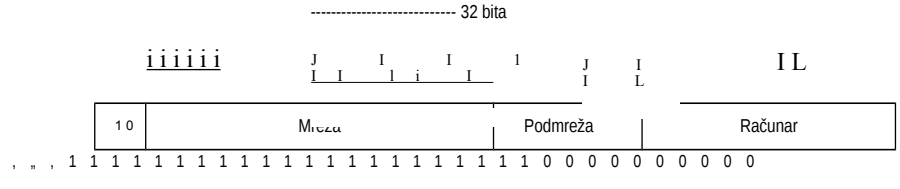
Slika 5-57. Univerzitetska mreža s lokalnim mrežama za svaku katedru.

U literaturi o Internetu, delovi mreže (u ovom slučaju to su Ethernet mreže) zovu se pođmreže (engl. *subnets).* Definicija se razlikuje od one koju smo naveli u 1. poglavlju, gde su podmrežu činili svi usmerivači i linije jedne mreže. Nadajmo se da če iz konteksta biti jasno na koje značenje mislimo. U ovom i sledećem odeljku kori- stičemo samo novouvedeno značenje pojma „podmreža".

Kada paket stigne u glavni usmerivač, kako ovaj zna na koju podmrežu (Ethernet) da ga postavi? Glavni usmerivač bi mogao da ima tabelu sa 65.536 odrednica za svaki računar/usmerivač na univerzitetu. To bi radilo, ali bi glavni usmerivač morao da ima vrlo veliku tabelu koja bi se morala često mčno održavati - kad god se uvede nov raču- nar, kada računar promeni mesto ili se isključi iz mreže.

U praksi se primenjuje druga šema. Umesto da se zadrži jedinstvena adresa klase B sa 14 bitova za mrežu i 16 bitova za računar, od računarskih bitova se jedan deo uzima za označavanje podmreže. Na primer, ako univerzitet ima 35 katedara, mogao bi da koristi 6-bitni broj za podmreže i 10-bitni broj za računare, što omogućava do 64 lokalne mreže sa po najviše 1022 računara (kao što smo ranije objasnili, ne mogu se koristiti 0 i -1). Kasnije bi se ovakva podela mogla promeniti ako se pokaže neuspešnom.

Da bi se ugradile podmreže, glavni usmerivač treba da ima masku podmreže (engl. *subnet mask)* koja naznačava podelu adrese između mreže + podmreže i računara (slika 5-58). Maska podmreže se takođe piše decimalnom notacijom s tačkom, s tim što se bitovi koji označavaju mrežu i podmrežu razdvajaju kosom crtom od bitova koji označavaju računar. Za primer na slici 5-58, maska podmreže može se pisati
25.255.252.0. Alternativno se može napisati samo /22, što znači da je maska podmreže dugačka 22 bita.



Slika 5-58. Mreža klase B izdeljena na 64 podmreže.

Podela na podmreže nije vidljiva izvan mreže, pa nije potrebno dobiti dozvolu od organizacije ICANN, niti menjati ijednu spoljnu bazu podataka. U našem primeru, prva podmreža bi mogla koristiti IP adrese počev od 130.50.4.1; draga bi mogla početi od 130.50.8.1, treća od 130.50.12.1 itd. Da biste razumeli zašto se adrese podmreža razlikuju za četiri, pogledajte ih u binarnom obliku:

Podmreža 1:10000010 00110010 000001|00 00000001
Podmreža 2: 10000010 00110010 000010|00 00000001
Podmreža 3:10000010 00110010 000011 [00 00000001

Ovde vertikalna crta (I) razgraničava broj podmreže od broja računara. Levo od nje je 6-bitni broj podmreže; desno je 10-bitni broj računara. (Brojevi podmreža su stvarno (decimalno) 1,2 i 3. Međutim, pošto se decimalnom notacijom s tačkom obuhvata cela treća grupa od 8 bitova, ti brojevi su onda 4, 8 i 12.)

Da biste videli kako podmreža radi, treba da objasnimo kako se IP paketi obrađuju u usmerivaču. Svaki usmerivač ima tabelu sa izvesnim brojem IP adresa tipa (mreža, 0) i tipa (ova mreža, računar). Pomoću adresa prvog tipa stiže se do udaljenih mreža, dok se pomoću adresa dragog tipa dosežu lokalni računari. Svakoj tabeli pridružen je i mrežni interfejs preko koga se pristupa odredištu, kao i neke druge informacije.

Kada stigne IP paket, traži se njegova odredišna adresa u tabeli za usmeravanje. Ako je paket upućen udaljenoj mreži, on se prosleđuje sledećem usmerivaču preko in- terfejsa označenog u tabeli. Ukoliko je paket upućen lokalnom računaru (npr. računara na usmerivačevoj lokalnoj mreži), on se direktno šalje na odredište. Ako u tabeli nema naznačene mreže, paket se upućuje podrazumevanom usmerivaču čije su tabele iscrpnije. Ovaj algoritam znači da svaki usmerivač treba da vodi računa samo o dragim mrežama i o lokalnim računarima, ne i o parovima (mreža, računar), što znatno smanjuje veličinu tabela za usmeravanje.

Kada se uvedu podmreže, menjaju se tabele za usmeravanje i dodaju odrednice sledeća dva tipa: (ova mreža, podmreža, 0) i (ova mreža, ova podmreža, računar). Tako, usmerivač na podmreži *k* zna kako da stigne do svih dragih podmreža i kako da stigne do svih računara na podmreži *k.* On ne mora ništa da zna o računarima na dragim podmrežama. U stvari, svaki usmerivač treba samo da sa adresom logički sabere masku podmreže i tako se oslobodi dela koji se odnosi na računar, a zatim da rezultat potraži u svojim tabelama (pošto odredi klasu mreže). Na primer, kada paket stigne u glavni usmerivač, njegova odredišna adresa (130.50.15.6) logički se sabere s maskom podmreže (255.255.252.0/22), što daje adresu 130.50.12.0. Ta adresa se traži u tabelama za usmeravanje da bi se pronašla izlazna linija ka usmerivaču podmreže 3. Na taj način, obrazovanjem podmreža stvara se trostepena hijerarhija: mreža, podmreža, računat-, što znatno smanjuje tabele usmerivača.

CTDR - Besklasno međudomensko usmeravanje

Internet protokol (IP) široko se koristi već decenijama. On radi sasvim dobro, o čemu svedoči eksponencijalni rast Interneta. Nažalost, IP postaje žrtva svoje sopstve- ne popularnosti: sve češće mu ponestaje adresa. Katastrofa na horizontu pobudila je mnogo žučnih rasprava između korisnika Interneta o tome šta preduzeti. U ovom odeljku detaljnije

ćemo opisati sam problem, kao i više predloženih rešenja.

Nekoliko vizionara je davne 1987. godine predvidelo da će Internet jednoga dana obuhvatiti možda 100.000 mreža. Stručnjaci su se podsmešljivo zgledali na takvu izjavu, smatrajući da će za to biti potrebne decenije, ako do toga uopšte dođe. Međutim, 1996. godine na Internet je priključena stohiljadita mreža i, kao što smo na početku napomenuli, pojavio se problem nedostatka adresa. U stvari, teorijski postoji preko dve milijarde adresa, ali se milioni njih ne mogu koristiti zbog prakse da se adresni prostor organizuje u klase (slika 5-55). Najveći rasipnik je, konkretno, klasa B. Za većinu organizacija, mreža klase A sa 16 miliona adresa je prevelika, a mreža klase C sa 256 adresa premala. Čini se daje pravi izbor mreža klase B sa 65.536 adresa. U Internet žargonu, ova situacija je poznata kao problem tri medvedića (kao u bajci *Zlatokosa i tri medvedića).*

U stvarnosti, adresni prostor klase B prevelik je za većinu organizacija. Ispitivanja su pokazala da više od polovine mreža klase B imaju manje od 50 računara. Njih bi mogla da opsluži mreža klase C, ali nema sumnje da je svaka organizacija koja je tražila klasu B mislila da će u bliskoj budućnosti prerasti 8-bitno polje za adresu računara. Kad se osvrnemo unazad, možda bi bilo bolje daje u mrežama klase C za adresu računara bilo previđeno deset, a ne osam bitova, pa bi ona mogla sadržati 1022 računara. Da je tako učinjeno na početku, verovatno bi se većina organizacija opredelila za mrežu klase C i njih bi bilo pola miliona (u odnosu na samo 16.384 mreže klase B).

Teško možemo kriviti projektante mreža za to što nisu obezbedili više (kraćih) adresa klase B. U doba kada se odlučivalo o obrazovanju tri klase mreža, Internet je bio eksperimentalna mreža koja je povezivala glavne istraživačke institucije (univerzitete) u SAD i još samo nekoliko kompanija i vojnih ustanova koje su istraživačke podatke razmenjivale preko mreže. Niko tada nije ni slutio da će Internet izrasti u sistem masovnih tržišnih komunikacija koji konkuriše telefonskoj mreži. U to vreme je neko verovatno pomislio: „U SAD ima oko 2000 koledža i univerziteta. Kada bi se svi oni povezali na Internet i kada bi im se pridružili i univerziteti iz inostranstva, još uvele ne bismo dostigli cifru od 16.000 jer na ćelom svetu nema toliko visokoškolskih ustanova. Osim toga, obrada je brža kada je deo adrese koji se odnosi na računar ceo broj bajtova“.

Međutim, kada bi se delu adrese koji se odnosi na mrežu klase B dodelilo 20 bitova, pojavio bi se dragi problem: eksplozija tabela za usmeravanje. S gledišta usmerivača, adresni IP prostor je dvostepena hijerarhija, s brojevima mreža i brojevima računara. Usmerivači ne moraju ništa da znaju o računarima, ali moraju sve da znaju o mrežama. Ako u upotrebu uđe pola miliona mreža klase C, svaki usmerivač na Internetu bi, između ostalog, morao da ima tabelu za usmeravanje s pola miliona odrednica, po jednom za svaku mrežu, na osnovu koje bi određivao izlaznu liniju za tu mrežu.

Samo fizičko skladištenje tabela sa po pola miliona odrednica verovatno je izvod- ljivo, premda skupo rešenje za kritične usmerivače koji tabele čuvaju u statičkom RAM-u na ulazno-izlaznim karticama. Ozbiljniji problem je to što složenost raznih algoritama koji rade s tabelama ne raste linearno s dužinom tabele, već brže. Gore je to što je većina postojećeg softvera i firmvera projektovana u vreme kada se Internet sastojao od 1000 povezanih mreža i kada je broj od 10.000 mreža izgledao nedostižan. Odluke koje su pri projektovanju donošene tada, sada su daleko od optimalnih.

Osim toga, neki algoritmi za usmeravanje (npr. protokoli zasnovani na vektoru rastojanja) zahtevaju da svaki usmerivač povremeno emituje svoju tabelu. Što je tabela veća, veća je i

verovatnoća da će se neki njen deo usput izgubiti, što na dragom kraju onemogućava dobijanje potpunih podataka i možda dovodi do nestabilnosti usmeravanja.

Problem tabela za usmeravanje mogao bi se resiti produžavanjem hijerarhije, tj. kada bi, na primer, svaka IP adresa sadržala polje za državu, pokrajinu, grad, mrežu i računar. Tada bi svaki usmerivač jedino morao da zna kako da podatke isporuči svakoj stranoj državi, pokrajinama u svojoj zemlji, gradovima u tim pokrajinama i mrežama u tim gradovima. Takvo rešenje, nažalost, zahtevalo bi IP adresu mnogo dužu od 32 bita i neefikasno bi koristilo adresni prostor (Lihtenštajn bi dobio isto onoliko bitova koliko i SAD).

Jednom reči,.neka rešenja određene probleme rešavaju, ali i stvaraju nove. Rešenje koje je realizovano i koje je Internetu omogućilo da za neko vreme predahne, zove se **besklasno međudomensko usmeravanje** (engl. *Classless InterDomain Routing, CIDR)*. Ono je opisano u RFC dokumentu 1519. Osnovna zamisao je bila da se preostali nedodeljen adresni prostor podeli u blokove različite veličine, ne vodeći računa o klasama. Kada bi lokacija zahtevala, recimo, 2000 adresa, dodeljivan joj je blok od 2048 adresa koji leži na 2048-bajtnoj granici unutar adresnog prostora.

Kada se napuste klase mreža, prosleđivanje postaje složenije. U starom sistemu klasa prosleđivanje funkcioniše na sledeći način. Kada paket stigne u usmerivač, IP adresa se logički pomera udesno za 28 bitova da bi se dobio 4-bitni broj klase. Paketi se zatim sortiraju na 16 načina (ako je to podržano) u klase A, B, C i D: osam slučajeva u klasu A, četiri u klasu B, dva u klasu C i po jedan u klase D i E. Kod za svaku klasu tada demaskira 8-, 16- ili 24-bitni mrežni broj koji se poravnava udesno u 32-bitnu reč. Tada se traži broj mreže u tabeli A, B ili C, pri čemu su tabele A i B obično indeksirane, a tabela C heširana. Kada se pronađe odgovarajuća odrednica, tu je i izlazna linija i paket se prosleđuje.

Sa sistemom CIDR, ovaj jednostavan algoritam više ne radi. Umesto toga, svaka odrednica tabele za usmeravanje proširuje se 32-bitnom maskom. Tako nastaje jedinstvena tabela usmeravanja za sve mreže, koja se sastoji od niza tripleta: IP adresa, maska podmreže, izlazna linija. Kada paket stigne, najpre se iz njega izvlači IP adresa. Zatim se (konceptualno) tabela usmeravanja pregleda - odrednica po odrednica - maskiranjem odredišne adrese i poređenjem sa odrednicama u tabeli da bi se našla odgovarajuća. Ako se nađe više odgovarajućih odrednica koje se razlikuju samo po dužini maske podmreže, koristi se ona s najdužom maskom. Na primer, ako se s traženom adresom slažu odrednice s maskama /20 i /24, koristi se odrednica s maskom /24.

Smišljeni su složeni algoritmi za ubrzanje postupka poređenja adresa (Ruiz-San- chez i saradnici, 2001). U komercijalnim usmerivačima koriste se posebni VLSI či- povi u koje su hardverski ugrađeni takvi algoritmi.

Da biste bolje razumeli kako radi algoritam za prosleđivanje, razmotrimo primer u kome su raspoloživi milioni adresa, počev od adrese 194.24.0.0. Pretpostavimo da Univerzitetu u Kembridžu treba 2048 adresa i da su mu dodeljene adrese između 194.24.0, 0 i 194.24.7.255, kao i maska 255.255.248.0. Posle toga, Oksfordski univerzitet traži 4096 adresa. Pošto blok od 4096 adresa mora ležati na 4096-bajtnoj granici unutar adresnog prostora, Oksfordu se ne mogu dati adrese počev od 194.24.8.0. Umesto toga, on dobija adrese između 194.24.16.0 i 194.24.31.255, kao i masku podmreže 255.255.240.0. Zatim, Edinburški univerzitet traži 1024 adrese i dobija ih između 194.24.8.0 i 194.24.11.255, i masku podmreže 255.255.252.0. Dodeljeni adresni prostor sumarno je prikazan na slici 5-59.

Univerzitet	Prva adresa	Posiednja adresa	Broj adresa	Piše se
Kembridž	194.24.0.0.	194.24.7.255.	2048	194.24.0.0/21
Edinburg	194.24.8.0.	194.24.11.255.	1024	194.24.8.0/22
(Nedodeljeno)	194,24,12.0.	194.24.15.255.	1024	194.24,12.0/22
Oksford	194.24.16.0,	194.24.31.255.	4096	194.24.16.0/20

Slika 5-59. Dodeljene IP adrese.

Tabele za usmeravanje širom sveta sada su ažurirane trima novim odrednicama. Svaka odrednica sadrži osnovnu adresu i masku podmreže. One u binarnom obliku izgledaju ovako:

Adresa	Maska
K: 11000010 00011000 00000000 00000000	*11111111 11111111 11111000 00000000*
E: 11000010 00011000 00001000 00000000	11111111 11111111 11111100 00000000
O: 11000010 00011000 00010000 00000000	11111111 11111111 11110000 00000000

Razmotrite sada šta se događa kada stigne paket sa odredišnom adresom 194.24.17.4, koja u binarnom zapisu izgleda kao sledeći 32-bitni niz:

Taj niz se najpre logički sabere s maskom Kembridža, pri čemu se dobija:

11000010 00011000 00010000 00000000

Ta vrednost se ne poklapa sa osnovnom adresom Kembridža, pa se prvobitni niz ponovo logički sabira sa Edinburškom maskom i dobija:

11000010 00011000 00010000 00000000

Ni ovo se ne slaže sa osnovnom adresom Edinburga, pa se zato proba sa Oksford- skom maskom, što daje:

11000010 00011000 00010000 00000000

Dobijena vrednost odgovara osnovnoj adresi Edinburga. Ako se dalje u tabeli ne pronađe još neko slaganje, upotrebljava se odrednica za Oksford i paket se šalje linijom koja je u njoj naznačena.

Razmotrimo sada tri pomenuta univerziteta s gledišta usmerivača koji se nalazi u Omahi (Nebraska) i koji ima samo četiri izlazne linije (za Mineapolis, Njujork, Dalas i Denver). Kada softver tog usmerivača dobije tri nove odrednice, on zapaža da sve tri može da kombinuje u jedinstvenu **grupnu odrednicu** (engl. *aggregate entry)*

194.24.0. 0/19, sa sledecom binarnom adresom i maskom podmreže:

11000010 0000000 00000000 00000000 11111111 11111111 11100000 00000000

Ta odrednica šalje pakete namenjene bilo kom od tri pomenuta univerziteta u Njujork. Grupišući odrednice, usmerivač u Omahi je njihov broj smanjio za dva.

Ako Njujork za sav saobraćaj s Velikom Britanijom ima jedinstvenu liniju do Londona, i on može da napravi grupnu odrednicu. Međutim, ako ima odvojene linije za London i Edinburg, onda mora da ima tri zasebne odrednice. Grupisanje odrednica uveliko se koristi širom Interneta da bi se uštedeo prostor u tabelama usmerivača.

Na kraju ovog primera napomenimo i to da grupna odrednica u Omahi šalje u Njujork i pakete s nedodeljenim adresama. Sve dok su adrese stvarno nedodeljene, to ne smeta, jer se ne pretpostavlja da one uopšte postoje. Međutim, ako one kasnije budu dodeljene nekoj kompaniji u Kaliforniji, za rad s njima mora se uneti nova odrednica 194.24.12.0/22.

NAT-Prevođenje mrežnih adresa

IP adresa nema mnogo. Davalac Internet usluga mogao je imati adresu /16 (bivšu klasu B), koja mu je omogućavala usluživanje 65.534 računara. Ako je imao više korisnika, nastajali su problemi. Za kućne korisnike koji su se povezivali modemom, IP adresa se mogla dodeljivati dinamički u trenutku prijavljivanja na mrežu i ponovo oduzimati kada se korisnik odjavi. Na taj način, jedinstvena adresa /16 mogla je zadovoljiti do 65.534 istovremeno aktivna korisnika, što je možda bilo dovoljno za davaoca Internet usluga koji ima više stotina hiljada mušterija. Kako se koja sesija završi, IP adresa se ponovo dodeljuje drugom korisniku. Dok takva strategija može da zadovolji davaoca Internet usluga koji ima priličan broj kućnih korisnika, ona ne zadovoljava u situacijama kada davalac ima prvenstveno poslovne korisnike.

Poslovni korisnici očekuju da stalno budu na mreži tokom radnog vremena. I male firme, kao što su turističke agencije sa samo tri zaposlena, imaju više računara povezanih u lokalnu mrežu, baš kao i velike korporacije. Neki od njih su PC računati zaposlenih, dok drugi mogu da budu Web serveri. Lokalna mreža je obično preko usmerivača i iznajmljene linije povezana s davaocem Internet usluga koji joj obezbe- đuje stalno prisustvo na mreži. Takav sistem zahteva da svaki računar čitavog dana ima istu IP adresu. Zbog toga, ukupan broj računara svih poslovnih korisnika jednog davaoca Internet usluga ne može da pređe broj IP adresa koje su dodeljene davaocu. Za adresu /16 to znači da ukupan broj računara ne može da pređe granicu od 65.534. Ako davalac Internet usluga ima na desetine hiljada poslovnih korisnika, on tu granicu brzo dostiže.

Situacija postaje sve ozbiljnija kako se sve veći broj kućnih korisnika pretplaćuje na ADSL liniju ili na kablovski Internet. Ove usluge su karakteristične po tome što (1) korisnik dobija stalnu IP adresu i (2) što se ne naplaćuju po uspostavljenoj vezi (već paušalno), tako da se mnogi korisnici uopšte ne odjavljuju s mreže. Takav razvoj situacije samo zaoštrava problem manjka IP adresa. Dodeljivanje IP adresa u hodu, kao što se radi s korisnicima koji se povezuju modemom, ovde nema efekta jer broj IP adresa koje su u svakom trenutku u opticaju može više puta da premaši broj adresa koje davalac ima.

Da bi sve bilo još komplikovanije, mnogi korisnici ADSL linija i kablovskog Interneta imaju više kućnih računara - često po jedan za svakog člana porodice, i svi žele da su stalno umreženi preko iste IP adrese koju im je dodelio davalac. Rešenje je u tome da se svi-računan u kući povežu u lokalnu mrežu snabdevenu usmerivačem. S gledišta davaoca Internet usluga, porodica je sada isto što i mala firma koja ima nekoliko računara. Dobro došli kod „Petrovića i sinova“!

Nedostatak IP adresa nije akademski problem koji nas može pogoditi negde u budućnosti - on je već oko nas. Dugoročno rešenje bi bilo da čitav Internet pređe na protokol IPv6 sa 128-bitnim adresama. Taj proces jeste u toku, ali će proći godine pre nego što se dovrši. Zbog toga su mnogi osetili potrebu za nekim brzim rešenjem. Ono se pojavilo u obliku sistema za **prevođenje mrežnih adresa** (engl. *Network Address Translation, NAT),* opisanog u RFC dokumentu 3022. U nastavku ćemo ga ukratko objasniti, a više detalja o njemu potražite kod Dutchera (2001).

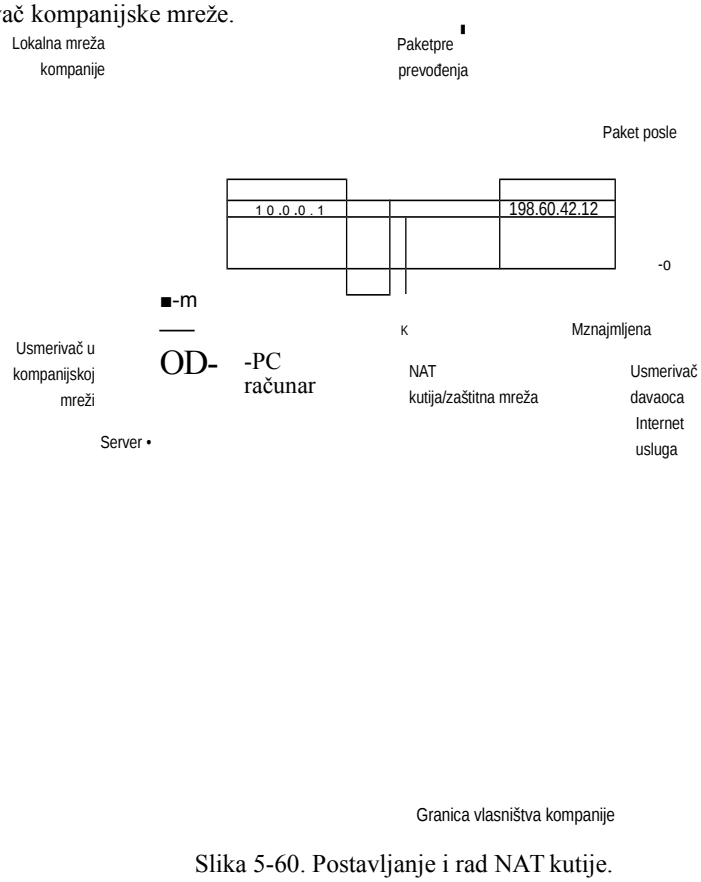
Sistem NAT svakoj kompaniji đodeljuje samo jednu IP adresu (ili najviše, manji broj adresa) za saobraćaj na Internetu. *Unutar* kompanije, svaki računar dobija jedinstvenu IP adresu koja služi za interni saobraćaj. Međutim, kada paket napušta kompaniju i odlazi ka davaocu Internet usluga, njegova adresa se prevodi. Da bi takav sistem mogao da radi, tri opsega IP adresa rezervisana su za privatne potrebe. Kompanije ih mogu interno koristiti

kako god žele. Jedino ograničenje je da se paketi s tim adresama ne smeju pojaviti na Internetu. Tri rezervisana opsega su:

10.0.0.0 - 10.255.255.255/8 (16.777.216 računara)
172.16.0.0 - 172.31.255.255/12 (1.048.576 računara)
192.168.0.0 - 192.168.255.255/16 (65.536 računara)

Prvi opseg obezbeđuje 16.777.216 adresa (osim adresa 0 i -1, kao i obično) i kompanije ga često koriste, čak i kada nemaju potrebe za tolikim brojem adresa.

Rad sistema NAT prikazanje na slici 5-60. Unutar kompanije, svaki računar ima jedinstvenu adresu oblika 10.*x.y.z.*. Međutim, kada paket napusti kompaniju, on prolazi kroz **NAT kutiju** (engl. *NAT box),* koja internu IP adresu izvorišta (na slici je to adresa 10.0.0.1) prevodi u pravu IP adresu kompanije (198.60.42.12). NAT kutija se često kombinuje u istom uređaju sa zaštitnom barijerom koja brižno nadzire sve što ulazi u kompaniju i sve što iz nje izlazi. Zaštitne barijere ćemo razmatrati u 8. poglavlju. NAT kutija se može smestiti i u usmerivač kompanijske mreže.



Slika 5-60. Postavljanje i rad NAT kutije.

Dosad smo ćutke prelazili preko jednog „sitnog“ detalja: kada stigne odgovor (npr. sa Web servera), on je, naravno, adresiran na 198.60.42.12. Kako onda NAT kutija zna kojom adresom da je zameni? To je problem koji NAT ne može da reši. Kada bi u IP zaglavlju bilo viška polja, u njima bi se mogao beležiti stvarni pošiljalac, ali - nažalost - postoji samo jedan neiskorišcen bit. U stvari, mogla bi se napraviti nova opcija za čuvanje prave izvorišne adrese, ali bi to zahtevalo da se kod protokola IP izmeni na svim usmerivačima na Internetu kako bi mogli da razumeju novu opciju. To baš ne obećava brzo rešavanje problema.

U stvari, radi se sledece. Tvorci sistema NAT zapazili su da IP paketi kao koristan teret većinom nose TCP ili UDP okvire. Kada u 6. poglavlju budemo razmatrali protokole TCP i

UDP, videćemo da zaglavlja njihovih okvira sadrže izvorišni i odredišni priključak. U nastavku ćemo govoriti samo o TCP priključcima, ali isto važi i za UDP priključke. Priključci su 16-bitni celi brojevi koji ukazuju gde počinje i gde se završava TCP veza. Oni se nalaze u polju koje omogućava da NAT radi.

Kada proces poželi da uspostavi TCP vezu sa udaljenim procesom, on se priključi na nezauzet TCP priključak na sopstvenom računaru. To je **izvorišni priključak** (engl. *source port)* koji TCP kodu saopštava gde da šalje dolazne pakete koji pripadaju priključku. Proces obezbeđuje i **odredišni priključak** (engl. *destinationport*) - mesto

na udaljenom računam na kome treba predavati pakete. Priključci 0-1023 rezervisani su za ustaljene usluge. Na primer, priključak 80 koriste Web serveri, tako da udaljeni klijenti uvek znaju da ih pronađu. Svaka poslata TCP poruka sadrži i izvorišni i odredišni priključak. Oni zajedno identifikuju procese koji koriste vezu, i to na oba kraja.

Korišćenje priključaka biće jasnije ako upotrebimo jednu analogiju. Zamislite kompaniju koja ima jedinstven glavni telefonski broj. Kada on zazvoni, javlja se operater i pita pozivaoca koji lokal želi, a zatim ga spaja s tim brojem. Glavni broj odgovara IP adresi kompanije, a lokali na dva kraja odgovaraju priključcima. Priključci su dodatnih 16 bitova adrese i određuju koji proces treba da dobije koji dolazili paket.

Pomoću polja *Izvorišni priključak* možemo da reširno naš problem preslikavanja (mapiranja) adresa. Kad god izlazni paket uđe u NAT kutiju, izvorišna adresa *10 oc.y.z* zamenjuje se pravom IP adresom kompanije. Osim toga, TCP polje *Izvorišna adresa* zamenjuje se indeksom u tabeli za prevođenje sa 65.536 odrednica u NAT kutiji. Ta odrednica sadrži originalnu IP adresu i originalni priključak izvorišta. Na kraju, kontrolni zbirovi IP zaglavlja i TCP zaglavlja ponovo se izračunavaju i umeću u paket. *Izvorišni priključak* je neophodno zameniti, pošto se može desiti da, na primer, računati 10.0.0.1 i 10.0.0.2 koriste isti priključak, 5000, pa se samo preko *Izvorišnog priključka* ne može identifikovati proces koji šalje poruku.

Kada paket stigne u NAT kutiju od davaoca Internet usluga, vadi se *Izvorišni priključak* iz TCP zaglavlja i koristi kao indeks u tabeli za prevođenje NAT kutije. Iz nađene odrednice čitaju se interna IP adresa i originalni TCP *Izvorišni priključak,* i kopiraju se u paket. Zatim se ponovo izračunavaju kontrolni zbirovi IP i TCP zaglavlja, i umeću se u paket. Paket se potom prosleđuje usmerivaču u kompanijskoj mreži, radi uobičajene isporuke na adresu 10.A'.y.z.

Sistem NAT se može iskoristiti i za ublažavanje problema manjka IP adresa kod korisnika ADSL linija i kablovskog Interneta. Kada davalac Internet usluga dodeljuje korisniku adresu, on tada koristi 10,*x.y.z* adrese. Kada korisnički paketi preko davaoca krenu na Internet, oni prođu kroz NAT kutiju koja im adrese prevede u pravu IP adresu davaoca Internet usluga. Na putu sa Interneta ka korisniku, adrese paketa se obrnuto preslikavaju. U pogledu adresa, davalac Internet usluga i njegovi ADSL/kablovski korisnici ostatku Interneta izgledaju kao jedna velika kompanija.

Iako opisani sistem na izvestan način rešava problem, mnogi ga smatraju najobičnijim ruglom. Evo, ukratko, šta mu se u osnovi zamera. Prvo, NAT narušava arhitekturu IP modela u kome svaka IP adresa jedinstveno identifikuje samo jedan računat na čitavom svetu. Softverska struktura Interneta u celini je izgrađena na toj pretpostavci. Uz sistem NAT, hiljade računara mogu da koriste adresu 10.0.0.1 (a to i čine).

Drugo, NAT pretvara Internet iz mreže koja radi bez uspostavljanja direktne veze u mrežu sa izvesnim uspostavljanjem direktne veze. Problem je u tome što NAT kutija mora da održava informacije (o preslikavanju) za svaku vezu koja prolazi kroz nju. Kada mreža treba da održava informacije o stanju veze, to onda liči na mrežu koja radi sa uspostavljanjem (a ne bez uspostavljanja) direktne veze. Ako NAT kutija otkaže i tabela preslikavanja propadne, kidaju se sve TCP veze. U odsustvu NAT-a, otkazivanje usmerivača nema uticaja na TCP prenos. Kada se posle nekoliko sekundi tajmer pošiljaoca isključi, njegov proces ponovo šalje sve nepotvrđene pakete. Uz NAT, Internet postaje ranjiv kao mreža s komutiranjem električnih kola.

Treće, NAT narušava osnovno pravilo raspoređivanja protokola po slojevima: sloj *k* ne sme da pravi nikakve pretpostavke o tome staje sloj *k* + 1 smestio u polje za korisničke podatke. Taj osnovni princip i održava međusobnu nezavisnost slojeva. Ako se pojavi nova verzija protokola TCP, npr. TCP-2, s drugačijim rasporedom zaglavlja (npr. sa 32-bitnim priključcima), NAT će zakazati. Čitava ideja protokola smeštenih po slojevima svodi se na obezbeđivanje nezavisnosti izmena pojedinih slojeva. NAT tu nezavisnost ruši.

Četvrto, od procesa na Internetu se ne traži da koriste protokole TCP ili UDP. Ako korisnik na računaru A odluči da za razgovor s korisnikom na računani *B* upotrebi neki nov transportni protokol (npr. za multimedijske aplikacije), NAT kutija će osujetiti aplikaciju jer neće biti u stanju da ispravno locira TCP *Izvorišni priključak.*

Peto, neke aplikacije umeću IP adrese u sam tekst. Primalac tada preuzima adrese i koristi ih. Pošto NAT ništa ne zna o tim adresama, on ih ne može zameniti, pa će propasti pokušaji da se one iskoriste na udaljenom kraju veze. Na opisani način radi **protokol za prenos datoteka** (engl. *File Transfer Protocol, FTP)* i može da otkaže kada postoji sistem NAT ako se ne preduzmu specijalne mere. Slično svojstvo ima protokol
H. 323 za Internet telefoniju (o kome ćemo govoriti u 7. poglavlju) i može da otkaže u prisustvu NAT-a. NAT bi se mogao naterati da sarađuje s protokolom H.323, ali krpiti kod u NAT kutiji svaki put kada se pojavi nova aplikacija nije baš preporučljivo.

Šesto, pošto TCP polje *Izvorišni priključak* ima 16 bitova, najviše 6.5.536 računara može biti preslikano u jednu IP adresu. U stvari, taj broj je nešto manji jer se prvih 4096 priključaka čuvaju za posebne namene. Međutim, da je na raspolaganju više IP adresa, svaka bi zadovoljila 61.440 računara.

U RFC dokumentu 2993 razmotreni su ovi i drugi problemi sa sistemom NAT. Protivnici NAT-a u načelu prigovaraju da se rešavanjem problema nedovoljnog broja IP adresa pomoću jednog privremenog i nakaradnog rešenja smanjuje pritisak za prelazak na pravo rešenje - IPv6, što i jeste najgore.

5.6.3 Protokoli za upravljanje na Internetu

Osim protokola IP, koji se koristi za prenos datoteka, na Internetu postoji više upravljačkih protokola koji se koriste u mrežnom sloju, uključujući i ICMP, ARP, RARP, BOOTP i DHCP. U ovom odeljku govorimo o njima.

Protokol za upravljanje porukama na Internetu

Rad Interneta budno nadziru usinerivači. Kada se dogodi nešto neočekivano, o događaju se izveštava **protokolom za upravljanje porukama na Internetu** (engl. *Internet Control Message Protocol, ICMP),* koji se koristi i za testiranje Interneta. Definisano je više od deset vrsta ICMP poruka, od kojih su najvažnije navedene na slici 5-61. ICMP poruka bilo koje vrste kapsulira se u IP paket.

Poruka ODREDIŠTE NEDOSTUPNO koristi se kada podmreža ili usmerivač ne mogu da lociraju odredište ili kada paket s postavljenim bitom *NF* ne može da se isporuči jer mu je na putu mreža koja ograničava veličinu paketa.

Vrsta poruke	Opis
Destination unreachable (Odredište nedostupno)	Paket se ne može isporučiti
Time exceeded (Isteklo vreme)	
Parameter problem (Greška u parametrima)	Neispravno polje u zaglavlju
Source quench (Prigušivanje izvorišta)	Prigušni paket
Redirect (Preusmeravanje)	Poučavanje usmerivača o topologiji
Echo (Eho)	Proveravanje aktivnosti računara
Echo reply (Odgovor na eho)	Potvrda aktivnosti računara
Timestamp request (Zahtev s vremenskom oznakom)	Isto što i Eho, s vremenskom oznakom
Timestamp reply (Odgovor na zahtev s vremenskom oznakom)	Isto što i Odgovor na eho, š vremenskom oznakom

Slika 5-61. Osnovne vrste ICMP poruka.

Poruka VREME ISTEKLO šalje se kada se odbaci paket jer mu je životni vele istekao. Ovaj događaj ukazuje na to da paketi kruže u petlji, da postoji izuzetno zagušenje ili da je rok tajmera suviše kratak.

Poruka GREŠKA U PARAMETRIMA ukazuje na to da je u nekom polju zaglavlja otkrivena nedozvoljena vrednost. Poruka signalizira na grešku u IP softvera pošiljaoca ili možda na greške u softveru usputnih usmerivača.

Poruka PRIGUŠIVANJE IZVORIŠTA prvobitno je korišćena za opominjanje računara koji prebrzo šalju pakete. Kada računar primi takvu poruku, očekuje se da uspori slanje. Danas se retko koristi jer pri zagušenju takvi paketi samo dodaju ulje na vatru. Upravljanje zagušenjem na Internetu danas se uglavnom obavlja u transportnom sloju; govoricemo o tome u 6. poglavlju.

Poruku PREUSMERAVANJE šalje usmerivač koji smatra da je paket pogrešno usmeren. On time pošiljaoca obaveštava o problemu.

Poruke EHO i ODGOVOR NA EHO koriste se pri utvrđivanju da li je određeno odredište dostupno i aktivno. Kada primi poruku EHO, odredište treba da vrati poruku ODGOVOR NA EHO. Slične su i poruke ZAHTEV S VREMENSKOM OZNAKOM i ODGOVOR NA ZAHTEV S VREMENSKOM OZNAKOM, osim što se u odgovoru beleže vreme stizanja zahteva i vreme slanja odgovora. Ovaj par poruka služi za određivanje performansi mreže.

Osim navedenih, definisano je još poruka. Njihov spisak se nalazi na Internetu, na adresi www.iana.org/assignments/icmp-parameters.

ARP - Protokol za razrešavanje adresa

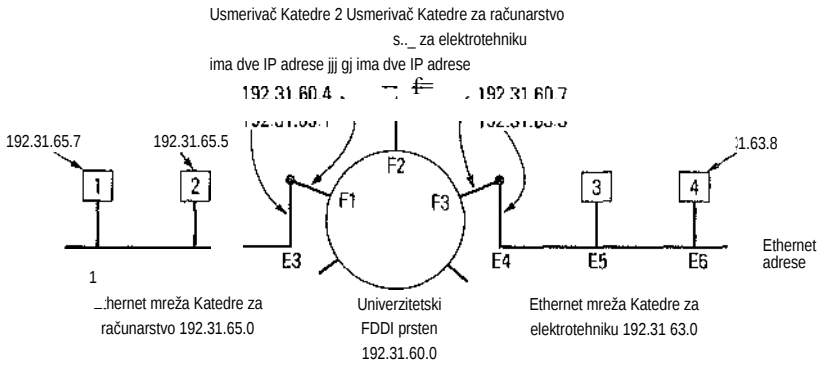
Iako svaki računar na Internetu ima jednu (ili više) IP adresa, one se u stvari ne mogu koristiti za slanje paketa jer hardver sloja veze podataka ne razume Internet adrese. Danas su računati u većini kompanija i univerziteta povezani u lokalne mreže preko mrežne kartice koja razume samo lokalne adrese. Na primer, Ethernet kartice

oduvek imaju ugrađenu 48-bitnu Ethernet adresu. Proizvođači Ethernet kartica zahtevaju od centralne ovlašćene organizacije da im dodeli blok adresa kako bi svaka kartica imala jedinstvenu adresu (i time se izbegli sukobi u situaciji kada se dve kartice sa istom adresom nađu na istoj lokalnoj mreži). Kartice šalju i primaju okvire zasnovane na 48-bitnim Ethernet adresama. One ništa ne znaju o 32-bitnim IP adresama.

Sada se postavlja pitanje: kako se IP adrese preslikavaju u adrese sloja veze podataka, npr. u Ethernet adrese? Da bismo to objasnili, poslužimo se primerom sa slike 5-62, na kojoj je prikazan mali univerzitet koji ima nekoliko mreža klase C (sada zvanih /24). Tu imamo dve Ethernet mreže, jednu na Katedri za računarstvo, sa IP adresom 192.31.65.0 ijednu na Katedri za elektrotehniku, sa IP adresom 192.31.63.0. One su povezane univezitetskom prstenastom okosnicom (npr. tipa FDDI), čija je adresa 192.31.60.0. Svaki računar na Ethernet mreži ima jedinstvenu Ethernet adresu, označenu sa *E1* do *E6,* a svaki računar na FDDI prstenu ima FDDI adresu, označenu sa *F1* do *F3.*

Počnimo tako što čemo posmatrati kako korisnik računara 1 šalje paket korisniku računara 2. Pretpostavimo da pošiljalac zna ime potencijalnog primaoca, npr. ime *morija® eagle.cs.uni.edu.* Prvi korak je pronaći IP adresu računara 2, zvanog *ea- gle.cs.uni.edu.* To se radi pomoću DNS sistema, o kome ćemo govoriti u 7. poglavlju. Zasad ćemo prihvatiti da DNS sistem vraća adresu računara 2 (192.31.65.5).

Softver gornjeg sloja na računaru 1 pravi sada paket s vrednošću 192.31.6.5.5 u polju *Odredišna adresa* i predaje ga IP softveru za slanje. IP softver može da pogleda adresu i da otkrije da se odredište nalazi na njegovoj mreži, ali ipak mora na neki način utvrditi njegovu Ethernet adresu. Jedno rešenje je da se negde u sistem smesti konfiguraciona datoteka koja IP adrese preslikava u Ethernet adrese. Iako bi to radilo, ažuriranje odgovarajućih datoteka u organizacijama s hiljadama računara podložno je greškama i oduzima vreme.



Slika 5-62. Tri međusobno povezane mreže tipa /24: dve Ethernet mreže i FDDI prsten.

Bolje je ako računar 1 difuzno emituje paket na Ethernet pitajući: Ko ima IP adresu 192.31.65.5? Pitanje će stići do svakog računara na Ethernet mreži 192.31.65.0 i svaki će proveriti svoju IP adresu. Samo će računar 2 odgovoriti sa svojom Ethernet adresom *(E2)*. Na taj način, računar 1 saznaje da IP adresa 192.31.65.5 pripada računaru sa Ethernet adresom *E2*. Protokol kojim se ovakvo pitanje šalje i na njega dobija odgovor zove se protokol za razrešavanje adresa (engl. *Address Resolution Protocol, ARP*). Izvršava ga skoro svaki računar' na Internetu. ARP je definisan u RFC dokumentu 826.

Prednost protokola ARP nad konfiguracionim datotekama jeste jednostavnost njegovog korišćenja. Administrator sistema treba samo da svakom računaru dodeli IP adresu i da odluči o maskama podmreže. Sve ostalo radi ARP.

U ovoj fazi, IP softver na računaru 1 pravi Ethernet okvir adresiran na *E2,* smešta IP paket (adresiran na 192.31.65.5) u polje za korisničke podatke, i šalje ga na Ethernet. Ethernet kartica računara 2 otkriva ovaj okvir, utvrđuje da je za nju, grabi ga i izaziva prekid. Upravljački Ethernet program vadi IP paket iz polja za korisničke podatke i prosleđuje ga IP softveru, koji utvrđuje da je paket ispravno adresiran i obrađuje ga.

Efikasnost protokola ARP može se povećati raznim optimizacijama. Najpre, kada računar jednom izvrši protokol ARP, on čuva rezultat za slučaj da uskoro treba da ponovo stupi u vezu sa istim računarom. Tada će moći da pronađe uparene (mapirane) adrese u sopstvenom kesu i neće morati da ponovo postavlja pitanje svima. U mnogim slučajevima, računar 2 moraće da pošalje odgovor, što i njega prisiljava da izvrši ARP da bi odredio pošiljaočevu Ethernet adresu. To difuzno ARP emitovanje može se izbeći ako računar 1 u ARP paket uključi preslikavanje sopstvene adrese (IP u Ethernet). Kada difuzna ARP emisija stigne računaru 2, u ARP keš računara 2 unosi se par (192.31.65.5, El) za buduću upotrebu. U stvari, svi računali na Ethernetu mogu ovo preslikavanje uneti u svoj ARP keš.

Druga optimizacija bi bila da svaki računar pri podizanju difuzno emituje preslikavanje sopstvenih adresa. To emitovanje se obično izvodi u obliku ARP pretraživanja sopstvene IP adrese. Ne očekuje se odgovor, ali je usputni efekat emitovanja to što svaki računar u ARP keš upiše odrednicu. Ako odgovor (neočekivano) stigne, to znači da postoje dva računara kojima je dodeljena ista IP adresa. Računar koji to otkrije treba da o tome obavesti administratora sistema i da se ne uključuje na mrežu.

Da bi se omogućilo menjanje podataka o preslikavanju, na primer, kada se neka Ethernet kartica pokvari i zameni novom (s novom Ethernet adresom), odrednice u ARP kešu treba da se automatski brišu nakon nekoliko minuta.

Vratimo se na sliku 5-62. Ovoga puta, računar 1 želi da pošalje paket računaru 4 (192.31.63.8). ARP tu neće pomoći jer računar 4 neće videti difuznu emisiju računara 1 (usmerivači ne prosleđuju difuzne emisije na nivou Ethernet mreže). Postoje dva rešenja. Prvo, usmerivač Katedre za računarstvo može se konfigurisati da odgovara na ARP zahteve za mrežu 192.31.63.0 (i možda i za druge lokalne mreže). U tom slučaju, računar 1 će u ARP keš uneti odrednicu s parom (192.31.63.8, E3) i sav saobraćaj za računar 4 poslati lokalnom usmerivaču. To je zastupnički (posrednički)
ARP (engl. *proxy ARP).* Drago rešenje je da računar 1 odmah shvati da se odredište nalazi na udaljenoj mreži i da sav saobraćaj pošalje na podrazumevanu Ethernet adresu koja obrađuje udaljeni saobraćaj, u ovom slučaju, na adresu *E3.* Za to rešenje nije neophodno da usmerivač Katedre za računarstvo zna koje udaljene mreže opslužuje.

Ovako ili onako, računar 1 pakuje IP paket u polje za korisničke podatke Ethernet okvira adresiranog na *E3.* Kada usmerivač Katedre za računarstvo dobije Ethernet okvir, on iz njega izvlači IP paket i traži njegovu IP adresu u svojim tabelama za usmeravanje. Pri tome otkriva da pakete za mrežu 192.31.63.0 treba slati usmerivaču 192.31.60.7. Ukoliko već ne zna FDDI adresu tog usmerivača, on difuzno šalje ARP paket u prsten i saznaje daje adresa usmerivača u prstenu *F3.* On zatim umeće IP paket u polje za korisničke podatke FDDI okvira, adresira ga na *F3* i postavlja na prsten.

U usmerivaču Katedre za elektrotehniku, upravljački FDDI program vadi paket iz FDDI okvira i predaje ga IP softveru, koji utvrđuje da treba da ga pošalje na adresu 192.31.63.8. Ako se ta IP adresa ne nalazi u njegovom ARP kešu, on difuzno šalje ARP zahtev na lokalnu mrežu Katedre za elektrotehniku i saznaje da je odredišna adresa paketa *E6,* pa pravi Ethernet okvir, adresira ga na *E6,* stavlja paket u njegovo polje za korisničke podatke i šalje ga na Ethernet. Kada Ethernet okvir stigne računam 4, paket se vadi iz okvira i prosleđuje IP softveru na obradu.

Slanje paketa od računara 1 na udaljenu mrežu preko regionalne mreže radi suštinski na isti način, osim što sada tabele usmerivača Katedre za računarstvo nalažu da se za slanje upotrebi usmerivač regionalne mreže, čija je FDDI adresa *F2.*

RARP, BOOTP i DHCP

ARP rešava problem nalaženja Ethernet adrese kada se zna IP adresa. Ponekada treba rešiti obrnut problem: naći IP adresu kada je poznata Ethernet adresa. Taj problem, konkretno, nastaje kada se pušta u rad stanica koja nema čvrsti disk. Takav računar uobičajeno dobija binarnu sliku svog operativnog sistema sa udaljenog ser- vera datoteka. Ali, kako da sazna njegovu IP adresu?

Prvo predloženo rešenje bio je obrnuti ARP (engl. *Reverse Address Resolution Protocol, RARP),* koji je definisan u RFC dokumentu 903. Taj protokol omogućava radnim stanicama koje se podižu da difuzno emituju poruku: „Moja 48-bitna Ethernet adresa je 14.04.05.18.01.25. Da li iko zna moju IP adresu?“ RARP server prima taj zahtev, traži Ethernet adresu u svojim konfiguracionim datotekama i kao odgovor šalje odgovarajuću IP adresu.

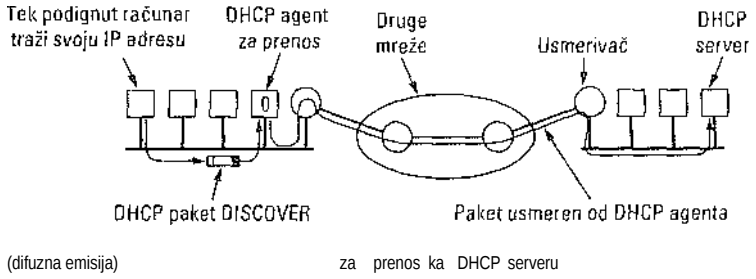
Bolje je upotrebiti RARP nego ugrađivati IP adresu u binarnu sliku jer se tako ista slika može koristiti za sve slične računaie. Kada bi IP adresa bila zapisana u binarnu sliku, svakom računam bi se morala slati zasebna slika.

Nedostatak protokola RARP je to što se za pristupanje RARP serveru koristi odredišna adresa sastavljena od samih jedinica (ograničeno difuzno emitovanje). Međutim, takve emisije usmerivači ne prosleđuju, pa RARP server mora da postoji na svakoj mreži. Da bi se taj problem zaobišao, smišljen je alternativni protokol za podizanje sistema (engl. *BOOTstraP, BOOTP).* Za razliku od protokola RARP, on koristi UDP poruke koje usmerivači prosleđuju. Stanicama bez čvrstih diskova on šalje i dopunske informacije, uključujući i IP adresu servera datoteka koji šalje binarnu sliku, IP adresu podrazumevanog usmerivača i masku podmreže koju će stanica koristiti. Protokol BOOTP opisan je u RFC dokumentima 951, 1048 i 1084.

BOOTP ima ozbiljan nedostatak: tabele preslikavanja IP adresa u Ethernet adrese moraju se ručno napraviti. Kada se u lokalnu mrežu doda nov računar, on ne može da koristi protokol BOOTP sve dok mu administrator ne dodeli IP adresu i par (Ethernet adresa, IP adresa) ne

unese ručno u konfiguracionu BOOTP tabelu. Da bi se izbegao ovaj postupak koji je podložan greškama, protokol BOOTP je proširen i preimenovan u protokol za dinamičko podešavanje računara (engl. *Dynamic Host Configuration Protocol, DHCP).* Protokol DHCP omogućava i ručno i automatsko dodeljivanje IP adresa. Opisan je u RFC dokumentima 2131 i 2132. On je u većini sistema zame- nio protokole RARP i BOOTP.

Slično protokolima RARP i BOOTP, protokol DHCP se takođe oslanja na specijalan server koji dodeljuje IP adrese onim računalima koji ih traže. Taj server ne mora da bude na istoj lokalnoj mreži kao i računar koji zahteva adresu. Pošto se DHCP ser- veru možda ne može pristupiti difuznim emitovanjem zahteva, u svakoj lokalnoj mreži mora da postoji DHCP agent za prenos (engl. *DHCP relay agent),* kao na slici 5-63.



Slika 5-63. Rad protokola DHCP.

Da bi saznao svoju IP adresu, računar koji se tek podigao emituje difuzno DHCP paket DISCOVER (otkrivanje). DHCP agent za prenos na njegovoj lokalnoj mreži presreće sve difuzne DHCP emisije. Kada otkrije DHCP paket DISCOVER, on ga usmereno šalje DHCP serveru koji se možda nalazi na udaljenoj mreži. DHCP agent za prenos treba da zna samo IP adresu DHCP servera.

Pri automatskom dodeljivanju IP adresa iz zajedničkog skladišta postavlja se pitanje koliki je rok korišćenja dodeljene adrese. Ako se računar isključi s mreže, a ne vrati svoju IP adresu DHCP serveru, ona se nepovratno gubi. Posle izvesnog vremena, na taj način bi se moglo izgubiti mnogo adresa. Da bi se to sprečilo, IP adrese se mogu dodeljivati za fiksni period tehnikom tzv. iznajmljivanja (engl. *leasing).* Računar mora od DHCP servera tražiti obnavljanje najma, pre nego što prethodni istekne. Ako to ne učini ili ako zahtev bude odbijen, računar ne može više koristiti prethodno dodeljenu IP adresu.

5.6.4 OSPF - unutrašnji protokol za mrežni prolaz

Završili smo proučavanje protokola za upravljanje na Internetu i vreme je da pređemo na drugu temu: usmeravanje na Internetu. Već smo više puta pomenuli da je Internet sačinjen od velikog broja autonomnih sistema. Svakim autonomnim sistemom upravlja druga organizacija i može unutar njega da koristi sopstveni algoritam za usmeravanje. Na primer, interne mreže kompanija *X, Y* i Z obično se vide kao tri autonomna sistema ako su sve tri priključene na Internet. Svaka od njih interno može da koristi drugačiji algoritam za usmeravanje. Bez obzira na to, kada postoje standardi barem za interno usmeravanje, jednostavnija je ugradnja interfejsa na granicama autonomnih sistema i moguće je više puta koristiti isti kod. U ovom odeljku bavićemo se usmeravanjem unutar autonomnog sistema, U

sledećem ćemo preći na usmeravanje između više autonomnih sistema. Algoritam za usmeravanje unutar autonomnog sistema zove se **unutrašnji protokol za mrežni prolaz** (engl. *interior gateway protocol),* a algoritam za usmeravanje između autonomnih sistema- **spoljni protokol za mrežni prolaz** (engl. *exterior gateway protocol*).

Prvobitni unutrašnji protokol za mrežni prolaz na Internetu bio je zasnovan na vektoru razdaljine (RIP), tj. na Belman-Fordovom algoritmu nasleđenom od ARPA- NET-a. On je dobro radio u malim autonomnim sistemima, ali sve lošije, kako su sistemi postajali veći. Talcođe je imao problema s približavanjem beskonačnosti i generalno loše vreme dostizanja ravnoteže, tako da je maja 1979. zamenjen protokolom zasnovanim na stanju veze. Godine 1988, IETF je počeo da radi na njegovom poboljšanju. Rezultat je bio **otvoren protokol najkraće putanje** (engl. *Open Shortest Path First, OSPF),* koji je 1990. postao standard. Danas ga proizvođači usmerivača uglavnom podržavaju, tako da je postao glavni unutrašnji protokol za mrežni prolaz. U nastavku ćemo u kratkim crtama opisati kako radi OSPF. Detalje potražite u RFC dokumentu 2328.

S obzirom na obimno iskustvo s protokolima za usmeravanje, radna grupa je novom protokolu postavila dugu listu zahteva. Prvo, algoritam je morao biti svakome dostupan (odatle ono „0“ u imenu OSPF, od engl. *open* - otvoren). Neko komercijalno rešenje, tj. vlasništvo određene kompanije, ne bi bilo prihvatljivo. Drugo, protokol je morao da podrži različite načine određivanja razdaljine: fizičko rastojanje, kašnjenje itd. Treće, algoritam je morao da radi dinamički - da se automatski i brzo prila- gođava promenama topologije sistema.

Četvrto, a to je novo za OSPF, algoritam je morao da podrži usmeravanje zasnovano na vrsti usluge. Protokol je morao da usmerava interaktivni saobraćaj jednim putem, a sav ostali drugim. U paketu IP protokola postoji polje *Vrsta usluge,* koje do sada nije koristio nijedan protokol. To polje je uključeno i u OSPF okvir, ali ga i dalje niko nije koristio, pa je kasnije uklonjeno.

Peto, a u vezi sa prethodnim, novi protokol je morao da ujednačava opterećenje, raspodeljujući ga na više linija. Raniji protokoli su slali pakete uglavnom najboljom putanjom. Drugu po redu „najbolju“ putanju uopšte nisu koristili. U mnogim slučajevima, raspodeljivanjem opterećenja na više linija postižu se bolje performanse.

Šesto, bilo je neophodno podržati hijerarhijske sisteme. Negde, 1988. godine, Internet je toliko porastao da se ni od jednog usmerivača nije očekivalo da zna čitavu njegovu topologiju. Trebalo je smisliti nov protokol za usmeravanje koji bi usmeri- vače u načelu oslobodio te obaveze.

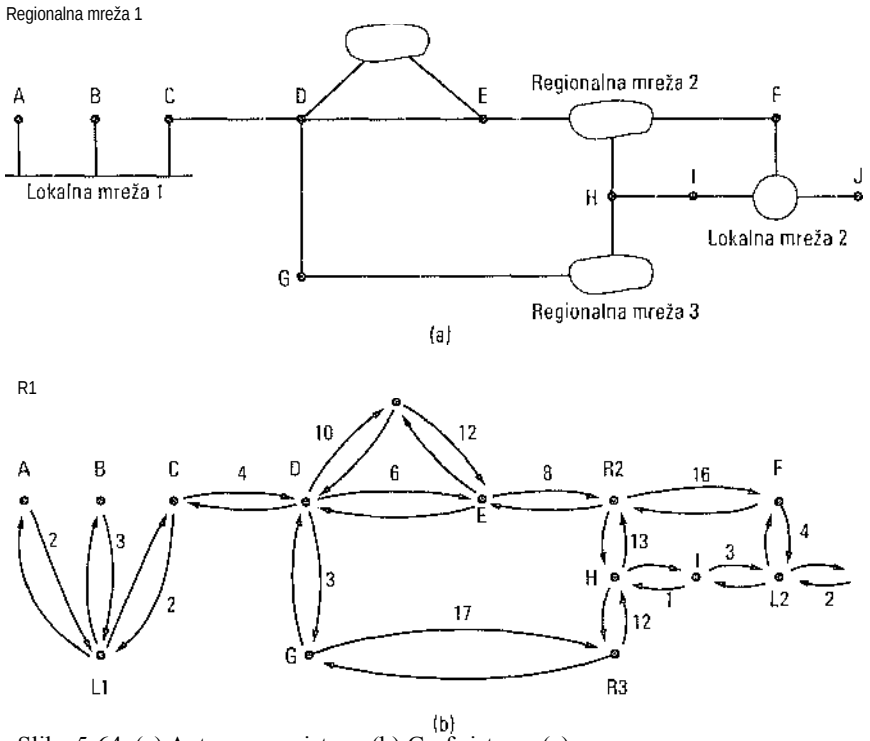
Sedmo, trebalo je predvideti izvesno obezbeđenje da besposleni studenti ne bi zatrpavali usmerivače pogrešnim podacima za usmeravanje. I na kraju, trebalo je obe- zbediti rad sa usmerivačima koji su na Internet povezani tunelom. Raniji protokoli to nisu dobro radili.

OSPF podržava tri vrste mreža i veza:

1. Linije od tačke do tačke između dva usmerivača.
2. Višepristupne mreže koje omogućavaju difuzno emitovanje (npr. većina lokalnih mreža).
3. Višepristupne mreže koje ne omogućavaju difuzno emitovanje (npr. većina regionalnih mreža koje rade s komutiranjem paketa).

Višepristupna (engl. *multiaccess*) mreža može da ima više usmerivača, koji međusobno

mogu da direktno komuniciraju. To svojstvo imaju sve lokalne i sve regionalne mreže. Slika 5-64(a) prikazuje autonoman sistem koji sadrži sve tri vrste mreža. Obratite pažnju na to da u protokolu OSPF računari nemaju nikakvu ulogu.



Slika 5-64. (a) Autonoman sistem, (b) Graf sistema (a).

OSPF radi tako što apstrahuje skup konkretnih mreža, usmerivača i linija u usme- ren graf u kome se svakom luku dodeljuje „težina“ (rastojanje, kašnjenje itd.). Protokol zatim izračunava najkrac'u putanju zasnovanu na težinama lukova. Serijsku vezu između dva usmerivača predstavlja par lukova, po jedan u svakom od dva smera. Njihove težine se mogu razlikovati. Višepristupna mreža se predstavlja jednim čvorom za mrežu i po jednim čvorom za svaki usmerivač. Lukovi od mrežnog čvora do usmerivača imaju težinu 0 i zato nisu prikazani na grafu.

Slika 5-46(b) prikazu je graf koji odgovara mreži sa slike 5-46(a). Ako nije drugačije označeno, težine su simetrične. OSPF u suštini na osnovu stvarne mreže pravi graf sličan prikazanom i onda izračunava najkraće putanje između usmerivača u svim kombinacijama.

Mnogi autonomni sistemi na Internetu i sami su veliki, pa je upravljanje njima složeno. OSPF im omogućava da se podele na oblasti (engl. *areas),* gde oblast predstavlja jednu mrežu ili skup susednih mreža. Oblasti ne treba da se prekrivaju, ali ni sve (npr. neki usmerivači) ne mora da uđe u neku oblast. Oblast predstavlja proširenje pojma podmreže. Topologija oblasti i dragi detalji nisu vidljivi izvan nje.

Svaki autonoman sistem ima oblast okosnice (engl. *backbone area)* - oblast 0. Sa okosnicom su povezane sve oblasti, možda pomoću tunela, tako da se preko okosnice može

stići iz bilo koje oblasti u bilo koju drugu oblast. Tunel je na grafu predstavljen lukom i ima svoju težinu. Svaki usmerivač koji je povezan s dve ili više oblasti deo je okosnice. Kao kod dragih oblasti, i topologija okosnice nije vidljiva izvan nje.

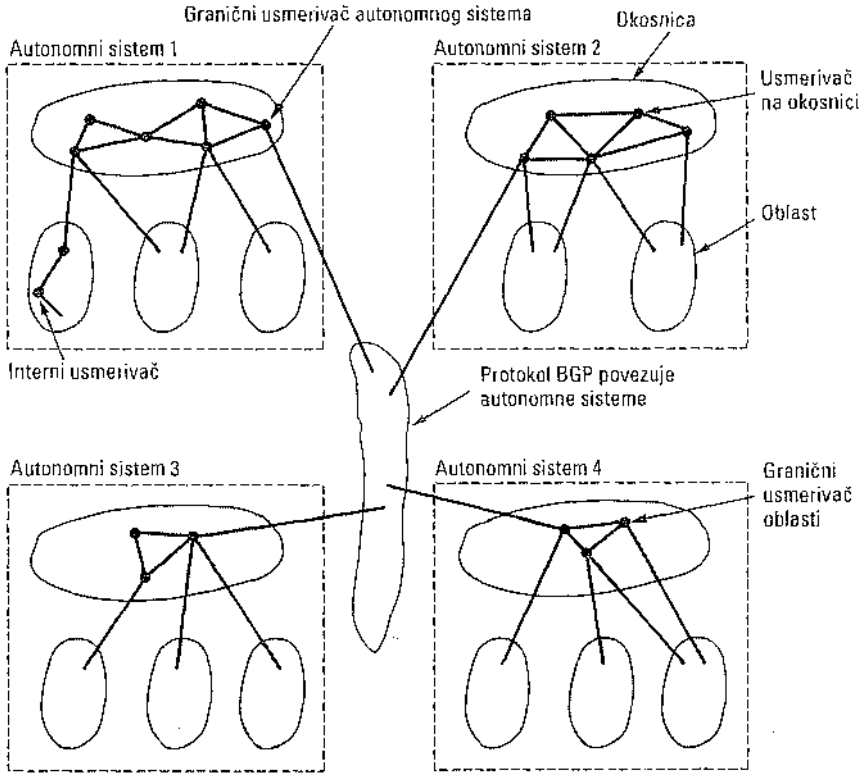
Unutar određene oblasti, svaki usmerivač ima istu bazu podataka sa stanjem veza i izvršava isti algoritam najkraće putanje. Glavni im je posao izračunavanje najkraće putanje do svih dragih usmerivača u oblasti, uključujući i usmerivač koji je povezan sa okosnicom, a barem jedan takav mora postojati u svakoj oblasti. Usmerivač koji se povezuje s dve oblasti mora imati baze podataka za obe i mora da za svaku izvršava zasebnu kopiju algoritma najkraće putanje.

Tokom normalnog rada, zahtevaju se tri vrste putanja: unutar oblasti, između oblasti i unutar autonomnog sistema. Najlakše je izračunati putanje unutar jedne oblasti, pošto izvorišni usmerivač već zna najkraći put do odredišnog usmerivača. Među- oblasno usmeravanje uvek se odvija u tri koraka: od izvorišta do okosnice, okosnicom do odredišne oblasti i kroz odredišnu oblast do samog odredišta. Ovakav algoritam nameće sistemu OSPF konfiguraciju zvezde - okosnica u centru, a oblasti na periferiji. Paketi se od izvorišta do odredišta šalju netaknuti. Ne kapsuliraju se i ne sprovode tunelom, osim ako treba ući u oblast koja je sa okosnicom povezana isključivo tunelom. Slika 5-65 prikazuje deo Interneta sa autonomnim sistemima i oblastima.

OSPF razlikuje četiri klase usmerivača:

1. Interni usmerivači u potpunosti pripadaju jednoj oblasti.
2. Granični usmerivači oblasti povezuju dve ili više oblasti.
3. Usmerivači okosnice nalaze se na okosnici.
4. Granični usmerivači autonomnog sistema komuniciraju sa usmerivačima drugih autonomnih sistema.

Navedene klase se mogu preklapati. Na primer, svi granični usmerivači automatski su deo okosnice. Osim toga, usmerivač na okosnici koji ne pripada nijednoj oblasti, takođe je interni usmerivač. Usmerivači sve četiri klase prikazani su na slici 5-65.



Slika 5-65. Međusobni odnosi između autonomnih sistema, okosnica i oblasti u sistemu OSPF.

Kada se usmerivač uključi, on šalje pozdravnu (HELLO) poruku na sve svoje linije od tačke do tačke, kao i svim usmerivačima na povezanim lokalnim mrežama. Za regionalne mreže gaje potrebno konfigurisati da bi znao s kime da stupi u vezu. Iz odgovora na ovu poruku svaki usmerivač saznaje ko su mu susedi. Usmerivači na istoj lokalnoj mreži susedi sujedan drugom.

OSPF radi tako što razmenjuje informacije između kontrolnih usmerivača, što nije isto kao kada bi ih razmenjivali susedi. U stvari, nije efikasno da svaki usmerivač u lokalnoj mreži komunicira sa svakim drugim usmerivačem u istoj lokalnoj mreži. Zato se bira tzv. **namenski** usmerivač (engl. *designated router).* On je istovremeno kontrolni usmerivač (engl. *adjacent router)* za sve druge usmerivače na istoj lokalnoj mreži i razmenjuje informacije s njima. Susedni usmerivači koji nisu kontrolni, ne razmenjuju međusobno informacije. Rezervni namenski usmerivač uvek je ažuran i spreman da uskoči za slučaj da glavni namenski usmerivač otkaže i mora odmah da se zameni.

Tokom normalnog rada, svaki usmerivač periodično plavi svoje kontrolne usmeri- vače porukama LINK STATE UPDATE. Te poruka sadrže stanje usmerivača i težine linija iz topološke baze podataka. One se potvrđuju da bi se obezbedila pouzdanost. Svaka poruka ima redni broj tako da usmerivač može utvrditi da li je poruka stvarno nova. Osim periodično, usmerivači ovakve poruke šalju i svaki put kada se linija uključi/isključi ili kada

se težina neke linije promeni.

Poruke DATABASE DESCRIPTION sadrže redne brojeve svih odrednica sa stanjem veze koje trenutno ima pošiljalac. Poredeći sopstvene vrednosti s dobijenim vrednostima, primalac može da utvrdi koje su novije. Poruke ovakve vrste razmenju- ju se prilikom uspostavljanja linija.

Svaki partner od dragoga može da zahteva informacije o stanju veze porukom LINK STATE REQUEST. Poruka prisiljava svaki par kontrolnih usmerivača da međusobno provere svežinu svojih podataka, pa se na taj način nove informacije šire po oblasti. Sve ove poruke šalju se kao sirovi IP paketi. Poruke pet pomenutih vrsta zajedno su navedene na slici .5-66.

Vrsta poruke	Opis
Hello	Koristi se za otkrivanje suseda
Link state update	Pruža podatke o težini veza između pošiljaoca i njegovih suseda
Link state ack	Potvrđuje poruku Link state update
Database description	Objavljuje svežinu informacija koje ima pošiljalac
Link state request	Traži informacije od partnera

Slika 5-66. Pet vrsta OSPF poruka.

Najzad možemo da sastavimo celu sliku. Svaki usmerivač plavljenjem obaveštava sve drage usmerivače u oblasti o svojim susedima i težini linija. Te informacije omogućavaju svakom usmerivaču da konstruiše graf svoje (svojih) oblasti i da izračuna najkraće putanje. To isto radi i oblast okosnice. Osim toga, usmerivači okosnice primaju informacije od graničnih usmerivača oblasti da bi mogli da izračunaju najbolju putanju od svakog usmerivača na okosnici do svakog drugog usmerivača. Ta informacija se vraća graničnim usmerivačima koji je objavljuju unutar svojih oblasti. Na osnovu ovih podataka, usmerivač koji želi da pošalje poruku u dragu oblast može da izabere najbolji usmerivač za izlazak na okosnicu.

5.6.5 BGP - spoljni protokol za mrežni prolaz

Za usmeravanje unutar jedinstvenog autonomnog sistema, preporučen je protokol OSPF (iako to nije i jedini protokol koji se za ovo koristi). Za usmeravanje između autonomnih sistema koristi se drugačiji protokol, tzv. **protokol graničnog mrežnog prolaza** (engl. *Border Gateway Protocol, BGP)*. Tu je neophodan drugačiji protokol jer se namene unutrašnjeg i spoljnog protokola za mrežni prolaz razlikuju. Unutrašnji protokol treba samo da što efikasnije prenosi pakete od izvorišta do odredišta. On ne treba da se zamara politikom.

Usmerivači koji rade sa spoljnim protokolom za mrežni prolaz moraju, međutim, da se poprilično pozabave i politikom (Metz, 2001). Na primer, korporacijski autonomni sistem može da ima potrebu da šalje pakete bilo kojoj lokaciji na Internetu i da ih od njih prima. Međutim, može se pojaviti otpor propuštanju paketa začetih u stranom autonomnom sistemu i namenjenih drugom stranom autonomnom sistemu, čak i onda kada se naš autonomni sistem nalazi na najboljoj putanji između ta dva sistema („To nije naš problem, već njihov“). Sasvim druga priča je tranzitni saobraćaj za su- sede ili čak i za strane autonomne sisteme

koji su voljni da za uslugu plate. Telefonske kompanije, na primer, vrlo rado pristaju da služe kao nosilac podataka za svoje mušterije, ali ne i za druge. Svi spoljni protokoli za mrežne prolaze, a protokol BGP posebno, namenski su projektovani da prihvate vrlo različita pravila usmeravanja koja se mogu pojaviti u saobraćaju između autonomnih sistema.

Pravila obično obuhvataju političke, bezbednosne i ekonomske elemente. Evo nekoliko ograničenja u pogledu usmeravanja, koja će vam to bliže objasniti:

1. Ne prolaziti kroz određene autonomne sisteme.
2. Nikada ne stavljati Irak na putanju koja počinje u Pentagonu.
3. Ne koristiti SAD za saobraćaj između kanadskih država Britanske Kolumbije i Ontarija.
4. Putanju usmeriti kroz Albaniju samo ako ne postoji drugi način.
5. Putanju koja počinje ili se završava u IBM-u ne usmeravati kroz Microsoft.

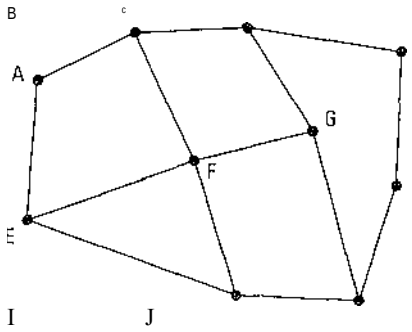
Pravila se najčešće ručno unose u svaki BGP usmerivač (ili poluautomatski, pomoću skripta). Ona nisu deo samog protokola.

S gledišta BGP usmerivača, svet se sastoji od autonomnih oblasti povezanih linijama. Dva autonomna sistema smatraju se povezanim ako postoji linija između njihovih graničnih usmerivača. Pošto protokol BGP ima poseban značaj u tranzitnom saobraćaju, mreže se svrstavaju u tri kategorije. Prvu kategoriju čine mreže povezane u jednoj tački (engl. *stub networks)* sa BGP grafom. One se ne mogu koristiti za tranzit jer nikuda ne vode. Dragu kategoriju čine mreže povezane u više tačaka (engl. *multiconnected networks)*. Kroz njih je tranzit moguć, osim kada ga odbiju. Treća je kategorija tranzitnih mreža (engl. *transit networks),* koje su, kao okosnice, spremne da prenose pakete za treća lica, možda uz izvesna ograničenja i najčešće uz naknadu.

Parovi BGP usmerivača međusobno komuniciraju uspostavljanjem TCP veze. Takav rad obezbeđuje pouzdanost i skriva sve detalje mreža kroz koje paketi prolaze.

Protokol BGP je u suštini zasnovan na vektoru razdaljine, ali se veoma razlikuje od drugih takvih protokola, npr. od protokola RIP Umesto da održava samo podatke o ukupnoj težini linije do svakog odredišta, BGP usmerivač vodi računa i o samoj putanji. Isto tako, umesto da susedima periodično šalje procenjene težine linija do svakog odredišta, BGP usmerivač im direktno ukazuje na liniju koju koristi.

Razmotrimo primer sa BGP usmerivačima prikazan na slici 5-67(a). Pogledajmo, konkretno, tabelu usmerivača *F*. Pretpostavimo da on do *D* koristi putanju *FGCD*. Kada mu susedi šalju informacije o usmeravanju, oni mu daju potpune putanje, kao na slici 5-67(b) (zbog jednostavnosti, prikazane su samo putanje do *D)*.



Informacije o usmerivaču D koje usmerivač F dobija od svojih suseda

Od B: "Ja koristim BCD"
Od G: "Ja koristim GOD"
Od I: "Ja koristim IFGCD"
Od E: "Ja koristim EFGCD"

(a) (b)

Slika 5-67. (a) Skup BGP usmerivača. (b) Informacije poslate usmerivaču *F*.

Pošto mu od suseda stignu sve putanje, *F* među njima traži najbolju. On brzo odbacuje putanje dobijene od / i *E,* pošto one prolaze kroz njega. Ostaje mu da bira između putanja dobijenih od *B* i *G.* Svaki BGP usmerivač ima modul koji ispituje putanje do određenog odredišta izračunavajući njihove „dužine". Svakoj putanji koja narušava postavljena pravila automatski se dodeljuje beskonačna dužina. Usmerivač na kraju prihvata najkraću putanju. Funkcija za izračunavanje dužine putanje nije deo protokola BGP; njen izbor je ostavljen administratoru sistema.

BGP lako rešava problem približavanja beskonačnosti koji toliko muči druge algoritme zasnovane na vektoru razdaljine. Pretpostavimo, na primer, da usmerivač *G* otkaže ili da se prekine linija *FG.* Usmerivač *F* tada prima informacije o putanjama od svoja tri preostala suseda. Te putanje su *BCD, IFGCD* i *EFGCD.* Pošto dve poslednje putanje automatski otpadaju jer prolaze kroz njega, usmerivač *F,* kao novu, bira putanju *FBCD.* Drugi algoritmi zasnovani na vektoru razdaljine često biraju pogrešnu putanju jer ne mogu da razlikuju nezavisne od zavisnih putanja koje im nude njihovi susedi. Definiciju protokola BGP naći ćete u RFC dokumentima 1771 i 1774.

5.6.6 Višesmerno emitovanje na Internetu

Normalna IP komunikacija obično se odvija između jednog pošiljaoca i jednog primaoca. Međutim, u nekim primenama je neophodno da proces istovremeno pošalje pakete na više odredišta. Primeri su ažuriranje replikovanih distribuiranih baza podataka, slanje izveštaja o akcijama velikom broju berzanskih posrednika i održavanje digitalnih konferencijskih telefonskih razgovora (između više učesnika istovremeno).

IP podržava višesmerno emitovanje, uz upotrebu adresa klase D. Svaka adresa klase D iđentifikuje grupu računara. Grupe se identifikuju pomoću 28 bitova, tako da istovremeno može postojati preko 2.50 miliona grupa. Kada proces pošalje paket na adresu klase D, paket

u načelu dobijaju svi članovi grupe, ali za to nema garancije. Neki članovi mogu da ostanu bez paketa.

Podržane su dve vrste grupnih adresa: stalne i privremene. Stalna grupa je uvek tu - ona se ne mora uvek ponovo uspostavljati. Svaka stalna grupa ima i stalnu adresu. Evo nekih primera:

224.0. 0.1 Svi sistemi na lokalnoj mreži
224.0. 0.2 Svi usmerivači na lokalnoj mreži
224.0. 0.5 Svi OSPF usmerivači na lokalnoj mreži
224.0. 0.6 Svi namenski OSPF usmerivači na lokalnoj mreži

Privremene grupe se pre korišćenja moraju uspostavljati. Proces može da od svog računara zatraži dozvolu da se pridruži grupi ili da iz nje izađe. Kada i poslednji proces na računam napusti grupu, ta grupa više ne postoji na tom računaru. Svaki računar vodi računa o grupama kojima pripadaju njegovi procesi.

Višesmerno emitovanje se realizuje pomoću specijalnih usmerivača koji mogu, ali i ne moraju, biti locirani uz standardne usmerivače. Približno svakog minuta, svaki usmerivač za višesmerno emitovanje šalje hardversku poruku (dakle, u sloju veze podataka) svim računalima na svojoj lokalnoj mreži (adrese 224.0.0.1), zahtevajući izveštaj u grupama kojima pripadaju njihovi procesi. Svaki računar odgovara svim adresama klase D za koje je zainteresovan.

Ovi zahtevi i odgovori razmenjuju se pomoću protokola za rad s grupama na Internetu (engl. *Internet Group Management Protocol*, *IGMP),* koji izdaleka liči na protokol ICMR On ima samo dve vrste paketa: zahteve i odgovore, svaki s jednostavnim fiksnim formatom koji u prvoj reči polja za koristan teret sadrže neke upravljačke informacije, a u drugoj adresu klase D. Opisan je u RFC dokumentu 1112.

Za usmeravanje na više adresa koristi se razgranato stablo. Svaki višesmerni usmerivač razmenjuje informacije sa svojim susedima pomoću modifikovanog protokola zasnovanog na vektoru razdaljine da bi svaki od njih za svaku grupu kon- struisao razgranato stablo koje obuhvata sve njene članove. Stablo se optimizuje na različite načine da bi se iz njega uklonili usmerivači i mreže koji nisu zainteresovani za određene grupe. Protokol intenzivno koristi tunele da bi zaobišao čvorove koji ne pripadaju stablu.

5.6.7 IP komuniciranje s pokretnim računarima

Mnogi korisnici Interneta imaju prenosive računate i žele da budu priključeni na Internet kada odu u posetu udaljenoj lokaciji, pa čak i tokom putovanja do nje. Nažalost, postojeći sistem IP ađresiranja otežava priključenje pokretnih korisnika. U ovom odeljku upoznaćemo se s problemom i načinima njegovog rešavanja, a iscrpniji opis možete da nađete kod Perkinsa (1998a).

Stvarni krivac je sam sistem ađresiranja. Svaka IP adresa sadrži broj mreže i broj računara. Razmotrimo, naprimer, računar čija je adresa 160.80.40.20/16. Broj 160.80 predstavlja broj mreže (decimalno 8272); 40.20 je broj računara (decimalno 10269). U tabelama usmerivača širom sveta navedeno je koju liniju treba upotrebiti za stizanje do mreže 160.80. Kad god im stigne paket sa odredišnom IP adresom oblika 160.80.xxx.yyy, oni ga šalju tom linijom.

Ako se taj računar odjednom pojavi na nekom udaljenom mestu, paketi za njega i dalje će se upućivati njegovoj matičnoj lokalnoj mreži (ili usmerivaču). Korisnik više neće dobijati e-poštu itd. Bilo bi nepraktično da se računam dodeli nova adresa jer bi o toj promeni trebalo

obavestiti veliki broj ljudi, programa i baza podataka.

Drugi moguć pristup bio bi da usmerivači koriste potpunu IP adresu, a ne samo njen mrežni deo. To bi, međutim, zahtevalo da usmerivači imaju tabele s milionima odrednica, što bi astronomski povećačo troškove Interneta.

Kada su korisnici počeli da zahtevaju mogućnost da svoje prenosive računare po vezu na Internet ma gde se nalazili, IETF je osnovao Radnu grupu za traženje rešenja. Radna grupa je brzo sastavila spisak uslova koje treba da zadovolji svako rešenje. Glavni su bili:

1. Svaki pokretni računar, ma gde se nalazio, i dalje može da koristi svoju matičnu IP adresu.
2. Softver fiksnih umreženih računara ne sme da se menja.
3. Softver i tabele usmerivača ne smeju da se menjaju.
4. Paketi za pokretne korisnike uglavnom treba da slede direktnu putanju.
5. Kada je pokretni računar na matičnoj lokciji, sve treba da teče normalno.

Izabrano je rešenje koje je već prikazano u odeljku 5.2.8. Ukratko, svaka lokacija koja korisnicima želi da dopusti lutanje mora da napravi domaćeg agenta (engl. *home agent)*. Svaka lokacija koja želi da prima goste, mora da ima agenta za strance (engl. *foreign agent)*. Kada se pokretni računar pojavi na stranoj lokaciji, on stupa u vezu sa agentom za strance i registruje se kod njega. Agent za strance stupa u vezu s korisnikovim domaćim agentom i daje mu privremenu adresu (engl. *care-of address),* obično sopstvenu IP adresu.

Kada se paket uputi u korisnikovu matičnu lokalnu mrežu, on dolazi preko određenog usmerivača povezanog s njom. Usmerivač pokušava da pronađe računar na uobičajen način - đifuzno šaljući ARP paket s pitanjem: Koja je Ethernet adresa računara 160.80.40.20? Agent za domaće računare na ovaj zahtev odgovara svojom sopstvenom Ethernet adresom. Usmerivač tada domaćem agentu šalje paket koji je namenjen računam 160.80.40.20. Ovaj ga posredstvom tunela upućuje na privremenu adresu kapsu- lirajući ga u polje za korisničke podatke IP paketa adresiranog na agenta za strance. Agent za strance vadi paket iz kapsule i isporučuje ga na adresu sloja veze podataka pokretnog računara. Osim toga, domaći agent daje pošiljaocu privremenu adresu, tako da se naredni paketi odmah mogu slati tunelom agentu za strance. Takvo rešenje ispunjava sve napred navedene zahteve.

Treba možda pomenuti jedan detalj. Dok pokretni računar menja mesto, usmerivač verovatno ima u kešu njegovu (uskoro nevažeću) Ethernet adresu. Zamena te adrese adresom domaćeg agenta izvodi se trikom zvanim dopunska usluga ARP-a (engl. *gratuitous ARP)*. To je specijalna, netražena poruka upućena usmerivaču, koja menja specifičnu odrednicu u njegovom kešu - u ovom slučaju, odrednicu računara koji upravo kreće na put. Kada se računar kasnije vrati kući, odrednica se ponovo ažurira istim trikom.

Opisano rešenje ne zabranjuje da pokretni računar bude sopstveni agent za strance, ali bi to radilo samo ako bi mobilni računar (kao agent za strance) bio logički povezan na Internet na svojoj trenutnoj lokaciji. Isto tako, takav „agent za strance“ morao bi da dobije privremenu IP adresu (koji bi davao dragim pokretnim korisnicima), a ta IP adresa morala bi pripadati lokalnoj mreži na koju je trenutno priključen.

Rešenje koje je IETF ponudio za pokretne računare razrešava i brojne drage pro bleme koje nismo pomenuli. Na primer, kako su locirani agenti? Rešenje je nađeno u tome da svaki agent periodično difuzno emituje svoju adresu i vrstu usluga koje je voljan da pruži (na primer, kao domaći agent, kao agent za strance ili oboje). Kada se pokretni računar negde

zaustavi, može sve da sazna slušajući ove neusmerene emisije, tzv. oglase (engl. *advertisements)*. Alternativno, on može i sam da difuzno pošalje paket oglašavajući svoj dolazak, u nadi da će na njega odgovoriti lokalni agent za strance.

Trebalo je rešiti i problem neuljudnih računara koji jednostavno napuste društvo bez pozdrava. Rešenje je nađeno u tome da se registracija ograniči 11a fiksan vremenski period. Ako se ona periodično ne obnavlja, automatski se ukida, i agent za strance briše takve računare iz svojih tabela.

Problem je i bezbednost. Kada domaći agent dobije ljubaznu poruku da sve pakete za Jelenu upućuje na određenu IP adresu, bolje je da se tome ne povinuje osim ako je siguran da je taj zahtev poslala glavom Jelena, a ne neko dragi u njeno ime. Za takve poruke koriste se protokoli za šifrovanu proveru identiteta, o kojima ćemo govoriti u
8. poglavlju.

Poslednje 0 čemu je razmišljala Radna IETF grupa bio je stepen polcretnosti računara. Zamislite avion sa ugrađenom Ethernet mrežom koju koriste navigacioni i upravljački računati u avionu. Na toj mreži je i standardni usinerivač koji radio-vezom komunicira sa zemaljskim Internetom. Jednog lepog dana, neki promućurni preduzet- nik instalira Ethernet priključke u naslone za ruke svih sedišta u avionu kako bi lokalnu mrežu mogli da koriste i putnici s prenosivim računarima.

Ovde se jasno izdvajaju dve vrste pokretnih računara: avionski računari koji su stacionarni u odnosu na Ethernet mrežu i putnički računari koji su u odnosu na nju pokretni. Osim toga, avionski usmerivač je pokretan u odnosu na usmerivače na tlu. Kretanje kroz sistem koji se i sam kreće, može se rešiti rekurzivnom upotrebom tunela.

5.6.8 IPv6

Dok sistemi CIDR i NAT mogu potrajati još neko vreme, svakom je jasno da su IP sistemu u današnjem obliku (IPv4) odbrojani dani. Uz opisane tehničke probleme, u pozadini vreba još jedna opasnost. Na samom početku, Internet su uglavnom koristili univerziteti, industrija koja se bavila savremenim tehnologijama i Američka vlada (naročito Ministarstvo odbrane). Sa eksplozijom zanimanja za Internet koja je počela sredinom devedesetih godina, njega su počele da koriste različite grupe ljudi i to grupe čiji su se zahtevi razlikovali. Najpre, tu su brojni vlasnici bežičnih prenosivih računara koji žele da su u stalnoj vezi sa svojom matičnom bazom. Zatim, kako se računari, komunikacija i zabava međusobno sve više približavaju, blizu je i trenutak kada će svaki telefon i televizor na svetu postati po jedan čvor Interneta koji uporedo s milijardama sličnih čvorova preuzima zahtevane audio i video sekvence. U takvoj situaciji, jasno je da protokol IP mora da se poboljša i da postane fleksibilniji.

Naslućujući ove probleme, IETF je 1990. počeo da razvija novu verziju protokola IP, takvu kojoj nikada neće ponestati adresa, koja će resiti brojne probleme i istovremeno biti fleksibilnija i efikasnija od prethodne. Trebalo je da nova verzija ispuni sle- deće zahteve:

1. Da podrži milijarde računara, čak i uz neefikasno korišćenje adresnog prostora.
2. Da smanji veličinu tabela za usmeravanje.
3. Da bude jednostavnija kako bi usmerivačima omogućila da brže obrađuju pakete.
4. Da bude bezbednija (identifikovanje i privatnost) od postojeće verzije.
5. Da obrati više pažnje na vrstu usluge, naročito na prenos u realnom vremenu.

6. Da podrži višesmerno emitovanje, omogućujući defmisanje opsega adresa.
7. Da računarima omogući kretanje bez menjanja adrese.
8. Da ostavi mogućnost za buduće razvijanje protokola.
9. Da omogući uporedno korišćenje stare i nove verzije tokom dužeg vremena.

Da bi mogao da napravi protokol koji bi ispunio sve pobrojane zahteve, IETF je u RFC dokumentu 1550 objavio poziv za raspravu i podnošenje predloga. Stigao je 21 odgovor, ali je stvarnih predloga bilo manje. Decembra 1992, na stolu je ostalo sedam ozbiljnijih predloga, počev od toga da se postojeći IP protokol samo malo zakrpi, pa do toga da se on sasvim odbaci i napiše potpuno nov protokol.

Jedan od predloga bio je da se TCP izvršava preko protokola CLNP, koji bi pomoću svojih 160-bitnih adresa zauvek rešio problem njihovog manjka i istovremeno objedinio dva glavna protokola mrežnog sloja. Međutim, mnogi su taj predlog shvatili kao priznanje da je i u OSI svetu nešto urađeno kako treba, a takav stav se u Internet krugovima smatra političkim zastranjivanjem. CLNP je pravljen po ugledu na IP, tako da se ti protokoli malo razlikuju. U stvari, protokol koji je na kraju izabran razlikuje se od protokola IP mnogo više nego protokol CLNP. Drugi argument protiv protokola CLNP bio je njegova slaba podrška vrstama usluga, nešto što je bilo neophodno za efikasan prenos multimedije.

Tri bolja predloga objavljena su u časopisu *IEEE Network* (Deering, 1993; Francis, 1993; Katz i Ford, 1993). Posle mnogo rasprave, prepravljanja i nadmetanja, odabrana je izmenjena kombinovana verzija Deeringovog i Francisovog predloga - **dopunjeni jednostavan protokol za Internet** (engl. *Simple Internet Protocol Plus, SIPP)* i data joj je oznaka **IPv6.**

IPv6 odgovara na postavljene zahteve prilično uspešno. U njemu su zadržane dobre osobine protokola IP, one loše su odbačene ili potisnute u drugi plan a - gde je trebalo - dodate su i nove. IPv6 u načelu nije kompatibilan s verzijom IPv4, ali jeste s drugim pomoćnim Internet protokolima, uključujući TCP, UDP, ICMP, IGMP, OSPF, BGP i DNS, ponekada uz male izmene (da bi se moglo raditi s dužim adresama). U nastavku opisujemo glavne osobine protokola IPv6. Više informacija o njemu možete da nađete u RFC dokumentima od 2460 do 2466.

Prvo i najvažnije, IPv6 radi s dužim adresama nego protokol IPv4. One su dugačke 16 bajtova, što rešava jedan od postavljenih zadataka: obezbeđuje praktično neograničen broj Internet adresa. O adresama ćemo ubrzo reći nešto više.

Drugo glavno poboljšanje koje je uveo protokol IPv6 odnosi se na uprošćenje zaglavlja. Ono sadrži samo sedam polja (u odnosu na 13 kod IPv4). Ta promena omogućava usmerivačima da brže obrađuju pakete, i tako povećaju njihov protok i smanje kašnjenje. I o zaglavlju ćemo ubrzo govoriti detaljnije.

Treće veće poboljšanje je bolja podrška opcijama. To je bilo neophodno kada je uvedeno novo zaglavlje, pošto su polja koja su ranije bila obavezna sada postala opciona. Osim toga, drugačiji je i način na koji se opcije predstavljaju, što usmerivačima omogućava da preskoče opcije koje nisu namenjene njima. Ta osobina ubrzava obradu paketa.

Četvrto područje koje je u protokolu IPv6 znatno unapređeno jeste bezbednost. IETF je imao u vidu punu policu novinskih članaka o 12-godišnjacima koji pomoću svojih PC računat a provaljuju u banke i vojne baze širom Interneta. Preovladalo je čvrsto ubeđenje da u pogledu toga treba nešto učiniti. Zbog toga su provera identiteta i privatnost ključne osobine novog IP protokola. Te osobine su kasnije dodate i u IPv4, tako da sada - na području

bezbednosti - nema velikih razlika između stare i nove verzije protokola.

Najzad, kvalitetu usluge posvećena je dužna pažnja. U prošlosti su primenjivana mnoga polovična rešenja, ali danas, kada se preko Interneta sve više prenosi multimedijski sadržaj, s time se ne treba šaliti.

Osnovno zaglavlje DPv6 paketa

Zaglavlje IPv6 paketa prikazano je na slici 5-68. Polje *Verzija* ima uvek vrednost 6 za IPv6 (4 za IPv4). Tokom perioda prelaska s protokola IPv4 na protokol IPv6, usmerivači će moći da ispituju ovo polje i da tako utvrde kakav paket imaju. Pomeni- mo uzgred da ova provera troši nekoliko instrukcija na kritičnoj putanji, pa će mnoge realizacije verovatno pokušati da je zaobiđu koristeći za razlikovanje paketa neko polje u zaglavlju sloja veze. Na taj način, paketi se direktno mogu proslediti odgovarajućem programu za obradu u mrežnom sloju. Međutim, ako sloj veze podataka počne da vodi brigu o vrstama paketa, to potpuno ruši princip da sloj ne treba ništa da zna

0 značenju bitova koje mu isporučuje viši sloj. Rasprava između onih koji misle da sve treba uraditi kako treba i onih koji smatraju da je važnije postići da protokol radi brzo, verovatno će biti dugotrajna i žestoka.

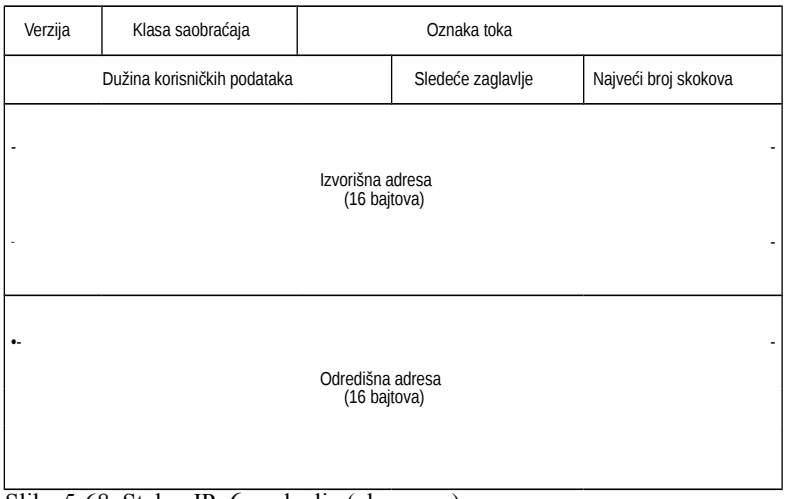
Polje *Klasa saobraćaja* koristi se za prepoznavanje paketa s različitim zahtevima u pogledu isporuke u realnom vremenu. Takvo polje je postojalo u protokolu IP od samog početka, ali su ga usmerivači retko koristili. Upravo se sprovode eksperimenti da bi se utvrdilo koliko ono pomaže pri isporuci multimedije.

I polje *Oznaka toka* još uvek je eksperimentalno, ali će se koristiti za uspostavljanje pseudoveze između izvorišta i odredišta za koju se mogu definisati određena svojstva

1 zahtevi. Na primer, tok paketa od izvesnog procesa na izvorišnom računaru do izvesnog procesa na odredišnom računam može da ima posebne zahteve u pogledu kašnjenja paketa i zato se za njega mora rezervisati propusni opseg. Takav tok se može unapred podesiti, pri čemu dobija određeni identifikator. Kada se pojavi paket sa *Oznakom toka* različitom od nule, svi usmerivači mogu da ga potraže u svojim internim tabelama da bi utvrdili kako s njim treba postupati. U stvari, takvi tokovi predstavljaju pokušaj da se objedine dva dobra: fleksibilnost datagramske podmreže i garantovanost isporuke u podmreži s virtuelnim kolima.

^ 32bita ,,

1 i i i i i i i I i i i i i i i I i i i i i i i I i i i i i i i I



Slika 5-68. Stalno IPv6 zaglavlje (obavezno).

Svaki tok se označava izvorišnom adresom, odredišnom adresom i brojem toka, tako da između istog para IP adresa istovremeno može da bude aktivno više tokova. Čak i kada dva toka koji potiču s različitih računara nose istu oznaku, usmerivač može da ih razlikuje na osnovu izvorišnih i odredišnih adresa. Smatra se da oznake tokova ne treba birati redom, već nasumično, jer se od usmerivača očekuje da ih heširaju.

Dužina korisničkih podataka govori koliko bajtova sledi iza 40-bajtnog zaglavlja sa slike 5-68. Ime odgovarajućeg polja IPv4 paketa (*Ukupna dužina)* promenjeno je jer je i značenje polja nešto drugačije: njegova vrednost više ne obuhvata 40 bajtova zaglavlja.

Polje *Sledeće zaglavlje* otkriva tajnu. Osnovno zaglavlje je tako jednostavno zato što se dopuštaju i (opciona) dodatna zaglavlja. Ovo polje saopštava koje od (za sada) šest dodatnih zaglavlja sledi iza aktuelnog zaglavlja, ukoliko je uopšte i upotrebljeno. Ako je aktuelno zaglavlje i poslednje IP zaglavlje, polje *Sledeće zaglavlje* ukazuje na program transportnog protokola (npr. TCP, UDP) kome treba proslediti paket na obradu.

Najveći broj skokova onemogućava večni život paketa. To polje ima istu funkciju kao polje *Životni vek* u protokolu IPv4, tj. njegova vrednost se pri svakom skoku smanjuje za jedan. U protokolu IPv4, *Životni vek* se teorijski izražava sekundama, ali ga nijedan usmerivač nije tako tumačio, pa mu je i ime promenjeno da bi bolje odražavalo način njegove stvarne upotrebe.

Slede polja *Izvorišna adresa* i *Odredišna adresa.* U Deeringovom prvobitnom predlogu (SIP) adrese su bile 8-bajtne, ali su tokom razmatranja protokola mnogi smatrali da će takvih adresa ponestati za nekoliko decenija, dok 16-bajtnih neće po- nestati nikada. Drugi su mislili da su 16-bajtne adrese preterivanje, dok su treći tvrdili da treba koristiti 20-bajtne adrese da bi se postigla kompatibilnost sa OSI protokolom za prenos datagrama. Postojala je i grupica koja se zalagala za adrese promenljive dužine. Posle mnogo prepucavanja, zaključeno je da su 16-bajtne adrese fiksne dužine najbolji kompromis.

Za zapisivanje 16-bajtnih adresa smišljena je nova notacija. Adrese se pišu u obliku osam grupa od po četiri heksadecimalna broja, razdvojenih dvotačkom:

8000:0000:0000:0000:0123:4567:89AB:CDEF

Pošto se očekuje da adrese sadrže brojne nule, odobrene su tri optimizacije. Prvo, vodeće nule u svakoj grupi mogu se ispustiti, tako da se 0123 piše kao 123. Drugo, jedna ili više grupa od po 16 bitova 0 može se zameniti parom dvotački. Na taj način, prethodna adresa postaje:

8000:: 123:4567:89AB :CDEF

Najzad, IPv6 adrese se mogu pisati u staroj decimalnoj notaciji s tačkom ako se na početku stavi par dvotački, na primer:

"192.31.20.46

Verovatno smo već time dosadili, ali zaista ima mnogo 16-bajtnih adresa. U stvari, ima ih 2^{128}, što je približno $3x10^{38}$. Kada bi čitava zemlja (kopno i more) bila prekrivena računarima, IPv6 bi dozvolio $7x10^{23}$ IP adresa po kvadratnom metru njene površine. Hemičari će zapaziti daje ovaj broj veći od Avogadrovog broja ($6{,}025x10^{23}$). Iako nije bila namera da se svakom molekulu na zemlji dodeli njegova sopstvena IP adresa, izgleda da nismo daleko od toga.

Međutim, adresni prostor se u praksi ne koristi tako efikasno, baš kao ni telefonski brojevi (pozivni broj za Menhetn - 212, jedva da dozvoljava još koji priključak, ali pozivni broj za Vajoming - 307 ima slobodnih brojeva koliko hoćete). Durand i Hui- tema su u RFC dokumentu 3194 definisali najnepovoljniji slučaj, koji bi mogao nastati ako bi se upotrebio sistem analogan dodeljivanju telefonskih brojeva, i utvrdili da bi i tada preostalo 1000 IP adresa po svakom kvadratnom metru zemljine površine. U svakoj realističnijoj situaciji preostali bi trilioni adresa po kvadratnom metru. Rečju, nema šanse da u skoroj budućnosti ponestane adresnog prostora.

Zanimljivo je uporediti zaglavlje protokola IPv4 (slika 5-53) sa zaglavljem protokola IPv6 (slika 5-68) da bi se videlo šta je u njemu zadržano od prethodne verzije. Nestalo je polje *DIZ* jer IPv6 zaglavlje ima fiksnu dužinu. Polje *Protokol* je izbačeno jer polje *Sledeće zaglavlje* saopštava šta dolazi iza poslednjeg IP zaglavlja (npr. UDP ili TCP segment).

Uklonjena su sva polja koja se odnose na fragmentiranje jer IPv6 ima drugačiji pristup deljenju paketa. Prvo, od svih računarakoji poštuju IPv6 očekuje se da dinamički određuju veličinu korišćenog datagrama. Zbog toga je manje verovatno da nastupi potreba za fragmentiranjem. Isto tako, minimum je podignut sa 576 na 1280 bajtova da bi se omogućilo postojanje 1024 bajta podataka uz mnoga zaglavlja. Osim toga, kada računar pošalje prevelik IPv6 paket, usmerivač ga ne fragmentira, već vraća po- mku o grešci. Ta poruka nalaže računaru da sve pakete za predmetno odredište smanji (izdeli). Mnogo je efikasnije da računar šalje pakete koji su od početka prave veličine, nego da ih usmerivač deli u hodu.

I na kraju, uklonjeno je polje *Kontrolni zbir* jer njegovo izračunavanje znatno pogoršava performanse. Uz pouzdane mreže koje se danas koriste, zajedno sa činjenicom da sloj veze podataka i transportni sloj normalno imaju svoje kontrolne zbirove, još jedan kontrolni zbir nije toliko vredan da bi se zbog njega dopustilo slabljenje performansi. Uklanjanjem svih navedenih polja protokol je postao jednostavan i moćan. Tako je postignut osnovni postavljeni cilj - IPv6 je brz i fleksibilan protokol sa ogromnim adresnim prostorom.

Dodatna zaglavlja

Neka od ispuštenih zaglavlja protokola IPv4 ponekad su ipak potrebna, pa je IPv6 uveo koncept (opcionih) **dodatnih zaglavlja** (engl. *extension headers).* Takva zaglavlja se mogu

uneti da bi se pružile dopunske informacije, ali moraju biti efikasno kodirana. Zasad je definisano šest vrsta dodatnih zaglavlja (slika 5-69). Svako od njih je opciono, ali ako se doda više od jednog, moraju slediti neposredno iza stalnog zaglavlja - najbolje, navedenim redom.

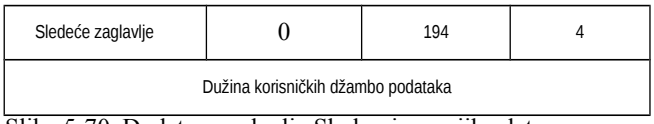
Neka od ovih zaglavlja imaju fiksan format, dok druga sadrže različit broj polja promenljive dužine. Kod ovih je svaka stavka kodirana tripletom: tip, dužina, vred- nost. *Tip* je 1-bajtno polje kojim se naznačava opcija. Vrednosti su tako izabrane da prva 2 bita tog polja saopštavaju „neobaveštenom" usmerivaču kako da obradi opciju. Mogućnosti su: da zanemari opciju; da odbaci paket; da odbaci paket i povratno pošalje ICMP paket; da uradi isto, samo da ne šalje ICMP pakete za grupne adrese (da ne bi jedan neispravan paket poslat grupi adresa izazvao milione ICMP izveštaja).

Dodatno zaglavlje	**Opis**
Skokovi	Različite informacije za usmerivače
Odredište	Dopunske informacije za odredište
Usmeravanje	Labava lista usmerivača koje treba proći
Fragmentiranje	Rad s fragmentima datagrama
Provera identiteta	Proveravanje identiteta pošiljaoca
Bezbednosno šifrovanje	Informacije o šifrovanom sadržaju

Slika 5-69. Dodatna IPv6 zaglavlja.

I *Dužina* je 1-bajtno polje. Ono saopštava dužinu polja *Vrednost* (od 0 do 255 bajtova). *Vrednost* su sve potrebne informacije, dužine do 255 bajtova.

Zaglavlje Skokovi se upotrebljava za informacije koje treba da koriste svi usmeri- vači na putanji. Dosad je definisana samo jedna opcija: podrška za datagrame vec'e od 64K. Format ovog zaglavlja prikazanje na slici 5-70. Kada se koristi, onda se *Dužina korisničkih podataka* u osnovnom zaglavlju postavlja na nulu.



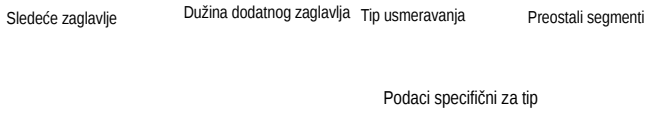
Slika 5-70. Dodatno zaglavlje Skokovi za veiike datagrame (džambograme).

Kao sva dodatna zaglavlja, i ovo počinje bajtom koji naznačava vrstu sledeceg zaglavlja. Iza njega dolazi bajt koji saopštava dužinu zaglavlja Skokovi posle prvih 8 obaveznih bajtova. Sva dodatna zaglavlja počinju na opisani način.

Sledeća 2 bajta naznačuju da ova opcija definiše veličinu datagrama (kod 194) i da njegova veličina predstavlja 4-bajtni broj. Poslednja 4 bajta predstavljaju veličinu datagrama. Veličine manje od 65.536 bajtova nisu dopuštene i takav paket će prvi usme- rivač odbaciti šaljući ICMP poruku o grešci. Datagrarni sa ovim dodatnim zaglavljem zovu se džambogrami (engl. *jumbograms)*. Džambogrami su važni za supeiračunare koji preko Interneta moraju da efikasno šalju gigabajte podataka.

Zaglavlje Odredište sadrži polja koja treba da tumači isključivo odredišni računar. U početnoj verziji protokola IPv6 definisana je samo nulta opcija: dopunjavanje ovog zaglavlja do umnoška od 8 bajtova - tako da u početku ono neće biti korišćeno. Ipak je zaglavlje definisano kako bi nov usmerivački i računarski softver imao šta da radi kada neko jednom izmisli opcije odredišta.

Zaglavlje Usmeravanje sadrži spisak usmerivača koji se moraju proći na putu do odredišta. To je vrlo slično mehanizmu približnog usmeravanja sa izvora kod protokola IPv4, gde se sve unete adrese moraju proći navedenim redosledom, ali se usput mogu posećivati i drugi nenavedeni usmerivači. Format zaglavlja Usmeravanje prikazan je na slici 5-71.



Slika 5-71. Dodatno zaglavlje Usmeravanje.

Prva 4 bajta dodatnog zaglavlja Usmeravanje sadrže četiri 1-bajtna cela broja. Polja *Sledeće zaglavlje* i *Dužina dodatnog zaglavlja* opisana su ranije. Polje *Tip usmeravanja* daje format ostatka zaglavlja. Tip 0 znači da iza prve reči sledi 32-bitna rezervisana reč, a zatim izvestan broj IPv6 adresa. U budućnosti će možda biti

definisani i drugi tipovi. Najzad, polje *Preostali segmenti* vodi računa o tome koliko adresa s liste još nije posečeno. Njegova vrednost se smanjuje za jedan kad god se po- seti adresa s liste. Kada vrednost tog polja padne na nulu, paket dalje može da putuje ka odredištu proizvoljnom putanjom. On je tada obično tako blizu odredišta daje najbolja putanja do njega sasvim očigledna.

Zaglavlje Fragmentiranje deli paket približno onako kao i IPv4. Zaglavlje sadrži identifikator datagrama, broj fragmenta i bit koji naznačava da li sledi još fragmenata. U protokolu IPv6 (nasuprot protokolu IPv4), samo izvorišni računar sme da fragmen- tira pakete. Usputni usmerivači to ne smeju da rade. Ova izmena predstavlja konačan raskid s prošlošću, i uprošćava i ubrzava rad usmerivača. Kao što smo već naglasili, ako usmerivač dobije prevelik paket, on ga odbacuje i izvorištu vrača ICMP poruku o grešci. Ta informacija služi izvorištu da izdeli paket u manje fragmente, da upotrebi ovo dodatno zaglavlje i ponovo pošalje paket.

Zaglavlje Provera identiteta predstavlja mehanizam pomoću koga primalac paketa može da bude siguran ko gaje poslao. Zaglavlje Bezbednosno šifrovanje omogućava da se šifruje sadržaj paketa tako da samo pretpostavljeni primalac može da ga pročita. Za oba ova zaglavlja koriste se kriptografske tehnike.

Nedoumice

Kada uzmemo u obzir javnost projektovanja protokola IPv6 i zagriženost stavova mnogih učesnika rasprave, ne iznenađuje to što su mnoge donete odluke, u najmanju ruku, kompromisne. U nastavku ćemo se kratko osvrnuti na neke od njih, a sve možete detaljno pročitati u odgovarajućim RFC dokumentima.

Već smo pominjali argumentaciju u vezi s dužinom adrese. Rezultat je kompromisan: 16-bajtne adrese fiksne dužine.

Druga bitka se vodila oko polja *Najveći broj skokova.* Jedni su bili čvrsto ubeđeni da ograničavanje maksimalnog broja skokova na 255 (što se mora u 8-bitnom polju) predstavlja katastrofalnu grešku. Ako su danas već uobičajene putanje sa 32 skoka, ko zna koliko će porasti prosečan broj skokova za desetak godina? Ta grupa je predložene ogromne adrese smatrala dalekovidim potezom, ali mali broj predviđenih skokova neoprostivom kratkovidošću. Po njihovom mišljenju, najgore je kada programer negde predvidi premalo bitova.

Na takve argumente moglo bi se odgovoriti protivargumentima za povećanje svakog polja, što bi samo „naduvalo“ zaglavlje. Osim toga, polje *Najveći broj skokova* treba da spreči dugo lutanje paketa po mreži, a za 65.535 skokova treba puno vremena. Najzad, kako Internet raste, gradi se sve više dugih linija koje omogućavaju da se bilo koje dve države povežu s najviše desetak skokova. Ako paketima treba više od 125 skokova da od izvorišta i odredišta stignu do odgovarajućih međunarodnih mrežnih prolaza, onda nešto nije u redu sa okosnicama u tim državama. Osmobitaši su ovde odneli pobedu.

Sledeći vruć krompir bila je maksimalna veličina paketa. Korisnici superračunara zahtevali su pakete veće od 64K. Kada superračunar počne da prenosi podatke, on stvarno to radi profesionalno i ne želi da bude prekidan na svaka 64K. Argument protiv ovoga bio je da paket većine 1 MB može da zaguši TI liniju brzine 1,5 Mb/s tokom više od 5 sekundi, što će izazvati primetno kašnjenje kod interaktivnih korisnika iste linije. I opet je nađen

kompromis: normalni paketi su ograničeni na 64K, ali se za džambograme moglo koristiti dodatno zaglavlje Skokovi.

Treću žestoku raspravu izazvalo je izbacivanje kontrolnog IPv4 zbira. Neki su taj potez upoređivali sa uklanjanjem kočnica iz automobila. Kada to uradite, auto je lakši i ide brže, ali ako se dogodi nešto nepredviđeno, onda ste u sosu.

Izbacivanje kontrolnog zbira branjeno je argumentom da svaka aplikacija koja stvarno brine o svojim podacima ionako predviđa kontrolni zbir u transportnom sloju, tako da još jedan kontrolni zbir (pored kontrolnog zbira u sloju veze podataka) predstavlja nepotreban višak. Štaviše, iskustvo je pokazalo da izračunavanje kontrolnog IP zbira stvara velike troškove u primeni protokola IPv4. Protivnici kontrolnog zbira pobedili su u ovoj raspravi, tako daje iz protokola IPv6 kontrolni zbir uklonjen.

I pokretni korisnici su bili predmet rasprave. Ako prenosivi računar obleti polovinu zemljine kugle, da li na odredištu može i dalje da radi sa istom IPv6 adresom ili mora da koristi sistem domaćih i stranih agenata? Pokretni računari u sistem usmera- vanja unose i asimetriju. Može se lako dogoditi da mali pokretni računar lako hvata snažan signal velikog stacionarnog usmerivača, ali da taj usmerivač ne može da čuje signal upućen s računara. Zbog toga su neki veoma glasno zahtevali da se u IPv6 ugradi izričita podrška za pokretne računare. Ta inicijativa je, međutim, propala jer se nije mogla postići saglasnost ni oko jednog konkretnog predloga.

Najveća bitka se verovatno vodila oko bezbednosti. Niko nije sporio daje bezbeđ- nost neophodna, ali nije bilo slaganja gde i kako je ugraditi. Najpre ono: gde. Argument u prilog smeštanja bezbednosnog mehanizma u mrežni sloj bio je da on tada postaje standardna usluga koju mogu da koriste sve aplikacije bez prethodnog planiranja. Protivargument je bio da aplikacije koje stvarno brinu o bezbednosti ne žele ništa manje nego šifrovanje od jednog do drugog kraja, gde izvorišna aplikacija šifru- je podatke, a odredišna ih dešifruje. Sve što je manje od toga ostavlja korisnika na milost i nemilost potencijalno loših realizacija mrežnog sloja na koje nema nikakvog uticaja. Odgovor na ovaj argument bio je da takve aplikacije mogu da odustanu od IP bezbednosti i da sve obave same. Replika je bila da korisnici koji sumnjaju da mreža obezbeđuje podatke na pravi način, neće želeti da plaćaju sporu i glomaznu IP realizaciju s mehanizmom obezbeđivanja koji moraju da isključe.

Drugi aspekt smeštanja mehanizma bezbednosti u vezi je sa činijenicom da mnoge (ali ne sve) države imaju stroge zakone o izvoženju kriptografskih sistema, tj. sistema za šifrovanje. Neke zemlje - na primer, Francuska i Irak - idu dotle da zabranjuju šifrovanje u domaćem saobraćaju kako korisnici ne bi nešto mogli da sakriju od policije. Zbog toga se IP realizacija sa sistemom šifrovanja koji je dovoljno moćan da bude od neke koristi ne bi mogla izvesti iz SAD (ni iz mnogih drugih zemalja) korisnicima širom sveta. Držanje dve verzije softvera- jednog za domaći i drugog za međunarodni saobraćaj - predstavlja nešto čemu se snažno protivi većina proizvođača.

Svi su se složili oko toga da ne treba očekivati da se IPv4 isključi u nedelju, a da IPv6 osvane u ponedeljak ujutro. Zamišljeno je da se najpre „konvertuju“ pojedina

izolovana ostrvca Interneta, koja bi na početku međusobno komunicirala posredstvom tunela. Kako bi ostrvca rasla, ona bi se spajala u veća ostrva i na kraju bi takva ostrva prerasla čitav Internet. Kada se uzmu u obzir investicije uložene u IPv4 usme- rivače, ne očekuje se da će prelaz na IPv6 biti kraći od desetak godina. Zbog toga je preduzeto sve što se može da taj prelazak bude što bezbolniji. Više detalja o protokolu IPv6 potražite kod Loshina (1999).

5.7 SAŽETAK

Mrežni sloj obezbeđuje usluge transportnom sloju. On se može zasnivati na virtu- elnim kolima ili na datagramima. Njegov glavni zadatak, u oba slučaja, jeste da usmeri pakete sa izvorišta na odredište. U podmrežama zasnovanim na virtuelnim kolima, odluka o usmeravanju donosi se pri uspostavljanju virtuelnog kola. U datagramskim podmrežama, ona se donosi nezavisno za svaki paket.

U računarskim mrežama koriste se mnogobrojni algoritmi za usmeravanje. Statički algoritmi obuhvataju usmeravanje najkraćom putanjom i plavljenje. Usmeravanje zasnovano na vektoru razdaljine i usmeravanje zasnovano na stanju veze jesu dinamički algoritmi. Postojeće mreže uglavnom koriste jedan od njih. U ostale važne teme koje se tiču usmeravanja spadaju hijerarhijsko usmeravanje, usmeravanje za pokretne korisnike, usmeravanje difuznog emitovanja, usmeravanje višesmernog emitovanja i usmeravanje u mrežama ravnopravnih računara.

Podmreža se lako može zagušiti, zbog čega paketi više kasne, a njihov protok se smanjuje. Projektanti nastoje da izbegnu zagušenja odgovarajućim planom mreže. Tehnike izbegavanja zagušenja obuhvataju pravila ponovnog emitovanja paketa, ke- širanje, kontrolu toka i drugo. Ako do zagušenja ipak dođe, s njim se treba izboriti. Mogu se povratno slati prigušni paketi, paketi se mogu odbacivati, a postoje i druge metode.

Od proste borbe sa eventualnim zagušenjem bolje je da se unapred razmišlja o načinu za obezbeđenje obećanog kvaliteta usluga. Za to se mogu koristiti metode ba- ferovanja kod klijenta, ujednačavanja saobraćaja, rezervisanja resursa i kontrole pristupanja. Opštiji pristupi za obezbeđivanje visokog kvaliteta usluga obuhvataju tzv. integrisane usluge (uključujući RSVP), diferencirane usluge i MPLS.

Mreže se međusobno razlikuju na brojne načine, tako da mogu nastati problemi kada se poveže više mreža. Ponekad se problem rešava tako što se paket prosledi tunelom kroz „nepodobnu“ mrežu, ali to ne radi ako se izvorišna i odredišna mreža razlikuju. Kada je u povezanim mrežama veličina paketa na različit način ograničena, u pomoć se može pozvati fragmentiranje.

Internet obiluje protokolima mrežnog sloja. Među njima je protokol za prenos podataka IP, ali i protokoli za upravljanje: ICMP, ARP i RARP, kao i protokoli za usmeravanje: OSPF i BGP. Internet brzo ostaje bez slobodnih IP adresa, tako da je razvijena nova verzija IP protokola - protokol IPv6.

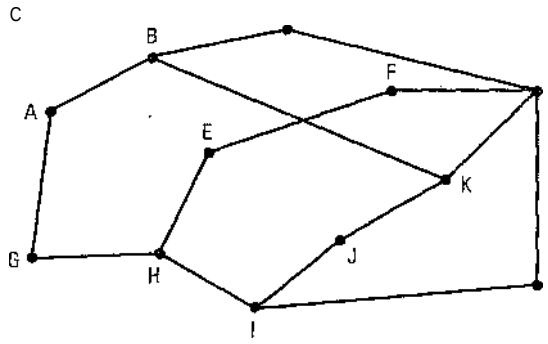
ZADACI

1. Navedite dve oblasti primene računara kojima najviše odgovara usluga sa uspostavljanjem direktne veze i dve za koje je najbolja usluga bez uspostavljanja direktne veze.
2. Postoje li okolnosti u kojima će usluga sa uspostavljanjem direktne veze isporučivati pakete preko reda (ili bi barem trebalo da ih tako isporučuje)? Obrazložite odgovor.

3. Datagramske podmreže nezavisno usmeravaju svaki paket. Podmreže zasnovane na virtuelnim kolima ne moraju to da rade, pošto svaki paket podataka sledi unapred utvrđenu putanju. Znači li to da podmreže s virtuelnim kolima ne treba da imaju mogućnost usmeravanja izolovanih paketa od proizvoljnog izvorišta do proizvoljnog odredišta? Objasnite svoj odgovor.
4. Navedite tri parametra protokola o kojima se može pregovarati tokom uspostavljanja veze.
5. Razmotrite sledec'i projektni problem koji se odnosi na ugradnju usluge virtuelnih kola. Ako se u podmreži interno koriste virtuelna kola, svaki paket podataka mora imati 3-bajtno zaglavlje, a svaki usmerivač mora potrošiti 8 bajtova memorije za iđentifikovanje kola. Ako se interno koriste datagrami, potrebna su 15-bajtna zaglavlja, ali se ne zauzima memorija usmerivača. Prenos 10 bajtova košta 1 cent po skoku. Vrlo brza usmerivačka memorija može se nabaviti za 1 cent po bajtu i amortizuje se tokom dve godine, uz pretpostavku o 40-časovnoj radnoj sedmici. Statistički prošek trajanja sesije je 1000 sekundi, tokom kojih se prenese 200 paketa. Prosečnom paketu do cilja treba četiri skoka. Koja ugradnja je jeftinija i za koliko?
6. Pretpostavljajući da svi usmerivači i računari rade ispravno i da njihov softver ne sadrži greške, postoji li šansa, ma kako mala, da paket bude isporučen na pogrešnu adresu?
7. Razmotrite mrežu sa slike 5-7, ali zanemarite „težine“ linija. Pretpostavite da se u njoj za usmeravanje koristi algoritam plavljenja. Ako paket poslat od *A* ka *D* ima maksimalan broj skokova jednak 3, navedite sve njegove putanje. Takođe odgovorite koliko njegovih skokova je primereno propusnom opsegu mreže?
8. Navedite jednostavan postupak pronalaženja dve putanje kroz mrežu od zadatog izvorišta do zadatog odredišta koje mogu da prežive gubitak bilo koje komunikacione linije (pod uslovom da takve dve putanje postoje). Pretpostavite da su usmerivači dovoljno pouzdani, tako da ne morate da razmišljate o njihovom otkazivanju.
9. Razmotrite podmrežu sa slike 5-13(a). Koristi se usmeravanje zasnovano na vektoru razdaljine, i u usmerivač *C* upravo su stigli sledeći vektori od *B:* (5, 0, 8, 12, 6, 2); od *D:* (16, 12, 6, 0, 9, 10); i od *E:* (7, 6, 3, 9, 0,4). Merena kašnjenja do *B, D* i *E* su 6, 3 i 5. Kako izgleda nova tabela usmerivača *Cl* Navedite izlazne linije i očekivana kašnjenja.
10. Ako se kašnjenja beleže 8-bitnim brojevima u mreži sa 50 usmerivača, a vektori razdaljine se izmenjuju dva puta u sekundi, koliko propusnog opsega svake (potpune dupleksne) linije „pojede“ distribuirani algoritam za usmeravanje? Pretpostavite da je svaki usmerivač s drugim usmerivačima povezan pomoc'u tri linije.
11. Na slici 5-14 logička disjunkcija (OR) dva skupa bitova ACF daje 111 u svakom redu. Da li je u pitanju koincidencija ili to uvek važi za sve podmreže?
12. Odredite tako veličinu oblasti i grupa za trostepeno hijerarhijsko usmeravanje pomoću 4800 usmerivača da tabele za usmeravanje budu minimalne. Dobro je poći od pretpostavke daje rešenje sa *k* grupa od po *k* oblasti sa po *k* usmerivača blisko optimalnom, sto znači daje *k* približno kubni koren iz 4800 (oko 16). Metodom isprobavanja proverite kombinacije u kojima sva tri parametra imaju vrednosti bliske 16.
13. U tekstu je navedeno da kada pokretni računar nije na matičnoj lokaciji, sve pakete koji stižu za njegovu lokalnu mrežu presreće njegov domaći agent. Kako domaći agent to obavlja za IP mrežu kada je lokalna mreža tipa 802.3?
14. Posmatrajte mrežu na slici 5-6 i odgovorite koliko *B* generiše paketa pri neusmere- nom emitovanju, ako se koristi
    a) usmeravanje ispitivanjem izvorišta?
    b) stablo optimalnih putanja?
15. Posmatrajte mrežu na slici 5-16(a). Zamislite da je između *F* i *G* dodata nova linija, ali

je stablo optimalnih putanja sa slike 5-16(b) ostalo neizmenjeno. Šta se menja na slici 5-16(c)?

16. Za mrežu na sledećoj slici izračunajte razgranato stablo za višesmerno emitovanje usmerivača C grupi računara vezanih za usmerivače *A, B, C, D, E, F, I* i *K*.



17. Da li čvor *H* ili čvor *I* na slici 5-20 ikada reaguju difuznom emisijom na prikazanu pretragu koja počinje u čvoru A?
18. Pretpostavite da je čvor *B* na slici 5-20 upravo pokrenut i da u svojim tabelama nema podatke o usmeravanju. Odjednom mu zatreba putanja do *II.* On difuzno emituje pakete s poljem *TTL* postavljenim na 1, 2, 3 itd. Iz koliko će takvih pokušaja pronaći putanju do *Hl*
19. U najjednostavnijoj verziji algoritma Chord za ravnopravno pretraživanje putanja, ne koristi se tabela prstiju, već se traženje obavlja linearno, duž kružnice u jednom ili drugom smeru. Može li čvor tačno predvideti smer u kome treba da traži? Obrazložite odgovor.
20. Posmatrajte krug za algoritam Chord na slici 5-24. Pretpostavite da se čvor 10 od-jednom uključio u igru. Da Ii to utiče na tabelu prstiju čvora 1 i ako utiče, kako?
21. Kao jedan od mogućih mehanizama za kontrolu zagušenja u podmreži koja interno radi s virtuelnim kolima, usmerivač može da odloži potvrđivanje primljenog paketa (1) sve dok ne sazna daje njegov poslednji prenos tim virtuelnim kolom uspešno završen i (2) sve dok ne bude imao slobodan bafer. Pretpostavite zbog jednostavnosti da usmerivači koriste protokol „stani i čekaj“ i da svako virtuelno kolo ima rezer- visan po jedan bafer za svaki smer saobraćaja. Ako prenos paketa (podataka ili potvrde) traje *T* sekundi, a na putanji ima *n* usmerivača, kojom brzinom se paketi isporučuju odredišnom računaru? Pretpostavite da su greške u prenosu retke i daje veza računar-usmerivač beskonačno brza.
22. Datagramska podmreža dozvoljava usmerivačima da odbacuju pakete kad god za tim osete potrebu. Verovatnoća da će usmerivač odbaciti paket iznosi *p.* Razmotrite slučaj izvorišnog računara povezanog sa izvorišnim usmerivačem, koji je povezan sa odredišnim usmerivačem, a zatim sa odredišnim računarom. Ako bilo koji usmerivač odbaci paket, tajmer izvorišnog računara ističe i on ponovo šalje paket. Ako se linije računar-usmerivač i usmerivač-usmerivač računaju kao skokovi, koji je pro- sečan broj
    a) skokova zajedno slanje paketa?
    b) slanja istog paketa?
    c) skokova potreban da paket bude primljen?
23. Objasnite glavne razlike između metode upozoravajućeg bita i metode RED.
24. Navedite jedan razlog zbog koga algoritam bušne kofe treba da propušta jedan paket po

otkucaju sistemskog sata, nezavisno od veličine paketa.

25. U određenom sistemu koristi se varijanta algoritma bušne kofe s prebrojavanjem bajtova. Pravilo je da se tokom jednog otkucaja sistemskog sata može poslati jedan paket od 1024 bajta ili dva paketa po 512 bajtova itd. Navedite ozbiljno ograničenje ovakvog sistema o kome nije bilo reči u tekstu.
26. ATM mreža koristi algoritam kofe sa žetonima za ujednačavanje saobraćaja. Nov žeton se stavlja u kofu svakih 5 ps. Svaki žeton važi za jednu ćeliju koja sadrži 48 bajtova podataka. Kolika je maksimalna podrživa brzina prenosa podataka?
27. Računarom na mreži brzine 6 Mb/s upravlja kofa sa žetonima. Kofa se puni brzinom 1 Mb/s. U početku je napunjena sa 8 megabita. Koliko dugo računar može da šalje pakete maksimalnom brzinom 6 Mb/s?
28. Zamislite specifikaciju toka s paketom maksimalne veličine 1000 bajtova, kofom sa žetonima brzine 10 miliona bajtova u sekundi i kapaciteta milion bajtova, i maksimalnom brzinom prenosa 50 miliona bajtova u sekundi. Koliko najduže traje rafal podataka maksimalne brzine?
29. U mreži na slici 5-37 koristi se protokol RSVP i višesmerna razgranata stabla za računare 1 i 2. Pretpostavite da računar 3 zahteva kanal propusnog opsega 2 Mb/s za tok od računara 1 i drugi kanal opsega 1 Mb/s za tok od računara 2. U isto vreme, računar 4 zahteva kanal opsega 2 Mb/s za tok od računara 1, a računar 5 zahteva kanal opsega 1 Mb/s za tok od računara 2. Koliko će ukupno propusnog opsega biti re- zervisano u usmerivačima A, *B, C, E, H, J, K* i L?
30. Mikroprocesor usmerivača može da obradi 2 miliona paketa u sekundi, a nudi mu se saobraćaj veličine 1,5 miliona paketa u sekundi. Ako na putanji od izvorišta do odredišta ima 10 usmerivača, koliko vremena mikroprocesori troše na svrstavanje u redove čekanja i obradu paketa?
31. Razmotrite primer korisnika diferenciranih usluga sa ekspresnim prosleđivanjem. Ima li garancije da ce ekspresni paketi kasniti manje od redovnih paketa? Ako ima, zašto? Ako nema, opet zašto?
32. Da li je fragmentiranje potrebno u mrežama sa ulančanim virtuelnim kolima ili je ono neophodno samo u datagramskim sistemima?
33. Prenos tunelom kroz podmrežu sa ulančanim virtuelnim kolima sasvim je jednostavan: višeprotokolarni usmerivač najednom kraju uspostavi virtuelno kolo do drugog kraja i samo njime šalje pakete. Da li se tuneli mogu primeniti i u datagramskim mrežama? Ako mogu, kako?
34. Pretpostavite daje računar *A* povezan sa usmerivačem *Rl,* daje *R1* povezan s drugim usmerivačem *R2,* a *R2* je povezan s računarom *B.* Pretpostavite da je TCP poruka koja sadrži 900 bajtova podataka i 20 bajtova TCP zaglavlja predata IP kodu na računaru A za isporuku računaru *B.* Prikažite sadržaj polja *Ukupna dužina, Identifikacija, NF, JF* i *Redni broj fragmenta* u IP zaglavlju svakog od paketa koji se šalju preko tri veze. Pretpostavite da veza *A-Rl* može da podrži pakete maksimalne veličine 1024 bajta, uključujući i 14-bajtno zaglavlje okvira, da veza *R1-R2* podržava okvir maksimalne veličine 512 bajtova, uključujući i 8-bajtno zaglavlje okvira, a da veza *R2-B* podržava okvire maksimalne veličine 512 bajtova, uključujući i 12-bajt- no zaglavlje okvira.
35. Usmerivač brzometno emituje pakete ukupne dužine (podaci plus zaglavlje) 1024 bajta. Pretpostavljajući da paketi žive 10 sekundi, pri kojoj maksimalnoj brzini linije može da radi usmerivač, a da ne zadre u identifikatorski prostor IP datagrama?
36. IP datagram sa opcijom *Strogo usmeravanje sa izvora* treba da se fragmentira. Mislite li da ta opcija treba da se kopira u svaki fragment ili je dovoljna samo u prvom? Obrazložite odgovor.

37. Pretpostavite da se za mrežni deo adrese klase B, umesto 16 bitova, koristi 20 bitova. Koliko bi tada postojalo mreža klase B?
38. Heksadecimalno zadatu IP adresu C22F1582 napišite koristeći decimalnu notaciju s tačkom.
39. Mreža na Internetu ima masku podmreže 2.55.255.240.0. Koliko najviše računara može ona da podrži?
40. Počev od adrese 198.16.0.0, na raspolaganju je veliki broj uzastopnih adresa. Pretpostavimo da četiri organizacije: *A, B, C i D* zahtevaju, redom, 4000, 2000, 4000 i 8000 adresa. Navedite prvu i poslednju adresu koja se svakoj od njih dodeljuje, kao i masku u notaciji *w.x.y.zJs*.
41. Usmerivač je upravo primio sledeće nove IP adrese: 57.6.96.0/21, 57.6.104.0/21, 57.6.112.0/21 i 57.6.120.0/21. Ako sve koriste istu izlaznu liniju, mogu li se one grupisati u istu odrednicu? Ako mogu, kako će izgledati odrednica tabele? Ukoliko ne mogu, objasniti zašto.
42. Skup IP adresa između 29.18.0.0. i 19.18.128.255 grupisan je u adresu 29.18.0.0/17. Međutim, postoji prazan prostor od 1024 nedodeljene adrese između 29.18.60.0 i 29.18.63.255, i on se najednom dodeljuje računaru koji koristi drugu izlaznu liniju. Da li sada treba da se grupisane adrese razbiju na sastavne delove, da se nov blok doda u tabelu i da se pokuša novo grupisanje? Ako to ne treba da se radi, šta treba?
43. Usmerivač ima sledeće (CIDR) odrednice u svojoj tabeli za usmeravanje:

Adresa/maska	Sledeci skok
135.46.56.0/22	Interfejs 0
135.46.60.0/22	Interfejs 1
192.53.40.0/23	Usmerivač 1
podrazumevana	Usmerivač 2

Šta će usmerivač raditi kada mu stignu paketi sa sledećim adresama:

a) 135.46.63.10
b) 135.46.57.14
c) 135.46.52.2
d) 192.53.40.7
e) 192.53.56.7

44. Mnoge kompanije se preko dva (ili više) usmerivača povezuju na Internet, što obezbeđuje rad i kada jedan od njih otkaže. Da li je takva strategija moguća u sistemu NAT? Obrazložite odgovor.
45. Upravo ste prijatelju objasnili kako radi protokol ARP. Pošto ste završili, on kaže: „Shvatio sam. ARP obezbeđuje uslugu mrežnom sloju, prema tome, on je deo sloja veze podataka“. Šta ćete mu odgovoriti?
46. Protokoli ARP i RARP preslikavaju adrese iz jednog prostora u drugi. U tom pogledu su slični. Međutim, pri ugradnji jednog i drugog protokola postoje fundamentalne razlike. Koje su najveće?
47. Opišite postupak sklapanja IP paketa iz fragmenata na odredištu.
48. Većina algoritama za sklapanje IP datagrama ima tajmer koji prazni bafer za sklapanje uprkos tome što izgubljeni fragmenti nikada ne stižu. Pretpostavite daje da- tagram izdeljen u četiri fragmenta. Prva tri fragmenta srećno stižu, ali četvrti kasni. Na kraju se isključuje tajmer i tri pristigla fragmenta se odbacuju iz bafera. Malo kasnije pristiže i zakasneli četvrti fragment. Šta s njim da se radi?
49. Kod IP i ATM paketa kontrolni zbir pokriva samo zaglavlje, ali ne i podatke. Šta mislite, zastoje to tako predviđeno?

50. Korisnik koji živi u Bostonu putiije za Mineapolis i nosi svoj prenosivi računar. Na njegovo iznenađenje, lokalna mreža u Mineapolisu je bežična IP mreža na koju ne mora da se (fizički) priključuje. Da li je još uvek neophodno proći čitav postupak sa stranim i domaćim agentima da bi e-pošta i ostali saobraćaj stizali ispravno?
51. Protokol IPv6 koristi 16-bajtne adrese. Ako se po jedan blok od milion adresa dodeljuje svake pikosekunđe, posle koliko vremena će se potrošiti sve adrese?
52. Polje *Protokol* zaglavlja IPv4 paketa ne postoji u osnovnom IPv6 zaglavlju. Zašto?
53. Kada se uvede protokol IPv6, znači li to da treba izmeniti protokol ARP? Ako je odgovor potvrdan, da li će te promene biti suštinske ili tehničke?
54. Napišite program koji simulira usmeravanje mehanizmom plavljenja. Svaki paket treba da ima brojač čija će se vrednost pri svakom skoku smanjivati za jedan. Kada vrednost brojača dostigne nulu, paket se odbacuje. Vreme je podeljeno u diskretne intervale i svaka linija obrađuje jedan paket tokom jednog vremenskog intervala. Napravite tri verzije programa: sve linije se plave, plave se sve linije osim ulazne, i plavi se samo k najboljih (statički izabranih) linija. Uporedite plavljenje uz determinističko usmeravanje ($k = 1$) u pogledu kašnjenja i korišćenog propusnog opsega.
55. Napišite program koji simulira računarsku mrežu s vremenom izdeljenim u diskretne intervale. Prvi paket u redu čekanja svakog usmerivača pravi jedan skok tokom jednog vremenskog intervala. Svaki usmerivač ima konačan broj bafera. Ako stigne paket, pa za njega nema mesta, on se odbacuje i ne šalje ponovo. Umesto toga, protokol koji povezuje dva kraja, pomoću svojih tajmera i potvrda odmah regeneriše pakete sa izvorišnog usmerivača. Prikažite protok kroz mrežu u funkciji roka tajmera (potrebnog da paket stigne s jednog kraja mreže na drugi), koristeći učestalost grešaka kao parametar.
56. Napišite funkciju koja obavlja prosleđivanje u IP usmerivaču. Procedura ima jedan parametar - IP adresu. Ona ima pristup i globalnoj tabeli čije su odrednice trokomponentni podaci - tripleti. Svaka odrednica sadrži tri cela broja: IP adresu, masku podmreže i izlaznu liniju. Funkcija mehanizmom CDIR pretražuje IP adrese u tabeli i kao vrednost vraća izlaznu liniju.
57. Pomoću programa *tracerute* (UNIX) ili *tracert* (Windows) ispratite putanje od vašeg računara do različitih univerziteta na drugim kontinentima. Napravite spisak prekookeanskih veza koje otkrijete. Pokušajte sa sledećim gradovima:

    www.berkeley.edu *(Kalifornija)*
    www.mit.edu *(Masačusets)*
    www.vu.nl *(Amsterdam)*
    www.ucl.ac.uk *(London)*
    www.usyd.edu.au *(Sidnej)* www.u-tokyo.ac.jp *(Tokio)* www.uct.ac.za *(Kejptaun)*

TRANSPORTNI SLOJ

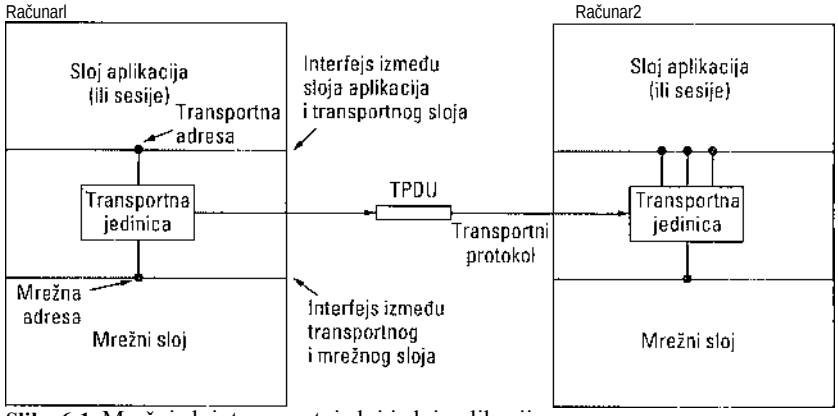
Transportni sloj nije samo još jedan od mnogih slojeva. On je sama srž hijerarhije protokola. Njegov zadatak je da obezbedi pouzdan, isplativ prenos podataka, bez obzira na fizičku mrežu ili mreže koje se trenutno nalaze između izvorišnog i odredišnog računara. Bez transportnog sloja, čitav koncept raspoređivanja protokola po slojevima gubi smisao. U ovom poglavlju, detaljno ćemo proučiti transportni sloj, zajedno s njegovim uslugama, protokolima i performansama.

6.1 USLUGA PRENOSA

U narednim odeljcima upoznacemo se sa uslugom prenosa. Utvrdičemo kakva se usluga obezbeđuje sloju aplikacija. Da bismo temu usluge prenosa izložili što kon- kretnije, ispitaćemo dva skupa njenih osnovnih oblika. Prvo ćemo obraditi jednostavan (ali hipotetički) skup usluga da biste shvatili šta se takvim uslugama želi postići, a zatim ćemo preći na interfejs koji se obično koristi na Internetu.

6.1.1 Usluge koje se **obezbeđuju** za više slojeve

Glavni zadatak transportnog sloja je obezbeđivanje efikasne, pouzdane i isplative usluge njegovim korisnicima - najčešće procesima u sloju aplikacija. Da bi taj zadatak ispunio, transportni sloj koristi usluge koje mu obezbeđuje mrežni sloj. Hardver i(ili) softver unutar transportnog sloja koji obavlja prenos zove se **transportna jedinica** (engl. *transport entity).* Transportna jedinica može da se nalazi u jezgru operativnog sistema, u posebnom korisničkom procesu, u biblioteci smeštenoj u mrežnu aplikaciju ili, razumljivo, na mrežnoj kartici. Odnosi (logički) između sloja aplikacija, transportnog i mrežnog sloja prikazani su na slici 6-1.



Slika 6-1. Mrežni sloj, transportni sloj i sloj aplikacija.

Kao što postoje dve vrste usluga mrežnog sloja: sa uspostavljanjem direktne veze i bez nje, tako postoje i dve vrste usluga prenosa. Usluga prenosa sa uspostavljanjem direktne veze prilično liči na odgovarajuću uslugu mrežnog sloja. Rad se u oba slučaja obavlja u tri faze: uspostavljanje veze, prenos podataka i raskidanje veze. Slični su takođe način adresiranja i kontrola toka podataka. Osim toga, i usluga prenosa bez uspostavljanja direktne veze liči na odgovarajuću uslugu mrežnog sloja.

Prirodno je da upitate: Ako usluga transportnog sloja toliko liči na uslugu mrežnog sloja, zašto uopšte postoje dva različita sloja? Zašto se sve ne bi moglo obaviti u jedinstvenom sloju? Odgovor na ovo pitanje nije sasvim očigledan, ali zadire u samu suštinu, zbog koje se moramo vratiti na sliku 1-9. Kod transportnog sloja izvršava se isključivo na korisničkim računarima, dok se kod mrežnog sloja izvršava pretežno na usmerivačima koje održava kompanija za prenos podataka (barem u regionalnim mrežama). Sta se dešava ako mrežni sloj ne obezbeđuje odgovarajuću uslugu - na pri- mer, ako često gubi pakete? Sta se dešava ako usmerivači povremeno otkazuju?

Nastaju problemi, eto šta. Korisnici nemaju stvarnog uticaja na mrežni sloj, pa ne mogu da poboljšaju njegovu uslugu ugrađujući kvalitetnije usmerivače ili uvodeći ozbiljnije ispravljanje grešaka u sloj veze podataka. Jedina mogućnost je da se iznad mrežnog sloja postavi još jedan sloj koji će poboljšati kvalitet usluge. Ako u podmreži koja radi sa uspostavljanjem direktne veze transportna jedinica usred prenosa duge serije podataka dobije informaciju daje veza u mrežnom sloju naglo prekinuta i da se ništa ne zna o podacima koji su se zatekli na putu, ona može da uspostavi novu vezu sa udaljenom transportnom jedinicom. Preko te nove veze ona može drugoj transportnoj jedinici da pošalje upit o tome koji su podaci stigli a koji nisu, i da nastavi da ih šalje od te tačke.

Postojanje transportnog sloja u suštini omogućava da usluga prenosa bude pouzdanija od usluge mrežnog sloja na koju se oslanja. Transportni sloj može da evidentira izgubljene ili oštećene pakete i da ih pošalje ponovo. Staviše, osnovne usluge prenosa mogu se realizovati kao procedure koje se pozivaju iz biblioteke, što ih čini nezavisnim od osnovnih usluga mrežnog sloja. Pozivanje usluga mrežnog sloja može se znatno razlikovati od jedne mreže do druge (npr. usluga za lokalnu mrežu bez uspostavljanja direktne veze može da izgleda sasvim drugačije od usluge za regionalnu mrežu sa uspostavljanjem direktne veze). Kada uslugu

mrežnog sloja zaklonite skupom osnovnih usluga transportnog sloja, tada ćete pri promeni mrežne usluge morati da samo jedan skup procedura u biblioteci zamenite dragim, a te procedure će obavljati isti posao sa sada drugačijom mrežnom uslugom.

Zahvaljujući postojanju transportnog sloja, programeri aplikacija mogu da pišu kod za različite mreže držeći se standardnog skupa osnovnih usluga i ne moraju da brinu o različitim mrežnim interfejsima i nepouzdanom prenosu. Kada bi sve stvarne mreže radile bez greške, sve imale iste osnovne oblike usluga i pri tome se ne bi nikada menjale, možda ne bi bilo potrebe za transportnim slojem. Ovako, u stvarnom svetu, on ispunjava svoju glavnu ulogu da više slojeve izoluje od tehnologije, strukture i nedostataka podmreže.

Iz navedenih razloga mnogi prave razliku između slojeva 1 do 4, sjedne strane, i sloja ili slojeva iznad četvrtog, s drage. Donja četiri sloja mogu se shvatiti kao davalac usluga prenosa (engl. *transport service provider),* dok viši slojevi predstavljaju korisnika usluga prenosa (engl. *transport service user).* Takva podela značajno utiče na način projektovanja slojeva i transportnom sloju nameće ključnu ulogu, pošto on predstavlja glavnu granicu između davaoca i korisnika pouzdane usluge prenosa podataka.

6.1.2 Osnovne operacije u uslugama prenosa

Da bi korisnicima omogućio pristup uslugama prenosa, transportni sloj mora da aplikacijama obezbedi neke *osnovne operacije* (engl. *primitives),* tj. interfejs ka uslugama prenosa. Svaka usluga prenosa ima sopstveni interfejs. U ovom odeljku prvo ćemo ispitati jednostavnu (hipotetičku) uslugu prenosa i njen interfejs da bismo se upoznali sa suštinom. U sledećem odeljku obradićemo primer iz realnog sveta.

Usluga prenosa slična je usluzi mrežnog sloja, ali između njih ima i važnih razlika. Glavna je to što je mrežna usluga namenjena stvarnim mrežama, sa svim njihovim nedostacima. Stvarne mreže znaju da gube pakete, pa je usluga mrežnog sloja u načelu nepouzdana.

Nasuprot tome, usluga prenosa (sa uspostavljanjem direktne veze) potpuno je pouzdana. Naravno, stvarne mreže ne rade bez grešaka, ali zato je glavni zadatak transportnog sloja da obezbedi pouzdanu uslugu preko nepouzdane mreže.

Zamislite, primera radi, dva procesa koja su u UNIX-u povezana kanalima (engl. *pipes).* Procesi smatraju da je veza između njih savršena i ne žele ni da čuju o gubljenju paketa, zagušenju i sličnim problemima. Oni zapravo žele stoprocentno pouzdanu vezu. Proces *A* stavlja podatke na jedan kraj kanala, a proces *B* ih preuzima na dragom kraju. To je sve što se može redi o usluzi prenosa sa uspostavljanjem direktne veze - zaklanjanje nesavršenosti usluge mrežnog sloja, tako da korisnički procesi mogu da računaju na tok bitova bez ijedne greške.

Uzgred, transportni sloj može da obezbedi i nepouzdanu (datagramsku) uslugu. Međutim, o tome gotovo da se nema šta novo redi, pa ćemo se u ovom poglavlju uglavnom usmeriti na uslugu prenosa sa uspostavljanjem direktne veze. Ipak, neke aplikacije, npr. za klijentsko-serverski rad i preuzimanje multimedijskog sadržaja u realnom vremenu, izvlače korist iz prenosa bez uspostavljanja direktne veze, pa demo
0 njima kasnije još nešto redi.

Druga razlika između usluga transportnog i mrežnog sloja ogleda se u korisnicima takvih usluga. Usluge mrežnog sloja koriste samo transportne jedinice. Retko lco sam piše kod

transportne jedinice, tako da mali broj korisnika ili programa uopšte dolazi u direktan dodir sa uslugama mrežnog sloja. Nasuprot tome, mnogi programi (dakle,
1 programeri) rade sa osnovnim uslugama prenosa. Prema tome, usluga prenosa mora da bude podesna i da se lako koristi.

Da biste dobili bližu predstavu o tome kako izgleda usluga prenosa, pogledajte njenih pet osnovnih operacija na slici 6-2. Prikazani transportni interfejs zaista je rudimentaran, ali ćete ipak uspeti da shvatite šta on mora da obezbedi pri uspostavljanju direktne veze. On aplikacijama omogućava da uspostave vezu, daje koriste i daje na kraju raskinu, što je većini sasvim dovoljno.

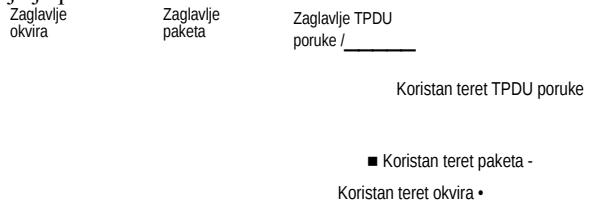
Osnovna operacija	**Paket koji se šalje**	**Značenje**
LISTEN	(ne šalje se)	Blokira rad dok neki proces ne pokuša da se uključi
CONNECT	CONNECTION REQUEST	Aktivno pokušava da uspostavi vezu
SEND	DATA	Šalje podatke
RECEIVE	(ne šalje se)	Blokira rad dok ne stigne paket DATA
DISCONNECT	DISCONNECTION REQUEST	Ova strana želi da raskine vezu

Slika 6-2. Osnovne operacije u jednostavnoj usluzi prenosa.

Da biste videli kako se koriste ove operacije, zamislite aplikaciju sa serverom i više udaljenih klijenata. Najpre server izvršava operaciju LISTEN (osluškujem), najčešće tako što iz biblioteke poziva odgovarajuću proceduru koja sistemski blokira server dok se ne pojavi neki klijent. Kada klijent poželi da stupi u vezu sa serverom, on izvršava operaciju CONNECT (želim vezu). Transportna jedinica izvršava ovu operaciju tako što blokira pozivaoca i šalje paket serveru. U koristan teret ovog paketa kapsulirana je poruka za transportnu jedinicu servera.

Treba redi nešto i o terminologiji. U nedostatku boljeg izraza, koristidemo pomalo rogobatnu skraćenicu *TPDU* (jedinica podataka transportnog protokola, engl. *Transport Protocol Data Unit)* za poruke koje između sebe razmenjuju transportne jedinice. Dakle, u paketima (koje razmenjuje mrežni sloj) nalaze se TPDU poruke (koje razmenjuje transportni sloj). Paketi se, sa svoje strane, nalaze u okvirima (koje

razmenjuje sloj veze podataka). Kada okvir stigne, sloj veze podataka obrađuje njegovo zaglavlje i koristan teret prosleđuje višem sloju - mrežnoj jedinici. Mrežna jedinica obrađuje zaglavlje paketa i prosleđuje njegov koristan teret višem sloju - transportnoj jedinici. To ugnežđivanje je prikazano na slici 6-3.



Slika 6-3. Ugnežđivanje TPDU poruka, paketa i okvira.

Ako se vratimo našem primem, poziv koji je lđijent uputio proceduri CONNECT re- zultuje slanjem TPDU poruke CONNECTION REQUEST (zahtev za vezu) serveru. Kada ona stigne, transportna jedinica proverava - operacijom LISTEN - da li je server blokiran (tj. da lije zainteresovan za obradu zahteva). Zatim ga deblokira i klijentu šalje TPDU poruku CONNECTION ACCEPTED (zahtev za vezu prihvaćen). Kada TPDU poruka stigne klijentu, on se deblokira i veza se uspostavlja.

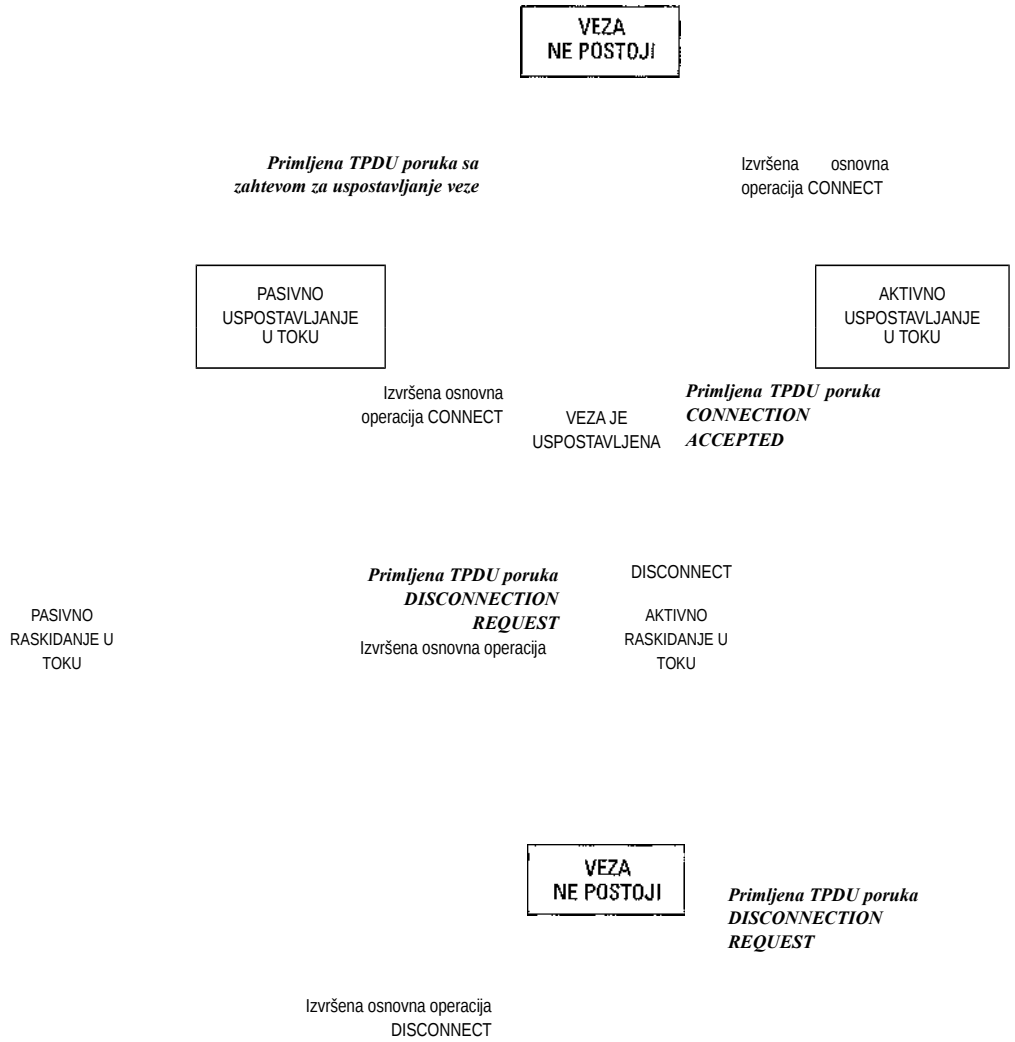
Sada se mogu razmenjivati podaci korišćenjem osnovnih operacija SEND (šaljem) i RECEIVE (primam). U najjednostavnijem slučaju, svaka strana može da se blokira (operacijom RECEIVE) i da čeka da draga strana izvrši operaciju SEND. Kada TPDU poruka stigne, primalac se deblokira. On tada može da obradi TPDU poruku i da pošalje odgovor. Opisani postupak radi dobro sve dok obe strane vode računa o redosledu slanja.

Obratite pažnju na to da je čak i jednostavna jednosmerna razmena podataka složenija u transportnom sloju nego u mrežnom. Svaki paket podataka mora da (jednom) bude i potvrđen. I paketi koji nose upravljačke TPDU poruke takođe se potvrđuju, posredno ili neposredno. Potvrdama rukuju transportne jedinice koristeći protokol mrežnog sloja - one su skrivene od korisnika prenosa. Transportne jedinice, isto tako, brinu o tajmerima i ponovnom slanju. Ništa od toga ne vide korisnici prenosa. Za njih je veza pouzdan kanal za prenos bitova: jedan korisnik ih na svom kraju trpa u kanal, a dragi, na dragom kraju, samo ih vadi. Raspoređivanje protokola po slojevima izuzetno je korisno upravo zbog takve mogućnosti skrivanja detalja.

Kada veza više nije potrebna, treba je raskinuti da bi se oslobodio prostor u tabelama transportnih jedinica. Način raskidanja može biti asimetričan i simetričan. U asimetričnoj varijanti, svaka strana može da izvrši operaciju raskidanja veze (DISCONNECT), pri čemu se dragoj strani šalje TPDU poruka DISCONNECT. Po stizanju poruke, veza se raskida.

U simetričnoj varijanti, svaki smer saobraćaja zasebno se zatvara, nezavisno od onog dragog. Kada jedna strana izvrši operaciju raskidanja veze, to znači da ona nema više šta da šalje, ali je još uvek voljna da primi podatke od partnera. Prema ovom modelu, veza se raskida kada obe strane izvrše operaciju DISCONNECT.

Na slici 6-4 prikazan je dijagram stanja pri uspostavljanju i raskidanju veze pomoću navedenih osnovnih operacija. Za svaki prelazak uzrok je neki događaj - izvršavanje osnovne operacije od strane lokalnog korisnika prenosa ili pristizanje paketa. Ovde smo zbog jednostavnosti pretpostavili da se svaka TPDU poruka zasebno potvrđuje. Pretpostavili smo i da se koristi simetrično raskidanje veze, pri čemu ga začinje klijent. Vidite i sami da model uopšte nije složen. Stvarne modele razmotrićemo kasnije.



Slika 6-4. Dijagram stanja pri jednostavnom radu s vezom. Prelasci označeni kurzivno izazvani su pristizanjem paketa. Pune linije prikazuju redosled stanja klijenta. Isprekidane linije prikazuju redosled stanja servera.

6.1.3 Berkli utičnice

Opišimo sada ukratko drugu vrstu osnovnih operacija prenosa, tzv. utičnice (engl. *sockets*) koje se u Berkli UNIX-u (Berkeley UNIX) koriste za TCP. Te osnovne operacije široko se koriste za programiranje na Internetu. Prikazali smo ih na slici 6-5. One približno odgovaraju modelu iz našeg prvog primera, s tim što nude više mogućnosti i fleksibilnije su. Nećemo se ovde baviti odgovarajućim TPDU porukama. To će sačekati trenutak kada kasnije u poglavlju budemo obrađivali TCP.

Prve četiri osnovne operacije izvršavaju serveri navedenim redom. Operacija SOCKET (utičnica) pravi novu utičnicu i za nju u transportnoj jedinici dodeljuje prostor u tabeli. Parametrima poziva zadaju se format adrese, vrsta usluge (npr. pouzdan tok bitova) i protokol. Uspešno izvedena operacije SOCKET vraća običan deskriptor datoteke za upotrebu pri sledećim pozivima, koji se koristi na isti način kao pri pozivanju procedure OPEN (otvori).

Osnovna operacija	Značenje
SOCKET	Pravi nov komunikacioni priključak (utičnicu)
BIND	Utičnici pridružuje lokalnu adresu
LISTEN	Objavljuje pristanak da prihvati veze; daje veličinu reda čekanja
ACCEPT	Blokira pozivaoca dok ne stigne zahtev za uspostavljanje veze
CONNECT	Aktivno pokušava da uspostavi vezu
SEND	Šalje podatke preko veze
RECEIVE	Prima podatke preko veze
CLOSE	Raskida vezu

Slika 6-5. Osnovne utičnice za TCP.

Nove utičnice nemaju mrežne adrese. One im se dodeljuju izvršavanjem usluge BIND (poveži). Kada server utičnici pridruži adresu, na nju se tada mogu priključivati udaljeni klijenti. Operacija SOCKET ne pridružuje odmah adresu utičnici jer bi to neke procese omelo (npr. one koji istu, opštepoznatu adresu koriste godinama).

Zatim se poziva osnovna operacija LISTEN (osluškuj), koja za dolazne pozive do- deljuje prostor u redu čekanja za slučaj da istovremeno više klijenata pokušava da se poveže. Za razliku od našeg prvog primera, u modelu utičnica operacija LISTEN ne blokira server.

Da bi se blokirao dok ne stigne neki poziv, server poziva operaciju ACCEPT (prihvati). Kada stigne TPDU zahtev za povezivanje, transportna jedinica pravi novu utičnicu koja ima ista svojstva kao i prethodna, i vraća njen deskriptor. Server tada može ostaviti jednu granu procesa ili programske niti da obrađuje novu utičnicu, a sam se vratiti na stara i nastaviti da čeka nove pozive. Operacija ACCEPT vraća standardan deskriptor datoteke koji se može koristiti pri čitanju i upisivanju, kao kod normalnih datoteka.

Pogledajmo sada šta se dešava kod klijenta. I tu se najpre mora napraviti utičnica pozivanjem operacije SOCKET, ali operacija BIND nije potrebna jer dodeljena adresa servera ništa ne znači. Operacija CONNECT (poveži) blokira pozivaoca i aktivno započinje proces povezivanja. Kada se proces završi (tj. kada se od servera dobije odgovarajuća TPDU poruka), klijentski server se deblokira i veza je uspostavljena. Obe strane sada mogu da pomoću usluga SEND (šaljem) i RECEIVE (primam) šalju i primaju podatke potpunom dupleksnom vezom. Mogu se koristiti i standardni UNIX-ovi sistemski pozivi procedurama READ i WRITE ukoliko nisu potrebne specijalne opcije koje imaju operacije SEND i RECEIVE.

Veza između utičnica raskida se simetrično, kada obe strane izvrše operaciju CLOSE.

6.1.4 Primer programiranja utičnica: server datoteka na Internetu

Primer korišćenja utičnica možete da vidite u klijentsko-serverskom kodu prikazanom na slici 6-6. Tu imamo sasvim elementaran server datoteka na Internetu i jednog klijenta koji ga koristi. Prikazani kod ima mnoga ograničenja (o kojima govorimo u nastavim), ali se njegov serverski deo može prevesti i izvršavati na svakom UNIX sistemu na Internetu. Posle toga se može prevesti klijentski deo koda i izvršavati na svakom

dragom UNIX računana širom sveta. Kada se klijentski kod izvrši uz odgovarajuće parametre, sa servera se može preuzeti svaka datoteka kojoj on ima pristup. Datoteka se šalje na standardni izlaz koji se, naravno, može preusmeriti u datoteku ili kanal.

```
/* Ova strana sadrži klijentski program koji može da zahteva datoteku od serverskog programa
* prikazanog na sledećoj strani. Server odgovara šaljući celu datoteku.
7
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

                                        /* proizvoljan broj priključka, ali se klijent i server
                                        moraju složiti oko njega 7 /* veličina prenosnog
                                        bloka 7

#include <netdb.h>
#define SERVEFLPORT 12345

#define BUF_SIZE 4096 int main(int argo, char **argv)
{
  int c, s, bytes; char buf[BUF_SIZE]; struct
hostent *h; struct sockaddrjn channel;          /* bafer za dolazne datoteke 7

                                        /*
                                                informacija o serveru
                                        7 /* sadrži IP adresu 7

  if (argc l= 3) fatal(„Uputstvo: klijent ime_servera ime_datoteke „); h =
  gethostbyname(argv[1]);                       /*    traženje IP adrese računara 7
  if (lh) fatal(„server nije pronađen");
  s = socket(PFJNET, SOCK STREAM, IPPROTO_TCP); if (s <0) fatal(„utičnica‘j;
  memset(&channel, 0, sizeof(channel)); channel.sin3family= AFJNET;
  memcpy(&channel.sin_addr.s_addr, h->h_addr, h->hjength); channel.sin_port=
  htons(SERVER_PORT);

  c = connect(s, (struct sockaddr *) &channel, sizeof(channel)); if (c < 0) fatal(„veza
  nije uspostavljena");

  I*      Veza je sada uspostavljena. Pošalji ime datoteke s bajtom 0 na kraju. 7
  write(s, argv[2], strlen(argv[2])+1);
  /* Uzmi datoteku i upiši je na standardni izlaz. 7 while (1) {
       bytes = read(s, buf, BUFJ3IZE);          /* čitaj sa utičnice 7
       if (bytes <= 0) exit(0);                 /*  provera kraja datoteke 7
       write(1, buf, bytes);                                        /*piši na standardni izlaz 7
  }
}
fatal(char *string)
{
  printf(„%s\n", string); exit(1);
}
```

Slika *6-6*. Klijentski deo koda sa utičnicama. Serverski deo koda je na sledećoj strani.

```
#include <sys/types.h>                    /* Ovo je serverski deo koda 7
#include <sys/fcntl.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#define SERVER_PORT 12345                 #define QUEUE_SIZE 10
#define BUF_SIZE 4096                     int main(int argo, char *argv[]) /* proizvoljan broj, ali se server i klijent moraju oko njega složiti 7 /* veličina prenosnog bloka 7
  int s, b, l, fd, sa, bytes, on = 1;     /* bafer za datoteke koje se šalju 7
  charbuf[BUF_SIZEj; struct               /* sadrži IP adresu 7
  sockađdrjn channel;
/* Gradi adresnu strukturu za pridruživanje utičnici. 7 memset(&channel, 0,
sizeof(channel)); /* kanal veličine nula 7 channel. sin_family = AFJNET; channel.
sin_addr.s_addr = htonl(INADDR_ANY); channel.sin_port =
htons(SERVER_PORT);
/* Pasivno otvoren. Čeka na vezu. 7
s = socket(AFJNET, SOCK_STREAM, IPPROTO_TCP); /* pravi utičnicu 7 if (s <
0) fatal(„utičnica nije napravljena“);
setsockopt(s, SOL_SOCKET, SO_REUSEADDR, (char *) &on, sizeof(on));
b = bind(s, (struct sockaddr *) &channel, sizeof(channel)); if (b < 0) fatai(„nije
uspelo pridruživanje adrese“);
l =   listen(s, QUEUE_SIZE);                                    /* zadaj  veličinu  reda
čekanja 7
if (l   < 0) fatal(„neuspešan poziv proceduri listen „);
/* Utičnica sa adresom sada je uspostavljena. Čekaj na vezu i obradi je. 7 while
(1) {
    sa = accept(s, 0, 0);                 /*  blokiranje i čekanje na zahtev za vezu */
    if (sa < 0) fatal(„nije uspelo prihvatanje veze“);
    read (sa, buf, BUF_SIZE);             /* učitaj ime datoteke sa utičnice */
    /* Uzmi i pošalji datoteku. */
    fd = open(buf, 0_RDONLY);             /*    otvori datoteku koja će biti poslata */
    if (fd < 0) fatal(„nije uspelo otvaranje“);
    while (1) {
        bytes = read(fd, buf, BUF_SIZE); /* čitaj iz datoteke */ if (bytes <= 0)
        break;                            /*    provera kraja datoteke 7
        write(sa, buf, bytes);            /*    piši bajtove na utičnicu 7
    }
    close(fd); /* zatvori datoteku 7
    close(sa); /* raskini vezu 7
```

Pogledajmo najpre serverski deo koda. On počinje nekim standardnim zaglavljima, od kojih tri poslednja sadrže glavne definicije i strukture podataka koji se odnose na Internet. Zatim dolazi definicija vrednosti (12345) serverskog priključka *(SER VER_PORT)*. Taj broj je proizvoljno izabran. Za priključak je dobar i svaki broj između 1024 i 6.5535 ukoliko ga ne koristi neki drugi proces. Naravno, klijent i server moraju da koriste isti priključak. Ako server ikada postane svetski poznat (što nije verovatno, s obzirom na njegovu primitivnost), biče mu dodeljen stalni broj priključka (manji od 1024) koji će se objaviti na adresi www.iana.org.

U sledeca dva reda serverskog koda definišu se dve neophodne konstante. Prva određuje veličinu elementarne jedinice (engl. *chunk)* za prenos datoteka. Drugom se zadaje maksimalan broj veza koje istovremeno mogu čekati na uspostavljanje; kada se on dostigne, odbacuje se svaki sledeći zahtev za uspostavljanje veze.

Posle deklarisanja lokalnih promenljivih, započinje sam kod. Prvo se inicijalizuje struktura podataka rezervisana za čuvanje IP adrese servera. Ona će uskoro biti povezana sa serverslcom utičnicom. Poziv proceduri *memset* postavlja sve vrednosti strukture podataka na nulu. Njena polja se popunjavaju pomoću tri dodele koje slede. Poslednje polje sadrži serverski priključak. Funkcije *htonl* i *htons* pretvaraju vrednosti u standardan format tako da se kod može ispravno izvršavati na računarima koji rade kako s formatom „big-endian“ (npr. na sistemu SPARC), tako i s formatom „lit- tle-endian“ (npr. na Pentiumu). Detalji semantike ovde nisu bitni.

Posle toga, server pravi utičnicu i proverava greške (što je označeno sa $s < 0$). U proizvodnoj verziji koda, poruka o grešci mogla bi biti i opširnija. Pozivanje procedure *setsockopt* neophodno je da bi se priključak mogao iznova koristiti, tako da server može stalno da radi prihvatajuci jedan zahtev za drugim. Sada se IP adresa pridružuje utičnici i proverava se da li je uspeo poziv proceduri *bind.* U poslednjoj fazi inicijalizacije, poziva se procedura *listen* kojom se objavljuje spremnost servera da prihvati dolazne pozive i sistemu nalaže da na čekanju drži najviše *QUEUE_SIZE* poziva dok on ne obradi tekuči poziv. Ako je red čekanja već pun, novi pozivi se ćutke odbacuju.

Sada server ulazi u svoju glavnu petlju koju nikada ne napušta. Ona se može prekinuti samo uništavanjem procesa spolja. Poziv proceduri *accept* blokira server sve dok neki klijent ne pokuša da s njim uspostavi vezu. Ako to uspe, procedura *accept* vrača deskriptor koji se može koristiti za čitanje i upisivanje, baš kao što se deskrip- tori datoteka koriste za učitavanje iz kanala i upisivanje u njega. Međutim, za razliku od programskih kanala (cevi) koji su jednosmerni, utičnice rade u oba smera, tako da se *sa* (adresa utičnice) može koristiti i za čitanje s veze i za upisivanje u nju.

Pošto se veza uspostavi, server s nje učitava ime datoteke. Ako ono još nije stiglo, server se blokira i čeka ga. Kada ga dobije, server otvara datoteku i ulazi u petlju u kojoj naizmenično čita blokove iz datoteke i zapisuje ih u utičnicu sve dok je celu ne kopira. Server tada zatvara datoteku, raskida vezu, i čeka zahtev za uspostavljanje nove veze. On se stalno vrti u toj petlji.

Pogledajmo sada klijentski deo koda. Da biste razumeli kako radi, treba da shvatite kako se poziva. Ako se kod, na primer, zove *client,* najčešće mu se upućuje sledeći poziv

```
client flits.cs.vu.nl /usr/tom/datoteka >f
```

Taj poziv će uspeti samo ako se server već izvršava na računam *flits.cs.vu.nl,* na kome postoji datoteka */usr/tom/datoteka,* kojoj je server već pristupio. Tada će datoteka biti poslata Internetom i biti zapisana u datoteku /, posle čega se klijentski program završava. Pošto server posle slanja nastavlja s radom, klijent se može ponovo pokretati da bi zahtevao i druge datoteke.

Klijentski kod počinje direktivama INCLUDE (za uključivanje drugih programskih modula) i deklaracijama protnenljivih. Izvršavanje počinje proverom sintakse poziva *(argc* = 3 znači ime programa uz dva argumenta). Obratite pažnju na to da urgv[l] sadrži ime servera (npr.*flits.cs.vu.nl)* i da se u IP adresu pretvara funkcijom *gethostbyname.* Ona ime dobija sa DNS servera. O sistemu DNS govorićemo u 7. poglavlju.

Zatim se pravi i inicijalizuje utičnica. Posle toga, klijent pokušava da uspostavi TCP vezu sa serverom služeći se procedurom *connect.* Ako na pozvanom računam postoji i radi server vezan za priključak *SERVER_PORT,* a povrh toga je i slobodan ili ima mesta u redu čekanja obrazovanom procedurom *listen,* veza će se (na kraju) uspostaviti. Koristeći uspostavljenu vezu, klijent šalje ime datoteke prepisujući ga na utičnicu. Broj poslatih bajtova zajedanje veći od potrebnog, pošto se na kraju mora poslati i nula da bi server znao gde je kraj imena datoteke.

Sada klijent ulazi u petlju, učitava sa utičnice blok po blok datoteke i kopira je na standardni izlaz. Kada to uradi, samo završi program.

Procedura *fatal* štampa poruku o grešci i završava se. Ista procedura je potrebna i za server, ali za nju nije bilo mesta na stranici. Pošto se klijent i server prevode odvojeno i obično izvršavaju na različitim računarima, ne mogu da dele isti *kad fatal.*

Dva opisana programa (kao i drugi materijal koji se odnosi na ovu knjigu) možete preuzeti sa Web lokacije posvećene knjizi:

http://www.prenhall.com/tanenbaum

tako što ćete pritisnuti hipervezu Web Site pored fotografije korica. Programe možete preuzeti, a zatim prevesti na bilo kom sistemu UNIX (Solaris, BSD, Linux) komandama

```
cc -o client client.c -lsocket -lnsl
cc -o server s erver.c -lsocket -lnsl
```

Server pokrećete komandom

```
server
```

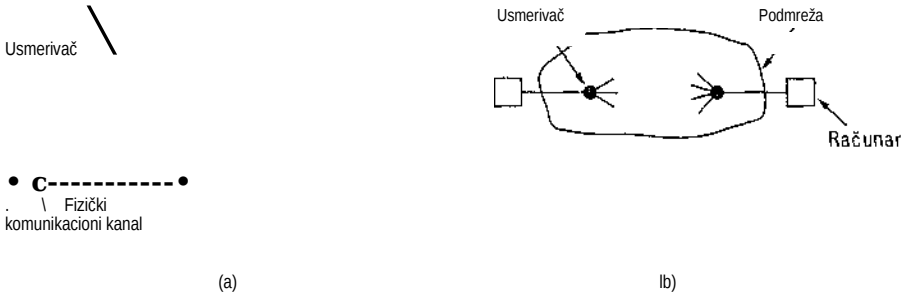
Kao što smo ranije naveli, za pokretanje klijenta potrebna su i dva argumenta. Na Web lokaciji naći ćete i verziju za Windows.

Imajte na umu da serverski kod nije poslednja reč tehnike. Njegov potprogram za proveru grešaka sasvim je „tanak“, a izveštaji šturi. U njemu nema pomena o bezbed- nosti, a korišćenje UNIX-ovih sistemskih procedura omogućava samo osnovnu nezavisnost od platforme. Program je napisan uz neke pretpostavke koje u tehničkom smislu nisu baš čvrste, npr. da ime datoteke staje u bafer i da se prenosi kao jedinstvena (nedeljiva) celina. Pošto sve zahteve obrađuje sekvencijalno (jer ima samo jednu programsku nit), performanse servera su slabe. Pa ipak, uz sve svoje nedostatke, to je potpun, funkcionalan server datoteka za Internet. U vežbama na kraju poglavlja čitalac će dobiti priliku da ga usavrši. Više informacija o programiranju utičnica nadi ćete kod Stevensa (1997).

6.2 ELEMENTI TRANSPORTNIH PROTOKOLA

Usluga prenosa se ostvaruje pomoću **transportnog protokola** (engl. *transport proto col)* koji povezuje dve transportne jedinice. Transportni protokoli na neki način liče na protokole sloja veze, koje smo opisali u 3. poglavlju. Protokoli obe vrste - između ostalog - treba da sprovedu kontrolu grešaka, da šalju podatke određenim redo- sledom i da upravljaju tokom.

Međutim, između njih postoje i upadljive razlike koje potiču od drugačijeg okruženja u kojima protokoli rade (slika 6-7). U sloju veze podataka, usmerivači međusobno komuniciraju direktno preko fizičkog kanala, dok u transportnom sloju taj fizički kanal predstavlja čitava podmreža. Ova razlika ima važne posledice po rad protokola, kao što ćemo videti u nastavku.



Slika 6-7. a) Okruženje sloja veze podataka, (b) Okruženje transportnog sloja.

Kao prvo, u sloju veze podataka usmerivač ne mora da naznači drugi usmerivač s kojim želi da komunicira jer svaka izlazna linija vodi samo do jednog usmerivača. U transportnom sloju, međutim, adresa odredišta mora se jasno zadati.

Drugo, postupak uspostavljanja veze preko žice na slici 6-7(a) sasvim je jednostavan: odredište je uvek na dragom kraju (osim ako je u kvara, u kom slučaju ga nema). Bilo kako bilo, tu nema mnogo posla. Uspostavljanje veze u transportnom sloju za- hteva više pripreme, kao što ćemo ubrzo videti.

Sledeća razlika između sloja veze podataka i transportnog sloja koja postaje sve neprijatnija odnosi se na potencijalni skladišni kapacitet podmreže. Kada usmerivač pošalje okvir, on može da stigne ili da se izgubi, ali ne može da pluta neznano gde, da

se privremeno sakrije i da se posle tridesetak sekundi pojavi baš u najnezgodnijem trenutku. Ako, međutim, datagramska podmreža za interno usmeravanje koristi prilagodljive algoritme, ipak postoji verovatnoca da će paket negde u njoj jedno vrerne biti uskladišten, a zatim kasnije isporučen. Posledice sposobnosti podmreže da privremeno skladišti pakete mogu ponekad biti pogubne, zbog čega se moraju koristiti specijalni protokoli.

Poslednja razlika između sloja veze podataka i transportnog sloja vise je kvantitativna nego kvalitativna. U oba sloja neophodni su privremeno skladištenje (baferovanje) i kontrola toka, ali zbog prisustva velikog i dinamički promenljivog broja veza, u transportnom sloju je možda potreban drugačiji pristup nego u sloju veze podataka. Kao što smo videli u 3. poglavlju, neki od protokola svakoj liniji dodeljuju fiksan broj bafera, tako da za pristigli paket uvek postoji slobodan bafer. U transportnom sloju, zbog većeg broja veza s kojima se istovremeno mora raditi, takva ideja je manje privlačna. U narednim poglavljima bavićemo se detaljnije svim navedenim, a i drugim problemima.

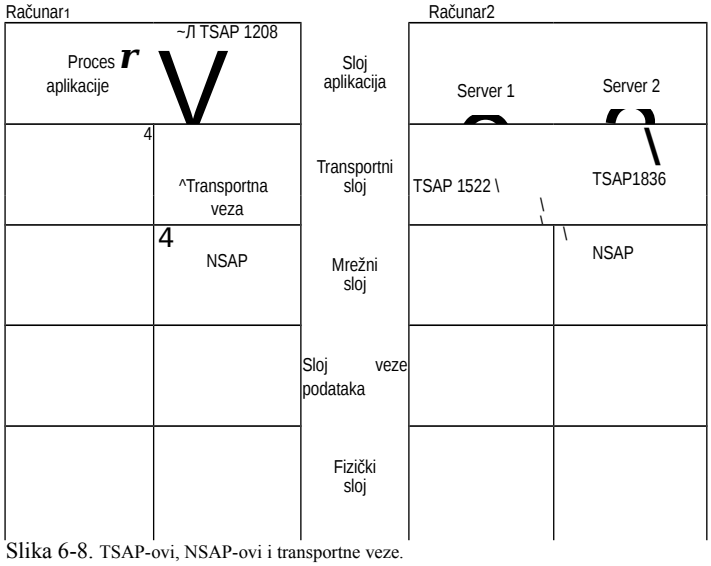
6.2.1 Adresiranje

Kada aplikacija (tj. korisnički proces) poželi da uspostavi vezu s procesom neke udaljene aplikacije, ona mora da ga eksplicitno navede. (Prenos bez uspostavljanja direktne veze ima isti problem: Kome poslati poruku?) Uobičajeni postupak podrazume- va navođenje adresa procesa koji treba da osluškuju zahteve za uspostavljanje veze. Takve krajnje tačke na Internetu nazivaju se priključcima (engl. *ports)*. U ATM mrežama one se zovu AAL-SAP -ovi (uslužne pristupne tačke ATM sloja za adaptaciju). Nadalje ćemo koristiti opšti izraz *TSAP* (pristupna tačka usluge prenosa, engl. *Transport Service Access Point)*. Odgovarajuće krajnje tačke u mrežnom sloju zovu se NSAP -ovi (uslužne pristupne tačke mrežnog sloja). Primer NSAP-a je IP adresa.

Slika 6-8 ilustruje odnose između NSAP-ova, TSAP-ova i transportne veze. Procesi aplikacija, i serverskih i klijentskih, mogu da se prikače na TSAP da bi ostvarile vezu sa udaljenim TS AP-om. Te veze na svakom računam, kao što se vidi, prolaze kroz NSAP- -ove. TS AP-ovi postoje zato što u nekim mrežama svaki računar ima samo jedan NSAP, pa je neophodno razlikovati više krajnjih transportnih tačaka koje dele taj NSAP.

Moguć je sledeći scenario povezivanja:

1. Serverski proces koji na zahtev distribuira tekuće vreme priključuje se na TSAP 1522 da bi čekao dolazne pozive. Kako on to čini nije stvar mrežnih protokola već isključivo lokalnog operativnog sistema. Čekanje se ostvaruje npr. već opisanim pozivanjem procedure LISTEN.
2. Proces aplikacije na računam 1 želi da utvrdi tekuće vreme, pa emituje zahtev CONNECT navodeći kao izvorište TSAP 1208, a kao odredište TSAP 1522. Ova akcija na kraju rezultuje uspostavljanjem veze između procesa aplikacije na računam 1 i servera 1 na računam 2.
3. Proces aplikacije tada šalje zahtev za tekuće vreme.
4. Server tekućeg vremena odgovara tako što šalje tekuće vreme.
5. Zatim se transportna veza raskida.



Slika 6-8. TSAP-ovi, NSAP-ovi i transportne veze.

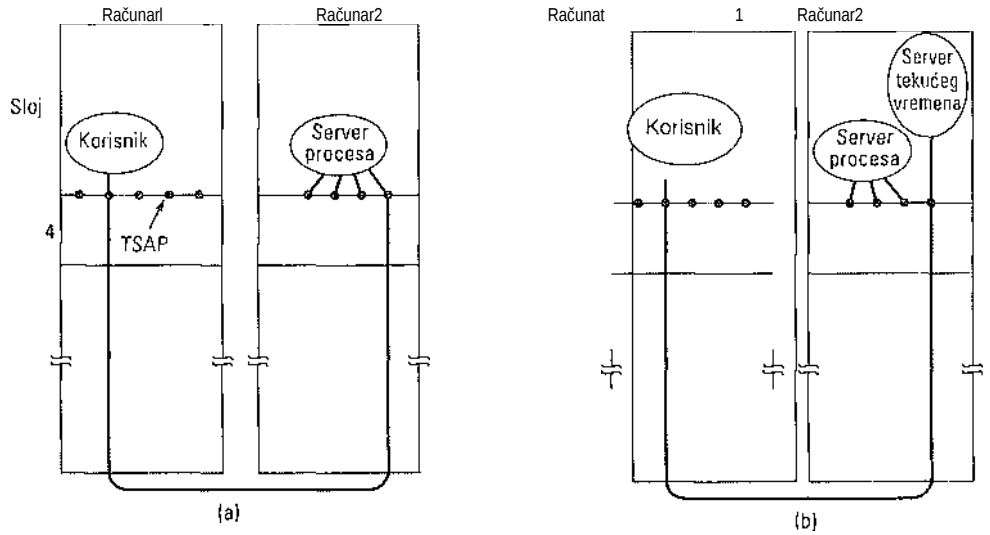
Obratite pažnju na to da na računaru 2 može biti i drugih servera, svaki sa svojim TSAP-om, lcoji čekaju na dolazne pozive preko istog NSAP-a.

Opisani scenario dobro izgleda, osim što smo sakrili jednu pojedinost: kako korisnički proces na računaru 1 zna da je server tekućeg vremena priključen na TSAP 1522? Jedna mogućnost je daje server tekućeg vremena već godinama priključen na TSAP 1522 i da su postepeno svi korisnici mreže to saznali. Prema ovom modelu, slične usluge imaju stabilne TSAP adrese popisane u datotekama na svima poznatim mestima, npr. u datoteci */etc/services* na sistemima UNIX, gde se nalaze serveri i odgovarajući priključci s kojima su trajno povezani.

Dok su stabilne TSAP adrese zgodne za mali broj ključnih usluga koje se nikada ne menjaju (npr. za Web servere), korisnički procesi u načelu žele da se povezuju s drugim korisničkim procesima koji kratko traju i nemaju unapred poznate TSAP adrese. Štaviše, ako ima potencijalno mnogo serverskih procesa, od kojih se većina retko koristi, rasipnički bi bilo da svaki od njih po ceo dan osluškuje stabilnu TSAP adresu. Ukratko, potreban je neki bolji sistem.

Jedan takav sistem prikazan je uprošćeno na slici 6-9. On je poznat kao protokol za početno povezivanje (engl. *initial connection protocol).* Umesto da svaki božji server osluškuje neku opštepoznatu TSAP adresu, svaki računar koji želi da ponudi usluge udaljenim korisnicima ima specijalan server procesa (engl. *process server)* koji zastupa slabije opterećene servere. On očeku je zahtev za povezivanje, istovremeno osluškujući više priključaka. Potencijalni korisnik usluge počinje tako što emituje zahtev CONNECT navodeći TSAP adresu usluge koju želi. Ako nijedan server ne odgovori na zahtev, veza se uspostavlja sa serverom procesa, kao na slici 6-9(a).

Pošto dobije zahtev, server procesa na licu mesta pravi zahtevani server i omogućuje mu da preuzme uspostavljenu vezu sa korisnikom. Taj server zatim odgovara na zahtev, a server procesa se vraća na osluškivanje novih zahteva, kao na slici 6-9(b).



Slika 6-9. Način na koji korisnički proces na računaru 1 uspostavlja vezu sa serverom tekućeg vremena na računaru 2.

Protokol za početno povezivanje radi odlično za servere koji se mogu praviti po potrebi, ali postoje mnoge situacije u kojima usluge postoje nezavisno od servera procesa. Server datoteka, na primer, mora da se izvršava na specijalnom hardvera (na računaru s diskom) i ne može se napraviti u hodu - onda kada neko poželi da se s njim poveže.

Takve situacije se često prevazilaze primenom drugog načina rada. Po tom modelu, postoji specijalan proces - server imena (engl. *name server)* ili server imenika (engl. *directory server).* Da bi pronašao TSAP adresu koja odgovara imenu određene usluge, npr. tekućeg vremena („time of day"), korisnik uspostavlja vezu sa serverom imena (koji osluškuje opštepoznati TSAP). Korisnik tada šalje poruku navodeći ime usluge, a server imena odgovara TSAP adresom. Zatim korisnik prekida vezu sa serverom imena i uspostavlja novu sa serverom željene usluge.

Prema ovom modelu, kada se ponudi nova usluga, ona se mora registrovati na servera imena svojim imenom (obično, u obliku tekstualnog niza) i svojom TSAP adresom. Server imena beleži ove podatke u svoju internu bazu podataka, tako da može uspešno odgovoriti na buduće upite.

Funkcija servera imena potpuno odgovara službi za brojeve pretplatnika na telefonskoj centrali (988) - obe obezbeđuju preslikavanje imena u brojeve. Kao u telefonskom sistemu, neophodno je da TSAP adresa servera imena (ili servera procesa u

protokolu početnog povezivanja) bude zaista opštepoznata. Ako ne znate telefonski broj službe informacija, ne možete je zvati da biste to saznali. Ukoliko, pak, mislite da je taj broj opštepoznat, isprobajte ga u nekoj stranoj zemlji.

6.2.2 Uspostavljanje veze

Uspostaviti vezu deluje lako, ali je u stvari veoma složeno. Na prvi pogled izgleda dovoljno da jedna transportna jedinica na odredište pošalje TPDU poruku CONNECTION REQUEST i zatim samo da čeka odgovor CONNECTION ACCEPTED. Problemi nastaju ako mreža može da izgubi, uskladišti ili duplira paket. Takvo ponašanje izaziva brojne komplikacije.

Zamislite podmrežu koja je toliko zagušena da potvrde gotovo nikada ne stižu na vreme, skoro svakom paketu ističe tajmer i on mora da se ponovo šalje još dva ili tri puta. Pretpostavite da podmreža interno koristi datagrame i da svaki paket sledi drugačiju putanju. Neki od njih mogu da se zaglave u saobraćajnoj gužvi i da zakasne, tj. da budu uskladišteni u podmreži i da znatno kasnije najednom vaskrsnu.

Najgore je ako se dogodi sledeće. Korisnik uspostavlja vezu s bankom, šalje banci nalog da prebaci veću sumu novca na račun osobe kojoj se ne može sasvim verovati, a zatim raskida vezu. Na nesreću, svaki paket poruke se duplira i skladišti u podmreži. Pošto se veza raskine, svi uskladišteni paketi ponovo se pojavljuju u podmreži i redom stižu na odredište zahtevajući od banke da uspostavi novu vezu, da (ponovo) prenese novac i zatim, da raskine vezu. Banka ne može da utvrdi da lije reč o duplikatima. Ona mora da pretpostavi daje reč o dragoj, nezavisnoj transakciji i da novac ponovo prenese. Do kraja ovog odeljka bavićemo se problemom zakasnelih duplikata s posebnim osvrtom na algoritme za uspostavljanje pouzdanih veza, tako da košmari slični opisanom ne mogu da se dogode.

Srž problema je postojanje zakasnelih duplikata. Oni se mogu napasti na različite načine, ali nijedan nije sasvim delotvoran. Jedan način je da se koriste jednokratne transportne adrese. Drugim recima, adresa se generiše svaki put kad je potrebna. Kada se veza raskine, adresa se odbacuje i više se ne koristi. Takva strategija onemogućuje primenu servera procesa sa slike 6-9.

Druga mogućnost je da inicijator veze svakoj vezi pridruži izabrani identifikator (tj. redni broj koji se uvećava za jedan sa uspostavljanjem svake nove veze) koji je dužan da stavlja u svaku TPDU poruku, uključujući i poruku kojom zahteva uspostavljanje veze. Pošto se veza raskine, svaka transportna jedinica može da ažurira tabelu zastarelim odrednicama sledećeg oblika: druga transportna jedinica, identifikator veze. Posle toga, kad god pristigne zahtev za uspostavljanje veze, on se može sravniti s tabelom i odbaciti ukoliko pripada vezi koja je ranije raskinuta.

Opisana šema, nažalost, ima jedan fatalan propust. Ona zahteva da svaka transportna jedinica neograničeno održava određenu količinu informacija o prethodno uspostavljenim vezama. Ako računar otkaže, sve će zaboraviti i više neće znati koji su identifikatori veza već korišćeni.

Zbog toga smo prinuđeni da izaberemo neki drugi način. Umesto da pakete pustimo da u mreži lutaju neograničeno, moramo smisliti mehanizam da ih pravovremeno uništimo. Ako obezbedimo da paketi ne žive duže od određenog vremena, problem se lakše može razrešiti.

Životni vek paketa može se ograničiti primenom jedne od sledecih tehnika (ili više njih):

1. Odgovarajućim projektovanjem mreže.
2. Smeštanjem brojača skokova u svaki paket.
3. Vremenskim označavanjem svakog paketa.

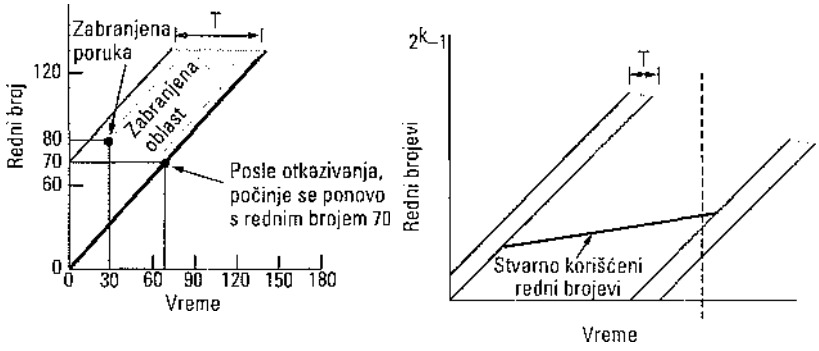
Pod odgovarajućim projektovanjem podrazumevamo mrežu bez petlji, u kojoj se kašnjenje prouzrokovano zagušenjem može izračunati kao vreme putovanja najdužom mogućom putanjom. Druga tehnika predviđa da se brojač skokova na početku postavi na odgovarajuću vrednost koja će se smanjivati za jedan pri svakom skoku. U tom slučaju, mrežni protokol će jednostavno odbacivati pakete čiji brojač padne na nulu. Treća tehnika podrazumeva da svaki paket nosi vreme nastanka, tako da ih usmerivači mogu odbaciti posle nekog, unapred dogovorenog vremenskog perioda. Ako se ona primeni, moraju se sinhronizovati sistemski satovi usmerivača što već nije jednostavno, osim ako se za sinhronizovanje ne iskoristi neki spoljni mehanizam, kao što je GPS ili radio-stanica koja periodično difuzno emituje tačno vreme.

U praksi, treba da uništimo ne samo paket, već i sve njegove potvrde, tako da ćemo uvesti T - mali umnožak maksimalnog životnog veka paketa. Vrednost T zavisi od primenjenog protokola, a uloga mu je da nam ostavi dovoljno vremena za uništenje. Ako, nakon što je paket poslat, sačekamo vreme T, bićemo sigurni da su nestali svi njegovi tragovi i da ni on, ni njegove potvrde ne mogu više da ometaju saobraćaj.

Kada ograničimo životni vek paketa, to nam pruža mogućnost da smislimo način za pouzdano uspostavljanje veze. Postupale koji opisujemo u nastavku potiče od Tom- linsona (1975). On rešava problem, ali uvodi i neke svoje specifičnosti. Taj postupak dalje su razradili Sunshine i Dalal (1978). Njegove varijante široko se koriste u praksi, između ostalog, i u protokolu TCP.

Da bi računari posle otkazivanja mogli da se snađu, Tomlinson je predložio da svaki bude opremljen satom tekućeg vremena. Satovi svih računara treba da budu sinhro- nizovani. Satovi su u obliku binarnog brojača čija se vrednost povećava za jedan u jednakim vremenskim razmacima. Osim toga, broj bitova u brojaču mora da bude jednak ili veći od broja bitova u rednim brojevima poruka. I na kraju, najvažnije: pretpostavlja se da sat nastavlja da radi i kada računar otkaže.

Tomlinsonov mehanizam u osnovi treba da obezbedi da se dve TPDU poruke sa istim rednim brojem nikada ne nađu u isto vreme na mreži. Kada se veza uspostavi, za početni redni broj (k bitova) koristi se k najmanje značajnih bitova sata. Na taj način, za razliku od naših protokola iz 3. poglavlja, svaka veza započinje obeleža- vanje svojih TPDU poruka različitim početnim rednim brojem. Rednih brojeva treba da ima barem toliko da su u momentu kada se s poslednjeg broja ponovo pređe na prvi sve TPDU poruke sa istim rednim brojevima davno iščezle. Linearni odnos između vremena i početnih rednih brojeva prikazan je na slici 6-10.



Slika 6-10. (a) TPDU poruke ne smeju da se nađu u zabranjenom području. (b) Problem ponovnog sinhronizovanja.

Kada se obe transportne jedinice slože o početnom rednom broju, za kontrolu toka podataka može se iskoristiti bilo koji protokol kliznih prozora. Zavisnost početnih rednih brojeva od vremena (prikazana na slici zadebljanom linijom) u stvarnosti nije linearna, već stepenasta, pošto sat menja vrednost u diskretnim vremenskim intervalima. Taj detalj smo zanemarili da bismo jače istakli suštinu.

Kada računar otkaže, nastaje problem, jer u trenutku kada se ponovo uključi, njegova transportna jedinica ne zna gde joj je mesto u nizu rednih brojeva. Jedno rešenje je da transportna jedinica tada miruje tokom intervala *T* i tako dozvoli da s mreže iščeznu sve stare TPDU poruke. Međutim, ako je mreža velika i složena, *T* može da ima veliku vrednost, pa takvo rešenje nije baš efikasno.

Da bi se izbeglo da računar, pošto se povrati od havarije, čeka tokom intervala *T,* bilo je potrebno dodatno ograničiti korišđenje rednih brojeva. Potrebu tog ograničenja najbolje ćete sagledati na jednom primeru. Neka *T,* maksimalni životni vek paketa, bude 60 sekundi i neka sat otkucava jednom u sekundi. Sledeći zadebljanu liniju na slici 6-10(a), početni redni broj za vezu uspostavljenu u trenutku *x* biće *x.* Zamislite da je u trenutku *t* = 30 s, jednom običnom TPDU paketu s podacima koji je poslat (ranije uspostavljenom) vezom .5, dat redni broj 80. Nazovimo taj paket *X.* Čim je poslao TPDU paket *X,* računar otkazuje, a zatim se brzo oporavlja. U trenutku *t* = 60 s, on počinje da ponovo uspostavlja veze 0 do 4. U trenutku *t* = 70 s, on ponovo uspostavlja vezu 5 i, kao što treba, koristi redni broj 70. U toku sledećih 15 s, on šalje TPDU pakete s podacima 70 do 80. Tako, u trenutku *t* = 85 s, na mreži se pojavljuje nov TPDU paket s rednim brojem 80 i vezom 5. Nažalost, TPDU Xjoš uvek postoji. Da je stigao do primaoca pre novog TPDU paketa 80, TPDU paket X bi bio prihvaćen, a ispravan TPDU paket 80 bio bi odbačen kao duplikat.

Da bi se izbegli slični problemi, između normalnog korišćenja rednih brojeva (njihovog dodeljivanja novim TPDU paketima) i njihovog ponovnog korišćenja kao početnih rednih brojeva, mora da protekne vreme *T.* Nelegalne kombinacije vremena i rednih brojeva na slici 6-10(a) označene su kao zabranjena oblast (engl.*forbidden regiori).* Pre nego što pošalje TPDU paket bilo kojom vezom, transportna jedinica uvek mora da očita sat i da proveri da li

se nalazi izvan zabranjene oblasti.

Protokol može sam sebe da satera u ćorsokak na dva jasno defmisana načina. Ako računar novouspostavljenom vezom šalje previše podataka prevelikom brzinom, brzina „trošenja“ rednih brojeva može znatno da premaši brzinu potencijalnog dodeljivanja početnih rednih brojeva. To znači da maksimalnu brzinu prenosa podataka bilo kojom vezom treba ograničiti na jedan TPDU paket po otkucaju sata. To takođe znači da transportna jedinica mora da sačeka jedan otkucaj sata pre nego što uspostavi novu vezu za računar koji se upravo oporavio, kako se isti redni broj ne bi još jednom koristio. Oboje govori u prilog bržeg otkucavanja sata (svakih nekoliko mikrosekundi ili češće).

Nažalost, prebrzo slanje podataka i ulazak u zabranjenu oblast odozdo nije i jedini način da se zapadne u nevolju. Sa slike 6-10(b) vidimo da će pri svakoj brzini slanja koja zaostaje za otkucajima sata zavisnost aktuelno korišćenih rednih brojeva na kraju ući u zabranjenu oblast sleva. Što je veći nagib te zavisnosti, to će se ulazak desiti kasnije. Kao što smo već naveli, pre nego što pošalje TPDU paket, transportna jedinica uvek mora da proveri da li će ući u zabranjenu oblast i ako to utvrdi, da sačeka sa slanjem tokom vremena *T*ili da se ponovo sinhronizuje u pogledu rednih brojeva.

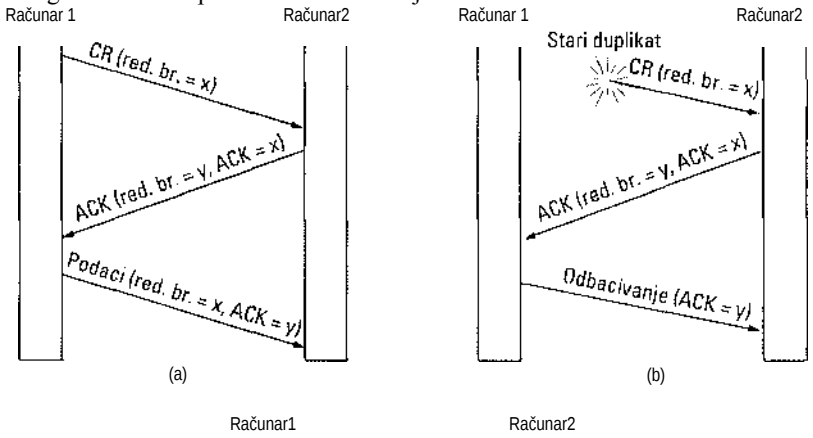
Upotreba sata rešava problem zakasnelih TPDU duplikata s podacima, ali se prvo mora uspostaviti veza. Pošto mogu kasniti i upravljački TPDU paketi, potencijalno može nastati problem usklađivanja dve strane oko početnog rednog broja. Pretpostavimo da se veze uspostavljaju tako što računar 1 udaljenom računaru 2 šalje TPDU zahtev CONNECTION REQUEST s predloženim početnim rednim brojem i brojem odredišnog priključka. Računar 2 potvrđuje zahtev šaljući nazad TPDU paket CONNECTION ACCEPTED. Ako se TPDU zahtev CONNECTION REQUEST usput izgubi, a računar 2 odjednom primi njegov zakasneli duplikat, veza će se uspostaviti neispravno.

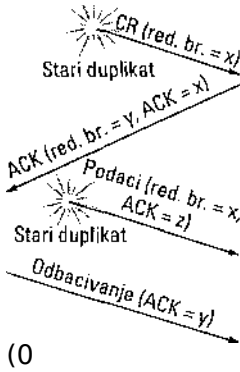
Da bi rešio opisani problem, Tomlinson je (1975) uveo tzv. trostepeno usaglaša- vanje (engl. *three-way handshake)*. Protokol ne zahteva da obe strane emisiju započnu istim rednim brojem, tako da za njegov rad nije neophodno sinhronizovanje prema globalnom vremenu. Na slici 6-11 (a) prikazanje normalan postupak uspostavljanja veze, pri čemu je inicijator računar 1. Računar 1 bira redni broj *x* i šalje ga računaru 2 sa TPDU zahtevom CONNECTION REQUEST. Računar 2 odgovara TPDU potvrdom kojom prihvata *x* i najavljuje svoj početni redni broj *y.* Na kraju, računar 1 potvrđuje da prihvata početni redni broj računara 2 u prvom TPDU paketu s podacima koji mu šalje.

Pogledajmo sada kako radi trostepeno usaglašavanje kada postoji zakasneli upravljački TPDU duplikat. Prvi TPDU paket na slici 6-11(b) predstavlja zakasneli duplikat CONNECTION REQUEST potekao iz stare veze. Taj TPDU paket stiže računaru 2 bez znanja računara 1. Računar 2 reaguje tako što računaru 1 šalje TPDU potvrdu (ACK) kojom, u stvari, i on traži potvrdu da računar 1 zaista želi da uspostavi novu vezu. Kada računar 1 odbije pokušaj računara 2 da uspostavi vezu, računar 2 shvata da gaje prevario zakasneli duplikat i odustaje od povezivanja. Na taj način, zakasneli duplikat ne pravi nikakvu štetu.

Najgore je kada podmrežom istovremeno lutaju paketi CONNECTION REQUEST i ACK. Taj slučaj je prikazan na slici 6-11 (c). Kao u prethodnom primeru, računar 2 do- bija zakasneli paket CONNECTION REQUEST i odgovara na njega. U ovom trenutku treba da uočite daje računar 2 predložio *y* kao početni redni broj za saobraćaj između računara 2 i računara 1, uverivši se prethodno da više nema ni TPDU paketa, ni TPDU potvrda s tim rednim brojem. Kada računam 2 stigne dragi zakasneli TPDU duplikat, on zna daje to stari

duplikat iz činjenice daje za početni redni broj prihvac'eno *z,* ume- sto *y.* Skrećemo vam pažnju na to da ne postoji kombinacija TPDU paketa koja bi mogla da protokol dovede u škripac i da ga natera da uspostavi novu vezu koju u stvari niko ne želi.





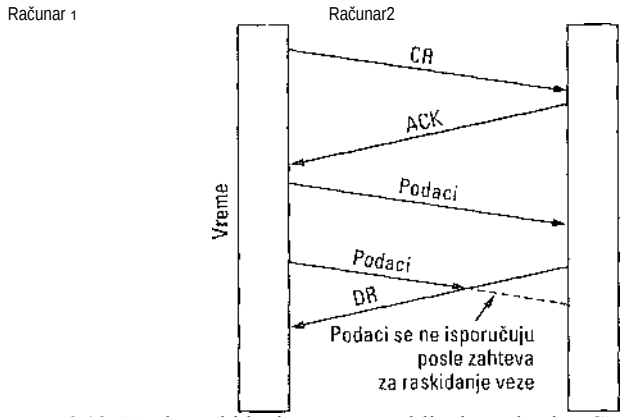
(0

Slika 6-11. Tri scenarija za uspostavljanje veze trostepenim usaglašavanjem. CR znači CONNECTION REQUEST; ACK znači potvrdu (ACKNOWLEDGMENT), (a) Normalan rad. (b) Stari duplikat CONNECTION REQUEST pojavljuje se neznano otkud, (c) Duplikat CONNECTION REQUEST i duplikat ACK.

6.2.3 Raskidanje veze

Lakše je raskinuti vezu nego je uspostaviti. Pa ipak, i tu ima više problema nego što se zamišlja. Kao što smo ranije naglasili, postoje dva stila raskidanja veze: asimetričan i simetričan. U telefonskom sistemu veza se raskida asimetrično: kada jedan od sago- vomika spusti slušalicu, veza se prekida. Kod simetričnog raskidanja, veza se smatra sistemom od dve paralelne jednosmerne veze od kojih se svaka mora zasebno raskinuti.

Asimetrično raskidanje je naglo i može da izazove gubitak podataka. Razmotrite scenario sa slike 6-12. Pošto se veza uspostavi, računar 1 šalje TPDU poruku koja ispravno stiže računam 2. Zatim računar 1 šalje drugu TPDU poruku. Nažalost, računar 2 emituje poruku DISCONNECT pre stizanja te druge TPDU ponike. Veza se prekida i podaci se gube.



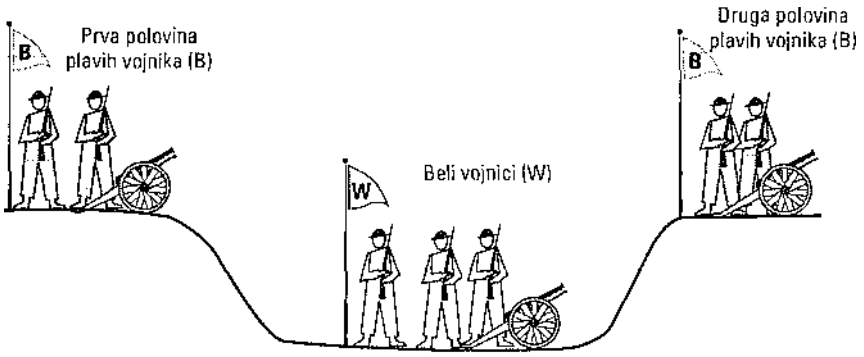
Slika 6-12. Naglo raskidanje veze uz gubljenje podataka. CR označava zahtev za uspostavljanje veze (CONNECTION REQUEST), ACK je potvrda (ACKNOWLEDGMENT), a DR je zahtev za raskidanje (DISCONNECT REQUEST).

Da bi se sprečilo gubljenje podataka, jasno je da je potreban složeniji protokol za raskidanje veze. Jedan način je simetrično raskidanje, u kome se nezavisno raskida svaka od dve jednosmerne veze. Tu računar može da prima podatke čak i pošto je sam rasldnuo vezu (za slanje drugom računam).

Simetrično raskidanje radi dobro kada svaki od dva procesa treba da pošalje fiksnu količinu podataka i tačno zna kada ih je sve poslao. U drugim slučajevima, određi vanje trenutka kada je posao završen, pa prema tome i kada veza treba da se raskine, nije tako očigledno. Možete da zamislite protokol u kome računar 1 kaže: „Ja sam završio. Jesi li i ti završio?“ Ako računar 2 odgovori: „1 ja sam završio, doviđenja, tada vezu možemo bezbedno raskinuti“.

Nažalost, ovaj protokol ne radi uvele. Postoji čuveni problem dve vojske (engl. *two-army problem),* koji ga ilustmje. Zamislite da su se beli vojnici ušančili u dolini (slika 6-13). Na oba okolna brda nalaze se plavi vojnici. Belih vojnika ima više nego plavih na svakom pojedinačnom brdu, ali je ukupan broj plavih vojnika veći od broja belih. Kada bi plavi vojnici s bilo kog brda sami krenuli u napad, bili bi poraženi, ali ako bi istovremeno napali sa obe strane, izvojevali bi pobedu.

Plavi vojnici s dva brda žele da sinhronizuju napad. Međutim, to mogu samo preko kurira koji treba da prođe kroz dolinu, gde može biti zarobljen, a poruka ostati neisporučena (tj. moraju da koriste nepouzdan komunikacioni kanal). Pitanje: postoji li protokol koji ce plavoj vojsci omogućiti da pobedi?



Slika 6-13. Problem dve vojske.

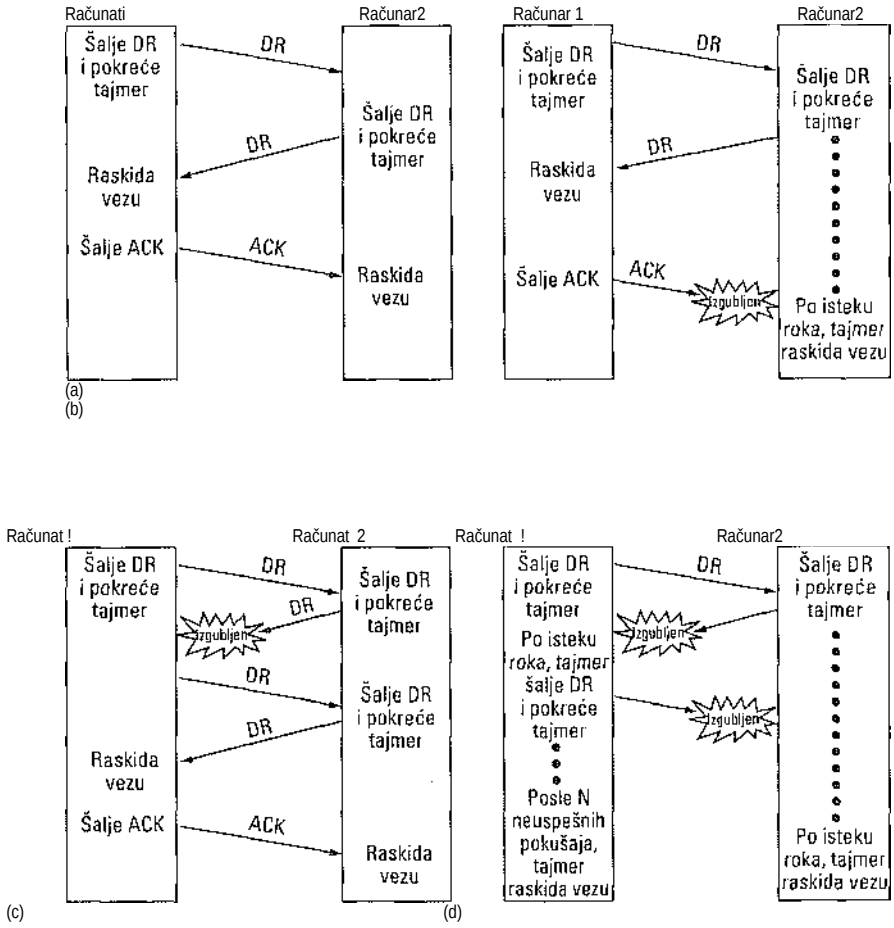
Pretpostavimo da komandant prve polovine plavih vojnika pošalje poruku koja glasi: „Predlažem da napadnemo 29. marta u zom. Šta misliš o tome?“ Pretpostavimo dalje da je ta poruka stigla, da se komandant drage polovine plavih vojnika složio s predlogom i da njegov odgovor bezbedno stiže komandantu prve polovine plavih vojnika. Da li će doći do napada? Verovatno da neće, jer komandant drage polovine plavih vojnika ne zna da lije njegov odgovor stigao dragoj strani. Ako nije stigao, to znači da prva polovina plavih vojnika neće napasti, pa bi bilo ludo da sam kreće u bitku.

Unapredimo sada protokol trostepenim usaglašavanjem. Inicijator prvog predloga mora da potvrdi daje dobio odgovor. Pretpostavljajući da se nijedna poruka ne gubi, draga polovina plavih vojnika dobiće potvrdu, ali će sada oklevati komandant prve polovine jer ne zna da li se njegova potvrda probila kroz dolinu, a ako nije, zna da druga polovina plavih vojnika neće napasti. Sada protokol možemo da usavršimo četvo- rostepenim, petostepenim usglašavanjem itd, ali ni to neće pomoći.

U stvari, može se dokazati da nema protokola kojim bi se problem rešio. Uprkos tome, pretpostavimo da takav protokol postoji. Poslednja poruka protokola može da bude bitna ili nebitna. Ako je nebitna, uklonimo je (kao i sve drage nebitne poruke), sve dok ne ostane protokol u kome su sve poruke bitne. Šta se dešava, ako se poslednja poruka ne probije do odredišta? Upravo smo rekli daje ona bitna, pa ako se izgubi, do napada neće doći. Pošto pošiljalac poslednje poruke nikada ne može biti siguran da je ona stigla, on neće rizikovati da napadne. Gore je što i druga polovina plavih vojnika to zna, pa ni oni neće napasti.

Da biste uočili sličnost problema dve vojske s problemom raskidanja veze, samo „napad“ zamenite „raskidanjem“. Ako nijedna od dve strane nije spremna da raskine vezu sve dok pouzdano ne sazna daje i draga strana spremna na to, do raskidanja nikada neće doći.

U praksi, međutim, situacija nije tako beznadežna, jer se pri raskidanju veze obično prihvata veći rizik nego pri pokretanju napada na bele vojnike. Na slici 6-14 prikazana su četiri scenarija raskidanja uz korišćenje trostepenog usaglašavanja. Iako taj protokol nije bez mana, obično je sasvim primeren svom zadatku.



Slika 6-14. Četiri scenarija s protokolom za raskidanje veze, (a) Normalan slučaj trostepenog usaglašavanja. (b) Gubi se poslednja ACK poruka, (c) Gubi se odgovor. (d) Gubi se i odgovor i naknadne DR poruke.

Na slici 6-14(a) vidimo normalnu situaciju u kojoj jedan od korisnika šalje TPDU poruku DR (DISCONNECTION REQUEST) i time inicira raskidanje veze. Kada ta poruka stigne, primalac takođe odgovara TPDU porukom DR i pokreće tajmer za slučaj da se DR usput izgubi. Kada ova DR poruka stigne, prvi pošiljalac vraća TPDU poruku ACK i raskida vezu. Na kraju, kada mu ACK stigne, i primalac raskida vezu. Raskidanje veze znači da transportna jedinica uklanja podatke o vezi iz svoje tabele s tekućim vezama i vlasniku veze (korisniku prenosa) na neki način to signalizira. Ova akcija se razlikuje od slučaja kada korisnik prenosa izvrši osnovnu operaciju DISCONNECT.

Ako se završna TPDU poruka ACK izgubi, kao na slici 6- 14(b), situaciju spasava tajmer. Veza se automatski prekida kada mu istekne rok.

Razmotrimo sada slučaj kada se izgubi druga DR poruka. Korisnik koji je inicirao raskidanje veze neće dobiti očekivani odgovor, tajmer će mu isteći i sve će početi iz početka. Kako to radi, vidimo na slici 6-14(c), pri čemu se drugi put ne gubi nijedna puruka, već sve

stižu ispravno i na vreme.

Poslednji scenario, na slici 6-14(d), isti je kao i scenario na slici 6-14(c), osim što sada - zbog gubljenja paketa - propadaju svi ponovni pokušaji slanja DR poruke. Posle *N* pokušaja, pošiljalac diže ruke i jednostrano raskida vezu, U međuvremenu, ističe tajmer kod primaoca, pa i on raskida vezu.

Iako protokol obično zadovoljava, on teorijski može da izneveri ako se izgubi početna DR poruka i ne uspe nijedan od *N* sledećih ponovnih pokušaja slanja. Pošiljalac će dići ruke i raskinuti vezu, a druga strana, ne znajući ništa o pokušaju raskidanja veze, radiće i dalje svom snagom. To rezultuje poluuspostavljenom (poluotvorenom) vezom.

Navedeni problem se može izbeći ako pošiljaocu ne dozvolimo da odustane posle *N* pokušaja, već ga nateramo da pokušava sve dok ne dobije odgovor. Međutim, ako je drugoj strani dozvoljeno da se tajmerom automatski isključuje, pošiljalac će do sudnjeg dana pokušavati da uspostavi vezu jer odgovor nikada neće stići. Ako drugoj strani ne dozvolimo da se automatski isključi, protokol će se zaglaviti u situaciji prikazanoj na slici 6-14(d).

Jedan način da se raskinu poluuspostavljene veze bilo bi pravilo automatskog raskidanja veze ukoliko tokom određenog vremena ne stigne nijedna TPDU poruka. Na taj način, ako jedna strana jednostrano raskine vezu, druga će primetiti nedostatak aktivnosti i takođe raskinuti svoju polovinu veze. Naravno, ako se uvede takvo pravilo, onda je neophodno da svaka transportna jedinica ima tajmer koji će se zaustaviti i ponovo pokrenuti kad god se pošalje nova TPDU poruka. Ako istekne rok ovog tajmera, šalje se prazna TPDU poruka da se draga strana ne bi isključila. S drage strane, ako se koristi pravilo automatskog isključivanja, pa se na inače slabo aktivnoj vezi uzastopno izgubi previše praznih TPDU poraka, prvo će se automatski isključiti jedna, pa draga strana.

Nećemo više razrađivati ovaj problem, ali bi moralo biti jasno jednom za svagda da raskidanje veze bez gubljenja podataka nije tako jednostavno kao što na prvi pogled izgleda.

6.2.4 Kontrola toka i privremeno skladištenje

Pošto smo malo detaljnije ispitali uspostavljanje i raskidanje veze, pogledajmo sada kako se upravlja uspostavljenom vezom. Već smo ranije opisali jedan od glavnih problema: kontrolu toka. On se u izvesnom smislu u transportnom sloju ispoljava kao i u sloju veze podataka, ali se pojavljuju i važne razlike. Osnovna sličnost je u tome da se u oba sloja na vezi moraju primeniti klizni prozori ili slična šema da neumereno brz pošiljalac ne bi zatrpao sporijeg primaoca porukama. Osnovna razlika je to što usmerivač obično ima srazmerno mali broj linija, dok računar može da ostvari brojne veze. Zbog te razlike nije praktično da se strategija privremenog skladištenja koja se koristi u sloju veze podataka, ugradi i u transportni sloj.

U protokolima sloja veze koje smo razmatrali u 3. poglavlju, okviri su privremeno skladištem u usmerivaču koji ih šalje i u usmerivaču koji ih prima. Na primer, u protokolu 6, i pošiljalac i primalac morali su da dodele po *MAX_SEQ* + 1 bafera svakoj liniji, pola za slanje, pola za primanje okvira. Ako pretpostavimo da računar ima maksimalno 64 veze i da koristi 4-bitne redne brojeve, on bi prema protokolu 6 morao da obezbedi 1024 bafera.

U sloju veze podataka, pošiljalac privremeno skladišti okvire za slučaj da mora ponovo da ih šalje. AJeo podmreža radi s datagramiina, i transportna jedinica pošiljaoca ih privremeno skladišti iz istog razloga. Ukoliko primalac zna da pošiljalac privremeno skladišti sve TPDU

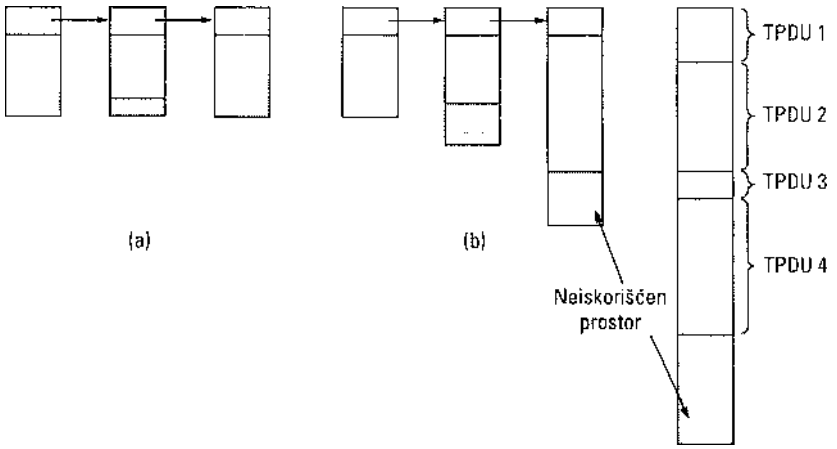
poruke dok se ne potvrde, on može, ali i ne mora da dodeli određene bafere određenim vezama. Primalac može, na primer, da bafere za sve veze čuva zajedno. Kada stigne TPDU poruka, čini se pokušaj da joj se bafer dodeli dinamički. Ako postoji slobodan bafer, TPDU poruka se prihvata; ako ne postoji, ona se odbacuje. Postoje pošiljalac spreman da ponovo šalje TPDU poruke koje se izgube u podmreži, to što ih primalac odbacuje ne pravi štetu, iako se time pomalo rasipaju resursi. Pošiljalac stalno iznova šalje poiuke sve dok ne dobije potvrdu.

Sve u svemu, ako je mrežna usluga nepouzdana, pošiljalac mora da privremeno uskladišti sve TPDU poruke koje šalje, kao u sloju veze podataka. Međutim, u pouzdanoj podmreži, moguć je drugačiji kompromis. Konkretno, ako pošiljalac zna da primalac uvek ima slobodan bafer, on ne mora da čuva kopije TPDU poruka koje šalje. Međutim, ako primalac ne može da garantuje da če svaka pristigla TPDU poruka biti i prihvaćena, pošiljalac će ipak morati da ih privremeno skladišti. U ovom drugom slučaju, pošiljalac ne može da se osloni na potvrdu mrežnog sloja jer ona samo znači daje paket stigao, a ne i daje prihvaćen. Na ovu važnu stvar vratićemo se kasnije.

Čak i kada se primalac složi da privremeno skladišti okvire, ostaje pitanje veličine bafera. Ako su TPDU poruke uglavnom iste veličine, prirodno je da organizuje skup bafera identične veličine, od kojih svaki može da primi jednu TPDU poruku, kao na slici 6-15(a). Međutim, ako veličina TPDU poruka znatno varira, od nekoliko znakova unetih s terminala do više hiljada znakova prilikom prenosa datoteka, skup bafera jednake veličine zapada u probleme. Ako se veličina bafera izabere prema veličini najveće TPDU poruke, prostor bafera neće biti efikasno iskorišćen kad god stigne neka kraća poruka. Ako se izaberu baferi veličine koja je manja od maksimalne veličine TPDU poruke, za skladištenje dugačkih poruka biće potrebno više bafera i složeniji rad s porukama.

Dragi pristup je da se koriste baferi različite veličine, kao na slici 6-15(b). Tu se memorija bolje iskorišćava, po cenu složenijeg rada s baferima. Treća mogućnost je da se svakoj vezi dodeli samo jedan veliki cirkularni bafer, kao na slici 6-15(c). I ovaj sistem dobro iskorišćava memoriju kada su sve veze potpuno opterećene, ali je islco- rišćenje malo ako je na nekim vezama saobraćaj redale.

Optimum koji se nalazi između privremenog skladištenja na izvorištu i privremenog skladištenja na odredištu zavisi od vrste saobraćaja na vezi. Za povremen saobraćaj niskih zahteva kakav, na primer, proizvodi interaktivan rad na terminalu, najbolje je da se ne predviđaju baferi za svaku vezu, već da se na oba kraja dodeljuju dinamički. Pošto pošiljalac ne može biti siguran da će primalac moći da obezbedi bafer, mora da zadrži kopiju TPDU poruke sve dok poruka ne bude potvrđena. S drage strane, za prenos datoteka i drugi saobraćaj koji zahteva veći propusni opseg, bolje je da primalac vezi dodeli čitav prozor bafera, da bi omogućio maksimalnu brzinu prenosa podataka. Prema tome, za povremen, niskozahtevan saobraćaj bolje je da se okviri privremeno skladište kod pošiljaoca, a za ravnomeran saobraćaj koji zauzima veliki propusni opseg - kod primaoca.



Slika 6-15. (a) Ulančani baferi jednake veličine, (b) Ulančani baferi različite veličine, (c) Jedan veliki cirkularan bafer po vezi.

Zbog stalnog uspostavljanja i raskidanja veza, i promena u tempu saobraćaja, pošiljalac i primalac moraju da dinamički dodeljuju bafere. Prema tome, transportni protokol treba da omogući pošiljaocu da na drugom kraju zahteva bafere. Baferi se mogu dodeljivati po pojedinačnoj vezi ili zbirno za sve veze između dva računara. Alternativno, primalac, koji zna s koliko bafera raspolaže (ali ne i ponuđen saobraćaj), može da poruči pošiljaocu „Rezervisao sam za tebe *X* bafera". Kada se broj uspostavljenih veza poveća, možda se mora smanjiti broj bafera po vezi, pa protokol treba da predvidi i takvu mogućnost.

Prilično opšti način za dinamičko dodeljivanje bafera, za razliku od protokola kliznih prozora iz 3. poglavlja, oslanja se na razdvajanje procesa skladištenja od rada s potvrdama. Dinamičko dodeljivanje bafera u stvari znači rad s prozorom promenljive veličine. Na početku, pošiljalac zahteva određen broj bafera na osnovu svojih trenutnih potreba. Primalac tada dodeljuje onoliko od traženih bafera koliko može. Svaki put kada pošiljalac pošalje TPDU poruku, on mora da smanji broj dodeljenih bafera za jedan i da potpuno prestane sa slanjem kada taj broj dostigne nulu. Primalac tada povratnim saobraćajem zasebno šlepuje potvrde i dodelu bafera.

Na slici 6-16 prikazanje primer mogućeg dinamičkog dodeljivanja bafera u datagramskoj podmreži u kojoj se koriste 4-bitni redni brojevi. Smatrajte da se podaci o dodeli bafera šalju posebnim TPDU porukama, kao što je prikazano, tj. ne šlepuju se uz povratne okvire. Na početku, A zahteva osam bafera, ali dobija samo četiri. On zatim šalje tri TPDU poruke, od kojih se treća gubi. TPDU porukom 6 potvrđuje se prijem svih TPDU poraka zaključno s rednim brojem 1 i tako omogućuje računaru *A* da oslobodi odgovarajuće bafere, i istovremeno se A obaveštava da sme da pošalje još tri sledeće TPDU poruke (tj. one s rednim brojevima 2, 3 i 4). A zna daje već poslao poruku 2, pa misli da bi mogao sada da pošalje poruke 3 i 4, i to i čini. Posle toga se blokira i čeka dodelu novih bafera. Tokom čekanja na liniji 9 može doći do ponovnog slanja okvira izazvanog istekom roka tajmera jer se za takvo slanje koriste već do- deljeni baferi. Na liniji 10, *B* potvrđuje prijem svih TPDU poruka zaključno s rednim brojem 4, ali odbija da dalje prima poruke od računara *A*. Takva situacija

nije moguća u protokolu kliznih prozora iz 3. poglavlja. Sledećom TPDU porukom, upućenom od A ka *B,* ipak se dodeljuje bafer i računani A dozvoljava da nastavi sa slanjem pomka.

	A	Poruka	B	Napomena
1	——-	< zahtevam 8 bafera>	——~	A zahteva 8 bafera
2		<potvrda = 15, baferi = 4>		B dozvoljava samo poruke 0-3
3	—	<red. br. = 0, podaci = m0>		Računaru A su ostala 3 bafera
4		<red. br. = 1, podaci = m1>	—	Računaru A su ostala 2 bafera
5		<red, br. = 2, podaci = m2>	…	Poruka se izgubila, ali A misli da je preostala poruka 1
6	—	<potvrda = 1, baferi = 3>	—	B potvrđuje poruke 0 i 1, dozvoljava poruke 2-4
7		<red. br. = 3, podaci = m3>	—	Računaru A je ostao 1 bafer
8	—	<red. br. = 4, podaci = m4>	—	A više nema bafera i mora da stane
9		<red, br. = 2, podaci = m2>	—	Računaru A ističe tajmer i ponovo emituje
10		<potvrda = 4, baferi = 0>	*	Sve je potvrđeno, ali A je i dalje blokiran
11	—	<potvrda = 4, baferi = 1>	—	A sada može da pošalje 5 poruka
12	—	<potvrda = 4, baferi = 2>	—	B je negde pronašao slobodan bafer
13		<red. br. = 5, podaci = m5>		Računaru A je ostao 1 bafer
14	——	<red. br, = 6, podaci = m6>	—	A je ponovo blokiran
15	—	<potvrda = 6, baferi = 0>		A je još uvek blokiran
16	…	<potvrda = 6, baferi = 4>	■*—	Potencijalna kružna blokada

Slika 6-16. Dinamičko dođeljivanje bafera. Strelice prikazuju smer prenosa. Tri tačke (...) označavaju izgubljenu TPDU poruku.

Ovakav način dodeljivanja bafera može da stvori probleme u datagramskoj mreži ako se izgubi upravljačka TPDU poruka. Pogledajte red 16. Računar *B* je računaru A sada dodelio još bafera, ali se TPDU poruka s podacima o dodeljenim baferima izgubila. Pošto se upravljačke TPDU poruke ne numerišu, niti se za njih koristi tajmer, računar A je trajno (kružno) blokiran. Da bi se takva situacija prevazišla, svaki računar treba da periodično šalje upravljačke TPDU poruke s potvrdom i stanjem bafera za svaku vezu. Na taj način, računar A će se, ranije ili kasnije, deblokirati.

Dosad smo pretpostavljali da brzinu slanja podataka ograničava samo broj slobodnih bafera primaoca. Kako cene drastično padaju, možda će biti moguće da se računari opreme s dovoljno memorije da nedostatak bafera gotovo nikada ne bude problem.

Kada prostor u baferima više ne ograničava protok podataka, pojavljuje se draga prepreka: prenosni kapacitet podmreže. Ako kontrolni usmerivači mogu da razmenju- ju najviše *x* paketa u sekundi, a između para računara ima *k* potpuno odvojenih putanja, računari mogu razmenjivati najviše *kx* TPDU poruka u sekundi, bez obzira na količinu bafera na dva kraja. Ako pošiljalac šalje više od *kx* TPDU poruka u sekundi, mreža đe se zagušiti jer ne može da isporučuje pakete brzinom kojom pristižu.

Tu je potreban mehanizam koji se ne zasniva na kapacitetu bafera primaoca, već na prenosnom kapacitetu mreže. Jasno je da se kontrola toka mora primeniti na pošiljaoca da na mreži ne bi istovremeno bilo previše nepotvrđenih TPDU poruka. Belsnes (1975) predložio je kontrolu toka s kliznim prozorima gde pošiljalac dinamički podešava veličinu prozora prema prenosnom kapacitetu mreže. Ako mreža može da obradi *c* TPDU poruka u sekundi, a *r* je vreme potrebno za slanje poruke, njen prenos, stajanje u redovima čekanja, obradu kod

primaoca i povratak potvrde, prozor pošiljaoca treba da bude veličine *cr*. S prozorom te veličine pošiljalac nema nikakvu rezervu. Neznatno slabljenje performansi mreže izazvaće njegovu blokadu.

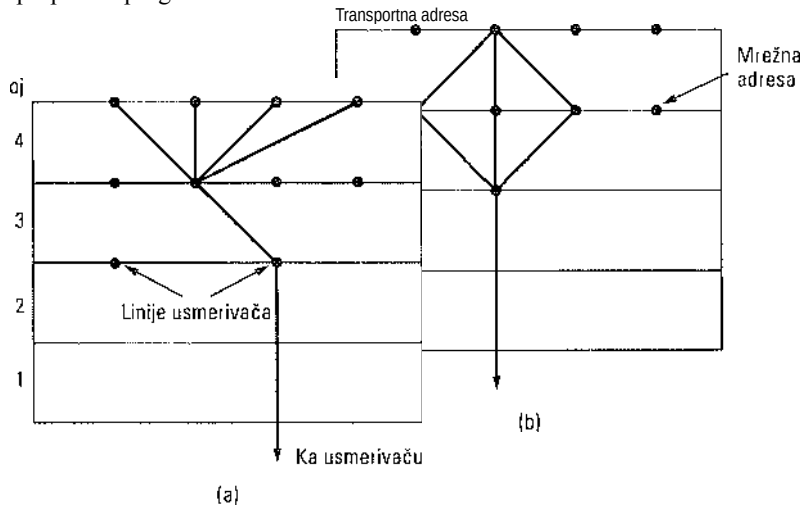
Da bi periodično podešavao veličinu prozora, pošiljalac mora istovremeno da nadzire parametre i da izračunava veličinu prozora. Prenosni kapacitet se može jednostavno odrediti brojanjem potvrđenih TPDU poruka u određenom vremenskom intervalu i deljenjem rezultata s tim intervalom. Tokom merenja, pošiljalac treba da šalje punom snagom da bi bio siguran da je registrovani broj potvrda rezultat maksimalnog preno- snog kapaciteta mreže, a ne niske brzine slanja. Vreme potrebno da se poslata TPDU poruka potvrdi može se meriti tačno i stalno treba određivati njegov tekući prošek. Pošto prenosni kapacitet mreže za svaki tok varira s vremenom, veličinu prozora treba često podešavati. Kao što ćemo kasnije videti, na Internetu se koristi slična šema.

6.2.5 Multipleksiranje

Multipleksiranje više „razgovora“ u vezama, virtuelnim kolima i fizičkim linijama igra određenu ulogu u nekoliko slojeva arhitekture mreže. Potreba za multipleksi- ranjem u transportnom sloju može da se javi iz više razloga. Na primer, ako računar ima samo jednu mrežnu adresu, sve njegove transportne veze moraju ići preko nje. Kada stigne TPDU poruka, mora postojati mehanizam pomoću koga će se ona dode- liti pravom procesu. Ova situacija, zvana **uzlazno multipleksiranje** (engl. *upward multiplexing),* prikazana je na slici 6-17(a). Na toj slici, četiri različite transportne veze ka udaljenom računani koriste isti mrežni priključak (tj. IP adresu).

Multipleksiranje u transportnom sloju može se iskoristiti i na drugi način. Pretpostavite, na primer, da podmreža interno radi s virtuelnim kolima i da u svakom od njih ograničava maksimalnu brzinu prenosa podataka. Ako je korisniku potreban veći propusni opseg nego što može da ponudi jedno virtuelno kolo, on može da otvori više mrežnih priključaka i da saobraćaj ciklično usmerava na njih, kao na slici 6-17(b). Takav način rada zove se **silazno multipleksiranje** (engl. *downward multiplexing).* Ako je istovremeno otvoreno *k* mrežnih priključaka, propusni opseg se efektivno povećava *k* puta. Čest primer silaznog multipleksiranja javlja se kod kućnih korisnika koji

imaju ISDN liniju. Takva linija obezbeđuje dve zasebne veze, brzine po 64 kb/s. Kada korisnik pomoću obe pozove davaoca Internet usluga i na njih ravnomerno podeli saobraćaj, može da poveća propusni opseg na efektivnih 128 kb/s.



Slika 6-17. (a) Uzlazno multipleksiranje. (b) Silazno multipleksiranje.

6.2.6 Oporavljanje posle pada sistema

Ako računari i usmerivači mogu da otkazu, onda problem može da bude i oporavljanje posle pada sistema. Ukoliko se transportne jedinice nalaze isključivo u računalima, oporavak posle pada mreže i usmerivača ima jasan tok. Ako mrežni sloj obezbeđuje uslugu datagrama, transportne jedinice sve vreme očekuju izgubljene TPDU poruke i znaju kako da rade s njima. Kada mrežni sloj obezbeđuje uslugu sa uspostavljanjem direktne veze, onda se gubitak virtuelnog kola nadoknađuje uspostavljanjem novog kola i anketiranjem udaljene transportne jedinice o TPDU paketima koje je primila i koje nije primila. Ovi drugi se tada mogu ponovo poslati.

Veći problem je oporavljanje posle pada računara. Konkretno, bilo bi možda poželjno da klijenti mogu da nastave s radom kada serveri „padnu“ i zatim se brzo oporave. Da bismo ilustrovali ovu teškoću, pretpostavimo da jedan računar, klijent, šalje dugačku datoteku drugom računani, serveru datoteka, koristeći jednostavan protokol „stani i čekaj“. Transportni sloj servera jednostavno prosleđuje dolazne TPDU poruke korisniku usluge prenosa, jednu po jednu. Usred prenosa server pada. Kada se povrati, njegove tabele se ponovo inicijalizuju, tako da on ne zna tačno gde se nalazi.

U nameri da se vrati u stanje u kom je bio neposredno pre otkazivanja, server može da difuzno emituje TPDU poruke svim računarima, objavljujući da se upravo oporavio od pada i zahtevajući od njih da ga izveste o stanju svih otvorenih veza. Svaki klijent može da bude u jednom od dva stanja: stanju *SI,* kada ima jednu poslatu TPDU poruku (za koju čeka potvrdu) i stanju *S2,* kada nema takvih poruka. Samo na osnovu tih informacija o stanju,

klijent mora da odluči da li da ponovo pošalje poslednju TPDU poruku.

Na prvi pogled, sve izgleda jasno: kada čuje za pad servera, klijent treba da ponovo pošalje samo poslatu nepotvrđenu TPDU poruku (tj, ako se nalazi u stanju *SJ).* Međutim, bližim analiziranjem otkrivamo teškoće u ovom naivnom pristupu. Razmotrite, na primer, situaciju u kojoj transportna jedinica servera najpre šalje potvrdu, a zatim, pošto je potvrda već poslata, upisuje podatke u proces aplikacije. Upisivanje TPDU poruke u izlazni tok i slanje potvrde dva su različita događaja koji se ne mogu istovremeno izvesti. Ako server otkaže nakon što je poslao potvrdu, ali pre nego što je upisao podatke, klijent će primiti potvrdu i biti u stanju *SO* kada do njega stigne obaveštenje o padu servera. Klijent, prema tome, neće ponovo slati TPDU poruku, smatrajući (pogrešno) daje ona stigla serveru. Takva odluka klijenta izaziva gubljenje TPDU poiuka.

U ovom trenutku možda mislite: „Taj problem se može lako rešiti. Samo treba re-programirati transportnu jedinicu da prvo upisuje podatke, a zatim da šalje potvrdu.“ Savetujemo vam da potražite neko drugo rešenje. Zamislite daje transportna jedinica izvršila upisivanje, pa je došlo do pada sistema, tako da potvrda nije poslata. Klijent će tada biti u stanju *SI* i ponovo će poslati TPDU poruku, što će izazvati nezapaženi TPDU duplikat u izlaznom toku ka procesu aplikacije na servera.

Bez obzira na to kako su programirani klijent i server, uvek postoje situacije od kojih protokol ne može potpuno da se povrati u pređašnje stanje. Server se može programirati na jedan od dva načina: da prvo šalje potvrdu i da prvo upisuje podatke. Klijent se može programirati na jedan od četiri načina: da uvek ponovo šalje posled- nju TPDU poruku, da je nikada ne šalje, da je šalje samo kada je u stanju *SO* i da je šalje samo kada je u stanju *SI.* To daje osam kombinacija, ali videćemo da za svaku od njih postoji skup događaja zbog kojih protokol neće moći da se povrati.

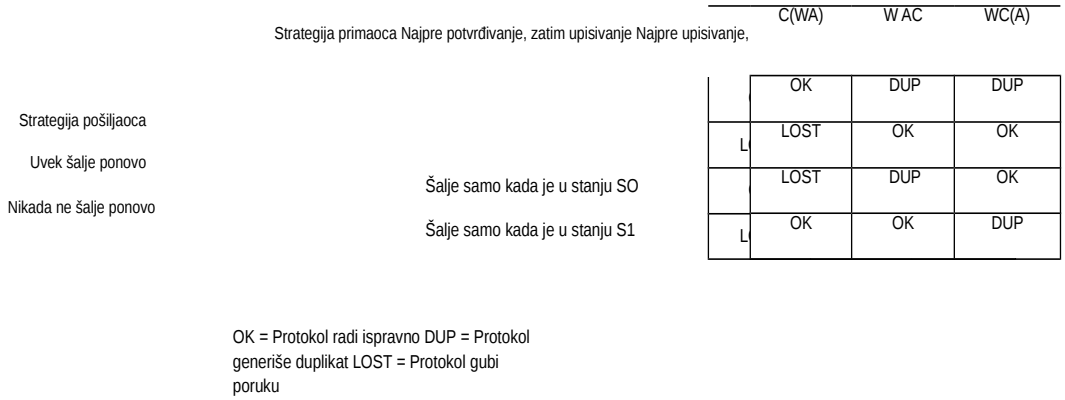
Na servera su moguća tri događaja: slanje potvrde *(A),* upisivanje u izlazni tok procesa (*W)* i pad sistema (C). Ta tri događaja mogu se dešavati prema šest različitih redosleda: *AC(W), AWC, C(AW), C(WA), WAC* i *WC(A),* gde se zagradama označava da ni A, ni *W* ne mogu da slede iza C (tj. kada sistem padne, onda je pao). Slika 6-18 prikazuje svih osam kombinacija mogućih strategija klijenta i servera, zajedno sa ispravnim tokom događaja. Obratite pažnju na to da za svaku strategiju postoji neki sled događaja iza koga protokol ne uspeva da se oporavi. Na primer, ako klijent uvek ponovo šalje TPDU poruku, kombinacija *A WC* generisaće neotkriveni duplikat, iako će za dve druge kombinacije događaja protokol raditi dobro.

Neće pomoći ni ako protokol još doteramo. Čak i kada klijent i server razmene više TPDU poruka pre nego što server išta upiše, tako da tačno zna šta će se dogoditi, klijent ne može da sazna da li je server otkazao pre ili posle upisivanaja. Neizbežno moramo zaključiti da se u uslovima neistovremenog odvijanja događaja, otkazivanje računara i njegov oporavak ne mogu jasno prikazati višim slojevima.

Ako ovaj zaključak uopštimo, sledi da oporavljanje sloja *N* može da izvede samo sloj *N + 1,* i to samo ukoliko viši sloj sačuva dovoljno informacija o poslednjem statusu. Kao što smo već pomenuli, transportni sloj se može oporaviti posle otkazivanja mrežnog sloja pod uslovom da svaki kraj veze stalno evidentira trenutno stanje.

Opisani problem nam u novom svetlu prikazuje značenje izraza „potvrđivanje od jednog do drugog kraja“. Transportni protokol je u načelu protokol koji radi s dva kraja veze - on nije ulančan kao protokoli nižih slojeva. Razmotrite sada slučaj korisnika koji

zahteva transakcije sa udaljenom bazom podataka. Pretpostavimo daje udaljena transportna jedinica tako programirana da prvo prosleđuje TPDU poruke sledecem sloju, a tek onda da za njih šalje potvrde. Čak i u tom slučaju, kada korisnik dobije potvrdu o prijemu, on nije siguran da lije server po prijemu poruke dovoljno dugo ostao aktivan da bi ažurirao bazu podataka. Stvarno potvrđivanje s kraja na kraj, pri čemu prijem potvrde znači daje posao stvarno urađen, a njeno nestizanje da posao nije urađen, vero- vatno se ne može postidi. Tu temu su detaljno obradili Saltzer i saradnici (1984).



Slika 6-18. Različite kombinacije strategija klijenta i servera.

6.3 JEDNOSTAVAN TRANSPORTNI PROTOKOL

Da bismo ono o čemu smo govorili približili praksi, u ovom odeljku ćemo detaljno proučiti jedan primer transportnog sloja. Pri tome ćemo koristiti apstraktne osnovne operacije sa slike 6-2 za slučaj rada sa uspostavljanjem direktne veze. Takvim izborom primer se približava popularnom protokolu TCP, ali je jednostavniji od njega.

6.3.1 Osnovne operacije korišćene u primeru

Najpre imamo problem da konkretno izrazimo osnovne operacije transportnih usluga. Sa operacijom CONNECT to ide lako: upotrebićemo samo proceduru *connect* iz biblioteke; veza se uspostavlja kada tu proceduru pozovemo uz odgovarajuće parametre. Parametri su lokalna i udaljena pristupna TSAP tačlca. Tokom izvršavanja usluge, tj. pokušaja da se uspostavi veza, pozivalac je (privremeno) blokiran. Ako se veza uspostavi, pozivalac se deblokira i može da počne sa slanjem podataka.

Kada proces poželi da prima dolazne pozive za uspostavljanje veze, on poziva proceduru *listen* zadajući istovremeno i pristupnu TSAP tačku na kojoj će osluškivati. Proces se tada blokira sve dok neki udaljeni proces ne pokuša da uspostavi vezu s tom TSAP tačkom.

Obratite pažnju na to da je ovaj model veoma asimetričan. Strana koja izvršava proceduru *listen* pasivno čeka da se nešto dogodi. Aktivna je samo druga strana koja inicira uspostavljanje veze. Zanimljivo je šta se događa ako protokol započne najpre aktivna strana. Prema jednoj strategiji, veza se ne može uspostaviti ako na udaljenoj TSAP tački nema slušaoca. Prema drugoj, inicijator se u tom stanju blokira (možda večno), sve dok se ne pojavi slušalac.

U našem primeru opredelili smo se za kompromisno rešenje: da se primalac zatrpava zahtevom za uspostavljanje veze tokom određenog vremenskog intervala. Ako proces na tom računani pozove proceduru *listen* pre nego što istekne pomenuti vremenski interval, veza se uspostavlja; u suprotnom, zahtev se odbacuje, pozivalac se deblokira i dobija poruku o grešci.

Za raskidanje veze koristićemo proceduru *disconnect.* Veza se raskida kada je izvrše obe strane. Drugim recima, koristićemo model simetričnog raskidanja veze.

Pri prenosu podataka odvija se isto što i pri uspostavljanju veze: pošiljalac je aktivan, a primalac pasivan. Zato ćemo taj problem resiti na isti način: aktivni proces poziva proceduru *send* da bi poslao podatke, a pasivni poziva procedura *receive* koja ga blokira dok ne stigne TPDU poruka.

Konkretna definicija naših usluga obuhvata dakle pet osnovnih operacija: CONNECT, LISTEN, DISCONNECT, SEND i RECEIVE. Za svaku operaciju u biblioteci postoji odgovarajuća procedura koja je izvršava. Slede parametri osnovnih operacija i odgovarajuće procedure:

```
connum = LISTEN(local) connum =
CONNECT(local, remote) status =
SEND(connum, buffer, bytes) status =
RECEIVE(connum, buffer, bytes) status =
DISCONNECT(connum)
```

Operacija LISTEN najavljuje da je pozivalac voljan da prihvati zahteve za uspostavljanje veze s naznačenom TSAP tačkom. Korisnik usluge se blokira sve dok neko s njim ne pokuša da uspostavi takvu vezu. Vreme čekanja nije ograničeno.

Za operaciju CONNECT potrebna su dva parametra: *local* - lokalna TSAP tačka (tj. transportna adresa) i *remote* - udaljena TSAP tačka. Kada se pozove uz njih, ona pokušava da uspostavi vezu između te dve tačke. Ako to uspe, vraća vrednost promenljive *connum* kao nenegativan broj koji se u narednim pozivima koristi za identifikovanje veze. Ako ne uspe da uspostavi vezu, vraća negativnu vrednost promenljive *connum* iz koje se može saznati razlog neuspeha. U našem jednostavnom modelu, svaka TSAP tačka može da učestvuje samo u jednoj vezi, pa uzrok nemogućnosti uspostavljanja veze može biti to što se jedna od transportnih adresa trenutno koristi. Ostali razlozi mogu biti: kvar udaljenog računara i nelegalna lokalna, odnosno udaljena adresa.

Usluga SEND šalje sadržaj bafera kao poruku preko naznačene transportne veze, ako je potrebno, izdeljenu u više elementarnih jedinica. Moguće greške, vraćene kao vrednost promenljive *status,* mogu biti: veza ne postoji, neispravna adresa bafera ili negativna vrednost (prekoračenje) brojača.

Operacijom RECEIVE pozivalac izražava želju da prima podatke. Željena veličina poruke naznačuje se parametrom *bytes* (bajtovi). Ako je udaljeni proces raskinuo vezu ili je adresa bafera neispravna (npr. izvan skupa adresa koje upotrevljava korisnikov program),

promenljiva *status* sadržače kod geške.

Operacija DISCONNECT raskida transportnu vezu označenu parametrom *connum.* Moguće greške su: *connum* pripada drugom procesu i *connum* nije ispravan identifi- kator veze. Promenljiva *status* sadržače vrednost 0 (u slučaju uspešne operacije) ili kod greške.

6.3.2 **Transportna** jedinica iz primera

Pre nego što razmotrimo kod transportne jedinice, imajte na umu daje ovaj primer analogan ranijim primerima iz 3. poglavlja: on je previše jednostavan za stvarni rad i ovde ga iznosimo najviše iz didaktičkih razloga. Iz njega su ispušteni mnogi tehnički detalji (kao što je opsežna provera grešaka), bez kojih ne može da se zamisli komercijalan sistem.

Transportni sloj koristi osnovne operacije usluga mrežnog sloja da bi slao i primao TPDU poruke. Za ovaj primer treba da ih izaberemo. Jedan izbor bio bi nepouzdana usluga datagrama. Pošto smo želeli da primer ostane jednostavan, nismo nju odabrali. S nepouzdanom uslugom datagrama, transportni kod bi postao dugačak i složen i uglavnom bi se bavio izgubljenim i zakasnelim paketima. Osim toga, najveći deo te problematike proradili smo u 3. poglavlju.

Zbog toga smo odlučili da upotrebimo pouzdanu mrežnu uslugu sa uspostavljanjem direktne veze. Tako se možemo koncentrisati na problematiku transporta koje nema u nižim slojevima. Ta problematika obuhvata - između ostalog - uspostavljanje veze, raskidanje veze i rad s kreditima. Tako bi mogla da izgleda jednostavna usluga prenosa ugrađena u gornji sloj ATM mreže.

Transportna jedinica bi u načelu mogla da bude deo operativnog sistema računara ili paket u biblioteci potprograma koji se izvršavaju unutar korisnikovog adresnog prostora. Zbog jednostavnosti primera, odlučili smo da naša transportna jedinica bude paket u biblioteci potprograma, ali se uz male izmene (u načinu pristupanja korisnikovim baferima) ona može smestiti i u operativni sistem.

Treba, međutim, napomenuti da „transportna jedinica“ iz ovog primera nije stvarno posebna jedinica, već deo korisničkog procesa. To, konkretno, znači da se tokom izvršavanja operacije koja prouzrokuje blokiranje, kao što je LISTEN, blokira i čitava transportna jedinica. Premda tako nešto radi dobro na računaru sa samo jednim korisnikom, na višekorisničkom računaru bi bilo prirodnije da transportna jedinica bude zaseban proces, nezavistan od korisničkih procesa.

Interfejs ka mrežnom sloju ostvaruje se procedurama *to_net* (ka mreži) i *fromjiet* (od mreže), koje nisu prikazane. Svaka od njih ima šest parametara. Prvi je identifi- kator veze koji se preslikava na odgovarajuće virtuelno kolo. Zatim dolaze bitovi *Q* i *M* koji, kada su 1, označavaju upravljačku poruku, odnosno poruku da u narednom paketu sledi još podataka iste poruke. Posle toga, dolazi tip paketa izabran između šest tipova prikazanih na slici 6-19. Na kraju imamo pokazivač na same podatke i ceo broj koji označava broj bajtova podataka.

Mrežni paket	Značenje
CALL REQUEST	Šalje se radi uspostavljanja veze
CALLACCEPTED	Odgovor na CALL REQUEST
CLEAR REQUEST	Šalje se radi raskidanja veze
CLEAR CONFIRMATION	Odgovor na CLEAR REQUEST
DATA	Koristi se za transport podataka
CREDIT	Upravljački paket za podešavanje prozora

Slika 6-19. Paketi mrežnog sloja koji se koriste u primeru.

Pri pozivanju procedure *tojiet,* transportna jedinica umeće sve parametre koje treba da očita mrežni sloj, dok pri pozivanju procedure *fromjiet,* mrežni sloj raščlanjava dolazni paket da bi ga prosledio transportnoj jedinici. Umesto da se bavi prosleđivanjem dolaznih i odlaznih paketa, transportni sloj mrežnom sloju prosleđuje samo parametre procedure i tako ne mora da zna detalje protokola mrežnog sloja. Ako bi transportna jedinica pokušala da pošalje paket u trenutku kad je klizni prozor virtuelnog kola u mrežnom sloju pun, bila bi blokirana procedurom *to_net* sve do trenutka dok se ne oslobodi prostor u prozoru. Taj mehanizam transportni sloj uopšte ne vidi jer ga sprovodi mrežni sloj služeći se komandama, analognim komandama *enable jra.nsportJ.ayer* i *disable^transportJayer* iz protokola u 3. poglavlju. Mrežni sloj podešava i veličinu prozora s paketima.

Osim ovog nevidljivog mehanizma za privremeno blokiranje, transportna jedinica poziva i izričite procedure *sleep* i *wakeup* koje takođe nisu prikazane. Procedura *sleep* se poziva kada transportna jedinica treba da se logički blokira i čeka na spoljni događaj, najčešće na stizanje paketa. Pošto se pozove procedura *sleep,* transportna jedinica prestaje da se izvršava (kao i korisnički procesi, naravno).

Kod transportne jedinice prikazanje na slici 6-20. Svaka veza je uvek u jednom od sedam stanja:

1. IDLE - Veza još nije uspostavljena.
2. WAITING - Izvršena je procedura CONNECT i poslat je paket CALL REQUEST.
3. QUEUED - Stigao je paket CALL REQUEST; još nije izvršena procedura LISTEN.
4. ESTABLISHED - Veza je uspostavljena.
5. SENDING - Korisnik čeka dozvolu da pošalje paket.
6. RECEIVING - Izvršena je procedura RECEIVE.
7. DISCONNECTING - Lokalno je izvršena procedura DISCONNECT.

Prelazak iz stanja u stanje izaziva jedan od sledećih događaja: izvršavanje osnovne operacije, stizanje paketa i isticanje roka tajmera.

Procedure prikazane na slici 6-20 dele se u dve vrste. Većina se može direktno pozvati iz korisničkih programa. Međutim, drugačije su procedure *packetjirrival* i *clock.* Njih spontano pokreću spoljni događaji: stizanje paketa, odnosno otkucavanje sata. To su u stvari potprogrami za obradu prekida. Pretpostavićemo da se nikada ne aktiviraju tokom izvršavanja procedura transportne jedinice. One se mogu pozvati samo kada je korisnički proces uspavan ili se izvršava izvan transportne jedinice. To svojstvo je neophodno za ispravan rad koda.

```
#define MAX_CONN 32                    /*  maksimalan broj istovremenih veza 7
#define MAX_MSG_SIZE 8192              /*  najveća poruka u bajtovima 7
```

```
#define MAX_PKT_SIZE 512                    /*    najveći paket u bajtovima 7
#define TIMEOUT 20
#define CRED 1
#define OK 0

#define ERR_FULL -1
#define ERR_REJECT -2
#define ERR_CLOSED -3
#define LOW_ERR -3

typedef int transport_address;
typedef enum {CALL_REQ,CALL_ACC,CLEAR_REQ,CLEAR_CONF,DATA_PKT,CREDIT}
pktjype;
typedef   enum   {IDLE,WAITING,QUEUED,ESTABLISHED,SENDING,RECEIVING,DISCONN}
estate;

/* Globalne promenljive. 7
transport_address listen_address;                                          /*
                                          lokalna adresa na kojoj se osluškuje 7
int listen_conn;                                                  /* identifikator  veze  za
listen 7
unsigned ehar data[MAX_PKT_SIZEj;          /* prostor za oblikovanje paketa s podacima7

struct conn {
  transport_address local_address, remote_address; estate state;
                                         /*  stanje ove veze 7
  unsigned char *user_buf_addr;            /*          pokazivač na prijemni bafer 7
  int byte_count;                                                  /*brojač
  poslatih/primljenih bajtova                                      7
  int clr_req_received;                                                      /*
                                          postavlja se pri prijemu paketa CLEAR_REQ 7
  int timer;                             /*tajmer za pakete CALL__REQ 7
  int credits;                                                     /*broj poruka koje
  se mogu poslati                                                  7
} conn[MAX_CONN + 1];                                          /*    element   0   se   ne
koristi 7

void sleep(void);                                  /*prototipovi          7
void wakeup(void);
void to_net(int cid, int q, int m, pktjype pt, unsigned char *p> int bytes); void
from_net(int *cid, int *q, int *m, pktjype *pt, unsigned char *p, int *bytes);
int listen(transport3address t)
{/* Korisnik želi da osluškuje zahteve za uspostavljanje veze. Proverava da li je CALL_REQ
već stigao. 7 int i, found = 0;

tor (i = 1; i <= MAX_CONN; i++)              /*  traži u tabeli CALL_REQ 7
    if (connpj.state == QUEUED && conn[i].local_address == t)
        { found = i; break;
    }
if (found == 0) {
    /* Ne čeka nijedan paket CALL_REQ. Idi na spavanje dok ne stigne ili dok ne istekne rok
tajmera. 7
```

```
    listen_address = t; sleepf); i = listen_conn ;
}
connjij.state = ESTABLISHED;                    /* veza je uspostavljena 7
connji],timer = 0;                              /*  tajmer se ne koristi 7

listen_conn = 0;                                /*  smatra se da je 0 pogrešna adresa 7
to_net(i, 0, 0, CALL_ACC, data, 0);             /*    nalaže se mreži da prihvati vezu 7
return(i);                                      /*    vraća se identifikator veze 7


int connect(transport_address l, transport_address r)
(/* Korisnik želi da se poveže sa udaljenim procesom; šalje paket CALL_REQ. 7 int i;
  struct conn *cptr;

data[0] = r; data[1]    =                       l;/*  ovo treba paketu CALL_REQ 7
i = MAX_CONN;                                   /*    pretraživanje tabele obrnutim redom 7
while (connpj.state != IDLE && i > 1) i = i 1; if (connpj.state == IDLE) {
    /* Upiši u tabelu odrednicu da je poslat paket CALL_REQ. 7 cptr = &conn[ij;
    cptr->locaLaddress = l; cptr->remote_address = r; cptr->state = WAITING; cptr-
    >clr_req_received = 0; cptr->credits = 0; cptr->timer = 0; to_net(i, 0, 0, CALL_REQ, data,
    2);
    sleep();                                    /*    čekaj CALL_ACC ili CLEAR_REQ7
    if (cptr->state  ==   ESTABLISHED) return(i);
    if (cptr->clr_req_received) {
        /* Druga strana je odbila poziv. 7
        cptr->state = IDLE;                     /*   vrati se u stanje IDLE 7
        to_net(i, 0, 0, CLEAR_CONF, data, 0);
        return(ERR_REJECT);
    }
  } else retum(ERR_FULL);                                   /*odbija     vezu:     nema
  mesta                                                     u tabeli 7
} int send(int cid, unsigned char bufptr[], int bytes)
{/* Korisnik želi da pošalje poruku. */ int i,
  count, m;
  struct conn *cptr = &conn[cid];

/* Ulazak u stanje SENDING. 7 cptr->state =
SENDING;
cptr->byte_count = 0;                           /* do ove poruke poslato je # bajtova */
if (cptr->clr_req_received == 0 && cptr->credits == 0) sleep();
if (cptr->clr_req_received == 0) {
    /* Kredit na raspolaganju; ako treba, podeli poruku u više paketa. */
    do {
        if (bytes cptr->byte_count > MAX_PKT_SIZE) {/* poruka u više paketa */
            count = MAX_PKT_SIZE; m = 1; /* sledi još paketa */
        } else {                                /* poruka u jednom paketu */
            count = bytes cptr->byte_count; m = 0; /* poslednji paket ove poruke 7
        }
        for (i = 0; i < count; i++) dataji] = bufptr[cptr->byte_count + i]; to_net(cid, 0, m,
    DATA_PKT, data, count);/* šalji 1 paket */ cptr->byte_count = cptr->byte:count + count;/*
    povećaj broj do sada poslatih bajtova 7 } while (cptr->byte_count < bytes);          /* radi u
    petlji dok ne pošalješ celu poruku */
```

```
        cptr->credits;                          /* svaka poruka koristi jedan kredit */
        cptr->state = ESTABLISHED;
        return(OK);
    } else {
        cptr->state = ESTABLISHED;
        return(ERR_CLOSED); /* slanje nije uspelo: druga strana želi da se isključi */
    }
}
int receive(int cid, unsigned char bufptrj], int *bytes)
{/* Korisnik je pripremljen da primi poruku. */
    struct conn *cptr = &conn[cidj; if (cptr-
    >clr_req_received == 0) {
        /* Veza i dalje postoji; pokušaj da primiš. */
        cptr->state = RECEIVING; cptr-
        >user_buf_addr = bufptr; cptr->byte_count
        = 0; data[0] = CRED; data[1] = 1;
        to_net(cid, 1, 0, CREDIT, data, 2);     /*   pošalji kredit */
        sleep();                                /*   blokiraj se iščekujući podatke */
        *bytes = cptr->byte_count;
    }
    cptr->state = ESTABLISHED; return(cptr-
    >clr_req_received ? ERR_CLOSED ; OK);
```

```
int disconnect(int cid)
{/* Korisnik želi da raskine vezu. 7
 struct conn *cptr = &conn[cid];

 if (cptr->clr_req__received)                    /* druga strana je inicirala raskidanje 7
     { cptr->state = IDLE;                       /* veza je sada raskinuta 7
     to_net(cid, 0, 0, CLEAR_CONF, data, 0);
                                                                  /* mi smo inicirali raskidanje 7
 } else}


     cptr->state = DISCONN; /* veza se ne raskida dok se druga strana ne složi 7
     to_net(cid, 0, 0, CLEAR_REQ, data, 0);
 }
 return(OK);
}
void packet_arrival(void)
{/* Paket je stigao, uzmi ga i obradi. 7
 int cid;                                        /*    veza preko koje je stigao paket 7
 int count, i, q, m;
 pkt_type ptype; /* CALL_REQ,CALL_ACC>CLEAR_REQ,CLEAR_CONF,DATA_PKT,
CREDIT 7
 unsigned char data[MAX_PKT_SIZE];       /*    deo s podacima iz dolaznog paketa 7
 struct conn *cptr;

 from_net(&cid, &q, &m, &ptype, data, &count); /* uzmi ih 7
 cptr = &conn[cidj;

 switch (ptype) {
   case CALL_REQ:                                /* udaljeni korisnik želi da uspostavi vezu7
     cptr->local_address = datajOj; cptr->remote_address = dataU[1 ]; if (cptr-
     >local_address == listen_address) {
           listen_conn = cid; cptr->state = ESTABLISHED; wakeup();
     } else {
           cptr->state = QUEUED; cptr->timer = TIMEOUT;
     }
     cptr->clr_req_received = 0; cptr->credits = 0; break;

   case CALL_ACC: /* udaljeni korisnik je prihvatio naš paket CALL_REQ 7
     cptr->state = ESTABLISHED; wakeup(); break;

   case CLEAR__REQ; /* udaljeni korisnik želi da raskine vezu ili da odbije poziv 7
     cptr->clr_req_jeceived = 1;
     if (cptr->state == DISCONN) cptr->state = IDLE; /* raščisti sukobljavanje 7 if (cptr-
     >state == WAITING || cptr->state == RECEIVING || cptr->state == SENDING)
wakeup();
     break;
```

```
   case CLEAR_CONF: /* udaljeni korisnik se slaže s raskidanjem veze 7
      cptr->state = IDLE; break;

   case CREDIT: /* udaljeni korisnik čeka podatke */ cptr->credits +=
      data[1 ]; if (cptr->state == SENDING) wakeup{); break;

   case DATA_PKT: /* udaljeni korisnik je poslao podatke 7
        tor (i = 0; i < count; i++) cptr->user_buf_addr[cptr->byte_count + i] = datajij;
        cptr->byte_count += count; if (m == 0 ) wakeup();
   }
 }
 void clock(void)
 {/* Sat je otkucao, proveri tajmere zahteva za uspostavljanje veze u redu čekanja. 7
   int i;
   struct conn *cptr;

   for (i = 1; i <= MAX_CONN; i++) { cptr = &conn[ij;
        if (cptr->timer > 0) {                    /*   tajmer je aktivan 7
              cptr->timer;


              if (cptr->timer == 0) {
                                                  rok tajmera je istekao 7
                    cptr->state = IDLE; to_net(i, 0, 0,
                    CLEAR_REQ, data, 0);
              }
        }
   }
}
```

Slika 6-20. Primer transportne jedinice.

Prisustvo bita *Q* (Kvalifikator) u zaglavlju paketa omogućava da izbegnemo dodatno zaglavlje transportnog protokola. Obične poruke s podacima šalju se s tim bitom postavljenim na nulu. Upravljačke poruke transportnog protokola, od kojih u našem primeru postoji samo jedna (CREDIT), šalju se kao paketi s podacima uz $Q = 1$. Upravljačke poruke otkriva i obrađuje transportna jedinica.

Glavna struktura podataka koju koristi transportna jedinica jeste niz *conn,* koji ima po jedan zapis za svaku potencijalnu vezu. U zapisu se beleži stanje veze, što podra- zumeva adrese na oba kraja, broj poruka koje su primljene i poslate vezom, tekuće stanje, pokazivač korisnikovog bafera, broj primljenih ili poslatih bajtova tekuće poruke, bit koji označava daje udaljeni korisnik izvršio proceduru DISCONNECT, tajmer i brojač dozvola koji se koristi za omogućavanje slanja poruka. U našem jednostavnom primeru nećemo koristiti sva ova polja, ali su sva ona neophodna stvarnoj transportnoj jedinici, a možda treba da ih bude i više. Smatra se da je svaka odrednica niza *conn* ini- cijalizovana stanjem *IDLE.*

Kada korisnik pozove proceduru CONNECT, time nalaže mrežnom sloju da udaljenom računaru pošalje paket CALL REQUEST i zatim pada u san. Kada paket CALL, REQUEST stigne drugoj strani, njena transportna jedinica prinudno izvršava proceduru *packet_arrival* proveravajući da li lokalni korisnik osluškuje na naznačenoj adresi. Ako korisnik osluškuje, odgovara se paketom CALL ACCEPTED i udaljeni korisnik se budi; ako ne osluškuje,

CALL REQUEST se smešta u red čekanja tokom *TIMEOUT* otkucaja sata. Ako se u tom intervalu izvrši procedura LISTEN, uspostavlja se veza; u drugom slučaju, paketu ističe rok i on se odbacuje uz slanje paketa CLEAR REQUEST da pozivalac ne bi večito ostao blokiran.

Iako smo izbegli zaglavlje transportnog sloja, i dalje nam je potreban način evidentiranja paketa koji pripadaju određenoj transportnoj vezi, pošto istovremeno može da se uspostavi više veza. Najjednostavnije je da se broj virtuelnog kola mrežnog sloja upotrebi kao broj transportne veze. Štaviše, broj virtuelnog kola može se upotrebitii kao indeks za niz *conn*. Kada paket stigne virtuelnim kolom *k* mrežnog sloja, on pripada transportnoj vezi *k,* čije se stanje nalazi u zapisu *conn[k]*. Za veze koje inicira računar, broj veze bira transportna jedinica s koje veza potiče. Za dolazne pozive, mrežni sloj bira bilo koji nekorišđen broj virtuelnog kola.

Bafere u transportnoj jedinici obezbeđujemo i radimo s njima pomoću mehanizma kontrole toka različitog od uobičajenog protokola kliznih prozora.

Kada korisnik pozove RECEIVE, transportnoj jedinici pošiljaoca šalje se specijalna kreditna poruka (engl. *credit message*) i tamo zapisuje u niz *conn*. Kada se pozove SEND, transportna jedinica proverava da li je kredit stigao preko zadate veze. Ako je stigao, poroka se šalje (ako treba, u više paketa) i kredit umanjuje; ako nije stigao, transportna jedinica se uspavljuje dok kredit ne stigne. Takvim mehanizmom se garantuje da nijedna poruka neće biti poslata ako druga strana nije izvršila operaciju RECEIVE. Zbog toga, svaku poruku koja stigne čeka rezervisan bafer za smeštanje. Opisana šema se lako može uopštiti tako da primaoci obezbeđuju više bafera i zahtevaju više poruka.

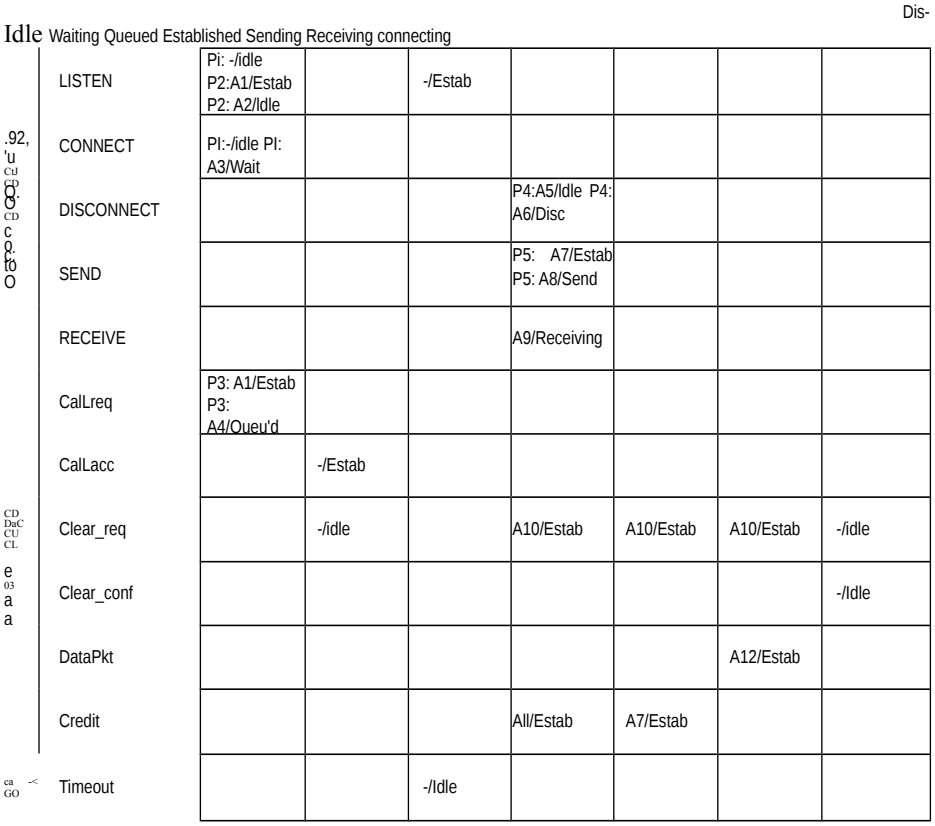
Imajte stalno na umu jednostavnost koda sa slike 6-20. Stvarna transportna jedinica realno bi proveravala svaki parametar koji stigne od korisnika, oporavljala sistem od havarija u mrežnom sloju, razrešavala sukobljavanje poziva, i podržavala opštiju uslugu prenosa, koja uključuje prekide, datagrame i verzije operacija SEND i RECEIVE koje se ne blokiraju.

6.3.3 Transportni protokol kao mašina konačnih stanja

Pisanje koda transportne jedinice predstavlja težak i zahtevan posao kada su u pitanju realistični protokoli. Da bi se umanjila mogućnost greške, često je korisno da se protokol predstavi mašinom konačnih stanja *(engl. finite state machine*).

Već smo videli da protokol iz našeg primera može da bude u sedam stanja po svakoj vezi. Takođe se može izdvojiti 12 događaja koji vezu mogu da prebace iz jednog stanja u drugo. Pet od njih su pet osnovnih operacija usluga. Preostalih šest su pristizanje paketa šest legalnih vrsta. Poslednji je isticanje roka tajmera. Na slici 6-21 prikazane su osnovne akcije protokola u obliku matrice. Kolone su stanja, a redovi predstavljaju događaje.

Stanje



	Idle	Waiting	Queued	Established	Sending	Receiving	Dis-connecting
LISTEN	Pi: -/idle P2:A1/Estab P2: A2/Idle		-/Estab				
CONNECT	PI:-/idle PI: A3/Wait						
DISCONNECT				P4:A5/Idle P4: A6/Disc			
SEND				P5: A7/Estab P5: A8/Send			
RECEIVE				A9/Receiving			
CalLreq	P3: A1/Estab P3: A4/Queu'd						
CalLacc		-/Estab					
Clear_req		-/idle		A10/Estab	A10/Estab	A10/Estab	-/idle
Clear_conf							-/Idle
DataPkt						A12/Estab	
Credit				All/Estab	A7/Estab		
Timeout			-/Idle				

Predikati

P1: Tabela veza puna
P2: Očekivanje Call_req
P3: Očekivanje LISTEN
P4: Očekivanje Clear_req
P5: Kredit na raspolaganju

Akcije

A1: Šalji CalLacc
A2: Čekaj Call_req
A3: Šalji Call_req
A4: Pokreni tajmer
A5: Šalji Clear_conf
A6: Šalji Clear_req
A7: Šalji poruku
A8: Čekaj kredit
A9: Šalji kredit
A10: Postavi indikator Clr_req_received
A11: Zabeleži kredit
A12: Prihvati poruku

Slika 6-21. Protokol iz primera kao mašina konačnih stanja. Svaka odrednica sadrži opcioni predikat, opcionu akciju i novo stanje. Tilda označava da nije preduzeta jedna od glavnih akcija. Podvučen predikat označava svoju negaciju. Prazne odrednice odgovaraju nemogućim ili pogrešnim događajima.

Svaki odrednica u matrici (mašini konačnih stanja) na slici 6-21 ima do tri polja: predikat, akciju i novo stanje. Predikat označava uslove potrebne za preduzimanje akcije. Na primer, u gornjoj levoj odrednici, ako se izvrši operacija LISTEN, a nema prostora u tabeli (predikat *Pl),* operacija završava neuspešno i stanje se ne menja.

S drage strane, ako je za adresu koja se osluškuje već stigao paket CALL REQUEST (predikat *P2),* veza se odmah uspostavlja. Draga mogućnost je daje predikat *P2* pogrešan, tj. da nije stigao paket CALL REQUEST, pa veza ostaje u stanju *IDLE,* očekujući paket CALL REQUEST.

Treba naglasiti da stanja u matrici nisu potpuno fiksirana samim protokolom. U ovom primera nema stanja *LISTENING* za koje bi se očekivalo da normalno sledi operaciju LISTEN. Tog stanja nema zato što je stanje povezano sa odrednicom zapisa o vezi, a LISTEN ne pravi zapis o vezi. Zašto ne pravi? Pa, zato što smo odlučili da brojeve virtuelnih kola mrežnog sloja iskoristimo kao identifikatore veza, a za operaciju LISTEN mrežni sloj bira broj virtuelnog kola tek kada stigne paket CALL. REQUEST.

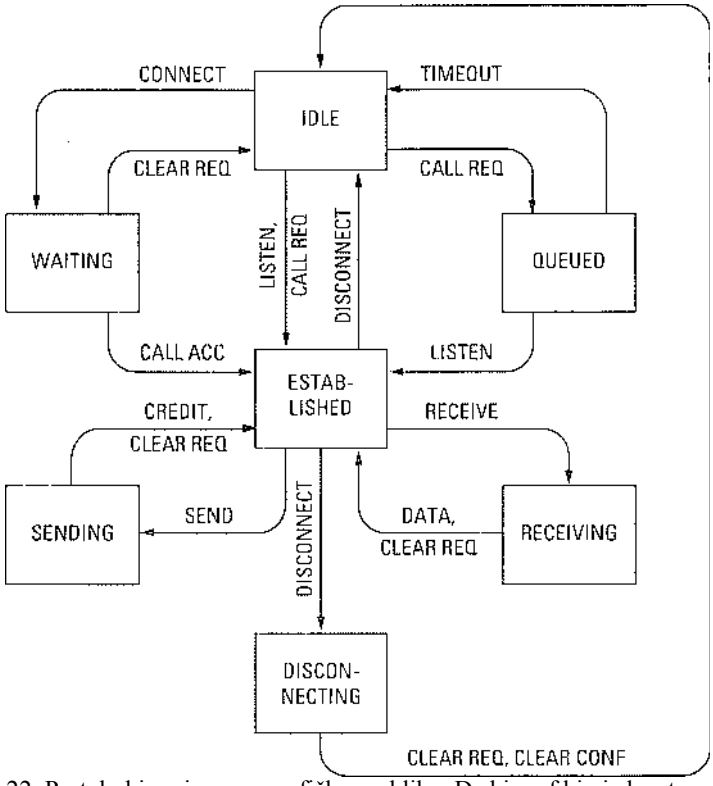
Akcije *A1* do *A12* su glavne akcije, kao što su slanje paketa i pokretanje tajmera. Nisu nabrojane sve sporedne akcije, kao što je, na primer, inicijalizivanje polja zapisa o vezi. Ako akcija podrazumeva buđenje usnulog procesa, računaju se i akcije koje slede buđenje. Na primer, ako stigne paket CALL REQUEST, a proces koji ga očekuje je uspavan, slanje paketa CALL ACCEPT posle buđenja procesa računa se kao deo akcije CALL REQUEST. Kao što se vidi na slici 6-21, pošto se akcija izvrši, veza može da pređe u novo stanje.

Predstavljanje protokola u obliku matrice ima tri prednosti. Prvo, programer može lako i sistematski da proveri svaku kombinaciju stanja, čak i da uvidi daje potrebna neka akcija. U realnim situacijama, neke od kombinacija bile bi iskorišćene za obradu grešaka. Na slici 6-21 nije pravljena razlika između nemogućih i nedozvoljenih situacija. Na primer, ako je veza u stanju čekanja *{waiting),* događaj DISCONNECT nije moguć jer je korisnik blokiran i ne može da izvrši nijednu operaciju. S druge strane, u stanju slanja *(sending),* ne očekuju se paketi s podacima jer za njih nije poslat kredit. Pristizanje takvog paketa je greška protokola.

Druga prednost matričnog predstavljanja protokola pojavljuje se pri njegovoj ugradnji. Možete da zamislite dvodimenzionalan niz u kome svaki element a[;][/] predstavlja pokazivač na proceduru za obradu događaja *i* kada se sistem nalazi u stanju *j.* Jedna moguća realizacija transportne jedinice je kratka programska petlja za očekivanje događaja. Kada se nešto dogodi, locira se odgovarajuća veza i izvlači njeno stanje. Uz poznat događaj i poznato stanje, transportna jedinica u nizu *a* lako pronalazi odgovarajuću proceduru za obradu. Takav pristup je mnogo ortodolcsniji i sistematičniji od projekta transportne jedinice iz našeg primera.

Treća prednost prikazivanja protokola kao mašine konačnih stanja ogleda se u njegovom opisivanju. U dokumentaciji koja prati neke standarde, protokoli se opisuju u obliku mašine konačnih stanja, slične matrici sa slike 6-21. Prelazak s takvog opisa na upotrebljivu transportnu jedinicu mnogo je lakši kada se stvarna transportna jedinica zasniva na standardizovanoj mašini konačnih stanja.

Osnovni nedostatak predstavljanja protokola u obliku mašine konačnih stanja jeste možda njegova srazmerna nerazumljivost u odnosu na programski primer koji smo ponudili. Međutim, taj problem se može delimično resiti ako mašinu konačnih stanja predstavimo grafom, kao na slici 6-22.



Slika 6-22. Protokol iz primera u grafičkom obliku. Da bi graf bio jednostavniji, nisu prikazani prelasci posle kojih se ne menja stanje veze.

6.4 TRANSPORTNI PROTOKOLI ZA INTERNET: **UDP**

Internet ima dva glavna protokola **u** transportnom sloju - jedan koji radi bez uspostavljanja direktne veze i drugi, s direktnom vezom. U narednim odeljcima raz- motrićemo oba. Protokol za rad bez uspostavljanja direktne veze je UDP, a onaj sa uspostavljanjem direktne veze - TCP. Pošto je UDP u osnovi isto što i IP, uz dodato kratko zaglavlje, počec'emo od njega. Razmotrićemo i dve primene UDP protokola.

6.4.1 Uvod u protokol UDP

U skupu protokola za Internet nalazi se i transportni protokol koji radi bez uspostavljanja direktne veze, tzv. **protokol za korisničke datagrame** (engl. *User Data- gram Protocol, UDP).* Protokol UDP omogućava aplikacijama da šalju kapsulirane IP datagrame za koje ne moraju prethodno da uspostavljaju vezu. Protokol UDP opisan je **u** RFC dokumentu 768.

UDP prenosi segmente koji se sastoje od 8-bajtnog zaglavlja i korisničkih podataka. Zaglavlje je prikazano na slici 6-23. Po jedan priključak (engl. *port)* na izvorišnom i na odredišnom računam identifikuju dva kraja „veze". Kada stigne UDP paket, njegov koristan teret predaje se procesu koji je pridružen priključku. Pridruživanje se vrši operacijom BIND

ili nekom sličnom osnovnom operacijom, kao što je prikazano na slici 6-6 za protokol TCP (proces povezivanja izgleda isto za UDP). U stvari, glavna prednost protokola UDP nad osnovnim IP protokolom leži u tome što on paketima dodaje izvorišni i odredišni priključak. Bez tih polja u zaglavlju, transportni sloj ne bi znao šta da radi sa paketima. Ovako, on ih isporučuje na pravu adresu.

Slika 6-23. Zaglavlje UDP paketa,

Izvorišni priključak u zaglavlju uglavnom je potreban kada treba poslati odgovor pošiljaocu. Kopirajući *izvorišni priključak* iz dolaznog segmenta u *odredišni priključak* odgovora, proces koji šalje odgovor može lako da naznači proces koji na dragom kraju treba da ga dobije.

Dužina UDP paketa obuhvata 8-bajtno zaglavlje i podatke. *Kontrolni zbir UDP paketa* nije obavezan i prikazuje se nulom kada nije izračunavan (kada je kontrolni zbir stvarno nula, to se prikazuje samim jedinicama). Isključivanje izračunavanja kontrolnog zbira nije preporučljivo, osim ako vam nije važan kvalitet podataka (npr. kod digitalizovanog glasa).

Verovatno odmah treba jasno reći šta UDP *ne radi.* On ne upravlja tokom, ne kon- troliše greške i ne šalje ponovo pogrešno primljene segmente. Sve to prepušta korisničkim procesima. On, međutim, predstavlja interfejs ka IP protokolu i dodatno demultipleksira procese koji istovremeno koriste isti priključak. I to je sve. Za aplikacije koje zahtevaju preciznu kontrolu toka paketa, grešaka ili vremenskog usklađivanja, protokol UDP radi tačno ono što mu naloži „viša instanca".

Protokol UDP je posebno koristan u klijentsko-serverskim kombinacijama. Klijent često pošalje serveru kratak zahtev i od njega očekuje isto tako kratak odgovor. Ako se zahtev ili odgovor izgube, klijentov tajmer automatski će se isključiti i on će zahtev poslati ponovo. Ne samo da je programski kod u ovom slučaju jednostavniji, već je i za obavljanje posla potrebno manje poruka (po jedna u svakom smera), nego uz protokol koji zahteva prethodno uspostavljanje veze.

Sistem imena domena (DNS), koji ćemo proučiti u 7. poglavlju, primer je aplikacije koja na takav način koristi protokol UDP. Ukratko, program koji truba da pronađe IP adresu koja odgovara imenu nekog računara, npr. imenu www.cs.berkeley.edu, može da pošalje UDP paket sa imenom DNS serveru. Server odgovara UDP paketom sa IP adresom tog računara. Za prethodnim uspostavljanjem veze nema potrebe, niti za njenim naknadnim raskidanjem. Preko mreže se samo razmene dve jednostavne poruke.

6.4.2 Daljinsko pozivanje procedure

Slanje poruke udaljenom računaru i dobijanje odgovora od njega u izvesnom smislu podseca na pozivanje funkcije u nekom programskom jeziku. U oba slučaja polazite od jednog ili više parametara i na kraju dobijate rezultat. To zapažanje je podstaklo programere da mrežne transakcije tipa zahtev-odgovor uobliče kao pozive procedurama. Takav način

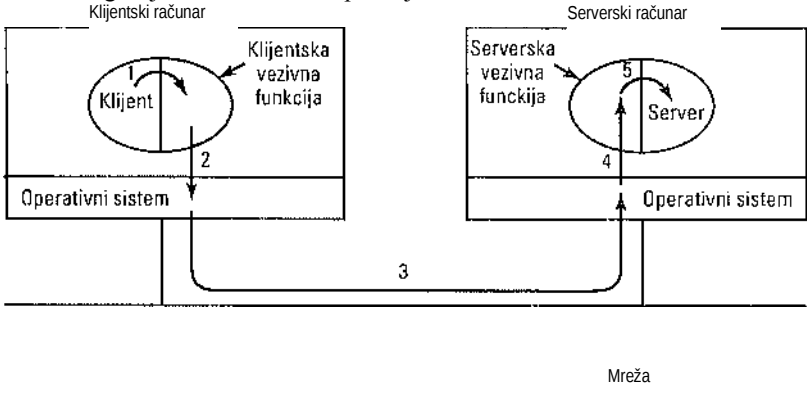
rada umnogome olakšava programiranje mrežnih aplikacija i rad s njima približava korisnicima. Zamislite, na primer, procedura *uzmi_IP_ adre- su(ime_racunara),* koja radi tako što DNS servera pošalje zahtev i očekuje odgovor, ali ako odgovor ne stigne dovoljno brzo, stalno se automatski isključuje i ponovo šalje zahtev. Na taj način, svi detalji mrežnog rada ostaju skriveni od programera.

Glavni posao u ovoj oblasti obavili su Birell i Nelson (1984). Oni su u osnovi predložili sistem u kome se programima dozvoljava da pozivaju procedure smeštene na udaljenim računalima. Kada proces na računaru 1 pozove proceduru na računaru 2, on se privremeno zaustavlja, a izvršava se pozvana procedura na računaru 2. Pozi- valac drugoj strani može da pošalje podatke u obliku parametara i da ih od nje dobije u obliku rezultata procedure. Programer uopšte ne vidi razmenu poruka. Tehnika koja je poznata kao **daljinsko pozivanje procedure** (engl. *Remote Procedure Call*, *RPC),* postala je osnova za mnoge mrežne aplikacije. Procedura koja šalje poziv obično se naziva klijentom, a pozvana procedura serverom, pa ćemo se i mi toga pridržavati.

Tehnikom RPC pokušava se da pozivanje procedure na udaljenom računaru bude što sličnije pozivanju lokalne procedure. U njenom najjednostavnijem obliku, da bi klijentski program mogao da daljinski pozove proceduru, mora u biblioteci imati jednu drugu kratku proceduru, tzv. **klijentsku vezivnu funkciju** (engl. *client stub)* koja reprezentuje serverslcu procedura u klijentskom adresnom prostoru. Slično tome, ser- ver je opremljen **serverskom vezivnom funkcijom** (engl. *server stub).* Te procedure maskiraju činjenicu da poziv klijenta serveru nije lokalan poziv.

Konkretni koraci sprovođenja tehnike RPC prikazani su na slici 6-24. U koraku 1, klijent poziva klijentsku vezivnu funkciju. To je pozivanje lokalne procedure, uz uobičajeno smeštanje parametara na stelc. U koraku 2, klijentska vezivna funkcija pakuje parametre u poruku i upućuje sistemski poziv za njeno slanje. Pakovanje parametara naziva se ustrojavanje (engl. *marshaling).* U koraku 3, jezgro operativnog sistema šalje poruku s klijentskog računara na server. U koraku 4, jezgro operativnog sistema servera prosleđuje dolazni paket serverskoj vezivnoj funkciji. Na kraju, u koraku 5, serverska vezivna funkcija poziva serverslcu proceduru uz parametre „izvučene iz stroja". Odgovor sledi istu putanju u obrnutom smeru.

Ovde je najvažnije to što klijentska procedura (koju piše korisnik) na najnormalniji način (tj. lokalno) poziva klijentsku vezivnu funkciju koja nosi isto ime kao i procedura na serveru. Pošto su klijentska procedura i klijentska vezivna funkcija u istom adresnom prostora, parametri se prosleđuju na uobičajen način. Slično tome, server- slcu proceduru poziva procedura iz njegovog adresnog prostora, uz očekivane parametre. Serverskoj proceduri sve izgleda normalno. Na ovaj način se imitiranjem poziva lokalnoj proceduri, u komunikaciji preko mreže izbegavaju ulazno-izlazne operacije sa utičnicama.



Slika 6-24. Koraci za daljinsko pozivanje procedure. Vezivne funkcije su prikazane sivo,

Uprkos spoljnoj eleganciji, tehnika RPC ima i nedostataka. Jedan od većih problema je upotreba pokazivača kao parametara. Pokazivač se normalno može proslediti proceduri koja će ga koristiti na isti način kao i pozivalac jer se i procedura i pozivalac nalaze u istom virtuelnom adresnom prostoru. Tehnikom RPC, međutim, pokazivači se ne mogu prosleđivati jer se klijent i server nalaze u različitim adresnim prostorima.

Pokazivači se u nekim slučajevima mogu proslediti upotrebom trikova. Pretpostavimo daje prvi parametar pokazivač na ceo broj *k.* Klijentska vezivna funkcija može da ustroji broj A: i da ga pošalje serveru. Serverska vezivna funkcija tada pravi pokazivač na *k* i prosleđuje ga serverskoj proceduri, baš kao što ova očekuje. Kada se programski tok vrati sa serverske procedure na serversku vezivnu funkciju, ova druga može da klijentu vrati broj *k* koji će prebrisati njegovu staru vrednost (za slučaj da ju je server izmenio). U stvari, standardan način pozivanja po referenci zamenjen je postupkom kopiranja i ponovnog uspostavljanja vrednosti. Opisani trik, nažalost, ne radi uvek - na primer, kada pokazivač ukazuje na graf ili na neku drugu složenu strukturu podataka. Zbog toga se parametrima koji se prosleđuju uz pozive udaljenim procedurama, moraju postaviti izvesna ograničenja.

Drugi problem stvaraju skromno tipizirani jezici, kao što je jezik C, u kojima je savršeno ispravno napisati proceduru koja izračunava skalarni proizvod dva vektora (niza), ne zadajuci njihovu veličinu. Svaki se može završiti specijalnom vrednošću poznatom samo proceduri koja poziva i proceduri koja je pozvana. U takvim uslovi- ma, klijentska vezivna funkcija suštinski ne može da ustroji parametre, tj. nema načina da utvrdi njihovu veličinu.

Treći problem je u tome što nije uvek moguće, čak ni pomoću formalne specifikacije ili i samog koda, utvrditi tipove parametara. Primer je procedura *printf,* koja može da ima proizvoljan broj parametara (najmanje jedan), a parametri mogu da budu smeša kratkih i dugačkih celih brojeva, slova, nizova, decimalnih brojeva i drugih tipova podataka. Daljinsko pozivanje procedure *printf* bilo bi praktično nemoguće zato što je jezik C tako tolerantan. Međutim, pravilo koje dozvoljava RPC procedura pod uslovom da ne programirate na jeziku C (ili na jeziku C++), bilo bi nepopularno.

Četvrti problem tiče se globalnih promenljivih. Procedura pozivalac i pozvana procedura, osim pomoću parametara, mogu normalno da međusobno komuniciraju i pomoću globalnih promenljnivih. Kada se pozvana procedura zatim premesti na udaljeni računar, kod neće raditi jer procedure više ne dele globalne promenljive.

Navedeni problemi ne znače da je daljinsko pozivanje procedura neupotrebljivo. U stvari, ta tehnika se široko koristi, ali su u praksi neophodna neka ograničenja da bi ona dobro radila.

Daljinsko pozivanje procedura ne mora se vršiti razmenom UDP paketa, ali to dvoje dobro sarađuju i obično se zajedno koriste. Međutim, kada parametri ili rezultati premaše maksimalnu veličinu UDP paketa, ili ako zahtevana operacija, npr. uvećavanje brojača za jedinicu, ne može da se ponavlja na identičan način, možda je potrebno uspostaviti TCP vezu i zahtev poslati njom.

6.4.3 Protokol za prenos u realnom vremenu

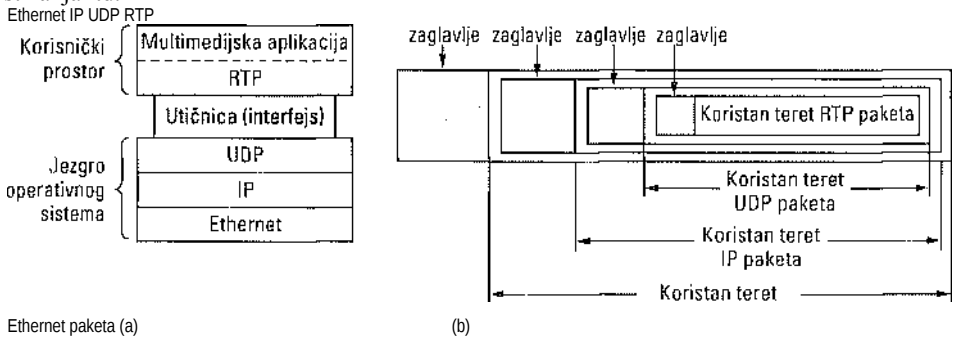
Daljinsko pozivanje procedura između klijenta i servera predstavlja jedno područje u kome se široko koristi protokol UDP. Drugo područje čine multimedijske aplikacije koje rade

u realnom vremenu. Dok je rasla popularnost Internet radija, Internet telefonije, muzike na zahtev, video-konferencija, videa na zahtev i drugih multimedijskih aplikacija, za svaku od tih aplikacija uvek je iznova izmišljan približno isti protokol za prenos podataka. Ubrzo je postalo jasno da bi bilo dobro imati opšti protokol za prenos podataka u realnom vremenu primenljiv za različite aplikacije. Tako je nastao **protokol za prenos u realnom vremenu** (engl. *Real-time Transport Protocol, RTP*), koji je opisan u RFC dokumentu 1889, a danas je u širokoj upotrebi.

Položaj protokola RTP u skupu protokola pomalo je neuobičajen. Odlučeno je da se on smesti u korisnički prostor i da se (normalno) izvršava pomoću protokola UDP. Radi na sledeći način. Multimedijsku aplikaciju sačinjava više zvučnih, video, tekstualnih i drugih tokova. Oni se upućuju u RTP biblioteku, koja se nalazi u korisničkom prostoru, zajedno sa aplikacijom. Biblioteka tada multipleksira tokove i kodira ih u RTP pakete koje zatim stavlja u utičnicu. S druge strane utičnice (u jezgra operativnog sistema), generišu se UDP paketi i smeštaju u IP pakete. Ako je računar na Ethernetu, IP paketi se stavljaju u Ethernet okvire koji se šalju. Skup protokola za opisanu situaciju prikazan je na slici 6-25(a). Ugnežđivanje paketa prikazano je na slici 6-25(b).

Zbog ovakvog rasporeda, teško je reći u kom se sloju nalazi protokol RTP. Pošto se izvršava u korisničkom prostoru i povezan je sa aplikacijom, izvesno je da liči na protokol sloja aplikacija. S druge strane, on je opšti protokol, nezavistan od bilo koje konkretne aplikacije, koji samo obezbeđuje transportne usluge, tako da liči na protokol transportnog sloja. Verovatno je najbolje smatrati ga transportnim protokolom ugrađenim u sloj aplikacija.

Osnovni zadatak protokola RTP jeste multipleksiranje u realnom vremenu više tokova podataka u jedinstven tok UDP paketa. UDP tok se može poslati na jedinstveno odredište (jednosmerno emitovanje) ili na više odredišta (višesmerno emitovanje). Pošto se protokol RTP služi UDP protokolom na uobičajen način, usmerivači ne obraćaju posebnu pažnju na njegove pakete, osim ako se aktiviraju neke uobičajene klase kvaliteta IP usluga. To znači da nema nikakvih posebnih garancija u pogledu isporuke paketa, ravnomernosti njihovog stizanja itd.



Slika 6-25. Položaj protokola RTP u skupu protokola, (b) Ugnežđivanje paketa.

Svakom paketu koji se šalje RTP tokom, dodeljuje se broj za jedan veći nego njegovom prethodniku. Numeracija omogućava odredištu da utvrdi da li su svi paketi na broju. Ako neki paket nedostaje, odredište će najbolje uraditi ako ga interpolira. Ponovno slanje nije

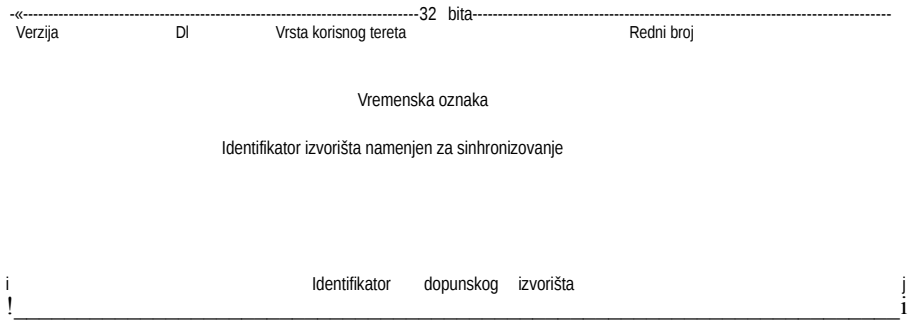
praktično, pošto bi ponovo poslat paket verovatno stigao prekasno da bi išta koristio. Zbog toga, u protokolu RTP nema upravljanja tokom, nema kontrole grešaka, nema potvrđivanja paketa, niti mehanizma za zahtevanje ponovnog slanja paketa.

Koristan teret svakog RTP paketa može da sadrži više uzoraka kodiranih na način kako zahteva aplikacija. U cilju obezbeđivanja saradnje, RTP definiše više profila (npr. jedinstven audio tok), a za svaki profil može se dozvoliti više formata. Na pri- mer, jedinstven audio tok može biti u formatu 8-bitnih PCM uzoraka pri 8 kHz, delta kodiran, prediktivno kodiran, GSM kodiran, u formatu MP3 itd. U zaglavlju RTP paketa postoji polje u kome izvorište može da naznači način kodiranja, ali sam protokol ne vrši kodiranje.

Dmga osobina potrebna mnogim aplikacijama koje rade u realnom vremenu jeste vremensko označavanje (engl. *timestamping).* Drugim recima, izvorište treba da vremenski označi prvi uzorak u svakom paketu. Vreme se označava u odnosu na početak toka, tako da su od koristi samo razlike između vremenskih oznaka - njihove apsolutne vrednosti nemaju smisla. Ovaj mehanizam omogućava odredištu da u izvesnoj meri privremeno skladišti pakete i da svaki uzorak reprodukuje nakon tačnog vremena od početka toka, nezavisno od toga kada je koji paket stvarno stigao. Vremensko označavanje ne samo što ublažuje efekat neravnomemog pristizanja paketa, već omogućava i da se više istovremenih tokova međusobno sinhronizuju. Na primer, digitalni televizijski program može da ima video tok i dva audio toka. Dva audio toka mogu da posluže za difuzno emitovanje sterofonskog zvuka ili za filmove kod kojih se gledaocu

omogućava da bira originalni jezik ili sinhronizovani prevod. Svaki tok potiče iz zasebnog fizičkog uređaja, ali ako ih vremenski označava isti brojač, oni se mogu reprodukovati sinhrono, čak i ako neravnomerno stignu na odredište.

RTP zaglavlje je prikazano na slici 6-26. Ono se sastoji od tri 32-bitne reči, a dozvoljena su i dodatna zaglavlja. Prva reč sadrži polje *Verzija,* koje već ima vrednost 2. Nadajmo se da verzije neće više često smenjivati jedna drugu jer je u polju preostalo samo još jedno kodno mesto (iako bi se moglo definisati da vrednost 3 znači da se broj verzije nalazi u 32-bitnoj reči dodatnog zaglavlja).



Slika 6-26. Zaglavlje RTP paketa.

Bit *D* označava da lije paket bio dopunjavan do umnoška od 4 bajta. Poslednji bajt dopune saopštava koliko je ukupno dodato bajtova. Bit Z ukazuje da postoji dodatno zaglavlje. Ne definišu se njegov format i značenje, već samo to da njegova prva reč označava dužinu. To je bilo solomonsko rešenje za slučaj da se pojave zahtevi koje nilco nije mogao unapred da predvidi.

Polje *Dl* govori koliko ima dopunskih izvora, 0 do 15 (videti u nastavku). Bit *0* je rezervisan za oznaku koju na to mesto postavlja aplikacija. Ona ga može iskoristiti da označi početak video okvira, početak reči na audio kanalu ili nešto treće što samo ona razume. *Vrsta korisnog tereta* govori o upotrebljenom algoritmu (npr. nekomprimo- vani 8-bitni audio, MP3 i slično). Pošto svaki paket ima ovo polje, tip korisnog tereta može se menjati tokom prenosa. *Redni broj je* samo brojač čija vrednost raste za jedan svaki put kada se pošalje RTP paket. On se koristi za otkrivanje izgubljenih paketa.

Vremensku oznaku generiše izvorište toka podataka kada napravi prvi uzorak koji paket nosi. Ona pomaže da se ublaži efekat neravnomernog stizanja paketa primaocu, tako što razdvaja proces reprodukovanja od procesa stizanja paketa. *Identifikator izvora namenjen za sinhronizovanje* ukazuje na tok kome pripada paket. Ta metoda se koristi za multipleksiranje i demultipleksiranje više tokova podataka u jedinstvenom toku UDP paketa. I na kraju, *Identifikatori dopunskih izvorišta,* ako ih ima, koriste se ako u studiju postoji uređaj za miksovanje. U tom slučaju, uređaj za miksovanje vrši i sinhronizovanje, a u polje se upisuju miksovani tokovi.

RTP ima mali „sestrinski“ protokol, koji se zove **protokol za upravljanje preno- som u realnom vremenu** (engl. *Real-time Transport Control Protocol, RTCP).* On povratno šalje informacije, sinhronizuje tok i obezbeđujue korisničko okruženje, ali ne prenosi podatke. Njegova prva funkcija može se iskoristiti za obaveštavanje izvorišta o kašnjenju, neravnomernosti pristizanja paketa, propusnom opsegu, zagušenju i dragim svojstvima mreže. Te informacije može da iskoristi proces koji kodira podatke da bi ih brže (i kvalitetnije) slao kada mreža radi dobro ili da bi brzinu slanja smanjio kada na mreži nastanu problemi. Kada se povratne informacije neprekidno šalju, algoritam za kodiranje može stalno da se prilagođava da bi u datim okolnostima obezbedio najviši mogući kvalitet. Na primer, ako se tokom prenosa menja širina propusnog opsega, za kodiranje se, u zavisnosti od njegove trenutne širine, može upotrebiti MP3, 8-bitna PCM ili delta tehnika. Kao što smo već rekli, tehnika kodiranja aktuelnog paketa naznačuje se u polju *Vrsta korisnog tereta,* tako da se može menjati po potrebi.

Protokol RTCP, takođe, međusobno sinhronizuje više tokova. Problem je u tome što različiti tokovi mogu koristiti različite satove, s različitim trajanjem jednog otkucaja i različitim odstupanjem. Protokol RTCP može da ih sve usaglasi.

Na kraju, protokol RTCP omogućava imenovanje različitih izvorišta (npr. tekstualnim imenima). Takvo ime se može uputiti na ekran primaoca da bi znao s kim trenutno razgovara.

Više informacija o protokolu RTP možete naći kod Perkinsa (2002).

6.5 TRANSPORTNI PROTOKOLI ZA INTERNET: TCP

UDP je jednostavan protokol koji ima svoje područje primene, npr. u klijentsko-serverskim interakcijama i multimediji, ali za većinu Internet aplikacija potrebna je pouzdana, uređena isporuka. UDP to ne može da obezbedi, pa je smišljen drugi protokol - TCP, koji je i glavna radna snaga Interneta. Razmotrimo ga detaljno.

6.5.1 Predstavljanje protokola TCP

Protokol za upravljanje prenosom (engl. *Transmission Control Protocol, TCP)* namenski je projektovan da obezbedi pouzdan tok bajtova s kraja na kraj „veze“ kroz nepouzdanu kombinovanu mrežu. Kombinovana mreža se razlikuje od jedinstvene mreže utoliko što njeni različiti delovi mogu imati drastično različite topologije, propusne opsege, kašnjenja, veličine paketa i druge parametre. TCP je projektovan tako da se dinamički prilagođava svojstvima kombinovane mreže i da bude otporan na mnoge havarije.

TCP je formalno definisan RFC dokumentom 793. Vremenom su otkrivane razne greške i nedoslednosti, a u nekim oblastima promenili su se i zahtevi. Razjašnjenja i ispravke nekih grešaka detaljno su opisani u RFC dokumentu 1122. Dopune su prikazane u RFC dokumentu 1323.

Svaki računar koji podržava protokol TCP ima i transportnu TCP jedinicu: kao proceduru u biblioteci, kao korisnički proces ili kao deo jezgra operativnog sistema. U sva tri slučaja, transportna TCP jedinica radi sa TCP tokovima i ostvaruje interfejs ka IP sloju. TCP jedinica prihvata tokove korisničkih podataka od lokalnih procesa, deli ih u blokove od najviše 64 KB (što u praksi često znači samo 1460 bajtova podataka da bi blok, zajedno sa IP i TCP

zaglavljima, stao u Ethernet okvir) i blokove šalje kao zasebne IP datagrame. Kada datagrami sa TCP podacima stignu u računar, predaju se TCP jedinici koja iz njih rekonstruiše originalne tokove bajtova.

IP sloj ne garantuje da će datagrami biti ispravno isporučeni, tako da transportna TCP jedinica mora da aktivira tajmere i da ih ponovo šalje ako zatreba. Datagrami na odredište mogu da stignu bilo kojim redom, a tu TCP jedinica treba da ih presloži ispravnim redosledom i da od njih sastavi poruku. Ukratko, protokol TCP mora da obezbedi pouzdanost koju želi većina korisnika, a koju ne obezbeđuje IP protokol.

6.5.2 Model TCP usluge

Kao što smo videli u odeljku 6.1.3, TCP usluga oživljava tako što i pošiljalac i primalac naprave krajnje tačke veze, tzv. utičnice. Svaka utičnica ima svoj broj (adresu), koji se sastoji od IP adrese računara i 16-bitnog lokalnog broja, zvanog priključak (engl. *port)*. Priključak je TCP naziv za TSAP. Da bi se ostvarila TCP usluga, između utičnice pošiljaoca i utičnice primaoca mora se izričito uspostaviti veza. Pozivi utičnicama navedeni su na slici 6-5.

Utičnica se može koristiti za više istovremenih veza. Drugim recima, dve ili više veza mogu završavati u istoj utičnici. Veze se raspoznaju prema identifikatorima utičnica na oba kraja (*utičnica*!, *utičnica!)*. Ne koriste se brojevi virtuelnih kola ni drugi identifikatori.

Priključci s brojem manjim od 1024 zovu se opštepoznati priključci (engl. *well- -known ports*) i rezervisani su za standardne usluge. Na primer, proces koji želi da uspostavi vezu s računarom da bi mu poslao datoteku protokolom FTP, može da se poveže na priključak 21 odredišnog računara i da pristupi sistemskoj FTP usluzi. Spisak opštepoznatih priključaka nađi čete na Web lokaciji www.iana.org. Definisano ih je prelco tri stotine. Nekoliko poznatijih navedeni su na slici 6-27.

Priključak	Protokol	Upotreba
21	FTP	Prenos datoteka
23	Telnet	Daljinsko prijavljivanje
25	SMTP	E-pošta
69	TFTP	Trivijalni protokol za prenos datoteka
79	Finger	Traženje informacija o korisniku
80	HTTP	Web
110	POP-3	Daljinski pristup e-pošti
119	NNTP	Diskusione grupe

Slika 6-27. Neki dodeljeni priključci.

Sigurno bi bilo moguće da se pri podizanju računara sistemska FTP usluga pridruži priključku 21, da se usluga telnet pridruži priključku 23 itd. Međutim, tako bi se memorija ispunila sistemskim uslugama koje većinu vremena ne rade ništa. Umesto toga, jedinstvena usluga koja se u UNIX-u zove *inetd* (od engl. *Internet daemon* **- sistemska usluga Interneta),** pridružuje se mnogim priključcima i čeka prvu dolaznu vezu. Kada se pojavi zahtev za uspostavljanje veze, inetd generiše nov proces i u njemu izvršava odgovarajuću sistemsku uslugu kojoj prepušta obradu zahteva. Na taj način, pored usluge inetd pojavljuju se samo sistemske usluge koje stvarno treba nešto da rade. Inetd zna koji će priključak za šta da upotrebi jer čita konfiguracionu datoteku. Shodno tome, administrator sistema može da trajno pridruži neke sistemske usluge najzaposlenijim priključcima (npr. priključku 80), a da ostale priključke prepusti usluzi inetd.

Sve TCP veze potpuno su dupleksne i tipa od tačke do tačke. Puni dupleks znači da se saobraćaj istovremeno može odvijati u oba smera. Od tačke do tačke znači da svaka veza ima samo dve krajnje tačke. Protokol TCP ne podržava višesmerno ili neusmereno (difuzno) emitovanje.

TCP veza prenosi tok bajtova, a ne tok poruka. Tokom prenosa se ne čuvaju granice poruka. Na primer, ako pošiljalac u TCP tok unese četiri zapisa od po 512 bajtova, oni se primaocu mogu isporučiti kao četiri bloka od po 512 bajtova, kao dva bloka od po 1024 bajta, kao jedan blok od 2048 bajtova (slika 6-28) ili na neki dragi način. Primalac nikako ne

IP zaglavlje

može da utvrdi jediničnu količinu podataka koje šalje draga strana.

TCP zaglavlje

A B C D

(b) (a)

Slika 6-28. (a) Četiri segmenta od po 512 bajtova poslati kao zasebni IP datagrami.
(b) Jednim pozivom procedure READ, aplikacija učitava svih 2048 bajtova podataka.

Datoteke u UNIX-u imaju isto svojstvo. Proces koji učitava datoteku ne može da utvrdi da li je datoteka upisana blok po blok, bajt po bajt ili odjednom cela. Slično UNIX-ovim datotekama, ni TCP softver ne tumači bajtove koje prenosi - njemu svaki bajt izgleda isto.

Kada aplikacija prosledi podatke prenosnoj TCP jedinici, ona može odlučiti da ih pošalje odmah ili da ih privremeno uskladišti dok ne prikupi veću količinu podataka koje će poslati zajedno. Međutim, ponekada aplikacija insistira na ažurnom slanju podataka. Pretpostavimo, na primer, da se korisnik prijavljuje na udaljeni računar. Kada unese podatke na komandnu liniju i pritisne taster ENTER, zgodno bi bilo da se oni odmah isporuče udaljenom računani, a ne da se uskladište i čekaju još podataka da bi svi zajedno bili poslati kasnije. Da bi prisilile TCP jedinicu da odmah pošalje prosleđene podatke, aplikacije mogu da upotrebe indikator (engl.*flag)* PUSH.

U nekim starijim aplikacijama, indikator PUSH se koristi za označavanje granice poruka. To, međutim, ne radi uvek jer se u nekim realizacijama TCP protokola indikator PUSH ne prosleđuje aplikaciji primaoca. Štaviše, ako se u transportnoj TCP jedinici nakupi više takvih indikatora (npr. kada je izlazna linija zauzeta), TCP jedinica „ima pravo“ da sve podatke označene indikatorom PUSH skupi u jedinstven IP datagram, ne razdvajajući pojedine delove.

Na kraju treba pomenuti još jednu važnu osobinu TCP usluge, a to je hitno slanje podataka (engl. *urgent data)*. Kada korisnik koji interaktivno radi sa udaljenom aplikacijom pritisne taster DEL ili CTRL-C da bi prekinuo već započeto izračunavanje, aplikacija pošiljalac umeće izvesne upravljačke informacije u tok podataka i predaje ga TCP jedinici zajedno sa indikatorom URGENT. Taj događaj prisiljava TCP jedinicu da prestane da skladišti podatke i da odmah drugoj strani pošalje sve što ima prikupljeno za tu vezu.

Kada podaci poslati hitnim postupkom stignu na odredište, aplikacija primalac prekida aktuelni posao (npr. u UNIX-u, dobija signal za prekid) i učitava tok tražeći hitne podatke. Kraj hitno poslatih podataka posebno je označen, tako da aplikacija zna gde se završavaju; međutim, njihov početak nije označen, pa aplikacija mora da se sama snađe. Opisana šema obezbeđuje osnovni, grub mehanizam signalizacije, a sve ostalo aplikacija treba da uradi sama.

6.5.3 Protokol TCP

U ovom odeljku pozabavićemo se protokolom TCP u glavnim crtama. U sledećem ćemo objasniti njegovo zaglavlje, polje po polje.

Čitav protokol TCP zasnovan je na ključnoj postavci da svaki bajt na TCP vezi ima svoj 32-bitni redni broj. Na početku ere Interneta, usmerivači su najčešće bili povezani iznajmljenim linijama brzine 56 kb/s, pa je računani koji emituje punom brzinom trebalo više od jedne sedmice da potroši sve redne brojeve. Danas, međutim, kada su brzine prenosa mnogo veće, i redni brojevi se, kao što ćemo kasnije videti, troše alarmantnom brzinom. Za potvrde 0 prijemu paketa i upravljanje prozorima koriste se zasebni 32-bitni redni brojevi.

Transportne TCP jedinice 11a dve strane veze izmenjuju podatke u obliku segmenata. TCP segment sadrži fiksno 20-bajtno zaglavlje (plus neobavezan deo) i podatke (kojih i ne mora da bude). Veličinu segmenta određuje TCP softver. On može da podatke iz više zapisa skupi u jedan segment ili da podatke iz istog zapisa podeli u više segmenata. Za veličinu segmenta postoje dva ograničenja. Prvo, svaki segment, zajedno sa TCP zaglavljem, mora da stane u polje za koristan teret IP paketa (maksimalno 65.515 bajtova). Drugo, za svaku mrežu postoji najveća jedinica prenosa (engl. *Maximum Transmission Unit, MTU),* koja se mora poštovati. MTU u praksi obično iznosi 1500 bajtova (veličina korisnog tereta Ethernet okvira), što definiše i maksimalnu veličinu TCP segmenta.

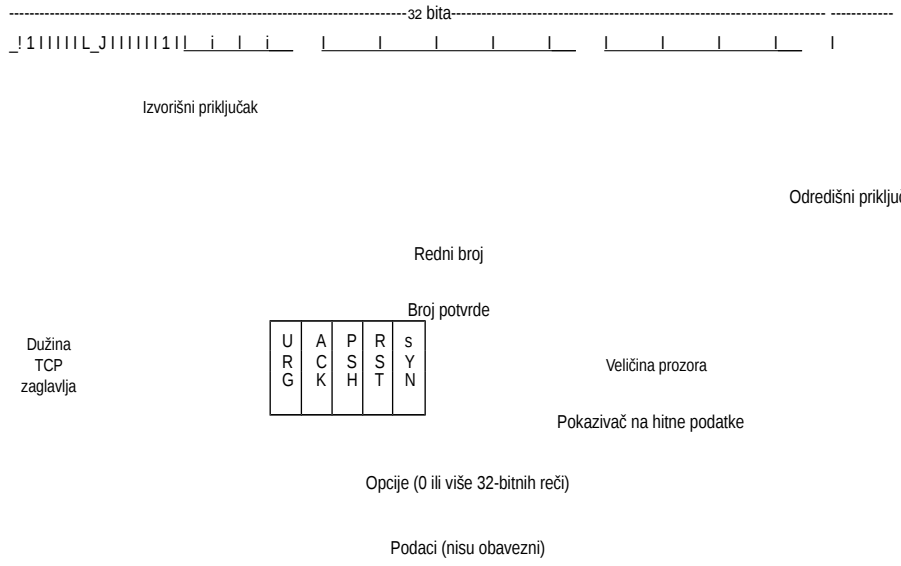
Kao svoj osnovni protokol, prenosne TCP jedinice koriste protokol kliznih prozora. Kada pošalje segment, pošiljalac aktivira i tajmer. Kada segment stigne na odredište, prijemna TCP jedinica odgovara potvrdom (uz koju se mogu slati i podaci), čiji redni broj odgovara rednom broju sledećeg segmenta koji očekuje. Ako se tajmer pošiljaoca isključi pre nego što ovaj dobije potvrdu, on ponovo šalje segment.

Iako protokol deluje jednostavno, postoji niz sitnica koje treba razrešiti. Segmenti mogu da stignu proizvoljnim redosledom, tako da pristigli bajtovi 3072—4095 ne mogu da budu potvrđeni dok se ne pojave i bajtovi 2048-3071. Putovanje segmenata može i da se oduži, pa je pošiljalac zbog isključivanja tajmera prinuđen da ih ponovo šalje. Pri ponovnom slanju, originalni podaci mogu da se drugačije segmentiraju, zbog čega je neophodno brižljivo pratiti šta je stiglo ispravno, a šta se mora ponovo slati. Iako složeno, takvo praćenje se može izvesti jer svaki bajt unutar toka ima svoj jedinstven redni broj.

Protokol TCP mora biti spreman da se efikasno izbori s navedenim problemima. Mnogo je truda uloženo u optimizovanje performansi TCP tokova, uprkos problemima koji postoje na mreži. U nastavku ćemo razmotriti više algoritama koji se za to koriste u mnogim realizacijama protokola TCP.

6.5.4 Zaglavlje TCP segmenta

Na slici 6-29 prikazana je organizacija TCP segmenta. Svaki segment počinje 20--bajtnim zaglavljem ustaljenog formata. Iza zaglavlja mogu da slede njegove opcije. Posle opcija, ako ih ima, dolazi najviše 65.495 bajtova podataka (65.53.5 - 20 - 20), gde prvih 20 bajtova otpada na IP zaglavlje, a drugih 20 na TCP zaglavlje. Segmenti bez podataka savršeno su ispravni - oni služe za potvrde i upravljanje.



Slika 6-29. TCP zaglavlje.

Analizirajmo TCP zaglavlje, polje po polje. *Izvorišni priključak* i *Odredišni priključak* identifikuju krajnje lokalne tačke veze. Opštepoznati priključci navedeni su na lokaciji www.iana.org, ali svaki računar može da dodeli priključke i drugim uslugama. Broj priključka plus IP adresa računara na kome se nalazi obrazuju jedinstvenu 48-bitnu krajnju tačku veze. Izvorišna i odredišna krajnja tačka zajedno definišu određenu vezu.

Funkcija polja *Redni broj* i *Broj potvrde* već je poznata. Skrećemo vam pažnju na to da *Broj potvrde* ne označava poslednji ispravno primljen bajt, već sledeći očekivani bajt. Oba broja su 32-bitna jer se u TCP toku svaki bajt zasebno nutneriše.

Dužina TCP zaglavlja označava broj 32-bitnih reći u zaglavlju. Taj podatak je neophodan zato stoje polje *Opcije* promenljive dužine, pa je samim tim promenljiva i dužina celog zaglavlja. *Dužina TCP zaglavlja* praktično označava mesto u segmentu (mereno 32-bitnim recima) od koga počinju podaci.

Zatim dolazi 6-bitno polje koje se ne koristi. Ono se već četvrt veka nije promenilo i predstavlja živi dokaz daje protokol TCP na samom početku dobro projektovan. Kod lošijih protokola to polje bi se koristilo za ispravljanje grešaka prvobitnog projekta.

Sledi šest jednobitnih indikatora. *URG* se postavlja na 1 kada se koristi *Pokazivač na hitne podatke* (engl. *urgent pointer)*. *Pokazivač na hitne podatke* sadrži priraštaj koji treba dodati tekućem rednom broju da bi se dobio redni broj segmenta s hitnim podacima i predstavlja zamenu za slanje zahteva za prekid. Kao što smo već naglasili, ovim jednostavnim mehanizmom jedna strana može drugoj da pošalje signal, a da se TCP pri tome ne pita šta je razlog prekida.

Kada je bit A *CK* jedinica, to znači daje *Broj potvrde* ispravan. Kada je *ACK* nula, segment ne sadrži potvrdu, pa se *Broj potvrde* zanemaruje.

Bit *PSH* označava da podatke treba odmah proslediti (PUSH). Od primaoca se zahteva da podatke ne čuva u baferu dok se ovaj ne napuni (što bi mogao činiti u cilju uvećanja efikasnosti prenosa), već da ih prosledi čim ih primi.

Pomoću bita *RSH ponovo* se uspostavlja veza oštećena zbog pada računara ili iz nekog drugog razloga. Taj bit se koristi i za odbijanje neispravnih segmenata, odnosno pokušaja da se uspostavi veza. Ukratko, ako dobijete segment s postavljenim bitom *RST,* to znači da negde postoji problem.

Bit *SYN* služi za uspostavljanje veza. Zahtev za uspostavljanje veze ima SYN = 1 i ACK = 0, što znači da se ne koristi polje za šlepovanje potvrde. Odgovor na zahtev, međutim, nosi potvrdu, tako da je u njemu *SYN* = 1 i *ACK* = 1. Bitom *SYN* u suštini se označavaju obe poruke: CONNECTION REQUEST i CONNECTION ACCEPTED, dole bit ACK služi za njihovo razlikovanje.

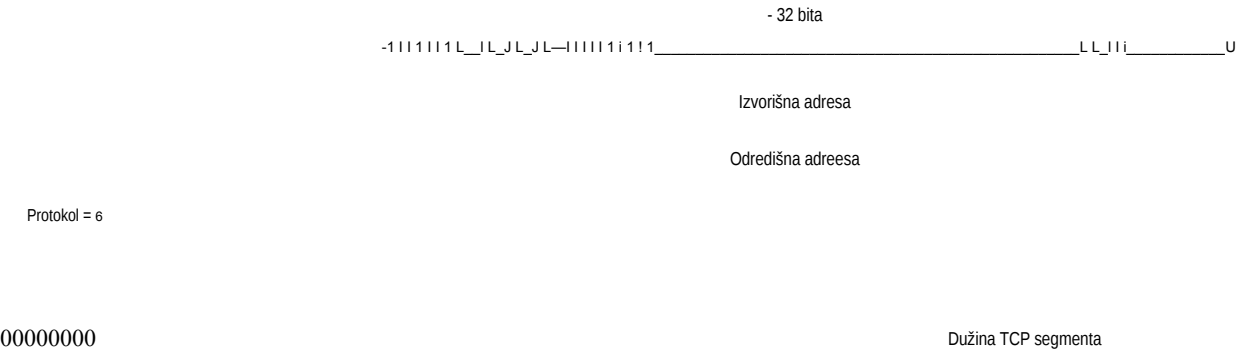
Bit *FIN* služi za zatvaranje veze. On označava da pošiljalac nema više podataka za slanje. Međutim, posle zatvaranja (jednog smera) veze, proces koji je raskida može i dalje da prima podatke. Segmenti *SYN* i *FIN* imaju svoje redne brojeve, što garantuje njihovu obradu ispravnim redosledom.

Protokol TCP upravlja tokom podataka pomoću kliznih prozora promenljive veličine. *Veličina prozora* saopštava broj bajtova koji se mogu poslati uključujući i potvrđen bajt. Nulta veličina prozora sasvim je legalna i govori da su primljeni svi

bajtovi zaključno s bajtom *Broj potvrde* - 1, ali da primalac trenutno više ne može da prima podatke. Primalac kasnije može da obnovi dozvolu za slanje tako što će poslati segment sa istim *Brojem potvrde* i *Veličinom prozora* različitom od nule.

U protokolima iz 3. poglavlja, potvrde za primljene okvire i dozvole za slanje novih okvira bile su međusobno vezane, što je posledica fiksne veličine prozora koja utiče na svaki protokol. U protokolu TCP, potvrde i dozvole za dodatno slanje potpuno su razdvojene. To znači da primalac može jednostavno da izjavi: Primio sam podatke do *k,* ali trenutno ih više ne želim. Pomenuto razdvajanje (u stvari, prozor promenljive veličine) daje protokolu dodatnu elastičnost. U nastavku ćemo je detaljnije razmotriti.

Kontrolni zbir je dodat zbog veće pouzdanosti prenosa. On obuhvata zaglavlje, podatke i konceptualno pseudozaglavlje, prikazano na slici 6-30. TCP polje *Kontrolni zbir* se pri izračunavanju postavlja na nulu, a polje s podacima se dopunjava bajtom nula ako je njegova dužina neparan broj. Algoritam za izračunavanje konrolnog zbira jednostavno sabira 16-bitne reči kao nepotpune komplemente i izračunava nepotpun komplement zbira. Zbog toga, kada primalac primeni isti postupak na čitav segment, uključujući i polje *Kontrolni zbir,* rezultat treba da bude 0.



Slika 6-30. Pseudozaglavlje koje se uključuje u kontrolni TCP zbir.

Pseudozaglavlje sadrži 32-bitne IP adrese izvorišnog i odredišnog računala, broj protokola koji odgovara protokolu TCP (6) i broj bajtova TCP segmenta (uključujući i zaglavlje). Uključivanjem pseudozaglavlja u izračunavanje kontrolnog zbira mogu se otkriti pogrešno isporučeni paketi, ali se time i narušava hijerarhija protokola pošto IP adrese u njemu pripadaju IP sloju, a ne TCP sloju. Protokol UDP koristi isto pseudozaglavlje za svoj kontrolni zbir.

Opcije omogućavaju uključivanje dodatnih mogućnosti koje ne predviđa redovno zaglavlje. Najvažnija je ona kojom svaki računar može da zada maksimalan korisni TCP teret koji je spreman da primi. Efikasnije je koristiti veće segmente jer se povećava količina prenetih podataka po jednom 20-bajtnom zaglavlju, ali skromniji računali ne mogu uvek da obrade veće segmente. Tokom uspostavljanja veze, svaka strana može da saopšti svoj

maksimum i da sazna maksimum druge strane. Ako računar ne koristi ovu opciju, dodeljuje mu se podrazumevani koristan teret od 536 bajtova. Od svih računara na Internetu zahteva se da prihvataju TCP segmente dužine 536 + 20 = 556 bajtova. Maksimalna veličina segmenta u dva smera iste veze ne mora da bude ista.

Za linije velikog propusnog opsega, linije s velikim kašnjenjem ili za linije sa obe ove osobine, često predstavlja problem prozor veličine 64 KB. Na T3 liniji (44,7.36 Mb/s), prozor veličine 64 KB isprazni se za samo 12 ms. Ako je vreme obilaska veze 50 ms (tipično za transkontinentalni optički kabl), pošiljalac će 3/4 vremena samo čekati potvrde. U satelitskoj vezi situacija je još gora. Veći prozor bi pošiljaocu omogućio da neprestano „pumpa" podadte, ali se u 16-bitnom polju takva veličina ne može izraziti. U RFC dokumentu 1323 predložena je opcija *Uvećanje prozora* koja pošiljaocu i primaocu omogućava da pregovaraju oko faktora uvećanja prozora. Taj broj obema stranama dozvoljava da polje *Veličina prozora* pomere ulevo za 1 do 14 bitova i time omoguće prozore veličine do 2^{30} bajtova. Savremene realizacije protokola TCP većinom podržavaju ovu opciju.

Druga opcija, koja je predložena u RFC dokumentu 1106 i danas široko ugrađena, odnosi se na primenu protokola selektivnog ponavljanja umesto protokola „vrati se n". Ako primalac dobije jedan neispravan segment iza koga sledi veliki broj ispravnih segmenata, „normalni" TCP protokol („vrati se n"), nakon isteka roka tajmera, ponovo bi poslao sve nepotvrđene segmente, uključujući i one koji su već stigli ispravno. RFC dokumentom 1106 uvedene su negativne potvrde (NAK), kako bi primalac mogao da zahteva samo određeni segment ili segmente. Kada ga (ih) dobije, on može da potvrdi prijem svih privremeno uskladištenih podataka i da tako smanji količinu podataka koje treba ponovo slati,

6.5.5 Uspostavljanje TCP veze

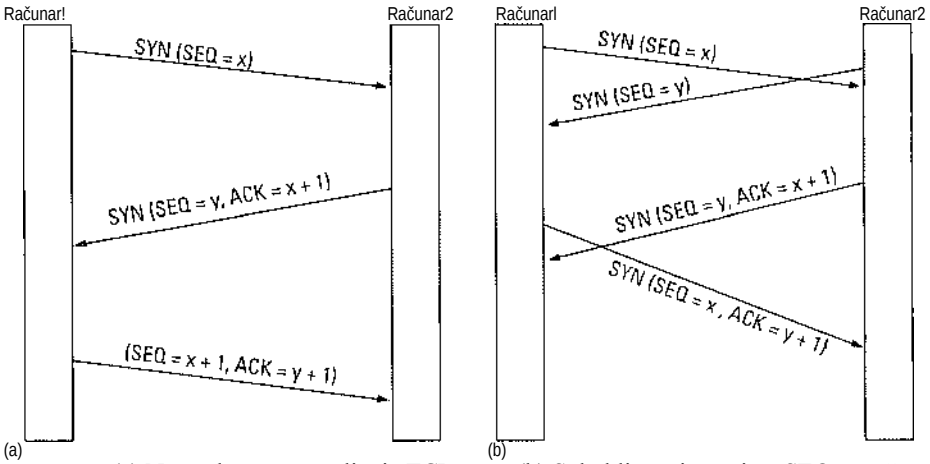
Veze u protokolu TCP uspostavljaju se mehanizmom trostepenog usuglašavanja, o kome smo govorili u odeljlcu 6.2.2. Da bijedna strana (npr. server) uspostavila vezu, ona pasivno čeka zahteve za uspostavljanje veze tako što izvrši operacije LISTEN i ACCEPT, pri čemu može, ali i ne mora da naznači izvorište.

Druga strana (npr. klijent) izvršava operaciju CONNECT, navodeći IP adresu i priključak na koji želi da se poveže, maksimalnu veličinu TCP segmenta koji je spremna da prihvati i, neobavezno, određene korisničke podatke (npr. lozinku). Operacija CONNECT šalje TCP segment s bitom *SYN* postavljenim na 1 i bitom *ACK* postavljenim na 0, i čeka odgovor.

Kada ovaj segment stigne na odredište, tamošnja TCP jedinica proverava da li postoji proces koji je izvršio operaciju LISTEN na priključku naznačenom u polju *Odredišni priključak.* Ako ne nađe takav proces, ona šalje odgovor s bitom *RST* postavljenim na 1 da bi odbila uspostavljanje veze.

Ako neki proces osluškuje priključak, predaje mu se pristigli TCP segment. On tada može da prihvati ili da odbije vezu. Ako je prihvati, to i potvrđuje. Redosled TCP segmenata koji se šalju u normalnom slučaju prikazan je na slici 6-31 (a). Obratite pažnju na to da segment SYN troši 1 bajt (jedno mesto) u nizu rednih brojeva, pa se može nedvosmisleno potvrditi.

Na slici 6-3 l(b) prikazan je redosled događaja za slučaj kada dva računara istovremeno pokušavaju da uspostave vezu između dve iste utičnice. Na kraju se uspostavlja samo jedna veza jer se svaka veza jednoznačno identifikuje svojim krajnjim tačkama. Ako prvi pokušaj rezultuje u vezi identifikovanoj krajnjim tačkama *(x, y),* a drugi da isti rezultat, u tabeli se pravi samo jedna odrednica za vezu (x, *y).*



Slika 6-31. (a) Normalno uspostavljanje TCP veze. (b) Sukobljavanje poziva. SEQ označava redni broj segmenta.

Početni redni broj segmenta na vezi nije 0 iz razloga koje smo ranije objasnili. Radni takt protokolu daje sistemski sat koji otkucava svake 4 ps. Kada računar otkaže, zbog dodatne sigurnosti mu se ne dozvoljava da se ponovo uključi pre nego što istekne maksimalni životni vek paketa poteklih iz prethodnih veza.

6.5.6 Raskidanje TCP veze

Iako su TCP veze potpuno dupleksne, da biste razumeli kako se raskidaju, bolje je da ih posmatrate kao dve paralelne jednosmerne veze. Svaka jednosmema veza raskida se nezavisno od svog parnjaka. Da bi raskinula vezu, svaka strana može da pošalje TCP segment s bitom *FIN* postavljenim na 1, što znači da nema više podataka za slanje. Kada se segment *FIN* potvrdi, taj smer se zatvara za nove podatke. Međutim, podaci i dalje mogu teći drugim smerom. Veza se stvarno raskida kada se zatvore oba njena smera. Za raskidanje je normalno potebno razmeniti četiri TCP segmenta: po jedan *FIN* i *ACK* segment za svaki smer. Međutim, prvi *ACK* segment i dragi *FIN* segment mogu se lcombinovati, tako da ukupan broj razmenjenih segmenata spada na tri.

Kao što oba sagovornika spuštajući slušalicu istovremeno mogu da raskinu telefonsku vezu, tako i obe strane TCP veze mogu istovremeno da pošalju segment *FIN.* Ti segmenti se potvrđuju na uobičajen način i veza se raskida. Nema, u stvari, bitne razlike između situacija u kojima računari vezu raskidaju jedan za dragim ili istovremeno.

Da bi se izbegao problem dve vojske, koriste se tajmeri. Ako odgovor na segment *FIN* ne stigne unutar dva maksimalna životna velca paketa, pošiljalac segmenta *FIN* raskida vezu. Druga strana će najzad primetiti daje više nilco ne sluša, pa c'e se i ona automatski isključiti. Iako opisano rešenje nije savršeno, ono ipak zadovoljava kada se ima u vidu da se savršeno rešenje teoretski i ne može postići. U praksi se problemi na ovom polju retko sreću.

6.5.7 Modelovanje rada **sa** TCP vezom

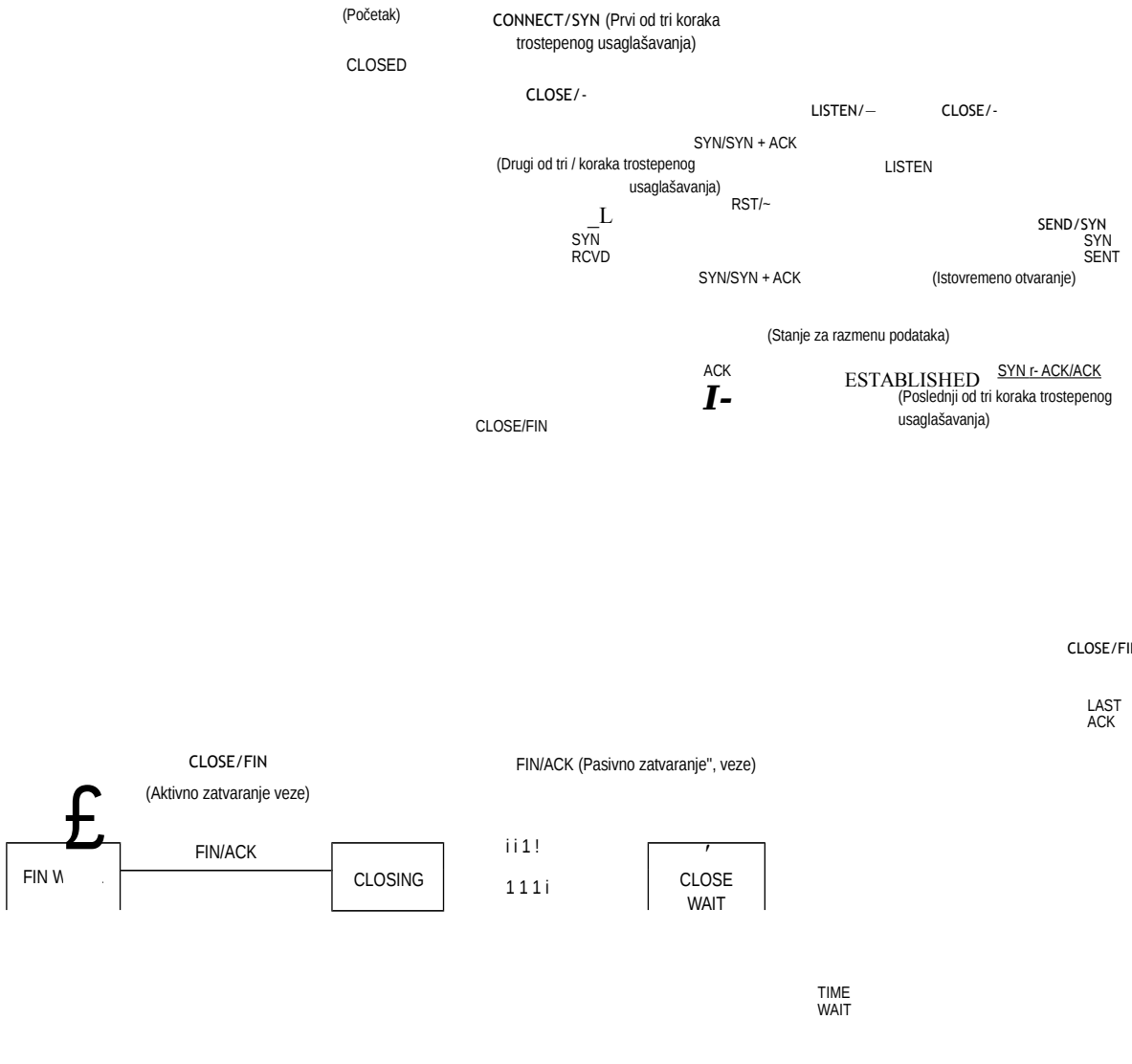
Postupale uspostavljanja i raskidanja veze može se prikazati pomoću mašine konačnih stanja, čijih je 11 stanja prikazano na slici 6-32. U svakom pojedinačnom stanju izvesni događaji su legalni. Kada se desi legalan događaj, može se preduzeti neka akcija. Ako je događaj nelegalan, generiše se izveštaj o grešci.

Svaka veza počinje od stanja *CLOSED* (ne postoji). Ona to stanje napušta pasivnim (LISTEN) ili aktivnim (CONNECT) otvaranjem. Ako draga strana izvrši suprotnu od dve navedene usluge,veza prelazi u stanje *ESTABLISHED* (uspostavljena veza). Raskidanje veze može da inicira bilo koja strana. Kada se dovrši, veza se ponovo vraća u stanje *CLOSED.*

Stanje	**Opis**
CLOSED	Nema aktivne veze, niti čekanja na vezu
LISTEN	Server čeka zahtev za uspostavljanje veze
SYN RCVD	Zahtev za uspostavljanje veze je stigao; čekanje potvrde
SYN SENT	Aplikacija je počela da otvara vezu
ESTABLISHED	Normalno stanje za razmenu podataka
FIN WAIT 1	Aplikacija saopštava da je završila
FIN WAIT 2	Druga strana se slaže s raskidanjem veze
TIMED WAIT	Čekanje da svi paketi izumru
CLOSING	Obe strane su pokušale da istovremeno zatvore vezu
CLOSE WAIT	Druga strana je inicirala raskidanje veze
LASTACK	Čekanje da svi paketi izumru

Slika 6-32. Stanja upotrebljena u mašini konačnih stanja kojom je predstavljen rad sa TCP vezom.

Sama mašina konačnih stanja prikazana je na slici 6-33. Uobičajena situacija kada se klijent aktivno povezuje s pasivnim serverom prikazana je zadebljanim linijama - punim za klijenta, isprekidanim za server. Tanke linije označavaju sekvence neuobičajenih događaja. Svaka linija na slici 6-33 označena je parom *događaj/akcija.* Događaj može da bude sistemski poziv koji je inicirao korisnik (CONNECT, LISTEN, SEND ili CLOSE), stizanje segmenta *(SYN, FIN,ACK* ili *RST)* ili, u jednom slučaju, isključivanje tajmera posle dvostrukog maksimalnog životnog veka paketa. Akcija je slanje upravljačkog segmenta *(SYN, FIN* ili *RST)* ili ništa, što je označeno crtom (-). Napomene su navedene u zagradi.


(Početak)
CONNECT/SYN (Prvi od tri koraka trostepenog usaglašavanja)
CLOSED
CLOSE/-
LISTEN/—
CLOSE/-
SYN/SYN + ACK
(Drugi od tri / koraka trostepenog usaglašavanja)
LISTEN
RST/~
SEND/SYN
SYN RCVD
SYN SENT
SYN/SYN + ACK
(Istovremeno otvaranje)
(Stanje za razmenu podataka)
ACK
ESTABLISHED
SYN r- ACK/ACK
(Poslednji od tri koraka trostepenog usaglašavanja)
CLOSE/FIN
CLOSE/FIN
LAST ACK
CLOSE/FIN
(Aktivno zatvaranje veze)
FIN/ACK (Pasivno zatvaranje", veze)
FIN W
FIN/ACK
CLOSING
CLOSE WAIT
TIME WAIT
ACK /-
ACK
FIN + ACK/ACK
FIN WAIT 2
FIN/ACK
sticanje roka tajmera/) ACK/-
CLOSED

(Vrati se na početak)

Slika 6-33. Mašina konačnih stanja za rad sa TCP vezom. Zadebljana puna linija označava normalan redosled događaja kod klijenta, Zadebljana isprekidana linija označava normalan redosled događaja na serveru. Tanke linije predstavljaju neuobičajene događaje. Uz svaki prelazak navedeni su događaj koji ga pokreće i rezultujuća akcija, razvdojeni kosom crtom.

Dijagram ćete najlakše razumeti ako najpre pratite put klijenta (zadebljana puna linija), a zatim servera (zadebljana isprekidana linija). Kada aplikacija na klijentskom računani pošalje zahtev CONNECT, lokalna TCP jedinica pravi zapis o vezi, u njega beleži da je veza u stanju *SYN SENT* i šalje segment *SYN*. Imajte na umu da istovremeno može biti otvoreno više veza (ili se istovremeno mogu otvarati različite aplikacije), pa se stanje beleži za svaku vezu u odgovarajućem zapisu. Kada stigne segment *SYN+ACK,* TCP jedinica šalje poslednji segment *ACK* iz šeme trostepenog usaglašavanja i prelazi u stanje *ESTABLISHED*. Sada se mogu slati i primati podaci.

Kada aplikacija završi s radom, ona izvršava osnovnu operaciju CLOSE, pa lokalna TCP jedinica šalje segment *FIN* i čeka odgovarajući segment *ACK* (pravougaonilc iscrtan isprekidanom linijom i označen kao „aktivno zatvaranje veze“)- Kada stigne segment *ACK,* veza prelazi u stanje *FIN WAIT* 2 - jedan smer veze je sada zatvoren. Kada i druga strana zatvori svoj smer, stiže segment *FIN* koji se potvrđuje. Sada su oba smera veze zatvorena, ali TCP jedinica čeka tokom vremena jednakog maksimalnom životnom veku paketa za slučaj da se segment ACK izgubio u putu i tek kada se tajmer isključi, briše zapis o vezi - veza je tada definitivno raskinuta, a na mreži više nema zaostalih paketa koji potiču od nje.

Razmotrimo sada uspostavljanje, rad i raskidanje veze s gledišta servera. Server izvršava osnovnu operaciju LISTEN i mimo čeka da se neko javi. Kada stigne segment *SYN,* on ga potvrđuje i prelazi u stanje *SYN RCVD.* Kada bude potvrđen i segment *SYN* koji je poslao server, dovršava se trostepeno usaglašavanje i server prelazi u stanje *ESTABLISHED.* Sada se mogu razmenjivati podaci.

Kada klijent završi posao, izvršava operaciju CLOSE, pa serveru stiže segment *FIN* (pravougaonilc iscrtan isprekidanom linijom i označen kao „pasivno zatvaranje veze“). Time je serveru poslat signal. Kada i on izvrši operaciju CLOSE, klijentu se šalje segment *FIN.* Po stizanju potvrde od klijenta, server raskida vezu i briše zapis o njoj.

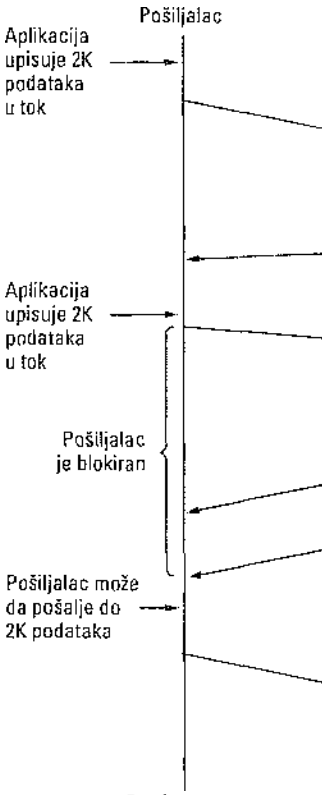
6.5.8 Pravila TCP prenosa

Kao što smo već pomenuli, rad s prozorima u protokolu TCP nije direktno vezan s potvrđivanjem, što je uglavnom slučaj u protokolima sloja veze podataka. Pretpostavimo, na primer, da primalac ima bafer veličine 4096 bajtova, kao na slici 6-34. Ako mu pošiljalac uputi segment veličine 2048 bajtova i ovaj bude ispravno primljen, primalac će ga potvrditi. Međutim, pošto u baferu sada ima slobodnih samo 2048 bajtova (dok aplikacija ne preuzme nešto podataka), on će objaviti prozor veličine 2048 bajtova koji počinje sledećim očekivanim bajtom.

Sada pošiljalac upućuje dragih 2048 bajtova, koji se potvrđuju, ali se oglašava prozor veličine 0. Slanje se mora prekinuti dok aplikacija na računam primaoca ne preuzme nešto podataka iz bafera i TCP jedinica ne oglasi veći prozor.

Kada prozor ima nultu veličinu, pošiljalac normalno ne bi smeo da šalje segmente, uz dva izuzetka. Prvo, mogu se poslati hitni podaci da bi se, na primer, prekinuo proces koji se izvršava na udaljenom računani. Drugo, pošiljalac može da uputi jedno- bajtni segment kojim će primaoca prinuditi da ponovo oglasi sledeći očekivani bajt i veličinu prozora. TCP stanard izričito podržava ovu opcije da bi se sprečilo kružno blokiranje ako se segment sa oglašenom veličinom prozora izgubi.

Pošiljalac ne mora da šalje podatke čim ih dobije od aplikacije, niti primalac mora odmah da šalje potvrdu. Na primer, na slici 6-34, kada joj stigne prvih 2 KB podataka od aplikacije, TCP jedinica, znajući da ima na raspolaganju prozor veličine 4096 KB, sasvim bi ispravno postupila ako bi ih smestila u bafer dok ne stigne još 2 KB podataka, kako bi mogla da pošalje segment s korisnim teretom od 4 KB. Takva mogućnost odlučivanja ugrađena je u cilju poboljšanja performansi.



Slika 6-34. Rad s prozorom u protokolu TCP. SEQ označava redni broj segmenta, a WIN veličinu prozora.

Razmotrite telnet vezu ka programu za uređivanje teksta koji reaguje na svaki pritisak tastera. U najnepovoljnijem slučaju, kada znak stigne u TCP jedinicu pošiljaoca, ona će napraviti 21-bajtni TCP segment i predati ga protokolu IP da od njega napravi 41-bajtni IP datagram. TCP jedinica primaoca odmah šalje 40-bajtnu potvrdu (po 20 bajtova TCP i IP zaglavlja). Kasnije, kada program za uređivanje teksta učita bajt, TCP jedinica ažurira prozor pomerajuđi ga 1 bajt udesno i oglašava ga paketom koji je takođe veličine 40 bajtova. Na kraju, kada program za uređivanje teksta obradi znak, on ga vraća kao odjek u paketu veličine 41 bajt. Sve u svemu, za svaki uneti znak potroše se 162 bajta propusnog opsega i četiri segmenta. Ako je propusni opseg skroman, ovo baš i nije najbolji način za obavljanje poslova.

Pristup koji se u mnogim TCP realizacijama koristi za optimizovanje ove situacije svodi se na odlaganje slanja potvrde i oglasa s novom veličinom prozora za .500 ms, u nadi da će se pojaviti neki podaci uz koje se oni mogu šlepovati. Uz pretpostavku da program za uređivanje teksta šalje odjek unutar vremena od .500 ms, udaljenom korisniku treba natrag poslati samo jedan paket veličine 41 bajt, čime se broj paketa i zauzeće propusnog opsega polove.

Iako se ovim pravilom smanjuje opterećenje mreže koje izaziva primalac, pošiljalac i

dalje radi neefikasno, šaljući 41-bajtne pakete sa samo jednim bajtom podataka. Takav neefikasan rad može se ublažiti jednostavnim **Nagleovim algoritmom** (Nagle, 1984): kada podaci pošiljaocu pristižu bajt po bajt, treba poslati samo prvi bajt, a ostale privremeno uskladištiti sve dok ne stigne potvrda za njega. Zatim treba poslati sve uskladištene znakove u jedinstvenom TCP segmentu i ponovo početi sa smeštanjem podataka u bafer dok svi poslati bajtovi ne budu potvrđeni. Ako korisnik brzo unosi podatke, a mreža je spora, jednim segmentom može se poslati priličan broj znakova, pa se postiže primetna ušteda propusnog opsega. Osim toga, algoritam dozvoljava i slanje novog paketa ako pristigne dovoljno podataka da ispune polovinu prozora ili segment maksimalne veličine.

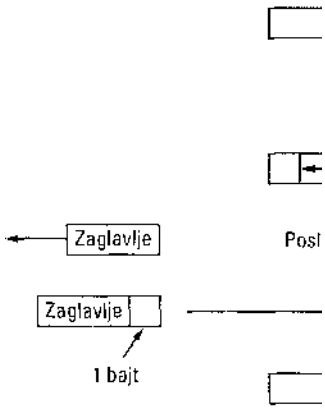
Nagleov algoritam se široko koristi u TCP realizacijama, ali ima situacija u kojima je bolje da se on deaktivira. Konkretno, ako se preko Interneta izvršava neka aplikacija grafičkog sistema X Window, udaljenom računaru treba slati pokrete miša. (X Window je grafički sistem prozora na vec'ini UNIX računara.) Kada bismo ih privremeno skladištili i slali u rafalima, pokazivač miša bi se po ekranu kretao na mahove, što bi potpuno zbunilo korisnika.

Još jedan problem koji može da pogorša TCP performanse zove se **sindrom luckastog prozora** (engl. *silly window syndrome*), prema Klarku (Clark, 1982). Problem nastaje kada se transportnoj TCP jedinici pošiljaoca podaci šalju u velikim blokovima, ali ih interaktivna aplikacija kod primaoca učitava bajt po bajt. Najbolje čete razumeti o čemu se radi ako pogledate sliku 6-35. Na početku je bafer primaoca pun i pošiljalac to zna (jer ima prozor veličine 0). Zatim interaktivna aplikacija učitava jedan bajt iz TCP toka. TCP jedinica primaoca odmah širi prozor i obaveštava pošiljaoca da može da pošalje 1 bajt. Pošiljalac prihvata predlog i šalje 1 bajt. Bafer je sada pun, tako da primalac potvrđuje stizanje jednobajtnog segmenta i postavlja veličinu prozora na nulu. Takav sled događaja može trajati večno.

Rešenje koje je za ovu situaciju predložio Klark svodi se na to da primalac ne treba da oglašava jednobajtni prozor, već da sačeka i oglasi prozor tek kada bude mogao da primi neku pristojniju količinu podataka. Dragim recima, primalac ne treba da oglašava nov prozor sve dok ne bude mogao da obradi segment maksimalne veličine koji je dogovorio prilikom uspostavljanja veze ili dok ne bude imao poluprazan bafer - šta god daje od to dvoje manje.

Osim toga, i pošiljalac može da doprinese efikasnosti tako što neće slati kratke segmente, već će se truditi da podacima dovoljno ispuni prozor kako bi mogao da pošalje segment maksimalne dužine ili takav koji će ispuniti barem polovinu bafera primaoca (tu veličinu on mora da proceni na osnovu istorije ažuriranja prozora).

Nagleov algoritam i Klarkovo rešenje za sindrom luckastog prozora međusobno se dopunjavaju. Nagle je pokušao da reši problem izazvan aplikacijom koja TCP jedinici šalje podatke bajt po bajt. Klark je pokušao da reši problem aplikacije koja iz TCP toka izvlači podatke, takođe, bajt po bajt. Oba rešenja su ispravna i mogu međusobno da sa- rađuju. Postignut je cilj da pošiljalac ne šalje kratke segmente, a da ih primalac ne traži.



Slika 6-35. Sindrom luckastog prozora.

Osim što će menjati prozor u većim priraštajima, TCP jedinica primaoca može i dodatno da poboljša performanse. Slično TCP jedinici pošiljaoca, i ona može da privremeno skladišti podatke i da blokira zahtev READ aplikacije sve dok ne bude imala dovoljno podataka da joj isporuči. Time se smanjuje broj poziva transportnoj TCP jedinici (i manje ometa njen rad). Naravno, time se produžava i vreme odgovaranja, ali za neinteraktivne aplikacije kao što je prenos datoteka, efikasnost je mnogo važnija nego vreme odgovaranja na pojedinačne zahteve.

Primalac takođe treba da zna šta da radi sa segmentima koji stignu preko reda. On ih može čuvati ili odbaciti, prema svom nahođenju. Naravno, potvrde se mogu slati tek kada stignu svi podaci, zaključno s potvrđenim bajtom. Ako primaocu stignu segmenti 0, 1,2, 4, 5, 6 i 7, on može da potvrdi sve do poslednjeg bajta u segmentu 2. Kada se tajmer pošiljaoca automatski isključi, on će tada ponovo poslati segment 3. Ako je primalac sačuvao segmente 4 do 7, on će po prijemu segmenta 3 moći da potvrdi sve do poslednjeg bajta segmenta 7.

6.5.9 TCP kontrola zagušenja

Kada se mreža optereti više nego što može da podnese, dolazi do zagušenja. Tu nije izuzetak ni Internet. U ovom odeljku razmotrićemo algoritme koji su poslednjih četvrt veka razvijani za borbu sa zagušenjem. Iako i mrežni sloj pokušava da upravlja zagušenjem, glavni posao ostaje protokolu TCP jer se zagušenje može razrešiti jedino smanjenjem brzine prenosa podataka.

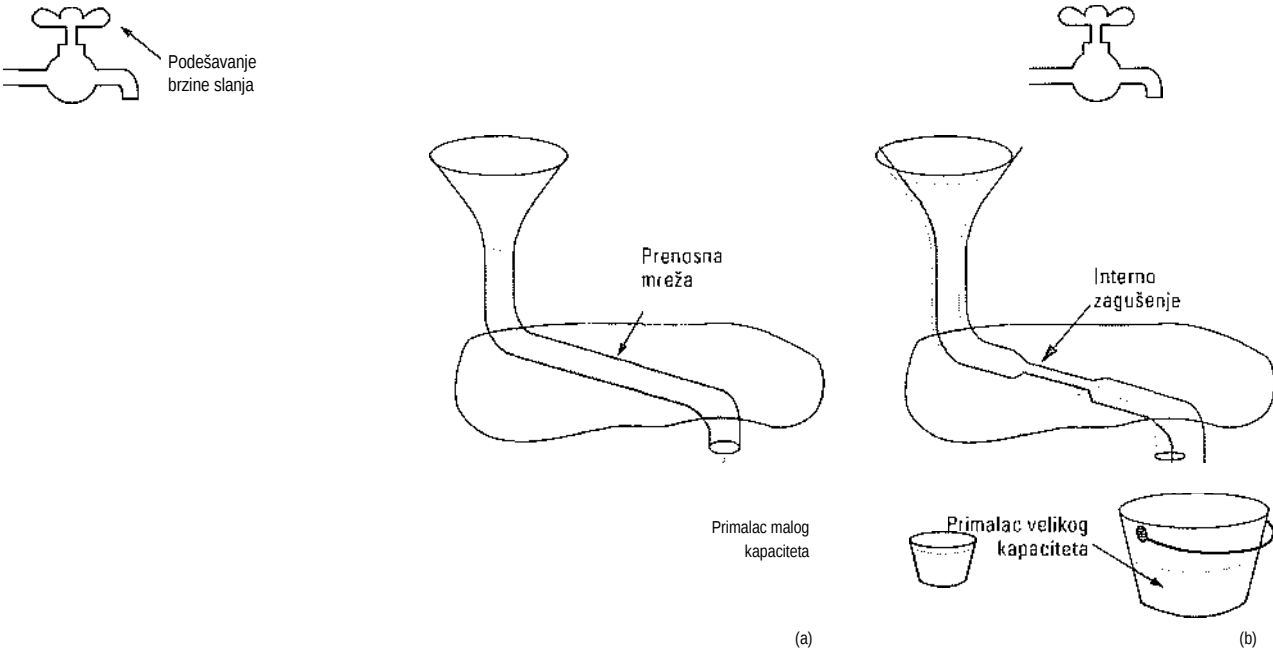
Teorijski, zagušenje se može izbeći ako se poslužimo analogijom zakona koji vladaju u fizici i držimo se „principa održanja broja paketa“. To znači da u mrežu ne treba slati nov paket dok je jedan paket koji je već na njoj ne napusti (tj. bude isporučen). To se u TCP protokolu pokušava postići dinamičkim podešavanjem veličine prozora.

Da bi se zagušenjem moglo upravljati, najpre ga treba otkriti. Ranije je to bilo skopčano s teškoćama. Tajmer pošiljaoca mogao se automatski isključiti (1) zbog velikog šuma na prenosnoj liniji ili (2) zato što je paket stvarno odbačen na nekom zagušenom usmerivaču. Pošiljalac nije imao načina da utvrdi razliku.

Danas se paketi srazmerno retko gube zbog grešaka u prenosu jer se na većini dugih linija koristi optički kabl (bežične mreže su ipak posebna priča). Zbog toga se na Internetu tajmeri isključuju uglavnom zbog zagušenja. Svi TCP algoritmi na Internetu pretpostavljaju da se tajmeri isključuju zbog zagušenja i budno prate njihov rad, baš kao što radari, zaziruć od metana, ne ispuštaju iz vida svoje kanarince.

Pre nego što objasnimo kako protokol TCP reaguje na zagušenje, razmotrimo šta on preduzima da do zagušenja ni ne dođe. Kada se uspostavi veza, treba izabrati podesnu veličinu prozora. Primalac može da predloži prozor na osnovu veličine svog bafera. Ako pošiljalac taj predlog prihvati, eventualni budući problemi neće nastati zbog prelivanja bafera kod primaoca, već verovatno zbog zagušenja u mreži.

Problem smo na slici 6-36 prikazali njegovom hidrauličkom analogijom. Na slici 6-36(a), voda se iz široke cevi (mreže) izliva u mali prihvatni sud primaoca. Sve dok pošiljalac ne šalje više vode nego što kofa može da prihvati, voda se neće prelivati. Na slici 6-36(b), kofa ima veliki kapacitet, ali je protok (podataka) ograničen suženjem u cevi (kapacitetom mreže). Ako voda pristiže prebrzo, prelivaće se preko ivice levka.



Slika 6-36. (a) Brza mreža napaja primaoca malog kapaciteta, (b) Spora mreža napaja primaoca velikog kapaciteta.

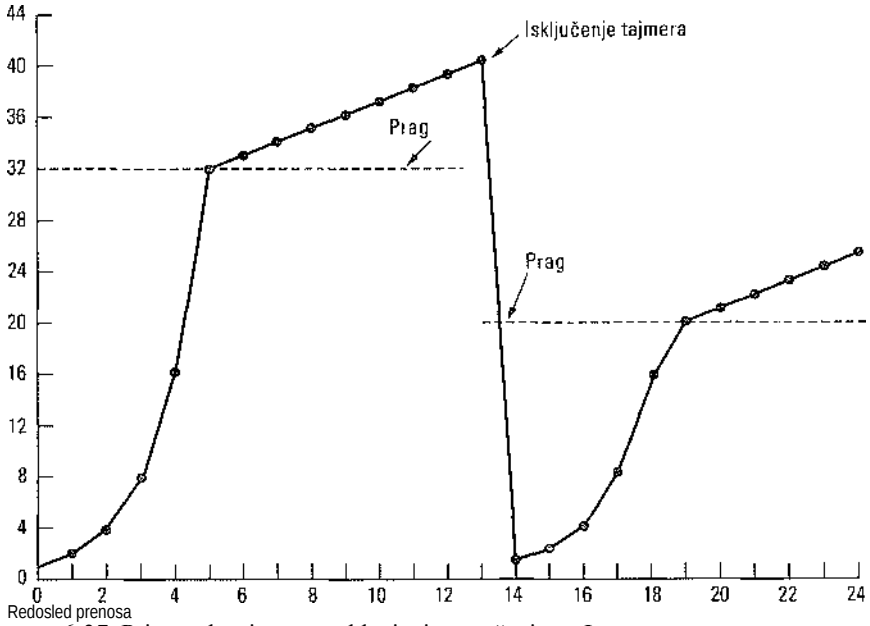
Rešenje za Internet može se nadi ako se shvati da postoje dva različita problema: kapacitet mreže i kapacitet primaoca, i svaki od njih zasebno rešava. U tom cilju, svaki pošiljalac treba da održava dva prozora: prozor koji je odobrio primalac i tzv. **prozor zagušenja** (engl. *congestion window).* Svaki prozor treba da odražava broj bajtova koje pošiljalac može da pošalje. Broj bajtova koje pošiljalac stvarno šalje odgovara manjem od dva prozora. Dakle, stvarno *će* biti poslat broj bajtova oko koga se slože i pošiljalac i primalac. Ako primalac poruči: „Pošalji 8 KB“, a pošiljalac zna da *će* više od 4 KB zagušiti mrežu, on će poslati 4 KB. S druge strane, ako primalac poruči: „Pošalji 8 KB“, a pošiljalac zna da rafali veličine i do 32 KB prolaze glatko, on će poslati svih zahtevanih 8 KB.

Kada se uspostavi veza, pošiljalac inicijalizuje prozor zagušenja na veličinu maksimalnog segmenta koji se koristi na vezi. Zatim šalje jedan segment maksimalne veličine. Ako za njega dobije potvrdu pre nego što se isključi tajmer, on širi prozor zagušenja za još jedan takav segment i šalje dva segmenta maksimalne veličine. Kako stižu potvrde za svaki od poslatih segmenata, on uvećava prozor zagušenja za po jedan maksimalan segment. Kada je veličina prozora zagušenja *n* segmenata i za svih *n* segmenata stignu potvrde, prozor zagušenja se proširuje za broj bajtova koji odgovara *n* segmenata. Dakle, posle svakog probnog rafala koji se završi uspešno, udvostručuje se veličina prozora zagušenja.

Prozor zagušenja raste eksponencijalno sve dok se prvi put ne isključi tajmer ili dok njegova veličina ne dostigne (početnu) veličinu prozora primaoca. Ako, na primer, prolaze rafali veličine 1024, 2048 i 4096 bajtova, ali se kod veličine 8192 bajta isključi tajmer, veličinu prozora treba podesiti na 4096 da bi se izbeglo zagušenje. Sve dok je veličina prozora zagušenja 4096 bajtova, neće se slati duži rafali, bez obzira na veličinu prozora koju odobrava primalac. To je tzv. **spori algoritam** (engl. *slow start),* ali on uopšte ne radi sporo (Jackobson, 1988). Kad prilike dozvole, on svoj rad ubrzava eksponencijalno. Obavezan je za sve TCP realizacije.

Razmotrimo sada algoritam za kontrolu zagušenja na Internetu. Osim prozora primaoca i prozora zagušenja, on sadrži i treći parametar - **prag** (engl. *threshold),* koji na početku iznosi 64 KB. Kada se tajmer automatski isključi, prag se smanjuje na polovinu tekuće veličine prozora zagušenja, a prozor zagušenja podešava na jedan segment maksimalne dužine. Posle toga se pomoću sporog algoritma utvrđuje maksimalna propusna moć mreže, osim što se eksponencijalno povećanje prozora prekida kada se pređe vrednost praga. Od tog trenutka, prozor zagušenja se povećava linearno (za jedan segment maksimalne dužine po uspešno prenetom rafalu), umesto za po jedan segment po uspešno prenetom segmentu. Ovaj algoritam, u stvari, nagađa da će za otklanjanje zagušenja verovatno biti dovoljno da prepolovi veličinu prozora, a zatim ga postepeno opet povećava.

Rad algoritma za otklanjanje zagušenja prikazan je na slici 6-37, gde maksimalna veličina segmenta iznosi 1024 bajta. Veličina prozora zagušenja na početku je bila 64 KB, ali se tajmer isključio, pa je prag podešen na 32 KB, a prozor zagušenja sveden na 1 KB (nulti prenos). Prozor zagušenja zatim raste eksponencijalno dok ne dostigne prag (32 KB), a odatle linearno.



Slika 6-37. Primer algoritma za otklanjanje zagušenja na Internetu.

U trinaestom prenosu sreća okreće leđa (to se moglo unapred znati) i tajmer se isključuje. Prag se podešava na polovinu veličine tekućeg prozora (20 KB, što je polovina od 40 KB) i ponovo se inicijalizuje spori algoritam. Kada potvrde četrnaestog prenosa počnu da pristižu, svaka od prve četiri potvrde udvostručava veličinu prozora zagušenja, ali posle toga, prozor nastavlja da raste linearno.

Ako se tajmer više ne bude isključivao, prozor zagušenja će nastaviti da raste dok ne dostigne veličinu prozora primaoca. Tu veličinu će zadržati sve dok se tajmer ponovo ne isključi ili primalac promeni veličinu svog prozora. Pomenimo uzgred da se stizanje prigušnog ICMP paketa SOURCE QUENCH u transportnu TCP jedinicu tretira kao i isključivanje tajmera. Alternativan (i savremeniji) pristup opisan je u RFC dokumentu 3168.

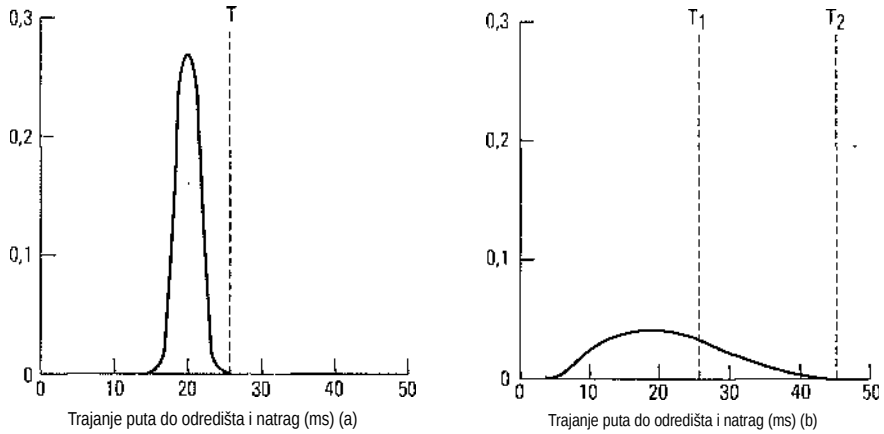
6.5.10 **Upravljanje** tajmerima **u protokolu TCP**

Za obavljanje svog posla, TCP jedinice koriste više tajmera (barem na papiru). Najvažniji među njima je tajmer za ponovno slanje (engl. *retransmission timer).* Taj tajmer se aktivira kada se pošalje segment. Ako potvrda stigne pre nego što se tajmer automatski isključi, tajmer se zaustavlja. Ukoliko se, s dinge strane, tajmer automatski isključi pre nego što stigne potvrda, segment se šalje ponovo (i tajmer ponovo uključuje). Postavlja se pitanje roka koji treba dodeliti tajmeru.

Problem je mnogo složeniji za rešavanje u transportnom sloju Interneta, nego u opštim protokolima sloja veze podataka iz 3. poglavlja. U ovom dragom slučaju, kašnjenje se može predvideti s visokim stepenom verovatnoće (tj. malo varira), tako

da se vreme isključivanja tajmera može podesiti na vrednost neznatno veću od očekivanog vremena stizanja potvrde, kao na slici 6-38(a). Pošto potvrde retko kasne u sloju veze podataka 0er tamo nema zagušenja), ako ne stignu do očekivanog vremena, to najčešće znači da su se okviri ili njihove potvrde izgubili u putu.

Protokol TCP radi u sasvim drugačijem okruženju. Raspodelaverovatnoće stizanja TCP potvrde više odgovara krivoj na slici 6-38(b), nego krivoj na slici 6-38(a). Trajanje puta do odredišta i natrag pomalo je neuhvatljivo. Čak i kada je to vreme poznato, teško je odrediti rok automatskog isključivanja tajmera. Ako se rok postavi na previše malu vrednost, npr. Tj na slici 6-38(b), okvir će se ponovo slati, što doprinosi zagušenju Interneta nepotrebnim paketima. Ukoliko je rok predugačak, npr. T_2, pogoršaće se performanse zbog dugog čekanja na ponovno slanje kad god se izgubi paket. Osim toga, srednja vrednost i varijansa raspodele vremena stizanja potvrde mogu se promeniti unutar nekoliko sekundi, u trenutku nastajanja ili razrešavanja zagušenja.



Slika 6-38. (a) Raspodela verovatnoće vremena pristizanja potvrda u sloju veze podataka, (b) Raspodela verovatnoće vremena pristizanja potvrda u TCP sloju,

Rešenje je u upotrebi algoritma koji će stalno dinamički podešavati rok tajmera, neprestano prateći performanse mreže. Algoritam koji to radi u TCP sloju dugujemo Jakobsonu (Jackobson, 1988). TCP jedinica za svaku vezu održava promenljivu *RTF,* čija vrednost odražava najbolju tekuću procenu vremena obilaska veze. Kada se pošalje segment, uključuje se tajmer da bi se utvrdilo najnovije vreme pristizanja potvrde i istovremeno ponovo poslao segment, ako se rok tajmera prekorači. Ako potvrda stigne pre nego što se tajmer isključi, dobija se vreme njenog stizanja *M.* TCP jedinica tu vrednost dodeljuje promenljivoj *RIT,* koristeći formulu

$$RTT = aRTT + (1 - a)M$$

gde je *a* „koeficijent izglađivanja“ kojim se prethodnoj vrednosti dodeljuje odgovarajući značaj. Najčešće, a = 7/8.

Čak i uz dobra vrednost promenljive *RTT,* određivanje pogodnog roka tajmera za ponovno slanje nije jednostavno. TCP jedinica će za rok obično upotrebiti vrednost J3R7T, ali treba odabrati p. U prvim realizacijama, koeficijent P je uvele bio 2, ali je iskustvo pokazalo da fiksna vrednost nije pogodna u situacijama kada raste varijansa.

Godine 1988, Jakobson je predložio da koeficijent P bude grubo proporcionalan standardnom odstupanju verovatnoce stizanja paketa, tako da velika varijansa znači i velileo P, i obrnuto. U stvari, on je predložio da se koristi *srednje odstupanje* (engl. *mean deviation*), kao jednostavna procena *standardnog odstupanja* (engl. *standard deviation).* Njegov algoritam treba da prati i drugu „izglađenu" promenljivu *D* - odstupanje. Kad god stigne potvrda, izračunava se razlika između očekivane i opaže- ne vrednosti *RTT-M*\. Dobijeni rezultat se u promenljivu *D* ugrađuje pomoću formule

$$D = aD + (1 - a) \backslash RTT\text{-}M \backslash$$

gde a može, ali ne mora, da ima istu vrednost kao što je ona korišćena za izglađivanje promenljive *RTT.* Iako *D* nije isto što i standardno odstupanje, ono predstavlja njegovu dovoljno dobra procenu, a Jakobson je pokazao kako se promenljiva *D* može izračunavati uz isključivu primenu aritmetike celih brojeva (sabiranje, oduzimanje, pomeranje bitova), što je velika prednost. Većina TCP realizacija sada koristi ovaj algoritam i pomoću sledeće formule podešava rok tajmera:

$$\text{Rok tajmera} = RTT + 4 \times D$$

Četvorka je pomalo proizvoljno odabrana kao množilac promenljive *D,* ali to nudi dve prednosti. Prvo, množenje četvorkom izvodi se jednom operacijom pomeranja bitiova. Drago, time se eliminišu nepotrebna isključivanja tajmera i ponovno slanje paketa jer manje od 1 % paketa kasni više od vremena jednakog četvoros trakom standardnom odstupanju. (Jakobson je, u stvari, na početku predložio vrednost 2, ali se kasnije pokazalo da vrednost 4 poboljšava performanse.)

Pri dinamičkom utvrđivanju vrednosti promenljive *RTT* nastaje problem kada istek- ne rok tajmera za određeni segment i on bude ponovo poslat. Kada stigne potvrda, ne zna se da li se ona odnosi na original ili na ponovo poslati segment. Pogrešna odluka može ozbiljno da poremeti procenu vrednosti *RTT.* Phil Karn je pomenuti problem ose- tio na sopstvenoj koži. On je pasionirani radio-amater koji voli da šalje TCP/IP pakete tzv. „ham radiom", poslovično nepouzdanim medijumom (po lepom danu probije se oko polovina paketa). On je izneo jednostavan predlog da se vrednost promenljive *RTT* ne ažurira ni za jedan segment koji se mora ponovo slati, već da se u svakom takvom slučaju rok tajmera udvostručava sve dok za segment prvi put ne stigne normalna potvrda. Ta popravka se zove Karnov algoritam. Koristi se u većini TCP realizacija.

Tajmer za ponovno slanje nije jedini tajmer koji koristi transportna TCP jedinica. Drugi je tzv. tajmer za ograničenje čekanja (engl. *persistence timer).* Njegov zadatak je da sprečava kružno blokiranje. Primalac šalje potvrdu s prozorom veličine 0, s.aopšta- vajući pošiljaocu da čeka. Primalac kasnije ažurira prozor, ali se paket kojim oglašava nov prozor gubi u putu. Sada i pošiljalac i primalac čekaju onog dragog da nešto učini. Kada se isključi tajmer za ograničenje čekanja, pošiljalac upućuje primaocu probni paket. Odgovor na njega sadrži veličinu prozora. Ako je veličina prozora i dalje **0,** ponovo se uključuje tajmer za ograničenje

čekanja i ciklus se ponavlja. Ako veličina prozora nije 0, mogu se slati podaci.

Treći tajmer koji se koristi u nekim realizacijama zove se **tajmer za proveru stanja veze** (engl. *keepalive timer)*. Kada se veza neko vreme potpuno utiša, ovaj tajmer može da se isključi i time opomene korisnike da pro vere da li je sagovornik još uvek na drugom kraju. Ako sagovornik ne odgovori, veza se raskida. Opisana mogućnost je sumnjive vrednosti jer opterećuje mrežu dodatnim saobraćajem, a može da raslcine inače aktivnu vezu koja se utišala zbog privremene blokade na mreži.

I na kraju, tajmer koji se koristi na svakoj TCP vezi koja se raskida održava stanje TIMED WAIT tokom dvostrukog maksimalnog životnog veka paketa, kako bi svi zaostali paketi „izumrli“ pre nego što se veza raskine.

6.5.11 Bežični TCP i UDP protokoli

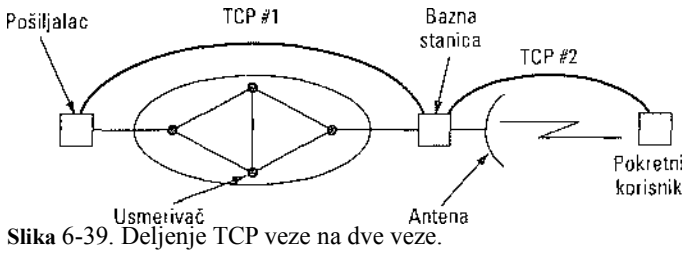
Transportni protokoli teorijski ne bi trebalo da zavise od tehnologije mrežnog sloja iznad koga se izvršavaju. Konkretno, protokol TCP ne bi trebalo da razmišlja o tome da li se protokol IP izvršava na optičkom kablu ili kroz radio-vezu. U praksi, međutim, to ima značaja jer su TCP realizacije uglavnom brižljivo optimizovane na osnovu pretpostavki koje važe za kablovske, ali ne i za bežične mreže. Zanemarivanje svojstava bežičnog prenosa može da dovede do TCP realizacije koja je logički ispravna, ali ima očajne performanse.

Glavni problem je algoritam za kontrolu zagušenja. U skoro svim TCP realizacijama danas se pretpostavlja da se tajmeri isključuju zbog zagušenja, a ne zato što se paketi gube. Zbog toga, kada se tajmer isključi, TCP jedinica uspori i pakete šalje blažim tempom (koristeći, npr. Jakobsonov spori algoritam). To treba da smanji opterećenje mreže i tako otkloni zagušenje.

Nažalost, bežične prenosne veze su veoma nepouzdane. Paketi se u njima sve vreme gube. Ispravan postupak sa izgubljenim paketima bio bi da se oni ponovo pošalju, i to što pre. Usporavanje slanja samo pogoršava stvari. Ako se gubi, recimo, **20%** svih paketa, tada, pri 100 paketa u sekundi koje upućuje pošiljalac, stvarni protok podataka iznosi **80** paketa u sekundi. Ako pošiljalac smanji brzinu slanja na **50** paketa u sekundi, i stvarni protok će se smanjiti na **40** paketa u sekundi.

U stvari, kada se paket izgubi na kablovskoj mreži, pošiljalac treba da uspori, a ako se izgubi na bežičnoj mreži, on treba da nastupi još žešće. Međutim, ako pošiljalac ne zna u kakvu mrežu šalje pakete, svaka odluka je rizična.

Putanja između pošiljaoca i primaoca često nije zasnovana na jedinstvenoj tehnologiji. Prvih **1000** km može da prolazi kroz kablovsku mrežu, ali na poslednjem kilometru kroz bežičnu. Tu je donošenje prave odluke još teže, jer ona zavisi od toga gde je nastao problem. Bakne i Badrinath **(1995)** predložili su rešenje, poznato kao **indirektni TCP protokol** (engl, *indirect TCP),* prema kome se TCP veza deli u dve zasebne veze, kao na slici **6-39.** Prva veza spaja pošiljaoca s baznom stanicom, a draga baznu stanicu s primaocem. Bazna stanica samo pakete za oba smera kopira s jedne veze na drugu.



Slika 6-39. Deljenje TCP veze na dve veze.

Prednost opisane šeme je to što je svaka veza sada homogena, tj. koristi jedinstvenu tehnologiju. Isključivanje tajmera na prvoj vezi sada izvesno znači da pošiljalac treba da uspori slanje, a isključivanje tajmera na dragoj upravo obrnuto - da ga ubrza. Za ove dve veze mogu se posebno podešavati i dragi parametri. Pomenuta šema, međutim, narušava semantiku protokola TCP. Pošto svaki deo veze predstavlja potpunu TCP vezu, bazna stanica obično potvrđuje svaki TCP segment. Ali sada, kada pošiljalac dobije potvrdu, to ne znači daje segment stigao primaocu, već daje stigao baznoj stanici.

Jedno drago rešenje, koje dugujemo Balakrishnanu i saradnicima (1995), ne narušava semantiku protokola TCP. Ono se svodi na više sitnih izmena u kodu mrežnog sloja bazne stanice. Jedna od izmena je i uvođenje „agenta za osmatranje“ (engl. *snooping agent)* koji otkriva i kešira TCP segmente koji odlaze pokretnom korisniku i potvrde koje dolaze od njega. Kada agent zapazi TCP segment koji odlazi pokretnom korisniku, ali za njega ne dobije potvrdu unutar (srazmerno kratkog) roka tajmera, on segment ponovo šalje i o tome ne izveštava izvorište. On ponovo šalje segment i kada od pokretnog korisnika dobije duplikat potvrde jer to neizostavno znači da je pokretni korisnik nešto propustio. Duplikati potvrda odbacuju se na licu mesta da izvorišni računar ne bi pogrešno zaključio daje došlo do zagušenja.

Mana ovakvog, neprimetnog rada bazne stanice dolazi do izražaja kada bežični deo veze gubi mnogo paketa. Tada se može isključiti tajmer na izvorištu, pa izvorišni računar pokreće algoritam za otklanjanje zagušenja. U indirektnom TCP protokolu, taj algoritam nikada ne bi bio pokrenut ako zaista nema zagušenja u kablovskom delu mreže.

Balakrishnan i saradnici su predložili i rešenje za slučaj kada se izgubi segment koji je poslao pokretni korisnik. Kada bazna stanica zapazi da u nizu nedostaje redni broj nekog poslatog segmenta, ona generiše zahtev za selektivno slanje nedostajućih bajtova koristeći odgovarajuću opciju TCP zaglavlja.

Bežična mreža uz opisane popravke postaje u oba smera pouzdanija, pri čemu izvorišni računar ne mora ništa da zna o tome, a i ne remeti se TCP semantika.

Iako protokol UDP radi drugačije od protokola TCP, bežično komuniciranje i njemu stvara teškoće. Osnovni problem je u tome što programi koji koriste protokol UDP očekuju da on radi veoma pouzdano. Oni, doduše, znaju da on ne nudi nikakve garancije, ali ipak očekuju da radi skoro savršeno. U bežičnom okruženju, rad protokola UDP daleko je od savršenstva. Za programe koji se od gubitka UDP poruka mogu oporaviti samo uz velike napore, nagli pelazak iz okruženja u kome se poroke, doduše, mogu izgubiti, ali se to retko dešava, u okruženje u kome se one redovno gube, može se pogubno odraziti na performanse.

Osim na performanse, bežično komuniciranje može da utiče i na drage aspekte pre- nosa. Postavlja se, na primer, pitanje kako pokretni računar može da pronađe lokalni štampač da ne

bi morao da koristi štampač iz svoje matične ćelije. Slično tome, kako on može da pristupi Web strani lokalne delije ako ne zna čak ni njeno ime? Pored toga, dizajneri Web strana skloni su da pretpostave postojanje velikog propusnog opsega. Kada na svaku Web stranu postave veliki logotip kome treba 10 sekundi da se učita preko spore bežične veze, onda to postaje kontraproduktivno i veoma nervira pokretne korisnike.

Sa širenjem upotrebe bežičnih mreža, sve su akutniji problemi izvršavanja TCP protokola u nijma. Detaljnije informacije o svemu što se u toj oblasti radi potražite kod Barakata i saradnika (2000), Ghanija i Dbđta (1999), Hustona (2001) i kod Xy- lomenosa i saradnika (2001).

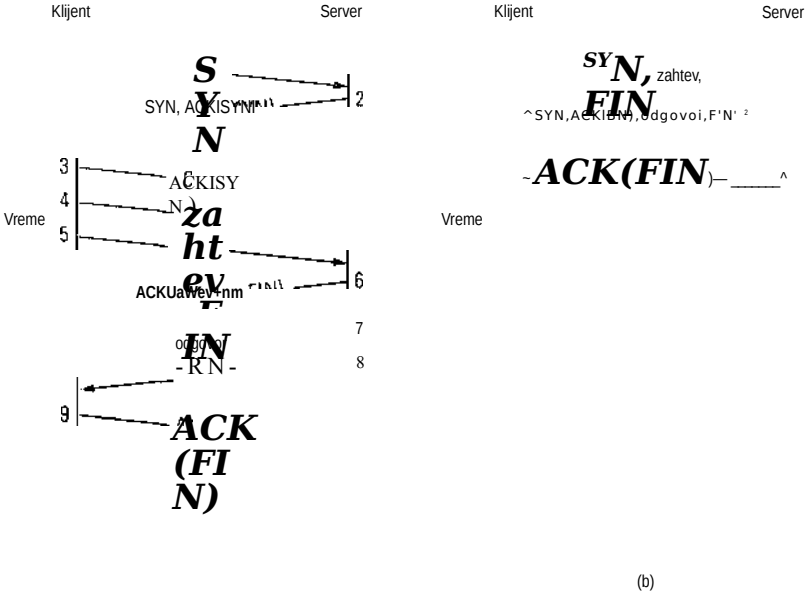
6.5.12 Transakcioni TCP protokol

U prvom delu ovog poglavlja razmatrali smo daljinsko pozivanje procedura kao način za realizovanje klijentsko-serverskih sistema. Ako su i zahtev i odgovor na njega dovoljno mali da stanu u po jedan paket, a operacija se sa istim rezultatom može neograničeno ponavljati, onda se UDP protokol može jednostavno koristiti; međutim, ukoliko nisu ispunjeni navedeni uslovi, UDP protokol je manje privlačan. Na primer, ako je odgovor veoma veliki, onda se mora podeliti u više delova i mora se obezbediti mehanizam za ponovno slanje izgubljenih delova. U stvari, od aplikacije se traži da ponovo izmisli TCP protokol.

Naravno, to ne dolazi u obzir, ali nije zgodno ni da se za to koristi TCP protokol. Problem leži u neefikasnosti. Redosled razmenjenih paketa za normalno daljinsko pozivanje procedure (RPC) uz protokol TCP prikazan je na slici 6-40(a). U najboljem slučaju, razmenjuje se devet paketa:

1. Klijent šalje paket *SYN* da bi uspostavio vezu.
2. Server šalje paket A *CK* da bi potvrdio prijem paketa *SYN*.
3. Klijent dovršava trostepeno usaglašavanje.
4. Klijent šalje stvarni zahtev.
5. Klijent šalje paket *FIN* da naznači kraj pošiljke.
6. Server potvrđuje zahtev i prijem paketa *FIN*.
7. Server šalje klijentu odgovor na zahtev.
8. Server šalje paket *FIN* da naznači kraj pošiljke.
9. Klijent potvrđuje prijem paketa *FIN*.

Imajte na umu da je u pitanju najpovoljniji slučaj. U najnepovoljnijoj situaciji, kli- jentov zahtev i paket FIN potvrđivali bi se nezavisno jedan od drugog, kao i odgovor, odnosno paket FIN sa servera.



(a) (b)

Slika 6-40. (a) RPC uz normalni TCP protokol, (b) RPC uz protokol T/TCP.

Brzo dolazimo do pitanja mogućnosti kombinovanja efikasnosti daljinskog pozivanja procedura pomoću protokola UDP (samo dve poruke) i pouzdanosti protokola TCP. Izgleda daje tako nešto *skoro* moguće izvesti, i to pomoću eksperimentalne varijante TCP protokola, zvane transakcioni TCP protokol (engl. *Transactional TCP, T/TCP),* koja je opisana u RFC dokumentima 1379 i 1644.

Osnovna zamisao je da se tokom uspostavljanja veze malo izmeni standardni re- dosled operacija da bi se i tokom uspostavljanja veze omogućio prenos podataka. Protokol T/TCP prikazan je na slici 6-40(b). Klijentov prvi paket sadrži bit *SYN,* sam zahtev, i bit *FIN.* On, u osnovi, saopštava: Želim da uspostavim vezu, evo ti podataka i s tim sam završio posao.

Kada server dobije zahtev, on traži podesan odgovor ili ga sastavlja, birajući način da ga pošalje. Ako odgovor staje u jedan paket, on daje odgovor prikazan na slici 6-40(b): Potvrđujem tvoj *FIN,* evo ti odgovora i time sam završio slanje. Klijent zatim potvrđuje *FIN* dobijen sa servera i protokol se prekida posle tri razmenjene poruke.

Međutim, ako odgovor servera ne staje u jedan paket, server ima mogućnost da ne šalje bit *FIN* sve dok ne pošalje redom sve pakete s odgovorom, posle čega zatvara svoj smer veze.

Treba možda istaći da protokol T/TCP nije i jedino predloženo poboljšanje protokola TCP. Drugi predlog je protokol za upravljanje tokom podataka (engl. *Stream Control Transmission Protocol, SCTP)*. *U* njemu se čuva granica između uzastopnih poruka, on podržava više režima isporuke (npr. prekoredno isporučivanje paketa), višematičnost (rezervna odredišta) i selektivno potvrđivanje (Stewart i Metz, 2001). Međutim, kad god neko predloži menjanje nečega što već dugo radi sasvim dobro, digne se prašina između onih koji zahtevaju nove osobine i onih koji odobravaju popravku samo ako je nešto zaista puklo.

6.6 PERFORMANSE

Performanse su vrlo važna tema u računarskim mrežama. Kada se međusobno po- vežu stotine ili hiljade računara, često dolazi do složenih interakcija s nepredvidljivim posledicama koje pogoršavaju performanse mreže iz razloga koji se teško mogu utvrditi. U narednim odeljcima razmotricemo brojne činioce koji utiču na performanse mreže da bismo otkrili moguće probleme i sagledali kako se oni mogu rešavati.

Nažalost, razumevanje performansi mreže pre je veština nego nauka. Još uvek ne postoji dovoljna teorijska podloga koja bi se mogla korisno upotrebiti u praksi. Možemo jedino da vam ukažemo na empirijska pravila i da vam iznesemo primere iz stvarnog sveta. Ovu priču smo namerno smestili iza objašnjenja transportnog sloja u TCP prenosu da bismo TCP mogli da upotrebimo u primerima.

Transportni sloj nije i jedino mesto gde se javljaju problemi s performansama. Vi- deli smo u prethodnom poglavlju da se oni javljaju i u mrežnom sloju. Pa ipak, mrežni sloj se uglavnom bavi usmeravanjem i kontrolom zagušenja. Širi, sistemski pristup problematici performansi vezanje za prenos podataka, tako daje ovo poglavlje pravo mesto gde ih treba pretresti.

U sledećih pet odeljaka upoznaćemo se s pet aspekata performansi mreže:

1. Problemi s performasama.
2. Merenje performansi mreže.
3. Projektovanje sistema za postizanje boljih performansi.
4. Brza obrada TPDU blokova.
5. Protokoli za buduće mreže visokih performansi.

Pre svega, potreban nam je opšti izraz za blokove podataka koje razmenjuju transportne jedinice. TCP segment je neprecizan izraz i nikada se u ovom kontekstu ne koristi izvan TCP sveta. ATM nazivi (CS-PDU, S AR-PDU i CPCS-PDU) važe samo za ATM mreže. Slično tome, paketi su vezani samo za mrežni sloj, a poruke pripadaju sloju aplikacija. Zbog nepostojanja standardnog imena za blok podataka koji međusobno razmenjuju transportne jedinice, vratićemo se na početak i taj blok zvati jedinica podataka transportnog protokola (engl. Transport Protocol Data Unit) ili kratko TPDU blok. Kada istovremeno mislimo na TPDU blok i na paket, upotrebićemo izraz „paket“, kao u rečenici „Mikroprocesor mora da bude dovoljno brz da bi dolazne pakete mogao da obradi u realnom vremenu“. Pod tim istovremeno podrazumevamo i paket mrežnog sloja i TPDU blok kapsuliran u njemu.

6.6.1 Problemi s performansama u računarskim mrežama

Neki problemi, kao što je zagušenje mreže, izazvani su privremenim preopte- rećenjem resursa. Ako usmerivaču najednom stigne više podataka nego što je u stanju da obradi, doći će do zagušenja i performanse će se pogoršati. O zagušenju smo detaljno raspravljali u prethodnom poglavlju.

Performanse su lošije i kada postoji strukturna neravnoteža resursa. Na primer, kada se skroman PC računar poveže na gigabitnu komunikacionu liniju, njegov mikroprocesor neće modi dovoljno brzo da obrađuje pakete koji pristižu i neki paketi će se gubiti. Oni će možda biti ponovo poslati, što povećava kašnjenje, troši propusni opseg i obično pogoršava performanse.

Do preopterećenja može doći i sinhronizovanom akcijom. Na primer, ako TPDU blok nosi pogrešan parametar, kao što je nepostojeći odredišni priključak, primalac će u mnogim slučajevima poslati odgovor sa upozorenjem na grešku, što je po sebi sasvim u redu. Zamislite sada šta će se dogoditi ako se neispravan TPDU blok neusmereno pošalje u mrežu sa 10.000 računara: svaki od njih može da odgovori porukom o grešci. Rezultat je bujica neusmerenih paketa (engl. *broadcast storm)* koja može da zaguši mrežu. Taj problem je postojao u protokolu UDP sve dok protokol nije tako izmenjen da računati više ne odgovaraju na greške u UDP paketima poslatim neusmereno.

Drugi primer sinhronizovanog preopterećivanja je situacija posle iznenadnog nestanka struje u mreži. Kada struja ponovo dođe, svi računati istovremeno pristupaju ROM-u da bi podigli svoje sisteme. Pri tome, računar najčešće prvo mora da od (DHCP) servera sazna svoj identitet, a zatim da od servera datoteka preuzme kopiju operativnog sistema. Ako to istovremeno urade stotine računara, serveri će verovatno otkazati.

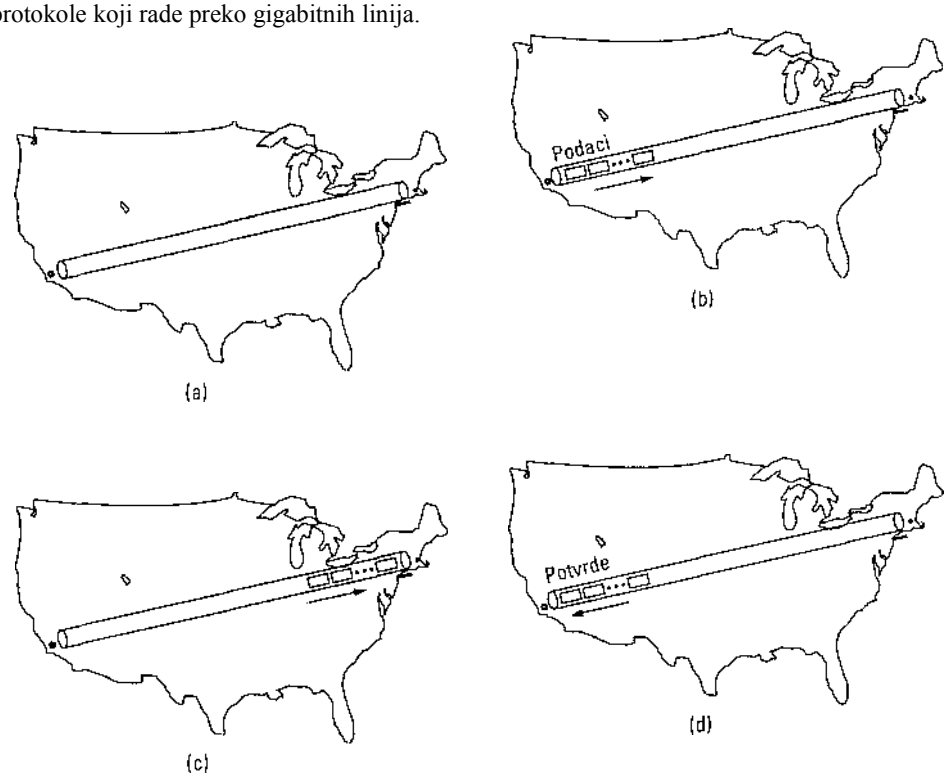
Čak i kada nema sinhronizovanog opterećivanja i kada postoji dovoljno resursa, performanse mogu da budu loše zbog nepodešenosti sistema. Na primer, ako računar ima brz procesor i veliku memoriju, ali nije dovoljno memorije dodeljeno baferima, dolaziće do njihovog prelivanja i podaci će se gubiti. Slično tome, ako algoritam za raspoređivanje poslova ne dodeli dovoljno visok prioritet obradi dolaznih TPDU blokova, neki od njih će se gubiti.

Isto tako, važno je i ispravno podešavanje tajmera. Kada se pošalje TPDU blok, tajmer se obično podešava tako da reaguje na njegov gubitak. Ako je rok tajmera pre- kratak, mreža će se opterećivati nepotrebno poslatim duplikatima. Ukoliko je rok tajmera predugačak, nepotrebno će se produžiti kašnjenje posle gubljenja TPDU bloka. U druge parametre koji se mogu podešavati spada period čekanja na podatke za šle- povanje potvrde pre nego što se pošalje zasebna potvrda, kao i broj pokušaja ponovnog slanja TPDU bloka pre definitivnog odustajanja.

Gigabitne mreže donose sa sobom i nove probleme s performansama. Razmotrite, na primer, slanje rafala od 64 KB podataka iz San Dijega u Boston, s ciljem da se ispuni isto toliko veliki bafer primaoca. Pretpostavimo da veza radi brzinom 1 Gb/s i da signalu treba 20 ms da stigne s jednog na drugi kraj. Na početku, u trenutku $t = 0$, na kanalu nema nijednog TPDU bloka, kao na slici 6-41 (a). Samo posle 500 ps, svi TPDU blokovi su na kanalu, kao na slici 6-41 (b). Vodeći TPDU blok nalazi se negde kod Browleya, još uvek duboko u Južnoj Kaliforniji. Međutim, pošiljalac mora da prekine slanje sve dok ne dobije novu veličinu prozora.

Posle 20 ms vodeći TPDU blok stiže u Boston, kao na slici 6-41 (c), i za njega se šalje potvrda. Konačno, 40 ms nakon početka, prva potvrda stiže pošiljaocu i on može da pošalje drugi rafal. Pošto se prenosna linija koristi samo tokom 0,5 ms od ukupno

40 ms, efikasnost njenog korišćenja je oko *1,25%.* Takva situacija je tipična za starije protokole koji rade preko gigabitnih linija.



Slika 6-41. Stanje prenosa jednog megabita podataka iz San Dijega u Boston.
(a) U trenutku $t = 0$. (b) Posle 500 fts. (c) posle 20 ms. (d) posle 40 ms.

Pri analiziranju performansi mreže koristan parametar je **proizvod opsega i kašnjenja** (engl. *bandwidth-delay product).* On se dobija množenjem propusnog opsega (b/s) s vremenom obilaska veze (s). Proizvod predstavlja kapacitet kanala od pošiljaoca do primaoca i natrag (u bitovima).

Za primer na slici 6-41 ovaj proizvod je 40 miliona bitova. Drugim recima, pošiljalac treba da šalje rafale od 40 miliona bitova da bi još uvek radio punom snagom kad mu stigne prva potvrda. Toliko je bitova potrebno da ispuni kanal (u oba smera). To je i razlog niske efikasnosti rafala veličine samo pola miliona bitova jer oni ispunjavaju samo 1,25% raspoloživog kapaciteta kanala.

Iz gornjeg razmatranja može se zaključiti da će performanse biti dobre ako je prozor primaoca jednak proizvodu opsega i kašnjenja, a bolje je da bude nešto veći jer primalac ne mora odmah da reaguje. Za transkontinentalne megabitne linije, veličina prozora treba da

bude barem 5 megabajta.

Ako je efikasnost katastrofalno niska za rafale veličine megabita, zamislite tek kakva je za kratke zahteve dužine par stotina bajtova. Ukoliko se linija nečim ne zaposli dok prvi klijent čeka odgovor, gigabitna linija će raditi kao megabitna, samo uz veće troškove.

Još jedan problem s performansama koji se javlja kod aplikacija u kojima je vreme važan parametar, kao što su audio i video, tiče se neravnomernosti pristizanja paketa. Nije dovoljno obezbediti kratko prosečno vreme prenosa, potrebno je da i standardno odstupanje bude malo. Obezbeđivanje kratkog prosečnog vremena prenosa uz malo standardno odstupanje zahteva znatan inženjerski trud.

6.6.2 Merenje performansi mreže

Kada mreža radi loše, korisnici najčešće okrivljuju one koji je održavaju i od njih zahtevaju da joj poboljšaju performanse. Da bi to mogli, operateri moraju prvo da tač- no utvrde šta se u mreži događa, tj. da izvrše odgovarajuća merenja. U ovom odeljku pozabavićemo se merenjem performansi mreže. Naredno izlaganje se zasniva na radu Mogula (1993).

Osnovni ciklus petlje pomoću koje se mogu poboljšati performanse mreže sadrži sledece korake:

1. Merenje odgovarajućih parametara i performansi mreže.
2. Sagledavanje stanja na osnovu izmerenih vrednosti.
3. Menjanje jednog parametra.

Navedeni koraci se ponavljaju sve dok se performanse dovoljno ne poboljšaju ili dok ne postane jasno da se više ništa ne može učiniti.

Merenja se mogu izvesti na više načina i na mnogim mestima (i na fizičkim lokacijama i u skupu protokola). U osnovna merenja spada i merenje vremena potrebnog za obavljanje neke aktivnosti pomoću tajmera. Na primer, osnovno je merenje vremena potrebnog da se potvrdi TPDU blok. Druga merenja podrazumevaju određivanje učestalosti nekog događaja (npr. učestalost gubljenja TPDU blokova). I na kr aju, često nas mogu zanimati neke količine, npr. broj bajtova obrađenih u određenom vremenu.

Pri sistematskom, ozbiljnom merenju performansi i parametara mreže moraju se izbeći mnoge skrivene zamke, od kojih nekoliko opisujemo u nastavku.

Uvek obezbedite dovoljno velik uzorak

Za merenje vremena slanja nemojte koristiti jedan TPDU blok, već ponovite merenje, recimo, milion puta i izračunajte prošek. Kada radite s velikim uzorkom, do- bićete pouzdaniju srednju vrednost i standardno odstupanje. Ta (ne)pouzdanost može se izračunati standardnim statističkim tehnikama.

Neka uzorak bude reprezentativan

Idealno bi bilo da čitav niz od milion merenja ponovite u različito doba dana i sedmice, i tako utvrdite uticaj različite opterećenosti sistema na merenu veličinu. Meriti zagušenje, na primer, ima malo smisla ako u tom trenutku mreža nije zagušena. Po- nekada rezultati na prvi pogled mogu da izgledaju besmisleno, na primer, veliko zagušenje u 10, 11, 1 i 2 sata i praktično prazna mreža oko 12 sati, ali se treba dosetiti da su u to vreme svi zaposleni na ručku.

Vodite računa o preciznosti sistemskog sata

Sistemski satovi u računalima rade tako što u redovnim intervalima uvećavaju vrednost

brojača za jedan. Na primer, milisekundni tajmer uvećava vrednost brojača svake milisekunde. Takvim tajmerom može se pratiti i događaj koji traje manje od 1 ms, ali je za to potrebna posebna pažnja. (Neki računari, naravno, imaju preciznije satove).

Da bi se izmerilo vreme slanja TPDU bloka, na primer, sistemski sat (npr. milisekundni) trebalo bi očitati u trenutku kada se započne izvršavanje koda transportnog sloja i zatim ponovo, kada se rečeni kod završi. Ako stvarno vreme slanja TPDU bloka iznosi 300 ps, razlika između dva očitavanja sata biće 0 ili 1 - oboje pogrešno. Međutim, ako se merenje ponovi milion puta, svi rezultati saberu i podele sa milion, izmereno srednje vreme slanja približiće se stvarnoj vednosti na manje od 1 ps.

Preduzmite sve da se tokom merenja ne dogodi ništa nepredviđeno

Ako merite performanse univerzitetske mreže baš onda kada se u rad pušta neki veliki projekat, možda ćete dobiti drastično drugačije rezultate od onih koje biste dobili da ste merenje odložili za sledeći dan. Slično tome, ako je neki istraživač odlučio da drži video konferenciju na mreži baš onog dana koji ste vi odredili za merenje njenih performansi, dobićete lažne rezultate. Najbolje je da merenje vršite na potpuno neaktivnom sistemu i da opterećenje generišete sami, ali i tu su moguća iznenađenja. Iako pouzdano znate da mrežu niko neće koristiti u 3 sata izjutra, to može baš da bude vreme kada se automatski uključuje program za rezervno kopiranje svih diskova na traku. Štaviše, možete naleteti i na gust saobraćaj iz drugih časovnih zona iz kojih udaljeni korisnici pokušavaju da se uključe na vašu božanstvenu Web lokaciju.

Keširanje podataka može potpuno da upropasti rezultate merenja

Merenje vremena potrebnog za prenos datoteka obuhvata otvaranje velike datoteke, njeno učitavanje, zatvaranje i, naravno, beleženje vremena pre i posle toga. Kada to ponovite mnogo puta, smatrate da ste dobili pouzdan prošek, ali je problem u tome što sistem može da kešira datoteku, tako da samo u prvom merenju imate stvarni saobraćaj na mreži. U svim ostalim, datoteka se učitava iz lokalnog keša. Rezultati takvog merenja bez daljeg su neupotrebljivi (osim ako ste želeli da merite efikasnost keširanja).

Keširanje često možete da prevarite tako što ćete namerno prepuniti keš. Na primer, ako je keš veličine 10 MB, eksperimentalna petlja treba da u svakom prolazu otvori, učita i zatvori dve datoteke od po 10 MB, kako bi keširanje svela na nulu. Pa ipak, savetujemo oprez jer morate detaljno znati kako radi algoritam za keširanje.

Privremeno skladištenje u bafer može da ima slične efekte. Zna se da je jedan popularan program za merenje TCP/IP performansi stalno davao rezultate UDP prenosa koji su znatno prevazilazili performanse same fizičke linije. Kako je to uopšte moguće? Pozvana UDP procedura obično vraća kontrolu programskom toku čim jezgro operativnog sistema prihvati poruku i svrsta je u red čekanja za slanje. Ako ima dovoljno prostora u baferu, ne znači da će posle 1000 uzastopnih poziva UDP proceduri svi podaci stvarno biti i poslati. Većina ih još uvek može boraviti u jezgru, ali program za merenje performansi to ne zna i misli da su svi poslati.

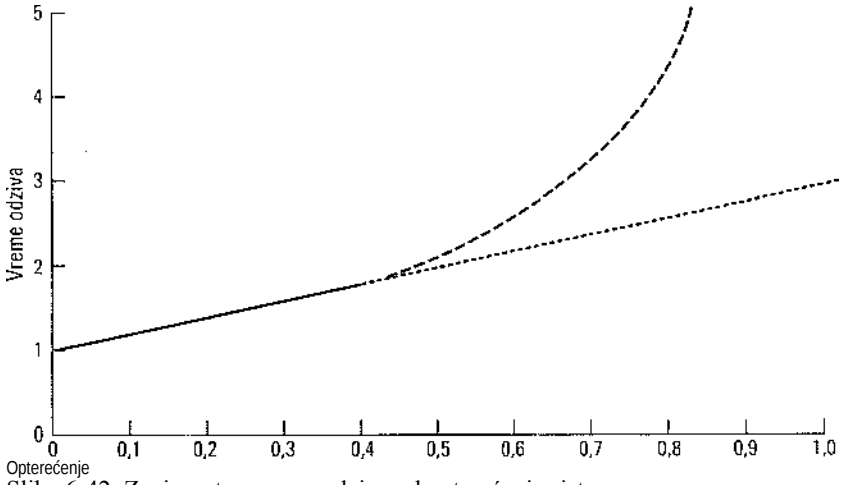
Morate znati šta merite

Ako merite vreme učitavanja udaljene datoteke, rezultat će zavisiti od mreže, operativnih sistema klijenta i servera, mrežnih kartica, njihovih upravljačkih programa itd. Ukoliko merenje izvedete s dužnom pažnjom, dobićete pouzdano vreme učitavanja udaljene datoteke za konkretnu konfiguraciju sistema. Ako vam je to bio i jedini cilj, sve je u redu.

Međutim, ukoliko sličnim merenjem proveravate tri različita sistema da biste utvrdili koji ima najbolju mrežnu karticu (jer takvu želite i sami da imate), možete da izvedete sasvim pogrešan zaključak zato što je, na primer, jedan od tri upravljačka programa za mrežne kartice zaista loše napisan, tako da iskorišćava jedva desetak procenata njenih mogućnosti.

Pazite kada uopštavate rezultate

Pretpostavimo da merite neki parametar simulirajući opterećenje mreže u opsegu od 0 (bez aktivnosti) do 0,4 (40 % kapaciteta) i da ste dobili rezultate prikazane punom linijom na slici 6-42. Padate u iskušenje da rezultate ekstrapolirate linearno (isprekidana prava). Budući da svrstavanje u redove čekanja često uključuje faktor 1/ (1 - p), gde p predstavlja operećenje, verovatnije je da će stvarni rezultati merenja (koje niste izveli) slediti isprekidanu krivu liniju koja raste mnogo brže.



Slika 6-42. Zavisnost vremena odziva od opterećenja sistema.

6.6.3 Projektovanje sistema za postizanje boljih performansi

Merenje i podešavanje često može znatno da poboljša performanse, ali ništa ne može da zameni dobar projekat mreže. Ako je mreža na početku loše projektovana, ona se može poboljšati do izvesne mere, ali sve preko toga zahteva njenu potpunu rekonstrukciju.

U ovom odeljku iznećemo izvesna pravila zasnovana na dugogodišnjem iskustvu s mnogim mrežama. Ona se ne odnose samo na projektovanje mreža, već i na projektovanje čitavih sistema jer su softver i operativni sistem često važniji od usmeri- vača i mrežnih kartica. Ta pravila godinama kruže među projektantima mreža, prenoseći se usmenim putem s kolena na koleno. Prvi ih je pismeno definisao Mogul (1993) i mi ćemo se držati njegovog pristupa. O njima je pisao i Metcalfe (1993).

Pravilo 1: Brzina procesora je važnija od brzine mreže

Dugogodišnje iskustvo, prikupljeno iz skoro svih mreža, pokazuje da je trajanje mrežnih operacija uglavnom uslovljeno operativnim sistemom i nekorisnim protokolarnim saobraćajem. Na primer, minimalno vreme daljinskog pozivanja procedure na Ethernetu

iznosi 102 ps, što odgovara zahtevu i odgovoru minimalne veličine (po 64 bajta). U praksi, to vreme se ni izbliza ne može postići upravo zbog trajanja softverske obrade.

Slično tome, u gigabitnom prenosu podataka najveći problem je da se bitovi iz korisnikovog bafera dovoljno brzo prebace na optički kabl i da ih mikroprocesor primaoca dovoljno brzo obradi. Ukratko, ako udvostručite takt mikroprocesora, često ćete skoro udvostručiti i protok podataka. Dupliranje kapaciteta mreže u ovoj situaciji nema efekta jer se usko grlo nalazi u računarima.

Pravilo 2: Smanjite broj paketa da biste skratili vreme obrade

Obrada svakog TPDU bloka obuhvata obradu zaglavlja i obradu svakog bajta (izračunavanje kontrolnog zbira). Kada se šalje milion bajtova, obrada svakog bajta traje isto, bez obzira na veličinu TPDU bloka. Međutim, ako za slanje upotrebite TPDU blokove veličine 128 bajtova, time ćete (neproduktivno) obrađivati 32 puta više zaglavlja nego kada biste upotrebili TPDU blokove od 4 KB. Taj višak obrade brzo se akumulira.

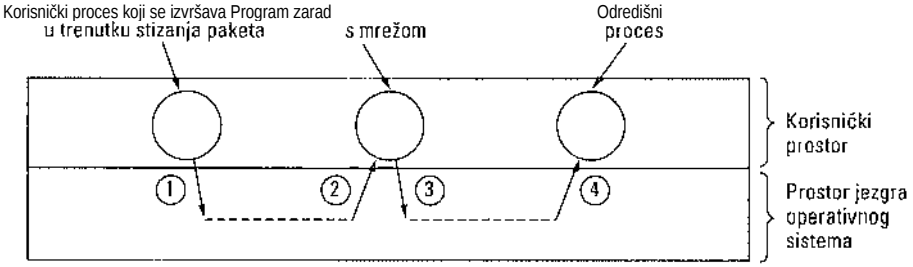
Osim obrade TPDU zaglavlja, treba razmotriti i trajanje obrade u nižim slojevima. Svaki pristigli paket izaziva softverski prekid. U savremenim mikroprocesorima za paralelnu obradu podataka (engl. *pipelineđ),* svaki softverski prekid raskida liniju obrade, utiče na kešmemoriju, zahteva izmene u kontekstu rada s memorijom i zauzima priličan broj registara mikroprocesora. AMostruko smanjenje broja TPDU blokova smanjilo bi i broj prekida, odnosno broja obrada paketa takođe *n* puta.

Navedeno zapažanje govori u prilog prikupljanja znatne količne podataka pre njihovog slanja da bi se smanjio broj prekida na dragom kraju. Nagleov algoritam i Klarkovo rešenje za sindrom luckastog prozora upravo predstavljaju pokušaje u tom pravcu.

Pravilo 3: Izbegavajte čest prelazak iz jednog konteksta u drugi

Menjanje konteksta (npr. iz režima jezgra u korisnički režim) deluje pogubno. Po- sledice su kao kod softverskih prekida, a najgora je dug niz instrukcija koje nisu pronađene u priručnoj memoriji (kešu). Menjanje konteksta može se proreditи ako se primeni procedura koja će podatke smeštati u interni bafer sve dok ih se ne nakupi pristojan broj. Slično tome, trebalo bi kod primaoca prikupljati kratke TPDU blokove i prosleđivati ih korisniku s vremena na vreme.

Dolazni paket u najboljem slučaju izaziva prelazak iz konteksta aktuelnog korisnika u kontekst jezgra, a zatim prelazak na proces kome paket isporučuje nove podatke. Nažalost, u mnogim operativnim sistemima kontekst se menja i dodatno. Na primer, ako se program za rad s mrežom izvršava kao specijalan proces u korisničkom prostom, stizanje paketa verovatno će izazvati prelazak iz korisničkog konteksta u kontekst jezgra, a zatim iz konteksta jezgra u kontekst programa za rad s mrežom, iza čega slede povratak u kontekst jezgra i prelazak u kontekst odredišnog procesa. Takav redosled događaja prikazanje na slici 6-43. Svi ti prelasci izazvani pristizanjem paketa intenzivno troše procesorsko vreme i imaju katastrofalne posledice na mrežne performanse.



Slika 6-43. Za obradu pristiglog paketa potrebna su četiri prelaska iz jednog konteksta u drugi kada se program za rad s mrežom izvršava u korisničkom prostoru.

Pravilo 4: Proredite kopiranje

Od višekratnog menjanja konteksta gore je višekratno kopiranje. Nije retko da se pristigli paket kopira tri ili četiri puta pre nego što se iz njega preuzme TPDU blok. Postoje paket stigao u specijalni, hardverski ugrađen bafer mrežne kartice, on se najčešće kopira u bafer jezgra operativnog sistema. Odatle se kopira u bafer mrežnog sloja, zatim u bafer transportnog sloja i na kraju, u proces odredišne aplikacije.

Inteligentan operativni sistem kopiraće reč po reč, ali nije neobično da mu za kopiranje jedne reči treba pet instrukcija (učitavanje, skladištenje, povećavanje vredno- sti indeksnog registra, provera kraja podataka i uslovno grananje). Kada pravite tri kopije svakog paketa uz pet instrukcija po svakoj 32-bitnoj reči, to iznosi 15/4 ili oko četiri instrukcije po bajtu. U procesora brzine 500 MIPS, instrukcija traje 2 ns, tako da je za obradu svakog bajta potrebno 8 ns procesorskog vremena ili oko 1 ns po bitu, što za maksimalnu brzinu obrade daje oko 1 Gb/s. Kada se uračuna i obrada zaglavlja, obrada prekida i menjanje konteksta, možda će se postići brzina 500 Mb/s, a još nismo ni stigli do stvarne obrade podataka. Jasno je da se mogućnosti Ethernet mreže brzine 1 Gb/s ni približno ne mogu iskoristiti.

U stvari, možda se ne može potpuno iskoristiti ni linija brzine 500 Mb/s. U prethodnom računu pretpostavili smo da procesor brzine 500 MIPS može da izvrši 500 miliona bilo kojih instrukcija u sekundi. U stvarnosti, računar može da radi takvom brzinom samo ako procesor ne pristupa memoriji. Operacije s memorijom često su desetak puta sporije od operacija između registara (oko 20 ns po instrukciji). Ako 20% instrukcija stvarno pristupa memoriji (jer ih nema u kesu), stoje moguće kada se prihvataju dolazni paketi, prosečno vreme izvršavanja instrukcije iznosi 5,6 ns (0,8 x 2 + 0,2 x 20). Uz četiri instrukcije po bajtu, za obradu jednog bajta trošimo 22,4 ns ili 2,8 ns po bitu, što ukupno daje oko 357 Mb/s. Ako na obradu zaglavlja itd. otpadne 50% procesorskog vremena, dobijamo vrednost 178 Mb/s. Imajte na umu da hardver ovde ne može da pomogne. Problem je u tome što operativni sistem previše kopira.

Pravilo 5: Možete da obezbedite veći propusni opseg, ali ne i manje kašnjenje

Sledeća tri pravila više se tiču komunikacija, nego protokolarne obrade podataka. Prvo pravilo kaže da ako želite veći propusni opseg, uvek možete da ga dokupite. Kada pored postojećeg optičkog kabla položite još jedan, udvostručavate propusni opseg, ali se kašnjenje ne menja. Kašnjenje se može smanjiti samo ako poboljšate protokol, operativni sistem ili mrežni interfejs. I uz sva navedena poboljšanja, kašnjenje se neće smanjiti ako je ograničavajući činilac vreme prenosa.

Pravilo 6: Zagušenje je bolje izbeći, nego se od njega oporavljati

Stara izreka „bolje sprečiti, nego lečiti“ direktno se može primeniti na zagušenja u mreži. Kada se mreža zaguši, gube se paketi, troši se propusni opseg, dolazi do nepotrebnog kašnjenja itd. Za oporavak od zagušenja potrebno je vreme i strpljivost. Bolje je sprečiti ga da nastane. Izbegavanje zagušenja je kao vakcina: na početku malo boli, ali ste kasnije mirni.

Pravilo 7: Ne dozvolite često isključivanje tajmera

Tajmeri su u mrežama neophodni, ali ih treba koristiti s merom i ne dozvoliti da se prečesto isključuju. Kada se tajmer automatski isključi, obično se ponavlja neka akcija. Ako je to neophodno, neka tako i bude, ali nepotrebno ponavljanje akcije samo traći vreme i propusni opseg.

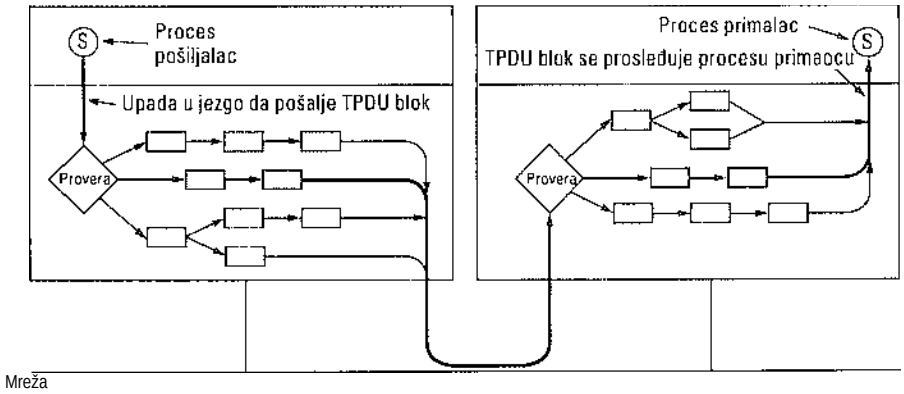
Zbog toga je pri podešavanju roka tajmera potreban malo konzervativniji pristup. Ako je rok isključivanja tajmera nešto duži, to će neznatno povećati kašnjenje na vezi u (malo verovatnom) slučaju gubljenja TPDU bloka. Tajmer koji se isključuje i kada treba i kada ne treba troši dragoceno procesorsko vreme i propusni opseg, i nepotrebno opterećuje možda desetine usmerivača.

6.6.4 Brza obrada TPDU blokova

Gornja priča nas uči da glavnu prepreku brzom radu u mreži predstavlja softver mrežnih protokola. U ovom odeljku razmotrićemo nekoliko načina za povećanje brzine njegovog rada. Detaljnije informacije potražite kod Clarka i saradnika (1989) i kod Chasea i saradnika (2001).

Administrativna (nekorisna) obrada TPDU bloka ima dve komponente: obradu po svakom TPDU bloku i obradu po svakom bajtu bloka. Obe se moraju skratiti. Ključ za brzu obradu TPDU blokova leži u izdvajanju normalnog slučaja (jednosmernog prenosa podataka) i njegovoj posebnoj obradi. Iako je za prelazak u stanje *ESTABLISHED* potrebno razmeniti niz specijalnih TPDU blokova, kada se to postigne, onda obrada TPDU blokova ide bez problema sve dok jedna strana ne inicira raskidanje veze.

Počnimo tako što ćemo u stanju *ESTABLISHED* analizirati pošiljaoca koji ima podatke spremne za slanje. Ovde ćemo zbog jednostavnosti pretpostaviti da je transportna jedinica smeštena u jezgro operativnog sistema, iako sve važi i za slučaj kad ona predstavlja proces u korisničkom prostom, odnosno biblioteku unutar izvorišnog procesa. Na slici 6-44, proces pošiljalac upada u jezgro da bi izvršio operaciju SEND. Transportna jedinica najpre proverava da li je u pitanju normalan slučaj: stanje je *ESTABLISHED,* nijedna strana ne pokušava da raskine vezu, šalje se regularan (dakle, ne izvanserijski) potpun TPDU blok, a prozor primaoca je dovoljno velik. Ako su zadovoljeni svi uslovi, ne treba dalje proveravati i može se uspostaviti brza putanja kroz transportnu jedinicu pošiljaoca. Ta putanja se koristi tokom najvećeg dela vremena.



Slika 6-44. Brza putanja od pošiljaoca do primaoca označena je zadebljanom linijom. Faze obrade na njoj prikazane su sivo.

Zaglavlja uzastopnih TPDU blokova s podacima najčešće su skoro ista. Tu činjenicu transportna jedinica iskorišćava tako što čuva prototip zaglavlja. Kada se uspostavi brza putanja, transportna jedinica što brže kopira prototip u priručni bafer, reč po reč. Polja koja se razlikuju od jednog do drugog TPDU bloka upisuju se preko postojećih vrednosti u baferu. Ta polja se često mogu izvesti iz promenljivih koje opisuju stanje, kao što je sledeći redni broj. Mrežnom sloju se tada prosleđuje pokazivač na potpuno TPDU zaglavlje i pokazivač na korisničke podatke. Tu se može primeniti slična strategija (nije prikazana na slici 6-44). Na kraju, mrežni sloj predaje rezul- tujući paket sloju veze podataka za slanje.

Pogledajmo kako ovaj princip radi u praksi na primeru protokola TCP/IP. Slika 6-45(a) prikazuje TCP zaglavlje. Polja koja se ne menjaju u uzastopnim TPDU blokovima jednosmernog toka označena su sivo. Transportna jedinica treba samo da kopira pet reči iz prototipa zaglavlja u izlazni bafer, da unese sledeći redni broj (kopirajući ga iz memorisane reči), da izračuna kontrolni zbir, i da za jedan poveća redni broj u memoriji. Ona tada može da preda zaglavlje i podatke specijalnoj IP proceduri za slanje regularnog TPDU bloka maksimalne veličine. IP tada kopira pet reči svog prototip- skog zaglavlja [slika 6-45(b)] u bafer, popunjava polje *Identifikacija* i izračunava kontrolni zbir. Paket je sada spreman za slanje.

Izvorišni priključak	Odredišni priključak
Redni broj	
Broj potvrde	
Dužina Ne koristi se	Veličina prozora
Kontrolni zbir	Pokazivač na hitne podatke

Verz. DIZ	Vrsta usluge	Ukupna dužina
Identifikacija		Redni broj fragmenta
Životni vek	Protokol	Kontrolni zbir zaglavlja
Adresa izvorišta		
Adresa odredišta		

(a) (b)

Slika 6-45. (a) TCP zaglavlje, (b) IP zaglavlje. Siva polja su u oba slučaja preuzeta od prototipa bez izmene.

Pogledajmo sada šta se događa kod primaoca na slici 6-44. U prvom koraku, locira se zapis o vezi za dolazni TPDU blok. U protokolu TCP, zapis o vezi se može čuvati u tabeli ključeva u kojoj ključ predstavlja neku jednostavnu funkciju dve IP adrese i dva priključka. Kada se locira zapis o vezi, moraju se proveriti obe IP adrese i oba priključka i time potvrditi daje pronađen pravi zapis.

Cesto se traženje zapisa o vezi može ubrzati ako se pokazivač ostavi na posled- njem korišćenom zapisu i on prvo pogleda. Clark i saradnici (1989) isprobali su ovaj trik i utvrdili da se tačan zapis dobija u preko 90% slučajeva. Kod McKenneyja i Do- vea (1992) možete naći još postupaka za pronalaženje zapisa o vezi.

Zatim se proverava regularnost TPDU bloka: stanje je ESTABLISHED, nijedna strana ne pokušava da raskine vezu, nema specijalnih indikatora, a redni broj odgovara očekivanom. Ove provere se obavljaju pomoću samo nekoliko instrukcija. Ako su svi uslovi zadovoljeni, poziva se specijalna TCP procedura brze putanje.

Procedura ažurira zapis o vezi i kopira podatke korisniku, istovremeno izračunavajući kontrolni zbir da ne bi još jednom prolazila kroz podatke. Ako je kontrolni zbir ispravan, ažurira se zapis o vezi i natrag šalje potvrda. Postupak kojim se zaglavlje najpre brzo proverava da li odgovara očekivanom, a zatim predaje specijalnoj proceduri koja obrađuje taj slučaj, naziva se **predviđanje zaglavlja** (engl. *header predic- tion).* Koristi se u mnogim TCP realizacijama. Kada se ova optimizacija iskoristi zajedno s drugim optimizacijama opisanim u ovom poglavlju, može se postići da TCP radi sa 90% brzine lokalnog kopiranja iz memorije u memoriju, pod pretpostavkom daje sama mreža dovoljno brza.

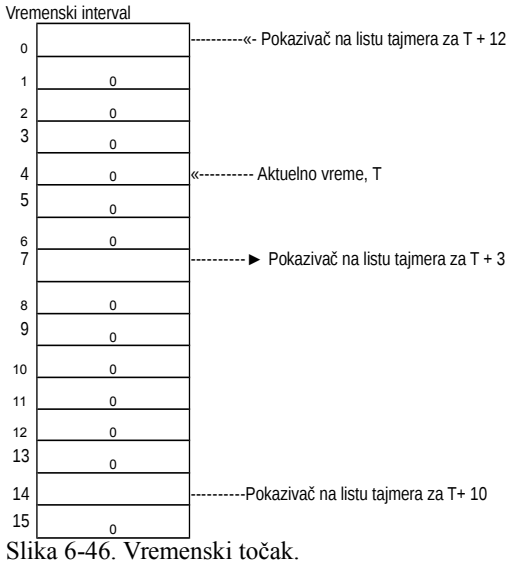
Podešavanje rada s baferima i tajmerima još su dva područja u kojima se značajno mogu poboljšati performanse. Kao što smo ranije istakli, u radu bafera treba izbeci nepotrebno kopiranje. Tajmere treba tako podesiti da se ne isključuju prečesto. Oni postoje kao obezbeđenje od gubljenja TPDU blokova, ali većina blokova stiže u redu, kao i potvrde za njih. Prema tome, tajmer treba podesiti tako da mu rok ne ističe prečesto.

Uobičajeno je da se napravi povezana lista akcija tajmera uređena prema roku. Naslovna odrednica sadrži brojač čija vrednost pokazuje broj otkucaja (sistemskog sata) do isključenja. Svaka sledeca odrednica sadrži broj otkucaja do isključenja, me- ren od prethodne odrednice. Tako, ako se tajmeri isključuju posle 3, 10, odnosno 12 otkucaja, vrednosti njihovih brojača su 3, 7, odnosno 2.

Vrednost brojača naslovne odrednice smanjuje se za jedan pri svakom otkucaju. Kada dostigne nulu, aktivira se akcija vezana za tajmer, a sledeća stavka na listi postaje naslovna. Njen brojač ne treba da se podešava. U ovakvom sistemu, skupo je umetati i brisati tajmere - trajanje operacije proporcionalno je dužini liste.

Ako se unapred zna maksimalan rok tajmera, problem se može mnogo efikasnije rešiti pomoću jednog niza, tzv. vremenskog točka (engl. *timing wheel),* prikazanog na slici 6-46. Svaki vremenski interval odgovara jednom otkucaju sistemskog sata. Aktuelnom trenutku na slici odgovara $T = 4$. Tajmeri su podešeni da se isključe posle 3, 10, odnosno 12 otkucaja u odnosu na alctuelni trenutak. Ako se odjednom pojavi nov tajmer s rokom isključenja 7 otkucaja, dovoljno je da se za interval 11 napravi nova odrednica. Slično tome, ako treba ukinuti tajmer s rokom $T + 10$, pretražuje se lista koja počinje intervalom 14 i iz nje uklanja

odgovarajuća odrednica. Obratite pažnju na to da niz sa slike 6-46 ne može da radi s tajmerima čiji rok prelazi $T + 15$.



Slika 6-46. Vremenski točak.

Pri svakom otkucaju sata pokazivač aktuelnog vremena prelazi u sledeći vremenski interval; kada stigne do kraja niza, vraća se na njegov početak. Ako u aktuelnom vremenskom intervalu postoji odrednica, obrađuju se svi tajmeri sa odgovarajuće liste. Kod Varghesea i Laucka (1987) naći ćete mnoge varijacije ove osnovne ideje.

6.6.5 Protokoli za gigabitne mreže

Čim su se devedesetih godina pojavile gigabitne mreže, na njih su primenjeni postojeći protokoli, ali su se odmah javili različiti problemi. Njima ćemo se pozabaviti u ovom odeljku i istovremeno ćemo razmotriti mere za njihovo rešavanje koje se pre- duzimaju u novim protokolima da bi mreže zaista radile sve brže i brže.

Prvi problem je u vezi sa 32-bitnim rednim brojevima. Na počecima Interneta, usmerivači su uglavnom bili povezani iznajmljenim linijama brzine 56 kb/s, tako da bi računar koji punom brzinom šalje podatke potrošio redne brojeve tek za nedelju dana. Za autore protokola TCP, broj 2 bio je prilično dobra aproksimacija beskonačnog jer praktično nije postojao rizik da na mreži zaostanu paketi poslati pre nedelju dana. U Ethernetu brzine 10 Mb/s ciklus reciklovanja rednih brojeva skratio se na 57 minuta, što je mnogo manje, ali ipak podnošljivo. U Ethernetu koji pumpa podatke na Internet brzinom 1 Gb/s, ciklus reciklovanja traje oko 34 sekunde, što je znatno manje od maksimalnog životnog veka paketa na Internetu (120 sekundi). I odjednom, broj 2^{32} više nije bio nikakva zamena za beskonačnost, pošto je pošiljalac mogao da potroši sve redne brojeve, a da stari paketi i dalje budu na mreži. U pomoć je pritekla jedna „kvaka“ opisana u RFC dokumentu 1323.

Za problem su u stvari krivi sami autori mnogih protokola, jer su (bez provere) po- drazumevali da ukupno vreme korišćenja celog skupa rednih brojeva uveliko preva- zilazi maksimalan životni vek paketa. Samim tim, nije se ni pretpostavljalo da na mreži postoje

duplikati starih paketa u trenutku pokretanja novog ciklusa rednih brojeva. Uz gigabitne brzine, te pretpostavke više ne važe.

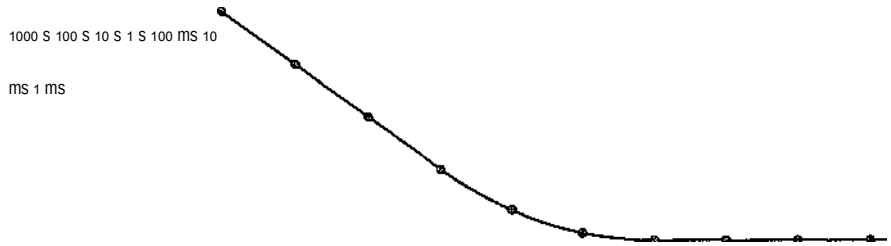
Drugi problem je u tome što se brzina komuniciranja povećala mnogo više od brzine računanja. (Poruka računarskim inženjerima: Krenite i pobedite već jednom te ,,telefondžije". Računamo na vas!) Sedamdesetih godina, mreža ARPANET je radila brzinom 56 kb/s, a imala je računare koji su radili brzinom 1 MIPS. Paketi su bili veličine 1008 bitova, pa je mreža mogla da isporučuje oko 56 paketa u sekundi. Imajući na raspolaganju skoro 18 ms, računar je za obradu svakog paketa mogao da potroši čak 18.000 instrukcija. Naravno, to bi potpuno zauzelo mikroprocesor, ali je on i dalje mogao da obradi paketa posveti 9000 instrukcija, ostavljajući polovinu procesorskog vremena za obavljanje stvarnog posla.

Uporedite ove brojeve s računarima brzine 1000 MIPS koji preko gigabitne linije razmenjuju pakete veličine 1500 bajtova. Svake sekunde može priticati više od
80.0 paketa, pa se obrada svakog od njih mora završiti unutar 6,25 ps ukoliko želimo da polovinu procesorskog vremena ostavimo za rad aplikacija. Računar brzine 1000 MIPS može da za 6,25 ps izvrši 6250 instrukcija, tj. samo trećinu onoga što su

imali na raspolaganju računari ARPANET-a. Osim toga, prosečna savremena RISC instrukcija obavlja manje posla nego stara CISC instrukcija, tako da je problem još veći nego što se čini. Sledi zaključak: pošto je na raspolaganju manje vremena za protokolarnu obradu, protokoli moraju biti jednostavniji.

Treći problem je u tome što protokol „vrati se n“ loše radi na linijama s velikim proizvodom opsega i kašnjenja. Razmotrite, na primer, liniju dužine 4000 km koja radi brzinom 1 Gb/s. Vreme obilaska linije iznosi 40 ms, a unutar njega pošiljalac može da emituje 5 megabajta. Ako dođe do greške, pošiljalac će za nju saznati tek nakon 40 ms. Ako se koristi protokol „vrati se n“, pošiljalac će morati da ponovo pošalje ne samo neispravan paket, već i 5 megabajta paketa koji slede iza njega. Očito je da se resursi tako razbacuju.

Četvrti problem potiče iz fundamentalne razlike između gigabitnih i megabitnih linija: dužinu gigabitne linije ne ograničava propusni opseg, već kašnjenje. Na slici 6-47 prikazujemo vreme potrebno za prenošenje datoteke od 1 megabita na daljinu od 4000 km, različitim brzinama. Do brzine od 1 Mb/s, vreme prenosa ograničava brzina slanja bitova. Pri brzini 1 Gb/s, vreme obilaska veze (40 ms) višestruko prevazilazi vreme (1 ms) potrebno da se podaci ubace u optičld kabl. Dalje povećanje propusnog opsega nema efekta.



Peti problem koji vredi pomenuti ne odnosi se ni na tehnologiju ni na protokole, već je rezultat pojave novih aplikacija. Jednostavno rečeno, za mnoge gigabitne aplikacije, kao što je multimedija, nije važno samo prosečno kašnjenje paketa, već i njegova varijansa. Spora, ravnomerna isporuka paketa često ima prednost nad njihovom brzom i skokovitom isporukom.

Pređimo sada s problema na načine njihovog rešavanja. Prvo ćemo izneti načelna zapažanja, a zatim razmotriti mehanizme rada protokola, organizaciju paketa i softver pomoću koga rade protokoli.

Osnovni princip koji naizust treba da znaju svi projektanti gigabitnih mreža glasi:

Pri projektovanju mislite na brzinu, a ne na propusni opseg.

Raniji protokoli su često projektovani s ciljem da smanje broj bitova na kablu, postižući to kratkim poljima pakovanim u bajtove i reči. Danas ne postoje problemi s propusnim opsegom. Problem predstavlja vreme protokolarne obrade podataka, pa treba praviti protokole koji će ga skratiti. Autori protokola IPv6 potpuno su prihvatili navedeni princip.

Privlači pažnju ideja da se brzi mrežni interfejsi izvode hardverski. Međutim, ako protokol ne bude dovoljno jednostavan, hardver će biti samo još jedna utična kartica sa sopstvenim mikroprocesorom i programom. Kada je koprocesor na kartici jeftiniji od glavnog procesora, često je i sporiji od njega. Posledica takvog rešenja je da glavni (brži) procesor uglavnom čeka da (sporiji) koprocesor obavi ključni posao. Pretpostavka da će on uporedo moći da radi i druge poslove spada u domen mitologije. Šta- više, kada međusobno komuniciraju dva mikroprocesora opšte namene, može doći do njihovog utrkivanja, pa su potrebni složeni protokoli za njihovo ispravno sinhroni- zovanje. Izlaz iz ove situacije obično je u tome da se napravi jednostavan protokol i ostavi glavnom mikroprocesoru da obavlja posao.

Razmotrimo sada problematiku povratnih informacija (engl. *feedback)* kod brzih protokola. Zbog (srazmemo) velikog kašnjenja, povratne informacije treba izbegava- ti jer signal od primaoca do pošiljaoca putuje dugo. Kao primer korišćenja povratnih informacija navodimo upravljanje brzinom slanja podataka pomoću protokola kliznih prozora. Da bi se izbegle (dugačke) zadrške skopčane sa slanjem podataka za ažuriranje prozora, bolje je koristiti neki protokol zasnovan na ograničenju brzine prenosa. Tada pošiljalac može da šalje šta god želi, samo to ne sme da šalje brže nego što je na početku dogovoreno.

Drugi primer korišćenja povratnih informacija predstavlja Jakobsonov spori algoritam. Tu se na početku isprobava koju količinu podataka mreža može da obradi. U visokobrzinskim mrežama, slanje nekoliko malih probnih paketa u cilju ispitivanja reakcije mreže, troši veliki deo propusnog opsega. Efikasnije je da tokom uspostavljanja veze pošiljalac, primalac i mreža rezervišu potrebne resurse. Takvim rezervi- sanjem može se smanjiti i neravnomernost pristizanja paketa. Ukratko, kako brzina prenosa u mreži raste, fokus projekta neizbežno se pomera ka radu sa uspostavljanjem direktne veze ili nečemu što je vrlo blisko tome. Naravno, ako u budućnosti bude propusnog opsega za bacanje, pravila projektovanja mreža mogu se potpuno preokrenuti.

Organizacija paketa predstavlja važnu stavku u gigabitnim mrežama. Zaglavlje treba da ima što manje polja da bi se skratilo vreme obrade, a ta polja treba da budu dovoljno velika da mogu da obave posao i da budu poravnata do granica reči kako bi se lakše obrađivala. „Dovoljno velika“ u ovom kontekstu znači da treba izbeći probleme vezane za ponovno lcorišcenje istih rednih brojeva, nemogućnost oglašavanja dovoljno velikog prozora itd.

Kontrolni zbir treba zasebno izračunavati za zaglavlje i zasebno za podatke, i to iz dva razloga. Prvo, zato da bi zaglavlje moglo da se posebno proveri i drugo, da se to učini pre nego što se podaci kopiraju u korisnički prostor. Poželjno je da se podaci proveravaju pomoću kontrolnog zbira tokom samog kopiranja, ali ako je zaglavlje neispravno, kopiranje će samo još više zabrljati stvar. Da bi se izbegla neispravna kopija i istovremeno omogućilo proveravanje podataka tokom kopiranja, neophodno je da se dva kontrolna zbira razdvoje.

Maksimalna količina prenetih podataka treba daje što veća da bi se obezbedio efikasan rad čak i u uslovima velikog kašnjenja. Osim toga, što je veći deo s podacima, propusni opseg se srazmerno manje troši na prenos zaglavlja. Količina od 1500 baj- tova previše je mala.

Bilo bi talcođe zgodno kada bi se zajedno sa zahtevom za uspostavljanje veze mogla poslati i neka uobičajena količina podataka. Na taj način bi se uštedela jedna šetnja mrežom.

Na kraju, treba reći nešto i o softveru pomoću koga se izvršavaju protokoli. Najvažnije je koncentrisati se na uspešan slučaj. U mnogim ranijim protokolima preterano se razmišljalo o tome šta činiti ako nešto pođe pogrešnim putem (npr. izgubi se paket). Da bi se protokol izvršavao brzo, mora se skratiti vreme svih operacija u situaciji kada sve ide kako treba. Skraćenje vremena obrade kada dođe do greške, manje je bitno.

Drugi problem koji treba da reši softver odnosi se na trajanje kopiranja. Kao što smo ranije videli, na kopiranje podataka može da ode najveći deo vremena. Idealno bi bilo da hardver prosleđuje svaki dolazni paket u memoriju i tamo ih skladišti jedan do drugog. Tada bi paket mogao da bude softverski iskopiran u korisnički bafer kao jedinstven blok podataka. U zavisnosti od toga kako radi keš, možda bi trebalo izbegavati kopiranje u petlji. Dragim recima, 1024 reči možda se može brže iskopirati pomoću 1024 uzastopne instrukcije MOVE (ili pomoću 1024 para instrukcija za učitavanje i skladištenje). Procedura za kopiranje toliko je važna daje treba ručno napisati u asembleru, osim ako se programski prevodilac može prevariti da tačno optimizuje kod.

6.7 SAŽETAK

Transportni sloj je ključ za razumevanje protokola raspoređenih po slojevima. On obezbeđuje različite usluge, od kojih je najvažnija održavanje toka bajtova s jednog na drugi kraj pouzdane, direktne veze između pošiljaoca i primaoca. Takav rad omogućavaju osnovne operacije za uspostavljanje, korišćenje i raskidanje veze. Uobičajeni interfejs transportnog sloja ostvaraje se pomoću Berkli utičnica.

Protokoli za prenos podataka moraju biti sposobni da održavaju vezu koja vodi kroz nepouzdane mreže. Uspostavljanje veze ometa postojanje zakasnelih dupliranih paketa koji se naknadno pojavljuju u najnezgodnijem trenutku. To se prilikom uspostavljanja veze prevazilazi mehanizmom trostepenog usaglašavanja. Veza se lakše raskida nego što se

uspostavlja, ali ni to nije jednostavno zbog postojanja problema dve vojske.

Čak i kada je mrežni sloj potpuno pouzdan, za transportni sloj ostaje dosta posla. On mora da radi sa svim osnovnim operacijama usluga, da održava veze i tajmere, i da dodeljuje i koristi kredite.

Na Internetu postoje dva glavna protokola za prenos podataka: UDP i TCP. Protokol UDP radi bez uspostavljanja direktne veze i obično je samo prenosilac IP paketa, ali je u stanju da multipleksira i demultipleksira više procesa služeći se jedinstvenom IP adresom. Protokol UDP se može iskoristiti za interakcije tipa klijent-server, na pri- mer, za daljinsko pozivanje procedura. Može se iskoristiti i za pravljenje protokola za rad u realnom vremenu, kao što je protokol RTP.

TCP je glavni protokol za prenos podataka na Internetu. On obezbeđuje pouzdan dvosmemi tok bajtova. Za sve segmente koristi 20-bajtno zaglavlje. Usmerivači mogu da fragmentiraju segmente, pa računari moraju znati kako da ih ponovo sklope. Mnogo je truda uloženo u optimizovanje performansi protokola TCP, između ostalog i pomoću Nagleovog, Klarkovog, Jakobsonovog, Karnovog i drugih algoritama. Bežične veze donose niz novih komplikacija. Transakcioni TCP predstavlja proširenje protokola TCP, koje u klijentsko-serverskim interakcijama radi sa smanjenim brojem paketa.

Performanse mreže najčešće zavise od funkcionisanja protokola i trajanja obrade TPDU blokova, i to je sve uočljivije što je prenos brži. Pri projektovanju protokola treba težiti da se minimizuju broj razmenjenih TPDU blokova, broj prelazaka iz jednog konteksta u drugi, i vreme kopiranja svakog TPDU bloka. Za gigabitne mreže potrebni su jednostavni protokoli.

ZADACI

1. U našem primeru osnovnih operacija transportnih usluga na slici 6-2, operacija LISTEN izaziva blokiranje. Da li je to stvarno neophodno? Ako mislite da nije, objasnite kako bi se mogla koristiti operacija koja ne izaziva blokiranje. Kakvu bi prednost ona imala u odnosu na postupak opisan u tekstu?
2. U modelu koji opisuje slika 6-4, pretpostavlja se da se paketi mogu izgubiti u mrežnom sloju i da se zato moraju pojedinačno potvrđivati. Pretpostavite da je mrežni sloj stoprocentno pouzdan i da nikada ne gubi pakete. Da li to menja sliku 6-4 i - ako je menja - kako?
3. Na obe polovine slike 6-6 napominje se da vrednost promenljive *SERVER_PORT* mora da bude ista kod klijenta i kod servera. Zašto je to tako važno?
4. Zamislite generator početnih rednih brojeva za čije se pokretanje koristi sistemski sat sa 15-bitnim brojačem. Sat otkucava svakih 100 ms, a maksimalan životni vek paketa je 60 s. Izračunajte posle kog vremena se sat mora ponovo sinhronizovati
   a) u najnepovoljnijem slučaju.
   b) u situaciji kada podaci troše 240 rednih brojeva u minutu.
5. Zašto maksimalan životni vek paketa, *T,* mora da bude dovoljno dugačak da u međuvremenu nestanu ne samo svi paketi, već i potvrde za njih?
6. Zamislite da se za uspostavljanje veze ne koristi trostepeno, već dvostepeno usaglašavanje. Drugim recima, treća poruka nije obavezna. Da li je u tom slučaju moguća kružna blokada? Ponudite primer ili pokažite da to nije moguće.
7. Zamislite uopšten problem *n* vojski, u kome je za pobedu dovoljan međusobni dogovor

bilo koje dve grupe plavih vojnika. Postoji li protokol koji bi omogućio pobedu plavim vojnicima?

8. Razmotrite problem oporavljanja od pada računara (slika 6-18). Ako se interval između upisivanja i slanja potvrde (ili obrnuto) može učiniti srazmerno kratkim, koje dve najbolje strategije mogu da primene pošiljalac i primalac da šansa otkazivanja protokola bude minimalna?
9. Da lije moguća kružna blokada s transportnom jedinicom opisanom u tekstu (slika **6-20)?**
10. Realizator transportne jedinice sa slike 6-20 iz radoznalosti je odlučio da u proceduru *sleep* postavi brojače i da tako prikuplja statistiku o nizu *conn*. Između ostalog, pratio je i broj veza koje se nalaze u jednom od sedam mogućih stanja «,■ (;' = 1,..., 7). Postoje napisao opsežan program u FORTRAN-u da bi analizirao statističke podatke, naš realizator je otkrio da izgleda uvek važi relacija = *MAX_CONN*. Postoje li i neke druge invarijante koje obuhvataju samo ovih sedam promenljivih?
11. Šta se dešava kada korisnik transportne jedinice sa slike 6-20 pošalje poruku nulte dužine? Kakav značaj ima vaš odgovor?
12. Za svaki moguć događaj u transportnoj jedinici sa slike 6-20, odgovorite da li je i legalan kada korisnik spava u stanju *sending*.
13. Objasnite prednosti i nedostatke upotrebe kredita u odnosu na protokole kliznih prozora.
14. Čemu protokol UDP? Zar ne bi bilo dovoljno dopustiti korisničkim procesima da šalju sirove IP pakete?
15. Razmotrite jednostavan protokol na nivou aplikacije koji preko protokola UDP omogućava klijentu da sa udaljenog servera na opštepoznatoj adresi preuzima datoteku. Klijent šalje zahtev sa imenom datoteke, a server odgovara nizom paketa s podacima koji sadrže različite delove tražene datoteke. U cilju pouzdanosti i očuvanja redosleda paketa, klijent i server koriste protokol „stani i čekaj“. Ako zanemarimo loše performanse, sagledavate li još neki problem sa ovim protokolom? Obratite pažnju na mogućnost zaglavljivanja procesa.
16. Preko optičkog kabla brzine 1 Gb/s, klijent šalje 128-bajtni zahtev serveru udaljenom 100 km. Izračunajte efikasnost rada linije tokom daljinskog pozivanja procedure.
17. Razmotrite ponovo situaciju iz prethodnog zadatka. Izračunajte minimalno moguće vreme odgovora za liniju brzine 1 Gb/s, kao i za liniju brzine 1 Mb/s. Kakav zaključak se nameće?
18. Pri slanju poruka, u protokolima UDP i TCP koriste se brojevi priključaka za identifikovanje odredišne transportne jedinice. Ponudite dva razloga zbog kojih je u ove protokole uveden nov apstraktni identifikator (broj priključka), umesto da se koristi identifikator procesa koji je već postojao u vreme njihovog nastanka.
19. Kolika je ukupna veličina minimalne MTU u protokolu TCP, uključujući TCP i IP zaglavlje, ali bez zaglavlja sloja veze podataka?
20. Fragmentiranje i ponovno sklapanje datagrama obavlja protokol IP, skrivajući to od protokola TCP. Znači li to da protokol TCP ne treba da brine o ispravnom redosledu stizanja podataka?
21. Protokol RTP se koristi za prenos zvuka CD kvaliteta - po jedan 16-bitni uzorak za svaki od dva stereo kanala, 44.100 puta u sekundi. Koliko paketa svake sekunde treba da prenese protokol RTP?
22. Da li bi bilo moguće smestiti RTP kod u jezgro operativnog sistema, zajedno sa UDP kodom? Obrazložite odgovor.

23. Procesu na računtaru 1 dodeljen je priključak *p,* a procesu na računaru 2 priključak *q.* Da li se između ovih priključaka može istovremeno uspostaviti još TCP veza?
24. Na slici 6-29 vidimo da, osim 32-bitnog polja *Broj potvrde,* u četvrtoj reči postoji i bit *ACK.* Da li on stvarno služi nečemu? Zastoje on tu?
25. Maksimalan koristan teret TCP segmenta iznosi 65.495 bajtova. Zašto je izabran ovako neobičan broj?
26. Opišite dva načina za stizanje u stanje *SYNRCVD* sa slike 6-33.
27. Navedite potencijalan nedostatak Nagleovog algoritma u gadno zagušenoj mreži.
28. Razmotrite efekat primene sporog algoritma na liniji s vremenom obilaska 10 ms i bez zagušenja. Primalac ima prozor veličine 24 KB, a najveći segment je 2 KB. Koliko će vremena proteći dok se ne pošalje prvi pun prozor podataka?
29. Pretpostavite daje TCP prozor zagušenja podešen na 18 KB i da se tajmer automatski isključio. Na koju vrednost će se povećati prozor ako naredna četiri poslata rafala budu uspešno primljena? Pretpostavite daje 1 KB maksimalna veličina segmenta.
30. Ako vreme obilaska TCP veze (*RTT)* trenutno iznosi 30 ms, a potvrde stignu posle 26, 32 i 24 ms, kako će Jakobsonov algoritam proceniti novu vrednost *RTT!* Za *a* upotrebite vrednost 0,9.
31. TCP računar šalje pune prozore podataka veličine 65.535 bajtova kanalom brzine 1 Gb/s u kome je kašnjenje u jednom smeru 10 ms. Koliki je najveći moguć protok podataka? Kolika je efikasnost rada linije?
32. Koliko brzo bez reciklovanja rednih brojeva može da radi linija u koju računar uzastopno šalje koristan TCP teret od 1500 bajtova, a maksimalan životni vek paketa je 120 s? Uzmite u obzir višak koji unose TCP, IP i Ethernet zaglavlja. Pretpostavite da se Ethernet okviri mogu slati kao neprekidan tok podataka.
33. Kolika je maksimalna brzina prenosa podataka po vezi u mreži u kojoj maksimalna veličina TPDU bloka iznosi 128 bajtova, maksimalan životni vek TPDU blokova je 30 s, a redni brojevi su 8-bitni?
34. Pretpostavimo da merite vreme primanja TPDU bloka. U trenutku kada nastane sistemski prekid, očitavate sat u milisekundama. Pošto TPDU blok bude u potpunosti obrađen, očitavate sat ponovo. Za trajanje obrade dobijate 270.000 puta nulu i 730.0 puta jedinicu. Koliko stvarno traje obrada TPDU bloka?
35. Mikroprocesor izvršava instrukcije brzinom 1000 MIPS. U jednom trenutku može se kopirati 64 bita podataka, a za svaku kopiranu reč troši se 10 instrukcija. Ako dolazni paket treba kopirati četiri puta, može li ovaj sistem da obrađuje liniju brzine 1 Gb/s? Pretpostavite, zbog jednostavnosti, da se sve instrukcije, pa i one za rad s memorijom, odvijaju brzinom 1000 MIPS.
36. Problem kratkog perioda reciklovanja rednih brojeva može se otkloniti ako se upotrebe 64-bitni redni brojevi. Međutim, optičko vlakno teorijski može da radi brzinom 75 Tb/s. Koliki maksimalan životni vek paketa treba predvideti da buduće mreže, brzine 75 Tb/s, ne bi imale problema s reciklovanjem čak i 64-bitnih rednih brojeva? Pretpostavite da svaki bajt ima svoj redni broj, kao u protokolu TCP.
37. Navedite jednu prednost daljinskog pozivanja procedura (RPC) pomoću protokola UDP, u odnosu na transakcioni protokol TCP. Navedite ijednu prednost protokola T/TCP nad protokolom RPC.
38. Na slici 6-40(a) vidimo da je za daljinsko pozivanje procedure (RPC) potrebno razmeniti 9 paketa. Postoji li situacija u kojoj treba razmeniti tačno 10 paketa?
39. U odeljku 6.6.5 izračunali smo da gigabitna linija sručuje računani 80.000 paketa u sekundi, ostavljajući mu samo 6250 instrukcija za njihovu obradu, pod uslovom da se

polovinom procesorskog vremena služe druge aplikacije. Tada smo pretpostavili daje paket veličine 1500 bajtova. Ponovite izračunavanje za ARPANET paket veličine 128 bajtova. U oba slučaja pretpostavite da zadata veličina paketa obuhvata i zaglavlja.

40. Za gigabitnu mrežu koja radi u području veličine 4000 km, ograničavajući faktor nije propusni opseg, već kašnjenje. Zamislite gradsku mrežu u kojoj prosečno rastojanje između izvorišta i odredišta iznosi 20 km. Pri kojoj brzini prenosa podataka, kašnjenje obilaska izazvano brzinom svetlosti postaje jednako kašnjenju paketa veličine 1 KB?
41. Izračunajte proizvod propusnog opsega i vremena obilaska za sledeće mreže: (1) TI (1,5 Mb/s), (2) Ethernet (10 Mb/s), (3) T3 (45 Mb/s) i (4) STS-3 (155 Mb/s). *ZaRTT* uzmite vrednost 100 ms. Setite se da TCP zaglavlje sadrži 16 bitova rezervisanih za veličinu prozora. Kako to utiče na vaš proračun?
42. Koliki je proizvod opsega i kašnjenja za kanal brzine 50 Mb/s koji se koristi za geo-stacionarni satelit? Ako su svi paketi (zajedno sa zaglavljem) veličine 1500 bajtova, koliku veličinu prozora treba naznačiti u paketima?
43. Server datoteka, prikazan na slici 6-6, daleko je od toga da bude savršen i mogao bi se poboljšati na više načina. Uvedite sledeće izmene:
    a) Dodajte klijentu i treći argument kojim se zadaje opseg bajtova.
    b) Dodajte klijentu indikator -w koji dozvoljava upisivanje datoteke na server.
44. Izmenite program sa slike 6-20 tako da omogući oporavljanje od grešaka. Dodajte nov tip paketa *reset,* koji može da stigne nakon što su obe strane otvorile vezu, a nijedna je nije zatvorila. Taj događaj, koji istovremeno nastupa na oba kraja veze, znači da su svi paketi koji se trenutno nalaze na vezi ili isporučeni ili uništeni, ali se nijedan od njih više ne nalazi u podmreži.
45. Napišite program koji simulira rad s baferima u transportnoj jedinici i za kontrolu toka ne koristi sistem kredita sa slike 6-20, već protokol kliznih prozora. Neka procesi viših slojeva nasumično uspostavljaju veze, šalju podatke i raskidaju veze. Da bi program bio jednostavniji, neka svi podaci teku od računara *A* ka računaru *B*. Uporedite različite strategije dodeljivanja bafera na računaru *B,* npr. namenjivanje bafera određenim vezama i dodeljivanje po potrebi iz skupa zajedničkih bafera, i iz- merite ukupno postignut protok podataka u svakom od slučajeva.
46. Projektujte i realizujte sistem za međusobno ćaskanje više grupa korisnika. Koordinator sesije nalazi se na opštepoznatoj mrežnoj adresi, koristi protokol UDP za komunikaciju sa ostalim klijentima, uspostavlja server za svaku sesiju i održava katalog sesija. Za svaku sesiju postoji zaseban server. Server komunicira s klijentima pomoću protokola TCP. Klijent korisnicima omogućava da započnu sesiju, da se pridruže već otvorenoj sesiji i da napuste sesiju. Napišite i ugradite kod koordinatora, servera i klijenta.

V

SLOJ APLIKACIJA

Pošto smo završili sve pripremne radnje, došli smo do sloja u kome se nalaze sve aplikacije. Slojevi ispod sloja aplikacija obezbeđuju pouzdan transport, ali korisnik nema utisak da oni nešto stvarno rade. U ovom poglavlju razmotrićemo nekoliko pravih mrežnih aplikacija.

Međutim, čak i u sloju aplikacija postoji potreba za protokolarnom podrškom kako bi aplikacije mogle da rade. Shodno tome, razmotrićemo jedan protokol pre nego što se upoznamo sa aplikacijama. Protokol se popularno zove DNS i radi sa imenima na Internetu. Posle njega, preći ćemo na tri prave aplikacije: elektronsku poštu, Web i multimediju.

7.1 DNS - SISTEM IMENOVANJA DOMENA

Iako se programi teoretski mogu obraćati računalima, poštanskim sandučićima i drugim resursima preko njihovih mrežnih (tj. IP) adresa, te adrese ljudi teško pamte. Isto tako, ako šaljete poruku na elektronsku adresu *jana@ 128.111.24.41,* pa Janin davalac Internet usluga ili njena radna organizacija premeste poštanski server na drugi računar s drugačijom IP adresom, to znači da ćete poruku sledeći put morati da šaljete na novu adresu. Zbog toga su uvedena tekstualna imena kojima se ime računara razdvaja od njegove adrese. Na taj način, Janina adresa mogla bi izgledati približno ovako: jana@art.ucsb.edu. Bez obzira na to, sama mreža razume samo brojčane adrese, tako daje neophodan mehanizam kojim će se tekstualna imena prevoditi u mrežne adrese. U narednim odeljcima proučićemo kako se ovo preslikavanje radi na Internetu.

Nekada davno, u doba ARPANET-a, postojala je datoteka *host.txt* s listom imena svih računara i njihovih IP adresa. Svi računari su je svake noći morali preuzimati s lokacije na kojoj je održavana. Za mrežu od nekoliko stotina računara koji su radili s podelom procesorskog vremena, ovakav pristup bio je sasvim zadovoljavajući.

Međutim, kada su se na mrežu priključile hiljade mini i PC računara, postalo je jasno da se takav pristup ne može večno održati. Prvo, datoteka bi postala ogromna. Drugo i važnije: ako se sa imenima ne bi radilo na jednom mestu, stalno bi dolazilo do njihovog sukobljavanja, što se zbog potencijalnog preopterećenja i kašnjenja ne bi smelo ni zamisliti na velikoj međunarodnoj mreži. U cilju rešavanja ovih problema, stvorenje sistem imenovanja domena (engl. *Domain Name System., DNS).*

Suštinu DNS-a čini hijerarhijska struktura imena zasnovana na domenima i sistem distribuiranih baza podataka za realizaciju te hijerarhijske strukture. On se prvenstveno koristi za preslikavanje imena računara i odredišta elektronske pošte u IP adrese, ali se može iskoristiti i za druge svrhe. DNS je definisan u RFC dokumentima 1034 i 1035.

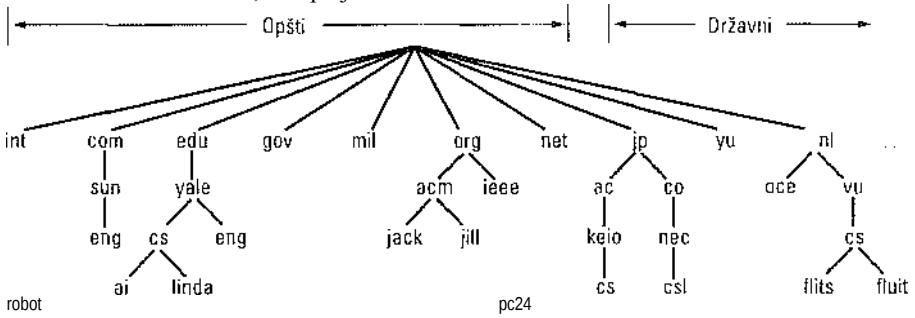
DNS u najopštijim crtama radi na sledeći način. Da bi preslikala ime u IP adresu, aplikacija (iz biblioteke) poziva proceduru razrešivač (engl. *resolver),* prosleđujući joj kao parametar ime. Primer razrešivača *gethostbyname* videli smo na slici 6-6. Razrešivač šalje UDP paket lokalnom DNS servera, koji traži ime i vraća razrešivaču odgovarajuću IP adresu, a razrešivač je vraća pozivaocu. Kada ima IP adresu, aplikacija može da uspostavi TCP vezu sa odredištem ili da mu pošalje UDP pakete.

7.1.1 Imenski DNS prostor

Rad s velikim skupom imena koja se stalno menjaju nije jednostavan. U poštanskom sistemu, sa imenima se radi tako što se zahteva da svako pismo sadrži (doslovce ili podrazumevano) ime države ili pokrajine, ime grada, ime ulice i broj kuće primaoca. Kada se upotrebi takva hijerarhija adresa, onda ne može da nastane zabuna između Petra Petrovića sa stanom u Beogradskoj ulici u Nišu i Petra Petrovića koji stanuje u Beogradskoj ulici u Beogradu. DNS radi na isti način.

Internet je organizaciono podeljen na preko 200 osnovnih domena (engl. *top-le- vel domains),* pri čemu svaki domen obuhvata brojne računare. Svaki domen je sa svoje strane izdeljen u poddomene, a ovi u pod-poddomene itd. Svi domeni se mogu predstaviti stablom, kao na slici 7-1. Listovi stabla predstavljaju domene koji nemaju podđomena (ali, naravno, imaju računare). List može da sadrži samo jedan računar ili može da predstavlja kompaniju s hiljadama računara.

Osnovni domeni imaju opšta imena (prema delatnostima) ili nose imena država. Prvobitni opšti osnovni domeni bili su *com* (engl. commercial - trgovina), *edu* (engl. educational institutions - obrazovanje), *gov* (engl. U.S. Federal Government - Američka savezna vlada), *int* (engl. certain international organizations - neke međunarodne organizacije), *mil* (engl. U.S. *Military* Forces - Oružane snage SAD), *net* (engl. network providers - davaoci mrežnih usluga) i *org* (engl. nonprofit organizations - neprofitne organizacije). Državnih domena, prema ISO standardu 3166, ima po jedan za svaku državu.



Slika 7-1. Deo imenskog DNS prostora na Internetu.

Novembra 2000. godine organizacija ICANN je odobrila četiri nova osnovna domena opšte namene: *biz* (engl. business - *„biznis“,* poslovi), *info* (engl. *information* - informacije), *name* (engl. people's *names* - imena ljudi) i *pro* (engl. professions - profesije: lekari, advokati). Osim toga, na zahtev izvesnih industrijskih grupacija uvedena su i tri specijalizovana osnovna domena: *aero* (engl. aerospace industry - vazduhoplovna industrija), *coop* (engl. eo-operatives - kooperacije) i *museum* (muzeji). Osnovni domeni biće dodavani i ubuduće.

Recimo uzgred daje Internet s porastom komercijalnog aspekta sve više postajao megdan za međusobna obračunavanja. Uzmite, na primer, domen *pro.* On je name- njen ovlašćenim profesionalcima. Ali, ko predstavlja profesionalce i ko ih ovlašćuje? Lekari i advokati su nesumnjivo profesionalci, ali šta reći o „slobodnim“ umetnicima: fotografima, učiteljima klavira, mađioničarima, majstorima svake vrste, bricama, istrebljivačima gamadi,

umetnicima za tetoviranje, plaćenicima i prostitutkama? Spadaju li ova zanimanja u profesije, i imaju li pravo na domen pro? Ako imaju, ko izdaje ovlašćenja pripadnicima tih profesija?

Dobiti domen drugog nivoa, npr. *ime_kompanije.com,* u načelu nije teško. Treba se samo obratiti registratora („matičaru“) osnovnog domena (u ovom slučaju, domena *com*) koji će proveriti da lije traženo ime raspoloživo i da li možda ne predstavlja zaštitni znak neke druge kompanije. Ako je u tom pogledu sve u redu, podnosilac zah- teva dobija ime za malu godišnju naknadu. Do ovog trenutka, izgleda daje u domenu *com* iskorišćena svaka normalna (engleska) reč. Proverite sa imenima kućnih aparata, životinja, biljaka, delova tela itd. Skoro sva su tu.

Svaki domen se označava uzlaznom putanjom do (neimenovanog) korena stabla. Komponente imena međusobno se razdvajaju tačkama. Tako, inženjersko odeljenje korporacije Sun Microsystems moglo bi imati domen *eng.sun.com,* umesto domena/ *com/sun/eng,* kakav bi bio u IJNIX-u. Obratite pažnju na to da ovakvo hijerarhijsko imenovanje domena omogućava da se domen *eng.sun.com* ne sukobljava s domenom *eng.yale.edu,* koji bi mogla imati Katedra za engleski jezik univerziteta Yale.

Imena domena mogu da budu apsolutna ili relativna. Za razliku od relativnog imena, apsolutno ime domena uvek se završava tačkom (npr. *eng.sun.com.).* Relativna imena moraju se tumačiti u okviru određenog konteksta da bi se jedinstveno utvrdilo njihovo značenje. U oba slučaja, imenovani domen se odnosi na određeni čvor stabla i na sve čvorove ispod njega.

U imenima domena ne pravi se razlika između velikih i malih slova, pa *edu, Edu* i *EDU* znače isto. Komponente imena mogu imati najviše po 63 znalca, a celo ime putanje ne sme da pređe 255 znakova.

Domeni se u načelu mogu umetati u stablo na dva načina. Na primer, domen *cs.ya- le.edu* mogao bi se kao *cs.yale.edu.us* isto tako dobro svrstati u državni domen *us.* U praksi, međutim, organizacije su u SAD uglavnom svrstane u opšte domene, a većina organizacija izvan SAD - u odgovarajuće državne domene. Nije zabranjeno biti istovremeno u dva osnovna domena, ali to uglavnom rade samo multinacionalne kompanije (npr. *sony.com* i *sony.nl).*

Svaki domen slobodno imenuje domene ispod sebe. Na primer, Japan ima domene *ac.jp* i *co.jp* koji odgovaraju opstim domenima *edu* i *com.* Holandija prelazi preko te razlike i sve poddomene direktno smesta u domen *ni.* Tako, sva tri sledeća domena obuhvataju katedre za računarstvo na različitim univerzitetima:

1. - *cs.yale.edu* (Univerzitet Yale u SAD)
2. - *cs.vu.nl* (Univerzitet Vrije, u Holandiji)
3. - *cs.keio.ac.jp* (Univerzitet Keio u Japanu)

Kada želite da napravite domen, potrebna vam je dozvola (hijerarhijski višeg) domena u koji želite da ga uključite. Na primer, ako je na univerzitetu Yale pokrenuta grupa za projektovanje integrisanih kola koja želi domen *vlsi.cs.yale.edu,* ona mora da dobije dozvolu od administratora domena *cs.yale.edu.* Slično tome, ako se na severe Južne Dakote otvori nov univerzitet (University of Northern South Dakota), on mora da od administratora domena *edu* zahteva dodeljivanje domena *unsd.edu.* Na taj način izbegavaju se sukobljavanja oko imena, a svaki domen može da vodi evidenciju svojih poddomena. Kada univerzitet iz gornjeg

primera dobije i registruje svoj domen, može u njemu po volji praviti poddomene, npr. poddomen *cs.unsd.edu.*

Imenovanje domena ne vezuje se za fizičke mreže, već sledi organizacione krite- rijume. Na primer, Katedra za računarstvo i Katedra za elektrotehniku mogu se nalaziti u istoj zgradi, čak deliti i istu lokalnu mrežu, pa ipak imati različite domene. I obratno, uprkos tome što Katedra za računarstvo može imati svoje prostorije i u Beogradu i u Nišu, računari iz oba odeljenja normalno će pripadati istom domenu.

7.1.2 Zapisi resursa

Svakom domenu, bez obzira na to da li sadrži samo jedan računar ili je u pitanju osnovni domen, može se pridružiti skup zapisa resursa (engl. *resource records)*. Za jedan računar, najčešći zapis resursa je njegova IP adresa, ali postoje i brojne druge vrste zapisa resursa. Kada razrešivač DNS serveru preda ime domena, od njega dobija zapise resursa pridružene tom imenu. Na taj način, osnovna funkcija DNS sistema jeste mapiranje (preslikavanje) imena domena u zapise resursa.

Zapis resursa je sastavljen od pet dubleta. Iako se oni zbog efikasnosti kodiraju binarno, obično se prikazuju tekstualno - po jedan red za svaki zapis. Mi ćemo koristiti sledeći format:

Ime_domena Životni_vek KlasaTip Vrednost

Ime_domena je domen za koji važi zapis. Obično za svaki domen postoji mnogo zapisa, a svaka kopija baze podataka sadrži informacije o više domena. Zato je ovo polje primarni ključ za pretraživanje zapisa pri odgovaranju na zahteve. Redosled zapisa u bazi podataka nije važan.

Životni_vek govori o stabilnosti zapisa. Vrlo stabilnim informacijama dodeljuje se velika vrednost, npr. 86400 (broj sekundi u jednom danu), a informacijama privremenog karaktera, mala, npr. 60 (jedan minut). Vratićemo se ponovo na životni vek pošto budemo obradili keširanje.

Treće polje svakog zapisa resursa jeste *Klasa.* Za podatke na Internetu, ono je uvek *IN.* Za podatke izvan Interneta mogu se koristiti drugi kodovi, ali se na njih retko nailazi u praksi.

Poljem *Tip* označava se vrsta zapisa. Najvažniji tipovi prikazani su na slici 7-2.

Tip	Značenje	Vrednost
SOA	Početak pouzdanih informacija	Parametri za aktuelnu zonu
A	IP adresa računara	32-bitni ceo broj
MX	Razmena pošte	Prioritetni domen koji je voljan da primi e-poštu
NS	Server imena	Ime servera za aktuelan domen
CNAME	Kanoničko ime	Ime domena
PTR	Pokazivač	Alijas IP adrese
HINFO	Opis računara	Mikroprocesor i operativni sistem (tekstualno)
TXT	Tekst	Nepreveden ASCII tekst

Slika 7-2. Glavni tipovi DNS zapisa resursa za IPv4.

Zapis tipa *SOA* sadrži ime osnovnog izvora informacija za zonu servera imena (opisanu u nastavku), elektronsku adresu njenog administratora, jedinstven serijski broj i različite

indikatore i rokove isključivanja tajmera.

Najvažniji su zapisi tipa A (adresni zapisi). Takav zapis sadrži 32-bitnu IP adresu nekog računara. Svaki računar na Internetu mora da ima barem jednu IP adresu preko koje komunicira s dragim računarima. Neki računari imaju dva i više priključaka na mrežu, pa zato imaju po jedan zapis *A* za svaki priključak (tj. za svaku IP adresu). DNS se može podesiti da ciklično bira ove adrese odgovarajući prvom adresom na prvi zahtev, dragom na drugi itd.

Sledeći zapis, *MX,* isto je tako važan. On sadrži ime računara spremnog da prihvati e-poštu za određeni domen. Zapis je potreban jer nije svaki računar pripremljen za primanje e-pošte. Ako neko, na primer, želi da pošalje elektronsku poruku korisniku bora@mikroknjiga.co.yu, pošiljalac mora da pronađe poštanski server domena *mikroknjiga.co.yu* koji je voljan da prihvati poruku. Zapis MX pruža te informacije.

Zapis *NS* nabraja servere imena. Na primer, svaka DNS baza podataka normalno ima po jedan zapis *NS* za svaki osnovni domen, tako da se, na primer, e-pošta može slati u udaljene delove stabla imena. Na ovo ćemo se kasnije ponovo vratiti.

Zapis *CNAME* omogućava pravljenje alijasa. Na primer, neko ko u opštim crtama poznaje sistem imenovanja na Internetu, može svom kolegi Pavlu koji radi na Katedri za računarske nauke Masačusetskog tehničkog instituta da pošalje poruku pogađajući adresu pavle@cs.mit.edu. Taj pokušaj bi propao jer je domen rečene Katedre u stvari *lcs.mit.edu.* Međutim, za korisnike koji to ne znaju, Masačusetski institut bi mogao da napravi zapis CNAME koji će korisnike i programe upućivati u pravom smeru. To bi se moglo izvesti pomoću sledeće odrednice:

```
cs.mit.edu 86400 IN CNAME lcs.mit.edu
```

Slično zapisu *CNAME* i zapis *PTR* ukazuje na drugo ime. Međutim, za razliku od zapisa CNAME, koji je u stvari definicija makroa, zapis PTR predstavlja regularan tip DNS podataka čije tumačenje zavisi od konteksta u kome je nađen. On se u praksi gotovo uvek koristi za pridruživanje imena IP adresi, tako da se na osnovu IP adrese može naći ime računara. To se zove obrnuto pretraživanje (engl. *reverse lookup).*

Zapis *H1NFO* omogućava korisnicima da saznaju kakvom računara i operativnom sistemu odgovara domen i na kraju, zapis *TXT* omogućava domenu da se predstavi na proizvoljan način. Zapisi ove dve vrste postoje radi korisnika. Oni nisu obavezni, tako da programi ne treba da računaju na njih (jer verovatno ne bi ni mogli da ih protumače ako bi uopšte uspeli da im pristupe).

Na kraju imamo polje *Vrednost.* Vrednost može da bude brojčana, da predstavlja ime domena ili da bude tekst. Semantika zavisi od tipa zapisa. Kratak opis polja *Vrednost* za svaki od glavnih tipova zapisa dat je na slici 7-2.

Na slici 7-3 prikazan je primer informacija o domenu koji možete da nađete u DNS bazi podataka. Upotrebljena je (poluhipotetička) baza podataka za domen *cs.vu.nl* sa slike 7-1. Baza sadrži sedam tipova zapisa resursa.

Prvi red posle komentara na slici 7-3 pruža izvesne osnovne informacije o domenu koje nas neće dalje zanimati. U sledeća dva reda teksta opisuje se mesto gde se domen nalazi. Naredni redovi ukazuju na prvo, odnosno drugo mesto na kojima treba pokušati isporuku e-pošte upućene na adresu korisnik@cs.vu.nl. Treba prvo probati sa specijalnim računarom *zephyr.* Ako to ne uspe, probati računar *top.*

```
; Authoritative data for cs.vu.nl
cs.vu.nl.       86400   IN  SOA     star boss
cs.vu.nl.       86400   IN  TXT     "Divisie Wiskunde en Informatica."
cs.vu.nl.       86400   IN  TXT     "Vrije Universiteit Amsterdam."
cs.vu.nl.       86400   IN  MX 1    zephyr.cs.vu.nl.
cs.vu.nl.       86400   IN  MX      2 top.cs.vu.nl.

flits.cs.vu.nl. 86400   IN  HINFO   Sun Unix
flits.cs.vu.nl. 86400   IN  A       130.37.16.112
flits.cs.vu.nl. 86400   IN  A       192.31.231.165
flits.cs.vu.nl. 86400   IN  MX      1 flits.cs.vu.nl.
flits.cs.vu.nl. 86400   IN  MX      2 zephyr.cs.vu.nl.
flits.cs.vu.nl. 86400   IN  MX      3 top.cs.vu.nl.

l.              86400   IN  CNAME   star.cs.vu.nl

ftp.cs.vu.nl.   86400   IN  CNAME   zephyr.cs.vu.nl

rowboat                 IN  A       130.37.56.201
                        IN  MX      1 rowboat
                        IN  MX      2 zephyr
                        IN  HINFO   Sun Unix

little-sister           IN  A       130.37.62.23
                        IN  HINFO   Mac MacOS

laserjet                IN  A       192.31.231.216
                        IN  HINFO   "HP Laserjet IIISi" Proprietary
```

Slika 7-3. Deo moguće DNS baze podataka za dornen *cs.vu.nl.*

Posle praznog reda, dodatog zbog bolje čitljivosti, dolaze redovi koji saopštavaju da j*e flits* radna stanica Sun sa operativnim sistemom UNIX i daju se obe njene IP adrese. Zatim se nude tri opcije za rukovanje e-poštom poslatom računam *flits.cs.vu.nl.* Prva je, prirodno, sam *računar flits*. Ako je on isključen, draga i treća opcija su računari *zephyr* i *top.* Zatim sledi alijas www.cs.vu.nl koji se može koristiti bez naznake određenog računara. Ovaj alijas omogućava domenu *cs. vu.nl* da promeni Web server, a da pri tome ne menja njegovu adresu. Slična argumentacija važi i za alijas *ftp. cs.vu.'nl.*

Sledeća četiri reda predstavljaju tipičnu odrednicu za radnu stanicu, u ovom slučaju, za stanicu *rowboat.cs.vu.nl.* Tu se nalaze IP adresa, primarni i sekundarni računar za isporuku pošte, i podaci o računam. Zatim dolazi odrednica za sistem koji ne radi pod UNIX-om i ne može da prima poštu, i na kraju, odrednica za laserski štampač koji je priključen na Internet.

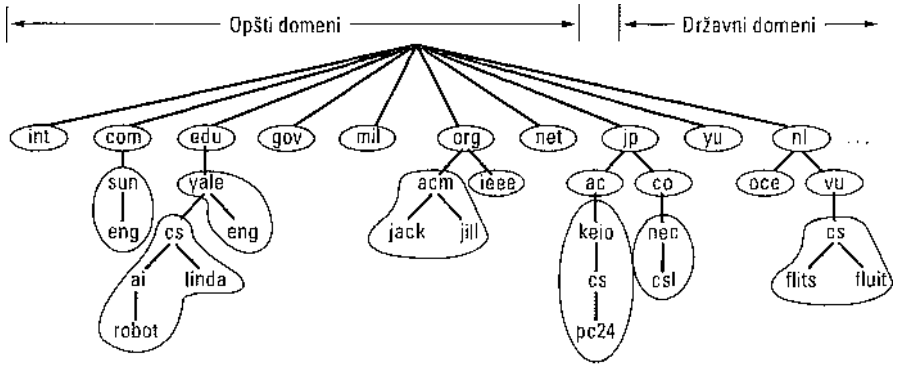
Nisu prikazane (a i ne nalaze se u ovoj datoteci) IP adrese koje se koriste za pretraživanje osnovnih domena. One su neophodne za pronalaženje udaljenih računara, ali pošto ti računari ne pripadaju domenu *cs.vu.nl,* adrese se ne nalaze u ovoj datoteci. Te adrese čuvaju tzv. korenski serveri (engl. *root setyers),* čije adrese postoje u kon- figuracionoj datoteci sistema i učitavaju se u DNS keš svaki put kada se pokrene DNS server. Širom sveta postoji desetak korenskih servera i svaki od njih zna IP adrese svih servera osnovnih domena (engl. *top-level*

domain servers). Na taj način, ako računar zna IP adresu barem jednog korenskog servera, on može da pronađe bilo koje DNS ime.

7.1.3 Serveri imena

Teorijski bi jedan jedini server imena mogao da sadrži čitavu DNS bazu podataka i da zadovolji sve zahteve u pogledu imena računara. Takav server bi, međutim, bio toliko opterećen da bi praktično bio beskorisan. Osim toga, ako bi ikada otkazao, čitav Internet bi se srušio.

Da bi se izbegli problemi koji proističu iz postojanja samo jednog izvora informacija, imenski DNS prostor podeljen je u više zona (engl. *zone)* koje se međusobno ne preklapaju. Jedna mogućnost deljenja imenskog prostora sa slike 7-1 prikazana je na slici 7-4. Svaka zona sadrži određeni deo stabla, kao i servere imena sa informacijama o dotičnoj zoni. Obično svaka zona ima primarni server imena koji informacije izvlači iz datoteke na svom disku i jedan ili više sekundarnih servera imena koji informacije dobijaju od primarnog servera imena. Radi veće pouzdanosti, neki serveri mogu da se nalaze i izvan zone za koju rade.



Slika 7-4. Deo imenskog DNS prostora, podeljen u zone.

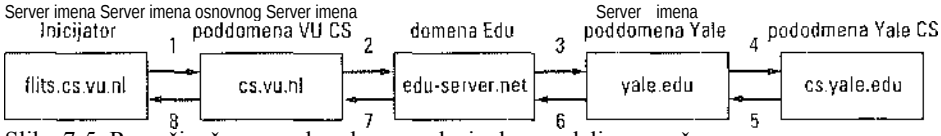
Gde se unutar zona postavljaju granice zone zavisi od njenog administratora. Ta odluka se donosi na osnovu broja servera imena i njihovog razmeštaja. Na primer, na slici 7- 4, Yale ima server za domen *yale.edu* koji obavlja poslove i za domen *eng.yale.edu,* ali ne i za domen *cs.yale.edu* koji predstavlja zasebnu zonu sa sopstvenim serverom imena. Takva odluka može se doneti ako, na primer, Katedra za engleski jezik ne želi da održava sopstveni server imena, a Katedra za računarstvo upravo to želi. Shodno tome, domen *cs.yale.edu* predstavlja zasebnu zonu, ali ne i domen *eng.yale.edu.*

Kada razrešivač dobije upit za ime domena, on ga prosleđuje nekom od lokalnih servera imena. Ako traženi domen spada u delokrug servera imena, kao što domen *ai.cs.yale.edu* potpada pod domen cs.yale.edu, server vraća meritorne zapise resursa.

Meritoran zapis (engl *authoritative record)* potiče iz autoritativnog izvora koji upravlja zapisima i zato je uvele ispravan. Meritorni zapisi su uvele ažurni, za razliku od keširanih

zapisa koji mogu i zastareti.

Ako je, međutim, domen udaljen i o njemu nema lokalnih informacija, server imena šalje poruku sa upitom serveru imena osnovnog domena kome pripada traženi domen. Postupak c'ete lakše razumeti pomoću slike 7-5. Tu, razrešivač na računaru *flits.cs.vu.nl* želi da sazna IP adresu računara *linda.cs.yale.edu.* U prvom koraku, on šalje upit lokalnom serveru imena, računaru *cs.vu.nl.* Upit sadrži ime traženog domena, tip *(A)* i klasu *(IN).*



Slika 7-5. Razrešivač u osam koraka pronalazi adresu udaljenog računara.

Pretpostavimo da lokalni server imena nikada ranije nije imao upit za takav domen i ne zna ništa o njemu. On može da se raspita kod susednih servera imena, ali ako ne sazna ništa, on šalje UDP paket serveru imena osnovnog domena *edit* koji postoji u njegovoj bazi podataka (slika 7-5), računani *edu-server.net.* Male su šanse da ovaj server zna adresu računara *linda.cs.yale.edu,* verovatno ni adresu domena *cs.ya- le.eclu,* ali on mora poznavati sve svoje potomke, tako da upit prosleđuje serveru imena domena *yale.edu* (korak 3). Ovaj, pak, prosleđuje upit serveru imena poddomena *cs.yale.edu* (korale 4), koji mora imati meritorne zapise resursa. Pošto se svaki upit prosleđuje od klijenta ka serveru, zapis resursa se vraća istim putem (koraci 5 do 8).

Kada zapisi stignu na server imena poddomena *cs.vu.nl,* beleže se u njegov keš, za slučaj da opet zatrebaju. Međutim, ti podaci nisu meritorni jer se izmene u domenu *cs.ycde.edu* neće automatski proširiti u sve keš memorije širom sveta. Zbog toga, podaci u kešu ne treba da se dugo čuvaju. To je i razlog što u svakom zapisu resursa postoji polje *Životni vek.* Njegova vrednost saopštava udaljenim serverima imena koliko dugo treba da čuvaju zapis. Ako neki računar svoju IP adresu ne menja godinama, onda se takva informacija može čuvati u kešu do jednog dana. Manje stabilne informacije treba brisati već posle nekoliko sekundi ili minuta.

Treba naglasiti da se opisani postupak uzastopnog slanja upita naziva rekurzivno ispitivanje (engl. *recursive query),* pošto svaki server koji nema traženu informaciju upit prosleđuje dalje, a odgovor istim putem prosleđuje natrag. Može se postupiti i drugačije. Ako lokalni server imena ne zna da odgovori na upit, on to klijentu i saopštava i istovremeno mu šalje ime sledećeg servera koji možda zna odgovor. Neki serveri ne vrše rekurzivno ispitivanje, već na upit uvele odgovaraju imenom sledećeg servera.

Treba pomenuti i to da DNS klijent koji tokom roka određenog tajmerom ne dobije odgovor na upit, sledeći upit šalje drugom serveru. Ovde se pretpostavlja da odgovornije stigao zato što je server isključen, a ne zato što su se upit ili odgovor izgubili.

Iako je sistem DNS izuzetno važan za ispravan rad Interneta, on u stvari samo preslikava simbolička imena računara u njihove IP adrese. On ne omogućava pronalaženje korisnika, resursa, usluga ili drugih objekata. Za pronalaženje takvih stvari, definisana je druga usluga, tzv. **jednostavan protokol za pristup imenicima** (engl. *Light-weight Directory Access Protocol, LDAP)*. To je pojednostavljena verzija usluge imenika po standardu OSI X.500, opisane u RFC dokumentu 2251. On podatke organizuje u stabla i omogućava pretraživanje prema njihovim različitim komponentama. Može se uporediti s „belim stranicama" telefonskog imenika. O njemu nećemo dalje govoriti u ovoj knjizi, a detalje potražite kod Weltmann i Dahburae (2000).

7.2 ELEKTRONSKA POŠTA

Elektronska pošta ili **e-pošta** (engl. *e-mail),* kako joj tepaju, postoji već više od dve decenije. Pre 1990. korišćena je uglavnom u akademskim krugovima. Tokom devedesetih izišla je u javnost i ubrzo se toliko proširila da danas broj elektronskih poruka poslatih u toku jednog dana daleko prevazilazi broj pisama poslatih običnom **zemaljskom poštom** (engl. *snail mail).*

E-pošta, kao i većina drugih vidova komunikacija, ima sopstvena pravila i stilove. Posebnost joj je što je prilično neformalna i pristupačna skoro svakom. Osobe koje ni u snu ne bi pisale nekom visokom zvaničniku ili ga pozvale telefonom, ne ok.leva.ju ni trenutka da mu e-poštom nešto naškrabaju.

E-pošta je prepuna šatrovačkih skraćenica, kao što su BTW (engl. *By The Way* - uzgred budi rečeno), ROTFL (engl. *Rolling On The Floor Laughing* - valjam se od smeha) i IMHO (engl. *In My Humble Opinion* - po mom skromnom mišljenju), a mnogi u svojim porukama koriste i ASCII simbole, npr. **smeška** (engl. *smiley)* i druge **emotikone** (engl. *emoticons).* Nekoliko najzanimljivijih prikazani su na slici 7-6. Biće vam jasniji ako knjigu zakrenete za 90 stepeni u smeru kazaljke. Sanderson i Dougherty (1993), sakupili su čak 650 takvih simbola u mini knjigu.

Smeško	Značenje	Smeško	Značenje	Smeško	Značenje
	Srećan sam	=I:")	Abraham Linkoln	:+)	Nosonja
>(	Tužan sam/ljut	=)>)	Ujka Sam	:-))	Debeljuca
	Apatičan sam	*<>)	Deda Mraz	:-0	Brkajlo
	Namigujem		Neznalica	#:-)	Čupavac
>(0)	Vičem		Australijanac	8-)	Ćora
	Bljujem	:-)X	Covek s leptir-mašnom	C:-)	Pametnjaković

Slika 7-6. Neki simboli smeška. Neće biti na završnom ispitu :-)

Prvi sistemi elektronske pošte imali su samo protokol za prenos datoteka i jedinstveno pravilo da prvi red poruke (tj. datoteke) mora da sadrži adresu primaoca. Kako je vreme prolazilo, ograničenja takvog rada bila su sve očiglednija.

Evo nekoliko karakterističnih primedaba upućenih takvim sistemima:

1. Sistem je nepogodan za slanje poruke grupi korisnika. Rukovodioci često žele da na taj način komuniciraju s podređenima.
2. Poruke nemaju svoju internu strukturu, što otežava računarsku obradu. Na primer, ako se prosleđena poruka uključi u tekst druge poruke, teško ju je odatle izvaditi.
3. Pošiljalac nikada ne zna da lije poruka stigla na odredište.
4. Ako neko planira da ode na put i želi da u međuvremenu sekretarica prima njegovu poštu, to nije lako organizovati.
5. Korisničko okruženje je loše integrisano sa sistemom prenosa, zbog čega korisnik mora prvo da napiše poruku u posebnom programu, da iziđe iz tog programa, a zatim da pozove program za prenos datoteka.
6. Nije moguće praviti i slati poruke koje istovremeno sadrže tekst, crtež, faks i glas.

Na osnovu stečenog iskustva, pravljeni su sve složeniji sistemi e-pošte. Godine 1982, u RFC dokumentima 821 (protokol za prenos podataka) i 822 (format poruke), objavljeni su predloži za ARPANET sistem e-pošte. Nešto izmenjene varijante ovih predloga, objavljene u RFC dokumentima 2821 i 2822, postale su standard za Internet, ali kada govore o e-posti na Internetu, svi se i dalje pozivaju na RFC dokument 822.

Godine 1984, CCITT je objavio preporuke X.400. Posle dvadesetogodišnjeg takmičenja, pobedio je sistem e-pošte zasnovan na RFC dokumentu 822, a sistem X.400 potpuno je iščezao. To što je sistem koji je skrpila šačica diplomaca pobedio zvanični međunarodni standard koji su podržavale sve PTT organizacije na svetu, mnoge vlade i znatan deo industrije računara, podseća na biblijsku priču o Davidu i Golijatu.

Sistem opisan u RFC dokumentu 822 svoj uspeh ne duguje kvalitetu, već tome što je standard X.400 projektovan tako neuko i komplikovano da niko nije uspeo da ga na pravi način ugradi. Birajući između jednostavnog, ali operativnog poštanskog sistema zasnovanog na RFC dokumentu 822 i navodno zadivljujućeg sistema X.400 koji ipak ne uspeva da proradi, većina organizacija se opredelila za prvi. Možda je to i pouka. Naše izlaganje, shodno tome, vezaćemo za sistem elektronske pošte na Internetu.

7.2.1 Arhitektura i usluge

U ovom odeljku ukratko ćemo razmotriti šta mogu da rade sistemi e-pošte i kako su organizovani. Svaki sistem e-pošte obično obuhvata dva podsistema: korisničkog agenta (engl. *user agent),* koji korisnicima omogućava da čitaju i šalju elektronske poruke, i agenta za prenos poruka (engl. *message transfer agent),* koji prenosi poruke od izvorišta do odredišta. Korisnički agenti su lokalni programi koji pomoću komandi, menija ili grafičkim putem, obezbeđuju interakciju sa sistemom e-pošte. Agenti za prenos poruka najčešće su sistemske usluge (engl. *daemons)* - procesi koji se izvršavaju u pozadini. Njihov zadatak je da prosleđuju poruku kroz sistem.

Sistemi e-pošte najčešće podržavaju pet osnovnih funkcija. Objasnimo ih pojedinačno.

Sastavljanje poruke (engl. *composition)* označava postupak pravljenja poruka i odgovora na poruke. Iako sam tekst poruke možete napisati u bilo kom programu za uređivanje teksta, sistem može da ponudi pomoć u pogledu adresiranja i dodavanja brojnih

zaglavlja koja se pridružuju svakoj poruci. Na primer, kada odgovarate na poruku, sistem može da iz nje izvuče adresu pošiljaoca i da je umetne na odgovarajuće mesto u odgovoru.

Prenos (engl. *transfer)* označava postupak prosleđivanja poruke od pošiljaoca ka primaocu. To uglavnom obuhvata uspostavljanje veze sa odredištem ili s nekim raču- narom posrednikom, slanje poruke i raskidanje veze. Sistem e-pošte to treba da radi automatski, ne tražeći pomoć od korisnika.

Izveštavanje (engl. *reporting)* ima za cilj da pošiljaoca upozna sa sudbinom poruke: da lije ona isporučena, odbačena ili izgubljena? Postoje brojne aplikacije u kojima je potvrđivanje poruka važno, pa čak može da ima i zakonske posledice („Poštovani sude, moj sistem e-pošte ne radi baš pouzdamo, pa pretpostavljam da se poziv za ročište negde usput igubio“).

Prikazivanje (engl. *displaying)* dolaznih poruka omogućava primaocima da ih čitaju. Ponelcada poruku treba preraditi ili pokrenuti specijalan čitač, na primer, ako je poruka u formatu PostScript ili predstavlja digitalizovan govor. Ponekada se pri prikazivanju isprobavaju različite jednostavne konverzije i formatiranje poruka.

Obrada (engl. *disposition)* obuhvata sve ono što korisnik na kraju radi s porukom postoje primi. On može daje odbaci pre čitanja, daje odbaci posle čitanja, daje sačuva itd. Potrebne su i mogućnosti ponovnog otvaranja i čitanja sačuvanih poruka, njihovog prosleđivanja ili obrađivanja na neki drugi način.

Osim navedenih osnovnih usluga, neki sistemi e-pošte, naročito interni sistemi korporacija, obezbeđuju i složenije usluge. Pomenimo kratko samo neke od njih. Kada korisnici privremeno ili trajno promene boravište, želeli bi da im se pošta pro- sleđuje, pa sistem mora biti u stanju da to radi automatski.

Većina sistema korisnicima omogućava da naprave **poštansko sanduče** (engl. *mailbox)* za dolaznu poštu i snabdeva ih komandama za pravljenje i brisanje poštanskih sandučića, za pregledanje njihovog sadržaja, za umetanje i vađenje poruka iz sandučića itd.

Rukovodioci u preduzećima često žele da pošalju cirkularnu poruku svim podređenim službenicima, svim kupcima ili svim dobavljačima. Tako dolazimo do koncepta **liste slanja** (engl. *mailing list)* - spiska adresa e-pošte. Kada se poruka pošalje listi slanja, svako s liste dobija po jednu njenu kopiju.

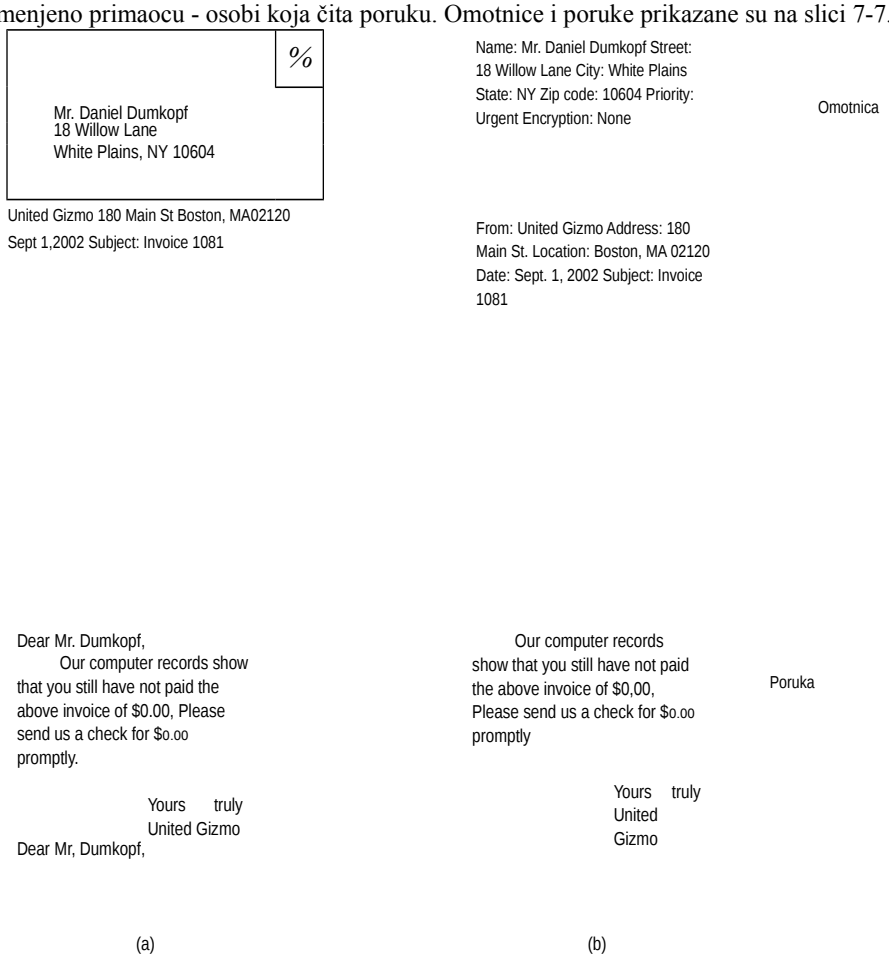
Ostale naprednije mogućnosti su: kopija poruke, nevidljiva kopija, prioritetne poruke, tajne (tj. šifrovane) poruke, alternativni primaoci (ako osnovni primalac nije na raspolaganju) i mogućnost da sekretarica čita poštu svog šefa i odgovara na nju.

E-pošta se danas široko koristi za komuniciranje unutar kompanija. Ona omogućava da na istom projektu rade saradnici, međusobno udaljeni i nekoliko časovnih zona. Kroz izostavljanje svake forme (naznake položaja, životnog doba ili pola), poruke

e-pošte omogućavaju da se saradnici usredsrede na suštinu projekta, umesto da brinu o tome *ko* je šta predložio. Uz e-poštu, veći utisak može da ostavi lepršava ideja nekog dopisnog studenta nego smrtno ozbiljan predlog nekog aktuelnog potpredsednika korporacije.

U sistemu e-pošte, ključna je razlika između omotnice poruke (engl. *envelope*) i njenog sadržaja. Omotnica kapsulira poruku. Ona sadrži sve podatke potebne za prenošenje poruke, kao što su odredišna adresa, prioritet i bezbednosni nivo, a svi ti podaci potpuno su odvojeni od same poruke. Agenti za prenos poruka koriste omotnice za usmeravanje, baš kao što bi činili i pravi poštari.

Poruka unutar omotnice sadrži dva dela: zaglavlje (engl. *header)* i telo poruke (engl. *body)*. Zaglavlje sadrži upravljačke podatke namenjene korisničkim agentima. Telo je namenjeno primaocu - osobi koja čita poruku. Omotnice i poruke prikazane su na slici 7-7.



Slika 7-7. Omotnice i poruke, (a) Klasično pismo, (b) Elektronska poruka.

7.2.2 Korisnički agent

Kao što smo videli, sistem e-pošte ima dva podsistema: korisničkog agenta i agenta za

prenos poruka. U ovom odeljku razmotrićemo korisničkog agenta. To je obično program (ponekada zvan i čitač pošte) koji prihvata komande za sastavljanje poruka, za njihovo primanje, za odgovaranje na njih, kao i za rad s poštanskim sandučićima.

Neki korisnički agenti imaju maštovita okruženja s menijima ili ikonicama koji se otvaraju pritiskom miša, dok dragi očekuju upisivanje jednoslovnih komandi s tastature. U funkcionalnom smislu, ti programi su isti. Neki sistemi imaju menije i ikonice, ali podržavaju i prečice s tastature.

Slanje e-pošte

Da bi poslao poruku elektronskim putem, korisnik mora da ima spremnu poruku, odredišnu adresu i možda još neke parametre. Poruka se može sastaviti u zasebnom programu za uređivanje teksta, u programu za obradu teksta ili u specijalnom programu za uređivanje teksta koji je ugrađen u korisničkog agenta. Odredišna adresa mora biti u formatu koji korisnički agent može da prepozna. Mnogi korisnički agenti očekuju adresu u obliku *korisnik@dns-adresa.* Pošto smo sistem DNS razmotrili ranije, nećemo ga ponovo objašnjavati.

Međutim, treba reći da postoje i dragi oblici adesa. Konkretno, adrese u sistemu X.400 izgledaju potpuno drugačije od DNS adresa. Njih čine parovi *atribut - vred- nost,* razdvojeni kosim crtama, na primer:

/C=US/ST=MASSACHUSETTS/L=CAMBRIDGE/PA=360 MEMORIAL DR./CN=KEN SMITH/

Ova adresa definiše saveznu državu, državu, mesto, ulicu i ime osobe (Ken Smith). Postoje i mnogi drugi atributi, tako da elektronsku poruku možete poslati i osobi čiju tačnu adresu e-pošte ne znate, pod uslovom da znate dovoljno drugih atributa (npr. kompaniju i radno mesto). Iako su adrese koje se koriste u sistemu X.400 mnogo ne- pogodnije od DNS adresa, mnogi sistemi e-pošte podržavaju alijase (nadimke) koji korisnicima omogućavaju da unesu ime osobe i da tako dobiju njenu tačnu adresu e-pošte. Prema tome, čak i u sistemu X.400 obično nije neophodno da stvarno unosite ove neobične redove teksta.

Većina sistema e-pošte podržava liste slanja, tako da korisnik jednom komandom može da pošalje istu poruku na više adresa. Ako se lista slanja održava lokalno, korisnički agent na svaku adresu s liste šalje po jednu kopiju poruke. Međutim, ako se lista održava na udaljenom računaru, poruka se na njemu i umnožava. Na primer, ako grupa ljubitelja ptica ima listu slanja *sova* instaliranu na računaru *ptice.arizona.edu,* tada će svaka poruka poslata na adresu sova@ptice.arizona.edu biti upućena na Univerzitet Arizone, tamo umnožena i poslata svakom članu s liste slanja, bez obzira na to gde se nalazi. Korisnici ove liste ne mogu da utvrde da je u pitanju lista slanja. Njima to može izgledati i kao lično poštansko sanduče Prof. Konstantina M. Sove.

Čitanje e-pošte

Kada se pokrene korisnički agent, on obično pogleda u korisnikovo poštansko sanduče za dolaznu poštu pre nego što išta prikaže na ekranu. Tada može da objavi broj pristiglih poruka ili da za svaku prikaže jednoredni sažetak i da čeka komandu.

Da biste razumeli kako radi korisnički agent, razmotrite jedan tipičan scenario. Pošto pokrene korisničkog agenta, korisnik traži da vidi sadržaj poštanskog sanduče- ta. Na ekranu se pojavljuje prikaz sličan onom sa slike 7-8. Svaki red odgovara jednoj poruci. U ovom primeru, poštansko sanduče sadrži osam poruka.

Svaki red prikaza sadrži više polja izvučenih iz omotnice ili zaglavlja odgovarajuće poruke.

U jednostavnim sistemima e-pošte, unapred su određena polja koja se prikazuju. U složenijim sistemima, korisnik može da izabere polja za prikazivanje de- finišući **korisnički profil** (engl. *userprofile),* tj. datoteku koja opisuje format prikaza. U našem jednostavnom primeru, prvo polje je redni broj poruke. Drugo polje, *Indikatori* (Flags), može da sadrži *K,* što znači da je poruka pročitana i ostavljena u poštanskom sandučetu; *A* znači da je na poruku već odgovoreno, a *F* da je poruka prosleđena nekom drugom. Postoje i dragi indikatori.

Redni broj	Indikatori	Bajtovi	Pošiljalac	Predmet poruke
1	K	1030	admin	Izmene za MINIX
2	KA	6348	tanja	Nisu sve Tanje navalentne
3	K F	4519	Amy N. Wong	Molim informacije
4		1236	bal	Bioinformatika
5		104110	kolega	Materijal za recenziju
6		1223	Boris	Re: Predlog za dodelu kredita
7		3110	Mika	Nas rad je prihvaćen za štampu
8		1204	dmr	Re: Studentska poseta

Slika 7-8. Primer prikaza sadržaja poštanskog sandučeta.

Treće polje sadrži dužinu poruke, a četvrto ime pošiljaoca. Pošto je polje jednostavno izvučeno iz poruke, ono može da sadrži lično ime, puno ime, incijale, korisničko ime ili već ono što je pošiljalac upisao. Najzad, polje *Predmet poruke* (Subject) sadrži kratak opis poruke. Ako ništa ne upišete u ovo polje, najčešće ćete utvrditi da odgovor na vašu poruku nema najviši prioritet.

Pošto se prikažu zaglavlja poruka, korisnik može da izvede jednu od više akcija: da prikaže poruku, da obriše poruku itd. Stariji sistemi su radili u tekstualnom režimu i ove akcije su najčešće pokretane jednoslovnim komandama s tastature, na primerT (odštampaj poruku), A (odgovori na poruku), D (obrisi poruku) i F (prosledi poruku). Argumentom se zadavala konkretna poruka. Savrerneni sistemi rade u grafičkom režimu. Korisnik obično izabere poruku mišem, a zatim pritiska odgovarajuće ikonice da bi poruku odštampao, da bi na nju odgovorio, obrisao je ili prosledio.

E-pošta je prevalila dug put od onog doba kada je bila smo sredstvo za razmenu datoteka. Složeni korisnički agenti danas omogućavaju manipulisanje velikom količinom e-poraka. Za one koji godišnje primaju i šalju na hiljade poruka, takva alatka je nezamenljiva.

7.2.3 Formati poruka

Pređimo sada iz korisničkog okruženja na sam format poruka e-pošte. Najpre ćemo razmotriti osnovni tekstualni format prema RFC dokumentu 822, a zatim obraditi i njegova multimedijska proširenja.

RFC 822

Poruke se sastoje od sasvim jednostavne omotnice (opisane u RFC dokumentu 821), zaglavlja s nekoliko polja, praznog reda i teksta (tela) same poruke. Svako polje zaglavlja (logički) predstavlja jedan red teksta sa imenom polja, dvotačkom i, za većinu polja, s

vrednošću. RFC dokument 821, napravljen pre više decenija, ne pravi jasnu razliku između polja omotnice i polja zaglavlja. Iako je dopunjen RFC dokumentom 2822, nije ga bilo moguće u potpunosti preraditi upravo zbog njegove široke upotrebe. Pri normalnom korišćenju, korisnički agent pravi poruku i prosleđuje je agentu za prenos poruka koji na osnovu nekih polja iz zaglavlja sklapa omotnicu - pomalo staromodna mešavina teksta i adresa.

Osnovna polja zaglavlja koja se tiču prenosa poruke prikazana su na slici 7-9. IJ polju *To:* nalazi se DNS adresa glavnog primaoca. Može postojati i više primalaca. U polju *Cc:* nalazi se jedna ili više adresa sporednih primalaca. U pogledu isporuke, nema nikakve razlike između glavnog i sporednih primalaca. Ta psihološka razlika (glavni - sporedan) može biti važna ljudima, ali ne i poštanskom sistemu. Izraz *Cc:* (engl. *Carbon copy* - indigo kopija) pomalo je zastareo, pošto računari ne koriste indigo, ali se i dalje održava. Polje *Bcc:* (engl. *Blind carbon copy* - nevidljiva kopija) slično je polju *Cc:,* osim što se taj red ne pojavljuje u kopijama koje se šalju glavnom i sporednim primaocima. To omogućava slanje kopija trećim licima, a da glavni i sporedni primaoci za to ne znaju.

Polje zaglavlja	**Značenje**
To:	.Jedna ili više adresa glavnih primalaca
Cc:	Jedna ili više adresa sporednih primalaca
Bcc:	Jedna ili više adresa za slanje nevidljivih kopija
From:	Osoba ili osobe koje su sastavile poruku
Sender:	Adresa stvarnog pošiljaoca
Received:	Red koji dodaje svaki agent za prenos duž putanje
Return-Path:	Može se navesti povratna putanja

Slika 7-9. Polja zaglavlja koja se prema RFC dokumentu 822 odnose na prenos poruke,

U sledeća dva polja, polju *From:* i polju *Sender:,* saopštava se ko je napisao, a ko poslao poruku. Pisac i pošiljalac poruke ne moraju biti ista osoba. Na primer, direktor može da sastavi poruku, ali je stvarno šalje njegova sekretarica. U tom slučaju, direktor bi se našao u polju *From:,* a njegova sekretarica u polju *Sender:*. Polje *From:* je obavezno, ali se polje *Sender:* može izostaviti ako je isto što i *From:*. Ta polja su potrebna za slučaj da se poruka ne može isporučiti, već se mora vratiti pošiljaocu.

Svaki agent za prenos na putu poruke unosi u nju po jedan red *Received:*. Red sadrži identifikator agenta, datum i vreme prijema poruke, kao i druge podatke koji se mogu iskoristiti za pronalaženje greške u sistemu usmeravanja.

Polje *Return-Path:* dodaje poslednji agent za prenos poruka. Ono definiše povratnu putanju do pošiljaoca. Ta informacija bi se teorijski mogla izvući analiziranjem svih polja *Received:* (zanemarujući ime poštanskog sandučeta pošiljaoca), ali se to retko radi, pa polje uglavnom sadrži samo pošiljaočevu adresu.

Osim polja sa slike 7-9, poruke prema RFC dokumentu 822 mogu sadržati i polja zaglavlja namenjena korisničkim agentima, odnosno samom korisniku. Najčešća takva polja prikazana su na slici 7-10. Većina ih je jasna po sebi, pa ih nećemo detaljno analizirati.

Polje zaglavlja	Značenje
Date:	Datum i vreme slanja poruke
Reply-To:	Adresa na koju treba slati odgovor
Message-Id:	Jedinstven identifikator preko koga se kasnije pristupa poruci
In-Reply-To:	Identifikator poruke na koju se odgovara
References:	Drugi relevantni identifikatori
Keywords:	Ključne reći koje bira korisnik
Subject:	Jednoredni sažetak poruke

Slika 7-10. Neka polja koja se prema RFC dokumentu 822 koriste u zaglavlju poruke e-pošte.

Polje *Reply-To:* popunjava se onda kada ni onaj koje sastavio poruku, ni onaj koji ju je poslao ne žele da dobiju odgovor. Na primer, šef marketinške službe šalje cirku- larnu poruku kupcima obaveštavajući ih o novom proizvodu. Poruku stvarno šalje njegova sekretarica, ali se u polje *Reply-To:* upisuje adresa šefa prodaje koji može da odgovori na potencijalna pitanja i da preuzme porudžbine. To polje je korisno i onda kada pošiljalac ima dve adrese e-pošte, pa želi da odgovori stižu na onu drugu adresu.

RFC dokumentom 822 izričito se dozvoljava korisnicima da prema svojim potrebama uvode nova polja, uz uslov da ona uvek počinju tekstom *X-*. Time se garantuje da buduća polja neće počinjati tekstom *X-,* pa su tako unaped izbegnuti sukobi između zva- ničnih i privatnih polja. Ponekada maloletni pametnjakovići smišljaju polja, kao što su *X-Fruit-of-th.e-Day* ili *X-Disease-of-the-Week,* koja su legalna, ali ne uvek i duhovita.

Posle zaglavlja dolazi telo poruke. U njega korisnici mogu da smeste šta god žele. Neki korisnici završavaju poruku svojevrsnim „potpisom“ koji sadrži jednostavne ASCII sličice, poznate i manje poznate citate, političke stavove i izjave svake vrste (npr. „Korporacija XYZ nije odgovorna za izneto mišljenje - u stvari, ona ne može ni da ga shvati“).

MIME - Višenamenski priključci za Internet poštu

U ranim danima ARPANET-a, e-pošta se sastojala isključivo od tekstualnih, ASCII poruka pisanih engleskim jezikom. U takvoj situaciji, sasvim je zadovoljavao RFC dokument 822: on je definisao zaglavlje, ali je sadržinu u potpunosti prepuštao korisnicima. Danas, na globalno povezanom Internetu, takav pristup više ne zadovoljava. Problemi nastaju pri slanju i primanju

1. Poruka na jezicima koji koriste dijakritičke znakove ili akcentovana slova (npr. na srpskom, hrvatskom, francuskom, nemačkom itd.).
2. Poruka na nelatiničnim pismima (npr. na ćirilici, hebrejskom ili ruskom).
3. Poruka sastavljenih ideografskim (neslovnim) pismom (npr. na kineskom ili japanskom)
4. Poruka koje uopšte ne sadrže tekst (već, npr. zvuk ili sliku).

Rešenje je predloženo u RFC dokumentu 1341 i dopunjeno u RFC dokumentima 2045-2049. To rešenje, poznato kao *MIME* (od engl. *Multipurpose Internet Mail Extensions* - Višenamenski priključci za Internet poštu), sada se široko primenjuje. U nastavku ćemo ga opisati, a detalje potražite u originalnim RFC dokumentima.

Standard MIME u telo poruke uvodi strukturu i definiše pravila kodiranja ne- tekstualnih

poruka, ali u osnovi i dalje omogućava korišćenje formata prema RFC dokumentu 822. Na taj način, poruke se i dalje mogu razmenjivati pomoću postojećih programa i protokola za poštu. Treba promeniti samo programe za slanje, odnosno primanje poruka, što mogu da učine sami korisnici.

Standard MIME definiše pet novih zaglavlja (slika 7-11). U prvom zaglavlju, korisničkom agentu se saopštava da ima posla sa MIME porukom i navodi se verzija MIME standarda. Poruka koja ne sadrži zaglavlje *MIME-Version:* smatra se tekstualnom porukom pisanom na engleskom jeziku i na taj način i obrađuje.

Zaglavlje	**Značenje**
MIME-Version:	Sadrži verziju MIME standarda
Content-Description:	Tekstualni opis sadržaja
Content-Id:	Jedinstven identifikator
Content-Transfer-Encoding:	Način prelamanja tela poruke pri prenosu
Content-Type:	Vrsta i format sadržaja

Slika 7-11. Zaglavlja koja je formatu RFC 822 dodao standard MIME.

Zaglavlje *Content-Description:* sadrži ASCII tekst sa opisom sadržaja poruke. Na osnovu njega primalac može da utvrdi isplati li se da poruku dekodira i pročita. Ako tamo piše: „Fotografija Barbarinog hrčka“, a osoba koja primi poruku nije veliki ljubitelj hrčaka, verovatno neće dekodirati ogromnu kolor fotografiju, već će poruku odbaciti.

Polje *Content-Id:* identifikuje sadržaj. Koristi se isti format kao i za polje *Messa- ge-Id:* osnovnog zaglavlja.

Polje *Content-Transfer-Encoding:* saopštava način prelamanja poruke na putu kroz mrežu koja možda ne prepoznaje ništa drugo oslim slova, brojeva i znakova interpunkcije. Postoji pet načina prelamanja (plus jedan koji omogućava uvođenje dodatnih načina). Najjednostavniji način je 7-bitno ASCII kodiranje (običan tekst), pri čemu poštanski protokol može da direktno prenese poruku, pod uslovom da nijedan red nije duži od 1000 znakova.

Sledeći načinje 8-bitno (prošireno) ASCII kodiranje, tj. znaci se kodiraju brojevima iz zatvorenog intervala 0-255. Ovaj način kodiranja narušava (originalni) protokol za Internet poštu, ali se koristi u nekim delovima Interneta gde je taj originalni protokol proširen. Iako način kodiranja ne postaje automatski legalan zbog toga što se navede, barem je kasnije lakše naći uzrok ako nešto pođe naopako. Ni 8-bitno kodirane poruke ne smeju sadržati redove duže od 1000 znakova.

Još gore su binarno kodirane poruke. To su binarne datoteke proizvoljnog formata koje ne samo što koriste svih 8 bitova, već ne poštuju ni ograničenje od 1000 znakova u jednom redu. Izvršne datoteke (programi) pripadaju ovoj kategoriji. Ništa ne garan- tuje da će binarna datoteka stići u ispravnom stanju, pa ipak, mnogi pokušavaju da ih na takav način prenose.

Ispravan način pravljenja binarnih poruka predstavlja **sistem kodiranja Base64** (engl. *base64 encoding),* ponekad zvan i ASCII **oklop** (engl. *ASCII armor).* Prema njemu, grupe od po 24 bita razbijaju se na 6-bitne jedinice i svaka jedinica se šalje kao legalan ASCII znak. Nula se kodira slovom „A“, jedinica slovom „B“ i tako dalje, zatim sledi 26 malih slova, deset

cifarai na kraju, „+“ je 62, a „/“je 63. Sekvenca „==“, odnosno „=“ znači da poslednja gnrpa sadrži samo 8, odnosno 16 bitova. Znakovi za vraćanje na početak reda (CR) i za prelazak u novi red (LF) ne uzimaju se u obzir, tako da se mogu proizvoljno umetati da bi se redovi na pravi način prelamali. Opisanim postupkom tekstualnog kodiranja bezbedno se može slati svaka binarna datoteka.

Tekstualne poruke koje sadrže samo nekoliko netekstualnih znakova neefikasno je kodirati sistemom Base64. Za njih se koristi **sistem kodiranja quoted-printable.** To je u suštini 7-bitno ASCII kodiranje, s tim što se znaci sa ASCII kodom većim od 127 zamenjuju znakom jednakosti i kodom znaka izraženim pomoću dve heksadeci- malne cifre.

Sve u svemu, binarni podaci se mogu kodirati sistemom Base64 ili sistemom quoted-printable. Kada postoji dobar' razlog da se ne kodira nijednim od ova dva sistema, može se u zaglavlju *Content-Transfer-Encoding:* zadati sistem kodiranja koji definiše sam korisnik.

Poslednje zaglavlje na slici 7-11, u stvari je najzanimljivije. Njime se zadaje vrsta tela poruke. RFC dokumentom 2045 definisano je sedam tipova, a svaki ima jedan ili više podtipova. Tip se od podtipa razdvaja kosom crtom:

```
Content-Type: video/mpeg
```

Podtip se u zaglavlju mora zadati; ne postoji podrazumevana vrednost. Prvobitna lista tipova i podtipova definisanih RFC dokumentom 2045 prikazana je na slici 7-12. Od tada je dodato mnogo novih tipova, a i dalje se po potrebi dodaju novi.

Tip	**Podtlp**	**Opis**
Text	Plain	Neformatiran tekst
	Enriched	Tekst s jednostavnim komandama za formatiranje
Image	Gif	Slika u formatu GIF
	Jpeg	Slika u formatu JPEG
Audio	Basic	Zvuk
Video	Mpeg	Film u formatu MPEG
Application	Octet-stream	Nepreveden tok bajtova
	Postscript	PostScript dokument za štampanje
Message	Rfc822	MIME poruka prema RFC 822
	Partial	Poruka je podeljena u prenosu
	External-body	Poruka se mora preuzeti s mreže
Multipart	Mixed	Nezavisni delovi po navedenom redu
	Alternative	Ista poruka u više formata
	Parallel	Delovi se moraju gledati istovremeno
	Digest	Svaki deo je potpuna poruka prema standardu RFC 822

Slika 7-12. M1ME tipovi i podtipovi definisani RFC dokumentom 2045.

Pogledajmo letimično listu tipova. Tip *text* odgovara ASCII tekstu. Kombinacija *text/plain* predviđena je za obične poruke koje se prikazuju onako kako su primljene, bez kodiranja i dalje obrade. Ova opcija omogućava da se šalju MIME poruke kojima je dodato samo nekoliko

novih zaglavlja.

Podtip *text/enriched* omogućava da se u poruci iskoristi jednostavan jezik za označavanje teksta. Taj jezik, nezavisno od sistema, omogućava da se tekst prikaže polu- crno, u kurzivu, s krupnijim ili sitnijim slovima, da se uvuče, poravna, prikaže kao eksponent ili indeks, i da se jednostavno rasporedi na stranici. Jezik za označavanje zasniva se na **standardnom opštem jeziku za označavanje** (engl. *Standard Generalized Markup Language, SGML),* na kome se zasniva i HTML koji se koristi na Webit. Na primer, poruka

Došlo je <bold> vreme </bold> reče <italic> morž </italic> ...

prikazala bi se ovako

Došlo je **vreme** reče *mori*...

Sistem primaoca treba da izabere način prikazivanja. Ako je u stanju da tekst prikaže polucrno ili u kurzivu, to će i učiniti; ako takvih opcija nema, tekst može da istakne bojom, treptanjem, podvlačenjem ili da ga prikaže inverzno. Različiti sistemi mogu različito da prikažu tako formatiran tekst, a to i čine.

Kada je Web postao popularan, dodat je nov podtip *text/html* (RFC dokumentom 2854), da bi se omogućilo slanje Web strana u porukama e-pošte prema standardu iz RFC dokumenta 822. Podtip *text/xml* za proširivi ježile za označavanje definisan je RFC dokumentom 3023. O jezicima HTML i XML govorićemo u nastavku poglavlja.

Sledeci MIME tip je *image* i koristi se za prenošenje slika. Danas se za skladištenje i prenošenje slika koriste mnogi komprimovani i nekomprimovani formati. Dva takva formata: GIF i JPEG, ugrađena su u skoro svaki čitač Weba, ali su prvobitnoj listi formata dodati i mnogi drugi.

Tipovi *audio* i *video* odnose se na zvuk i video sekvence. Imajte na umu da video obuhvata samo sliku, a ne i ton. Ako treba da prenesete ozvučeni film, možda morate da ih zasebno prenosite, što zavisi od upotrebljenog sistema kodiranja. Prvi definisani video format napravila je tzv. Grupa eksperata za film (engl. *Moving Picture Expert Group, MPEG),* po kojoj je i dobio ime, ali je od tada dodato mnogo novih formata. Pored tipa *audio/basic,* RFC dokumentom 3003 dodat je i nov tip *audio/mpeg* da bi se korisnicima omogućilo da e-poštom razmenjuju zvučne MP3 datoteke.

Tip *application* okuplja formate za koje je potrebna posebna obrada. Podtip *octet- -stream* predstavlja samo sekvencu sirovih bajtova. Kada primi takvu selcvencu, korisnički agent će je možda prikazati i ponuditi korisniku daje snimi u datoteku. Dalja obrada datoteke je dužnost korisnika.

Drugi definisani podtip je *postscript* i odnosi se na Adobeov jezik PostScript za opisivanje štampanih strana. Mnogi štampači imaju ugrađene PostScript prevodioce. Iako korisnički agent može da pozove nezavisnog PostScript prevodioca da prikaže datoteku, to je skopčano sa izvesnim rizikom. PostScript je potpun programski ježile u kome neki mazohista, ako ima dovoljno vremena, može da napiše programski prevodilac za C ili sistem za rad s bazama podataka. Dolazna PostScript poruka prikazuje se tako što se pokrene PostScript program koji ona sadrži. Osim što će prikazati tekst, taj program može da učitava, menja ili briše korisničke datoteke, kao i da izazove neke druge neugodne efekte.

Tip *message* omogućava da se jedna poruka potpuno kapsulira drugom. Ta šema je, na

primer, korisna za prosleđivanje poruka. Kada se potpuna poruka prema standardu RFC 822 kapsulira unutar spoljne poruke, treba upotrebiti podtip *rfc822.*

Podtip *partial* omogućava da se kapsulirana poruka izdeli na više delova i oni pošalju zasebno (npr. u slučaju kada je kapsulirana poruka predugačka). Parametrima se postiže da se svi delovi na odredištu ispravno sklope u celinu.

Najzad, podtip *external-body* može se upotrebiti za veoma dugačke poruke (npr. za video filmove). Umesto da se u poruku uključi MPEG datoteka, navodi se FTP adresa, a tada primaočev korisnički agent može da preko mreže preuzme datoteku u trenutku kada mu je potrebna. Ova mogućnost je posebno korisna ako film šaljete na listu slanja, a očekujete da će ga pogledati samo nekoliko korisnika s nje (imajte na umu zatrpavanje korisnika reklamnim video porukama).

Poslednji tip, *multipart,* omogućava da poruka bude iz više jasno odeljenih delova. Kod podtipa *mixed,* svaki deo može da bude drugačiji, bez nametanja dodatne strukture. Mnogi programi za e-poštu dozvoljavaju korisnicima da tekstualnoj poruci pridruže jedan i više priloga. Prilozi se šalju uz korišćenje podtipa *multipart.*

Za razliku od podtipa *multipart,* podtip *alternative* omogućava da ista poruka bude uključena više puta, ali u različitim formatima. Na primer, poruka može istovremeno biti poslata kao običan tekst, kao delimično formatiran tekst (enriched) i u PostScript formatu. Ispravno podešen korisnički agent koji primi takvu poruku, prikazaće je - ako može - u PostScript formatu. Sledeći pokušaj je prikazivanje delimično formatiranog teksta. Ako ništa od toga ne uspe, prikazaće se običan ASCII tekst. Delove treba rasporediti od najjednostavnijeg ka složenijim, da bi i primaoci sa starijim korisničkim agentima (bez MIME standada) mogli nešto da izvuku iz poruke, jer i takvi agenti mogu da protumače običan ASCII tekst.

Podtip *alternative* može se koristiti i za višejezične poruke. U ovom smislu, čuvena kamena ploča iz Rozete s trojezičnim tekstom (pisanim grčki, hijeroglifima i de- motskim pismom) predstavlja prastaru poruku tipa *multipart/alternative.*

Na slici 7-13 prikazanje multimedijski primer. Tom porukom se kao rođendanska čestitka istovremeno šalju tekst i pesma. Ako korisnik, može da sluša zvuk, korisnički agent će preuzeti zvučnu datoteku birthday.snd i reprodukovati je. Ako korisnik nema mogućnost slušanja zvuka, u potpunoj tišini će se na ekranu prikazati samo tekst pes- mice. Delovi su razgraničeni udvojenom crticom iza koje sledi (softverski generisan) tekstualni niz zadat parametrom boundary.

From: dragana@abcd.com To: vesna@xyz.com MIME-Version: 1.0
Message-Id: <0704760941 .AA00747@abcd.com>
Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
Subject: Svaki put kad Zemlja obiđe Sunce imaš jednu godinu više

Ovo je preambula koju korisnički agent zanemaruje. Želim ti lep provod.

-qwertyuiopasdfghjklzxcvbnm Content-Type: text/enriched

Happy birthday to you
Happy birthday to you
Happy birthday dear
<bold> Vesna </bold>
Happy birthday to you

```
-qwertyuiopasdfghjklzxcvbnm Content-Type: message/external-body; access-
type="anon-ftp"; site="bicycle.abed.com"; directory="pub";
name="birthday.snd"

content-type: audio/basic content-transfer-encoding: base64
-qwertyuiopasdfghjklzxcvbnm-
```

Slika 7-13. Višedelna poruka. Alternative su: delimično formatiran tekst i audio.

Obratite pažnju na to da se zaglavlje *Content-Type* u primeru pojavljuje na tri me- sta. Najpre se u vrhu njime naznačava da poruka ima tri dela. Unutar svakog dela ono označava tip i podtip tog dela. Na kraju, unutar drugog dela, ono je neophodno i da bi korisnički agent znao koju vrstu spoljne datoteke da preuzme. Tu malu razliku istakli smo tako što smo zaglavlje napisali malim slovima, ali je to programski nebitno jer se u zaglavljima ne pravi razlika između velikih i malih slova. Slično tome, zaglavlje *content-transfer-encoding* neophodno je za svaku spoljnu datoteku koja nije kodirana kao 7-bitni ASCII tekst.

Ako se vratimo na tip *multipart,* tu postoje još dva podtipa. Podtip *parallel* se koristi kada sve delove poruke treba prikazati istovremeno. Na primer, filmovi obično imaju i zvučni i video kanal. Filmovi ostavljaju bolji utisak ako se kanali reprodukuju istovremeno, nego jedan za drugim.

Na kraju, tip *digest* se koristi onda kada se mnoge poruke pakuju u jednu kombi- novanu poruku. Na primer, neke diskusione grupe na Internetu prikupljaju poruke od učesnika i šalju ih grupi kao poruku tipa *multipart/digest.*

7.2.4 Prenos poruka

Sistem prenosa poruka bavi se upućivanjem poruke od pošiljaoca do primaoca. Najjednostavnije je da se za to prethodno uspostavi prenosna veza između izvorišnog i odredišnog računara. Pošto objasnimo kako se to normalno radi, razmotricemo situacije gde to ne ide i ukazati kako da se takvi problemi prevaziđu.

SMTP - Jednostavan protokol za prenos elektronske pošte

E-pošta se na Internetu isporučuje tako što izvorišni računar uspostavi TCP vezu s priključkom 25 odredišnog računara. Na tom priključku osluškuje sistemska poštanska usluga koja razume **jednostavan protokol za prenos elektronske pošte** (engl. *Simple Mail Transfer Protocol, SMTP).* Ona prihvata uspostavljanje veza i kopira poruke s njih u odgovarajuće poštanske sandučiće. Ako se poruka ne može isporučiti, pošiljaocu se vraća izveštaj o grešci s prvim delom neisporučene poruke.

SMTP je jednostavan tekstualni protokol. Pošto uspostavi TCP vezu s priključkom 25, pošiljalac, r adeći kao klijent, čeka da prvo progovori primalac koji radi kao server. Server počinje tako što šalje red teksta u kome se predstavlja i saopštava da lije spreman da primi poruku. Ako nije spreman, klijent raskida vezu i kasnije ponovo pokušava daje uspostavi.

Ako je server voljan da primi poruku, klijent saopštava od koga je poruka i kome je namenjena. Ako takav primalac postoji na odredištu, server daje klijentu zeleno svetio da pošalje poruku. Zatim klijent šalje poruku i server je potvrđuje. Nije potrebno izračunavanje kontrolnih zbirova jer TCP obezbeđuje pouzdan tok bajtova. Ako postoji još poruka, i one se

šalju tada. Kada se razmene sve poruke u oba smera, veza se raskida. Primer dijaloga upotrebljenog za slanje poruke sa slike 7-13, uključujući i numeričke kodove koje koristi protokol SMTP, prikazan je na slici 7-14. Redovi koje šalje klijent označeni su sa *K:*. Oni koje šalje server nose oznaku *S:*.

Potrebno je nekoliko napomena o slici 7-14, Prva klijentova komanda zaista je *HELO*. Od različitih skraćenica za reč HELLO, ova ima najviše prednosti. Zašto sve komande treba da su četvoroslovne verovatno se može nadi u istoriji nastanka protokola.

Na slici 7-14, poruka se šalje samo jednom primaocu, talco da se koristi samo jedna komanda RCPT. Takvim komandama se jedna poruka može slati na više adresa. Svaka poruka se pojedinačno potvrđuje ili odbacuje. Čak i kada se za neke primaoce poruka odbaci (jer se primaoci ne nalaze na odredištu), ona se može poslati drugima.

Na kraju, iako je sintaksa četvoroslovnih komandi klijenta strogo delinisana, sintaksa odgovora je fleksibilnija. Samo je numerički kod stvarno bitan. Posle njega, svaka realizacija može da dopiše proizvoljan tekst.

Da biste stekli osećaj kako radi protokol SMTP i drugi protokoli koje opisujemo u ovom poglavlju, isprobajte ih. Najpre se smestite uz računar koji je priključen na Internet. Ako je to UNIX računar, upišite

```
telnet pošta.davalac.com 25
```

stavljajući umesto *pošta.davcilac.com* DNS ime poštanskog servera vašeg davaoca Internet usluga. Na Windows računani, pritisnite Start, Run, a zatim upišite komandu u okvir za dijalog. Ta komanda će uspostaviti telnet (tj. TCP) vezu s priključkom 25 računara. Priključak 25 je rezervisan za SMTP (na slici 6-27 prikazani su nelđ uobičajeni priključci). Verovatno će odgovor glasiti približno ovako:

```
Trying 192.30.200.66...
Connected to pošta.davalac.com
Escape character is
220 pošta.davalac.com Smail #74 ready at Thu, 25 Sept 2004 13:26 +0200
```

U prva tri reda telnet vas izveštava šta radi. Poslednji red je sa SMTP servera udaljenog računara i saopštava daje taj računar voljan da s vama razgovara i da prihvati e-poštu. Da biste videli koje komande on prihvata, upišite

```
HELP
```

Od tog trenutka, počev od klijentove komande *HELO,* komande se mogu nizati kao na slici 7-14.

```
                S: 220 xyz.com SMTP service ready
K: HELO abcd.com
                S: 250 xyz.com says hello to abcd.com
K: MAIL FROM: <dragana@abcd.com>
   S; 250 sender ok K: RCPT TO:
                <vesna@xyz.com>
                S: 250 recipient ok
K: DATA
                S: 354 Send mail; end with on a line by itself K:
From: dragana@abcd.com K: To: vesna@xyz.com K: MIME-
Version: 1.0
K: Message-Id: <0704760941.AA00747@abcd.com>
K: Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
```

K: Subject: Svaki put kad Zemlja obiđe Sunce imaš jednu godinu više K:
K: Ovo je preambula koju korisnički agent zanemaruje. Želim ti lep provod.
K:
K: -qwertyuiopasdfghjklzxcvbnm K: Content-Type: text/enriched K:
K: Happy birthday to you
K: Happy birthday to you
K: Happy birthday dear
<bold> Vesna </bold>
K: Happy birthday to you K:
K: -qwertyuiopasdfghjklzxcvbnm K: Content-Type: message/external-body;
K: access-type="anon-ftp";
K: site="bicycle. abed, com";
K: directory="pub";
K: name="birthday.snd"
K:
K: content-type: audio/basic K: content-transfer-encoding: base64 K:
-qwertyuiopasdfghjklzxcvbnm
K:.
S: 250 message accepted
K: QUIT

S: 221 xyz.com closing connection Slika 7-14. Slanje poruke od

dragana@abccl.com do vesna@xyz.com.

Treba naglasiti da korišćenje običnog ASCII teksta za komandovanje nije slučajno izabrano. Većina protokola za Internet radi na taj način. Kada su komande tekstualne, u protokolima se pri proveravanju lako pronalaze greške. Protokoli se mogu provera- vati ručnim upisivanjem komandi, kao u primeru, a pristigle poruke su lako čitljive.

Iako je protokol SMTP sasvim dobro definisan, jos uvek mogu da nastanu problemi. Jedan se odnosi na dužinu poruka. Neke stare realizacije ne mogu da obrade poruku dužu od 64 KB. Drugi problem se tiče tajmera. Ako klijent i server imaju različito podešene tajmere, jedan od njih može da odustane u trenutku kada je onaj drugi u najvećem poslu i da raskine vezu. I na kraju, u izvesnim retkim situacijama mogu se aktivirati beskonačne bujice poruka. Na primer, ako računar 1 ima listu slanja A, a računar 2 listu slanja B, pri čemu svaka lista kao odrednicu sadrži onu dragu listu, poruka koja se uputi bilo kojoj od dve liste pokrenuće niz uzastopnih poruka koji nema kraja osim ako ga neko ne uoči i nasilno prekine.

Da bi se prevazišli neki od navedenih problema, RFC dokumentom 2821 definisan je prošireni protokol SMTP (engl. *extended SMTP, ESMTP).* Klijenti koji žele da ga koriste, treba da na početku, umesto poruke *HELO,* pošalju poruku *EHLO.* Ako takva poruka bude odbijena, to znači da je s druge strane običan SMTP server, i da klijent treba da se prikloni uobičajenom protokolu. Ako se, pale, *EHLO* prihvati, dozvoljava se slanje novih komandi i parametara.

7.2.5 Konačna isporuka

Do sada smo pretpostavljali da svi korisnici rade na računalima koji mogu da šalju i primaju e-poštu. Kao što smo videli, e-pošta se isporučuje tako što pošiljalac uspostavi TCP vezu s primaocem, a zatim njom šalje poruku. Taj model je decenijama dobro radio dok su svi ARPANET računari (kasnije i računari na Internetu), u stvari sve vreme bili na mreži i spremni da prihvate TCP vezu.

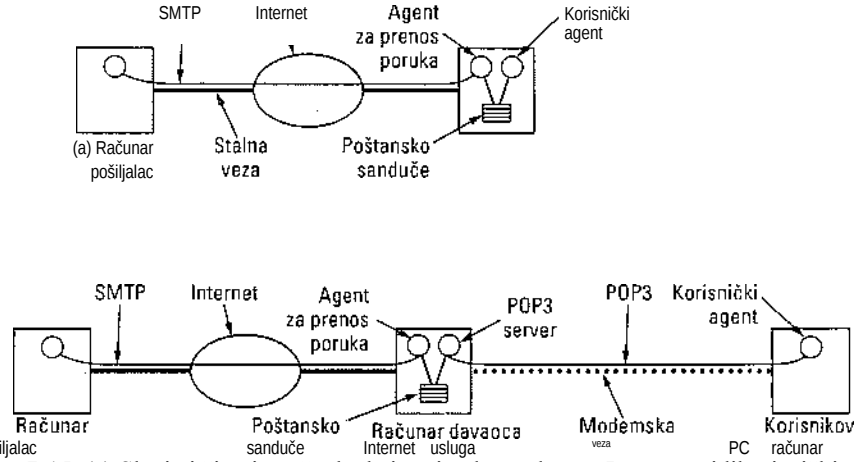
Međutim, s pojavom korisnika koji se na Internet uključuju modemom preko svog davaoca Internet usluga, model je zatajio zbog osnovnog problema: šta raditi kada Dragana želi da pošalje poruku Vesni, a ova trenutno nije na mreži? Dragana ne može da uspostavi vezu s Vesnom i tako ne može da pokrene protokol SMTP.

Jedno rešenje je da agent za prenos poruka na računaru davaoca Internet usluga prihvata poštu za svoje korisnike i da je smešta u njihove poštanske sandučiće na računaru davaoca. Pošto agent može da bude na vezi svih 24 sata, pošta mu se može slati u svako doba dana.

POP3

Predloženo rešenje, nažalost, rađa drugi problem: kako da korisnik preuzme svoju poštu od agenta za prenos poruka na računaru davaoca Internet usluga? I tu je nađeno rešenje u stvaranju novog protokola koji korisničkim agentima za prenos poruka (na njihovim PC računarima) omogućava da stupe u vezu sa agentom za prenos poruka na računaru davaoca Internet usluga i da odatle kopiraju pristigle poruke. Jedan takav protokol je tzv. poštanski protokol verzije 3 (engl. *Post Office Protocol Version 3, POP3),* koji je opisan u RFC dokumentu 1939.

Situacija u kojoj su i pošiljalac i primalac morali stalno biti na Internetu prikazana je na slici 7-15(a). Situacija u kojoj je pošiljalac (trenutno) na mreži, ali primalac nije, prikazana je na slici 7-15(b).



Slika 7-15. (a) Slanje i čitanje pošte kada je primalac stalno na Internetu, idikorisnički agent se izvršava na istom računaru gde i agent za prenos poruka, (b) Gitapje e-pošte kada primalac ima modemsku vezu s davaocem Internet usluga.

Protokol POP3 se aktivira kada korisnik pokrene svoj čitač pošte. Čitač pošte poziva davaoca Internet usluga (osim ako je veza već uspostavljena) i uspostavlja TCP vezu sa agentom za prenos poruka na priključku 110. Po uspostavljanju veze, protokol prolazi redom kroz sledeće tri faze:

1. Ovlašćivanje.
2. Transakcije.
3. Ažuriranje.

U fazi ovlašćivanja (engl. *authorization),* korisnik mora da se prijavi. U fazi transakcija (engl. *transactions),* korisnik preuzima svoju poštu i obeležava je za brisanje iz poštanskog sandučeta. U fazi ažuriranja (engl. *update),* brišu se obeležene poruke. Takvo ponašanje biste mogli posmatrati ako biste upisali nešto slično sledećem:

```
telnet pošta.davalac.com 110
```

gde*pošta.davalac.com.* predstavlja DNS ime poštanskog servera vašeg davaoca Internet usluga. Telnet uspostavlja TCP vezu s priključkom 110, na kome osluškuje POP3 server. Po prihvatanju TCP veze, server tekstualno potvrđuje svoje postojanje. Ta poruka obično počinje sa +OK, iza čega sledi napomena. Na slici 7-16 prikazan je jedan moguć scenario koji počinje po uspostavljanju TCP veze. Kao i ranije, redovi označeni sa *K:* potiču od klijenta (korisnika), a oni označeni sa *S:* - od servera (od agenta za prenos poruka na računani davaoca Internet usluga).

```
                S: +OK POP3 server ready
K: USER vesna
                S: +OK K: PASS vegetables
                S: +OK login successful
K: LIST
                S: 1 2505 S: 2 14302 S: 3
                8122 S:.
K: RETR 1
                S: (šalje poruku 1)
K: DELE 1
K: RETR 2
                S: (šalje poruku 2)
K: DELE 2
K: RETR 3
                S: (šalje poruku 3)
K: DELE 3
K: QUIT

                S: +OK POP3 server disconnecting Slika 7-16. Preuzimanje tri
```

poruke pomoću protokola POP3.

Tokom faze ovlašćivanja, klijent šalje svoje korisničko ime i lozinku. Posle uspe- šnog prijavljivanja, klijent može da pošalje komandu *LIST,* na koju server odgovara spiskom sadržaja poštanskog sandučeta, navodeći dužinu svake poruke u posebnom redu. Spisak se završava tačkom.

Zatim korisnik može da preuzima poruke komandom *RETR* i da ih označava za brisanje komandom *DELE.* Kada preuzme sve poruke (i možda ih označi za brisanje), klijent izdaje komandu *QUIT,* na koju server iz faze transakcija prelazi u fazu ažuriranja. Kada server obriše sve poruke, on šalje odgovor da je izvršio naloženo i raskida TCP vezu.

Iako protokol POP3 omogućava da se sa servera preuzme kopija jedne ili više poruka, a da se originali zadrže na serveru, većina programa za e-poštu samo preuzme svu poštu i isprazni poštansko sanduče. U toj situaciji, jedine kopije poruka postoje na korisnikovom čvrstom disku. Ako on otkaže, sve poruke mogu da nestanu.

Sumirajmo ukratko rad sistema e-pošte za korisnika koji se na Internet uključuje preko davaoca Internet usluga. Dragana sastavlja poruku za Vesnu koristeći neki program za e-poštu

(tj. korisničkog agenta) u kome pritiska određenu ikonicu da bi je poslala. Program za e-poštu predaje poruku agentu za prenos poroka na Draganinom računani. Agent za prenos poroka uočava da je poroka upućena na adresu *ves-* na@xyz.com i koristi sistem DNS da bi pronašao zapis *MX* za domenxyz.com (domen Vesninog davaoca Internet usluga). Ovo pretraživanje vraća DNS ime poštanskog servera domena *xyz.com.* Agent za prenos poruka sada traži IP adresu tog računara koristeći ponovo DNS, pomoću, recimo, procedure *gethostbyname.* On tada uspostavlja TCP vezu sa SMTP serverom na priključku 2.5 ovog računara. Koristeći SMTP komande slične onima prikazanim na slici 7-14, on prenosi poroku u Vesnino poštansko sanduče i raskida TCP vezu.

Posle određenog vremena, Vesna uključuje svoj PC računar, povezuje se sa svojim davaocem Internet usluga i pokreće program za e-poštu. Program uspostavlja TCP vezu sa POP3 serverom na priključku 110 serverskog računara davaoca Internet usluga. DNS ime ili IP adresa ovog računara najčešće su uneti u program za e-poštu prilikom instaliranja ili prijavljivanja davaocu. Po uspostavljanju TCP veze, Vesnin program za e-poštu izvršava protokol POP3 da bi preneo sadržaj poštanskog sandučeta na njen čvrsti disk koristeći komande slične onima na slici 7-16. Kada preuzme svu poštu, program raskida TCP vezu. U stvari, sada se može raskinuti i veza s davaocem Internet usluga, pošto je sva pošta sada na Vesninom čvrstom disku. Naravno, da bi se poslao odgovor, veza se ponovo mora uspostaviti, pa se zato i ne raskida odmah po preuzimanju poruka.

IMAP

Za korisnika s jednim nalogom za e-poštu koji se sa istog PC računara povezuje preko jednog jedinog davaoca Internet usluga, protokol POP3 radi odlično i omiljen je zbog svoje jednostavnosti i stabilnosti. Međutim, čim nešto počne da radi dobro, odmah se - po nepisanom pravilu računarske industrije - pojavljuje neko ko traži još više mogućnosti (s kojima dobija i više problema). To se dogodilo i sa elektronskom poštom. Na primer, mnogi na svom poslu imaju samo jedan nalog za e-poštu, a žele da joj pristupe i kad su na poslu i kad su kod kuće, sa svog prenosivog računara kada su na putu, a i iz Internet kafića kada su na odmoru. Iako protokol POP3 dopušta preuzimanje svih pristiglih poruka pri svakom stupanju u vezu, korisnikove poruke se tako, bez nekog reda rasipaju na više računara, od kojih mu neki čak i ne pripadaju.

Uočeni nedostatak dao je povoda da se napravi protokol za krajnju isporuku, tzv. protokol za pristupanje porukama na Internetu (engl. *Internet Message Access Protocol, IMAP),* koji je definisan u RFC dokumentu 2060. Za razliku od protokola POP3, gde se u osnovi pretpostavlja da će korisnik pri svakom stupanju u vezu „očistiti“ svoje poštansko sanduče i pregledati poruke tek pošto se isključi s mreže, kod protokola IMAP se pretpostavlja da sva e-pošta neograničeno dugo ostaje na serveru u različitim poštanskim sandučićima. IMAP obezbeđuje složene mehanizme za čitanje poruka, čak i delova poruka, što je korisno kada preko sporog modema čitate tekstualni deo poruke s velikim zvučnim i grafičkim prilozima. Pošto je radna pretpostavka da korisnik neće preuzeti poruke da ih stalno čuva na svom računaru, IMAP obezbeđuje mehanizme za sastavljanje i brisanje poruka, kao i za manipulisanje poštanskim sandučićima na serveru. Na taj način, korisnik može da ima zasebno poštansko sanduče za svakog korespodenta i da u njega prebaci pristiglu poruku čim je pročita.

Protokol IMAP nudi mnoge mogućnosti, kao što je mogućnost pristupanja porukama ne

prema rednom broju (kao što je urađeno na slici 7-8), već koristeći atribute (npr. daj mi prvu Perinu poruku). Za razliku od protokola POP3, protokol IMAP, osim toga što isporučuje dolaznu poštu, može i da prihvata odlaznu poštu da bije isporučio na odredište.

Opšti stil rada protokola IMAP sličan je stilu protokola POP3 (slika 7-16), osim što protokol IMAP ima na desetine komandi. IMAP server osluškuje priključak 143. Na slici 7-17 upoređeni su protokoli POP3 i IMAP. Međutim, treba naglasiti da ne podržavaju svi davaoci Internet usluga i svi programi za e-poštu i jedan i drugi protokol. Zbog toga, kada birate program za e-poštu, proverite koji protokol ili protokole on podržava, a takođe proverite da li ga (ih) podržava i vaš davalac Internet usluga.

Osobina	**POP3**	**IMAP**
Gde je definisan protokol	U RFC dokumentu 1939	U RFC dokumentu 2060
TCP priključak koji koristi	110	143
Gde se čuva e-pošta	Na PC računaru korisnika	Na serveru
Kada se čita e-pošta	Posle isključivanja s mreže	Tokom rada na mreži
Potrebno vrerne na vezi	Kratko	Dugo
Korišćenje resursa servera	Minimalno	Intenzivno
Više poštanskih sandučića	Ne	Da
Ko pravi rezervne kopije sadržaja poštanskih sandučića	Korisnik	Davalac Internet usluga
Dobar za pokretne korisnike	Ne	Da

nastavlja se

Osobina	POP3	IMAP
Korisnik upravlja preuzimanjem	Minimalno	U znatnoj meri
Delimično preuzimanje poruka	Ne	Da
Da li su problem kvote na disku	Ne	Vremenom mogu postati
Jednostavan za ugradnju	Da	Ne
Široka podrška	Da	Raste

Slika 7-17. Poređenje protokola POP3 i IMAP.

Različite mogućnosti isporučivanja pošte

Bez obzira na protokol za isporuku pošte (POP3 ili IMAP), mnogi sistemi imaju mogućnost dodatne obrade pristiglih poruka. Među njima je izuzetno korisna mogućnost filtriranja poruka. Filtri su pravila koja se primenjuju kada stigne poruka ili kada se pokrene korisnički agent. Svakim pravilom zadaje se uslov i akcija. Na pri- mer, može da postoji pravilo da se sve poruke od šefa smeštaju u poštansko sanduče broj 1, da se sva pošta od izabrane grupe prijatelja smešta u poštansko sanduče broj 2, a da se svaka poruka s nepristojnim recima u redu Subject odbacuje bez komentara.

Neki davaoci Internet usluga nude filtar koji pristigle poruke automatski razvrstava na važne i neželjene (engl. *spam),* i smešta ih u odgovarajuće sandučiće. Takvi filtri najčešće prvo proveravaju da li je pošiljalac neki poznati „Internet davitelj", a zatim ispituju zaglavlje s predmetom poruke (red Subject). Ako su stotine korisnika upravo primili poruku sa istovetnim redom Subject, verovatno je u pitanju davitelj. Za otkrivanje neželjene pošte koriste se i druge tehnike.

Druga mogućnost koju često obezbeđuje davalac Internet usluga jeste (privremeno) prosleđivanje dolazne poruke na neku drugu adresu. Na toj adresi može čak da bude i računar komercijalne pejdžing službe, koja će korisnika pronaći radio putem ili preko satelita i na njegovom pejdžeru prikazati red *Subject:*.

Još jedna česta mogućnost je instaliranje sistemske usluge „na odmoru sam do" (engl. *vacation daemon).* To je program koji pregleda svaku pristiglu poruku i pošiljaocu šalje standardan odgovor, na primer:

Zdravo. Ja sam na odmoru. Vratiću se 24. avgusta. Odmori se i ti.

Odgovor i ne mora da bude sasvim šablonski, već pošiljaoca može da uputi kako da postupi ako je u pitanju nešto hitno, s kim da stupi u vezu itd. Većina takvih sistemskih usluga vodi evidenciju o tome kome su poslale odgovore, pa ih istim osobama ne šalju ponovo. Kada je takva usluga zaista dobro osmišljena, ona proverava i da li je poruka došla preko liste slanja, a tada uopšte ne odgovara. (Osobe koje tokom leta šalju poruke na velike liste slanja, verovatno ne žele da dobiju stotine odgovora s planovima godišnjih odmora stotina korisnika).

Autor ove knjige upoznao se s jednim ekstremnim oblikom obrade pošte kada je poslao poruku izvesnom Džonu koji tvrdi da dnevno prima 600 poruka. Pravi Džonov identitet nećemo otkrivati da čitaoci ne bi pali u iskušenje da mu odmah pošalju poruku. Neka ostane samo Džon.

Džon je instalirao poštanskog robota koji je svakom novom pošiljaocu odgovarao unapred pripremljenim tekstom da Džon više nije u mogućnosti da lično čita sve svoje poruke i zato je pripremio lični dokument sa odgovorima na često postavljana pitanja (engl. *Frequently Asked Questions, FAQs).* Diskusione grupe uobičajeno koriste takve dokumente - na pitanja se ne

odgovara lično.

U Džonovom FAQ dokumentu navodi se njegova adresa, broj faksa i broj telefona, i objašnjava kako se može stupiti u vezu s njegovom kompanijom. U dokumentu se objašnjava kako se on može angažovati kao predavač, gde se mogu dobiti njegovi radovi i druga dokumenta. Tu su i ukazatelji na softver koji je napisao, na konferenciju koju održava, na standard na kome radi itd. Možda je ovakav pristup neophodan, možda je lični FAQ dokument vrhunski statusni simbol.

Web pošta

Na kraju treba pomenuti i Web poštu (engl. *WebmaiT).* Neke Web lokacije, kao što su Hotmail i Yahoo, obezbeđuju usluge e-pošte za svakog lco ih zatraži, a rade na sledeći način. One imaju normalne agente za prenos poruka koji osluškuju na priključku 25 očekujući dolazne SMTP veze. Da biste pristupili, recimo, Hotmailu, morate da preuzmete njihov DNS zapis *MX,* tako što ćete, na primer, (na UNIX računam) otkucati

```
host -a -v hotmail.com
```

Pretpostavimo da je adresa poštanskog servera *nvclO.hotmail.com.* Ako otkucate

```
telnet mx10.hotmail.com 25
```

moći ćete da uspostavite TCP vezu preko koje se na uobičajen način mogu slati SMTP komande. Do sada ništa novo, osim što su ovi veliki serveri često zauzeti, pa se veza može uspostaviti tek posle nekoliko ponovljenih pokušaja.

Nas najviše zanima ispomka e-pošte. U načelu, kada korisnik pristupi Web strani za e-poštu, suoči se sa obrascem u koji treba da upiše svoje korisničko ime i lozinku. Kada korisnik pritisne dugme Sign In, serveru se šalju korisničko ime i lozinka, i tamo potvrđuju. Ako prijavljivanje uspe, server pronalazi korisnikovo poštansko san- duče i pravi listing sličan onom na slici 7-8, formatiran jezikom HTML u Web stranu. Tada se Web strana šalje čitaču i u njemu prikazuje. Na njoj se mogu pritisnuti mnoge stavke, tako da se poruke mogu čitati, brisati itd.

7.3 WEB - GLOBALNA RAČUNARSKA MREŽA

Globalna računarska mreža (engl. *World Wide Web, WWW)* predstavlja strukturu namenjenu za pristupanje povezanim dokumentima na milionima računara širom Interneta. Tokom samo jedne decenije ona je od alatke za razmenu podataka u oblasti fizike brzih čestica evoluirala u aplikaciju koja milionima ljudi predstavlja sinonim za Internet. Za svoju izuzetnu popularnost Web treba da zahvali grafičkom korisničkom okruženju prepunom boja, u kome se početnici lako snalaze, i obilju informacija koje nudi o svim mogućim temama - od A do Z.

Web (poznat i kao WWW), začet je 1989. u CERN-u, Evropskom centru za nuklearna istraživanja. CERN ima više akceleratora na kojima timovi naučnika iz evropskih zemalja vrše istraživanja iz oblasti fizike brzih čestica. Timovi su često sastavljeni od naučnika iz više evropskih zemalja. Eksperimenti su veoma složeni i zahtevaju višegodišnje planiranje i razvijanje specijalne opreme. Web je izrastao iz potrebe za permanentnom saradnjom međunarodnih timova na razmenjivanju izveštaja, planova, crteža, fotografija i dragih dokumenata.

Mrežu povezanih dokumenata prvi je 1989. predložio fizičar Tim Berners-Li (Tim Berners-Lee) iz CERN-a. Osamnaest meseci kasnije, pušten je u rad prvi (tekstualni) prototip takve

mreže. Decembra 1991, na konferenciji Hypertext '91, održanoj u San Antoniju (Teksas), održana je prva javna demonstracija.

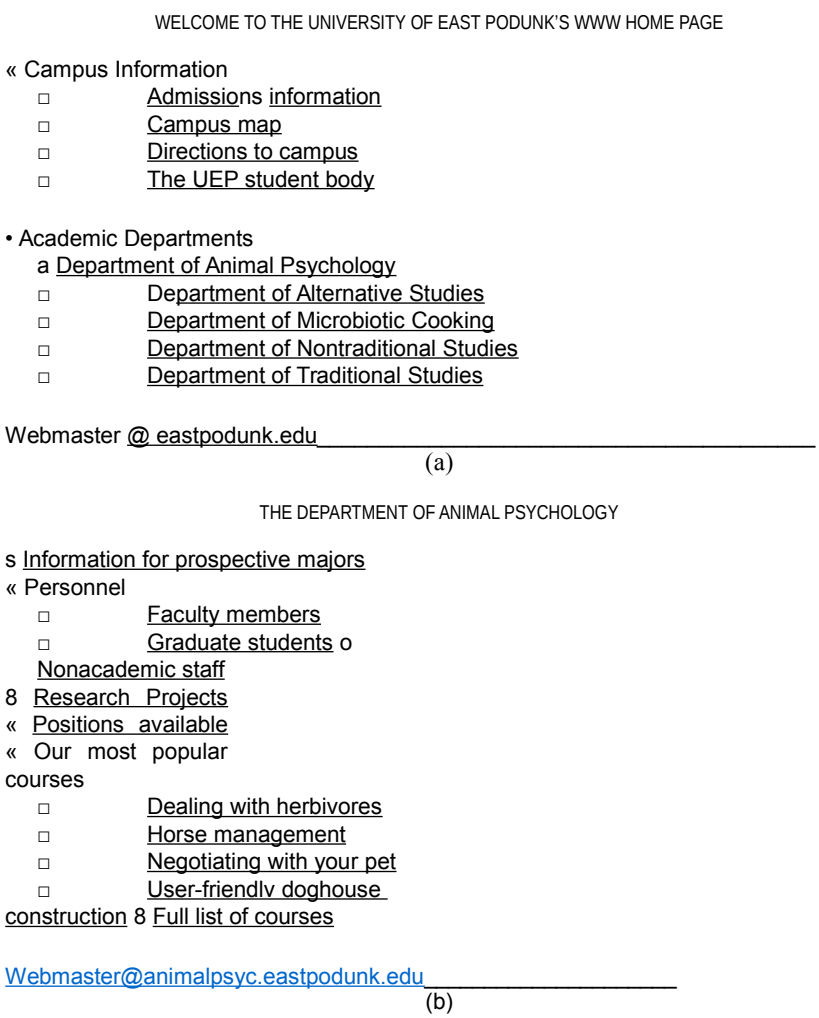
Ta demonstracija i publicitet koji joj je dat privukli su pažnju drugih istraživača, između ostalih i Marka Andresena (Mare Andreesen) sa Univerziteta Ilinois, da započne rad na prvom grafičkom čitaču Weba (engl. *Web browser)* - Mosaicu. Mosaic je pušten u rad februara 1993. Postao je toliko popularan da je Andresen godinu dana kasnije napustio univerzitet da bi osnovao sopstvenu kompaniju, Netscape Communications Corp., sa ciljem da razvija klijente, servere i drugi softver za Web. Kada je kompanija Netscape 1995. ponudila akcije na javnu prodaju, investitori su ih (misleći da bi to mogao da bude novi Microsoft) otkupili za milijardu i po dolara. Takav uspeh zaista iznenađuje, s obzirom na to daje kompanija imala samo jedan proizvod, daje opstajala na ivici propasti i da je najavila da u doglednoj budućnosti ne očekuje da radi profitabilno. Tokom sledeće tri godine, Netscapeov Navigator i Microsoftov Internet Explorer vodili su čuveni „rat čitača Weba", uvodeći bezumnim tempom sve više novih mogućnosti (i sve više programskih grešaka). Godine 1998, America Online je za 4,2 milijarde dolara kupila Netscape Communication Corp., okončavši kratak period nezavisnosti ove kompanije.

Godine 1994, CERN i MIT (Masačusetski tehnički institut) potpisali su sporazum o uspostavljanju **Konzorcijuma za upravljanje Webom** (engl. *World Wide Web Consortium, W3C),* organizacije koja treba da dalje razvija Web, da standardizuje protokole i da ohrabruje saradnju između lokacija. Berners-Li je postao direktor Konzorcijuma. Od tog vremena, Konzorcijumu su se pridružili brojni univerziteti i kompanije. Iako je o Webu napisano nebrojeno knjiga, ako želite da saznate najnovije informacije, treba (naravno) da ih potražite na samom Webu. Matičnu stranu Konzorcijuma naći ćete na adresi www.w3.org. Zainteresovanim čitaocima skrećemo pažnju na hiperveze na toj strani - one će ih odvesti do svih dokumenata i aktivnosti Konzorcijuma.

7.3.1 Pregled arhitekture Weba

S gledišta korisnika, Web je ogromna globalna zbirka dokumenata zvanih **Web strane** (engl. *Web pages)* ili kratko - **strane** (engl. *pages).* Na svakoj strani mogu postojati hiperveze ka drugim stranama bilo gde na svetu. Korisnici slede vezu najčešće tako što je pritisnu mišem i odmah dolaze na odgovarajuću stranu. Taj postupak se može ponavljati u nedogled. Mogućnost da jedna strana ukazuje na drugu, danas zvanu **hipertekst** (engl. *hypertext),* smislio je vizionar Vanevar Buš (Vannevar Bush), profesor elektrotehnike na MIT-u, još 1945. - mnogo pre nego što se pojavio Internet.

Strane se gledaju pomoću programa zvanog **čitač** (engl. *browser),* čiji su popularni primeri Netscapeov Navigator i Microsoftov Internet Explorer. Čitač preuzima traženu stranu, tumači njen tekst i komande za formatiranje, i prikazuje je ispravno formatiranu na ekranu. Jedan primer vidite na slici 7-18(a). Kao i mnoge druge Web strane, i ova počinje naslovom, sadrži neke informacije i završava se adresom e-pošte osobe koja održava stranu. Delovi teksta koji predstavljaju veze ka drugim stranama zovu se **hi- perveze** (engl. *hyperlinks)* i često su istaknuti podvlačenjem, dragom bojom ili i jednim i drugim. Kada želi da sledi vezu, korisnik postavlja pokazivač miša iznad istaknutog područja (pokazivač pri tome promeni oblik) i pritisne taster. Iako postoje i isključivo tekstualni čitači, npr. Lynx, oni nisu tako popularni kao grafički, pa ćemo nadalje govoriti samo o ovim drugim. Pojavljuju se i čitači koji reaguju na glas.

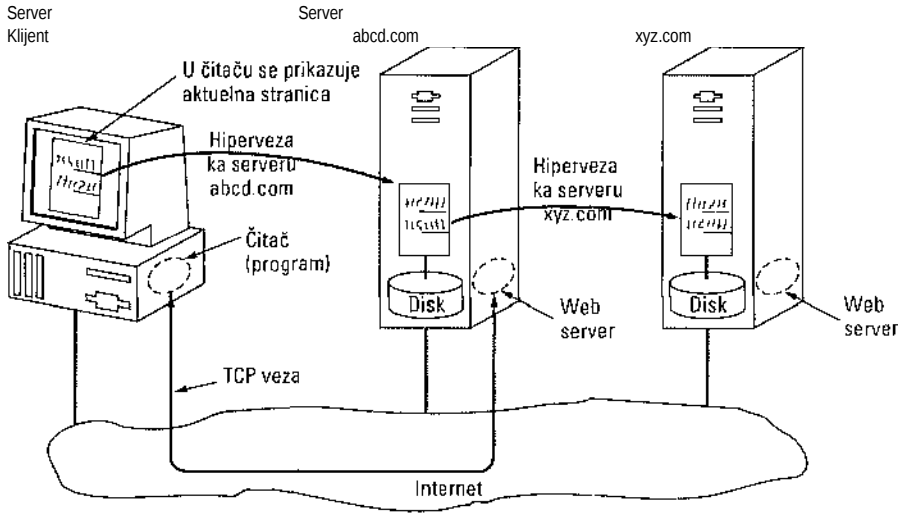


Slika 7-18. (a) Web strana, (b) Strana na koju se stiže kada se pritisne hiperveza Department of Animal Psychology.

Korisnici koje posebno zanima psihologija životinja mogu da pritisnu (podvučenu) hipervezu Department of Animal Psychology. Čitač će preuzeti stranu na koju ukazuje hiperveza i prikazati je, kao na slici 7-18(b). Podvučene stavke na njoj takođe se mogu pritisnuti da bi se pristupilo i drugim stranama itd. Nova strana se može nalaziti na istom računaru kao i prva, ali i na suprotnom kraju sveta. To korisnik ne zna. Čitač samostalno pristupa stranama, bez pomoći korisnika. Ako se korisnik kojim slučajem vrati na glavnu stranu, možda će primetiti da su hiperveze koje je sledio podvučene isprekidanim linijama (i verovatno, drugačije boje) da bi se razlikovale od hiperveza koje još nije koristio. Ako pritisnete red *Campus Information* na glavnoj strani, neće se ništa dogoditi. Taj red nije podvučen, što znači daje u pitanju običan tekst, a ne hiperveza.

Osnovni model rada Weba prikazanje na slici 7-19. Tu čitač prikazuje Web stranu na klijentskom računam. Kada korisnik pritisne red teksta koji je povezan sa stranom na servem *abcd.com,* čitač sledi hipervezu tako što mu šalje zahtev za stranu. Kada strana stigne,

prikazuje se. Ako ova strana ima hipervezu ka strani na servem *xyz.com*, i korisnik je pritisne, čitač tada šalje zahtev tom računaru itd.



Slika 7-19. Delovi modela Weba.

Klijentski deo

Razmotrimo detaljnije klijentski deo modela sa slike 7-19. Čitač je, u načelu, program koji može da prikaže Web stranu i da „oseti" kada se na njoj mišem pritiskaju određene stavke. Kada korisnik izabere stavku, čitač sledi hipervezu i prikazuje odabranu stranu. Prema tome, ugrađena hiperveza mora na neki način ukazivati na tu stranu. Svaka strana na Webu ima tzv. **jedinstvenu adresu resursa** (engl. *Uniform Resource Locator, URL).* Evo tipične URL adrese:

http://www.abcd.com/products.html

Na URL adrese ćemo se vratiti kasnije. Za sada je dovoljno da znate da se URL adresa sastoji iz tri dela: imena protokola *(http),* DNS imena računara na kome se strana nalazi (www.abcd.com) i (obično) imena datoteke sa stranom (*products.html*).

Kada korisnik pritisne hipervezu, čitač preduzima niz koraka da bi preuzeo odabranu stranu. Pretpostavimo da korisnik pretražuje Web i daje na strani o Internet te- lefoniji pronašao hipervezu ka matičnoj strani organizacije ITU (http://www.itu.org/ *home/index.html).* Opišimo šta se događa pošto korisnik izabere ovu vezu.

1. Čitač određuje URL (na osnovu onoga što je korisnik izabrao).
2. Čitač od DNS servera traži IP adresu računara www.itu.org.
3. DNS server odgovara IP adresom 156.106.192.32.
4. Čitač uspostavlja TCP vezu s priključkom 80 na računaru 156.106.192.32.
5. Čitač upućuje zahtev za datoteku */home/index.html.*
6. *Server www.itu.org šalje datoteku* /home/index.html.
7. TCP veza se raskida.
8. Čitač prikazuje sav tekst datoteke */home/index.html.*

9. Čitač preuzima i prikazuje sve slike u ovoj datoteci.

Mnogi čitači, na statusnoj traci u dnu prozora, prikazuju šta trenutno rade. Na taj način, ako postupak dugo traje, korisnik može da sazna da lije to zato što ne odgovara DNS server, zato što ne odgovara Web server ili je u pitanju zagušenje mreže tokom preuzimanja strane.

Da bi mogao da prikaže novu (i svaku drugu) stranu, čitač mora da razume njen format. Da bi svi čitači mogli da razumeju sve Web strane, Web strane se pišu stan- dardizovanim jezikom zvanim HTML, koji ih opisuje. O jeziku HTML govorićemo kasnije u ovom poglavlju.

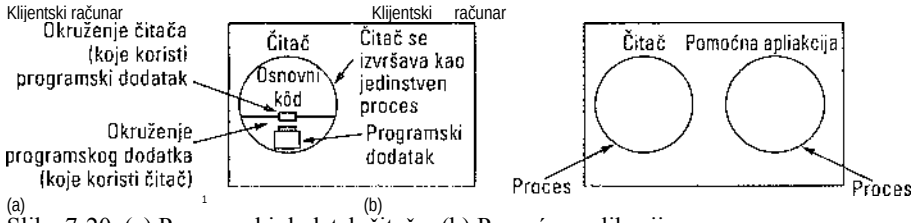
Iako je svaki čitač u osnovi interpretator jezika HTML, većina čitača nudi brojnu dugmad i druge mogućnosti za lalјše snalaženje na Webu. Tu je najčešće dugme Back za vraćanje na prethodno posećenu stranu, dugme Forward za prelazak na sledeću stranu (ukoliko ste je ranije već posetili) i dugme za direktan prelazak na stranu s koje ste krenuli. Većina čitača korisniku daje mogućnost da pomoću određenog dugmeta ili stavke iz menija obeleži Web stranu koja mu je važna, kao i mogućnost prikazivanja spiska obeleženih strana, sa koga se samo jednim pritiskom miša može otići na neku već posećenu stranu. Preuzete Web strane mogu se sačuvati na disku ili odštampati. Obično postoje i mnoge opcije za upravljanje prikazom na ekranu i zadavanje parametara korisničkog profila.

Osim običnog teksta (nepodvučenog) i hiperteksta (podvučenog), Web strane mogu sadržati i ikonice, crteže, mape i fotografije. Svaka od ovih stavki može se (po želji) vezati za neku drugu Web stranu. Tada ona postaje osetljiva na pritisak mišem i može da vas odvede na neku drugu Web stranu, baš kao kada mišem pritisnete hi- pertekst. Kod fotografija i mapa, nova prikazana strana može da zavisi od toga koji ste deo slike ili mape pritisnuli.

Nisu sve strane pisane jezikom HTML. Strana može da bude i dokument 11 formatu PDF, ikonica u formatu GIF, fotografija u formatu JPEG, muzička datoteka u formatu MP3, video datoteka u formatu MPEG ili dokument u nekom od stotina dragih formata. Pošto se standardne HTML strane mogu povezati sa svakim od tako formatiranih dokumenata, čitač će imati teškoća da protumači stranu koju ne razume.

Umesto da se čitači stalno proširuju novim interpretatorima za formate koji se pojavljuju skoro svakog dana, izabrano je opštije rešenje. Kada server pošalje zahtevanu stranu, on s njom šalje i dodatnu informaciju koja sadrži MIME tip strane (slika 7-12). Strane tipa *text/html* prikazuju se direktno, kao i strane još nekih unapred ugrađenih tipova. Ako MIME tip ne spada u ugrađene tipove, čitač u svojoj tabeli MIME tipova traži savet kako da je prikaže. U toj tabeli, MIME tipovi su nabrojani uporedo sa odgovarajućim programima za prikazivanje.

Postoje dve mogućnosti: programski dodaci i pomoćne aplikacije. Programski dodatak (engl. *plug-in)* predstavlja modul s kodom koji čitač preuzima iz specijalnog direktorijuma 11a disku i instalira kao sopstveno proširenje, kao na slici 7-20(a). Pošto se programski dodaci izvršavaju unutar čitača, oni mogu da pristupe aktuelnoj strani da bi izmenili njen izgled. Kada programski dodatak obavi posao (obično nakon što korisnik pređe na neku dragu Web stranu), on se uklanja iz memorije čitača.



Slika 7-20. (a) Programski dodatak čitaču, (b) Pomoćna aplikacija.

Svi programski dodaci moraju imati ugrađen skup procedura za svaki čitač kako bi ovaj mogao da ih poziva. Na primer, tipična je procedura kojom osnovni kôd čitača programskom dodatku prosleđuje podatke za prikazivanje. Taj skup procedura predstavlja okruženje programskog dodatka koje je različito za svaki čitač.

Osim toga, čitač pravi i sopstveni skup procedura preko kojih pruža usluge programskom dodatku. Tipičnim procedurama iz okruženja čitača dodeljuje se i oslobađa memorija, prikazuju se poruke na statusnoj traci čitača i pronalaze parametri čitača.

Programski dodatak se pre korišćenja mora instalirati. Postupak obično ide tako što korisnik odlazi na Web lokaciju gde se nalazi programski dodatak i preuzima njegovu instalacionu datoteku. U Windowsu, to je najčešće samoraspakujuća zip datoteka s nastavkom *.exe.* Kada zip datoteku dvaput pritisnete mišem, izvršiće se kratak program koji joj je pridružen. Taj program raspakuje programski dodatak i kopira ga u direktorijum u kome čitač drži programske dodatke. Tada poziva odgovarajuću procedura da bi registrovao njegov MIME tip i pridružio mu ga. U UNIX-u, program za instaliranje je obično komandni skript koji kopira i registraje programski dodatak.

Drugi način za proširivanje mogućnosti čitača jeste pomoćna aplikacija (engl. *helper application).* To je pravi program koji se izvršava kao zaseban proces, što je prikazano na slici 7-20(b). Postoje pomoćna aplikacija zaseban program, ona čitaču ne nudi svoje okruženje, niti koristi njegove usluge. Umesto toga, ona obično prihvata ime privremene datoteke u kojoj se nalazi datoteka sa sadržajem, otvara je i prikazuje sadržaj. Pomoćne aplikacije su najčešće velild programi, potpuno nezavisni od čitača, npr. Adobeov Acrobat Reader za prikazivanje PDF datoteka ili Microsoftov Word. Za neke programe (kao Acrobat) postoje programski dodaci koji sami pozivaju pomoćnu aplikaciju.

Mnoge pomoćne aplikacije koriste MIME tip *application.* Definisano je i prilično podtipova, na primer, *application/pdf* za PDF datoteke i *application/msword* za Wordove datoteke. Na taj način, URL adresa može da ukaže direktno na PDF ili Wordovu datoteku. Kada je korisnik pritisne mišem, automatski se pokreću Acrobat, odnosno Word, i predaje im se ime privremene datoteke sa sadržajem koji treba prikazati. Shodno tome, čitači se mogu podesiti za prikazivanje praktično neograničenog broja tipova dokumenata, a da se sami ne proširuju. Konfiguracije savremenih Web servera sadrže stotine kombinacija tip/podtip i stalno se dodaju nove kako se instaliraju novi programi.

Pomoćna aplikacija ne mora da koristi samo MIME tip *application.* Adobeov Photoshop, na primer, koristi i tip *image/x-photoshop,* a RealOne Player - tip *audio/mp3.*

U Windowsu, kada se instalira program, istovremeno se registruju i MIME tipovi s kojima on „želi“ da radi. To izaziva sukobljavanje u slučajevima kada za isti podtip, npr. za podtip *video/mpg,* postoji više programa za prikazivanje. Sve se završava time što poslednji instalirani program upiše sebe preko postojeće odrednice koja povezuje MIME tip sa odgovarajućom

pomoćnom aplikacijom i tako je nasilno izgura iz igre. Zbog toga, pri instaliranju novog programa može da se promeni način na koji čitač radi s postojećim tipovima podataka.

U UNIX-u registrovanje obično ne ide automatski. Korisnik mora da ručno ažurira određene konfiguracione datoteke. Takav pristup zahteva više rada, ali i sprečava neprijatna iznenađenja.

Umesto da preuzimaju datoteke sa udaljenih Web servera, čitači mogu da otvaraju i lokalne datoteke. Pošto lokalne datoteke ne sadrže MIME tipove, čitač na neki način mora da pronađe programski dodatak ili pomoćnu aplikaciju da bi prikazao tipove podataka koji se razlikuju od tipova ugrađenih u njega, kao što su *text/html* i *image/ jpeg*. Za rad s lokalnim datotekama pomoćna aplikacija može da se pridruži nastavku imena datoteke, kao i njenom MIME tipu. Uz standardan način pridruživanja, *doLpdf* će se otvoriti u Acrobatu, a datoteka *tekst.doc* u Wođu. Neki čitači, određujući pomoćnu aplikaciju za prikazivanje, osim nastavka imena datoteke koriste i MIME tip, čak i informacije iz same datoteke. Naglasimo na kraju da se Internet Explorer, kad god može, više oslanja na nastavak imena datoteke, nego na MIME tip.

I ovde može da dođe do sukobljavanja pošto su mnogi programi voljni, čak željni da obrade datoteku s nastavkom, recimo, *.mpg*. Pri instaliranju programa namenjenih profesionalcima, često se nudi izbor MIME tipova i nastavaka datoteka koje program može da obradi, tako da korisnik ručno može da ih izabere vodeći računa da time ne poremeti postojeće veze između tipova podataka i pomodnih aplikacija za njihovo prikazivanje. Kod programa namenjenih običnim korisnicima, obično se pretpostavlja da korisnik baš ne poznaje MIME tipove, pa zato ti programi pri instaliranju grabe sve što mogu, bez obzira na to šta je ranije instalirano.

Mogućnost proširivanja čitača mnogim novim tipovima - iako zgodna - može da donese i nevolje. Kada Internet Explorer preuzme datoteku s nastavkom. *exe,* on zna da je to izvršna datoteka i da za nju ne postoji pomoćna aplikacija (za prikazivanje). Očito je da treba samo da je pokrene. Međutim, to može da bude veliki bezbednosni previd. Zlonamerna Web lokacija treba samo da napravi Web stranu sa slikama, recimo, filmskih zvezda ili sportskih asova i da ih sve poveže s virusom. Dovoljno je da takvu sliku pritisnete i preuzećete na svoj računar i pokrenuti zloćudni program. Da bi se sprečio ulazak takvim nezvanim gostima, Internet Explorer može da se podesi da ne izvršava automatski programe koje ne poznaje, ali ne znaju svi korisnici kako se to radi.

U TJNIX-u postoji sličan problem s komandnim skriptovima, ali to od korisnika zahteva da svesno instalira komandno okruženje kao pomoćnu aplikaciju. To instaliranje je, srećom, dovoljno složeno da niko ne može nehotice da ga izvede (a malo ih je kojima to uspe i kada se trude).

Serverski deo

Toliko o klijentu. Razmotrimo sada kako to izgleda kod servera. Kada korisnik upiše URL adresu ili pritisne hipertekst na Web strani, već smo videli da čitač tada analizira adresu i njen deo između http://i sledeće kose crte tumači kao DNS ime čiju IP adresu treba naći. Dobivši IP adresu servera, čitač uspostavlja TCP vezu s njegovim priključkom 80, a zatim mu šalje komandu koja sadrži ostatak URL adrese - ime datoteke na tom serveru. Server tada vraća datoteku čitaču i ovaj je prikazuje.

Web server grubo liči na server čiji je kod prikazan na slici 6-6. I tom serveru se daje ime datoteke koju treba da nađe i pošalje. Koraci koje server preduzima u svojoj glavnoj petlji u oba slučaja su:

1. Prihvatanje TCP veze s klijentom (čitačem).
2. Dobijanje imena zahtevane datoteke.
3. Preuzimanje datoteke (s diska).
4. Slanje datoteke klijentu.
5. Raskidanje TCP veze.

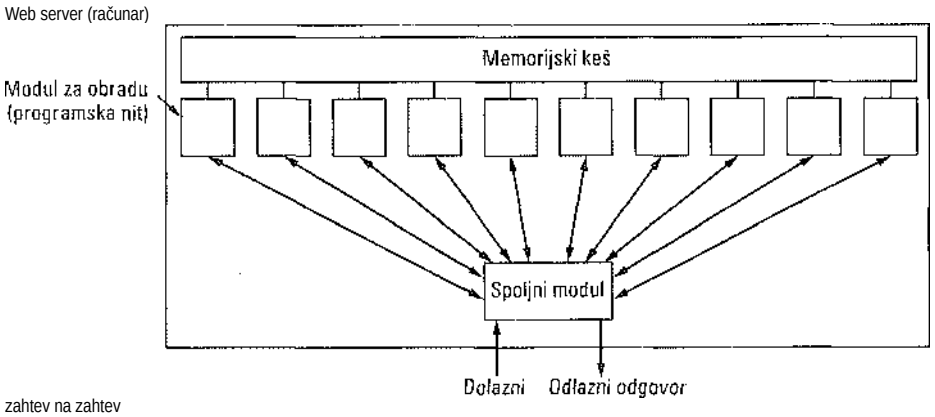
Savremeni Web serveri imaju više mogućnosti, ali u osnovi rade na isti način.

Mana opisanog postupka je što se pri svakom zahtevu mora pristupati disku da bi se preuzela datoteka. Zbog toga Web server može da obradi onoliko zahteva u sekundi koliko puta može da pristupi disku. Najsavremeniji SCSI disk ima vreme pristupa od oko 5 ms, što ograničava brzinu rada servera na najviše 200 zahteva u sekundi, a i na manje ako se često traže velike datoteke. Za veće Web lokacije, takva brzina rada ne zadovoljava.

Neposredno poboljšanje (koje koriste svi Web serveri) postiže se memorijskim keširanjem *n* poslednjih zahtevanih datoteka. Pre nego što pristupi disku, server će prvo pretražiti keš. Ako u njemu nađe datoteku, može je poslati direktno iz memorije.

Iako je za efikasno keširanje potrebna velika glavna memorija i nešto dodatnog vremena za pretraživanje i održavanje keša, ukupne vremenske uštede su tolike da se taj višak potrošenog vremena i finansijska ulaganja skoro uvek isplate.

Sledeći korale ka ubrzavanju servera jeste uvođenje višenitnog rada. U jednoj varijanti, server se sastoji od spoljnog modula (engl. *front-end-module)* koji prihvata sve dolazne zahteve i *k* modula za obradu (engl. *processing module),* kao na slici 7-21. Svih *k*+ 1 programskih niti pripadaju istom procesu tako da svi moduli za obradu imaju pristup kešu unutar adresnog prostora procesa. Kada stigne zahtev, spoljni modul ga prihvata i opisuje ga kratkim zapisom. Zatim zapis predaje jednom od modula za obradu. U drugoj mogućoj varijanti, spoljni modul ne postoji i svaki modul za obradu pokušava da preuzme zahtev, ali je tu potreban i protokol za blokiranje modula kako ne bi došlo do sukobljavanja.



Slika 7-21. Višenitni Web server sa spoljnim modulom i modulima za obradu zahteva.

Zahtevanu datoteku modul za obradu najpre traži u kešu. Ako je datoteka tamo, ažurira zapis tako što postavlja pokazivač na nju. Ako datoteka nije tamo, modul počinje postupak učitavanja datoteke s diska u keš (i možda briše neke ranije keširane datoteke da bi za nju napravio mesta). Kada se datoteka preuzme s diska, istovremeno se smešta u keš a njena kopija

se šalje klijentu.

Premda su neki moduli blokirani jer čekaju da se završi operacija učitavanja s diska (ali pri tome ne troše procesorsko vreme), ova šema ima tu prednost što drugi moduli istovremeno mogu da aktivno obrađuju zahteve. Naravno, da biste izvukli neku stvarnu korist iz višenitnog modela, neophodno je da imate više diskova, tako da istovremeno više njih može da bude zaposleno. Uz *k* modula za obradu i *k* diskova, ukupan protok podataka kroz sistem može se povećati *k* puta u odnosu na jednonitni server i jedan disk.

I jednonitni server sa *k* diskova teorijski bi mogao da ubrza rad *k* puta, ali su tu sam kod i administriranje mnogo složeniji jer se za pristupanje diskovima ne može koristiti uobičajen sistemski poziv blokirajućoj proceduri READ. Procedura READ može se koristiti na višenitnom servera jer tu blokira samo programsku nit iz koje je pozvana, a ne čitav proces.

Osim prihvatanja imena datoteka i njihovog isporučivanja, savremeni Web serveri rade još štošta drugo. U stvari, obrada svakog zahteva može da ispadne veoma složena. Iz tog razloga, svaki modul za obradu na mnogim serverima izvršava niz koraka. Spoljni modul dolazni zahtev prosleđuje prvom raspoloživom modulu, a ovaj ga zatim obrađuje prema sledećoj šemi, potpuno ili delimično - u zavisnosti od toga šta je sve potrebno za konkretan zahtev.

1. Razrešavanje imena zahtevane Web strane.
2. Provera identiteta klijenta.
3. Provera pravila pristupanja klijentu.
4. Provera pravila pristupanja Web strani.
5. Provera lceša.
6. Preuzimanje zahtevane strane s diska.
7. Određivanje MIME tipa koji se uključuje u odgovor.
8. Sređivanje raznih sitnica.
9. Vraćanje odgovora klijentu.
10. Beleženje usluge u dnevnik.

Prvi korak je neophodan jer zahtev možda ne sadrži ime datoteke kao doslovan tekst. Na primer, razmotrite URL adresu http://www.cs.vu.nl, koja ne sadrži ime datoteke. Ona se mora proširiti nekim podrazumevanim imenom datoteke. Isto tako, savremeni čitači mogu da naznače podrazumevani govorni jezik korisnika (npr. italijanski ili engleski), na osnovu čega server može da odabere odgovarajuću verziju Web strane, ukoliko takva verzija postoji. Proširivanje imena u načelu nije jednostavno kao što izgleda, jer postoje brojna pravila o imenovanju datoteka.

U drugom koraku proverava se identitet korisnika. To je potrebno zbog Web strana koje nisu namenjene svima. Kasnije, u nastavku poglavlja, pokazaćemo kako se to radi.

U trećem koraku, pošto se sada zna identitet i lokacija klijenta, proverava se mogućnost odgovaranja na zahtev. U četvrtom koraku proveravaju se moguća ograničenja u vezi sa samom stranom. Ako u direktorijumu u kome se nalazi tražena datoteka postoji i izvesna druga datoteka (npr. datoteka s nastavkom *.htciccess),* ona može da ograniči slanje tražene datoteke samo na izvesne domene, na primer, samo korisnicima unutar kompanije.

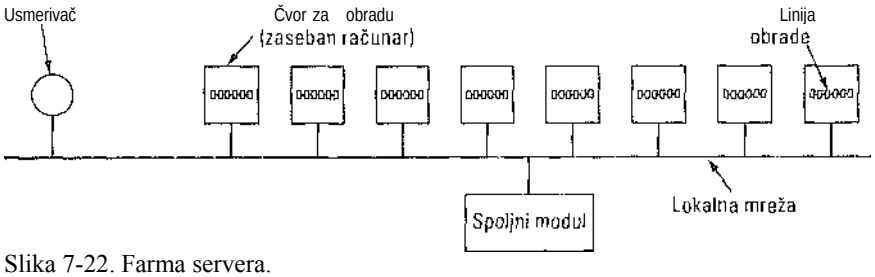
U petom i šestom koraku datoteka se preuzima iz keša ili sa diska. U šestom koraku mora postojati mogućnost istovremenog učitavanja s više diskova.

MIME tip se u sedmom koraku određuje na osnovu nastavka imena datoteke, prvih nekoliko

reči iz datoteke, na osnovu konfiguracione datoteke, a možda i na osnovu drugih izvora. U osmom koraku obavljaju se različiti sporedni poslovi, kao stoje pravljenje korisničkog profila ili prikupljanje statistike.

U devetom koraku rezultat se šalje, a u desetom se obavljen posao beleži u sistemski dnevnik za administrativne svrhe. Kasnije se u takvim dnevnicime mogu naći vredne informacije o ponašanju korisnika, na primer, redosled kojim korisnici pristupaju stranama.

Alco svake sekunde stiže previše zahteva, mikroprocesor neće moći sve da ih obradi, bez obzira na broj paralelno vezanih diskova. Rešenje je u tome da se napravi više čvorova (računara), možda s repliciranim diskovima, da sada diskovi ne bi postali usko grlo. To vodi modelu farme servera (engl. *serverfarm)* sa slike 7-22. Spoljni modul i dalje prihvata sve zahteve, ali ih više ne raspodeljuje u programske niti, već u niz mikroprocesora, rasterećujući tako svaki od njih. Pojedini računari mogu, kao i ranije, imati više programskih niti, svaku sa navedenom linijom obrade.

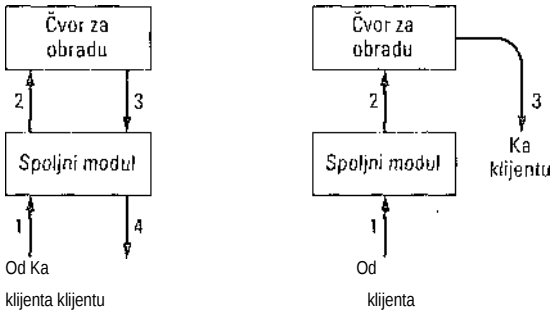


Slika 7-22. Farma servera.

U farmi servera postoji problem što nema zajedničkog kesa, već svaki čvor za obradu održava sopstveni keš - osim ako se ne upotrebi skup računar s paralelnim procesorima i deljenom memorijom. Problem se može umanjiti alco spoljni modul be- leži kome šalje koji zahtev i kasnije zahtev za istu stranu šalje istom računani. Tako svaki čvor postaje specijalista za određene strane, a keš svakog računara ne zatrpava se svim mogućim datotekama.

Problem s farmama servera predstavlja i to što se klijentska TCP veza završava u spoljnom modulu, pa i odgovor mora da ide kroz njega. Ta situacija je prikazana na slici 7-23(a), gde i dolazni zahtev (1) i odlaziti odgovor (2) prolaze kroz spoljni modul. Ponekada se ovaj problem zaobilazi trikom zvanim predavanje TCP upravljanja (engl. *TCP handoff).* Tu se završna TCP tačka provlači sve do modula za obradu, tako da ovaj direktno može da odgovori klijentu, što je prikazano strelicom (3) na slici 7-23(b). Predavanje TCP upravljanja zaklonjeno je od klijenta.

Slika 7-23. (a) Normalan put zahteva i odgovora, (b) Put zahteva i odgovora uz predavanje TCP upravljanja.


Čvor za obradu
2
3
Spoljni modul
1
4
Od Ka klijenta klijentu
(a)
Čvor za obradu
2
3
Ka klijentu
Spoljni modul
1
Od klijenta
(b)

URL - Jedinstvena adresa resursa

Više puta smo ponovili da Web strane mogu da sadrže pokazivače na druge Web strane. Razmotrimo sada detaljnije kako su ti pokazivači ugrađeni. Kada je Web nastao, odmah je bilo jasno da se mora obezbediti mehanizam za imenovanje i lociranje strana. Zapravo, prikazivanje bilo koje Web strane zahtevalo je prethodne odgovore na sledeća pitanja:

1. Kako se strana zove?
2. Gde se strana nalazi?
3. Kako se strani može pristupiti?

Kada bi svaka strana imala jedinstveno ime, mogle bi se nesumnjivo identifikova- ti. Pa ipak, problem time ne bi bio rešen. Razmotrite sledeću analogiju između ljudi i Web strana. U SAD, skoro svako ima broj socijalnog osiguranja koji jedinstveno identifikuje osobu pošto ni dve osobe nemaju isti broj. Pa ipak, ako imate broj nečijeg socijalnog osiguranja, nema načina da dođete do njegove (njene) adrese, pogotovu ne možete pogoditi da li toj osobi treba pisati na engleskom, španskom ili kineskom jeziku. Web u osnovi ima iste probleme.

Izabrano rešenje identifikuje Web strane na način kojim se jednim potezom otklanjaju sva tri problema. Svakoj strani se dodeljuje **jedinstvena adresa resursa** (engl. *Unijom Resource Locator, URL),* koja služi kao jedinstveno ime strane i važi za ceo svet. URL identifikatori imaju tri dela: protokol (poznat i kao **šema),** DNS ime računara na kome se strana nalazi i lokalno ime koje jedinstveno identifikuje određenu stranu (to je obično ime datoteke na računani na kome se strana nalazi). Na primer, Web lokacija katedre na kojoj radi autor sadrži više video filmova o univerzitetu i samom Amsterdamu. URL adresa strane s video filmovima glasi

http://www.cs.vu.nl/video/index-en.html

Taj URL ima tri dela: protokol (*http*), DNS ime računara (www.cs.vu.nl) i ime datoteke (*video/index-en.html*), sa odgovarajućom interpunkcijom između delova. Ime datoteke je dato kao relativna putanja u odnosu na podrazumevani Web direktorijum na računaru *cs.vu.nl.*

Mnoge lokacije imaju ugrađene skraćenice za imena datoteka. Na mnogim lokacijama, prazno ime datoteke podrazumeva glavnu matičnu stranu organizacije. Ako navedeno ime predstavlja direktorijum, to najčešće podrazumeva datoteku *index.ht- ml.* Na kraju, ime *~korisnik/*može da se preslika u korisnički WWW direktorijum, a zatim u datoteku *index.html* u njemu. Na taj način, autorovoj matičnoj strani može se pristupiti preko adrese

http://www.cs.vu.ni/~ast/

iako je stvarno ime datoteke *index.html,* a ona se nalazi u određenom podrazumeva- nom direktorijumu.

Sada možemo da detaljnije ispratimo kako radi hipertekst. Da bi neki deo teksta reagovao na pritisak mišem, autor strane mora da uradi dvoje: treba da prikaže tekst koji se može pritisnuti mišem i da obezbedi URL strane na koju se prelazi kada se tekst pritisne. Sintaksu komandi objasnicemo kasnije.

Kada se izabere tekst, čitač traži ime računara pomoću sistema DNS. Kada sazna IP adresu računara, čitač s računarom uspostavlja TCP vezu. Preko te veze on šalje ime datoteke koristeći navedeni protokol. Opa! Eto Web strane.

Opisana šema URL adresiranja elastična je po tome što čitači mogu da koriste različite protokole da bi pristupili različitim resursima. U stvari, definisane su URL adrese i za razne

druge uobičajene protokole. U nešto uprošćenijem vidu, nekoliko takvih adresa prikazano je slici 7-24.

Ime	Koristi se za	Primer
http	Hipertekst (HTML)	http://www.es. vu.nl/~ast/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Lokalne datoteke	file:///usr/suzana/prog.c
news	Diskusione grupe	newsreomp.os.minix
news	Poruke diskusionih grupa	news:AA0134223112@cs.utah.edu
gopher	Gopher	gopher://gopher.tc.umn.edu/11/Libraries
mailto	Slanje e-pošte	mailto:Jovan@acm.org
telnet	Daljinsko prijavljivanje	telnet://www.w3. org:80

Slika 7-24. Neke uobičajene URL adrese.

Pregledajmo brzo tu listu. Protokol *http* je „maternji“ ježile Weba; njime se spora- zumevaju Web serveri. On se zove **protokol za prenos hiperteksta** (engl. *HyperText Transfer Protocol, HTTP*) - otuda ono *http*. O njemu čemu govoriti više u dragom delu poglavlja.

Oznaka *ftp* znači da se datotekama pristupa protokolom FTP - protokolom za prenos datoteka na Internetu. FTP koji postoji i radi već više od dve decenije i dalje se dobro drži. Brojni FTP severi širom sveta omogućavaju svakom na Internetu da se prijavi i preuzme bilo koju datoteku koja se nalazi na FTP servera. Web ovo nije pro- menio; samo je pomogao da se datoteke lakše preuzimaju jer je okruženje protokola FTP pomalo nerazumljivo (ali je protokol moćniji od protokola HTTP jer, na primer, omogućava korisniku računara A da prenese datoteku s računara *B* na računar C).

Lokalnoj datoteci možete pristupiti kao Web strani bilo protokolom *file* ili, jednostavnije, ako je samo imenujete. Postupak je sličan protokolu FTP, samo bez servera. Naravno, radi samo za lokalne datoteke, ne i za udaljene.

Davno pre Interneta postojao je sistem diskusionih grupa USENET. On se sastojao od oko 30.000 grupa u kojima su milioni korisnika razgovarali o najrazličitijim temama tako što su objavljivali i čitali poruke u odgovarajućim diskusionim grupama. Protokol *news* možete iskoristiti da prikažete poruke diskusionih grupa kao Web strane. To znači daje Web čitač istovremeno i čitač poruka diskusionih grupa. U stvari, mnogi čitači imaju dugmad ili stavke u meniju pomoću kojih se poruke diskusionih grupa čitaju lakše nego u njihovom originalnom čitaču.

Za protokol *news* postoje dva formata. Prvim se zadaje diskusiona grupa i on se može iskoristiti za dobijanje spiska poruka s lokacije posvećene diskusionim grupama koja je unapred naznačena u čitaču. Za drugi format neophodan je identifikator tražene poruke, u našem primera to je AA0134223112@cs.utah.edu. Čitač tada preuzima traženu pontku sa unapred konfigurisane lokacije posvećene diskusionim grupama pomoću **protokola za prenos poruka diskusionih grupa** (engl. *Network News Transfer Protocol, NNTP*). O tom protokolu nećemo više govoriti u ovoj knjizi, dovoljno je reći da on liči na protokol SMTP.

Protokol *gopher* koristi sistem Gopher, stvoren na Univerzitetu Minesote i nazvan tako prema simpatičnom glodani - maskoti univerzitetskih sportskih timova, a i „gopher“ u žargonu znači „go for“, tj. „idi i uzmi“. Gopher je nastao više godina pre Weba. To je sistem za

preuzimanje informacija, koncepcijski sličan Webu, ali samo za preuzimanje teksta (bez slika). Danas je zastareo i retko se koristi.

Poslednja dva protokola u stvari ne služe za preuzimanje Web strana, ali su ipak korisni. Protokol *mailto* omogućava korisnicima da pošalju e-poruku iz čitača Weba. Treba da pritisnu dugme OPEN i da zadaju URI, koji se sastoji od reči *mailto:,* iza koje sledi adresa e-pošte primaoca. Čitači uglavnom reaguju tako što otvore podrazu- mevani program za e-poštu s već upisanom adresom primaoca i popunjenim još nekim poljima zaglavlja.

Protokol telnet omogućava povezivanje sa udaljenim računarom. Koristi se na isti način kao i program telnet, što ne iznenađuje, jer većina čitača poziva taj program kao pomoćnu aplikaciju.

Sve u svemu, URL adrese omogućavaju korisnicima ne samo da putuju Webom, već i da rade sa FTP protokolom, diskusionim grupama, Gopherom, e-poštom i tel- netom, što čini nepotrebnim specijalizovana korisnička okruženja za te druge usluge jer se gotovo sve čemu se može pristupiti na Internetu može doseći iz čitača Weba. Da ne znamo da je začetnik ove ideje fizičar, pomislili bismo da je ona potekla iz reklamnog odeljenja neke softverske kompanije.

Uprkos svim pobrojanim divnim mogućnostima, s porastom Weba izišla je na vi- delo ijedna urođena slabost sistema URI- adresa. URL adresa ukazuje na jedan određeni računar. Ako se neke Web strane mnogo traže, zgodnije bi bilo da se njihove kopije razmeste na više međusobno udaljenih lokacija kako bi se smanjio saobraćaj na Internetu. Problem je u tome što sistem URL adresa ne može da ukaže na Web stranu a da istovremeno ne naznači i gde se ona nalazi. Ne može se reći: Tražim stranu xyz, i nije me briga odakle ćeš je dobaviti. Da bi razrešila ovaj problem i omogućila repli- kovanje Web strana, grupa DETF počela je da razrađuje sistem **jedinstvenih imena resursa** (engl. *Universal Resource Names, URNs).* URN ime možete zamisliti kao uopštenu URL adresu. Ta tema je još uvek predmet istraživanja, iako je sintaksa URN imena definisana u RFC dokumentu 2141.

Nepostojanje stanja i kolačići

Kao što smo se već više puta uverili, Web je sistem u kome ne postoje stanja. Tu ne postoji pojam sesije prijavljivanja. Čitač šalje zahtev serveru i od njega dobija datoteku. Posle toga, server zaboravlja da je ikada stupio u vezu s tim klijentom.

Na samom početku, kada je Web uglavnom služio za preuzimanje javno dostupnih dokumenata, takav model rada bio je sasvim zadovoljavajući. Ali kada je Web počeo da se koristi i za druge svrhe, počeli su da se javljaju problemi. Na primer, neke Web lokacije zahtevaju da se korisnici registruju (možda i da plate) da bi mogli da ih koriste. To pokreće pitanje kako server razlikuje zahteve registrovanih korisnika od za- hteva onih koji se nisu registrovali. Drugi primer nalazimo u oblasti e-trgovine. Ako korisnik „šparta“ virtuelnom prodavnicom i s vremena na vreme stavlja artikle u korpu, kako server evidentira sadržaj korpe? Treći primer pružaju prilagodljivi Web portali kao što je Yahoo. Korisnici mogu da svoju polaznu Web stranu detaljno podese tako da sadrži samo informacije koje njih zanimaju (npr. stanje njihovih deonica na tržištu i njihove omiljene sportske timove), ali kako server može da prikaže odgovarajuću stranu ako ne zna ko je korisnik?

U prvi mah biste pomislili da server može da evidentira korisnike prema njihovim IP adresama. Međutim, taj pristup je promašaj. Pre svega, mnogi rade na višekorisničkim računalima, naročito kad su na poslu, a IP adresa identifikuje samo računar, ne i korisnika. Drugo, što je možda i gore, mnogi davaoci Internet usluga koriste sistem za prevođenje

mrežnih adresa NAT, tako da odlazni paketi svih korisnika nose istu IP adresu. S gledišta servera, hiljade korisnika istog davaoca Internet usluga imaju istu IP adresu.

Da bi razrešio opisani problem, Netscape je uveo toliko kritikovane kolačiće (engl. *cooki.es)*. Samo ime potiče iz prastarog programerskog žargona i označava situaciju kada je program pozivao proceduru i od nje dobijao nešto što joj je morao kasnije ponovo pokazati da bi mu obavila neki posao. U tom istom smislu, deslcriptor datoteka u UNIX-u i identifikator objekta u Windowsu mogu se smatrati kolačićima. Kolačići su kasnije formalno definisani u RFC dokumentu 2109.

Kada klijent zatraži Web stranu, server s njom može da pošalje i dodatne informacije. Ta informacija može da bude i u obliku kolačića - male datoteke (najviše 4 KB) ili teksta. Čitači čuvaju ponuđene kolačiće u direktorijumu namenjenom njima na kli- jentovom čvrstom disku, osim ako korisnik onemogući njihovo prihvatanje. Kolačići su samo datoteke ili tekst, oni se ne izvršavaju. Kolačić u načelu može da sadrži virus, ali pošto se s kolačićima postupa kao s podacima, zvanično ne postoji mogućnost da se virus pokrene i napravi štetu. Međutim, uvek postoji rizik da će zlonamerni haker iskoristiti neki propust u kodu čitača da bi aktivirao virus.

Kolačić može da sadrži do pet polja (slika 7-25). Polje *Domen* saopštava odakle je došao kolačić. Pretpostavlja se da čitači proveravaju istinitost te informacije. Svaki domen srne da pošalje klijentu najviše 20 kolačića. *Putanja* određuje deo strukture di- rektorijuma servera koju sme da koristi kolačić. Ta putanja je često /, što znači daje kolačiću otvoren put kroz čitavo stablo direktorijuma.

Domen	Putanja	Sadržaj	Rok	Bezbednost
toms-casino.com	/	CustomerID=497793521	15-10-02 17:00	Yes
joes-store.com	/	Cart=1-0501;1-0703;2-1321	11-10-02 14:22	No
aportal.com	/	Prefs=Stk:SUNW+ORCL;Spt: Jets	31-12-10 23:59	No
dugoprsti.com	/	UserID=3627239101	31-12-12 23:59	No

Slika 7-25. Primeri kolačića

Sam *Sadržaj* kolačića je u obliku *ime = vrednost.* I *ime* i *vrednost* zadaje server prema svojim potrebama.

Sledeće polje sadrži *Rok* važnosti kolačića. Ako tog polja nema, čitač pri zatvaranju odbacuje kolačić. To je privremen kolačić (engl. *nonpersistent cookie)*. Ako *Rok* sadrži vreme i datum, to je trajan kolačić (engl. *persistent cookie)* koji se čuva dok mu naznačeni rok ne istekne. Rok važnosti kolačića izražen je u skladu sa srednjim griničkim vremenom. Da bi uklonio kolačić s klijentovog čvrstog diska, server ponovo šalje isti kolačić, ali s rokom važnosti koji je istekao.

Na kraju, u polju *Bezbednost* može se naznačiti da čitač srne da vrati kolačić samo bezbednom serveru. Ta opcija se koristi za elektronsku trgovinu, bankarske transakcije i druge aplikacije koje moraju biti posebno obezbeđene.

Do sada smo saznali kako se kolačići dobijaju, ali još uvele ne znamo kako se koriste. Pre nego što pošalje zahtev za stranu s neke Web lokacije, čitač proverava da li u direktorijumu gde se čuvaju postoje i kolačići dobijeni iz domena kome se upućuje zahtev. Ako pronađe takve kolačiće, sve ih uključuje u poruku sa zahtevom. Kada server dobije kolačiće, on može da ih tumači kako god želi.

Potražimo neke namene kolačića. Na slici 7-25, prvi kolačić koji šalje elektronski kazino *toms-casino.com,* koristi se za identifikovanje korisnika. Kada se klijent sledeće nedelje prijavi da bi potrošio malo para, čitač šalje kolačić serveru da bi ovaj znao o kome se radi. Imajući korisnikov identifikator, server može da potraži korisnikov zapis u bazi podataka i da na osnovu njega napravi Web stranu koju će prikazati. U zavisnosti od ličnih kockarskih sklonosti korisnika, ta strana može da sadrži poker, spisak konjskih trka ili automat za kockanje.

Dragi kolačić dolazi iz elektronske prodavnice *joes-store.com.* Korisnik tu obično luta radnjom i zagleda šta bi kupio. Kada pronađe ono što traži i pritisne ga mišem, server pravi kolačić s brojem artikla i odgovarajućom šifrom za svaki artikal, i šalje ga klijentu. Klijent nastavlja da se kreće kroz prodavnicu, a kolačić se šalje posle svakog zahteva za novu stranu. Kako klijent pritiska nove artikle, server ih dodaje u kolačić. Korpa na slici sadrži tri artikla, a poslednji je tražen u duplikatu. Na kraju, kada klijent pritisne *PROCEED TO CHECKOUT,* kolačić - koji sada sadrži potpun spisak kupovine - šalje se zajedno sa zahtevom. Na taj način, server tačno zna šta je kupljeno.

Treći kolačić poslao je Web portal. Kada korisnik pritisne hipervezu ka portalu, time mu upućuje uskladišteni kolačić. Na osnovu njega, portal će napraviti Web stranu sa stanjem akcija korporacija Sun Microsystems i Oracle, i rezultatima njujorškog fudbalskog kluba Jets. Pošto je dozvoljena veličina kolačića 4 KB, ima dovoljno me- sta i za detaljnije zahteve, kao što su novinski naslovi, lokalna vremenska prognoza, specijalne ponude itd.

Kolačiće server može da upotrebi i u sopstvenu korist. Pretpostavimo, na primer, da server želi da evidentira broj različitih korisnika koji su mu pristupili, kao i to koliko je strana svaki korisnik posetio pre nego što je napustio lokaciju. Kada mu stigne prvi zahtev od nekog korisnika, uz njega neće biti kolačića, pa će mu server uzvratiti kolačićem sa sadržajem *Brojač = 1.* Pri svakom sledećem zahtevu kolačić će se vraćati serveru, a server će uvećati brojač za jedinicu i slati ga ponovo klijentu. Analizirajući brojače, server može da utvrdi koliko je korisnika odustalo odmah posle prve strane, koliko ih je posetilo dve strane itd.

Kolačići se često i zloupotrebljavaju. Oni bi, teorijski, trebalo da se vraćaju samo na mesto odakle su potekli, ali su hakeri iskoristili svaki mogući bezbednosni propust čitača Weba da bi se dočepali tuđih kolačića. Pošto neke lokacije za elektronsku trgovinu u kolačiće stavljaju brojeve kreditnih kartica, očigledna je mogućnost njihove zloupotrebe.

Kontroverzna je i upotreba kolačića u cilju neprimetnog skupljanja informacija o tome kako korisnik upotrebljava svoj čitač Weba. To radi ovako. Na primer, reklamna agencija Dugi Prsti ugovara s velikim Web lokacijama da, uz plaćanje, na njih postavi propagandne poruke za proizvode svojih klijenata. Umesto da Web lokaciji, za postavljanje na svaku stranu, dostavi GIF ili JPEG datoteku, reklamna agencija joj dostavlja URL adresu. Takva URL adresa sadrži jedinstven broj u svom delu za datoteku, na primer:

```
http//www.dugoprsti,com/382674902342.gif
```

Kada korisnik prvi put poseti stranu *P* koja sadrži takvu poruku, čitač preuzima HTML datoteku. Zatim je čitač pregleda i pronalazi u njoj vezu ka datoteci sa slikom na adresi www.dugoprsti.com, pa na nju šalje zahtev za sliku. Zaista mu stiže GIF datoteka s propagandnom porukom, ali i kolačić s jedinstvenim korisničkim identifikatorom *3627239101* sa slike 7-25. Agencija beleži činjenicu da je korisnik s tim identifikatorom posetio stranu *P.* To je lako, jer se na traženu datoteku (*382674902342.gif)* ukazuje samo na strani *P.* Naravno, ista poruka se može pojaviti na hiljadama strana, ali uvele s različitim imenom datoteke. Agencija verovatno dobija neki dinar od svojih klijenata kad god na ovaj način pošalje propagandnu poruku.

Kasnije, kada korisnik poseti dragu Web stranu koja sadrži oglas agencije Dugi Prsti, pošto čitač preuzme HTML stranu sa servera, on vidi, recimo, vezu ka slici http://www.dugoprsti.com/493654919923.gifi zahteva tu datoteku. Pošto već ima kolačić iz domena *dugoprsti.com,* čitač uključuje agencijin kolačić koji sadrži korisnički identifikator. Agencija sada zna i drugu stranu koju je korisnik posetio.

Kako prolazi vreme, agencija može da izgradi potpun profil ponašanja korisnika, čak i ako korisnik nikada nije pritisnuo mišem samu propagandnu poruku. Naravno, ona još uvek nema njegovo korisničko ime (premda ima njegovu IP adresu, što može biti dovoljno da se ime izvuče iz drugih baza podataka). Međutim, ako korisnik ikada dostavi svoje korisničko ime bilo kojoj lokaciji koja sarađuje sa agencijom Dugi Prsti, time će svoj profil kompletirati za prodaju svakome ko bude želeo da ga kupi. Prodajom te informacije agencija Dugi Prsti verovatno može dovoljno zaraditi da bi postavila još propagandnih poruka na dodatne Web lokacije i tako skupila informacije o više korisnika. U čitavom tom poslu najcrnje je to što većina korisnika i ne sluti da neko o njima prikuplja informacije, čak misle da su bezbedni jer nikada nisu pritisnuli samu propagandnu poruku.

Ako agencija Dugi Prsti želi da postane majstor svog posla, njena propagandna poruka uopšte ne mora da liči na oglase koje svakodnevno gledate. „Oglas“ veličine jednog piksela, u boji pozadine (i zato nevidljiv), ima isti efekat kao i reklamna poruka koja se širi preko celog ekrana: ona nagoni čitač da preuzme GIF sliku veličine 1x1 piksel i da pošalje sve kolačiće prispele iz zavičajnog domena slike.

Da bi sebe u ovom pogledu prividno umirili, neki korisnici podešavaju svoje čitače da odbijaju sve kolačiće. Međutim, to može da stvori probleme s legitimnim Web lokacijama koje koriste kolačiće. Taj problem korisnici ponekada rešavaju tako što instaliraju softver koji proždire kolačiće. To su specijalni programi koji svaki pristigli kolačić pregledaju i zatim ga prihvataju ili odbacuju shodno pravilima koja je zadao korisnik (npr. prema spisku Web lokacija u koje se može imati poverenja). To korisniku omogućuje da precizno upravlja prihvatanjem kolačića. Savremeni čitači, kao što je Mozilla (■www.moz.illa.org), i sami nude korisnicima složene mogućnosti rada s kolačićima.

7.3.2 Statični Web dokumenti

Osnovu Weba čini prenos Web strana od servera do klijenta. U svom najjednostavnijem obliku, Web strane su statične, odnosno predstavljaju datoteke koje čuče na ser- veru i samo čekaju da ih neko preuzme. U tom smislu, čak i video je statična Web strana jer je i to samo datoteka. U ovom odeljku detaljno ćemo razmotriti statične Web strane. U sledećem ćemo se pozabaviti njihovim dinamičnim sadržajem.

HTML - Jezik za označavanje hiperteksta

Web strane se za sada pišu jezikom za označavanje hiperteksta (engl. *Hypertext Markup Language, HTML),* HTML omogućava korisnicima da prave Web strane s tekstom, grafikom i pokazivačima na druge Web strane. HTML služi za označavanje, to je jezik koji opisuje formatiranje strane. Izraz „markiranje“ (engl. *markup)* potiče iz dana kada su redaktori doslovce unosili oznake u dokumente da bi štamparima - tada još uvek ljudskim bićima - naznačili fontove, uvlačenje itd. Zbog toga jezici za označavanje sadrže izričite komande za formatiranje. Na primer, u jeziku HTML, znači „započni ispis polucrnim fontom“, a znači „kraj polucrnog ispisa“. Prednost jezika koji za označavanje koristi izričite komande

ogleda se u tome što je lako napraviti čitač koji ga razume; čitač samo treba da pravilno razume komande. TeX i troff predstavljaju još dva primera poznatijih jezika za označavanje.

Kada ugradite sve komande za označavanje u svaku HTML datoteku i standardi- zujete ih, svaki Web čitač će moći da učita Web stranu i da je formatira. Ključno je to što čitač može da formatira Web stranu posto je preuzme jer je Web strana možda napravljena u prozoru dimenzija 1600 x 1200 piksela, 24-bitnim bojama, a mora se prikazati u prozoru veličine 640 x 320 piksela 8-bitnim bojama.

U nastavku ćemo predstaviti jezik HTML samo toliko da steknete utisak o čemu se radi. Iako HTML dokumente možete pisati u svakom standardnom programu za uređivanje teksta, možete ih pisati i u specijalnim programima za uređivanje ili obradu HTML teksta, ali ćete tada manje moći da utičete na detalje.

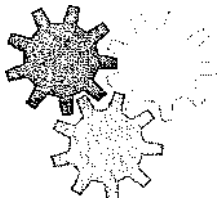
Web strana ima zaglavlje i telo, zajedno smeštene između komandi za formatiranje, tzv. oznaka (engl. *tags)* <html> i </html>, iako se mnogi čitači neće buniti ako tih oznaka nema. Kao što se vidi sa slike 7-26(a), samo zaglavlje smešteno je između oznaka <head> i </head>, a telo između oznaka <body> i </body>. Između oznaka se nalaze direktive. Većina HTML oznaka sledi prikazani format, tj. oznaka <nešto> saopštava da nešto počinje, a oznaka </nešto> kaže da se to nešto završava. Većina čitača u meniju
imaju stavku VIEW SOURCE ili neku sličnu stavku. Kada je izaberete, možete da vidite
izvorni HTML kod aktuelne Web strane.

(a)

```
<html>
<head> <title> AMALGAMATED WIDGET, INC. </title> </head>
<body> <h1> Welcome to AWI's Home Page </h1>
<img src="httpi//www.widget.com/images/logo.gif“ ALT="AWI Logo"> <br>
We are so happy that you have chosen to visit <b> Amalgamated Widget's</b> home page.
We hope <i> you </i> will find all the information you need here.
<p>Below we have links to information about our many fine products.
You can order electronically (by WWW), by telephone, or by fax. </p>
<hr>
<h2> Product information </h2>
<ul>
   <li> <a href="http://widget.com/products/big”> Big widgets </a>
   <li> <a href="http://widget.com/products/little"> Little widgets </a>
</ul>
<h2> Telephone numbers </h2>
<ul>
   <li> By telephone: 1-800-WIDGETS <li> By fax: 1-415-765-4321
</ul>
</body>
</html>
```

(a)

Welcome to AWI's Home Page



We are so happy that you have chosen to visit **Amalgamated Widget's** home page. We hope ***you*** will find all the information you need here.

Below we have links to information about our many fine products. You can order electronically (by WWW), by telephone, or by FAX.

Product Information

[9] Big widgets •
Little widgets

Telephone numbers

o 1-800-WIDGETS ® 1-
415-765-4321

(b)

Slika 7-26. (a) Izvorni HTML kod jedne Web strane, (b) Izgled formatirane Web strane.

Oznake možete pisati velikim ili malim slovima. Prema tome, <head> i <HEAD> znače isto, ali nove verzije standarda prepoznaju samo mala slova. Raspored HTML teksta na strani nije važan. Programi za analiziranje HTML teksta zanemaruju višak praznih mesta i prelom redova jer tekst formatiraju tako da stane u raspoloživ prostor za prikazivanje. Prema tome, da bi HTML tekst bio čitljiviji (što je veoma poželjno), ubacujte u njega beline onako kako to vama odgovara. Imajte na umu da pasuse ne možete razdvajati praznim redovima jer se oni jednostavno zanemaruju. Za kraj pasusa neophodna je posebna oznaka.

Neke oznake imaju svoje (imenovane) parametre, tzv. atribute. Na primer,
<img.src="abc" alt="tekst">

predstavlja oznaku <img> s parametrom *src* postavljenim na vrednost *abc* i parametrom *alt* postavljenim na vrednost *tekst.* Za svaku oznaku, HTML standard definiše listu dozvoljenih parametara i daje njihovo značenje. Pošto svaki parametar ima svoje ime, redosled njihovog navođenja nije važan.

Tehnički gledajući, HTML dokumenti se pišu skupom ISO 8859-1 znakova Latin-1, ali za korisnike čije tastature podržavaju samo skup ASCII znakova, definisane su kontrolne sekvence za prikazivanje specijalnih znakova, npr. znaka e. Spisak specijalnih znakova naveden je u standardu. Svi oni počinju ampersendom i završavaju se tačkom i zarezom. Na primer,   znači razmak, è proizvodi znak e, a é znale e. Pošto znakovi <, > i & imaju specijalno značenje, oni se doslovno mogu prikazati samo pomoću kontrolnih sekvencii &.lt;, > i &.

Glavna stavka zaglavlja je naslov, smešten između oznaka <title> i </title>, ali zaglavlje može sadržati i određene metainformacije. Naslov se inače ne prikazuje na strani. Neki čitači njime označavaju prozor Web strane.

Razmotrimo sad i neke druge oznake na slici 7-26. Sve oznake sa slike 7-26, a i još neke, prikazane su na slici 7-27. Podnaslovi se generišu oznakom <h«>, gde je *n* cifra između 1 i 6. Dakle, <h 1 > je oznaka za najvažniji, a <h6> za najmanje važan podnaslov. Čitaču se prepušta da podnaslove pravilno formatira na ekranu. Najčešće će važniji podnaslovi biti prikazani

krupnijim fontom i deblje. Čitač može nivoe podnaslova da prikaže i u različitim bojama. Podnaslovi nivoa <h 1 > obično su krupni i debelo ispisani, s po jednim praznim redom ispred i iza podnaslova. Nasuprot tome, podnaslovi <h2> sitniji su i sa manje praznog prostora ispred i iza podnaslova.

Oznaka	Opis
<html>... </html>	Deklariše Web stranu pisanu jezikom HTML
<head> ... </head>	Odvaja zaglavlje strane
<title>... </title>	Definiše naslov (koji se ne prikazuje na strani)
<body> ... </body>	Odvaja telo strane
	Označava podnaslov n-tog nivoa
...	Poiucrni ispis
	Kurzivni ispis

nastavlja se

Oznaka	Opis
...	Horizontalno centriranje na strani
...	Obuhvata neuređenu (označenu) listu ai
...	Obuhvata uređenu (numerisanu) listu
...	Obuhvata stavku u uređenoj ili neuređenoj listi
 	Izaziva prelom reda
	Započinje pasus
<hr>	Umeće horizontalnu liniju
<img src="„...">	Prikazuje sliku na tekućoj poziciji
... 	Definiše hipervezu

Slika 7-27. Odabrane HTML oznake. Neke imaju i dodatne parametre.

Oznakama i <i> počinje polucrni (engl. *bolđ),* odnosno kurzivni ispis (engl. *italic)*. Ako čitač ne može slova da prikaže polucrno ili u kurzivu, mora da ih prikaže na neki drugi način, npr. drugom bojom ili možda inverzno.

U jeziku HTML liste se mogu praviti na više načina, a i ugnežđivati jedna u drugu. Liste počinju oznakom <ul> ili <ol>, a u oba slučaja pojedinačne stavke liste počinju oznakom <li>. Oznakom <ul> počinje neuređena lista. Njene stavke, budući neuređene, označene su bulitima (•). Uređene (numerisane) liste počinju oznakom <ol>. Njihove stavke čitač obeležava brojevima. Komande <ul> i <ol>, iako se različito pišu, imaju istu sintaksu i daju slične rezultate.

Oznakama
, i <hr> naglašava se granica između pojedinih delova teksta. Tačan format se može definisati opisom stila (pogledajte u nastavku) koji je pridružen strani. Oznaka
 samo prelama red. Čitači najčešće ne umeću prazan red iza nje. Nasuprot tome, oznaka započinje pasus, pri čemu se ispred njega može ubaciti prazan red i prvi red pasusa uvući. (Oznakom teoretski treba da se završi pasus, ali se ona retko koristi; mnogi autori HTML strana čak i ne znaju da takva oznaka postoji.) I na kraju, oznaka <hr> pravi prekid u tekstu i umeće horizontalnu liniju.

Jezik HTML omogućava umetanje slika u tekst Web strane. Oznaka <img> znači da sliku treba prikazati na tekućoj poziciji na strani. Uz nju se može zadati više parametara. Parametrom *src* zadaje se URL adresa slike. HTML standard ne ograničava pri- menu različitih grafičkih formata. U praksi se može računati s tim da svi čitači podržavaju formate GIF i JPEG. Čitači mogu da podržavaju i drage formate, ali je to rizično. Ako se korisnik navikne na čitač koji podržava, na primer, format BMP, on takve datoteke može da uključi u svoje Web strane i kasnije da se iznenadi kad utvrdi da drugi čitači jednostavno ne obraćaju pažnju na njegova remek-dela.

U druge parametre oznake <img> spadaju *align,* koji upravlja poravnanjem slike u odnosu na osnovnu liniju teksta *{top, middle, bottom* - gore, u sredini, dole), *alt,* koji umesto slike prikazuje tekst u slučajevima kada je korisnik onemogućio prikazivanje slika, i *ismap,* indikator koji označava da slika predstavlja aktivnu mapu (tj. sliku s područjima osetljivim na pritisak mišem).

Na kraju, dolazimo do hiperveza, za koje se koriste oznake <a> (anker, sidro) i . Kao i uz oznaku <img>, i uz <a> se mogu koristiti različiti parametri, uključujući /7./'<?/(URL) i *name* (ime hiperveze). Tekst obuhvac'en oznakama <a> i prikazuje se na ekranu. Ako ga korisnik izabere, preći će na Web stranu na koju ukazuje hiper- veza. Između oznaka <a> i može se umetnuti i slika pomoću oznake <img>, pa se hiperveza aktivira pritiskom na sliku.

Evo primera HTML koda:

```
<a href="http://www.nasa.gov"> NASA - matična strana </a>
```

Kada se prikaže strana sa ovim delićem HTML koda, na ekranu će se po javiti NASA - matična strana

Ako korisnik pritisne ovaj tekst, čitač će preuzeti stranu sa URL adrese *http://* www.nasa.gov i prikazati je.

Sada pogledajte dragi primer:

```
<a href="http://www.nasa.gov'1> <img src="shuttle.gif“ alt="NASA"> </a>
```

Prikazana strana sadržaće sliku (npr. spejs šatl). Pritisak na sliku odvešće vas na matičnu stranu agencije NASA, kao i pritisak na podvučeni tekst u prethodnom primera. Da je korisnik isključio prikazivanje slika u svom čitaču, na ekranu bi se umesto nje pojavio alternativni tekst NASA.

Pomoću parametra *name* oznake <a> može se postići da hiperveza ukaže na neko mesto na Web strani. Na primer, neke Web strane počinju sadržajem čije su stavke osetljive na pritisak mišem. Kada korisnik neku od njih pritisne, prikazaće se odgovarajuće mesto na strani.

HTML se i dalje usavršava. Njegove verzije 1.0 i 2.0 ne podržavaju tabele; one su dodate u verziji 3.0. HTML tabela sadrži jedan ili više redova, a svaki red ima jednu ili više ćelija. Same ćelije mogu sadržati različit materijal, uključujući tekst, slike, ikonice, fotografije, čak i drage tabele. Ćelije se mogu spajati, tako da, na primer, naslov može da se ispiše u više kolona. Autori Web strana mogu u izvesnoj meri da upravljaju poravnanjem, izgledom ivica i marginama ćelija, ali odlučujuću reč pri prikazivanju tabela ima čitač.

Definicija jedne HTML tabele prikazana je na slici 7-28(a), a njen mogući prikaz na slici 7-28(b). Primer prikazuje samo neke osnovne mogućnosti za pravljenje HTML tabela. Definicija tabele počinje oznakom <table>. Zatim se pomoću dopunskih informacija mogu definisati opšta svojstva tabele.

(a)

```
<html>
<head> <title> Primer strane s tabelom </title> </head>
<body>
•ctable border=1 rules=all>
<caption> Neke razlike između verzija HTML jezika </caption>
<col align=left>
<col align=center>
<col align=center>
<col align=center>
<col align=center>
<tr> <th>Mogućnost <th>HTML 1.0 <th>HTML 2.0 <th>HTML 3.0 <th>HTML 4.0 </tr>
<tr> <th> Hiperveze <td> x <td> x <tđ> x <td> x </tr>
<tr> <th> Slike <td> x <td> x <td> x <td> x </tr>
<tr> <th> Liste <td> x <td> x <td> x <td> x </tr>
<tr> <th> Aktivne mape i slike <td>   <td> x <td> x <td> x </tr>
<tr> <th> Obrasci <td>   <td> x <td> x <td> x </tr>
<tr> <th> Jednačine <td>   <td>   <td> x <td> x </tr>
<tr> <th> Palete alatki <td>   <td>   <td> x <td> x </tr>
<tr> <th> Tabele <td>   <td>   <td> x <td> x </tr>
<tr> <th> Pomoć hendikepiranima <td>   <td>   <td>   <td> x </tr>
<tr> <th> Ugrađivanje objekata <td>   <td>   <td>   <td> x </tr>
<tr> <th> Skriptovanje <td>   <td>   <td>   <td> x </tr>
</table>
</body>
</html>
```

(b)

Neke razlike između verzija HTML jezika

Mogućnost	HTML 1.0	HTML 2.0	HTML 3.0	HTML 4.0
Hiperveze	X	X	X	X
Slike	X	X	X	X
Liste	X	X	X	X
Aktivne mape i slike		X	X	X
Obrasci		X	X	X
Jednačine			X	X
Palete alatki			X	X
Tabele			X	X
Pomoć hendikepiranima				X
Ugrađivanje objekata				X
Skriptovanje				X

Slika 7-28. (a) HTML tabela, (b) Jedan od mogućih prikaza ove tabele.

Oznakom <caption> definiše se naslov tabele. Svaki red počinje oznakom <tr> (engl. *table row* - red tabele), a pojedine delije nose oznake <th> (engl. *table header* - zaglavlje tabele) ili <td> (engl. *table data* - podaci). To je potrebno da bi čitač mogao različito da prikaže zaglavlje i podatke, kao što smo i mi uradili u primeru.

U tabelama se mogu koristiti i brojni atributi. Pomoću njih se može zadavati horizontalno i

vertikalno poravnanje sadržaja delija, poravnanje uz obe ivice, izgled ivica delija, grupisanje delija, jedinice i drugo.

HTML u verziji 4.0 sadrži i dodatne mogućnosti. Među njima su pomoć za hendikepirane osobe, ugrađivanje objekata (generalizacija oznake <img>, tako da se u Web stranu mogu ugrađivati i drugi objekti), podrška skriptovima (za dinamičko generi- sanje sadržaja strane) i druge mogućnosti.

Kada je Web lokacija složena i sadrži mnogo strana koje su pravili različiti autori iste kompanije, često je poželjno obezbediti da sve takve strane imaju isti izgled. To se postiže tzv. **opisima stilova** (engl. *style sheets).* Kada se oni upotrebe, strane se više ne podešavaju pojedinačnim stilovima, kao što su polucrni ili kurzivni ispis, već se za to koriste logički stilovi, kao što su <dn> (definiši), (blago isticanje), (snažno isticanje) i <var> (programske promenljive). Logički stilovi su definisani u odgovarajućem opisu stila na koji upućuje početak svake strane. Na taj način sve strane imaju isti stil i ako urednik Web lokacije odluči da logički stil izmeni tako da ne označava više plav, kurzivni ispis fontom od 14 tačaka, već provokativno ružičast, podebljan ispis veličine 18 tačaka, treba samo da izmeni jednu definiciju u opisu stila, pa će se promeniti i izgled čitave lokacije. Opis stila može se uporediti s datotekom koja se poziva direktivom #include u kodu pisanom na jeziku C: izmenom jedne makro definicije u njoj menjaju se svi programi koji tu direktivu sadrže u zaglavlju.

Obrasci

HTML u verziji 1.0 omogućavao je jednosmeran saobraćaj. Korisnici su mogli da preuzimaju strane od davalaca informacija, ali su im teško mogli slati poruke. S porastom broja komercijalnih organizacija na Webu, potreba za dvosmernim saobraćajem sve više je rasla. Na primer, mnoge kompanije su želele mogućnost prikupljanja narudžbina za određene proizvode preko svojih Web strana, prodavci softvera su želeli da od kupaca prikupe podatke za registraciju softvera takođe preko Weba, a kompanije koje su nudile usluge pretraživanja želele su da korisnicima omoguće slanje ključnih reči za pretraživanje Weba.

Zbog opisanih potreba, u HTML jezik, počev od verzije 2.0, uvedeni su **obrasci** (engl. *forms).* Obrasci sadrže polja ili dugmad, koja korisnicima omogućavaju da upišu podatke ili da nešto izaberu, a zatim da te informacije pošalju natrag vlasniku Web strane. Za to se koristi oznaka <input>, uz koju se mogu navesti različiti parametri kojima se određuje veličina, priroda i način korišćenja prikazanog polja. Najčešći elementi obrazaca su prazna polja u koja korisnik može da upiše tekst, polja za potvrdu, aktivne mape i dugmad tipa *submit* (pošalji, podnesi obrazac). Primer na slici 7-29 prikazuje neke od ovih mogućnosti.

```
<html>
<head> <title> AWI CUSTOMER ORDERING FORM </title> </head>
<body>
<h1> Widget Order Form </h1>
<form ACTION="http://widget.com/cgi-bin/widgetorder“ method=POST>
<p> Name cinput name="customer“ size=46> </p>
<p> Street Address cinput name="address" size=40> </p>
<p> City cinput name="city" size=20> State cinput name="state“ size =4>
Country cinput name="country“ size= 10> c/p> cp> Credit card # cinput
name="cardno“ size=10>
Expires cinput name="expires“ size=4>
M/C cinput name="cc" type=radio value="mastercard">
```

VISA cinput name="cć‘ type=radio value="visacard"> c/p>

```
<p> Widget size Big <input name="producf type=radio value="expensive"> Little <input
name=”producf type=radio value="cheap">
Ship by express courier <input name="express“ type=checkbox> </p> <pxinput type=submit
value="submit order"> </p>
Thank you for ordering an AWI widget, the best widget money can buy! </form>
</body>
</html>
```

(a)

Widget Order Form

Name

Street address

State I Country^

City

Credit card # Expires M/C O Visa O

Widget size Big Q Little Q Ship by express courier Q

Submit order

Thank you for ordering an AWI widget, the best widget money can buy!

(b)

Slika 7-29. (a) HTML kod za jednu narudžbenicu, (b) Formatirana strana.

Počnimo naše objašnjavanje obrazaca ovim primerom. Kao i svi obrasci, i ovaj je smešten između oznaka <form> i </form>. Tekst izvan ovih oznaka prikazuje se na ekranu kao i svaki drugi tekst. Unutar obrasca su dozvoljene uobičajene oznake za forma- tiranje, npr. oznaka <b>. U našem obrascu koriste se tri vrste polja za unos podataka.

Polje prve vrste smešteno je u produžetku teksta „Name“. Ono je dužine 46 znakova. Od korisnika se očekuje da u njega upiše svoje ime i prezime, koje se zatim smešta u promenljivu *customer* i čuva za kasniju obradu. Oznaka <o> nalaže čitaču da naredni tekst i polja prikaže u sledecem redu, čak i ako ima mesta u tekućem. Pomoću oznake <p> i sličnih oznaka, autor strane može da utiče na njen izgled na ekranu.

U sledecem polju (iza teksta „Street address“) od korisnika se traži da upiše svoju adresu. To polje je talcođe u zasebnom redu i dužine 40 znakova. Zatim dolazi red s poljima za upisivanje grada („City“), države („State“) i savezne države („Country“). Između polja nema oznake <p>, tako da se sva prikazuju u istom redu, ako mogu da stanu u njega. S gledišta čitača, ovaj pasus sadrži šest stavki: tri tekstualna niza koja se smenjuju s tri polja. On ih prikazuje uzastopno sleva udesno i prelazi u sledeći red čim tekući postane tesan. Stoga se može zamisliti da će se na ekranu veličine 1600 x 1200 tačaka sva tri tekstualna niza i odgovarajuća polja prikazati u istom redu, ali da

će na ekranu veličine 1024 x 768 oni možda biti podeljeni u dva reda. U najnepovoljnijem slučaju, reč „Country“ ostade na kraju jednog reda, a polje koje joj odgovara preći đe u sledeći red.

U sledećem redu od korisnika se traži da upiše broj svoje kreditne kartice („Credit card #“) i datum do kada ona važi („Expires“). Broj kreditne kartice treba slati Internetom samo ako su preduzete odgovarajuće bezbednosne mere. O tome ćemo nešto redi u 8. poglavlju.

Iza polja „Expire“ nailazimo na nove elemente, tzv. radio-dugmad (engl. *radio buttons),* previđene za opcije koje se međusobno isključuju. Ovaj element je napravljen po ugledu na radio u automobilu s gomilom dugmida za biranje različitih stanica. Čitač tu dugmad prikazuje na način koji korisniku omogućava da ih izabere mišem ili s tastature. Kada pritisne jedno dugme, isključuju se sva druga u istoj grupi. Sam vizuelni prikaz zavisi od čitača. Veličina izabranog proizvoda („Widget size“) takođe se bira pomoću dva radio-dugmeta: veliki („Big“) i mali („Little“). Dve grape dugmadi razlikuju se po imenu (polju *name),* a ne po nekakvim hipotetičkim oznakama <radiobutton> ... </radiobutton>.

Za prepoznavanje pritisnutog dugmeta služi parametar- *value* (vrednost). U zavisnosti od kreditne kartice koju imate (M/C ili Visa), na osnovu čega ste pritisnuli odgovarajuće dugme, promenljiva *cc* će dobiti vrednost „mastercard“, odnosno „visacard“.

Posle dva skupa radio-dugmadi nailazimo na opciju koja se tiče načina isporuke, predstavljenu poljem za potvrdu (engl. *checkbox).* To polje možete da izaberete ili da ga ne izaberete. Za razliku od radio-dugmadi, gde iz skupa dugmadi morate izabrati samo jedno, svako polje za potvrdu može da bude izabrano (ili neizabrano), nezavisno od ostalih sličnih polja. Na primer, kada naručujete picu s Web strane Electropiz- za, možete da izaberete picu sa sardinom *i* lukom *i* ananasom (ako to možete da svarite), ali ne možete istovremeno da izaberete veliku *i* malu picu. Sadržaj pice biće ponuđen pomoću tri zasebna polja za potvrdu, dok ćete njenu veličinu moći da birate pomoću radio-dugmadi.

Pomenimo uzgred da su radio-dugmad nezgodna za prikazivanje dugačkih lista stavki od kojih se mora izabrati jedna. Tu se primenjuju oznake <select> i </select> za obuhvatanje liste mogućih izbora, ali uz tumačenje kao kod radio-dugmadi (osim ako se zada parametar *multiple,* kada se lista tumači kao skup polja za potvrdu). Lista između oznaka <select> i </select> u nekim čitačima se prikazuje kao padajući meni.

Dosad smo obradili dva ugrađena tipa za oznaku <input>: *radio* i *checkbox.* U stvari, obradili smo i treći: *text.* Pošto je to podrazumevani tip, nismo se trudili da u kod upisujemo *type = text,* ali i da jesmo, ne bismo pogrešili. Dva sledeća tipa su *password* (lozinka) i *textarea* (višeredno polje). Polje *password* je isto što i polje *text,* samo što se znakovi ne prikazuju na ekranu dok se unose. Polje *textarea* prihvata više redova teksta.

U primeru na slici 7-29 dolazimo do dugmeta tipa *submit* (pokrivenog tekstom „Submit order“). Kada ga korisnik pritisne, podaci koje je uneo u obrazac šalju se računani s koga je potekao obrazac. Slično drugim tipovima, i *submit* je rezervisana reč koju čitač razume. Parametrom *value* (vrednost) zadaje se tekst koji se pojavljuje na dugmetu. Sva polja mogu da imaju parametar *value,* ali je jedino ovde on i neophodan. Kod polja tipa *text,* vrednost se (ako je definisan parametar' *value*) prikazuje zajedno s ponuđenim obrascem, ali je korisnik može izmeniti ili obrisati. I polja tipa *checkbox* i *radio* mogu se inicijalizovati, ali atributom *checked* (izabrano), jer parametar *value* daje samo tekst, ali ne pokazuje koje je dugme podrazumevano izabrano.

Kada korisnik pritisne dugme tipa *submit,* čitač pakuje podatke unete u obrazac u jedan

jedini dugačak red teksta i šalje ga serveru na obradu. Znakom & razdvajaju se polja, a znale + predstavlja razmak. Za obrazac iz našeg primera, taj red bi mogao izgledati kao na slici 7-30 (iako je tekst prelomljen u tri reda zbog ograničene širine stranice, sve to predstavlja jedan logički red podataka).

```
customer=John+Doe&address=100+Main+St.&city=White+Plains&
state=NY&country=USA&cardno=1234567890&expires=6/98&cc=mastercard&
product=cheap&express=on
```

Slika 7-30. Moguć odgovor čitača serveru: podaci koje je upisao korisnik.

Tekstualni niz neće biti poslat servent u tri reda, već u jednom. Ako polje za potvrdu (*checkbox*) nije izabrano, neće ga biti ni u tekstualnom nizu. Servent se prepušta da protumači dobijeni niz. Kako se to radi, videćemo u nastavku poglavlja.

XMLiXSL

Ježile HTML, sa obrascima ili bez njih, ne omogućava nikakvo strukturiranje Web strane. U njemu su sadržaj i formatiranje izmešani. Međutim, s razvojem elektronske trgovine i drugih aplikacija narasla je potreba za strukturnijim Web stranama kod kojih je sadržaj odvojen od formatiranja. Na primer, program koji pretražuje Web u potrazi za najnižom cenom neke knjige ili CD-a mora da analizira mnoge Web strane tražeći u njima ime artikla i cenu. Kada su Web strane napisane jezikom HTML, programu je teško da utvrdi gde se na njima nalazi ime artikla i cena.

Zbog toga je konzorcijum W3C proširio jezik HTML da bi se mogle praviti Web strane pogodno strukturirane za automatsku obradu. Za tu svrhu su napravljena dva nova jezika. Prvi, **proširivi jezik za označavanje** (engl. *extensible Markup Language, XML)* opisuje strukturirano sadržaj Web strane, a dragi, **proširivi jezik za pravljenje stilova** (engl. *extensible Style Language, XSL)* opisuje formatiranje nezavisno od sadržaja. Oba jezika su teme za sebe, pa ćemo u nastavku pokušati samo da vam približimo kako rade.

Razmotrite primer XML dokumenta na slici 7-31. U njemu se definiše struktura bookjist, koja predstavlja spisak knjiga. Za svaku knjigu u njemu postoje tri polja: naslov, autor i godina izdanja. Ove strukture su izuzetno jednostavne. Dozvoljene su, međutim, strukture s ponovljenim poljima (npr. kada ima više autora), neobaveznim poljima (npr. naslov priloženog CD-a) i alternativnim poljima (npr. URL adresa knjižare, ako se knjiga može naći u slobodnoj prodaji ili URL adresa lokacije za aukcije, ukoliko se više ne može naći kod izdavača).

Svako od tri polja u ovom primeru predstavlja nedeljivu celinu, ali je u načelu dozvoljeno da se svako polje dalje deli. Na primer, polje za autora moglo je biti izdelje- no na sledeći način da bi se omogućilo preciznije pretraživanje i formatiranje:

```
<author>
   <first_name> Andrew </first_name>
   <last_name> Tanenbaum </last_name>
</author>
```

Svako polje se može deliti na potpolja, ova na pod-potpolja i tako u beskraj. Dokument na slici 7-31 samo đefiniše listu knjiga s tri odrednice. On ne određuje način na koji će se lista prikazati na Web strani. Za formatiranje je potrebna druga datoteka, *bookjist.xsl,* koja sadrži definicije na jeziku XSL. Ta datoteka je opis stila kojim će se Web strana prikazati. (Postoje i alternative upotrebi opisa stilova, npr. pretvaranje XML datoteke u HTML datoteku, ali

njihovo opisivanje prevazilazi okvire ove knjige.)

```
<?xml version="1.0“ ?>

<?xml-stylesheet type="text/xsl" href="book_list.xsl"?>

<book_list>

<book>
   <title> Computer Networks, 4/e </title>
   <author> Andrew S. Tanenbaum </author>
   <year> 2003 </year>
</book>

<book>
   <title> Modern Operating Systems, 2/e </title>
   <author> Andrew S. Tanenbaum </author>
   <year> 2001 </year>
</book>

<book>
   <title> Structured Computer Organization, 4/e </title>
   <author> Andrew S. Tanenbaum </author>
   <year> 1999 </year>
</book>

</bookJist>
```

Slika 7-31. Jednostavna Web strana na jeziku XML,

Primer XML datoteke za formatiranje datoteke sa slike 7-31, prikazan je na slici 7-32. Posle nekoliko neophodnih deklaracija koje uključuju i URL adresu XSL standarda, u datoteci se pojavljuju oznake <html> i <body> koje, kao i obično, najavljuju početak Web strane. Zatim sledi definicija tabele, uključujući i zaglavlja tri njene kolone. Obratite pažnju na to da pored oznaka <th> postoje i oznake </th> o kojima dosad nismo brinuli. Specifikacije jezika XML i XSL. mnogo su strožije od
HTML specifikacija. One nalažu da se sintaksno neispravna datoteka odbaci, čak i onda kada čitač može da pogodi staje autor Web strane nameravao. Čitač koji prihvati sintaksno neispravnu XML ili XSL datoteku i sam ispravi greške, smatra se neprikladnim i bice odbačen pri proveri podobnosti. Čitači, međutim, smeju da uka- žu na greške. Ovakve, pomalo drakonske mere bile su potrebne da bi se izišlo na kraj sa nepreglednom hrpom kojekako napisanih Web strana.

```
<?xrnl version='1,0'?>
<xsl:stylesheet xmlns:xsl="http://www. w3.org/1999/XSL/Transform“
version="1.0"> <xsl:template match="/">

<html>
<body>

ctable border='’2">
  <tr>
     <th> Title</th>
     <th> Author</th>
```

```
    <th> Year </th>
  </tr>

  <xsl :for-each select="bookjist/book">
  <tr>
    <td> <xsl:value-of select="title"/> </td>
    <td> <xsl:value-of select="author"/> </td>
    <td> <xsl:value-of select="year"/> </td>
  </tr>
  </xsl:for-each>
</table>

</body>
</html>
</xslitemplate>
</xsl:stylesheet>
```

Slika 7-32. Stil opisan jezikom XSL.

Naredba

```
<xsl:for-each select="book_list/book">
```

potpuno je analogna naredbi for u jeziku C. Ona nalaže čitaču da za svaku knjigu (book) ponovo prođe kroz petlju (koja se završava oznakom <xsl:for-each>). Pri svakom prolasku generiše se pet redova: <tr>, naslov, autor, godina i </tr>. Po izlasku iz petlje dolazimo do završnih oznaka </body> i </html>. Čitač će protumačiti ovaj opis stila kao da Web strana ima tabelu u tekstu. Međutim, kada se Web strana napiše u ovom formatu, programi će moći da analiziraju XML datoteku i da u njoj lako pronađu, na primer, knjige štampane od 2000. godine do danas. Iako naša XML datoteka već ima programsku strukturu (petlju), treba naglasiti da su Web strane pisane jezicima XML i XSLjoš uvek statične jer sadrže u stvari samo instrukcije čitaču u vezi s načinom prikazivanja strane, baš kao i HTML strane. Naravno, da bi mogao da upotrebi XML i XSL strane, čitač mora znati da protumači ta dva jezika, a većina čitača to danas može. Ostaje otvoreno pitanje da li de XSL prevagnuti nad klasičnim opisima stilova.

O tome nismo govorili, ali XML omogućava urednicima Web lokacija da prave datoteke u kojima su strukture unapred definisane. Te definicione datoteke mogu se kasnije uključivati pri pravljenju konkretnih Web strana i omogućiti izradu veoma složenih struktura. Dopunska objašnjenja u vezi s jezicima XML, i XSL potražite u nekoj od brojnih knjiga posvećenih tim temama. Dve su i u spisku literature na kraju ove knjige (Livingston, 2002 i Williamson, 2001).

Pre nego što završimo s jezicima XML i XSL, osvrnimo se na ideološku borbu između konzorcijuma W3C i zajednice dizajnera Weba. Prvobitni zadatak jezika HTML bio je da definiše *strukturu* dokumenta, a ne njegov *izgled.* Na primer, naredba

```
<h1 > Milenine fotke </h1 >
```

nalaže čitaču da istakne podnaslov, ali pritom ne zadaje ni font, ni veličinu, ni boju ispisa. To se prepušta čitaču koji zna mogućnosti ekrana (broj pilcsela, boja itd.). Međutim, mnogi Web dizajneri poželeli su mogućnost da potpuno upravljaju izgledom svojih strana, pa su u HTML dodate nove oznake, na primer

```
<font face="helvetica“ size="24“ color="red”> Milenine fotke </font>
```

Dodate su i mogućnosti za precizno pozicioniranje objekata na ekranu. Nažalost, takav format nije prenosiv. Iako je strana možda izgledala savršeno **u** čitaču **u** kome je napravljena, u drugom čitaču, drugoj verziji istog čitača ili na ekranu drage rezolucije mogla se pretvoriti u bezobličnu mrlju. Jezik XMLbio je delimičan pokušaj vraćanja prvobitnom cilju - opisivanju strukture, a ne izgleda dokumenta. Istovremeno je stvoren i XML jezik za opisivanje izgleda. Oba formata se, međutim, mogu zloupotrebiti. Samo čekajte i posmatrajte.

Osim za opisivanje Web strana, XML se može upotrebiti i za druge svrhe. On sve češće nalazi mesto kao jezik za međusobno komuniciranje aplikacija. Tu je **jednostavan protokol za pristupanje objektima** (engl. *Simple Object Access Protocol, SOAP)* posebno pogodan za daljinsko pozivanje procedura između aplikacija na način koji prevazilazi različitosti jezika i sistema. Klijent sastavlja zahtev kao XML poruku i šalje ga serveru koristeći protokol HTTP (o kome govorimo u nastavku). Server vraća odgovor kao formatiranu XML poruku. Na taj način, međusobno mogu da komuniciraju različite platforme.

XHTML - Prošireni jezik za označavanje hiperteksta

HTML nastavlja da se razvija odgovarajući sve novijim i novijim zahtevima. Mnogi u industriji veruju da budućnost Weba ne pripada stacionarnim PC računari- ma, već bežičnim, ručnim uređajima, tipa ličnog digitalnog pomoćnika (LDA). Takvi uređaji nemaju dovoljno memorije za složene čitače Weba, prepune mehanizama za ispravljanje sintaksno neispravnih Web strana. Zbog toga, sledeći korak posle 4. verzije HTML-a mogao je da bude samo neki Veoma Izbirljiv jezik. Taj jezik je **prošireni jezik za označavanje hiperteksta** (engl. *extended HyperText Markup Language,*
XHTML), a ne HTML5, jer je XHTMLu suštini HTML 4 preformulisan u XML. Pod time podrazumevamo da u njemu oznaka kao <h1 > nema značenja. Da bi se dobio efe- kat jezika HTML 4, potrebna je definicija u XML datoteci. XHTML je nov standard za Web i treba ga koristiti za sve nove Web strane da bi se postigla maksimalna pre- nosivost između različitih platformi i čitača.

Postoji šest glavnih razlika između jezika XHTML i jezika HTML 4, kao i niz manjih. Počnimo od onih glavnih. Prvo, XHTML strane i čitači moraju se strogo pokoravati standardu. Nema više „zbrdazdolisanih“ Web strana. To svojstvo je preuzeto iz jezika XML.

Drugo, sve oznake i atributi moraju biti pisani malim slovima. Oznaka <HTML> u jeziku XHTML nije ispravna. Mora se pisati <html>. Isto tako, nije ispravno ni <img SRC="pic001.jpg"> jer sadrži atribut pisan velikim slovima.

Treće, obavezna je primena oznaka u paru, čak i za takve oznake kao što su i . Za oznake koje nemaju svoju završnu varijantu, kao što su oznake
, <hr> i <img>, ispred završnog znalca „>“ mora se staviti kosa crta, na primer

```
<img src=sl001 .jpg“ />
```

Četvrto, atributi se moraju stavljati u navodnike. Na primer, ovakvo nešto

```
<img SRC="sl001.jpg“ height=500 />
```

nije više dozvoljeno. Broj 500 mora biti u navodnicima, kao i ime JPEG datoteke, iako je to broj, a ne tekst.

Peto, oznake se moraju ispravno ugnežđivati jedna u drugu. To ranije nije bilo neophodno, sve dok je postizan željeni efekat. Na primer,

```
<center> <b> Slike sa odmora </center> </b>
```

ranije je moglo da prođe. Jezik XHTML to ne dopušta. Oznake se moraju zatvarati redosledom obrnutim od njihovog otvaranja.

Šesto, u svakom dokumentu mora se naznačiti njegov tip. To smo, na primer, vi- deli u dokumentu sa slike 7-32. Opširnija objašnjenja i velikih i malih razlika možete naći na adresi www.w3.org.

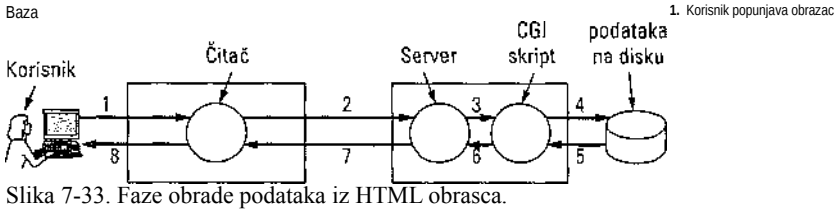
7.3.3 Dinamični Web dokumenti

Sve do sada smo koristili model prikazan na slici 6-6: klijent šalje ime datoteke serveru, koji mu tada vraća datoteku. Na početku razvoja Weba, sav sadržaj je bio ovako statičan (sve same datoteke). U novije vreme, sadržaj je postao dinamičniji - više se ne isporučuje gotov s diska, već se generiše na zahtev. Sadržaj se može gene- risati na serveru ili na klijentskom računam. Pogledajmo kako radi ijedno i drugo.

Dinamičko generisanje Web strana na serveru

Da biste razumeli zašto je potrebno da se sadržaj generiše na serveru, razmotrite ponovo upotrebu obrazaca. Kada korisnik ispuni obrazac i pritisne dugme *submit,* serveru se šalje poruka i naznačava da je u njoj sadržaj obrasca, zajedno s poljima koje je korisnik popunio. Ta poruka ne predstavlja ime tražene datoteke, već podatke

koje treba proslediti nekom programu ili skriptu na obradu. Obrada obično podrazu- meva traženje zapisa u bazi podataka na serveru, koje se obavlja na osnovu podataka dobijenih od korisnika, i generisanje prilagođene HTML strane koja se šalje klijentu. Na primer, u nekom programu za elektronsku trgovinu, pošto korisnik pritisne *PROCEED TO CHECKOUT* (idi na kasu), čitač vraća kolačić sa sadržajem korpe izabranih artikala, ali neki program ili skript na serveru treba da se pokrene da bi ga obradio i kao odgovor generisao HTML stranu. Ta HTML strana može da sadrži obrazac sa spiskom izabranih artikala i korisnikovom adresom, zajedno sa zahtevom da korisnik proveri podatke i naznači način plaćanja. Koraci neophodni za obradu podataka iz HTML obrasca prikazani su na slici 7-33.



Slika 7-33. Faze obrade podataka iz HTML obrasca.

2. Obrazac se vraća serveru 3 Obrazac se predaje CGI skriptu
4. CGI skript pretražuje bazu podataka
5. Zapis je nađen
6. CGI skript generiše stranu
7. Strana se vraća klijentu 8 Strana se prikazuje

Sa obrascima i drugim interaktivnim Web stranama najčešće se radi pomoću sistema zvanog **opšti interfejs za mrežni prolaz** (engl. *Common Gateway Interface, CGI)*- Taj standardni interfejs omogućava Web serverima komunikaciju s programima ili skriptovima za obradu koji mogu da prihvate ulazne podatke (npr. iz obrasca) i da kao odgovor generišu HTML strane. Ti programi su u stvari skriptovi, obično pisani na Perlu, zato što se Perl skriptovi pišu brže i lakše nego Perl programi (ako uopšte znate da programirate na Perlu). Prema opštem dogovoru, oni se smeštaju u di- rektorijum *cgi-bin,* koji je vidljiv u URL adresi. Ponekad se za pisanje skriptova, umesto Perla, koristi drugi jezik - Python.

Kao primer čestog korišćenja interfejsa CGI, razmotrite slučaj proizvoda hipote- tičke Stvarno Najbolje Svetske Firme, koji se isporučuje bez pismene garancije. Umesto toga, korisnik treba da ode na mrežnu adresu www.snsf.com i da se tamo re- gistruje. Na toj adresi nađi će sledeću hipervezu:

Pritisnite ovde da biste registrovali svoj proizvod

Ta veza ukazuje, na primer, na Perl skript www.snsf.com/cgi-bin/reg.perl. Kada se ovaj skript pozove bez parametara, on odgovara HTML stranom sa obrascem za registraciju. Kada korisnik ispuni obrazac i pritisne dugme *submit,* skriptu se šalje poruka s podacima u stilu slike 7-30. Perl skript tada analizira parametre, pravi za novog kupca odrednicu u bazi podataka i šalje mu HTML stranu s registarskim brojem i telefonskim brojem službe za pomoć. Obrasci se najčešće obrađuju ovako, ali ne uvek. Postoji mnogo knjiga o pisanju CGI skriptova i programiranju na Perlu. Nekoliko pri- mera naći ćete kod Hanegana (2001), Lasha (2002) i Meitzera i Michaiskog (2001).

CGI skriptovi nisu jedini način za generisanje dinamičkog sadržaja na serveru. Drugi uobičajen načinje da se u HTML kod ugrade kratki skriptovi koje će izvršavati server da bi generisao stranu. Popularni jezik za pisanje tih skriptova je **pretprocesor hiperteksta PHP** (engl. *PHP: Hypertext Preprocesor, PHP)*. Da bi koristio PHP, server mora da ga razume (kao što i čitač mora da razume XML da bi protumačio Web strane pisane tim jezikom). Serveri obično očekuju da se datoteke pisane jezikom PHP završavaju sa *php* (a ne sa *html* ili *htm)*.

Na slici 7-34 prikazanje malecki PHP skript koji bi trebalo da radi na svakom serveru koji razume PHP. On sadrži normalan HTML kod, izuzev PHP skripta unutar oznaka <?php ... ?>, i generiše Web stranu na kojoj samo saopštava šta zna o čitaču koji gaje pozvao. Čitači zajedno sa zahtevom obično šalju i neke podatke o sebi (kao i sve prigodne kolačiće), a ti podaci se smeštaju u promenljivu *HTTP_USER_AGENT*. Kada se ovaj listing smesti u datoteku *test.php* u Web direktorijumu kompanije AB- CD, tada ćete po upisivanju adrese www.abcd.com/test.php dobiti Web stranu sa opisom čitača, jezika i operativnog sistema koje koristite.

```
<html>
<body>

<h2> Evo šta znam o tebi </h2>
<?php echo $HTTPJJSER_AGENT ?>

</body>
</html>
```

Slika 7-34. Primer HTML strane sa ugrađenim PHP skriptom.

PHP je posebno zgodan za rukovanje obrascima, a PHP skriptovi su jednostavniji od CGI skriptova. Način njegovog rada razumećete iz primera na slici 7-35(a). Ta slika sadrži normalan HTML kod sa obrascem, osim što se u prvom redu zadaje pozivanje datoteke za obradu parametara (*akcija.php*), nakon što korisnik ispuni i pošalje obrazac. Strana prikazuje dva tekstualna polja za upis imena, odnosno godina starosti. Pošto se polja popune i obrazac pošalje, server analizira tekstualni niz dobijen u stilu slike 7-30, smešta ime u promenljivu *ime,* a godine starosti u promenljivu *starost.* Zatim te parametre obrađuje datoteka *akcija.php,* prikazana na slici 7-35(b) i generiše odgovor. Tokom obrade, izvršavaju se PHP komande. Ako su podaci koje je uneo korisnik: „Barbara“ i „24“, HTML datoteka koja se šalje kao odgovor izgledaće kao na slici 7-35(c). Kao što vidite, kada upotrebite PHP, rukovanje obrascima postaje izuzetno jednostavno.

```
<html>
<body>
<form action="action.php" method="post”>
<p> Upišite svoje ime: <input type="text" name=”ime”> </p>
<p> Upišite starost: cinput type="text“ name="starost"> </p>
cinput type="submit">
</form>
</body>
</html>
```

(a)

```
<html>
<body>
```

```
<h1> Odgovori </h1>
Zdravo <?php echo $ime; ?>.
Predviđam: sledeće godine imaćeš <?php echo $staros1 + 1; ?>
</body>
</html>
```

(b)

```
<html>
<body>
<h1> Odgovor: </h1>
Zdravo Barbara.
Predviđam: sledeće godine imaćeš 25 </body>
</html>
```

(o)

Slika 7-35. (a) Web strana sa obrascem, (b) PHP skript za obradu podataka iz obrasca, (c) Web strana koju generiše PHP skript kada korisnik kao podatke unese „Barbara“ i „24“,

Premda se lako koristi, PHP je moćan programski jezik usmeren na komuniciranje između Weba i severskih baza podataka. On radi s promenljivama, tekstom i nizovima, i ima većinu upravljačkih struktura koje postoje i u jeziku C, ali je u ulazno-iz- laznim operacijama mnogo jači od proste komande *printf.* Kod jezika PHP otvorenje i svakom dostupan. Projektovan je tako da se posebno dobro slaže s najpoznatijim serverom na Webu - serverom Apache, čiji je kod takođe otvoren. Više obaveštenja o jeziku PHP potražite kod Valadea (2002).

Upravo smo videli dva načina za dinamičko generisanje HTML strana: korišcenje CGI skriptova i ugrađivanje PHP koda. Postoji i treća tehnika, **strane Java servera** (engl. *JavaServer Pages*, *JSP),* koja je slična tehnici PHP, osim što se dinamički deo piše programskim jezikom Java, a ne jezikom PHP. Imena strana na kojima je isko- rišćena ova tehnika, imaju nastavak *jsp.* Četvrta tehnika, **aktivne serverske strane** (engl. *Active Server Pages, ASP),* predstavlja Microsoftovu verziju tehnika PHP i JSP. U njoj se za generisanje dinamičnog sadržaja koristi Microsoftov zaštićeni jezik za skriptovanje, Visual Basic Script. Imena strana napravljenih ovom tehnikom nose nastavak *asp.* Izbor između tehnika PHP, JSP i ASP obično je uslovljen više politikom (otvoreni kod u odnosu na Sunov ili Microsoftov zaštićeni kod), nego tehnologijom, pošto su sva tri jezika međusobno uporediva.

Skup tehnologija za generisanje sadržaja **u** hodu, ponekad se naziva **dinamički** HTML (engl. *dynamic HTML).*

Dinamičko generisanje Web strana kod klijenta

CGI slcriptovi i skriptovi pisani na jezicima PHP, JSP i ASP rešavaju problem rukovanja obrascima i komuniciranja s bazama podataka na serveru. Svi oni mogu da prihvate podatke unete u obrasce, da na osnovu njih pretraže jednu ili više baza podataka i da generišu HTML stranu s rezultatima. Nijedan od njih, međutim, ne može da uhvati kretanje miša i da direktno sarađuje s korisnikom. Za tako nešto potrebno je u HTML strane ugraditi skriptove koji će se izvršavati kod klijenta, a ne na serveru. Počev od 4. verzije HTML-a, takvi skriptovi su dozvoljeni ako se ugrade uz oznaku <script>. Najpopularniji jezik za pisanje skriptova koji se izvršavaju kod klijenta jeste JavaScript, pa ćemo mu posvetiti malo pažnje.

JavaScript je ježile za pisanje skriptova, koji *sasvim malo* podseća na programski ježile Java. To izvesno nije Java. Kao i drugi jezici za pisanje skriptova, i ovaj je vrlo visokog nivoa. Na primer, u jednom jedinom redu pisanom na JavaScriptu može se prikazati okvir za

dijalog, sačekati da se unese tekst i zatim ga pridružiti promenljivoj. Takve velike mogućnosti čine JavaScript idealnim jezikom za pravljenje interaktivnih Web strana. S druge strane, zbog toga što nije standardizovan i što mutira brže od vinske mušice izložene gama-zracima, izuzetno je teško napisati programe na JavaScriptu koji rade na svim platformama, ali će se i JavaScript jednog dana možda stabilizovati.

Na slici 7-36 vidite primer programa napisanog na JavaScriptu. Slično programu sa slike 7-35(a), i on prikazuje obrazac s dva polja: imenom i starošću, a zatim - kada korisnik unese podadce - predviđa koliko će godina korisnik imati sledeće godine. Telo programa je gotovo isto kao u PHP primeru; glavnu razliku čini deklaracija dugmeta submit i naredba dodele u njoj. Kada korisnik pritisne dugme mišem, tom naredbom se nalaže čitaču da pozove skript *response* i da mu kao parametar prosledi obrazac.

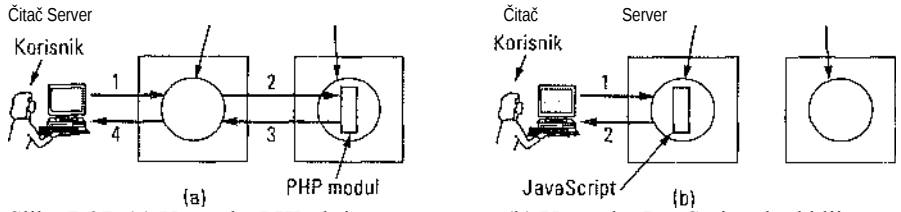
```
<html>
<heađ>
<script language="javascript" type="text/javascript">
funetion response(testjorm) { var korisnik = test_form.ime
.value; var godine = eval(test_form.starost.value) + 1;
document.open();
   documentwriteln(„<html> <body>") ;
   document.writeln(„Zdravo „ + korisnik + ,,.<br>‘j;
   document.writeln(„Predviđam: siedeće godine imaćeš „ + godine +
   document.writeln(„</body> </html>‘j;
   document.close();
}
</script>
</head>
<body>
<form>
Upišite svoje ime: <input type="text“ name="ime”>
<P>
Upišite starost: <input type="text“ name="starost">
```

```
<P>
<input type=”button" value="submit“ onclick="response(this.form)">
</form>
</body>
</html>
```

Slika 7-36. Korišćenje JavaScripta za obradu obrasca.

Potpuno je nova deklaracija funkcije *response* JavaScripta u zaglavlju HTML datoteke, području koje je obično namenjeno naslovima, boji pozadine i sličnim deklaracijama. Ta funkcija izvlači vrednost polja *ime* iz obrasca i smešta je u promenljivu *korisnik* kao tekst. Ona takođe izvlači vrednost polja *starost,* pretvara ga u ceo broj koristeći funkciju *eval,* dodaje mu 1 i rezultat smešta u promenljivu *godine.* Zatim otvara dokument za slanje, upisuje četiri reda metodom *writeln* i zatvara dokument. Dokument je HTML datoteka, što se može zaključiti po raznim HTML oznakama u njemu. Čitač tada prikazuje dokument na ekranu.

Važno je da shvatite da se podaci sasvim različito obrađuju programima sa slika 7-35 i 7-36, iako sami programi međusobno liče. Na slici 7-35, pošto korisnik pritisne dugme *submit,* čitač objedinjuje podatke u dugačak tekstualni niz (kao na slici 7-30) i vraća ga serveru od koga je dobio obrazac. Server vidi ime PHP datoteke i izvršava je. PHP skript generiše novu HTML stranu i šalje je čitaču za prikazivanje. Na slici 7-36, kada korisnik pritisne dugme *submit*, čitač istovremeno prevodi i izvršava (interpretira) funkciju JavaScripta sadržanu u strani. Sve se odvija lokalno, unutar čitača. Ne uspostavlja se veza sa serverom. Zbog toga se rezultat pojavljuje naizgled trenutno, dok uz PHP može da protekne više sekundi dok rezultujuća HTML strana ne stigne do klijenta. Razlike između upotrebe skriptova na serveru i kod klijenta prikazane su na slici 7-37, zajedno sa neophodnim koracima. U oba slučaja, korake smo počeli da numerišemo tek pošto je na korisnikovom ekranu prikazan obrazac. Korak 1 označava prihvatanje podataka koje je upisao korisnik. Zatim dolazi obrada podataka, koja se razlikuje za dva prikazana slučaja.



Slika 7-37. (a) Upotreba PHP skripta na serveru. (b) Upotreba JavaScripta kod klijenta.

Postojeće razlike ne znače daje JavaScript bolji od jezika PHP jer je namena ova dva jezika različita. Jezik PHP (pa, u tom smislu, i jezici JSP i ASP) koristi se kada treba ostvariti komunikaciju sa udaljenom bazom podataka. JavaScript služi za

komunikaciju s korisnikom na njegovom računaru. Moguće je (čak često) da ista HTML strana sadrži i PHP i JavaScript, iako ta dva skripta rade različite stvari i ne mogu da dele isto dugme.

JavaScript je potpun programski jezik koji objedinjuje svu moć jezika C i Java. On ima promenljive, tekstualne i drage nizove, objekte, funkcije i sve uobičajene upravljačke strukture. Prepun je i mogućnosti za rad sa Web stranama, uključujući i rad s prozorima, okvirima, kolačićima, obrascima i hipervezama. Primer JavaScript programa koji koristi rekurzivnu funkciju prikazanje na slici 7-38.

```
<html>
<head>
<script language="javascript“ type=”text/javascript">

function response(test_form) { function factorial(n) {if (n == 0) return 1; else
return n * factorial(n - 1);} var r = eval(test_form.broj.value); // r = upisani
argument document.mojobrazac.mojtekst.value = „Evo rezultataAn"; for
(var i = 1; i <= r; i++) // prikaži jedan red za svaku vrednost od 1 do r
document.mojobrazac.mojtekst.value += (i + „! = „ + factorial(i) + ,,\n‘j;
}
</script>
</head>

<bođy>
<form name="mojobrazac">
Upišite broj: <input type="text“ name="broj">
<input type="button" value='izračunavanje tabele faktorijela" onclick=’’response(this.form)">
<P>
<textarea name="mojteksf rows=25 cols=50> </textarea>
</form>
</body>
</html>
```

Slika 7-38. JavaScript program za izračunavanje i prikazivanje faktorijela.

JavaScript može i da prati kretanje pokazivača miša iznad objekata na ekranu. Mnoge Web strane s JavaScriptom imaju to svojstvo da se pri kretanju pokazivača miša iznad određenog teksta ili slike nešto događa. Taj događaj često je promena slike ili pojava menija. Takvo ponašanje lako se programira u JavaScriptu i omogućava pravljenje „živih“ Web strana. Primer je prikazan je na slici 7-39.

```
<html>
<head>
<script language="javascript“ type="text/javascript"> if
((document.mojurl) document.mojurl = new Array();
document,mojurl[0]= „http://www.cs.vu.nl/ ast/im/maca.jpg";
document.mojurl[1] = „http://www.cs.vu.nl/ ast/im/kuca.jpg”;
document.mojurl[2] = „http://www.cs.vu.nl/ ast/im/zeka.jpg”;
function pop(m) {
    var urx = „http://www.cs.vu.nl/ ast/im/zoo.jpg";
    popupwin = window.open(document.mojurl[m],"mojprozor","width=250,height=250‘‘);
}
</script>
</head>
```

```
<body>
<p> <a href=”#“ onMouseover="pop(0); return false;“ > Maca </a> </p>
<p> <a href="#“ onMouseover="pop(1); return false;“ > Kuca </a> </p>
<p> <a href=”#“ onMouseover="pop(2); return false;“ > Zeka </a> </p>
</body>
</html>
```

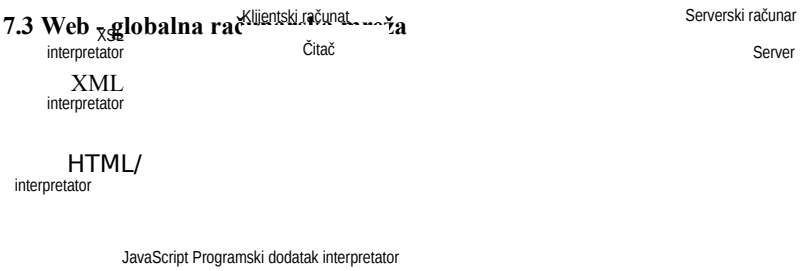
Slika 7-39. Interaktivna Web strana koja reaguje na kretanje miša.

JavaScript nije jedino sredstvo pomoću koga se Web strane mogu oživeti. Druga popularna metoda koristi aplete (engl. *applet*). To su mali programi pisani na Javi i prevedeni u instrukcije mašinskog jezika za virtuelni računar zvan Javina virtuelna mašina (engl. *Java Virtual Machine, JVM).* Apleti se mogu ugraditi u HTML strane (između oznaka <applet> i </applet>) i interpretirati u čitačima koji imaju JVM. Pošto se Javini apleti ne izvršavaju direktno, već se prethodno prevode (interpretiraju), Ja- vin interpretator može da ih spreči da ne „rade ružne stvari“. Barem u teoriji. U praksi su, međutim, autori apleta uspeli da zloupotrebe skoro beskonačan niz propusta u Ja- vinim ulazno-izlaznim bibliotekama.

Microsoftov odgovor na Sunove Java aplete bile su ActiveX kontrole (engl. *ActiveX Controls)* - programi prevedem na mašinski jezik Pentiuma koji se hardverski izvršavaju. Zbog toga su mnogo brži i fleksibilniji od interpretiranih Java apleta jer mogu da urade sve što i svaki drugi program. Kada Internet Explorer ugleda ActiveX kontrolu na Web strani, on je preuzme, proveri joj identitet i izvrši je. Međutim, preuzimanje i izvršavanje programa drugih proizvođača skopčano je s bezbednosnim problemima, o čemu ćemo govoriti u 8. poglavlju.

Pošto skoro svi čitači mogu da interpretiraju programe pisane na Javi i JavaScrip- tu, autorima koji žele da naprave pretežno interaktivnu Web stranu stoje na raspolaganju barem dve tehnike, a ako se ne postavlja pitanje prenosivosti između različitih platformi, tu su i ActiveX kontrole. U načelu važi da se JavaScript programi najlakše pišu, da se Javini apleti izvršavaju brže, a da se ActiveX kontrole izvršavaju najbrže. Takođe, pošto svi čitači sadrže potpuno istu Javinu virtuelnu mašinu, ali ni dva čitača nemaju istu verziju JavaScripta, Javini apleti su prenosiviji od programa u JavaScrip- tu. O JavaScriptu je napisano mnogo debelih knjiga, često i sa više od 1000 stranica. Neke ćete naći i u pregledu literature na kraju ove knjige (Easttom, 2001; Harris, 2001; McFedries, 2001).

Pre nego što zaključimo temu dinamičnog Web sadržaja, napravimo pregled onoga što smo o tome naučili. Potpune Web strane mogu se praviti u hodu, pomoću različitih skriptova na serveru. Kada ih preuzme, čitač s njima postupa kao s normalnim HTML, stranama i samo ih prikazuje. Skriptovi se mogu pisati na Perlu i na jezicima PHP,JSP i ASP (slika 7-40),



Slika 7-40. Načini generisanja i prikazivanja sadržaja.

Sadržaj se može dinamički generisati i kod klijenta. Web strane se mogu pisati na jeziku XML, a zatim XML datoteka prevesti u HTML. Pomoc'u programa pisanih na JavaScriptu mogu se vršiti proizvoljna izračunavanja. Na kraju, za prikazivanje sadržaja u različitim formatima mogu se koristiti programski dodaci i pomoćne aplikacije.

7.3.4 HTTP - protokol za prenos hiperteksta

Protokol za prenos podataka koji se koristi širom Weba zove se **protokol za prenos hiperteksta** (engl. *HyperText Transfer Protocol, HTTP)*. Njime se definišu poruke koje klijenti mogu da šalju serverima, kao i odgovori servera. Svaka interakcija sadrži jedan tekstualni (ASCII) zahtev i odgovor tipa MIME prema standardu RFC 822. Svi klijenti i svi serveri moraju da poštuju ovaj protokol. On je definisan u RFC dokumentu 2616. U ovom odeljku osvrnucemo se na njegova važnija svojstva.

Veze

Čitač se sa serverom obično povezuje tako što uspostavi TCP vezu s njegovim priključkom 80, premda taj postupale s formalnog stanovišta nije obavezan. Vrednost TCP veze ogleda se u tome što ni čitači ni serveri ne moraju da brinu o izgubljenim porukama, duplikatima, dugačkim porukama ili potvrdama. O svemu tome misli TCP.

U verziji 1.0 protokola HTTP, pošto se veza uspostavi, šalje se samo jedan zahtev i dobija samo jedan odgovor. Tada se TCP veza raskida. Kada su se Web strane sastojale isključivo od HTML teksta, takav postupak je sasvim odgovarao. Međutim, Web strane su se ubrzano kitile ikonicama, slikama i drugim vizuelnim ukrasima, pa je uspostavljanje TCP veze za prenošenje samo jedne ikonice ili slike postalo preskupo.

Iz tog razloga nastala je verzija 1.1 protokola HTTP, koja podržava **trajne veze** (engl. *persistent connections).* Sada su se protokolom HTTP, nakon uspostavljanja TCP veze i jednokratne razmene zahteva i odgovora, mogli slati i dodatni zahtevi i primati odgovori na njih. Ukidanjem uspostavljanja i raskidanja TCP veze za svaki zahtev smanjenje broj nepotrebnih operacija po zahtevu. Zahtevi se mogu slati i serijski, tj. može se poslati zahtev 2 pre nego što stigne odgovor na zahtev 1.

Metode

Iako je protokol HTTP prvenstveno projektovan za rad na Webu, namemo je napravljen šire nego što je potrebno da bi se eventualno mogao iskoristiti za buduće objektno orijentisane aplikacije. Zbog toga su u njemu, osim operacija koje se direktno tiču slanja zahteva i odgovaranja na njih, podržane i druge operacije, tzv. metode. Takva širina pristupa omogućila je da zaživi i protokol SOAP. Svaki zahtev se sastoji od jednog ili više redova ASCII teksta, pri čemu prva reč u prvom redu označava zahtevanu metodu. Spisak ugrađenih metoda prikazan je na slici 7-41. Za pristupanje objektima uopšte mogu se koristiti i dopunske metode specifične za te objekte. U imenima se pravi razlika između velikih i malih slova, tako da je *GET* legalna metoda, a *get* nije.

Metoda	Opis
GET	Zahtev za učitavanje Web strane
HEAD	Zahtev za učitavanje zaglavlja Web strane
PUT	Zahtev za skladištenje (upisivanje) Web strane
POST	Dopunjavanje imenovanog resursa (npr. Web strane)
DELETE	Uklanjanje Web strane
TRACE	Slanje odjeka dolaznog zahteva
CONNECT	Rezervisano za buduću upotrebu
OPTIONS	Pregledanje izvesnih opcija

Slika 7-41. Ugrađene HTTP metode zahtevanja.

Metodom *GET* od servera se zahteva da pošalje stranu (u najopštijem slučaju - ob- jekat, ali najčešće samo datoteku). Pogodno je ako je strana kodirana prema standardu MIME. Velika većina zahteva serverima sadrži metodu *GET.* Ona se najčešće koristi u obliku

```
GET datoteka HTTP/1.1
```

gde *datoteka* predstavlja ime traženog resursa (datoteke), a 1.1 je verzija korišćenog protokola.

Metodom *HEAD* zahteva se samo zaglavlje poruke - bez odgovarajuće Web strane. Ona se koristi za provera datuma poslednje izmene strane, za prikupljanje podataka za indeksiranje ili za proveru ispravnosti URL adrese.

Metoda *PUT* je suprotna metodi *GET.* umesto da stranu učitava, ona je upisuje. Tom metodom se može napraviti zbirka Web strana na udaljenom serveru. Telo zahteva sadrži stranu. Ona se može kodirati sistemom MIME, kada redovi koji slede metodu *PUT* mogu da uključe kako zaglavlje *Content-Type,* tako i zaglavlje za potvrdu identiteta i time utvrdi da lije pozivalac zaista ovlašćen za zahtevanu operaciju.

Metoda *POST* pomalo liči na metodu *PUT.* I ona sobom nosi URL adresu, ali umesto da zameni postojeće podatke, ona ih u određenom smislu njima priključuje. Slanje poruke diskusionoj grupi ili njeno objavljivanje na elektronskoj oglasnoj tabli predstavljaju u ovom kontekstu „priključivanje“ već postojećim porukama. U praksi se metode *PUT* i *POST* ne koriste prečesto.

Metoda *DELETE* radi ono što i očekujete: uklanja postojeće strane. Kao kod metode *PUT,*

i ovde identifikovanje i dozvola za obavljanje operacije igraju važnu ulogu. Nema garancije za uspešan ishod primene metode *DELETE-,* čak i kada je udaljeni HTTP server voljan da obriše stranu, možda je njena datoteka u režimu koji HTTP serveru ne dozvoljava daje izmeni ili obriše.

Metoda *TRAČE* služi pronalaženju grešaka. Ona nalaže serveru da zahtev pošalje tamo odakle je došao. Korisna je onda kada klijent shvati da se zahtevi ne obrađuju na odgovarajući način i želi da sazna kakav je u stvari zahtev server dobio.

Metoda *CONNECT* se trenutno ne koristi. To je rezerva za budućnost.

Pomoću metode *OPTIONS* klijent može da pregleda svojstva servera ili zadate datoteke.

Na svaki zahtev šalje se odgovor koji sadrži statusni red, a možda i dopunske informacije (npr. celu Web stranu ili samo njen deo). Statusni red sadrži trocifren statusni kod kojim se saopštava da li je zahtev zadovoljen ili nije, a ako nije, i zašto. Prvom cifrom se odgovor svrstava u jednu od pet glavnih grupa (slika 7-42). Kodovi lxx retko se koriste u praksi. Kodovi 2xx znače daje zahtev uspešno obrađen i daje kao odgovor poslat sadržaj (ako postoji). Kodovi 3xx upućuju klijenta da odgovor potraži na nekom drugom mestu: na drugoj URL adresi ili u sopstvenom kešu (o čemu ćemo govoriti kasnije). Kodovi 4xx znače da zahtev nije uspeo zbog klijentove greške ili zbog nepostojeće Web strane. Na kraju, kodovi 5xx znače da sam server ima problema, bilo zbog greške u kodu ili zbog preopterećenja.

Kod	Značenje	Primerl
1xx	Informisanje	100 = server se slaže da obradi klijentov zahtev
2xx	Uspeh	200 = zahtev uspešan; 204 = ne postoji sadržaj
3xx	Preusmeravanje	301 = strana je premeštena; 304 = strana još postoji u kešu
4xx	Greška kod klijenta	403 = zabranjena strana; 404 = strana nije nađena
5xx	Greška na serveru	500 = interna greška servera; 503 = pokušaj opet kasnije

Slika 7-42. Grupe odgovora prema statusnom kodu.

Zaglavlja poruka

Iza reda sa zahtevom (npr. reda u kome je metoda *GET)* mogu slediti redovi s dodatnim informacijama. To su zaglavlja zahteva (engl. *request headers).* Podaci u njima mogu se uporediti s parametrima koji se prosleđuju pri pozivanju procedure. I odgovori mogu da imaju zaglavlja odgovora (engl. *response headers).* Neka zaglavlja se mogu koristiti u oba smera. Na slici 7-43 prikazana su najvažnija zaglavlja.

Zaglavlje *User-Agent* omogućava klijentu da obavesti server o svom čitaču, operativnom sistemu i dragim svojstvima. Na slici 7-34 videli smo da server ima te informacije i da ih na zahtev može proslediti PHP skriptu. Te informacije mu je poslao klijent u navedenom zaglavlju.

Zaglavlje	Tip	Sadržaj
User-Agent	Zahtev	Informacije o čitaču i platformi klijenta
Accept	Zahtev	Tip strana s kojima klijent može da radi
Accept-Charset	Zahtev	Skupovi znakova koje klijent podržava
Accept-Encoding	Zahtev	Načini kodiranja strana koje klijent podržava
Accept-Language	Zahtev	Govorni jezici koje klijent podržava
Host	Zahtev	DNS ime servera
Authorization	Zahtev	Lista klijentovih akreditiva
Cookie	Zahtev	Šalje serveru prethodno primljen kolačić
Date	Oboje	Datum i vreme slanja poruke
Upgrade	Oboje	Protokol na koji želi da pređe pošiljalac
Server	Odgovor	Informacije o serveru
Content-Encoding	Odgovor	Način kodiranja sadržaja (npr. gzip)
Content-Language	Odgovor	Govorni jezik korišćen na strani
Content-Length	Odgovor	Dužina strane u bajtovima
Content-Type	Odgovor	MIME tip strane
Last-Modified	Odgovor	Vreme i datum poslednje izmene strane
Location	Odgovor	Komanda klijentu da zahtev pošalje na drugo mesto
Accept-Ranges	Odgovor	Server prihvata zahteve za stranu u delovima
Set-Cookie	Odgovor	Server želi da pošalje kolačić klijentu

Slika 7-43. Neka zaglavlja HTTP poruka.

Četiri zaglavlja *Accept* obaveštavaju server da je klijent spreman da prihvati odgovor u slučaju kada je podešen da ne prihvata sve i svašta. Prvo zaglavlje navodi dobrodošle MIME tipove (npr. text/html). Drago navodi skup znakova (npr. ISO-8859-5 ili Unicode-1-1). Treće sadrži metodu komprimovanja podataka (npr. gzip). Četvrto zaglavlje ukazuje na govorni jezik (npr. španski). Ako server ima više verzija Web strane, moći će klijentu da pošalje njenu odgovarajuću verziju. Ako to ne bude mogao, vratiće kod greške i zahtev neće uspeti.

Zaglavlje *Host* imenuje server. Ono se preuzima iz URL adrese. Zaglavlje je obavezno jer neke IP adrese opslužuju više DNS imena, a server mora da zna kom računani da uputi zahtev.

Zaglavlje *Authorization* je obavezno kada se traži zaštićena strana. U tom slučaju, klijent mora da dokaže daje ovlašćen da joj pristupi, a dokaz šalje upravo u pomenu- tom zaglavlju.

Iako kolačići zapravo ne spadaju u RFC dokument 2616, već u RFC dokument 2109, i za njih postoje dva zaglavlja. Zaglavlje Cookie koriste klijenti da bi servera vratili kolačić koji su ranije dobili od nekog računara iz njegovog domena.

Zaglavlje *Date* se može koristiti i u zahtevu i u odgovoru, a sadrži vreme i datum slanja poruke. Zaglavlje *Upgrade* treba da olakša prelazak na buduću (možda nekompatibilnu) verziju protokola HTTP. Ono omogućava klijentu da objavi šta može da podrži, i serveru - da potvrdi šta koristi.

Dolazimo do zaglavlja koja koristi isključivo server kada odgovara na zahteve. Prvo zaglavlje, *Server,* omogućava serveru da se predstavi i da uz to, ako želi, navede i neka svoja

svojstva.

Sledeća četiri zaglavlja koja počinju na *Content-,* omogućavaju serveru da opiše sadržaj strane koju šalje.

Zaglavlje *Last-Modified* sadrži vreme i datum poslednje izmene strane, i važno je za keširanje strana.

Pomoću zaglavlja *Location* server upućuje klijenta na drugu URL adresu u slučajevima kada je tražena Web strana premeštena ili se ista strana može naći na više URL adresa (možda na različitim serverima). Njega koriste i kompanije koje glavnu Web stranu drže u domenu *com,* ali klijente usmeravaju na nacionalne ili regionalne Web strane u zavisnosti od njihovih IP adresa ili govornog jezika.

Ako je strana vrlo velika, skromniji klijenti je možda neće želeti odjednom celu. Neki serveri će prihvatiti zahteve za najvećom količinom bajtova koja se u jednom trenutku može prihvatiti i stranu poslati u više manjih delova. Zaglavljem *Accept- Ranges* server najavljuje spremnost da obradi ovu vrstu zahteva za delimičnim slanjem strane.

Set-Cookie je zaglavlje pomoću koga server šalje kolačiće klijentima. Od klijenta se očekuje da dobijeni kolačić sačuva i vrati ga pri sledećem zahtevu serveru.

Primer korišćenja protokola HTTP

Pošto je HTTP tekstualni protokol, korisnik ne mora da pokreće čitač, već direktno s terminala može da komunicira sa Web serverom. Treba mu samo TCP veza s priključkom 80 servera. Preporučujemo čitaocima da ovo sami isprobaju (najbolje s nekog UNIX sistema jer neki drugi sistemi ne prikazuju status veze). Za to je potrebno izdati sledeće komande:

```
teinet www.ietf.org 80 >log
GET /rfc.html HTTP/1.1
Host: www.ietf.org

close
```

pomoću kojih se prvo uspostavlja teinet (tj. TCP) veza s priključkom 80 Web servera organizacije IETF (www.ielf.org). Rezultat sesije se preusmerava u datoteku *log* za kasnije pregledanje. Zatim dolazi komanda *GET* sa imenom datoteke i protokolom. U sledećem redu obavezno je zaglavlje *Host.* I prazan red je obavezan. On saopštava serveru da nema više zaglavlja zahteva. Komandom *close* programu teinet se nalaže da raskine vezu.

Datoteku *log* možete pregledati u bilo kom programu za uređivanje teksta. Ona treba da počinje približno onako kao što je prikazano na slici 7-44, osim ako ju je IETF nedavno izmenio.

Prva tri reda na slici 7-44 ne potiču sa udaljene lokacije, već ih ispisuje program telnet. Redom koji počinje sa HTTP/l. 1, lokacija IETF saopštava spremnost da s vama razgovara pomoću protokola HTTP/1.1. Zatim sledi niz zaglavlja, pa onda sadržaj. Već smo videli sva zaglavlja, osim zaglavlja *ETag,* koje je jedinstven identi- fikator strane u smislu njenog keširanja, i zaglavlja *X-Pad,* koje nije standardno i tu je da pomogne nekom nesolidnom čitaču Weba.

```
Trying 4.17.168.6...
Connected to www.ietf.org.
Escape character is HTTP/1.1 200 OK
Date: Wed, 08 May 2002 22:54:22 GMT
Server: Apache/1.3.20 (Unix) mod_ssl/2.8.4 OpenSSL/0.9.5a
```

```
Last-Modified: Mon, 11 Sep 2000 13:56:29 GMT
ETag: ,,2a79d-c8b-39bce48d"
Accept-Ranges: bytes Content-Length: 3211 Content-Type:
text/html X-Pad: avoid browser bug

<html>
<head>
<title>IETF RFC Page</title>

<script language="javascript"> function url() {
var x = document.forml.number, value if (x.length == 1) {x =
„000“ + x} if (x.length == 2) {x = „00“ + x} if (x.length == 3) {x
= „0“ + x} document.forml.action = ,,/rfc/rfc“ + x + ,,.txt"
document.forml .submit }
</script>

</head>
```

Slika 7-44. Početak ispisa datoteke www.ieft.org/rfc.html.

7.3.5 Poboljšanje performansi

Popularnost Weba je skoro dovela do njegovog uništenja jer su serveri, usmerivači i linije često preopterećeni. Mnogi su Globalnu računarsku mrežu (engl. *World Wide Web,* WWW) počeli ironično da nazivaju Globalno čekanje (engl. *World Wide Wait,* WWW). Zbog sve češćeg - naizgled beskonačnog - čekanja da se Web strana učita, istraživači su razvili različite tehnike za poboljšavanje performansi. Na ovom mestu opisaćemo tri takve tehnike: keširanje, kopiranje Web servera, i mreže za isporuku sadržaja.

Keširanje

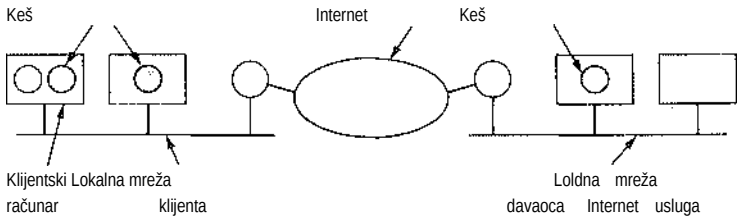
Prilično jednostavan način poboljšavanja performansi sastoji se u privremenom skladištenju već isporučenih strana za slučaj da ih klijent ponovo traži. Ta tehnika posebno daje rezultate za strane koje se često posecuju, kao što su www.yahoo.com i www.cnn.com. Privremeno skladištenje strana za eventualno buduće lcorišćenje naziva se **keširanje** (engl. *caching).* To privremeno skladište (keš) obično održava poseban **zastupnički Web proces** (engl. *proxy).* Tada se čitač može podesiti da zahteve za strane ne šalje udaljenom serveru, već tom procesu. Ako proces u kešu pronađe traženu stranu, on je odmah prikazuje. Ako u kešu nema tražene strane, zastupnički Web proces je preuzima sa udaljenog servera, stavlja je u keš za kasniju upotrebu i prikazuje je klijentu.

U vezi s keširanjem postavljaju se dva važna pitanja:

1. Ko treba da koristi keširanje?
2. Koliko dugo treba čuvati keširane strane?

Postoji više odgovora na prvo pitanje. Na ličnim PC računalima često se izvršava zastupnički Web proces tako da se ranije posećene strane mogu skoro trenutno prikazati. U lokalnim mrežama preduzeća, zastupnički proces je u stvari poseban zastupnički server koji dele svi računati, tako da istu stranu iz zastupnikovog lceša može da preuzme svaki korisnik na lokalnoj mreži. Mnogi davaoci Internet usluga takođe imaju zastupničke servere, kako bi svojim korisnicima što više ubrzali pristup često traženim Web stranama. Obično zastupnički Web procesi (serveri) rade istovremeno, tako da se zahtev najpre upućuje lokalnom

zastupničkom Web procesu. Ako se tražena strana ne nađe, lokalni proces pretražiće je zastupnički server lokalne mreže. Ako strane ni tamo nema, zastupnički server lokalne mreže će je potražiti u kešu zastupničkog servera davaoca Internet usluga. Ovo poslednje mora da uspe; strana se mora naći u samom kešu, u kešu višeg nivoa ili na samom serveru. Sistem kesa razmeštenog na više nivoa koji se na zahtev uzastopno pretražuju zove se **hijerarhijsko Web keširanje** (engl. *hierarchical caching*). Njegova moguća realizacija prikazana je na slici 7-45.



Slika 7-45. Hijerarhijsko keširanje u tri stupnja.

Dužina čuvanja Web strana u kešu pomalo je neodređena. Neke Web strane uopšte ne treba keširati. Na primer, strana koja sadrži 50 najaktivnijih berzanskih akcija menja se praktično svake sekunde. Ako bi se keširala, korisnik kome bi se prikazala kopija iz keša dobio bi zastarele podatke. S druge strane, kada se na kraju dana berza zatvori, strana će biti sveža tolcom više sati ili dana, sve dok ne počne nov promet akcija. Prema tome, potreba za keširanjem iste strane može da se menja od trenutka do trenutka.

Pri određivanju perioda čuvanja strane u kešu, osnovni činilac je stepen svežine podataka koji korisnici mogu da prihvate. (Pošto se keširane strane čuvaju na disku, veličina skladišnog prostora retko predstavlja problem.) Ukoliko zastupnički server prebrzo odbacuje keširane strane, gotovo uvek će imati sveže podatke, ali neće raditi baš efikasno (tj. tražena strana će se retko naći u kešu). Ako, pak, strane čuva predugo, korisnici će ih uglavnom tamo i naći, ali često s bajatim podacima.

Rešavanju opisanog problema može se pristupiti na dva načina. Prvi način je pronalaženje sistema na osnovu koga bi se moglo zaključiti koliko dugo treba čuvati stranu. Osnova za to obično je zaglavlje *Last-Modified* (slika 7-43). Ako je strana izmenjena pre jedan sat, ona se u kešu čuva takođe jedan sat. Ukoliko je izmenjena pre godinu dana, očigledno je veoma stabilna (neka je to npr. spisak božanstava iz grčke i rimske mitologije), i zato se čuva godinu dana uz razumno očekivanje da se tokom tog perioda neće izmeniti. Premda ovakav sistem u praksi često radi dobro, s vremena na vreme se i u njemu mogu pronaći zastarele strane.

Drugi načinje skuplji, ali otklanja rizik od zastarevanja strana pomoću specijalnih mogućnosti rada s kešom, opisanih u RFC dokumentu 2616. Najkorisnija mogućnost je zaglavlje *If-Modified-Since,* koje zastupnički Web proces može da pošalje serveru. Njime se naznačava strana koju zastupnik traži i vreme kada je ona poslednji put menjana (iz zaglavlja *Last-Modified).* Ako strana od tada nije menjana, server odgovara kratkom porukom *Not Modified* (statusni kod 304 na slici 7-42), što znači da zastupnik može da upotrebi keširanu stranu. Ukoliko je strana od tada menjana, server šalje svežu stranu. Iako je za ovaj pristup uvek potrebno razmeniti poruke sa zahte- vom i odgovorom, odgovor je veoma kratak ako je

strana u kešu još uvek sveža.

Opisana dva načina rada s kešom mogu se spojiti. Tokom zadatog perioda *AT* od prvog preuzimanja strane, zastupnik je šalje svakom ko je zahteva. Posle izvesnog vremena, zastupnik proverava njenu svežinu porukom *If-Modified-Since*. Za biranje perioda *AT* potrebna je određena strategija koja zavisi od toga kada je strana prethodno bila izmenjena.

Web strane s dinamičnim sadržajem (npr. generisane PHP skriptom), nikada ne treba keširati jer se parametri mogu izmeniti od zahteva do zahteva. Za takve i slične slučajeve postoji opšti mehanizam pomoću koga server obaveštava sve zastupnike na putanji do klijenta da tekuću stranu ne koriste ponovo dok ne provere njenu svežinu. Taj mehanizam se može upotrebiti za sve strane za koje se očekuje da se često menja- ju. U RFC dokumentu 2616 definisani su i drugi mehanizmi rada s kešom.

Još jedan način poboljšavanja performansi je keširanje unapred (engl. *proactive caching*). Kada zastupnik preuzme stranu sa servera, on na njoj može da potraži hiperveze, a zatim da od odgovarajućih servera preuzme strane na koje hiperveze ukazuju i smesti ih u svoj keš - za svaki slučaj. Ta tehnika inože da skrati vreme pristupanja pri uzastopnim zahtevima, ali i zatrpava komunikacione linije stranama koje nikada nilco ne traži.

Jasno je da keširanje Web strana nije jednostavno. O tome bi se moglo pričati i pričati. U stvari, toj temi su posvećene i mnoge knjige (na primer, Rabinovich i Spatscheck, 2002; Wessels, 2001). Međutim, vreme je da pređemo na sledecu temu.

Kopiranje servera

Keširanje je tehnika poboljšavanja performansi koja se primenjuje kod klijenta, ali postoje i tehnike koje se primenjuju na serveru. Serveri najčešće pokušavaju da poboljšaju svoj rad tako što sadržaj koji nude kopiraju na više međusobno udaljenih lokacija. Ta tehnika se ponekad zove kopiranje (ili preslikavanje) servera (engl. *mirroring)*.

Tipičnu upotrebu ove tehnike možete videti na glavnim Web stranama nekih kompanija, gde su, pored nekoliko slika, navedene i veze ka njihovim, recimo, Istočnim, Zapadnim, Severnim i Južnim Web lokacijama. Korisnik tada bira lokaciju koja mu je najbliža i dalje sav saobraćaj ide preko nje.

Kopirane lokacije (lokacije blizanci) u načelu su potpuno statične. Kompanija odlučuje gde će postaviti kopije, zakupljuje servere u svakom području i na svaku lokaciju postavlja manje-više isti sadržaj (možda izostavljajući snežne pejzaže s lokacije u Majamiju i suncobrane za plažu s lokacije na Aljasci). Lokacije se najčešće ne menjaju mesecima, čak ni godinama.

Nažalost, na Webu je česta pojava iznenadnih zagušenja (engl. *flash crowds*), kada do tada potpuno nepoznata, neposećena i zabačena Web lokacija odjednom postaje centar sveta. Na primer, do 6. novembra 2000. godine, lokacija Državnog sekretara Floride (www.dos.state.fl.us) tiho je objavljivala detalje sa sastanaka Državnog kabineta Floride i uputstva tipa „kako da postanete beležnik u Floridi“. Sve to, međutim, izmenilo se 7. novembra, u trenutku kada je izbor sledećeg predsednika SAD zavisio samo od par hiljada glasova iz nekoliko okruga na Floridi. Pomenuta Web lokacija je u tom trenutku postala jedna od pet najposećenijih lokacija na svetu. Ne treba ni naglašavati da lokacija nije mogla da izdrži taj pritisak i da je gotovo „pukla“ pokušavajući da mu se odupre.

Iz navedene priče sledi daje neophodan mehanizam pomoću koga će Web lokacija koja oseti masovno povećanje saobraćaja, moći da se automatski klonira na onoliko lokacija

koliko je potrebno i da ih održava u radu sve dok oluja ne prođe, posle čega mnoge od njih može ugasiti, ako ne i sve. Web lokacija to može ako od kompanija koje poseduju mnoge servere unapred dobije saglasnost da se u slučaju potrebe repli- lcuje na njih i da plaća za realno zauzete kapacitete.

Još fleksibilnija strategija predviđa dinamičko pravljenje kopija samo sa onim stranama koje se u određenoj geografskoj oblasti traže. Neka istraživanja u tom smislu obavili su Pierre i saradnici 2001. i 2002. godine.

Mreže za isporuku sadržaja

Vrhunac kapitalističke ekonomije je činjenica da je neko uspeo da napravi biznis i od Globalnog čekanja. To ide ovako. Kompanije zvane **mreže za isporuku sadržaja** (engl. *Content Delivery Networks, CDNs)* pregovaraju s davaocima sadržaja (s muzičkim i novinskim lokacijama, kao i sa svim drugim lokacijama kojima je stalo da ga brzo i lako stave na raspolaganje korisnicima) i nude im da uz naknadu njihov sadržaj efikasno isporučuju krajnjim korisnicima. Pošto se ugovor potpiše, vlasnik sadržaja predaje CDN-u sadržaj svoje Web lokacije na prethodnu obradu (o kojoj ćemo ubrzo govoriti) i potom, distribuciju.

CDN zatim pregovara s brojnim davaocima Internet usluga i nudi dobar novac da na njihovu lokalnu mrežu postavi daljinski upravljan server krcat vrednim sadržajem. Za davaoca Internet usluga to nije samo jednokratan izvor prihoda, već se time umnogome skraćuje i vreme pristupa sadržaju koji obezbeđuje CDN, pa davalac dobija relativnu prednost u odnosu na drage davaoce koji nisu prihvatili ponudu CDN-a. Pod ovim uslo- vima, kada davalac Internet usluga potpiše ugovor s CDN-om, nema šta da brine. Iz tih razloga, najveće mreže za isporuku sadržaja imaju i više od 10.000 servera širom sveta.

Kada se sadržaj kopira na hiljade lokacija širom sveta, jasno je da se mogu postići odlične performanse. Međutim, da bi to radilo, mora postojati način da se klijentov zahtev preusmeri na najbliži CDN server, najbolje na lokalni CDN server kod njegovog davaoca Internet usluga. To preusmeravanje, takođe, ne sme da poremeti sistem DNS, niti bilo koju drugu standardnu infrastrukturu na Internetu. Sledi neznatno pojednostavljen opis rada najveće CDN kompanije, Akamai.

Čitav proces započinje kada davalac sadržaja preda CDN kompaniji svoju Web lokaciju. CDN kompanija tada svaku stranu lokacije propušta kroz pretprocesor - program koji u njima menja sve URL adrese. Prema viđenju CDN kompanije, Web lokacija davaoca sadržaja najčešće ima mnogo „tankih“ Web strana (uglavnom sa HTML tekstom), koje su, međutim, povezane s velikim grafičkim, audio ili video datotekama. Zbog toga, CDN kompanija ostavlja HTML strane sa izmenjenim URL adresama na servera davaoca sadržaja kako bi se mogle preuzimati kao i do tada, a slike, audio i video datoteke prebacuje na svoj server.

Da biste razumeli kako stvarno radi ova sema, razmotrite Web stranu Krzneni Video na slici 7-46(a). Posle prethodne obrade, ta strana izgleda kao na slici 7-46(b) i postavlja se na server firme Krzneni Video, na adresu www.krznenivideo.com/index.html.

```
<html>
<head> <title> Krzneni Video </title> </head>
<body>
<h1> Krzneni Video - Spisak proizvoda </h1>
<p> Pritisnite da biste dobili besplatne uzorke. </p>
```

```
<a href="medvedi,mpg"> Medvedi danas </a> <br>
<a href="zecevi.mpg"> Sve sami zečevi </a> <br>
<a href="misevi.mpg"> Slatki mali mišići </a> <br>
</body>
</html> <btml>
<head> <title> Krzneni Video </title> </head>
<body>
<h1 > Krzneni Video - Spisak proizvoda </h1 >
<p> Pritisnite da biste dobili besplatne uzorke. </p>

<a href="httpi//cdn-servercom/krznenivideo/medvedi.mpg"> Medvedi danas </a> <br>
<a href="http://cdn-server.com/krznenivideo/zecevi.mpg"> Sve sami zečevi </a> <br>
<a href="http://cdn-server.com/krznenivideo/misevi.mpg"> Slatki mali mišići </a> <br>
</body>
</html>
```

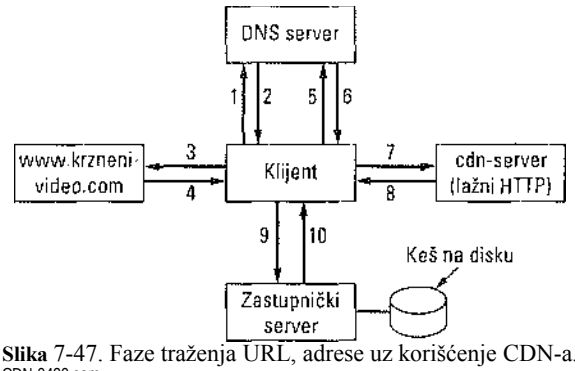
(b)

Slika 7-46. (a) Originalna Web strana, (b) Ista strana, nakon transformacije.

Kada korisnik upise URL adresu www.krznenivideo.com, DNS vrača autentičnu IP adresu Web lokacije Krzneni Video i omogućuje da se njena glavna (HTML) strana preuzme na normalan način. Kada korisnik na njoj pritisne bilo koju hipervezu, čitač traži od DNS servera IP adresu domena *cdn-server.com,* a ovaj mu je isporučuje. Čitač tada šalje HTTP zahtev na dobijenu IP adresu, očekujući da s nje dobije MPEG datoteku.

To se, međutim, ne događa jer *cdn-server.com* nema nikakav sadržaj. To je lažni HTTP server CDN kompanije. On ispituje ime datoteke i servera da bi utvrdio koja se strana traži i od kog davaoca sadržaja. On ispituje i IP adresu dolaznog zahteva, i pregleda svoju bazu podataka da bi utvrdio gde se korisnik približno nalazi. Snabde- ven tim podacima, on određuje CDN server sadržaja koji će korisniku pružiti najbolju uslugu. Ta odluka nije laka, jer geografski najbliži server ne mora biti najbliži i u smislu topologije mreže, a onaj koji je topološki najbliži može u tom trenutku biti veoma zaposlen. Pošto donese odluku, *cdn-server.com* šalje odgovor sa statusnim kodom 301 i zaglavljem *Location* sa URL adresom datoteke na CDN servera sadržaja koji je najbliži klijentu. Primera radi, pretpostavimo daje ta URL adresa *www.CDN- 0420.com/krz.nenivideo/medvedi.mpg.* Čitač tada obrađuje ovu URL adesu na uobičajen način da bi dobio traženu MPEG datoteku.

Pojedini koraci ovog postupka prikazani su na slici 7-47. Prvi korak je traženje IP adrese domena www.krz.nenivideo.com. Posle toga se HTML strana može preuzeti i prikazati na uobičajeni način. Strana sadrži tri hiperveze ka domenu *cdn-server.com* [slika 7-46(b)]. Kada se izabere, recimo, prva hiperveza, od DNS servera se traži njena IP adresa (korale 5) i vraća čitaču (korak 6). Kada zahtev za datoteku *medve- di.mpg* stigne servera *cdn-seiyer.com* (korak 7), klijent se upućuje na *CDN-0420.com* (korak 8). Kada klijent učini kako mu je naloženo (korak 9), dobija datoteku iz zastupničkog lceša (korak 10). Komponenta koja omogućava da ovaj sistem radi nalazi se u koraku 8; to je lažni HTTP server koji klijenta preusmerava na zastupnički CDN server koji mu je najbliži.



Slika 7-47. Faze traženja URL, adrese uz korišćenje CDN-a.
CDN-0420.com

1. Traženje IP adrese domena www.krznenivideo.com
2. Vraćanje IP adrese lokacije Krzneni Video
3. Traženje HTML strane od lokacije Krzneni Video
4. Vraćanje HTML strane
5. Posle pritiska na hipervezu, traži se IP adresa domena cdn-server.com
6. Vraćanje IP adrese domena cdn-server.com
7. Traženje datoteke medvedi.mpg od servera cdn-server.com
8. Klijentu se nalaže da se preusmeri na CDN-0420,com
9. Zahtevanje datoteke medvedi.mpg 10. Vraćanje keširane datoteke medvedi.mpg.)

CDN server na koji se klijent preusmerava najčešće je zastupnički server s velikim kesom prepunim najvažnijeg sadržaja. Ako, međutim, neko traži stranu koja nije keši- rana, ona se preuzima s pravog servera i smešta u keš za eventualnu buduću upotrebu. Kada umesto potpune kopije servera upotrebi zastupnički server, CDN može da pravi različite kompromise između veličine diska, vremena keširanja i parametara performansi.

Više o mrežama za isporuku sadržaja saznaćete kod Hulla (2002), i Rabinovicha i Spatschecka (2002).

7.3.6 Bežični Web

Prilična je potražnja za malim prenosivim uređajima koji mogu da pristupe Webu bežičnim putem. U stvari, već su načinjeni prvi grubi koraci u tom pravcu. Nema sumnje da će narednih godina u ovoj oblasti doći do velikih promena, ali je ipak korisno ispitati neke sadašnje pristupe bežičnom Webu da biste sagledali gde smo sada i kuda stremimo. Vezaćemo se za dve široke oblasti bežičnih sistema Weba koje dominiraju tržištem: protokol WAP i i-režim.

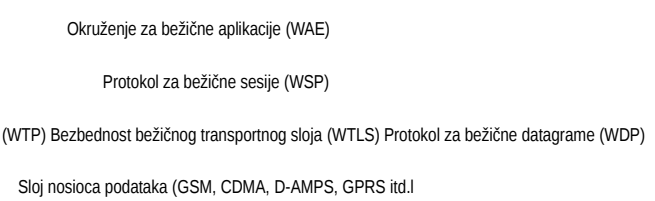
WAP - Protokol za bežične aplikacije

Kada su Internet i mobilni telefoni postali obična stvar, nije trebalo dugo da se neko doseti da ih smesti u zajednički uređaj: mobilni telefon sa ugrađenim ekranom za bežičan pristup e-pošti i Webu. Taj „neko“ je u ovom slučaju prvobitno bio kon- zorcijum firmi Nokia, Ericsson, Motorola i phone com (bivša Unwired Planet), dok sada broji na stotine članova. Sistem se zove protokol za bežične aplikacije (engl. *Wireless Application Protocol, WAP).*

WAP uređaj može da bude mobilni telefon novije generacije, lični digitalni pomoćnik (LDA) ili prenosivi računar bez ikakve mogućnosti komunikacije glasom. Specifikacija obuhvata sve te uređaje, a i mnoge druge. Osnovna zamisao je bila da se iskoristi postojeća digitalna bežična infrastruktura. Korisnici bukvalno mogu da

bežičnim putem pozovu mrežni WAP prolaz i da preko njega pošalju zahtev za Web stranu. Mrežni prolaz tada traži zahtevanu stranu u svom kesu. Ako je tamo nađe, odmah je šalje korisniku; ako je ne nađe, preuzima je sa ožičenog Interneta. To, u suštini, znači daje WAP 1.0 sistem s komutiranim kolima i prilično visokom cenom po minutu veze. Da bismo skratili priču, recimo samo to da korisnici nisu baš oduševljeno pristupali Internetu na maleckom ekranu, plaćajući pri tome svaki minut na vezi, tako da se WAP 1.0 ne može smatrati uspehom (mada je tu bilo i drugih problema). Međutim, WAP i njegov konkurent i-režim (o kome ćemo govoriti malo kasnije) izgleda da streme sličnoj tehnologiji, tako da će WAP 2.0 možda ipak postići veliki uspeh. Pošto je WAP 1.0 bio prvi korak ka bežičnom Internetu, vredelo bi ga ukratko opisati.

WAP je, u osnovi, skup protokola za pristupanje Webu, optimizovan za veze malog propusnog opsega i bežične uređaje sa sporim procesorima, skromnom memorijom i malim ekranom. Takav scenario se prilično razlikuje od mogućnosti standardnih stonih PC računara, pa je, prirodno, i skup protokola prilično različit. Slojevi sistema WAP prikazani su na slici 7-48.

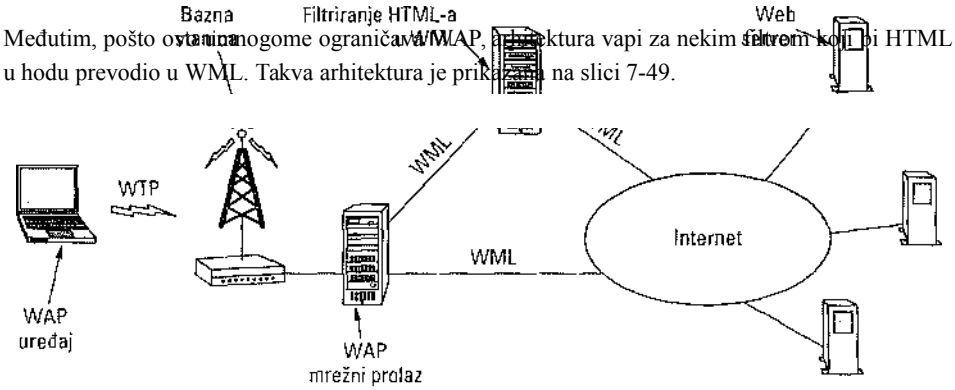


Slika 7-48. Skup WAP protokola.

Najniži sloj podržava sve postojeće sisteme mobilne telefonije, uključujući sisteme GSM, D-AMPS i CDMA. Brzina prenosa podataka protokolom WAP 1**.0** iznosi **9600** b/s. Iznad ovog sloja je **protokol za bežične datagrame** (engl. *Wireless Datagram Protocol, WDP),* koji je u suštini protokol UDP. Zatim dolazi sloj za bezbednost koji je očigledno neophodan na bežičnim vezama. WTLS je podskup Netscapeovog sistema SSL, o kome ćemo govoriti u 8. poglavlju. Sledi sloj za transakcije koji radi sa zahtevima i odgovorima, na pouzdan ili nepouzdan način. Taj sloj zamenjuje TCP koji se zbog neefikasnosti ne koristi na bežičnim vezama. Sledeći je sloj sesije koji liči na HTTP/l. 1 uz neka ograničenja i proširenja izvedena u cilju optimizacije. Na vrhu je milcročitač (WAE).

Pored cene, činilac koji nesumnjivo utiče na prihvaćenost WAP-a jeste i to što WAP ne koristi HTML. Umesto njega, sloj WAE koristi **jezik za označavanje pri bežičnom prenosu** (engl. *Wireless Markup Language, WML),* jednu primenu jezika XML. Zbog toga, WAP uređaj u principu može da prihvati samo strane koje su prevedene u WML.

Međutim, pošto ovaj mnogome ograničava WAP, arhitektura vapi za nekim filtrom koji bi HTML u hodu prevodio u WML. Takva arhitektura je prikazana na slici 7-49.



Slika 7-49. Arhitektura sistema WAP.

Ruku na srce, sa WAP-om se malo požurilo. Kada je jezik XML, prvi put isproban, znali su ga samo oni iz konzorcijuma W3C, pa su se u štampi pojavili naslovi tipa WAPNE KORISTI HTML. Tačniji naslov bi glasio: WAP VEĆ KORISTI NOVI STANDARD ZA HTML. Međutim, učinjenu štetu bilo je teško popraviti i WAP 1.0 nikada nije zaživeo. Vratićemo se ponovo na WAP pošto budemo opisali njegovog glavnog konkurenta.

I-režim

Dole je međugranski konzorcijum telefonskih i računarskih kompanija usavršavao otvoreni standard koristeći najnapredniju verziju jezika HTML, u Japanu su uporedo tekla slična istraživanja. Jedna Japanka, Mari Matsunaga, smislila je drugačiji pristup bežičnom Webu, tzv. **informacioni** ili **i-režim** (engl. *information* mode ili *i-mode)*. Kompaniju za bežične veze - jednog od naslednika čuvenog japanskog telefonskog monopola - ubedila je u ispravnost svog pristupa i februara 1999, NTT DoCoMo (bukvalno: Japanska telefonska i telegrafska kompanija, gde god da ste) lansirala je svoju novu uslugu u Japanu. Tokom 3 godine imala je preko 35 miliona japanskih pretplatnika koji su mogli da se povezuju s preko 40.000 Web lokacija specijalizova- nih za i-režim pristupanja. Taj uspeh nije nezapaženo promakao ispred očiju većine svetskih telefonskih kompanija, naročito zato što je WAP naizgled zapao u škripac. Pogledajmo šta je to i-režim i kako on radi.

I-režim se kao sistem sastoji iz tri glavne komponente: novog prenosnog sistema, novog korisničkog uređaja i novog jezika za pravljenje Web strana. Prenosni sistem se sastoji od dve zasebne mreže: postojeće mreže mobilne telefonije s komutiranim kolima (pomalo slične sistemu D-AMPS) i nove mreže s komutiranjem paketa napravljene posebno za uslugu i-režima. U govornom režimu koristi se mreža s komutiranim kolima i naplaćuje po minutu trajanja veze. U i-režimu koristi se mreža s komutiranjem

paketa koja je uvele aktivna (slično liniji ADSL ili kablu), tako da se ne naplaćuje vre- me na vezi, već svaki poslati paket. Trenutno nije moguće da se istovremeno koriste obe mreže.

Korisnički uređaj liči na mobilni telefon uz dodatak malog ekrana. Kompanija NTT DoCoMo ne reklamira ove uređaje kao bežične Web terminale, već kao bolje mobilne telefone, iako oni upravo služe za pristupanje Webu. U stvari, većina korisnika vero- vatno nije ni svesna daje na Internetu. Oni svoje uređaje koji rade u i-režimu smatraju mobilnim telefonima s dodatnim uslugama. U skladu s konceptom i-režima kao usluge, korisnik ne može da programira svoj uređaj, iako je on ravan PC računam iz 1995. i verovatno bi mogao da radi pod Windowsom 95 ili UNIX-om.

Kada se i-uređaj uključi, prikazuje se spisak kategorija zvanično odobrenih usluga. Ima više od 1000 usluga svrstanih u oko 20 kategorija. Svaku uslugu, koja realno predstavlja malu i-Web lokaciju, pruža neka nezavisna kompanija. Glavne kategorije u zvaničnom meniju su: e-pošta, diskusione grupe, vremenska prognoza, sport, igrice, šoping, mape, horoskopi, zabava, putovanja, turistički vodiči, razni pozivni signali, recepti, kocka, elektronske bankarske usluge i berzanslci izveštaji. Usluga je pomalo usmerena na tinejdžere i dvadesetogodišnjalce koji obožavaju elektronske spravice, naročito ako su šarene. Činjenica da preko 40 kompanija prodaje samo pozivne signale govori sama za sebe. Najpopularnija aplikacija je e-pošta s porukama do 500 bajtova, što je veliko poboljšanje u odnosu na uslugu prenosa kratkih poruka (engl. *ShortMes- sage Service, SMS)* od 160 bajtova. Popularne su i računarske igrice.

Postoji i preko 40.000 Web lokacija koje rade u i-režimu, ali se njima ne pristupa iz menija, već se mora upisati njihova URL adresa. Zvanična lista u izvesnom smislu podseća na Internet portal iz koga se Web lokacijama može pristupiti direktno (bez upisivanja URL adrese).

NTT DoCoMo strogo kontroliše zvanične usluge. Da bi bila stavljena na listu, usluga mora da zadovolji niz unapred objavljenih kriterijuma. Na primer, usluga ne srne da ispoljava loš uticaj na društvenu zajednicu, japansko-engleski rečnici moraju imati dovoljno reči, usluge koje nude pozivne signale moraju često svoj asortiman dopunjavati novim zvučnim sekvencama i nijedna lokacija ne sme da podržava trenutno pomodarstvo, niti bilo šta drugo što bi se loše odrazilo na NTT DoCoMo (Frengle, 2002). Za onih dragih 40.000 Internet lokacija koje rade u i-režimu ne postoje nikakva ograničenja u pogledu ponašanja.

Poslovni model i-režima toliko se razlikuje od konvencionalnog Interneta da ga vredi objasniti. Osnovna pretplata za korišćenje i-režima iznosi nekoliko dolara me- sečno. Pošto se naplaćuje i svaki primljen paket, osnovna pretplata obuhvata i određen mali broj primljenih paketa. Korisnik alternativno može da izabere pretplatu s više besplatnih paketa, a preko toga sa cenom po paketu koja drastično pada s povećanjem broja primljenih paketa. Ako se besplatni paketi potroše pre kraja meseca, dodatni paketi se mogu dokupiti preko mreže.

Da biste koristili uslugu, morate se na nju pretplatiti, a to se radi tako što je jednostavno pritisnete i otkucate svoj PIN kod. Većina zvaničnih usluga košta 1-2 dolara mesečno. NTT DoCoMo uslugu naplaćuje preko telefonskog računa, prosleđujući 91% prihoda davaocu usluge i zadržavajući za sebe 9%. Ako postoji milion korisnika neke nezvanične usluge, njen davalac mora svakog meseca da šalje milion računa (za oko 1 dolar) korisnicima. Ako ta usluga postane zvanična, naplatu vrši NTT DoCoMo i davaocu usluge samo svakog meseca prosleđuje 910.000 dolara. Takva mogućnost da se izbegne teret naplaćivanja veliki je podsticaj kompanijama da postanu zvanični davaoci usluga, a kompaniji NTT DoCoMo samo

otvara mogućnost dodatnog priho- dovanja. Kada ste zvaničan davalac usluge, pojavićete se i u meniju, što znači da će vam korisnici lalcše pristupati. Korisnikov telefonski račun obuhvata telefonske pozive, pretplatu za i-režim, naknade za usluge i za dodatne pakete.

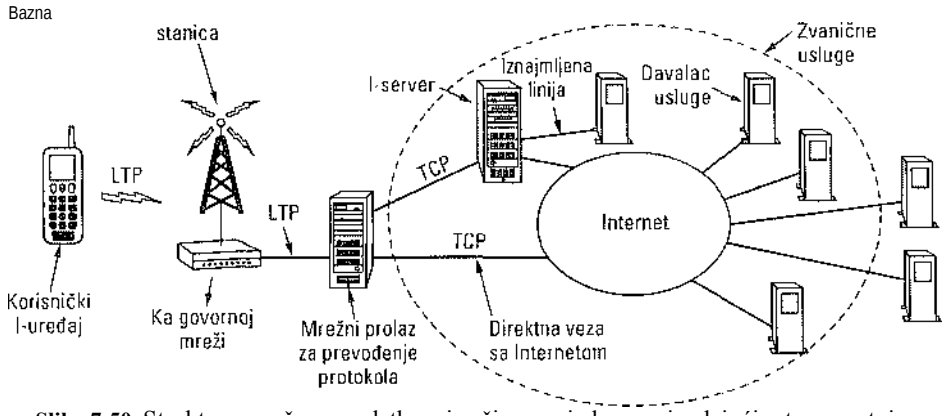
Uprkos velikom uspehu u Japanu, ostaje nejasno da li će i-režim zaživeti i u SAD i Evropi. Japanski način života unekoliko se razlikuje od načina života na Zapadu. Prvo, većina potencijalnih korisnika na Zapadu (npr. tinejdžeri, studenti, poslovni ljudi) već imaju kućne PC računare s velikim ekranom i gotovo sigurno su povezani na Internet, barem brzinom 56 kb/s. U Japanu je malo korisnika kućnih PC računara koji su povezani na Internet, nešto zbog nedostatka prostora, a više zbog izuzetno visokih cena lokalnog telefonskog saobraćaja (oko 700 dolara za instaliranje linije i 1,5 dolara po satu utrošenom na lokalne razgovore). Za većinu korisnika, i-režim je njihova jedina veza sa Internetom.

Drago, korisnici na Zapadu nisu navikli da plaćaju dolar mesečno da bi pristupili lokaciji CNN-a, dragi dolar da bi pristupili portalu Yahoo, a treći da bi koristili Google - da i ne pominjemo nekoliko dolara mesečno po megabajtu paketa. Većina davalaca Internet usluga na Zapadu danas paušalno naplaćuje svoje usluge, bez obzira na stvaran korisnikov promet, što je uvedeno uglavnom na zahtev korisnika.

Treće, većina Japanaca koristi i-režim dok putuje vozom ili podzemnom železni- com između kuće i posla ili škole. U Evropi se manje ljudi prevozi vozom nego u Japanu, a u SAD skoro niko. Kada ste kod kuće, pored 17-inčnog monitora, megabitne ADSL linije i toliko besplatnih megabajta, praktično nema smisla koristiti i-režim. Pa ipak, ni popularnost mobilnih telefona niko nije mogao da predvidi, pa će i-režim možda pronaći svoje mesto i na Zapadu.

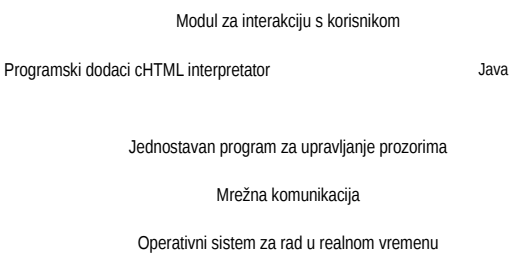
Kao što smo već objasnili, i-uređaj za govor koristi postojeću mrežu s komutiranim kolima, a za podatke novu mrežu s komutiranjem paketa. Mreža za prenos podataka zasniva se na sistemu CDMA i prenosi pakete veličine 128 bajtova brzinom 9600 b/s. Na bežičnoj vezi i-uređaji komuniciraju pomoću **jednostavnog transportnog protokola** (engl. *Lightweight Transport Protocol, LTP)* da bi se povezali s mrežnim prolazom za prevođenje protokola. Mrežni prolaz je širokopojasnim optičkim kablom povezan sa i-serverom, koji je sa svoje strane povezan sa svim uslugama. Kada korisnik iz zvaničnog menija izabere uslugu, njegov zahtev se šalje i-serveru koji kešira većinu strana da bi poboljšao performanse. Zahtevi upućeni lokacijama izvan zvaničnog menija mimoilaze i-server i odlaze direktno na Internet.

Današnji i-uređaji imaju procesore koji rade brzinom 100 MHz, više megabajta fleš ROM-a, možda megabajt RAM-a i mali ugrađeni ekran. Za i-režim potreban je ekran veličine barem 72 x 94 piksela, ali neki vrhunski i-uređaju imaju i 120 x 160 piksela. Ekrani su obično sa 8-bitnim bojama (256 boja). To nije dovoljno za fotografije, ali jeste za crteže i jednostavne crtaće. Pošto nema miša, po ekranu se krećete pomoću kursorskih tastera.



Slika 7-50. Struktura mreže za podatke **u** i-režimu, zajedno s pripadajućim transportnim protokolima.

Struktura softvera prikazana je na slici 7-51. Najniži sloj sadrži jednostavan operativni sistem za rad u realnom vremenu koji upravlja hardverom. Zatim sledi modul za mrežno komuniciranje koji koristi verziju LTP protokola u vlasništvu kompanije NTT DoCoMo. Iznad njega je jednostavan program za upravljanje prozorima koji radi s tekstom i jednostavnim slikama (GIF datotekama). Kada je ekran veličine 120 x 160 piksela, njime se lako upravlja.



Slika 7-51. Struktura softvera za i-režim.

Četvrti sloj sadrži interpretator Web strana (tj. čitač). U i-režimu se ne koristi potpuni jezik HTML, vec' negov podskup **cHTML (kompaktni HTML)** koji se ovlaš oslanja na HTML 1.0. Ovaj sloj dopušta i pomoćne aplikacije, odnosno programske dodatke, kao što čini i PC čitač. Jedna od standadnih pomodnih aplikacija je i interpretator jedne malo izmenjene verzije Javine virtuelne mašine (JVM).

Na vrhu je modul za interakciju, odnosno komuniciranje s korisnikom.

Razmotrimo sada detaljnije cHTML. Vec smo rekli da on podseca na HTML 1.0, bez nekoliko njegovih mogućnosti i s nekoliko novih osobina koje mu omogućuju da radi na pokretnom i-uređaju. On je podnet konzorcijumu W3C radi standardizovanja, ali je W3C pokazao malo zanimanja, pa izgleda da će to ostati proizvod u privatnom (korporacijskom) vlasništvu.

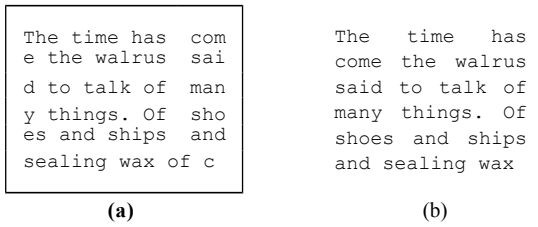
Dozvoljena je većina HTML oznaka: <html>, <head>, <title>, <body>, <hn>, , <ul>, <ol>, <menu>, <li>,
, <p>, <hr>, <img>, <form> i <input>. Oznake <b> i <i> nisu dozvoljene.

Za povezivanje s drugim stranama postoji oznaka <a>, ali uz dodatnu šemu (protokol) *tel* za biranje telefonskog broja. Sema *tel* u izvesnom smislu podseđa na šemu *mailto.* Kada se izabere veza s šemom *mailto,* čitač otvara obrazac za slanje e-poruke na adresu naznačenu u hipervezi. Kada se izabere hiperveza sa šemom *tel,* čitač bira odgovarajući telefonski broj. Na primer, u adresaru umesto imena mogu da budu sličice vaših poznanika. Kada izaberete jednu od njih, i-uređaj poziva odgovarajući telefonski broj. Telefonske URL adrese razmatraju se u RFC dokumentu 2806.

Čitač cHTML jezika ima svoja ograničenja. On ne podržava JavaScript, okvire, opise stilova, pozadinske boje i pozadinske slike. On ne podržava ni JPEG slike pošto im treba dugo da se raspakuju. Dozvoljeni su Java apleti, ali su (trenutno) ograničeni na veličinu od 10 KB zbog sporog prenosa bežičnom vezom.

Iako je NTT DoCoMo odbacila neke HTML oznake, dodala je i neke nove. Tekst sa oznakom <blink> naizmenično se „pali" i „gasi". Iako može izgledati nedosledno to što su odbacili <b> (jer Web lokacije ne treba da se bave izgledom strane), a uveli oznaku <blink> koja se isključivo odnosi na izgled, upravo su tako uradili. Još jedna nova oznaka je <marquee> koja omogućava da njen sadržaj teče ekranom kao na traci.

Nov je i atribut *align* oznake
. On je neophodan jer uz ekran od šest redova sa po 16 znakova u svakom redu postoji velika opasnost da se red prelomi usred reči, kao na slici 7-52(a). Atribut *align* umanjuje tu opasnost, pri čemu se dobija nešto kao na slici 7-52(b). Zanimljivo je da Japanci nemaju taj problem. Za tzv. *konji* ispis (kineskim ideogramima), ekran se deli u pravougaonu mrežu ćelija veličine 9x10 ili 12 x 12 piksela, u zavisnosti od podržanog fonta. Svaka ćelija sadrži samo jedan kanji znak, koji u evropskim jezicima predstavlja čitavu reč. Prelamanje reda između dve reči uvek je dozvoljeno u japanskom jeziku.



Slika 7-52. Originalni tekst iz „Alise u zemlji čuda" uklapa se u ekran od **6** redova sa po **16** slova.

Iako u japanskom pisanom jeziku postoje desetine hiljada kanji znakova, kompanija NTT DoCoMo izmislila je 166 potpuno novih znakova, zvanih **emoji,** koji su izuzetno dopadljivi i predstavljaju u suštini piktograme tipa smeška sa slike 7-6. Među njima su simboli za astrološke znake, pivo, pljeskavicu, zabavni park, rođendan, mobilni telefon, psa, mačku, Božić, ucveljeno srce, poljubac, raspoloženje, po- spanost i dopadljivost.

Nov atribut je i mogućnost da korisnici hipervezu biraju pomoću tastature, što je sigurno veoma značajno za računar koji nema miša. Primer korišćenja ovog atributa vidi se iz cHTML datoteke prikazane na slici 7-53.

```
<html>
<body>
<h1> Izaberi opciju </h1>
<a href="messages.chtml“ accesskey=”1"> Proveri govornu poštu </a> <br>
<a href="mail.chtml“ accesskey="2"> Proveri e-poštu </a> <br>
<a href="games.chtml“ accesskey=”3"> Pokreni igricu </a>
</body>
</html>
```

Slika 7-53. Primer cHTML datoteke.

Iako kod klijenta sve može da bude pomalo tesno, i-server predstavlja potpuno opremljen računar, sa svim mogućim pomagalima. On podržava CGI, Perl, PHP, JSP, ASP i sve drugo što inače podržavaju Web served.

Sažeto poređenje sistema WAP i i-režima, onako kako su realizovani u sistemima prve generacije, prikazano je na slici 7-54. Iako vam neke od razlika mogu izgledati neznatne, one su često veoma važne. Na primer, petnaestogodišnjaci ne mogu da dobiju kreditnu karticu, pa su zato veoma zainteresovani za sistem elektronske kupovine kod koga se kupljeni artikli naplaćuju putem telefonskog računa, što predstavlja značajnu razliku.

O i-režimu možete dodatno da se obavestite kod Frenglea (2002) i Vacca (2002).

Osobina	WAP	I-režim
Šta predstavlja	Skup protokola	Usluga
Uređaj	Ručni uređaj, LDP, prenosivi računar	Ručni uređaj
Pristup	Biranjem telefonskog broja	Neprekidan
Mreža na kojoj se izvršava	S komutiranim kolima	Dve: električna kola + paketi
Brzina prenosa podataka	9600 b/s	9600 b/s
Ekran	Monohromatski	U boji
Jezik za označavanje	WML (aplikacija XML-a)	cHTML
Jezik za skriptove	WMLScript	Ne postoji
Naknada korišćenja	Po minutu	Po paketu
Plaćanje kupovine	Kreditnom karticom	Uz telefonski račun
Piktogrami	Ne	Da
Standardizacija	Otvoren standard foruma WAP	Vlasništvo NTT DoCoMo
Gde se koristi	U Evropi i Japanu	U Japanu
Tipičan korisnik	Poslovni čovek	Mlada žena

Slika 7-54. Poređenje prve generacije sistema WAP i i-režima.

Bežični Web druge generacije

Za sistem WAP 1.0, zasnovan na priznatim međunarodnim standardima, verovalo se da treba da postane ozbiljna alatka poslovnih ljudi koji su stalno u pokretu. Bio je to promašaj. S druge strane, i-režim, razvijen na vlasničkim resursima kao elektronska igračka za japanske

tinejdžere, postigao je veliki uspeh. Šta nas, dakle, čeka u budućnosti? Svaka strana je ponešto naučila od bežičnog Weba prve generacije. Konzorcijum sistema WAP naučio je da je sadržina važna. Ako nemate dovoljno Web lokacija koje govore vašim jezikom za označavanje, propast je neminovna. Kompanija NTT DoMoCo naučila je da zatvoren, vlasnički sistem, kruto povezan sa i-ureña- jima skromnih mogućnosti i japanskom kulturom nije dobar izvozni proizvod. Obe strane su došle do istog zaključka: da biste mogli da ubedite mnoge Web lokacije da svoj sadržaj prevedu u vaš format, morate imati otvoren, stabilan i univerzalno prihvaćen ježile za označavanje. Sukobljavanje formata nije dobro za posao.

Obe strane upravo uvode tehnologiju za drugu generaciju bežičnog Weba. WAP 2.0 je istupio prvi, pa ćemo za primer uzeti njega. U protokolu WAP 1.0 neke stvari su urađene kako valja, pa je njihov razvoj nastavljen. Treba reći da se WAP može izvršavati na veoma različitim mrežama. U prvoj generaciji korišcene su mreže s komutiranim kolima, ali su mreže s komutiranjem paketa bile alternativa i onda i sada. Sistemi druge generacije verovatno će koristiti komutiranje paketa, na primer, sistem GPRS. Takođe, WAP je od početka projektovan da podrži široku lepezu uređaja, od mobilnih telefona, do moćnih prenosivih računara, a ta težnja postoji i danas.

WAP 2.0 ima i neke nove osobine, od kojih su najznačajnije sledeće:

1. Model guranja (engl. *push),* kao i model povlačenja (engl.*pult)*.
2. Podrška za integrisanje telefonije u aplikacije.
3. Razmenjivanje multimedijskog sadržaja.
4. Uključivanje 264 piktograma.
5. Interfejs ka skladišnom uređaju.
6. Podrška za programske dodatke u čitaču.

Model povlačenja je dobro poznat: klijent zahteva stranu i dobija je. Prema modelu guranja, podaci stižu i kada se ne traže, na primer, stalno ažuriranje podataka o stanju na berzi ili u saobraćaju.

Govor i podaci počinju da se zbližavaju i WAP 2.0 to podržava na više načina. Već smo kod i-režima videli primer da se pozivanje telefonskog broja može povezati s biranjem odgovarajuće ikonice ili teksta na ekranu. Pored e-pošte i telefonije, podržana je i razmena multimedijskog sadržaja.

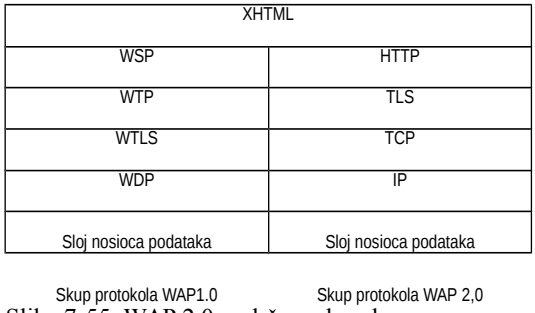
Ogromna popularnost koju su japanski emoji stekli kroz uslugu i-režima podstakla je konzorcijum WAP da izmisli 264 sopstvena emoji lika. Njihove kategorije obu- hvataju životinje, aparate, odeću, emocije, hranu, ljudsko telo, pol, mape, muziku, biljke, sportove, vreme, alatke, vozila, oružje i (meteorološko) vreme. Zanimljivo je da se u standardu svaki piktogram samo imenuje; ne daje se njegova bit mapa, vero- vatno iz straha da slike kojima bi bili predstavljeni, na primer, pojmovi „uspavan“ ili „zagrljaj“ ne uvrede pripadnike drugih kultura. Kod i-režima takav problem nije postojao jer je i-režirn od početka bio namenjen jednoj kulturi (zemlji).

Obezbeđivanje interfejsa ka skladišnom uređaju ne znači da svaki WAP 2.0 telefon treba da ima i veliki čvrsti disk. I fleš ROM je skladišni uređaj. Bežični fotoaparat koji radi uz protokol WAP može da privremeno skladišti slike u fleš ROM-u, pre nego što najuspelije pošalje na Internet.

Na kraju, programski dodaci treba da povećaju sposobnosti Web čitača. Obez- beđen je i ježile za pisanje skriptova.

Sam protokol WAP 2.0 takođe je tehnički unapređen. Dva najvec'a poboljšanja odnose se

na skup protokola i na jezik za označavanje. WAP 2.0 nastavlja da podržava stari skup protokola sa slike 7-48, ali podržava i standardni skup za Internet (TCP i HTTP/ 1.1). Međutim, uvedene su četiri manje (ali kompatibilne) izmene u protokol TCP (da bi mu se uprostio kod): (1) Korišćenje prozora fiksne veličine 64 KB, (2) odbacivanje sporog algoritma, (3) maksimalna jedinica prenosa (MTU) veličine 1500 bajtova i (4) neznatno izmenjen algoritam ponovnog prenosa. TLS je bezbednosni protokol transportnog sloja koji je standardizovala grupa IETF; njega ćemo razmotriti u 8. poglavlju. Mnogi od prvih uređaja verovatno će sadržati oba skupa, kao na slici 7-55.



Slika 7-55. WAP 2.0 podržava dva skupa protokola.

Druga razlika u odnosu na WAP 1.0 jeste jezik za označavanje. WAP 2.0 podržava XHTML Basic koji je namenjen malim bežičnim uređajima. Pošto se i kompanija NTT DoCoMo složila da podrži taj podskup, Web dizajneri mogu da koriste ovaj format znajući unapred da će se njihove strane videti i na fiksnom Internetu i na bežičnim uređajima. Ove odluke će konačno prekinuti rat formata jezika za označavanje koji je pretio da ugrozi razvoj bežičnog Weba.

Nekoliko reči o XHTML Basicu verovatno bi dobro došlo. On je namenjen mobilnim telefonima, televiziji, LDP uređajima, prodajnim automatima, pejdžerima, automobilima, automatima za igra, čak i ručnim satovima. Iz tog razloga, on ne podržava opise stilova, skriptove i okvire, ali ima većinu standardnih HTML oznaka. One su

grupisane uli modula, od kojih su neki obavezni, a neki nisu. Sve su definisane u XML-u. Moduli i primeri nekih oznaka prikazani su na slici 7-56. Ovde ih nećemo detaljno objašnjavati, već vas upućujemo na adresu www.w3.org.

Modul	Obavezan	Funkcija	Primeri oznaka
Struktura	Da	Struktura dokumenta	body, head, html, title
Tekst	Da	Informacije	br, code, dfn, em, hn, kbd, p, strong
Hipertekst	Da	Hiperveze	a
Liste	Da	Liste stavki	dl, dt, dd, ol, ul, li
Obrasci	Ne	Obrasci za ispunjavanje	form, input, label, option, textarea
Tabele	Ne	Pravougaone tabele	caption, table, td, th, tr
Slike	Ne	Slike	img
Objekti	Ne	Apleti, mape itd.	object, param
Metainformacije	Ne	Dopunske informacije	meta
Veze	Ne	Slično oznaci <a>	link
Osnovni	Ne	Početna URL adresa	base

Slika *7-56.* Moduli i oznake XHTML Basica.

Uprkos dogovoru da i jedan i drugi koriste isti jezik (XHTML Basic), sistemu WAP i i-režimu preti zajednički suparnik: mreža 802.11. Druga generacija bežičnog Weba treba da radi brzinom 384 kb/s, stoje mnogo više od 9600 b/s u prvoj generaciji, ali mnogo manje od 11 Mb/s ili 54 Mb/s, koliko nudi mreža 802.11. Naravno, mreža 802.11 nije prisutna svuda, ali sve više restorana, hotela, trgovina, kompanija, aerodroma, autobuskih stanica, muzeja, univerziteta, bolnica i drugih organizacija odlučuju da instaliraju bazne stanice za svoje zaposlene i za svoje mušterije, pa će možda gradska područja biti dovoljno pokrivena mrežom 802.11 da se ljudi neće libiti da odšetaju blok ili dva da bi seli u neki „umreženi“ kafić i uz kafu proverili svoju e-poštu. Firme rutinski mogu da postavljaju logotip mreže 802.11 pored logotipa kojim obaveštavaju korisnike koje kreditne kartice prihvataju - oba iz istog razloga: da privuku mušterije. Na planu grada (koji se, naravno, može preuzeti s mreže) mogu se zeleno označiti područja koja pokriva mreža, a crveno ona druga, da bi se korisnici mreže mogli kretati od jedne do dinge bazne stanice, kao nomadi koji kroz pustinju putuju od jedne oaze do druge.

Iako će restorani brze hrane možda lako prihvatiti da instaliraju bazne stanice mreže 802.11, poljoprivrednici verovatno neće, tako da će područja pokrivena mrežom biti sporadična i ograničena na centralna područja gradova zbog njenog ograničenog dometa (najviše nekoliko stotina metara). To može da bude podsticaj za dvorežimske bežične uređaje koji bi koristili mrežu 802.11 onda kada mogu da uhvate signal, a u suprotnom, sistem WAP.

7.4 MULTIMEDIJA

Bežični Web je uzbudljivo novo ostvarenje, ali nije jedino. Multimedija je za mnoge vrhunac rada u mreži. Na samo pominjanje multimedije rastu zazubice i raču- nardžijama i poslovnim ljudima. Oni prvi smatraju neodoljivim tehničkim izazovom da u svaki dom dovedu (interaktivni) video na zahtev. Oni drugi samo trljaju ruke brojeći zamišljenu zaradu. Postoje za prenos multimedije neophodan veliki propusni opseg, već je dovoljno teško

obezbediti njen rad pomoću fiksnih priključaka. Čak je i bežični video VHS kvaliteta još uvek nekoliko godina daleko od nas, tale da ćemo se ovde usredsrediti samo na ožičene sisteme.

Multimedija doslovno znači dva ili više medija. Daje izdavač ove knjige želeo da se prikloni opštoj histeriji za multimedijom, mogao je knjigu da reklamira kao multimedijsku. U krajnjoj liniji, ona sadrži dva medija: tekst i slike. Pa ipak, većina pod multimedijom podrazumeva kombinaciju dva ili više **medija za kontinualno repro- dukovanje** (engl. *continuous medici),* odnosno medija koji se reprodukciju tokom određenog vremena, obično uz saradnju korisnika. Ta dva medijuma su u praksi najčešće audio i video, tj. pokretne slike uz zvuk.

Međutim, mnogi multimediju često svode na čist audio, kao što je Internet telefo- nija ili Internet radio, što oni svakako nisu. U stvari, za njih je bolji izraz **mediji za reprodukovanje tokom preuzimanja** (tj. **u realnom vremenu)** (engl. *streaming medio),* ali ćemo se mi pridružiti opštem trendu i pod multimedijom podrazumevati i audio koji se reprodukuje u realnom vremenu. U narednim ođeljcima proučićemo kako računari obrađuju audio i video, kako se ta dva medija komprimuju, i neke mrežne aplikacije koje se njima bave. Iscrpnu raspravu (u tri toma) o multimediji na mreži naći ćete kod Steinmetza i Nahrstedta (2002; 2003a; 2003b).

7.4.1 Uvod u digitalni audio

Zvučni talas je longitudinalni poremećaj pritiska: čestice vazduha titraju u pravcu njegovog prostiranja. Kada takav talas dopre do uha, on izaziva treperenje bubne opne koje se prenosi na sitne kosti unutrašnjeg uha i dalje nervnim putevima do mozga. Takve nervne impulse čovek tumači kao zvuk. Slično tome, kada zvučni talas pogodi mikrofon, ovaj generiše električni signal koji odražava veličinu zvučnog pritiska u funkciji vremena. Predstavljanje, obrada, skladištenje i prenošenje takvih audio signala čini glavninu aktivnosti u proučavanju multimedijskih sistema.

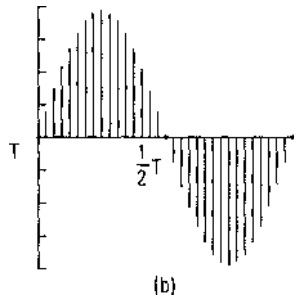
Frekvencije koje čuje ljudsko uho kreću se u opsegu od 20 Hz do 20.000 Hz. Neke životinje, naročito psi, mogu da čuju i više frekvencije. Uho čuje „logaritamski“, tako da se odnos subjektivne jačine dva zvuka čije su snage A i B dogovorno izražava **decibelima (đB),** prema formuli

$$dB = 10 \log_{10}(A/B)$$

Ako donju granicu čujnosti (zvučni pritisak od oko 0,0003 din/cm") za smusm talas frekvencije 1 lcHz označimo kao 0 dB, običan razgovor ima jačinu 50 dB, a granica bola iznosi oko 120 dB. Unutar ovog opsega uho može da razlikuje oko milion različitih vrednosti jačine zvuka.

Za razliku od oka, uho registruje i promene zvuka koje traju samo nekoliko mili- sekundi. Zbog toga se prilikom preuzimanja multimedijskog sadržaja neravnomer- nost pristizanja zvuka od nekoliko milisekundi primećuje kao pad njegovog kvaliteta, dok oko gotovo da ne može da primeti takvu neravnomernost u pristizanju slike.

Audio signali, potekli od zvučnih talasa, mogu se prevesti u digitalni oblik **analog- no-digitalnim pretvaračem** (engl. *Analog Digital Converter, ADC),* U pretvarač ulazi električni signal, a iz njega izlazi odgovarajući binarni broj. Na slici 7-57(a) vidimo primer sinusnog talasa. Da bismo taj signal predstavili digitalno, možemo ga uzorko- vati svakih *AT* sekundi, što prikazuju visine stubića na slici 7-57(b). Ako audio signal nije čist sinusni talas, već linearna kombinacija sinusnih talasa koji ne prelaze frekvenciju /, tada je prema Nikvistovoj teoremi (2. poglavlje) dovoljno uzorkovati učestalošću 2/. Češće uzorkovati nema smisla jer više frekvencije koje bi takvo uzor- kovanje moglo da uhvati ne postoje u audio signalu.



Slika 7-57. (a) Sinusni talas, (b) Uzorkovanje sinusnog talasa, (c) Kvantizacija uzoraka na 4 bita.

Digitalnim uzorkovanjem nikada se tačno ne predstavlja ulazni signal. Uzorci sa slike 7-57(c) obuhvataju samo 9 vrednosti u rasponu od -1,00 do 1,00, s korakom 0,25. Osmobitni uzorak bi u istom rasponu omogućio 256 vrednosti, a šesnaestobitni uzorak čak 65.536 vrednosti. Greška koja se uvodi time što uzorak sadrži konačan broj bitova, naziva se **šum kvantizacije** (engl. *quantization noise).* Alco je on prevelik, uho će ga zapaziti.

Opštepoznati primeri korišćenja uzorkovanog zvuka jesu telefon i audio CD. Tehnikom impulsno-kodne modulacije u telefonskom sistemu uzimaju se 8-bitni uzorci 8000 puta u sekundi. U Severnoj Americi i Japanu, sedam bitova su podaci, a osmi je upravljački bit; u Evropi su svih osam bitova podaci. Dve varijante ovog sistema omogućavaju brzinu prenosa 56.000 b/s, odnosno 64.000 b/s. Uz samo 8000 uzoraka u sekundi, gube se frekvencije iznad 4 kHz.

Za audio CD, zvuk se uzorkuje 44.100 puta u sekundi, čime se registruju frekvencije do 22.050 Hz, što ljudima sasvim odgovara (ali ne i psima). Uzorci su 16-bitni i linear no pokrivaju raspon amplituda. Imajte na umu da 16-bitni uzorci omogućavaju samo 6.5.536 različitih vrednosti, dok ljudsko uho može da razlikuje oko milion vrednosti,

mereno od granice čujnosti. Prema tome, sa samo 16 bitova po uzorku uvodi se izvestan šum kvantizacije (iako se ne pokriva pun dinamički opseg - audio CD ne treba da korisniku probije bubnu opnu). Uz 44.100 šesnaestobitnih uzoraka u sekundi, za prenos digitalnog audio signala u mono-režimu potreban je propusni opseg 705,6 kb/s, a u ste- reo-režimu 1,411 Mb/s. Iako je ovo manje od propusnog opsega potrebnog za prenos videa (pogledajte u nastavku), još uvek je za prenos u realnom vremenu nekomprimo- vanog stereo-zvuka kvaliteta za CD potreban gotovo ceo TI kanal.

Računarski softver lako može da obrađuje digitalizovan zvuk. Postoje desetine programa za lične računare koji korisnicima omogućavaju da snimaju, prikazuju, menjaju, miksuju i skladište audio signale iz različitih izvora. Skoro svi profesionalni sistemi za snimanje i obrađivanje zvuka danas su digitalni.

Muzika je, naravno, samo specijalan, ali važan deo audija. Drugi, isto tako važan specijalan slučaj jeste govor. Ljudski glas obično pokriva raspon od 600 do 6000 Hz. Govor čine samoglasnici i suglasnici koji imaju različita svojstva. Samoglasnici se proizvode neprigušeno, tako da njihova osnovna frekvencija odgovara rezonantnoj frekvenciji govornog aparata i položaju jezika i vilica govornika. Ti zvuci su gotovo periodični tokom tridesetak milisekundi. Suglasnici nastaju iz delimično prigušenog govornog aparata. Oni su manje periodični od samoglasnika.

Neki sistemi za generisanje i prenos govora zasnivaju se na modelima govornog aparata, pokušavajući da govor svedu na nekoliko parametara (npr. na veličinu i oblik različitih rezonantnih šupljina), umesto da samo uzorkuju zvučni talas. Kako rade ovi tzv. vokoderi ne spada u domen ove knjige.

7.4.2 Komprimovanje zvuka

Kao što smo upravo videli, za audio CD kvaliteta potreban je prenosni propusni opseg od 1,411 Mb/s. Odatle proizlazi da za efikasan prenos Internetom, audio treba znatno komprimovati. U tom cilju razvijeni su brojni algoritmi za komprimovanje zvuka. Možda je najpopularniji sistem MPEG, koji ima tri sloja (varijante), od kojih je najmoćniji i najpoznatiji **audio sloj** MPEG 3 (engl. *MPEG audio layer 3)* ili skraćeno - MP3. Na Internetu se nudi mnogo muzike u formatu MP3, što nije uvek u skladu sa zakonom, pa autori i nosioci autorskih prava stalno pokreću sudske sporove. MP3 pripada audio delu MPEG standarda za komprimovanje videa. Komprimovanje videa obradićemo u drugom delu poglavlja, a sada razmotrimo kako se komprimuje zvuk.

Zvuk se može komprimovati na jedan od dva načina. Kada se **kodira oblik talasa** (engl. *waveform coding*), signal se matematički - pomoću Furijeovih transformacija - razlaže na svoje frekventne komponente. Na slici 2-1(a) prikazana je jedna proizvoljna zavisnost signala od vremena i odgovarajuće amplitude njegovih Furijeovih komponenata. Amplituda svake komponente tada se kodira na najjednostavniji moguć način. Cilj je da se oblik talasa tačno reprodukuje na dragom Imaju pomoću što manje bitova.

Drugi način, **perceptivno kodiranje** (engl. *perceptual coding),* iskorišćava izve- sne nedostatke ljudskog slušnog aparata i signal kodira tako da čoveku zvuči isto kao i original, iako se ta dva signala na osciloskopu prilično razlikuju. Perceptivno kodiranje se zasniva na **psihoakustici** - naučnoj disciplini koja proučava način na koji ljudi primaju zvuk. MP3 se zasniva na perceptivnom kodiranju.

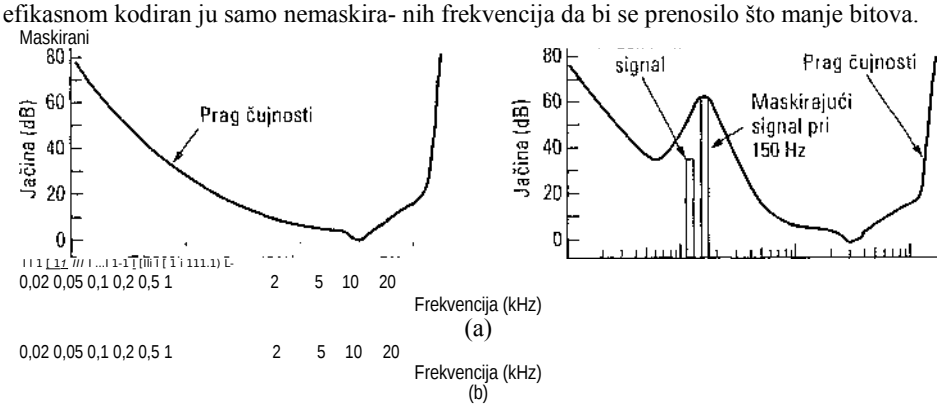
Kod perceptivnog kodiranja ključno je zapažanje da izvesni zvuci mogu da **maskiraju** drage zvuke. Zamislite da uživo prenosite koncert za flautu i orkestar usred vrelog letnjeg

dana. Odjedared, grupa radnika na ulici uključuje kompresor i počinje da razbija asfalt. Niko više ne čuje flautu. Njene zvuke maskira zvuk pneumatskog čekića. Za svrhe prenosa, dovoljno je kodirati samo frekventno područje u koje spada zvuk čekića jer slušaoci ionako ne čuju flautu. To se zove **maskiranje frekvencija** (engl. *frequency masking)* - sposobnost jačeg zvuka jedne frekvencije da priguši slabiji zvuk drage frekvencije koji bi se inače u odsustvu jačeg zvuka čuo. U stvari, i kada radnici završe posao, još neko vreme nećete moći da čujete flautu jer je uho smanjilo svoju osetljivost čim su radnici uključili kompresor i treba mu vremena da osetljivost ponovo povrati kada ga isključe. Taj efekat se zove **privremeno maskiranje** (engl. *temporal masking).*

Da bismo ušli u kvantitativni aspekt maskiranja, razmotrimo eksperiment 1. Osoba u tihoj sobi stavlja slušalice povezane sa zvučnom karticom računara. Računar proizvodi čist sinusni talas frekvencije 100 Hz malom snagom koja se postepeno povećava. Osoba treba da pritisne taster kada prvi put začuje zvuk. Računar tada beleži snagu zvuka i ponavlja eksperiment s frekvencijom 200 Hz, zatim sa 300 Hz i sve tako, do gornje granice koju čuje ljudsko uho. Kada se eksperiment ponovi s mnogo osoba, na log-log dijagramu sa slike 7-58(a) možete da vidite koliko prosečno treba da je jak ton da bi ga čulo ljudsko uho. Iz dobijene zavisnosti direktno sledi da nikada ne treba da kodirate frekvencije čija jačina leži ispod praga čujnosti. Na primer, ako jačina zvuka frekvencije 100 Hz iznosi 20 dB, taj zvuk se pri kodiranju može preskočiti bez primetnog gubitka kvaliteta, pošto 20 dB pri 100 Hz leži ispod granice čujnosti na slici 7-58(a).

Razmotrimo sada eksperiment 2. Računar ponovo sprovodi eksperiment 1, ali na testiranu frekvenciju ovoga puta superponira sinusni talas konstantne amplitude i frekvencije, recimo, 150 Hz. Otkrićemo da prag čujnosti u blizini frekvencije 150 Hz raste, kao na slici 7-58(b).

Iz ovog novog zapažanja sledi da broj prenesenih bitova možemo da smanjimo ako vodimo računa o signalima koji su maskirani jačim signalima u obližnjim frekventnim područjima i ne kodiramo ih. Na slici 7-58, signal frekvencije 125 Hz može se potpuno preskočiti i niko neće primetiti razliku. Čak i kada se u nekom frekventnom području jači signal prekine, i dalje možemo neko vreme da ne kodiramo frekvencije koje je maskirao, sve dok uho ne povrati osetljivost (efekat privremenog maskiranja). Suština algoritma MP3 sastoji se u Furijeovom razlaganju zvuka na frekventne komponente odgovarajućih jačina i efikasnom kodiran ju samo nemaskira- nih frekvencija da bi se prenosilo što manje bitova.



Slika 7-58. (a) Prag čujnosti u zavisnosti od frekvencije, (b) Efekat raaskiranja.

Kada sve ovo znamo, možemo da razumemo kako se obavlja kodiranje. Audio se

komprimuje tako što se signal uzorkuje učestalošću 32 kHz, 44,1 kHz ili 48 kHz. Mogu se uzorkovati jedan ili dva kanala, u bilo kojoj od četiri konfiguracije:

1. Monofonski (jedinstven ulazni tok).
2. Dvojno monofonski (npr. engleski i japanski zvučni zapis).
3. Razdvojeno stereofonski (svaki kanal se zasebno komprimuje).
4. Spojeno stereofonski (maksimalno se koristi redundancija između kanala).

Najpre se bira izlazna brzina zvučnog toka. MP3 može da komprimuje rokenrol na stereo-CD brzinom 96 kb/s a da niko ne primeti gubitak kvaliteta, čak ni rokenrol fa- novi koji (još) nisu ogluveli. Za klavirski koncert, međutim, potrebna je brzina barem 128 kb/s. Ova razlika postoji jer je kod rokenrola odnos signala i šuma mnogo veći nego kod klavirslcog koncerta (u tehničkom smislu, naravno). Mogu se izabrati i manje izlazne brzine uz izvestan pad kvaliteta.

Uzorci se tada obrađuju u grupama od po 1152 (obrada grupe uzoraka traje oko 26 ms). Svaka grupa se najpre propušta kroz 32 digitalna filtra da bi se dobila 32 frekventna područja. Ulazni signal se istovremeno uvodi u psihoakustički model da bi se utvrdilo koje su frekvencije maskirane. Zatim se svako od 32 frekventna područja dalje razlaže na finije spektralne komponente.

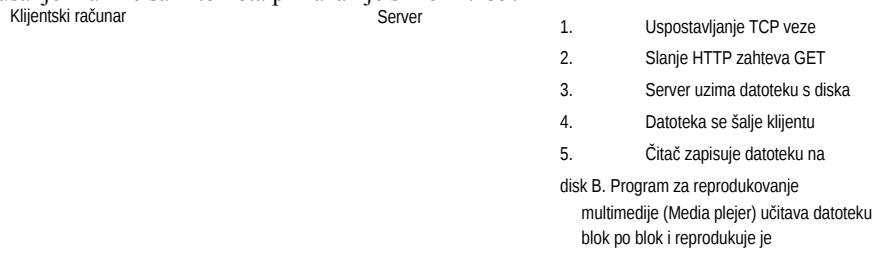
U sledećoj fazi, područjima se dodeljuju bitovi iz raspoloživog skupa, pri čemu se više bitova dodeljuje područjima s najmanje maskiranom spektralnom jačinom, manje bitova nemaskiranim područjima manje spektralne jačine, a maskiranim područjima se uopšte i ne dodeljuju. Bitovi se na kraju kodiraju Hafmanovim (Huffman) algoritmom koji pridružuje kratke oznake brojevima koji se pojavljuju često, a duge oznake onima koji se pojavljuju retko.

Priča se ovde ne završava. Koriste se i različite tehnike za smanjenje šuma, ublažavanje prelaza i moguće iskorišćavanje redundancije između kanala, ali one prevazila- ze temu ove knjige. Formalan matematički opis procesa možete da nađete kod Pana

7.4.3 Audio koji se reprodukuje u realnom vremenu

Pređimo sada sa tehnologije digitalnog zvuka na njegove tri mrežne primene. Prva je audio lcoji se reprodukuje u realnom vremenu (engl. *streaming audio)* ili, prostije, slušanje zvuka preko Interneta. To se još zove i muzika na zahtev. Naredne dve primene su Internet radio i prenos glasa preko Interneta (engl. *voice over IP).*

Internet je prepun muzičkih Web lokacija s listama pesama koje korisnik može da čuje ako pritisne odgovarajuću hipervezu. Neke od njih su besplatne (npr. lokacije novih muzičkih grupa koje tek stiču publicitet); druge zahtevaju naknadu za preuzimanje muzike, premda i one imaju izvestan broj besplatnih uzoraka (npr. prvih 15 sekundi pesme). Najdirektniji način za slušanje muzike sa Interneta prikazan je slikom 7-59.



Slika 7-59. Direktan način da se na Web stranu ugradi muzika za preuzimanje.

Proces započinje kada korisnik pritisne naslov odgovarajuće pesme. Tada na scenu stupa čitač Weba. U koraku 1 on uspostavlja TCP vezu sa Web serverom na kome se nalazi pesma. U koraku 2 čitač šalje serveru HTTP zahtev GET tražeći izabranu pe- smu. Zatim (koraci 3 i 4) server preuzima pesmu (datoteku u formatu MP3 ili nekom drugom formatu) sa diska i šalje je čitaču. Ako je datoteka veća od operativne memorije servera, on je može slati u blokovima.

Ispitujući MIME tip, na primer, *audio/mp3,* (ili nastavak imena datoteke), čitač utvrđuje kako da prikaže datoteku. Normalno će toj vrsti datoteke biti pridružena neka pomoćna aplikacija, kao što je RealOne Player, Windows Media Player ili Winamp. Pošto čitač sa pomoćnom aplikacijom obično sarađuje tako što sadržaj upisuje u privremenu datoteku, on će prvo u nju na disku upisati celu muzičku datoteku (korak 5). Zatim će pokrenuti program za reprodukovanje multimedije i proslediti mu ime privremene datoteke. U koraku 6, program počinje da učitava i reprodukuje muziku blok po blok.

Opisani pristup je u principu potpuno ispravan i pomoću njega se može slušati muzika. Ne

(1995).

valja samo to što se pre slušanja cela pesma mora preuzeti sa servera. Ako je muzička datoteka velika 4 MB (tipična veličina MP3 numere), a modem brzine 56 kb/s, korisnik će „uživati“ u desetominutnoj tišini pre nego što krene muzika. Neće se baš svako time oduševiti. Naročito zato što će i sledeća pesma započeti sa istim zakašnjenjem, a i ona posle nje.

Da bi problem resile ne zadiruci u način rada čitača, muzičke lokacije su prešle na drugi sistem. Datoteka na koju ukazuje naslov pesme nije datoteka koja je sadrži, već vrlo kratka **metadatoteka** koja sadrži samo naslov pesme. Tipična metadatoteka može da sadrži samo red ASCII teksta, na primer, ovakav:

```
rtsp://joes-audio-servei7song-0025.mp3
```

Kada čitač dobije takvu jednorednu datoteku, zapisuje je na disk u privremenu datoteku, pokreće pomoćnu aplikaciju (program za reprodukovanje) i predaje joj ime privremene datoteke, kao i obično. Program za reprodukovanje učitava datoteku i uviđa da ona sadrži URL adresu. Tada stupa u vezu sa *joes-audio-serverom* i od njega traži pesmu. Obratite pažnju na to da čitač više nije u igri.

Server naveden u metadatoteci najčešće nije prvobitni Web server. To obično nije čak ni HTTP server, već server specijalizovan za slanje multimedije. Na primer, takav server koristi **protokol za preuzimanje podataka u realnom vremenu** (engl. *Real Time Streaming Protocol, RTSP),* što se vidi i iz početka URL adrese (*rtsp:).* On je opisan u RFC dokumentu 2326.

Program za reprodukovanje ima četiri glavna zadatka:

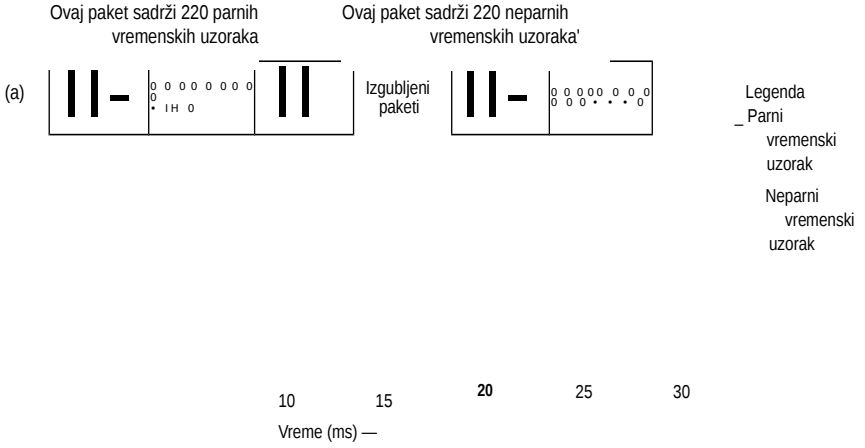
1. Da obezbedi korisničko okruženje.
2. Da obradi greške pri prenosu.
3. Da dekomprimuje muziku.
4. Da otkloni neravnomernost pri stizanju paketa.

Današnji programi za reprodukovanje multimedije većinom nude bogato korisničko okruženje koje ponekad simulira stereo uređaj, sa svim onim dugmićima, prekidačima, klizačima i vizuelnim efektima. Cesto se njegova spoljašnost, tzv. **maska** (engl. *skin),* može menjati. Program za reprodukovanje treba sve to da radi i da istovremeno sarađuje s korisnikom.

Njegov dragi zadatak je da ispravlja greške. Za prenošenje muzike u realnom vremenu retko se koristi protokol TCP jer nastanak greške i ponovno slanje podataka mogu da uvedu neprihvatljivo dugu pauzu pri njenom reprodukovanju. Za prenošenje muzike obično se koristi neki protokol sličan protokolu RTP, o kome smo govorili u 6. poglavlju. Slično dragim protokolima za prenos u realnom vremenu, i RTP radi preko protokola UDP, tako da se paketi možda gube. Program za reprodukovanje treba da se s tim izbori.

U nekim slučajevima, pri slanju se koristi preplitanje, tako da se greške lakše otklanjaju. Na primer, paket može da sadrži 220 stereo uzoraka, svaki s parom 16-bitnih brojeva, koji zajedno čine 5 ms muzike. Ali protokol može da šalje sve neparne uzorke tokom intervala od 10 ms u jednom paketu, a sve parne uzorke u drugom. Izgubljeni paket tada ne predstavlja pauzu od 5 ms u reprodukovanju, već gubitak svakog drugog uzorka tokom 10 ms. Program za reprodukovanje lako može da ublaži takav gubitak interpolirajući nedostajuće vrednosti na osnovu susednih uzoraka.

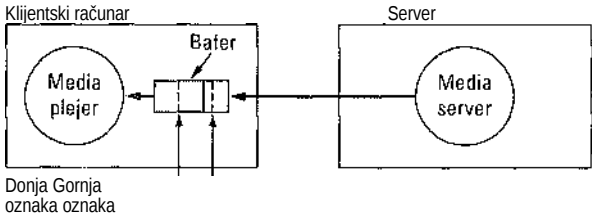
Korišćenje preplitanja za otklanjanje grešaka u prenosu prikazano je na slici 7-60. Ovde svaki paket sadrži naizmenične vremenske uzorke uzete tokom 10 ms. Shodno tome, gubitak paketa 3, kao što se vidi, ne izaziva pauzu u reprodukovanju, već samo tokom određenog vremena snižava rezoluciju. Nedostajuće vrednosti mogu se nadoknaditi interpoliranjem i tako opet obezbediti kontinualno reprodukovnje. Opisana šema radi samo s nekomprimovanim uzorkovanim zvukom, ali jasno pokazuje kako se pauza izazvana gubitkom paketa može mudrim kodiranjem prevesti samo u pad kva- liteta.U RFC dokumentu 3119 opisana je šema koja radi s komprimovanim zvukom.



Slika 7-60. Kada paketi nose naizmenične uzorke, gubitak paketa izaziva privremen pad kvaliteta (rezolucije), ali ne i pauzu pri reprodukovanju.

Treći zadatak programa za reprodukovanje jeste dekomprimovanje muzike. Iako je taj posao računarski zahtevan, prilično je jasan.

Četvrti zadatak je ublažavanje neravnomernosti stizanja paketa - neophodan uslov za svaki sistem koji radi u realnom vremenu. Svi audio sistemi za reprodukovanje u realnom vremenu privremeno skladište (baferuju) muziku tokom 10-15 sekundi pre nego što počnu daje reprodukuju (slika 7-61). Server u idealnom slučaju nastavlja da puni bafer podacima istom brzinom kojom ih iz bafera uzima program za reprodukovanje, ali pošto stvarnost nije idealna, dobro bi mu došle i povratne informacije.



Slika 7-61. Baferi programa za reprodukovanje pune se sa servera multimedije. Muzika se u stvari reprodukuje iz bafera, a ne direktno s mreže.

Za stalno održavanje punog bafera primenjuju se dva pristupa. Kod vučnog serve- ra (engl. *pull server),* sve dok u baferu ima mesta za još jedan blok, program za repro- dukovanje mu stalno šalje nove zahteve. Cilj mu je da bafer uvek bude pun do vrha.

Nedostatak vučnog servera je pojavljivanje brojnih nepotrebnih zahteva za podatke. Server zna daje poslao celu datoteku, pa zašto klijent i dalje šalje zahteve? Zbog toga se vučni serveri retko koriste.

Kod gurajučeg servera (engl. *push server),* program za reprodukovanje šalje ser- veru zahtev *PLAY* i ovaj samo gura podatke. Postoje dve mogućnosti: server multimedije radi normalnom brzinom reprodukovanja ili radi brže od toga. U oba slučaja, pre nego što počne reprodukovanje, privremeno se skladišti nešto podataka. Ako server radi normalnom brzinom reprodukovnja, podaci koji s njega stižu dodaju se na zadnji kraj bafera, a program za reprodukovanje ih uzima s prednjeg kraja (režim FIFO. Prim. prev.). Ako sve radi kako treba, u baferu se stalno održava ista količina podataka. Šema je zgodna jer ne zahteva razmenjivanje upravljačkih poruka.

Druga mogućnost je da server gura podatke brže nego što se mogu reprodukovati. Ona postoji zato da bi server koji ne može da garantuje redovnu normalnu brzinu slanja, mogao da ponovo uhvati korak ukoliko se nađe u zaostatku. Tu postoji potencijalan problem da će se bafer prepuniti ako server podatke gura brže nego što se troše (a on to mora da bi izbegao pauze u reprodukovanju).

Program za reprodukovanje nalazi rešenje u tome što u baferu definiše donju i gornju oznaku napunjenosti. Server, u osnovi, gura podatke sve dok ne ispuni bafer do gornje oznake. Tada mu program za reprodukovanje naređuje da stane. Pošto će podaci pristizati sve dok serveru ne stigne naređenje, rastojanje od gornje oznake i kraja bafera mora da bude veće od proizvoda propusnog opsega i kašnjenja mreže. Kada server prestane da šalje podatke, bafer će početi da se prazni. Kada njegov sadržaj dostigne donju oznaku, program za reprodukovanje naređuje serveru da nastavi sa slanjem podataka. I donja oznaka mora biti dovoljno udaljena od drugog kraja bafera, tako da se bafer nikada potpuno ne isprazni.

Da bi radio s gurajućim serverom, program za reprodukovanje mora da njime daljinski upravlja. To obezbeđuje protokol RTSP. On je definisan u RFC dokumentu 2326 i obezbeđuje mehanizam kojim program za reprodukovanje može da upravlja serverom. Njega se ne tiče tok samih podataka kojim obično upravlja protokol RTR Glavne komande protokola RTSP prikazane su na slici 7-62.

Komanda	Reakcija servera
DESCRIBE	Izlistava parametre multimedije
SETUP	Uspostavlja logički kanal između programa za reprodukovanje i servera
PLAY	Počinje da šalje podatke klijentu
RECORD	Počinje da prihvata podatke od klijenta
PAUSE	Privremeno prestaje da šalje podatke
TEARDOWN	Raskida logički kanal

Slika 7-62. RTSP komande koje program za reprodukovanje šalje serveru.

7.4.4, Internet radio

Kada je postalo moguće da se zvuk preko Interneta prenosi u realnom vremenu, komercijalne radio-stanice došle su na ideju da, osim bežičnim putem, svoj program emituju i preko Interneta. Nedugo zatim, i visokoškolske ustanove su počele da svoj signal emituju preko Interneta. Potom su *studenti* tih visokoškolskih ustanova pokrenuli sopstvene radio-stanice. Pomoću savremene tehnologije, praktično svako može da započne radio-prenos. Cela ideja Internet radija potpuno je nova i stalno se razvija, ali je vredi kratko razmotriti.

Za Internet radio postoje dva opšta pristupa. Prema prvom, programi se snimaju unapred i skladište na disku. Slušaoci mogu da se povezu Sa arhivom radio-stanice i da povuku i preuzmu iz nje svaki program za slušanje. To je u stvari isto što i audio prenos u realnom vremenu, o čemu smo malopre govorili. Svaki program koji se emi- tuje „uživo“ može se zatim sačuvati, tako da će ga arhiva imati, recimo, pola sata kasnije. Prednosti ovog pristupa su to što je jednostavan, što sve tehnike koje smo opisali uz njega rade i što slušaoci mogu da izaberu bilo koji program iz arhive.

Drugi pristup je emitovanje preko Interneta „uživo“. Neke stanice istovremeno emituju i u etar i preko Interneta, ali je sve više onih koje emituju samo preko Interneta. Pojedine tehnike koje se primenjuju za audio prenos u realnom vremenu pri- menljive su i za Internet radio, ali postoje i neke važne razlike.

Slično je to što i ovde postoji potreba za privremenim skladištenjem podataka kod korisnika da bi se ublažila neravnomernost njihovog stizanja. Ako unapred prikupite materijal za 10 do 15 sekundi reprodukovanja, to je dovoljno za glatko slušan je radija, čak i kada paketi vrlo neravnomemo pristižu. Sve dok paketi stižu pre nego što su potrebni, nije važno koliko kasne.

Jedna od glavnih razlika ogleda se u tome što se pri audio prenosu u realnom vremenu podaci mogu gurati brzinom većom od brzine reprodukovanja pošto ih primalac može zaustaviti kada u njegovom baferu dostignu gornju oznaku. To otvara mogućnost da se tada pošalju izgubljeni paketi, ali se ona retko koristi. Nasuprot tome, radio-prenos uživo uvek se odvija brzinom jednakom brzini reprodukovanja.

Druga razlika je to što radio-stanice koje emituju uživo obično istovremeno imaju stotine i hiljade slušalaca, dok se audio prenos u realnom vremenu odvija od tačke do tačke. IJ ovim okolnostima, za Internet radio treba koristiti višesmerno emitovanje pomoću protokola RTP/RTSP. To je nesumnjivo najefikasniji način rada.

U današnjoj praksi, međutim, ne ide sve tako glatko. Korisnik mora da uspostavi TCP

vezu sa stanicom preko koje se šalje sadržaj. Naravno, to izaziva razne probleme, kao što je pauziranje kada se napuni prozor, isključivanje tajmera za izgubljene pakete i njihovo ponovno slanje, itd.

Postoje tri razloga zašto se umesto višesmernog RTP emitovanja koristi jedno- smerno TCP emitovanje. Prvo, malo davalaca Internet usluga podržava višesmerno emitovanje, tako da to rešenje nije praktično. Drugo, protokol RTP manje je poznat od protokola TCP, a radio-stanice su obično male i bez dovoljno stručnjaka, pa zato češće koriste opštepoznat protokol koji podržava svaki softver. Treće, mnogi slušaju

Internet radio na poslu, što najčešće znači, iza zaštitne barijere. Administratori većinom podešavaju zaštitne barijere tako da bi svoje lokalne mreže zaštitili od nepoželjnih posetilaca. Oni obično dopuštaju TCP veze sa udaljenog priključka 25 (SMTP za e-poštu), UDP pakete sa udaljenog priključka 53 (DNS) i TCP veze sa udaljenog priključka 80 (HTTP za Web). Sve drago mogu da blokiraju, uključujući i RTP. Prema tome, alco Web lokacija želi da pošalje radio-signal kroz zaštitnu barijeru, može jedino da se predstavi kao HTTP server (barem zaštitnoj barijeri) i da koristi HTTP servere koji razumeju protokol TCP. Pomenute bezbednosne mere minimalno obez- beđuju lokalnu mrežu, ali prinuđuju multimedijske aplikacije na drastično neefika- sniji način rada.

Pošto je Internet radio nova komponenta multimedije, rat formata je na vrhuncu. RealAudio, Windows Media Audio i MP3 agresivno se nadmeću za mesto dominantnog formata za Internet radio. Nov takmac je Vorbis, tehnički sličan standardu MP3, ali otvorenog koda i dovoljno različit da ne koristi patente na kojima je zasnovan format MP3.

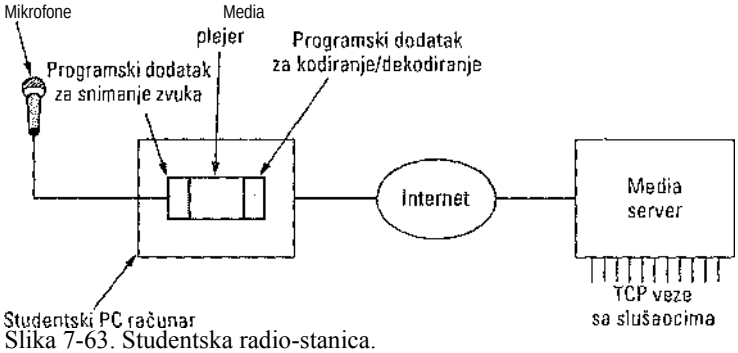
Tipična radio-stanica na Internetu ima Web stranu s vremenskim rasporedom emi- tovanja programa, informacijama o disk-džokejima i najavljivačima, i s mnogo oglasa. Tu su jedna ili više ikonica za biranje podržanih formata (ili ikonica LISTEN NOW, ako je podržan samo jedan format). Pomenute ikonice ukazuju na metadatote- ke, koje smo ranije objasnili.

Kada korisnik pritisne ikonicu, šalje se kratka metadatoteka. Pomoću njenog MIME tipa ili nastavka imena datoteke čitač određuje odgovarajuću pomoćnu aplikaciju (npr. program za reprodukovanje). Zatim zapisuje metadatoteku u privremenu datoteku na disku, pokreće program za reprodukovanje i predaje mu ime privremene datoteke. Program za reprodukovanje čita privremenu datoteku, zapaža u njoj URL adresu (koja obično ne počinje šemom *rtsp,* već šemom *http,* zbog problema sa zaštitnom barijerom, a i zato što neke popularne multimedijske aplikacije rade na taj način), stupa u vezu sa serverom i počinje da se ponaša kao radio. Recimo uzgred da audio ima samo jedan tok, pa zato *http* radi, ali je za video koji ima barem dva toka, neophodno nešto slično *rtsp*.

Ovde je zanimljivo to što svako, čak i neki nesvršeni student, može da pokrene i održava radio-stanicu na Internetu. Njene glavne komponente prikazane su na slici 7-63. Osnovu stanice čini običan PC računar opremljen zvučnom karticom i mikrofonom. Od softvera je potreban program za reprodukovanje multimedije, kao što su Winamp ili Freeamp, s programskim dodatkom za snimanje zvuka i koderom/deko- derom (engl. *codec*) za izabrani izlazni format, npr. za MP3 ili Vorbis.

Audio tok koji generiše stanica šalje se Internetom do velikog servera koji može da ga distribuira preko brojnih TCP veza. Server najčešće podržava veliki broj malih stanica. On održava katalog stanica za koje radi i prati šta svaka od njih trenutno emituje u etar.

Potencijalni slušaoci se povezuju sa serverom, biraju stanicu i od njega dobi- jaju podatke TCP vezom. Postoje komercijalni softverski paketi za upravljanje svim delovima procesa, kao i paketi otvorenog koda kao sto je icecast. Postoje i serveri voljni da obavljaju ovakvu distribuciju uz naknadu.



Slika 7-63. Studentska radio-stanica.

7.4.5 Govor preko Interneta

Nekada davno, javni komutirani telefonski sistem korišđen je prvenstveno za govorni saobraćaj, s tu i tamo nešto prenosa podataka. Ali je razmenjivanje podataka sve više raslo, sve dok se 1999. godine broj prenetih bitova podataka nije izjednačio s brojem prenetih govornih bitova (pošto se govor preko regionalnih linija modulira impulsno- -kodno, može se izmeriti broj prenetih bitova u sekundi). Godine 2002, saobraćaj podataka je za red veličine premašio govorni saobraćaj i nastavio da raste eksponencijalno, dok je govorni saobraćaj sve vreme imao konstantnu brzinu rasta (oko 5% godišnje).

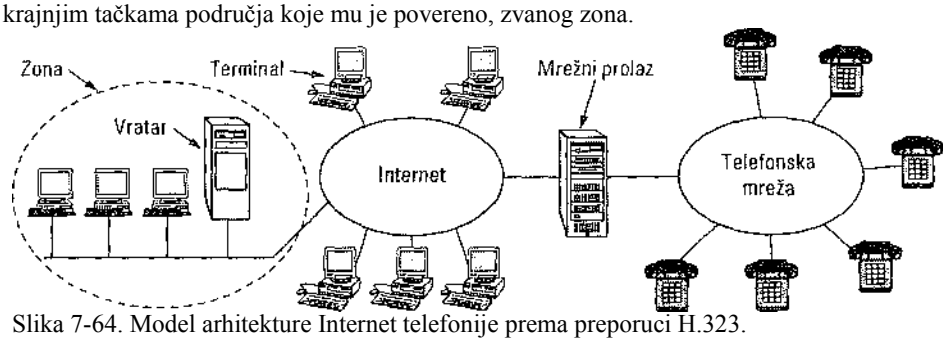
Zbog svega toga, mnogi operateri mreža koje rade s komutiranjem paketa najednom su počeli da se zanimaju za prenošenje govora njihovim mrežama. Tu je dodatni propusni opseg potreban za prenos govora minimalan, jer su mreže s komutiranjem paketa dimenzionisane za saobraćaj podataka. Međutim, telefonski račun prosečnog korisnika verovatno je veći od njegovog računa za Internet, pa su operateri mreža u Internet telefoni ji videli način da zgrnu velik novac a da ne moraju da razvuku ni jedno jedino novo optičko vlakno. Tako je rođena **Internet telefonija** (poznata i kao **govor preko Interneta,** engl. *voice over IP).*

H.323

Jedno je svima bilo jasno od samog početka: ako svaki proizvođač napravi sopst- veni skup protokola, sistem nikada neće proraditi. Zbog toga su se brojne zaintereso- vane strane okupile (pod pokroviteljstvom organizacije ITU) da zajednički izrade standarde. Godine 1996, organizacija ITU je izdala preporuku H.323 pod naslovom „Vizuelni telefonski sistemi i oprema za lokalne mreže koje obezbeđuju negaranto- van kvalitet usluge“. Samo je telefonska industrija mogla da smisli takvo ime. Preporuka je prerađena 1998. godine i tako prerađena postala osnova za prve široko prihvaćene sisteme Internet telefonije.

Preporuka H.323 predstavlja više pregled arhitekture Internet telefonije, nego određen protokol. Umesto da sama definiše protokole, ona se oslanja na niz konkretnih postojećih

protokola za kodiranje govora, uspostavljanje veze, signaliziranje, prenos podataka i drugo. Opšti model arhitekture Internet telefonije prema preporuci H.323 prikazan je na slici 7-64. U centru je mrežni prolaz (engl. *gateway)* koji povezuje Internet s telefonskom mrežom. On svojom stranom okrenutom Internetu razume protokole prema preporuci H.323, a stranom okrenutom telefonskoj mreži protokole PSTN. Uređaji za komuniciranje nazvani su **terminali.** Lokalna mreža može da ima **vratara** (engl. *gatekeeper)* koji vodi računa o krajnjim tačkama područja koje mu je povereno, zvanog zona.

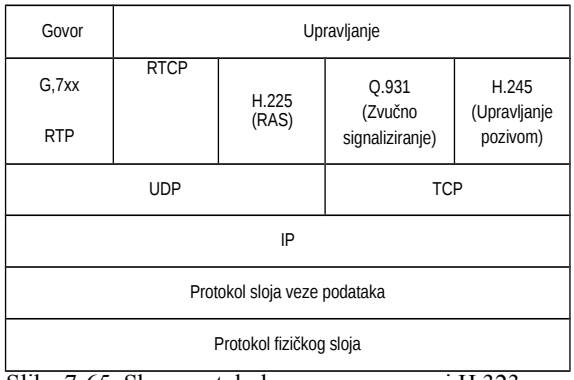


Slika 7-64. Model arhitekture Internet telefonije prema preporuci H.323.

Za telefonsku mrežu potrebno je više protokola, a najpre protokol za kodiranje i dekodiranje govora. Sistem PCM (impulsno-kodna modulacija), o kome smo govorili u 2. poglavlju, definisan je u ITU preporuci **G.711.** Pomoću njega se kodira prost govorni kanal tako što se uzorci od 8 bitova uzimaju 8000 puta u sekundi, čime se ge- neriše nekomprimovan tok podataka brzinom 64 kb/s. Svi H.323 sistemi moraju da podržavaju preporuku G.711. Dozvoljeni su i drugi protokoli za komprimovanje govora, ali nisu obavezni. Oni koriste drugačije algoritme za komprimovanje koji pronalaze različite kompromise između kvaliteta i propusnog opsega. Na primer, prema sistemu **G.723.1,** uzima se blok od 240 uzoraka (30 ms govora) i prediktivnim kodiranjem smanjuje na 24 ili 20 bajtova. Taj algoritam omogućava izlaznu brzinu podataka 6,4 kb/s ili 5,3 kb/s (kompresioni odnos 10, odnosno 12), uz neznatan gubitak subjektivnog kvaliteta. Dozvoljeni su i drugi sistemi za kodiranje/dekodiranje.

Pošto je dozvoljeno više algoritama za komprimovanje, potreban je protokol za do- govaranje terminala o algoritmu koji c'e koristiti. Taj protokol nosi ime **H.245.** Pomoću njega se usaglašavaju i drugi aspekti veze, npr. brzina prenosa. Protokol RTCP neophodan je za upravljanje RTPkanalima (za prenos u realnom vremenu). Takođe je neophodan i protokol za uspostavljanje i raskidanje veze, za obezbeđivanje zvučnih signala za biranje broja i pozivnog signala, kao i za sve drugo što spada u standardnu telefo- niju. Za to se koristi ITU protokol **Q.931.** Terminalima treba protokol pomoću koga će komunicirati s vratarem (ukoliko postoji). Za to se koristi protokol **H.225.** Kanal od PC računara do vratara, o kome brine taj protokol, zove se kanal za registrovanje, propuštanje i status (engl. *Registration/Ađmission/Status, RAS*). Taj kanal između ostalog omogućava terminalima da se pridruže zoni ili da je napuste, da zahtevaju propusni opseg i da ga vraćaju, kao i da ažuriraju svoj status. I na kraju, potreban je protokol kojim se prenose sami podaci. Za to se koristi protokol RTP. Njime, kao i obično, upravlja protokol RTCP. Međusobni odnosi svih ovih

protokola prikazani su na slici 7-65.



Slika 7-65. Skup protokola prema preporuci H.323.

Da biste razumeli kako ti protokoli međusobno sarađuju, razmotrite slučaj PC terminala u lokalnoj mreži (s vratarem) koji poziva udaljeni telefonski broj. PC računar mora prvo da pronađe vratara, pa difuzno preko priključka 1718 emitu je UDP paket za traženje vratara. Kada vratar odgovori, PC računar saznaje njegovu IP adresu. Sada se PC računar registruje kod vratara šaljući mu RAS poruku u UDP paketu. Pošto bude prihvaćen, PC računar šalje vrataru RAS poruku za propuštanje zahtevajući propusni opseg. Tek kada se dodeli propusni opseg, može da počne uspostavljanje veze. Kada računar unapred zahteva propusni opseg, vratar može da ograniči broj poziva da bi izbe- gao preopterećenje izlazne linije i tako pomogao da se ostvari potreban kvalitet usluge.

PC računar sada stupa u TCP vezu s vratarem da bi počeo da uspostavlja telefonski poziv. Telefonski poziv se uspostavlja postojećim telefonskim mrežnim protokolima koji rade s direktnom vezom, pa je zbog toga potrebna TCP veza. Nasuprot tome, telefonski sistem nema ništa slično protokolu RAS kojim telefoni najavljuju svoje postojanje, tako da su autori preporuke H.323 za RAS mogli da upotrebe UDP ili TCP. Opredelili su se za protokol UDP jer se njime razmenjuje manje paketa.

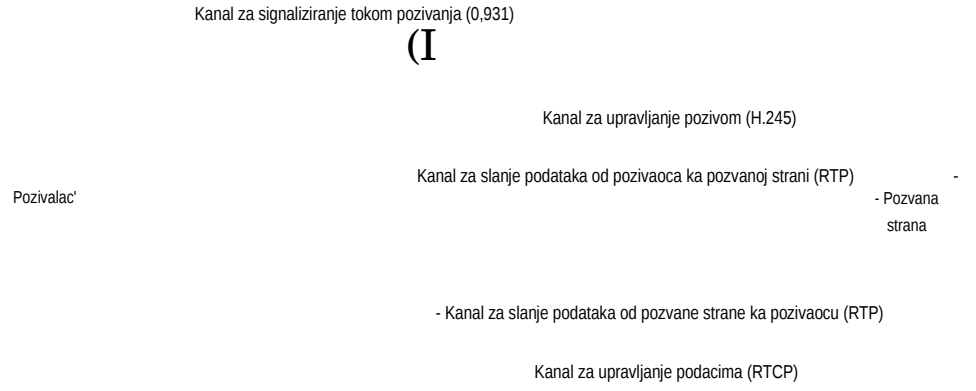
Sada, kada mu je dodeljen propusni opseg, PC računar može TCP vezom da pošalje Q.931 poruku *SETUP*. Njom se zadaje broj telefona koji se poziva (ili IP adresa i priključak, ako se poziva računar). Vratar odgovara Q.931 porukom *CALL PRO- CEEDING* da bi potvrdio ispravan prijem zahteva. Vratar tada prosleđuje poruku *SETUP* mrežnom prolazu.

Mrežni prolaz koji je pola računar, a pola telefonska centrala, upućuje je tada normalan telefonski poziv traženom telefonskom broju. Lokalna telefonska centrala za koju je priključen traženi telefon upućuje mu zvučni signal i istovremeno povratnim putem vraća Q.931 poruku *ALERT* kojom pozivaocu saopštava da je počela sa zvučnim

pozivanjem broja. Kada neko na traženom telefonskom broju podigne slušalicu, lokalna centrala vraća Q.931 poruku *CONNECT*kojom PC računaru signalizira daje uspostavila vezu.

Kada se veza uspostavi, vratar više nema šta da radi, ali mrežni prolaz, naravno, ima. Naredni paketi zaobilaze vratara i odlaze direktno na IP adresu mrežnog prolaza. U toj fazi imamo prostu cev koja povezuje dva kraja. To je samo veza u fizičkom sloju koja prenosi bitove i ništa više. Nijedna strana veze ne zna ništa o onoj drugoj.

Sada se pomoću protokola H.245 pregovara o parametrima ove konkretne telefonske veze. Za to se koristi upravljački H.245 kanal koji je uvek otvoren. Svaka strana započinje pregovore tako što objavljuje svoje mogućnosti, na primer, može li da prihvati video (H.323 može da radi s videom) ili konferencijske pozive, koje kodere/ dekodere podržava i slično. Kada svaka strana sazna šta može ona druga, uspostavljaju se dva jednosmerna kanala za podatke kojima se dođeljuju koderi/dekoderi i drugi dogovoreni parametri. Pošto dve strane mogu da imaju različitu opremu, sasvim je moguće da su koderi/dekoderi u dva kanala različiti. Pošto se s pregovorima završi, može se početi sa slanjem podataka pomoću protokola RTP. Njime upravlja protokol RTCP, naročito pri zagušenju. Ako se prenosi video, RTCP obavlja audio/video sinhronizovanje. Na slici 7-66 prikazani su različiti kanali. Kada bilo koja od dve strane prekine vezu, za njeno potpuno raskidanje koristi se Q.931 kanal za signaliziranje tokom poziva.



Slika 7-66. Logički kanali između pozivaoca i pozvane strane tokom telefonskog razgovora.

Kada se razgovor završi, pozivalac (PC računar) ponovo vrataru šalje RAS poruku da bi vratio dodeljeni propusni opseg. Umesto toga, može odmah da „okrene“ dragi telefon.

Ništa nismo rekli o kvalitetu usluge, iako je on suština uspešnosti Internet telefo- nije. Preporuka H.323 u tom pogledu ništa ne precizira. Ako u mreži između pozivaoca (PC računara) i mrežnog prolaza može da se uspostavi stabilan i ravnomeran tok podataka (npr. tehnikama o kojima smo govorili u 5. poglavlju), kvalitet će biti visok; u suprotnom, biće nizak i tu se ništa ne može. Telefonski deo veze koristi impulsno- -kodnu modulaciju (PCM) i kroz njega je tok podataka uvek ravnomeran.

SDP - Protokol za otvaranje sesije

Preporuku H.323 dala je organizacija ITU. Mnogi korisnici Interneta smatraju je tipičnim proizvodom „telefondžija": velikom, složenom i neelastičnom. Zbog toga je grupa IETF sazvala komitet za smišljanje jednostavnijeg i modularnijeg načina raz- govaranja preko Interneta. Njegov najznačajniji rezultat do danas jeste protokol za otvaranje sesije (engl. *Session Initiation Protocol, S1P),* definisan u RFC dokumentu 3261. Taj protokol opisuje uspostavljanje telefonskih poziva preko Interneta, video konferencije i druge multimedijske veze. Za razliku od sistema H.323, koji je potpun skup protokola, SIP predstavlja samo jedan modul, ali taj modul dobro sarađuje sa svim postojećim aplikacijama za Internet. On, na primer, telefonske brojeve definiše kao URL adrese, tako da ih mogu sadržati i Web strane, omogućujući da se pozove telefonski broj jednim pritiskom miša (na isti način kao što šema *mailto* omogućava da se pritiskom miša otvori program za slanje poruke e-pošte).

SIP može da uspostavi sesiju između dve strane (običnu telefonsku vezu), sesiju između više strana (od kojih svaka može da sluša i da govori) i višesmernu sesiju (između jednog pošiljaoca i više primalaca). Sesije mogu da sadrže audio, video ili podatke - ovo poslednje je zgodno, na primer, za igrice u kojima u realnom vremenu učestvuje više igrača. SIP je zadužen samo za uspostavljanje sesije, upravljanje njome, i njeno prekidanje. Za prenos podataka koriste se drugi protokoli, kao RTP/RTCP. SIP je protokol sloja aplikacija i može da se izvršava bilo preko protokola UDP, bilo preko protokola TCP.

SIP podržava niz usluga, uključujući pronalaženje pozvane strane (korisnik ne mora biti uz svoj kućni računar) i određivanje njenih mogućnosti, kao i rutinske poslove uspostavljanja i prekidanja veze. U najjednostavnijem slučaju, koji ćemo prvo razmotriti, SIP uspostavlja sesiju između pozivaočevog računara i računar a pozvane strane.

Telefonski brojevi u protokolu SIP prikazani su URL adresama uz šemu *sip,* na primer, *sip:*elza@cs.university.edu za korisnika Elzu na računam DNS imena *cs.uni- versity.edu.* URL adrese u protokolu SIP mogu sadržati i IPv4 adrese, IPv6 adrese ili stvarne telefonske brojeve.

Protokol SIP je tekstualan i modelovan prema protokolu HTTP. Jedna strana šalje tekstualnu (ASCII) poruku sa imenom metode u prvom redu i sledećim redovima sa zaglavljima za parametre koji se prosleđuju. Mnoga zaglavlja su preuzeta od sistema MIME da bi SIP mogao da sarađuje s postojećim aplikacijama za Internet. Šest metoda definisanih osnovnom specifikacijom prikazane su na slici 7-67.

Kada želi da otvori sesiju, pozivalac napravi TCP vezu s drugom stranom i preko nje pošalje poruku *INVITE* ili samo pošalje poruku *INV1TE* u UDP paketu. U oba slučaja, zaglavlja u drugom i sledećim redovima opisuju strukturu tela poruke koja sadrži pozivaočeve mogućnosti, tipove multimedije i formate. Ako pozvana strana prihvati poziv, ona odgovara kodom u stilu HTTP odgovora (trocifrenim brojem prema slici 7-42, pri čemu 200 znači pristanak). Posle reda s kodom odgovora, pozvana strana može da unese podatke o njenim mogućnostima, tipovima multimedije i formatima.

Metoda	Opis
INVITE	Zahtevanje početka sesije
ACK	Potvrđivanje da je sesija počela
BYE	Zahtevanje završetka sesije
OPTIONS	Upit računaru o njegovim mogućnostima
CANCEL	Poništavanje zahteva koji čeka
REGISTER	Obaveštenje serveru za preusmeravanje o korisnikovoj trenutnoj lokaciji

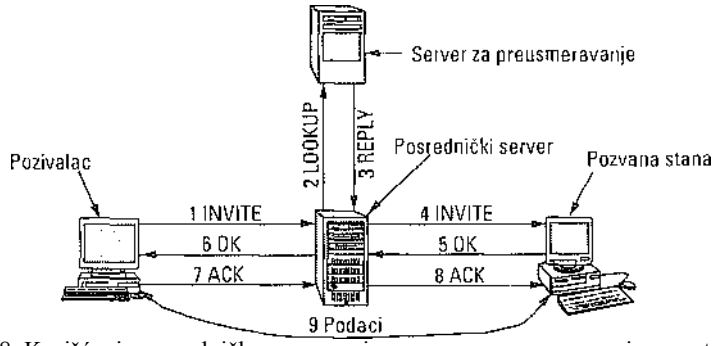
Slika 7-67. Metode protokola SIP definisane u osnovnoj sepecifikaciji.

Veza se uspostavlja trostepenim usaglašavanjem, tako da pozivalac završava protokol porukom ACK kojom potvrđuje primljenu poruku s kodom 200.

Bilo koja strana može da zahteva zatvaranje sesije šaljući poruku s metodom *BYE.* Kada druga strana potvrdi prijem poruke, sesija se zatvara.

Pomoću metode *OPTIONS,* računani se šalje upit o njegovim mogućnostima. Metoda se najčešće koristi pre otvaranja sesije da bi se utvrdilo da li je pozvana strana sposobna za razgovor preko Interneta ili za neku dragu vrstu sesije.

Metoda *REGISTER* je u vezi sa sposobnošću protokola SIP da pronađe korisnika koji nije kod kuće i poveže se s njim. Poruka se šalje SIP serveru za preusmeravanje koji vodi računa o tome gde se ko nalazi. Tom serveru se kasnije može poslati upit o korisnikovoj trenutnoj lokaciji. Operacija preusmeravanja prikazana je na slici 7-68. Ovde pozivalac šalje poruku *INVITE* posredničkom serveru da bi sakrio eventualno preusmeravanje. Posrednički server traži lokaciju korisnika i šalje poruku *INVITE* na pronađenu lokaciju. On zatim služi kao relej za naredne poruke u toku trostepenog usaglašavanja. Poruke *LOOKUP* i *REPLY* nisu deo protokola SIP; može se upotrebiti bilo koji pogodan protokol, u zavisnosti od toga kakav se server za preusmeravanje izabere.



Slika 7-68. Korišćenje posredničkog servera i servera za preusmeravanje u protokolu SIP.

Protokol SIP ima i brojne druge osobine koje na ovom mestu nećemo razmatrati, a među njima su stavljanje pozivaoca na čekanje (engl. *call waiting),* prosejavanje poziva (engl. *call screening),* šifrovanje (engl. *encryption),* i provera identiteta (engl. *authentication).* On ima i mogućnost pozivanja običnog telefona direktno s računara, ukoliko postoji odgovarajući

mrežni prolaz između Interneta i telefonskog sistema.

Poređenje sistema H.323 i protokola SEP

H. 323 i SIP imaju mnoge sličnosti, ali i neke razlike. Oba sistema omogućavaju razgovor između dve i vise strana, bez obzira na to da li su u pitanju računali ili obični telefoni. Oba podržavaju pregovaranje o parametrima, šifrovanje i protokole RTP/ RTCP. Njihove sličnosti i razlike sumirane su na slici 7-69.

Stavka	H.323	SIP
Autor	ITU	IETF
Kompatibilnost s javnom komutiranom telefonskom mrežom	Da	Prilična
Kompatibilnost sa Internetom	Ne	Da
Arhitektura	Monolitna	Modularna
Potpunost	Potpun skup protokola	SIP samo uspostavlja vezu
Pregovaranje o parametrima	Da	Da
Signaliziranje poziva	Q.931 preko TCP	SIP preko TCP ili UDP
Format poruke	Binaran	ASCII
Prenos multimedije	RTP/RTCP	RTP/RTCP
Povezivanje više strana	Da	Da
Multimedijske konferencije	Da	Ne
Adresiranje	Broj računara ili telefona	URL
Prekidanje poziva	Eksplicitno ili TCP raskidanje	Eksplicitno ili po isteku tajmera
Neposredno razmenjivanje poruka	Ne	Da
Šifrovanje	Da	Da
Obim standarda	1400 stranica	250 stranica
Ugradnja	Obimna i složena	Umereno složena
Status	Široko rasprostranjen	Tek dolazi

Slika 7-69. Poređenje sistema H.323 i protokola SIP.

Iako im je skup osobina sličan, dva protokola se razilaze po filozofiji. Sistem H.323 tipičan je detaljan, telefonski industrijski standard koji specificira potpun skup protokola i precizno definiše sve što je dozvoljeno, kao i ono što nije. Takav pristup vodi veoma dobro definisanim protokolima u svakom sloju, što olakšava međuoperativnost sistema. Cena za to je veliki, složen i krut standard koji će se teško prilagođavati budućim aplikacijama.

Za razliku od njega, SIP je tipičan protokol za Internet koji radi tako što razmenju- je kratice tekstualne poruke. On predstavlja jednostavan modul koji dobro sarađuje sa svim ostalim protokolima za Internet, ali lošije sa postojećim protokolima za signa- liziranje unutar telefonskog sistema. Pošto je IETF model Internet telefonije visoko- modularan, elastičan je i lako se može prilagoditi novim aplikacijama. Mogući nedostatak ogleda se u

manjoj međuoperativnosti, ali to predupređuju zainteresovane strane koje ga ugrađuju, tako što se često sastaju i proveravaju svoje sisteme.

Govor preko Interneta upravo kuca na naša vrata. Zbog toga već postoji niz knjiga koje su mu posvećene, a nekoliko takvih nalaze se i u spisku literature na kraju ove knjige (Collins, 2001; Davidson i Peters, 2000; Kumar i saradnici, 2001; Wright, 2001). Časopis *Internet Computing* u broju za maj/jun 2002. posvetio je više članaka toj temi.

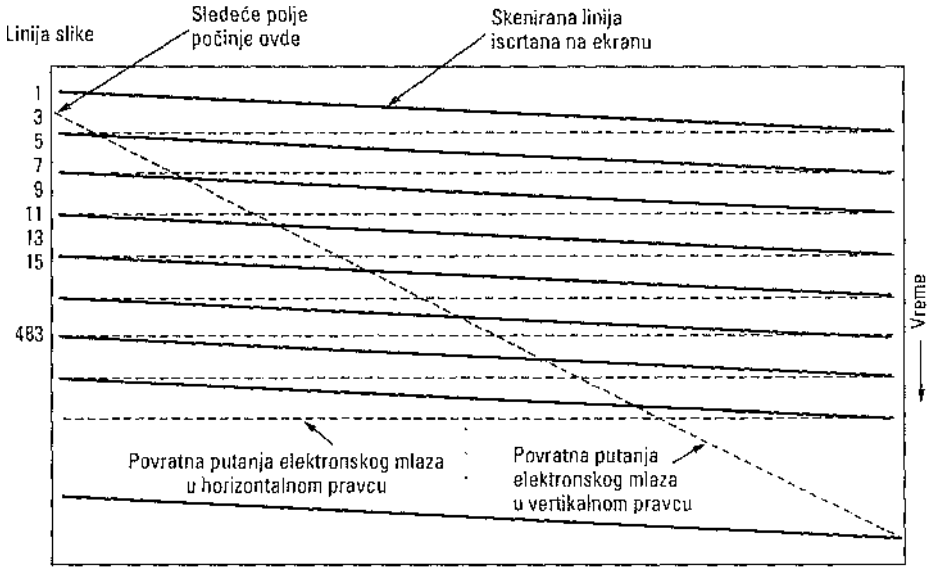
7.4.6 Uvod u video

Baš smo se raspričali o ušima; red je da pređemo na oči (ne plašite se, sledeci ode- Ijak nije o nosu). Ljudsko oko reaguje tako da slika koja se stvori na njegovoj mrežnjači traje još nekoliko milisekundi posle nestanka stimulusa. Ako se ispred očiju smenjuje redom 50 slika u sekundi, oko ih neće pojedinačno razlikovati. Svi video (televizijski) sistemi za pravljenje „pokretnih slika" zasnivaju se na ovom principu.

Analogni sistemi

Da biste razumeli video, treba poći od jednostavne, već skoro zaboravljene crno- -bele televizije. Da bi dvodimenzionalnu sliku ispred sebe predstavila jednodimenzionalnim električnim naponom u funkciji vremena, kamera pomoću elektronskog mlaza skenira njen odraz odozgo nadole, liniju po liniju, beležeći osvetljaj pojedinih njegovih područja. Kada skenira jednu celu sliku, tzv. **okvir** ili **kadar** (engl *,frame),* elektronski mlaz sve počinje od početka. Intenzitet osvetljenosti pojednih područja slike u funkciji vremena emituje se u mrežu i TV gledaoci u svojim domovima ponavljaju skeniranje da bi rekonstruisali sliku. Načini skeniranja koje koriste i TV kamera i TV prijemnik prikazani su na slici 7-70. (Pomenimo da CCD kamere ne skeniraju, već hvataju integralnu sliku, ali neke kamere i svi monitori koriste skeniranje).

Parametri skeniranja variraju od jedne do drage zemlje. Sistem koji se koristi u Se- vernoj i Južnoj Americi, i u Japanu razlaže sliku na 525 linija, ima odnos horizontalne i vertikalne dimenzije slika 4:3 i radi sa 30 okvira u sekundi. Evropski sistem ima 625 linija, istu razmeru slike 4:3 i 25 okvira u sekundi. Nekoliko prvih i poslednjih redova slike ne prikazuju se ni u jednom od dva sistema (da bi se dočarao pravougaoni oblik slike na prvobitnim okruglim katodnim cevima). Prikazuje se samo 483 od 525 redova sistema NTCS (i 576 od 625 redova sistema PAL/SECAM). Elektronski mlaz se isključuje prilikom povratka na početak okvira, tako da mnoge stanice (naročito u Evropi) koriste taj vremenski interval da bi emitovale teletelcst (tekstualne strane s ve- stima, vremenskom prognozom, sportskim rezultatima, stanjem na berzi itd.).



Slika 7-70. Način skeniranja prema nacionalnom NTSC standardu (SAD) za video i televiziju

Iako je za obmanu oka i postizanje glatkog efekta kretanja potrebno samo 25 slika u sekundi, pri toj brzini smenjivanja slika mnogi, naročito stariji gledaoci primećuju treperenje slike (jer je prethodna slika nestala sa mrežnjače pre nego što ju je smenila nova). Da se ne bi povećavala brzina smenjivanja slika, što bi dodatno opteretilo već tesan propusni opseg, nađeno je drugačije rešenje. Umesto da se skenirani redovi prikazuju redom, prvo se prikazuju svi neparni redovi, a zatim svi parni. Svaka polovina okvira naziva se **polje** (*engl. field*). Eksperimentima je dokazano da iako gledaoci za- pažaju treperenje pri 25 okvira u sekundi, oni ga ne zapažaju pri 50 polja u sekundi. Opisana tehnika zove se **preplitanje** (engl. *interlacing).* Neprepleteni televizija i video nazivaju se **progresivnim.** Imajte na umu da se filmovi prikazuju brzinom 24 slike u sekundi, ali daje svaki okvir potpuno vidljiv tokom 1/24 sekunde.

Za video u boji koristi se isti način skeniranja kao za monohromni (crno-beli), osim što sliku ne prikazuje jedan pokretni elektronski mlaz, već tri mlaza koji se zajedno kreću. Za svaku od tri osnovne aditivne boje: crvenu, zelenu i plavu (engl. *red, green, blue, RGB*) koristi se po jedan elektronski mlaz. Tom tehnikom se može predstaviti svaka nijansa - inešanjem crvene, zelene i plave boje odgovarajućih intenziteta. Međutim, za prenos jednim kanalom, signali sve tri boje moraju se kombinovati u jedinstven **kompozitni** signal.

Na početku televizije u boji, tehnički su bile izvodljive različite metode prikazivanja boja, pa su razne zemlje usvojile različite, međusobno nekompatibilne sisteme. (Ove razlike nemaju ništa s razlikama između sistema VHS, Betamax i P2000, koji predstavljaju metode snimanja videa.) Politički zahtev u svakoj zemlji bio je da se program koji se emituje u boji može gledati i na crno-belim televizorima. Zbog toga i nije bio prihvatljiv najjednostavniji sistem pojedinačnog kodiranja RGB signala. Sistem RGB takođe nije previše efikasan.

Prvi sistem u boji u SAD standardizovao je **Nacionalni komitet za televizijske standarde** (engl. *National Television Standards Committee, NTSC),* kome standard **NTSC** duguje ime. U Evropi je televizija u boji uvedena nekoliko godina kasnije, a u međuvremenu je tehnologija toliko uznapredovala, da je Evropa usvojila sisteme otpornije na šum i s boljim bojama. Ti sistemi se zovu **SECAM (SEquentiel Couler Avec Memoire),** koji se koristi u Francuskoj i u istočnoj Evropi, i **PAL (Phase Alternating Line),** koji se koristi u ostatku Evrope. Zbog razlika u kvalitetu boje kod sistema NTSC i sistema PAL/SECAM nastao je vic koji kaže da NTSC u stvari znači „nikad dvaput ista boja“ (engl. Never Twice the Same Color).

Da bi televizijski signal u boji mogao da se vidi na crno-belim televizorima, sva tri sistema linearno kombinuju RGB signale u signal **luminanse** (osvetljenosti) i dva signala **hrominanse** (obojenosti), iako svaki sistem koristi različite koeficijente pri sklapanju ovih signala iz RGB signala. Prilično neobično, oko je mnogo osetljivije na osvetljaj nego na boju, pa se signali hrominanse ne moraju tako precizno prenositi. Shodno tome, signal luminanse može se ernitovati na istoj frekvenciji kao i stari crno- -beli signal kako bi mogli da ga primaju crno-beli prijemnici. Dva signala hrominanse emituju se u dva uska područja viših frekvencija. Neki televizori imaju komande za osvetljenost, ton (nijansu) i zasićenje boje (ili za osvetljenost, dubinu i boju), pomoću kojih mogu posebno da podešavaju ova tri signala. Razumevanje luminanse i hrominanse neophodno je za razumevanje video komprimovanja.

Poslednjih nekoliko godina znatno je porastao interes za **televizijom visoke rezolucije** (engl. *High Definition Television, HDTV),* koja daje oštrije slike zahvaljujući udvostručenom broju skeniranih linija slike. Sjedinjene Države, Evropa i Japan razvili su različite i međusobno nekompatibilne HDTV sisteme. Zar se i moglo očekivati nešto drugo? Osnovni principi sistema HDTV u smislu skeniranja, luminanse, hrominanse itd, slični su postojećim sistemima. Međutim, sva tri formata imaju istu razmeru slike 16:9 (urnesto 4:3), da bi se bolje prilagodili formatu koji se koristi u filmskoj industriji (na 35-milimetarslcom filmu, razmera slika je 3:2).

Digitalni sistemi

Digitalni video se najjednostavnije može predstaviti sekvencom okvira koji sadrže pravougaonu mrežu elemenata slike, tj. **piksela** (engl. *pixels).* Piksel može biti jedno- bitan, kada predstavlja crno ili belo. Kvalitet takvog sistema sličan je onome što dobijete kada kolor fotografiju pošaljete faksom - jednom reci, nikakav. (Isprobajte to sami ili fotokopirajte kolor fotografiju na mašini koja ne pravi polutonove.)

Sledeći, viši stupanj je da svakom pikselu dodelite 8 bitova da biste prikazali 256 sivih nijansi (polutonova). Takvom šemom dobijate visokokvalitetan crno-beli video. Kod videa u boji, dobri sistemi koriste 8 bitova za svaku od RGB boja, iako ih gotovo svi naknadno mešaju u kompozitni signal koji emituju. Kada po pikselu potrošite 24 bita, možete da prikažete 16 miliona nijansi koje ljudsko oko ionako ne može da razlikuje.

Digitalne slike u boji prave se pomoću tri skenirajuća elektronska mlaza, po jednim za svaku boju. Geometrija je istakao kod analognog sistema sa slike 7-70, osim što umesto neprekidnih skeniranih linija ovde imamo nizove pojedinačnih piksela.

Da bi se postigao efekat glatkog kretanja, i kod digitalnog videa se mora smenji- vati

barem 25 okvira u sekundi. Međutim, pošto kvalitetni računarski monitori na svom ekranu ponovljeno iscrtavaju slike koje drže u memoriji (75 puta u sekundi i češće), preplitanje nije potrebno i normalno se i ne koristi. Ponovno iscrtavanje svakog okvira triput za redom dovoljno je da otkloni treperenje slike.

Drugim recima, utisak glatkog kretanja postiže se prikazivanjem određenog broja *različitih* slika u sekundi, dok na treperenje utiče učestalost iscrtavanja ekrana. Ta dva parametra ne znače isto. Nepokretna slika koja se prikazuje „brzinom 20 okvira u sekundi" neće se, naravno, pomerati, ali će treperiti jer jedna slika na mrežnjači oka nestaje pre nego što se u nju utisne druga. Film koji se prikazuje brzinom 20 različitih okvira u sekundi, pri čemu se svaki okvir iscrtava uzastopno četiri puta, neće treperiti, ali će pokreti u njemu biti iseckani.

Značaj ova dva parametra postaje jasniji kada razmotrimo propusni opseg koji je potreban za prenošenje digitalnog videa mrežom. Sadašnji računarski monitori uglavnom imaju razmeru prikaza 4:3 tako da se u njima koriste obične katodne cevi koje se proizvode za televizijske namene. Uobičajene konfiguracije su 1024 x 768, 1280 x 960 i 1600 x 1200. Pri 24 bita po pikselu i 25 okvira u sekundi, čak i najmanju od ovih konfiguracija treba napajati brzinom 472 Mb/s. Za to bi bio potreban SONET nosilac OC-12, a on se ne može uvoditi baš u svaku kuću. Udvostručenje navedene brzine prenosa da bi se otklonilo treperenje još manje je privlačno. Bolje rešenje je da se prenosi 25 okvira u sekundi, a da računar skladišti svaki okvir i iscrtava ga dva puta. TV difuzija ne koristi ovu strategiju pošto TV prijemnici nemaju memoriju. Čak i kada bi imali memoriju, analogne signale bi najpre trebalo pretvoriti u digitalne da bi se mogli uskladištiti u njoj, što zahteva dodatan hardver. Zbog svega toga, preplitanje je za TV difuziju neophodno, ali ne i za digitalni video.

7.4.7 Komprimovanje video zapisa

Do sada bi već trebalo da bude jasno daje prenošenje videa u nekomprimovanom obliku iluzija. Jedinu nadu pruža intenzivno komprimovanje. Srećom, poslednjih decenija je mnogo istraživačkih napora uloženo u tehnike i algoritme komprimovanja koji omogućuju prenos video signala. U ovom odeljku proučićemo način komprimovanja videa.

Svi sistemi za komprimovanje sadrže barem dva algoritma. Jedan za komprimovanje podataka na izvorištu i dragi za njihovo dekomprimovanje na odredištu. U literaturi se oni nazivaju algoritmima za kodiranje i dekodiranje videa. I mi ćemo ih ovde tako zvati.

Ovi algoritmi u radu pokazuju izvesnu asimetriju koju treba pravilno razumeti. Prvo, za mnoge svrhe - recimo, za multimedijske dokumente - video sekvencu treba kodirati samo jednom (pre skladištenja na serveru multimedije), ali je treba dekodirati

hiljadama puta (kad god sekvencu gleda korisnik). Takva asimetrija znači da algoritam za kodiranje može da bude spor i da se može izvršavati na skupom hardveru, pod uslo- vom daje algoritam za dekodiranje brz i malo zahtevan u pogledu hardvera. Na kraju krajeva, operater multimedijskog servera može da bude voljan da iznajmi superraču- nar koji će za nekoliko sedmica kodirati celu njegovu videoteku, ali ne bi bilo zgodno to zahtevati od korisnika. Mnogi realni sistemi komprimovanja usredsređuju se na to da dekodiranje bude brzo i jednostavno, čak i ako je kodiranje sporo i složeno.

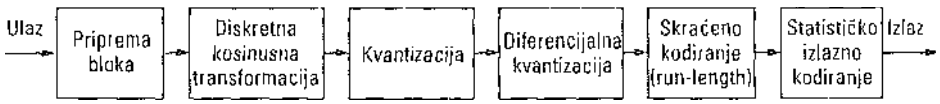
S druge strane, za multimediju u realnom vremenu, kao što su video konferencije, sporo kodiranje je neprihvatljivo. Kodiranje se mora obavljati u hodu, tj. stvarno u realnom vremenu. Zbog toga se za multimediju koja se prenosi u realnom vremenu koriste drugačiji algoritmi i parametri nego kada se ona skladišti na disk, često uz znatno manji stepen kompresije.

Druga asimetrija se ogleda u tome što dekodiranje ne mora stvarno da bude reverzibilno u odnosu na proces kodiranja. Kada se datoteka komprimuje, prenese do korisnika i tamo dekomprimuje, korisnik očekuje da se pred njim pojavi original - do poslednjeg bita. Za mutlimediju se takav zahtev ne postavlja. Obično je sasvim prihvatljivo da se delcodirani video signal malo razlikuje od originala. Kada dekodirani signal nije potpuno jednak originalnom signalu, taj sistem kodiranja radi s gubicima (engl. *lossy)*. Ako su ulazni i izlazni signal identični, sistem kodiranja radi bez gubitaka (engl. *lossless)*. Sistemi koji rade s gubicima važni su jer se prihvatanje malog gubitka informacija može isplatiti preko većeg stepena komprimovanja.

Standard JPEG

Video nije ništa drugo do sekvenca (ozvučenih) slika. Kada bismo imali dobar algoritam za kodiranje jedne slike, taj algoritam bi se mogao primeniti na sve slike redom, dakle, i za komprimovanje videa. Dobar algoritam za komprimovanje nepokretnih slika zaista postoji, pa počnimo od njega. Standard JPEG (engl. *Joint Photographic Experts Group)* za komprimovanje nepokretnih višetonskih slika (npr. fotografija) razvila je grupa stručnjaka za fotografiju pod pokroviteljstvom organizacija ITU, ISO i IEC (još jedna organizacija za standardizovanje). On je važan za multimediju, zato što standard MPEG grabo liči na standard JPEG jer zasebno kodira svaki okvir, premda ima i dodatne osobine (komprimovanje prostora između okvira, otkrivanje kretanja). JPEG je definisan kao Međunarodni standard 10918.

Sistem JPEG ima četiri režima i mnogo opcija. On više liči na spisak za kupovinu, nego na jedinstven algoritam. Za nas je, međutim, bitan samo sekvencijalni režim za komprimovanje s gubicima, a on je prikazan na slici 7-71. Osim toga, usredsredičemo se na uobičajeno JPEG komprimovanje 24-bitnih RGB video slika, preskačući manje važne detalje.

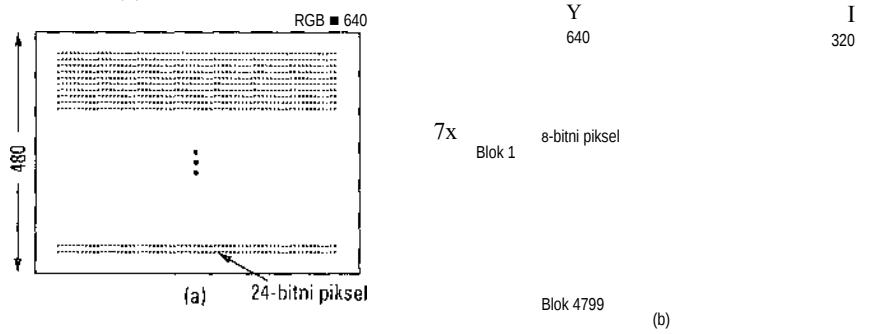


Prvi korak kodiranja slike sistemom JPEG jeste priprema bloka podataka. Da bismo bili određeniji, pretpostavimo da je ulazni signal RGB slika veličine 640 x 480 piksela sa 24 bita po pikselu, kao na slici 7-72(a). Pošto se korišćenjem luminanse i hrominanse dobija bolja kompresija, prvo ćemo izračunati luminansu *Y,* a zatim dve hrominanse / i *Q* (za sistem NTSC), prema sledećim formulama:

Y = 0,30R + 0,59G + 0,11B
I = 0,60R - 0,28G - 0,32B
Q = 0.21R- 0,52G + 0,31B

Za sistem PAL hrominanse su U i V, i koeficijenti su drugačiji, ali je način izračunavanja isti. Kod sistema SECAM izračunavanje je sasvim drugačije.

Za T / i <2 konstruišu se zasebne matrice sa elementima između 0 i 255. Zatim se u matricama I i Q izračunavaju prosečne vrednosti kvadratnih bolokova od po četiri piksela, da bi se matrice smanjile na 320 x 240. Smanjenjem matrica gube se podaci, ali to oko jedva zapaža, posto je osetljivije na osvetljaj nego na boju. Pa ipak, podaci se tako komprimuju na polovinu prvobitne količine. Sada se od svakog elementa sve tri matrice oduzima 128 da bi se njihove vrednosti premestile u oblast oko nule. Svaka matrica se na kraju deli na blokove veličine 8x8. Matrica *Y*ima 4800 blokova; ostale dve imaju po 1200 blokova, kao sto se vidi na slici 7-72(b).



Slika 7-72. (a) Ulazni RGB podaci, (b) Podaci posle svrstavanja u blokove.

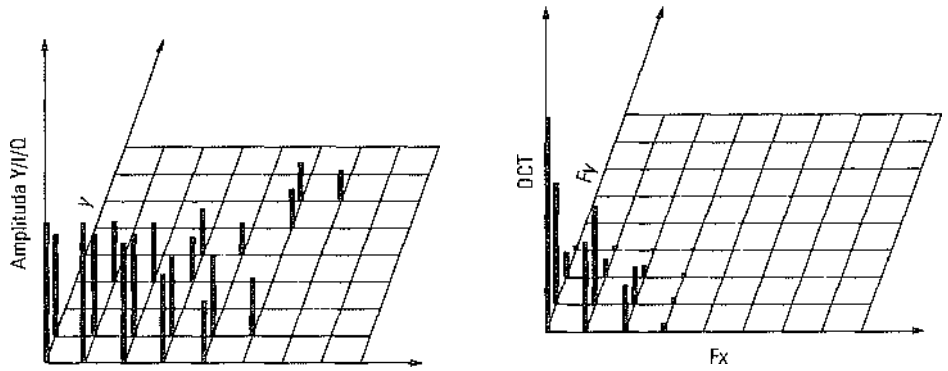
Drugi korak je primena **diskretne kosinusne transformacije** (engl. *Discrete Cosine Transformation, DCT)* pojedinačno na svaki od 7200 blokova. Rezultat svake transformacije je matrica DCT koeficijenata veličine 8x8. DCT element (0, 0) predstavlja srednju vrednost za blok. Drugi elementi govore koliku spektralnu jačinu ima svaka prostorna frekvencija. DCT teorijski radi bez gubitaka, ali upotreba brojeva u formatu s pokretnim zarezom i

Slika **7-71. Sekvencijalno JPEG komprimovanje s gubicima.**

transceđentnih funkcija uvek unosi grešku zaokruživanja koja rezultuje malim gubitkom informacija. DCT elementi normalno brzo opadaju sa udaljenošću od početka (0, 0), kao na slici 7-73.

Kada se faza DCT završi, prelazi se na treći korale - **kvantizaciju** (engl. *quantization),* u kome se manje značajni DCT koeficijenti brišu. Ovo transformisanje (koje radi s gubicima) izvodi se tako što se svaki koeficijent DCT matrice, veličine 8x8,

deli „težinom“ uzetom iz tabele. Kada su sve težine 1, transformisanjem se ne menja ništa. Međutim, ako težine naglo rastu sa udaljenjem od početka, više prostorne frekvencije brzo nestaju.



Slika 7-73. (a) Jedan blok matrice *Y,* (b) DCT koeficijenti.

Primer rada u ovoj fazi prikazanje na slici 7-74. Tu vidimo početnu DCT matricu, tabelu kvantizacije i rezultat dobijen deljenjem svakog DCT elementa odgovarajućim elementom tabele kvantizacije. Vrednosti u tabeli kvantizacije nisu deo standarda JPEG. Svaka aplikacija mora da ima svoju tabelu, čime utiče na odnos gubitka informacija i stepena kompresije.

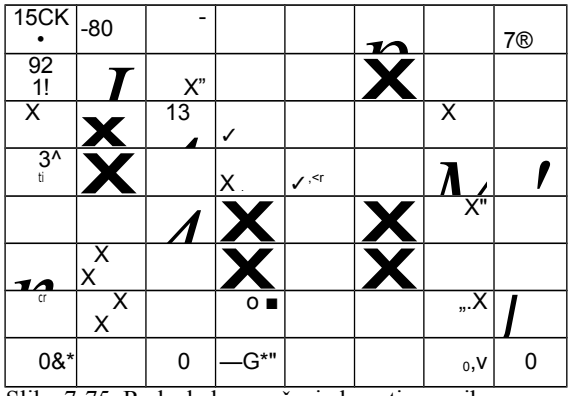
DCT koeficijenti Tabela kvantizacije Kvantizovani koeficijenti

1	1	2	4	8	16	32	64
1	1	2	4	8	16	32	64
2	2	2	4	8	16	32	64
4	4	4	4	8	16	32	64
8	8	8	8	8	16	32	64
16	16	16	16	16	16	32	64
32	32	32	32	32	32	32	64
64	64	64	64	64	64	64	64

Slika 7-74. Izračunavanje kvantizovanih DCT koeficijenata.

Četvrti korak svodi vrednost elementa (0, 0) svakog bloka (onog u gornjem levom uglu) na razliku od vrednosti odgovarajućeg elementa u prethodnom bloku. Pošto su ti elementi srednje vrednosti za svaki blok, trebalo bi da se sporo menjaju; kada se umesto apsolutnih vrednosti uzmu razlike, dobijaju se najčešće mali brojevi. Razlike se ne izračunavaju za ostale elemente bloka. Vrednosti elementa (0,0) nazivaju se DC komponentama, a ostale - AC komponentama. DC elementi („jednosmerni“) predstavljaju opšti nivo signala, a AC („naizmenični“) njegovu promenljivu komponentu.

U petom koraku se 64 elementa linearizuju i na listu se primenjuje run-length kodiranje. Skeniranjem bloka sleva udesno i odozgo nadole neće se skupiti sve nule jedna do druge, pa se zato koristi cik-cak skeniranje, kao na slici 7-75. U ovom slučaju, takvim skeniranjem se na kraju matrice nakupi 38 nula. Odgovarajući tekstualni niz tada se može skratiti jer samo treba saopštiti da sledi 38 nula. Ovaj način skraćenog kodiranja istih podataka zove se run-length kodiranje (engl. *run-length encoding).*



Slika 7-75. Redosled prenošenja kvantizovanih vrednosti.

Sada imamo listu brojeva koji predstavljaju sliku (u prostoru transformacija). U šestom koraku, da bi se brojevi uskladištili ili poslali, kodiraju se po Hafmanu - brojevi koji se pojavljuju češće dobijaju kraće oznake od onih koji se pojavljuju rede.

Algoritam JPEG možda izgleda komplikovano, ali on i *jeste* komplikovan. Pa ipak, često se koristi jer komprimuje slike u odnosu 20:1, pa i većem. Za dekodiranje slike komprimovane algoritmom JPEG, potrebno ga je izvršiti u obrnutom smeru. To ne važi za sve algoritme komprimovanja, kao što ćemo se ubrzo i sami uveriti.

Standard MPEG

Konačno smo stigli do suštine: standarda MPEG, nazvanog prema autorima, grupi eksperata za film (engl. *Motion Picture Experts Group).* Ti standardi sadrže glavne algoritme za komprimovanje videa i međunarodno su prihvaćeni od 1993. Pošto filmovi sadrže i slike i zvuk, standard MPEG može da komprimuje i audio i video. Već smo razmotrili komprimovanje zvuka i komprimovanje nepokretnih slika, pa pogledajmo sada kako se komprimuje video.

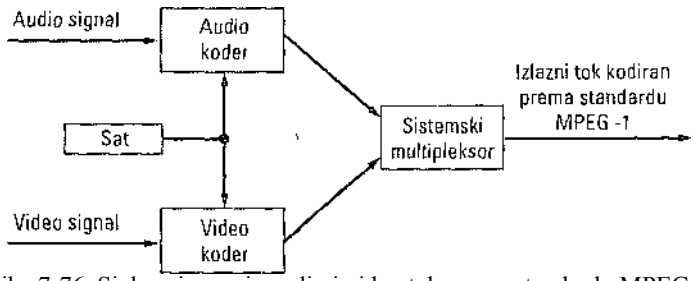
Prvo je dovršen standard MPEG-1 (Međunarodni standard 11172), sa ciljem da ostvari nivo kvaliteta za video rikorđere (352 x 240 za sistem NTSC), uz brzinu pre- nosa 1,2 Mb/s. Za prenos slika veličine 352 x 240, sa 24 bita po pilcselu i uz 25 okvira u sekundi potrebna je brzina 50,7 Mb/s, pa njeno smanjivanje na 1,2 Mb/s nije bilo baš najjednostavniji zadatak. Tu je bilo potrebno primeniti komprimovanje u odnosu 40:1. Video kodiran prema standardu MPEG-1 može se na umerene udaljenosti prenositi običnom upredenom paricom. Algoritam

MPEG-1 koristi se i za skladištenje filmova na CD diskovima.

Sledeći standard iz istog skupa bio je MPEG-2 (Međunarodni standard 13818) i prvobitno je razvijen za komprimovanje videa kvaliteta za difuzno emitovanje u propusni opseg od 4 do 6 Mb/s da bi mogao da stane u TV kanal sistema NTSC ili PAL. Kasnije je MPEG-2 dopunjen za više rezolucije, uključujući i HDTV. Sada je veoma raširen jer predstavlja osnovu za DVD i digitalnu satelitsku televiziju.

Standardi MPEG-1 i MPEG-2 zasnivaju se na sličnim principima, razlika je samo u detaljima. Najgrublje rečeno, MPEG-2 je nadskup standarda MPEG-1, s dodatnim funkcijama, formatima okvira i opcijama za kodiranje. Prvo čemo razmotriti MPEG-1, pa onda MPEG-2.

Standard MPEG-1 ima tri dela: audio, video i sistemski deo, koji upravlja s prva dva dela (slika 7-76). Koderi za audio i video rade nezavisno jedan od drugog, što pokreće pitanje sinhronizovanja dva toka podataka kod primaoca. Taj problem se rešava sistemskim satom frekvencije 90 kHz koji i jednom i drugom kođeru istovremeno šalje tekuće vreme. Vremenske oznake su 33-bitne, što omogućava 24 časa gledanja filma bez reciklovanja oznaka. Oznake se uključuju u kodirane izlazne podatke i pro- sleđuju do prijemnika, koji ih koristi za sinhronizovanje audio i video toka.



Slika 7-76. Sinhronizovanje audio i video tokova po standardu MPEG-1.

Razmotrimo sada komprimovanje videa pomoću algoritma MPEG-1. U filmovima postoje dve vrste redundantnih podataka: prostorni i vremenski. MPEG-1 isko- rišćava obe. Prostorna redundansa se može iskoristiti tako što se svaka pojedinačna slika kodira algoritmom JPEG. Taj sistem se koristi naročito ako je neophodno slobodno pristupiti svakom okvira, kao pri video produkciji. U tom režimu postižu se propusni opsezi od 8 do 10 Mb/s.

Ako se iskoristi činjenica da su uzastopni okviri međusobno gotovo identični, video se može i dodatno sažeti. Taj efekat je manji nego što izgleda na prvi pogled jer mnogi autori svake 3 ili 4 sekunde „seckaju“ scenu (da bi film uklopili u predviđeno trajanje). Pa ipak, tako čak i niz 75 veoma sličnih okvira nudi mogućnost priličnog sažimanja u odnosu na situaciju kada se svaki pojedinačan okvir zasebno kodira algoritmom JPEG.

Slika 7-77. Tri **uzastopna okvira.**

Za scene u kojima su kamera i pozadina nepokretni, a u prednjem planu se jedan ili dva glumca kredu sporo, skoro svi pikseli će u uzastopnim okvirima biti isti. Tu dobro dođe da se svaki okvir oduzme od prethodnog i samo ta razlika kodira algoritmom JPEG. Međutim, u scenama gde se kamera krede ili zumira, ta tehnika ne radi. Potreban je mehanizam koji de kompenzovati kretanje, a to je upravo ono što radi algoritam MPEG. To je i glavna razlika između algoritama MPEG i JPEG.

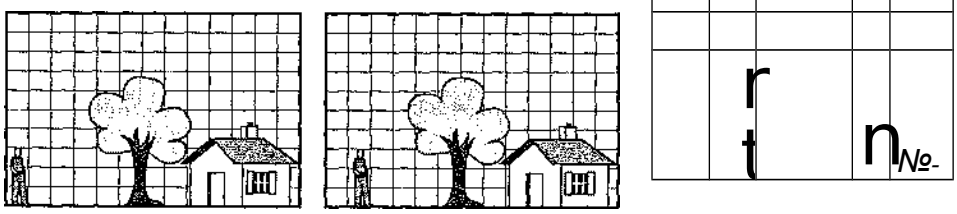
Izlazni tok algoritma MPEG-1 sadrži četiri vrste okvira:

1. I-okvire (interno kodirane): potpune nepokretne slike kodirane algoritmom JPEG.
2. P-okvire (prediktivne): razlike u odnosu na prethodni okvir, blok po blok.
3. B-okvire (dvosmerne): razlike između prethodnog i sledeđeg okvira.
4. D-okvire (DC kodirane): srednje vrednosti blokova korišcene za przo pre-motavanje unapred.

I-okviri su samo nepokretne slike kodirane varijantom algoritma JPEG, s luminan- som u punoj rezoluciji i hrominansom u polovini rezolucije duž svake ose. I-okviri se moraju periodično pojavljivati u izlaznom toku iz tri razloga. Prvo, MPEG-1 se može koristiti za višesmerno emitovanje, pri čemu se korisnici uključuju po želji. Ako svaki okvir zavisi od svog prethodnika i tako sve do prvog okvira, svako ko propusti prvi okvir nikada nede može da dekodira ostale. Drugo, ako ikada stigne pogrešan okvir, dalje dekodiranje je nemoguće. Treće, kada ne bi postojali I-okviri, dekoder bi pri brzom premotavanju unapred ili unazad morao da izračunava parametre za svaki okvir koji prelazi da bi imao potpune informacije o okviru na kome se zaustavlja. Zbog toga se I-okviri umeću u izlazni tok podataka jednom ili dvaput u sekundi.

Za razliku od toga, P-okviri kodiraju razlike između susednih okvira. Oni se zasnivaju na ideji makroblokova, koji zauzimaju 16x16 piksela u prostora luminanse i 8x8 piksela u prostoru hrominanse. Makroblok se kodira tako što se traži istovetan blok u prethodnom okviru ili se utvrđuje da se malo promenio.

Na slici 7-77 prikazan je primer scene kao poručen za kodiranje pomoću makroblokova. Tu vidimo tri uzastopna okvira sa istom pozadinom koji se razlikuju samo po položaju osobe u prednjem planu. Makroblokovi koji sadrže pozadinsku scenu bide potpuno isti, ali de makroblokovi sa osobom biti pomereni za neku nepoznatu vrednost koja se mora pratiti.



Standard MPEG-1 ne propisuje kako treba tražiti razlike, koliko ih detaljno treba tražiti ili koliko neslaganje mora da bude da bi se makroblolc proglasio izmenjenim. To sve zavisi od njegove konkretne realizacije. Na primer, u jednoj realizaciji mogao bi se tražiti makroblok na tekućoj poziciji u prethodnom okviru i na svim drugim pozicijama koje su od nje udaljene ±Ax u pravcu ose *x* i ±*t**y* u pravcu ose *y*. Za svaku poziciju mogao bi se izračunati broj poklapanja u matrici luminansi, a pobedu bi odnela pozicija s najvećom sličnošću, pod uslovom da se broj istovetnih piksela nalazi iznad nekog unapred određenog praga. U suprotnom slučaju, makroblok bi se smatrao različitim. Naravno, mogu se primeniti i mnogo složeniji algoritmi.

Kada se makroblok pronađe, kodira se njegova razlika u odnosu na prethodni okvir (za luminansu i obe hrominanse). Matrice ovih razlika podvrgavaju se tada diskretnoj kosinusnoj transformaciji, kvantizaciji, run-length kodiranju i Hafmanovom kodiranju, baš kao u algoritmu JPEG. Vrednost makroblolca u izlaznom toku predstavlja sada vektor kretanja (pomak makrobloka u svakom pravcu, u odnosu na svoju prethodnu poziciju), iza koga sledi lista brojeva kodiranih po Hafmanu. Ako se makroblok ne nađe u prethodnom okvira, tekuća vrednost se kodira algoritmom JPEG, baš kao I-okvir.

Vidi se da je ovaj algoritam veoma asimetričan. Načinom njegovog realizovanja treba da se očajnički pretraži svaka moguća pozicija u prethodnom okviru u nadi da će se pronaći nedostajući okvir, ma gde da se premestio. Takav pristup bi zaista rni- nimizovao kodirani MPEG-1 tok, ali bi to trajalo i trajalo. Možda bi se to moglo dopustiti za jednokratno kodiranje filma, ali bi bilo potpuno neprihvatljivo za video- -konferencije u realnom vremenu.

Slično tome, u svakoj realizaciji postoje nezavisna pravila o tome šta se smatra „pronađenim“ makroblokom. Ta sloboda omogućava realizatorima da nađu kompromis između kvaliteta i brzine rada njihovih algoritama, pri čemu je rezultat uvek saglasan standardu MPEG-1. Bez obzira na to koji se algoritam za pretraživanje upotrebi, rezultat je uvek tekući makroblok kodiran algoritmom JPEG ili, pak, razlika između tekućeg makrobloka i odgovarajućeg makrobloka u prethodnom okvira, ta- kođe kodirana algoritmom JPEG.

Dekodiranje formata MPEG-1, barem zasada, ide glatko. I-okviri se dekodiraju isto kao i JPEG slike. Za dekodiranje P-okvira potreban je bafer za prethodni okvir, kao i bafer u kome će dekoder sklopiti nov okvir na osnovu makroblokova koji su kodirani celi i makroblokova koji sadrže razlike u odnosu na prethodni blok. Nov okvir se sklapa jedan po jedan makroblok.

B-okviri liče na P-okvire, osim što dopuštaju da referentni makroblok bude bilo u prethodnom, bilo u narednom okviru. Ovom dodatnom slobodom omogućava se poboljšano kompenzovanje kretanja, a ona je korisna i onda kada objekti prolaze ispred ili iza drugih objekata. Da bi dekodirao B-okvire, koder istovremeno mora da u memoriji čuva tri okvira: prethodni, tekući i sledeći. Iako se pomoću B-okvira postiže najbolji stepen kompresije, ne podržavaju ih sve realizacije.

D-okviri isključivo služe za prikazivanje uprošćenih slika pri brzom premotavanju unapred ili unazad. Teško je izvesti normalno dekodiranje formata MPEG-1 u realnom vremenu. Bilo bi i previše očekivati od dekodera da samo protoči kroz video zapis

Slika 7-77. Tri uzastopna okvira.

brzinom desetak puta većom od normalne. Upravo zato i postoje D-okviri koji gene- tišu uprošćene slike. Svaki D-okvir nosi samo srednju nekodiranu vrednost jednog bloka, koju je lako prikazati u realnom vremenu. Ta mogućnost je važna onima koji pretražuju video zapis tražeći određenu scenu. D-okviri se u načelu smeštaju neposredno ispred odgovarajućih I-okvira, tako da se - nakon premotavanja - video odmah može reprodukovati normalnom brzinom.

Pošto smo završili sa formatom MPEG-1, pređimo na MPEG-2. Kodiranje sistemom MPEG-2 u osnovi liči na MPEG-1 kodiranje, sa svim onim I, P i B-okvirima. D-okviri, međutim, nisu podržani. Isto tako, pri diskretnom kosinusnom transfor- misanju, umesto blokova 8x8, koriste se blokovi 10 x 10, što daje 50% više koeficijenata, znači i viši kvalitet. Pošto je standard MPEG-2 namenjen za TV emitovanje, kao i za DVD, on podržava i progresivne i prepletene slike; MPEG-1 podržava samo progresivne slike. Između dva standarda postoji i više sitnijih razlika.

Umesto da podržava samo jednu rezoluciju, standard MPEG-2 podržava četiri rezolucije: nisku (352 x 240), glavnu (720 x 480), visoku-1440 (1440 x 1152) i visoku rezoluciju (1920 x 1080). Niska rezolucija je namenjena za video rikordere, a postoji i zbog povratne kompatibilnosti sa standardom MPEG-1. Glavna rezolucija je namenjena za normalnu TV difuziju po sistemu NTSC. Ostale dve su za HDTV. Rezultat visokog kvaliteta se sa standardom MPEG-2 obično dobij a uz brzinu podataka 4-8 Mb/s.

7.4.8 Video na zahtev

Video na zahtev ponekad se poredi s videotekom, gde korisnik bira neku od raspoloživih video traka i nosi je kuci. Pa ipak, kod videa na zahtev korisnik bira traku *od kuće* pomoću daljinskog upravljača svog televizora, a izabrani film odmah započinje da se prikazuje. Nema potrebe da se ide do videoteke. Ne treba ni naglašavati da je realizovanje videa na zahtev mnogo teže nego njegovo opisivanje. U ovom odeljku ćemo razmotriti osnovne pristupe i njihove realizacije.

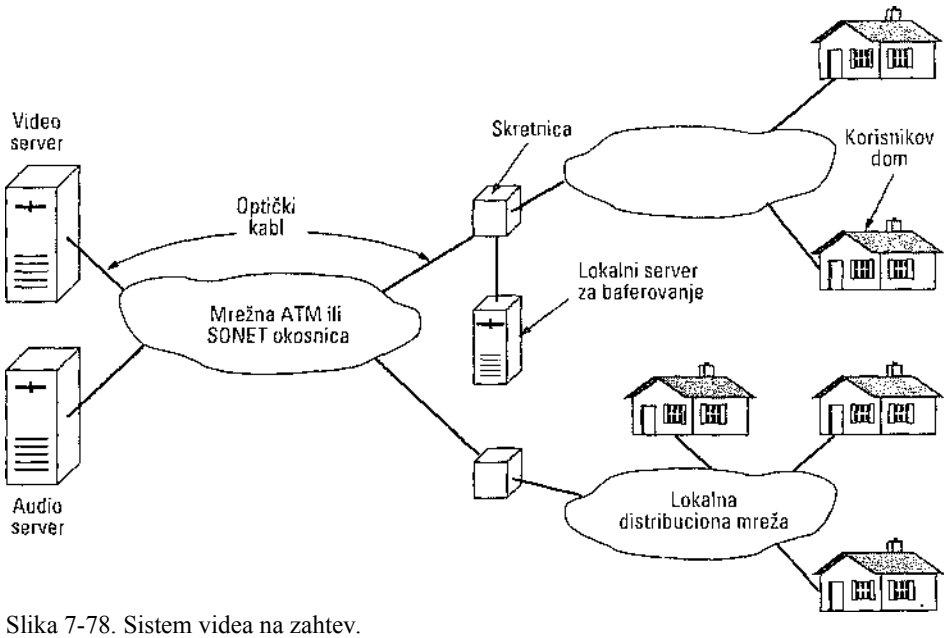
Da li video na zahtev zaista više liči na iznajmljivanje trake iz videoteke ili pre na biranje filma iz kablovskog sistema sa 500 kanala? Odgovor na ovo pitanje ima važne tehničke implikacije. Korisnici videoteka, konkretno, imaju naviku da film zaustave da bi nešto obavili u kuhinji ili kupatilu - a zatim kada se vrate - da nastave da ga gledaju od tačke gde su ga zaustavili. Gledaoci normalnog TV programa obično ne očekuju takvu mogućnost.

Ako video na zahtev želi da uspešno konkuriše videotekama, možda korisnicima treba da omogući slobodno zaustavljanje, ponovno pokretanje i premotavanje videa. Ta mogućnost znači da davalac videa svakom korisniku treba da distribuira zasebnu kopiju filma.

S druge strane, ako se video na zahtev smatra samo naprednijim televizijskim sistemom, onda je možda dovoljno da davalac videa svaku popularniju sekvencu na različitim kanalima pokreće, recimo, svakih deset minuta. Korisnik koji želi da je pogleda, možda će morati da sačeka desetak minuta. Iako u takvom sistemu nije moguće privremeno zaustavljanje i ponovno pokretanje videa, gledalac koji se pred televizor vrati posle nekoliko minuta može da se prebaci na drugi kanal na kome se ista sekvenca odvija 10 minuta kasnije. Nešto će verovatno videti ponovo, ali sigurno ništa neće propustiti. Takva šema se zove skoro video na

zahtev (engl. *near video on demand)*. Ona nudi šira uslugu po znatno nižoj ceni jer ista emisija sa servera istovremeno ide mnogim korisnicima. Između videa na zahtev i skoro videa na zahtev razlika je slična kao između vožnje sopstvenog automobila i vožnje autobusom.

Gledanje filmova (skoro) na zahtev samo je jedna od niza novih usluga koje će biti moguće kada se ostvari širokopojasni rad u mreži. Opšti model koji mnogi koriste prikazanje na slici 7-78. Tu vidimo širokopojasnu (nacionalnu ili međunarodnu) regionalnu okosnicu u centru sistema. Na nju su povezane hiljade lokalnih distribucionih mreža, kao što su kablovske televizije ili distribucioni telefonski sistemi. Lokalni di- stribucioni sistemi stižu do domova korisnika, gde se završavaju u lokalnim TV pretvaračima (engl. *set-top boxes)* u stvari, moćnim, specijalizovanim ličnim računarima.



Slika 7-78. Sistem videa na zahtev.

Za okosnicu su pomoću optičkih kablova velike propusne moći vezani brojni davaoci informacija. Neki od njih će nuditi gledanje/slušanje video/audio diskova i uslugu naplaćivati po zahtevu. Drugi će nuditi specijalizovane usluge, kao sto je „kupovina iz fotelje“, omogućavajući gledaocima da virtuelno okreću konzervu supe i čitaju na njenoj etiketi šta supa sadrži ili da sa ekrana nauče kako da rukuju motornom kosilicom. Nema sumnje da će ubrzo biti na raspolaganju sportski rezultati, vesti, stare epizode „Sage o Forsajtima“, pristup Webu i štošta drugo.

U sistemu je i lokalni server za baferovanje koji služi da korisnicima (unapred) približi sadržaj, kako bi se uštedeo propusni opseg u špicu saobraćaja. Kako će svi ovi delovi međusobno sarađivati i ko će biti vlasnik čega, predstavlja predmet žučnih rasprava u

Slika 7-79. **Hijerarhija skladištenja na video serveru.**

industrijskim krugovima. Mi ćemo u nastavku proučiti konstrukciju glavnih delova ovog sistema: video servera i distribucione mreže.

Video serveri

Da bismo ostvarili video (skoro) na zahtev, moramo imati video servere sposobne za skladištenje i istovremeno slanje velikog broja filmova. Ocenjuje se da je ukupan broj ikada napravljenih filmova oko 65.000 (Minoli, 1995). Kada se komprimuje algoritmom MPEG-2, normalan film zauzme oko 4 GB, tako da bi za 65.000 filmova bilo potrebno skladište kapaciteta 260 terabajtova. Dodajte tome sve stare, ikada napravljene televizijske programe, sportske prenose, dnevnike, reklamne spotove itd. i postade jasno da se nalazimo pred grdnim problemom skladištenja koji se mora reša- vati na industrijskom nivou.

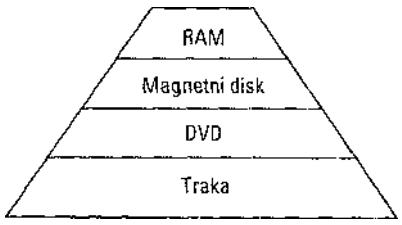
Velika količina informacija najjeftinije se skladišti na magnetnoj traci. To je odu- vek bilo tako, a izgleda da ce tako biti i nadalje. Traka Ultrium može da uskladišti 200 GB (50 filmova) po ceni od oko 1 do 2 dolara po filmu. Sada postoje komercijalni mehanički serveri koji sadrže hiljade traka i koji traženu traku automatski mogu da stave u uređaj za učitavanje. Njihovi problemi su vreme pristupanja (naročito ako treba pristupiti pedesetom filmu na traci), brzina prenosa podataka i ograničen broj čitača trake (da bi odjednom mogao da ponudi *n* filmova, ovakav server mora da ima *n* čitača trake).

Srećom, iskustvo s videotekama, javnim bibliotekama i drugim sličnim organizacijama pokazuje da nisu svi sadržaji jednako popularni. Kada je na raspolaganju *N* filmova, može se pokazati daje udeo svih zahteva za fc-ti, najpopularniji film, približno jednak *C/k,* pri čemu je C tako izabrano da sumu svih zahteva svede na 1:

$$C = 1/(1 + 1/2 + 1/3 + 1/4 + 1/5 + .. . + 1//V)$$

Na taj način, u nizu popularnosti, najpopularniji film je sedam puta popularniji od sedmog po redu. Ovaj rezultat je poznat kao Zipfov zakon (Zipf, 1949).

To što su neki filmovi mnogo popularniji od drugih nameće rešenje u obliku hijerarhije skladištenja (slika 7-79). Što je viši stupanj u hijerarhiji, performanse su bolje.

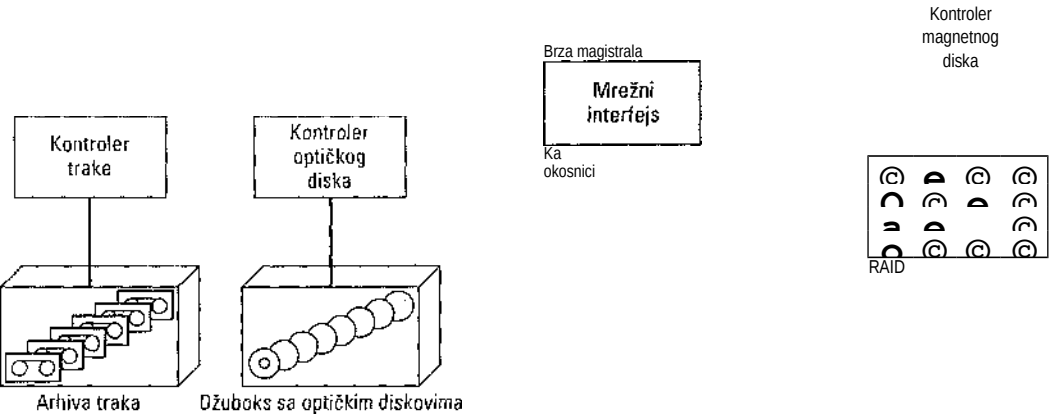


Alternativa traci je optičko skladištenje. Današnji DVD diskovi primaju 4,7 GB podataka - dovoljno za jedan film, ali sledeća generacija DVD diskova moc'i će da uskladišti dva filma. Iako podacima pristupaju sporo u odnosu na magnetne diskove (50 ms u odnosu na 5 ms), džuboksovi s hiljadama DVD diskova zbog svoje niske cene i visoke pouzdanosti predstavljaju dobru alternativu u situacijama kada se neki film izuzetno često traži.

Slede magnetni diskovi. Njihovo vreme pristupa je kratko (5 ms), brzina prenosa im je velika (320 Mb/s za SCSI 320), a kapacitet znatan (> 100 GB), zbog čega su pogodni za filmove koji se stvarno traže (za razliku od filmova koji su uskladišteni samo za slučaj da ih neko nekada zatraži). Zbog visoke cene magnetnih diskova, na njima se ne isplati čuvati filmove koji se retko traže.

U vrhu hijerarhije na slici 7-79 nalazi se radna memorija - RAM. Radna memorija je najbrži medijum za skladištenje, ali istovremeno i najskuplji. Kada cena RAM-a padne na 50 dolara po gigabajtu, film od 4 GB zauzeće RAM vredan 200 dolara, a za 100 takvih filmova trebace 200 GB memorije, vredne 20.000 dolara. Pa ipak, za video server od koga se u svakom trenutku zahteva 100 filmova, već izgleda isplativo da ih sve drži u RAM-u. A ako video server ima 100 korisnika koji ukupno gledaju samo 20 različitih filmova, to nije samo isplativo, već i dobro rešenje.

Pošto je video server u stvari ulazno-izlazni uređaj velikog kapaciteta koji radi u realnom vremenu, njegova hardverska i softverska arhitektura treba da se razlikuje od arhitekture PC računara ili UNIX radne stanice. Hardverska arhitektura tipičnog video servera prikazana je na slici 7-80. Server ima jedan ili više brzih mikroprocesora (sa izvesnom količinom lokalne memorije), deljenu glavnu memoriju, veliki keš u RAM-u za popularne filmove, niz uređaja za čuvanje filmova, kao i nešto mrežnog hardvera - normalno, optički interfejs ka SONET ili ATM okosnici na nosiocu OC-12 ili moćnijem. Ovi podsistemi su međusobno povezani magistralom izuzetno visoke brzine (barem 1 GB/s).



Slika **7-80. Hardverska arhitektura tipičnog video servera.**

Bacimo kratak pogled i na softver video servera. Mikroprocesori se koriste za prihvatanje zahteva korisnika, za pronalaženje filmova, za prenošenje podataka između uređaja, za naplaćivanje usluga i za mnoge druge svrhe. Neke od tih aktivnosti ne moraju se odvijati u realnom vremenu, ah neke moraju, pa određeni mikroprocesori (ako ne i svi) moraju da koriste operativni sistem za rad u realnom vremenu, kao što je mi- krojezgro za rad u realnom vremenu (engl. *real-time microkernel).* Takvi sistemi obično dele posao na manje zadatke, svaki s definisanim rokom izvršenja. Program za raspoređivanje poslova tada može da izvršava, na primer, algoritam najskorijeg roka ili algoritam ravnomerne brzine izvršavanja (Liu i Layland, 1973).

Softver mikroprocesora definiše i prirodu radnog okruženja koje server nudi klijentima (serverima za baferovanje i lokalnim TV pretvaračima). Popularna su dva dizajna. Prvi je klasičan sistem datoteka, u kome klijenti mogu da otvaraju, učitavaju, upisuju i zatvaraju datoteke. Izbegavajući komplikacije koje uvodi hijerarhija skladištenja i rad u realnom vremenu, takav server može da ima sistem datoteka kao u UNIX-u.

Drugo popularno okruženje imitira model video rikordera. Njegove komande naređuju serveru da datoteku otvori, reprodukuje, privremeno zaustavi i brzo premota unapred ili unazad. Za razliku od UNIX modela, dovoljna je samo komanda *PLAY* da bi server automatski nastavio da pumpa podatke konstantnom brzinom.

Samu srž softvera za video server čini softver za rad s diskom. On ima sledeća dva zadatka: da filmove iz skladišta na optičkom disku ili na traci prenosi na magnetni disk i da obrađuje zahteve za sadržajem koji dolaze iz mnogih spoljnih tokova. Mesto gde se film nalazi na serveru veoma utiče na performanse.

Skladišni diskovi se mogu organizovati kao farme i kao grupe. Kod **farme diskova** (engl. *disk farm),* svaki disk sadrži izvestan broj celih filmova. Zbog boljih performansi i pouzdanosti, svaki film treba da se nalazi na barem dva diska. Drugi način organizovanja diskova su **redundantne grupe jeftinih diskova** (engl. *Redundant Array of Inexpensive Disks, RAID)* ili, jednostavno, **grupe diskova** (engl. *disk arrays),* kod kojih je svaki film „razvučen“ na više diskova, na primer, blok 0 je na disku 0, blok 1 na disku 1 itd, do bloka *n*-1 na disku *n* - 1. Posle toga, ciklus se ponavlja, tj. blok *n* dolazi na disk 0 itd. Takva organizacija se naziva **rasparčavanje** (engl. *striping)* sadržaja na više diskova.

Grupa diskova s rasparčanim filmovima ima više prednosti u odnosu na farmu diskova. Prvo, svih *n* diskova mogu istovremeno da rade, što poboljšava performanse *n* puta. Drugo, sistem se može napraviti redundantnim ako se na svakih *n* diskova doda još jedan, pri čemu će taj redundantni disk sadržati rezultat isključive đisjunkcije (XOR) odgovarajućih blokova drugih diskova, što omogućava spašavanje podataka u slučaju otkazivanja nekog od diskova. Na kraju, rešava se problem uravnotežavanja opterećenja (više nije potrebno ručno premeštanje filmova da ne bi svi trenutni hitovi bili na istom disku). S druge strane, grupu diskova je teže organizovati od farme diskova, a ona je i osetljivija na kvarove. Isto tako, ona je nepogodna za operacije uobičajene za video rikordere, na primer, za brzo premotavanje filma.

Softver za rad s diskom treba i da u realnom vremenu opslužuje sve izlazne tokove, zadovoljavajući sve njihove vremenske zahteve. Za ovo su još do pre nekoliko godina bili potrebni složeni algoritmi za raspoređivanje poslova, ali s današnjim padom cena memorije moguć je i mnogo jednostavniji pristup. Za svaki izlazni tok u RAM-u se održava bafer

dovoljan za 10 sekundi filma (oko 5 MB). On se puni s diska, a prazni u mrežu. Kada imamo 500 MB RAM-a, iz njega se može istovremeno direktno opslužiti 100 izlaznih tokova. Naravno, podsistem za rad s diskovima mora puniti bafere brzinom 50 MB/s, ali grupa sastavljena od vrhunskih SCSI diskova može taj zahtev lako da ispuni.

Distribuciona mreža

Distribuciona mreža predstavlja skup skretnica i linija između izvorišta i odredišta. Kao što smo videli na slici 7-78, ona se sastoji od okosnice povezane s lokalnom distri- bucionom mrežom. Okosnica je obično komutirana, dok distribuciona mreža nije.

Za okosnicu je najvažnije da ima veliki propusni opseg. Ranije je ona morala da obezbedi i ravnomerno stizanje paketa, ali sada, kada i najmanji PC računar može da baferuje desetak sekundi visokokvalitetnog MPEG-2 videa, to više nije potrebno.

Lokalno razvođenje je oblast u potpunom haosu: razne kompanije isprobavaju različite mreže u različitim područjima. Sve telefonske kompanije, kablovske televizije i novi igrači - na primer, elektrodistribucije - ubeđeni su da će pobediti onaj ko prvi uleti u posao. Zbog toga smo danas svedoci instaliranja najrazličitijih tehnologija. U Japanu, u posao sa Internetom ušla je i Gradska kanalizacija, sa argumentom da oni imaju najširu cev do svakog doma (oni kroz kanalizaciju provlače optički kabl, ali moraju dobro da paze gde će se kabl na kraju pojaviti). Četiri glavne šeme lokalne distribucije videa na zahtev nose skraćena imena ADSL, FTTC, FTTH i HFC. Objasnićemo ih redom.

ADSL je prvi adut telefonskih kompanija u nadmetanju za lokalnu distribuciju. Proučili smo ovaj sistem u 2. poglavlju, pa ga nećemo ovde ponovo opisivati. Ta šema se zasniva na pretpostavci da do svake kuće u SAD, Evropi i Japanu već stiže bakama upredena parica (analogna telefonska linija). Ako bi ove parice bile upotrebljive za prenos videa na zahtev, telefonske kompanije bi zaista mogle da pokupe kajmak.

Problem je, naravno, u tome što parice na svojoj prosečnoj dužini od 10 km, ne mogu da podrže čak ni MPEG-1, a kamoli MPEG-2. Za film visoke rezolucije u punoj boji, u zavisnosti od traženog kvaliteta, potrebno je 4 do 8 Mb/s. Sistem ADSL jednostavno nije dovoljno brz, osim na kratkim rastojanjima.

Drugi kandidat telefonskih kompanija je tzv. optički kabl u **susedstvu** (engl. *Fiber To The Curb, FTTC)*. U sistemu FTTC, kompanija razvlači optički kabl od lokalne telefonske centrale do pojedinih gradskih blokova i završava ga u uređaju zvanom **priključak na optičku mrežu** (engl. *Optical Network Unit, ONU)*. Na njega se može priključiti oko 16 bakarnih lokalnih linija. Te linije su sada tako kratice, da mogu da rade kao TI ili T2 nosilac u potpunom duplelcsnom režimu, omogućavajući prenos MPEG-1, odnosno MPEG-2 filmova. Osim toga, tako je moguće održavati i video konferencije sa učesnicima koji rade kod kuće i malim preduzetnicima, jer je sistem FTTC simetričan.

Treće rešenje telefonskih kompanija je optički kabl do kuće (engl. *Fiber To The Home, FTTH)*. Prema ovoj šemi, ukoliko je potrebno, svako može da ima nosilac OC-1, OC-3 ili čak bolji. Sistem FTTH je vrlo skup i sigurno ga nećemo videti još godinama, ali kada nastupi, otvoriće čitavu lepezu novih mogućnosti. Na slici 7-63 vi- deli smo kako svako može da uspostavi sopstvenu radio-stanicu. Šta kažete na to da svaki član domaćinstva otvori svoju privatnu TV stanicu? Sva tri sistema: ADSL, FTTC i FTTH predstavljaju lokalne distribucione mreže tipa od tačke do tačke, što ne treba da vas čudi kada znate kako je

organizovan današnji telefonski sistem.

Potpuno drugačiji pristup važi za hibridni optičko-koaksijalni kabl (engl. *Hybrid Fiber Coax, HFC),* rešenje koje se trenutno najviše dopada distributerima kablovske televizije. Taj sistem je prikazan na slici 2-47(a), a pretpostavljena priča ide nekako ovako; sadašnji koaksijalni kablovi propusnog opsega 300 do 400 MHz za- menjuju se koaksijalnim kablovima propusnog opsega 750 MHz, čime im se kapacitet sa 50 do 75 kanala od 6 MHz povećava na 125 istih takvih kanala. Sedamdeset pet od pomenutih 125 kanala biće iskorišćeni za analogni TV prenos.

Preostalih 50 kanala biće pojedinačno modulisani sistemom QAM-256, što obezbe- đuje brzinu prenosa 40 Mb/s, ukupno 2 Gb/s. Razvodnici će biti postavljani dublje u naselja, tako da svaki kabl prolazi samo pored 500 domova. Prostim deljenjem možemo zaključiti da će svaka kuća dobiti 4 Mb/s, što je dovoljno za prenošenje MPEG-2 filma.

Premda sve zvuči divno, najpre kompanije treba da sve postojeće kablove zamene koaksijalnim kablovima propusnog opsega 750 MHz, da instaliraju nove razvodnilce i da uklone sve jednosmerne pojačivače, jednom reči - da zamene čitav kablovski TV sistem. Zbog toga se infrastruktura potrebna za ovaj sistem može meriti sa infrastrukturom sistema FTTC telefonskih kompanija. U oba slučaja, lokalni davalac mrežnih usluga mora da provuče optički kabl do stambenog područja. I opet, u oba slučaja, kabl se završava u optičko-električnom pretvaraču. U sistemu FTTC, poslednji segment linije je lokalna parica od tačke do tačke. U sistemu HFC, poslednji segment je deljeni koaksijalni kabl. S tehničkog aspekta, ova dva sistema nisu toliko različita koliko to tvrde njihovi pobornici.

Pa ipak, postoji jedna razlika koju ne treba preskočiti. HFC koristi deljeni medijum bez komutiranja ili usmeravanja. Svaki pretplatnik bez daljeg može da s kabla preuzme svaku informaciju. FTTC, koji predstavlja potpuno komutiran sistem, nema ovo svojstvo. Zbog toga, operateri sistema HFC žele da video serveri šalju šifrovane tokove kako korisnici koji nisu obnovili pretplatu ne bi mogli da ih primaju. Operateri sistema FTTC ne žude baš za šifrovanjem jer ono komplikuje stvari, slabi performanse, a sistemu ne povećava bezbednost. Da li je šifrovanje poželjno za kompaniju koja drži video server? Server koji drži telefonska kompanija, neka od njenih firmi-ćerki ili neki od partnera, može namerno da odluči da ne šifruje video i da pri tome javno zagovara potrebu za efikasnošću rada, a da u stvari želi da naudi svojim HFC konkurentima.

U svakoj od ovih lokalnih distribucionih mreža moguće je svako stambeno naselje opremiti lokalnim serverom za baferovanje ili s više njih. Oni, u stvari, predstavljaju umanjene verzije pravog video servera. Velika im je prednost to što u određenoj meri rasterećuju okosnicu mreže.

Na njih se unapred mogu uskladištiti filmovi na osnovu rezervacija. Ako korisnici davaocu usluga unapred saopšte koje bi filmove želeli da gledaju, on bi te filmove mogao preneti na lokalni server tokom zatišja na mreži. Verovatno će davalac tada naknadu naplaćivati po ugledu na avio-prevoznike. Može se zamisliti, na primer, da će se za filmove naručene 24 do 72 sata unapred za gledanje utorkom ili četvrtkom popodne pre 18 časova ili uveče posle 11 časova, davati popust od 27%. Filmovi koji se naruče prve nedelje u mesecu pre 8 časova izjutra, za gledanje u sreću popodne onog dana čiji je datum prost broj, imaće popust 43% itd.

7.4.9 MBone - višesmerna okosnica

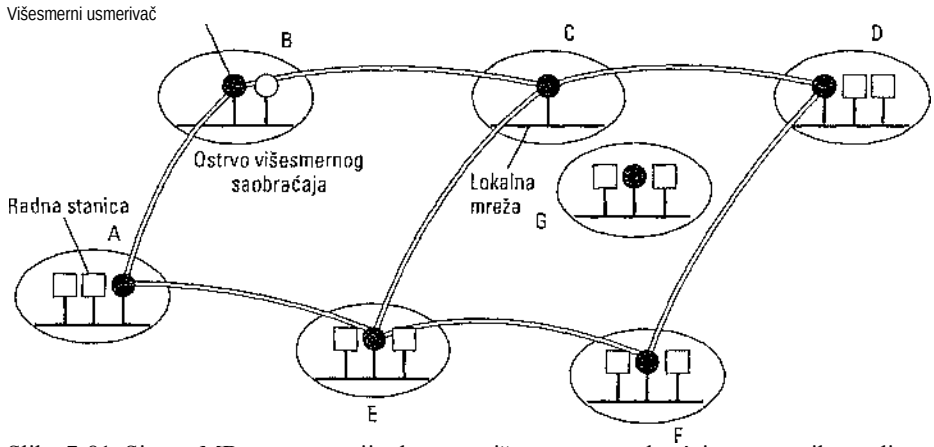
Dok sve pomenute delatnosti prave velike planove za budućnost (možda i međunarodnog) videa na zahtev, propraćene velikom pompom, zajednica korisnika Interneta tiho ugrađuje sopstveni digitalni multimedijski sistem, tzv. **višesmernu okosnicu** (engl. *Multicast Backbone, MBone)*. Objasnićemo u ovom odeljlcu šta ona predstavlja i kako radi.

Sistem MBone je zamišljen kao Internet televizija. Za razliku od videa na zahtev, gde je naglasak na zahtevanju i gledanju prethodno komprimovanih filmova uskladištenih na serveru, MBone se koristi za difuzno emitovanje živog videa u digitalnom obliku širom sveta preko Interneta. Taj sistem radi još od 1992. godine. Tako su prenošene mnoge naučne konferencije, naročito IETF skupovi, kao i važni naučni događaji, npr. lansiranje spejs-šatla. Ijedan koncert Rolingstounsa prenošen je sistemom MBone, kao i delovi Filmskog festivala u Kanu. Može se raspravljati da li i to spada u zanimljive naučne događaje.

Sa tehničkog stanovišta, MBone predstavlja viituelnu mrežu koja prekriva Internet. Ona se sastoji od ostrva sposobnih za višesmerno emitovanje, međusobno povezanih tunelima, kao na slici 7-81. Na njoj je prikazan sistem MBone sa šest ostrva, od *A* do *F,* povezanih pomoću sedam tunela. Svako ostivo (najčešće lokalna mreža ili grupa međusobno povezanih lokalnih mreža) hardverski podržava višesmerno emitovanje svojim računalima. Tuneli sprovode pakete okosnice između ostrva. Jednog lepog dana u budućnosti, kada svi usmerivači budu mogli da direktno obrađuju višesmerni saobraćaj, takva nadstruktura više neće biti potrebna, ali za sada, ona obavlja posao.

Svako ostrvo ima jedan ili više specijalnih **višesmernih usmerivača** (engl. *multicast router, mrouter)*. Neki od njih su obični usmerivači, ali većina su samo UNIX radne stanice koje izvršavaju korisnički softver (ali kao administratori). Višesmerni usmerivači su logički povezani tunelima. Paketi okosnice se kapsuliraju u IP pakete i usmeravaju kao obični jednosmerni paketi na IP adresu odredišnog višesmernog usmerivača.

Tuneli se lconfigurišu ručno. Tunel obično vodi putanjom za koju postoji fizička veza, ali to nije obavezno. Ako se zbog neke nezgode putanja kojom vodi tunel prekine, višesmerni usmerivači s dve strane tunela to neće čak ni primetiti, pošto će sam Internet automatski preusmeriti sav IP saobraćaj između njih na druge linije.



Slika 7-81. Sistem MBone se sastoji od ostrva višesmemog saobraćaja povezanih tunelima.

Kada se pojavi novo ostrvo koje želi da se priključi višesmernoj okosnici, na primer, ostrvo *G* na slici 7-81, njegov administrator to objavljuje porukom koju šalje listi slanja sistema MBone. Posle toga, s njim stupaju u vezu administratori okolnih lokacija da bi dogovorili uspostavljanje tunela. Ponekada se mreža postojećih tunela preuređuje da bi se nova ostrva iskoristila za optimizovanje topologije. Na kraju krajeva, tuneli ne postoje fizički. Oni postoje samo u virtuelnom svetu tabela višesmernih usmerivača i zato se mogu dodavati, uklanjati ili premeštati prostim menjanjem odrednica tabela. Svaka država na višesmernoj okosnici obično ima i sopstvenu okosnicu na koju su priključena regionalna ostrva. Konfiguracija sistema MBone normalno obuhvata po jedan ili dva tunela „ispod“ Atlantika i Pacifika, što joj daje globalno značenje.

Na taj način, sistem MBone u svakom trenutku predstavlja specifičnu topologiju koja obuhvata ostrva i tunele, bez obzira na broj trenutno korišćenih adresa za više- smerno slanje i nezavisno od toga ko gleda ili sluša sadržaj. To je veoma slično normalnoj (fizičkoj) podmreži, tako da se u sistemu MBone koriste uobičajeni algoritmi za usineravanje. Shodno tome, u sistemu MBone prvobitno je korišđen **protokol za višesmerno usmeravanje zasnovan na vektoru razdaljine** (engl. *Distance Vector Multicast Routing Protocol, DVMRP),* u čijoj osnovi leži Belman-Fordov algoritam vektora razdaljine. Na primer, na slici 7-81, ostrvo C može da pakete usmerava ka A preko *B* ili preko *E* (čak i preko *D).* Ostrvo C donosi odluku na osnovu rastojanja do Akoje mu šalju ta ostrva na koja dodaje rastojanje od sebe do njih. Na taj način, svako ostrvo određuje najbolju putanju do svakog drugog ostrva. Međutim, same putanje se ne koriste na opisani način, kao što ćemo ubrzo videti.

Razmotrimo sada kako se stvarno odvija višesmerno slanje. Da bi moglo da emitu- je audio ili video program na više adresa, izvorište prvo mora da dobije adresu za višesmerno emitovanje klase D, koja igra ulogu frekvencije radio-stanice ili broja kanala. Adrese klase D rezerviše program koji u bazi podataka traži slobodne adrese za višesmerno emitovanje. Mnoge višesmerne emisije mogu da se odvijaju istovremeno, pa računar može da se selektivno uključi u jednu od njih ako osluškuje odgovarajuću adresu za višesmerno emitovanje.

Svaki višesmerni usmerivač povremeno u svoje ostrvo difuzno šalje paket IGMP, pitajući koje zainteresovan za koji kanal. Računari koji žele (da nastave) da primaju jedan ili više kanala, kao odgovor šalju drugi IGMP paket. Odgovori se vremenski kombinuju da se ne bi preopteretila lokalna mreža. Svaki višesmerni usmerivač vodi tabelu kanala koje mora da obezbedi sopstvenoj lokalnoj mreži, da se ne bi trošio propusni opseg na kanale koje niko ne želi.

Višesmerno emitovanje kroz okosnicu MBone izgleda ovako. Kada audio ili video izvor generišu nov paket, on se višesmerno šalje lokalnom ostrvu pomoću odgovarajućeg hardvera. Paket stiže i do lokalnog višesmernog usmerivača koji ga kopira na sve tunele s kojima je spojen.

Svaki višesmerni usmerivač koji tunelom primi takav paket proverava da li je paket stigao najboljom putanjom, tj. putanjom za koju njegova tabela kaže daje treba koristiti do izvorišta (kao da je ono odredište). Ako je paket stigao najboljom putanjom, višesmerni usmerivač ga kopira na sve svoje druge tunele. Ako je paket stigao nekom manje optimalnom putanjom, on se odbacuje. Tako, na primer, na slici 7-81, ako tabela usmerivača C kaže da do *A* treba ići preko *B,* pa višesmerni paket od *A* stigne do C preko *B,* paket se kopira *z & D i E*. Međutim, kada višesmerni paket stigne od *A* do C preko *E* (nije najbolja putanja), jednostavno se odbacuje. To je uprvo algoritam prosleđivanja paketa ispitivanjem izvorišta o kome smo govorili u 5. poglavlju. Iako nije savršen, prilično dobro radi i vrlo se lako realizuje.

U cilju sprečavanja plavljenja Interneta, osim algoritma zasnovanog na ispitivanju izvorišta, koristi se i polje *Životni vek* IP paketa. Svaki paket polazi na put sa određenom vrednošću ovog polja (koju zadaje izvorište). Svakom tunelu se dodeljuje „težina“. Paket se propušta kroz tunel samo ako je dovoljno težak, inače se odbacuje. Prekookeanskim tunelima se, na primer, dodeljuje težina 128, tako da se njihov saobraćaj može ograničiti na izvorišni kontinent ako im se dodeli *Životni vek* 127 ili kraći. Pošto paket prođe kroz tunel, *Životni vek* mu se skraćuje za težinu tunela.

Iako algoritam za usmeravanje u sistemu MBone zadovoljavajuće radi, mnogo je napora utrošeno da se on poboljša. U jednom predlogu zadržava se osnovna ideja usmeravanja prema vektora razdaljine, ali se u algoritam uvodi hijerarhija grupisa- njem MBone lokacija u oblasti i usmeravanjem najpre u njih (Thyagarajan i Deering, 1995).

Drugi predlog zagovara upotrebu usmeravanja na osnovu stanja veze, umesto na osnovu vektora razdaljine. Konkretno, radna IETF grupa modifikovala je algoritam OSPF da bi ga prilagodila visesmernom emitovanju unutar jedinstvenog autonomnog sistema. Rezultat je bio MOSPF (Moy, 1994). Izinene su se sastojale u tome što je, osim uobičajenih informacija o usmeravanju, trebalo čuvati i potpunu MOSPF mapu višesmernih ostrva i tunela. Kada se raspolaže podacima o celokupnoj topologiji, lako je izračunati najbolju putanju za tunel od svakog ostrva do svakog drugog ostrva. Za to se može, na primer, upotrebiti Dijkstrin algoritam.

Drago područje istraživanja predstavlja usmeravanje između autonomnih sistema. Tu je radna IETF grupa razvila algoritam za višesmerno usmeravanje nezavisno od protokola (engl. *Protocol Independent Multicast, PIM)*. Algoritam PIM postoji u dve verzije, u zavisnosti od toga da li su ostrva raspoređena gusto (skoro svako želi da gleda program) ili retko (skoro niko ne želi da gleda program). U obe verzije koriste se standardne tabele za jednosmerno usmeravanje i ne pravi se nikakva nadtopologija kao u sistemima DVMRP i MOSPF.

Kod algoritma PIM koji radi u režimu gusto raspoređenih ostrva (PIM-DM), osnovno je da se „potkrešu" nekorisne putanje. Kada višesmerni paket stigne kroz „pogrešan" tunel, kroz tunel se odgovara paketom za potkresivanje koji pošiljaocu nalaže da od naznačenog izvorišta više ne šalje pakete. Kada paket stigne kroz „pravi" tunel, kopira se na sve druge tunele koji se prethodno nisu isključili. Ako su se svi dragi tuneli isključili, a za kanal nema zanimanja u lokalnom ostrvu, višesmerni us- merivač šalje poruku za potkresivanje kroz taj „pravi" tunel. Na taj način se višesmerno emitovanje automatski prilagođava i paketi se šalju samo tamo gde ih žele.

Algoritam PIM radi drugačije u režimu retko raspoređenih ostrva (PIM-SM). Taj režim je opisan u RFC dokumentu 2362. Algoritam treba da spreči zasićenje Interneta koje će, recimo, nastati samo zato što tri korisnika s Berklija žele da drže konferenciju preko adresa klase D. PIM-SM radi tako što uspostavlja tačke susreta. Svako izvorište iz višesmeme grupe PIM-SM šalje svoje pakete tačkama susreta. Svaka lokacija koja želi da se priključi šalje upit jednoj od tačaka susreta da bi s njom uspostavila tunel. Na taj način, sav PIM-SM saobraćaj transportuje se jednosmerno, umesto višesmerno. Algoritam PIM-SM postaje sve popularniji, a sistem MBone ga sve više podržava. Kako se korišćenje algoritma PIM-SM širi, algoritam MOSPF postepeno nestaje. S druge strane, sistem MBone pomalo stagnira i verovatno nikada neće uzeti većeg maha.

Pa ipak, prenos multimedije mrežom još uvek je uzbudljivo područje koje se brzo menja, čak i ako sistem MBone ne postigne veliki uspeh. Nove tehnologije i aplikacije najavljuju se svakog dana. Višesmerno emitovanje i kvalitet usluga sve se više međusobno približavaju (Striegel i Manimaran, 2002). Druga vraća tema je bežično višesmerno emitovanje (Gossain i sar., 2002). Izgleda da će čitava oblast višesmernog emitovanja i sve srodne teme privlačiti pažnju još mnogo godina.

7.5 SAŽETAK

Imena na Internetu prate hijerarhijsku šemu zvanu sistem imenovanja domena (DNS). Na samom vrhu su opštepoznati osnovni domeni, među kojima su *com* i *edu,* kao i oko 200 domena pojedinih država. DNS se realizuje kao sistem distribuiranih baza podataka sa serverima širom sveta. DNS sadrži zapise sa IP adresama, zapise za razmenu pošte i drage informacije. Šaljući upit DNS servera, proces može da prevede ime domena na Internetu u njegovu IP adresu koja se koristi za komuniciranje s tim domenom.

E-pošta je jedna od dve najomiljenije aplikacije za Internet. Koriste je svi ~ od 7 do 77 godina. Većina sistema e-pošte u svetu danas koriste poštanski sistem definisan u RFC dokumentima 2821 i 2822. Poruke poslate tim sistemom sadrže sistemsko ASCII zaglavlje s

definicijom svojstava poruke. Pomoću protokola MIME može se slati veoma različit sadržaj. Poruke se šalju protokolom SMTP koji uspostavlja direktnu TCP vezu od izvorišta do odredišta i njom šalje poruku.

Druga najomiljenija aplikacija je Web. Web je sistem povezivanja hipertekstualnih dokumenata. Na početku je svaki dokument bio strana pisana jezikom HTML, sa hi- pervezama ka drugim dokumentima. Danas XML polako preuzima posao od HTML-a. Isto tako, velika količina sadržaja generiše se automatski pomoću skriptova na servers- kom kraju veze (PHP, JSP i ASP) ili na klijentskom kraju (uglavnom JavaScript). Čitač može da prikaže dokument tako što uspostavlja TCP vezu sa serverom na kome se on nalazi, zahteva dokument i zatvara vezu. Zahtevi sadrže niz zaglavlja u kojima se saop- štavaju dodatne informacije. Performanse Weba se poboljšavaju keširanjem, uspostavljanjem više kopija servera i pomoću mreža za isporuku sadržaja.

Bežični Web se tek rađa. Prvi sistemi su WAP i i-režim, oba s malim ekranima i skromnim propusnim opsegom, ali će sledeća generacija biti moćnija.

Prenos multimedije mrežom takođe tek hvata zalet. On omogućuje da se audio i video digitalizuju i elektronskim putem pošalju slušaocima/gledaocima. Za audio je potreban manji propusni opseg, zato je njegov prenos više i napredovao. Reprodukovanje zvuka u realnom vremenu, Internet radio i Internet telefonija sada su stvarnost i za njih se svakoga dana pojavljuju nove aplikacije. Video na zahtev je područje koje tek nastaje ali se brzo razvija jer za njega postoji veliko zanimanje. I na kraju, MBone je eksperimentalna, globalna usluga digitalnog prenosa televizijskog programa preko Interneta.

ZADACI

1. Mnogi poslovni računari imaju tri jasna i globalno jedinstvena identifikatora. Koji su to identifikatori?
2. Prema podacima sa slike 7-3, da lije *little-sister.cs.vu.nl* na mreži klase A, B ili C?
3. Na slici 7-3 nema tačke posle *rowboat.* Zašto?
4. Pogodite šta bi značio emotikon :-X (ili ponekad
5. DNS koristi UDP umesto TCP. Ako se DNS paket izgubi, neće se automatski nadok- naditi. Da lije to problem, i ako jeste, kako se rešava?
6. Osim što se mogu izgubiti, maksimalna veličina UDP paketa može da bude i samo 576 bajtova. Šta se događa ako DNS ime čiju adresu treba naći prelazi ovu dužinu? Može li se zahtev poslati u dva paketa?
7. Može li računar s jedinstvenim DNS imenom imati više IP adresa? Kako do toga dolazi?
8. Može li računar imati dva DNS imena koja spadaju u različite osnovne domene? Ako može, navedite ubedljiv primer. Ako ne može, objasnite zašto.
9. Broj kompanija koje su otvorile svoje Web lokacije poslednjih godina je naglo narastao. Zbog toga se hiljade kompanija registrovalo za domen *com,* što preopterećuje server tog osnovnog domena. Dajte predlog za otklanjanje tog problema bez remećenja postojeće šeme imenovanja domena (tj. bez uvođenja novih osnovnih domena). Vaše rešenje može da izmeni klijentski kod.
10. Neki sistemi e-pošte podržavaju polje zaglavlja *Content Return* (Povraćaj sadržaja),

koje određuje da li telo poruke treba vratiti pošiljaocu u slučaju neuspešne isporuke. Da li to polje pripada omotnici ili zaglavlju?

11. Sistemi e-pošte treba da imaju imenike sa e-adresama korisnika. Da bi se takvi imenici mogli napraviti (i pretraživati), imena treba razbiti na standardne komponente (npr. ime, prezime). Navedite moguće probleme i način njihovog rešavanja ako takav imenik treba da postane svetski standard.
12. Adresa e-pošte sastavljena je od korisničkog imena, znaka @ i imena DNS domena sa zapisom *MX*. Korisničko ime može da bude stvarno ime ili prezime korisnika, njegovi inicijali ili bilo šta drugo. Pretpostavimo da je velika kompanija uočila da se previše elektronskih poruka gubi zato što pošiljaoci ne znaju korisničko ime primaoca. Postoji li način da ona taj problem reši a da ne menja DNS? Ako mislite da postoji, dajte predlog i objasnite kako radi. Ako mislite suprotno, objasnite zašto to nije moguće.
13. Binarna datoteka je dugačka 3072 bajta. Koliko će biti dugačka ako se kodira sistemom base64 i ako se posle svakih 80 bajtova i na kraju umeću znakovi CR+LF?
14. Razmotrite kodiranje MIME šemom quoted-printable. Navedite problem o kome nije bilo govora u tekstu i predložite rešenje.
15. Navedite pet MIME tipova pored onih navedenih u knjizi. Potražite odgovor u svom čitaču ili na Internetu.
16. Pretpostavimo da prijatelju želite da pošaljete MP3 datoteku, ali davalac Internet usluga vašeg prijatelja ograničava veličinu dolaznih poruka na 1 MB, a MP3 datoteka ima 4 MB. Ima li načina da se ova situacija prevaziđe na osnovu RFC dokumenta 822 i sistema MIME?
17. Pretpostavimo da je neko aktivirao sistemsku uslugu „na odmoru sam do“, a zatim šalje poruku neposredno pre odjavljivanja. Nažalost, i potencijalni primalac je na odmoru, pa je i on aktivirao istu sistemsku uslugu. Šta će se dogoditi? Hoće li se šablon- ski odgovori šetati tamo-amo sve dok se najednom kraju stvarno ne pojavi neko živ?
18. U svakom standardu, npr. u standardu RFC 822, potrebne su precizne definicije da bi različite realizacije mogle međusobno da sarađuju. Čak se i jednostavne stvari moraju pažljivo definisati. U SMTP zaglavljima dozvoljene su beline između oznaka. Ponudite još *dve* ubedljive definicije belina između oznaka.
19. Da li je sistemska usluga „na odmoru sam do“ deo korisničkog agenta ili deo agenta za prenos poruka? Naravno, ona se podešava pomoću korisničkog agenta, ali da li korisnički agent stvarno šalje odgovore? Objasnite svoj odgovor.
20. POP3 omogućava korisnicima da preuzmu e-poštu iz udaljenog poštanskog san- dučeta. Znači li to da treba standardizovati interni format poštanskih sandučića da bi svaki klijentski POP3 program mogao da čita poštanske sandučiće na svakom poštanskom serveru? Obrazložite odgovor.
21. S gledišta davaoca Internet usluga, protokoli POP3 i IMAP fundamentalno se razlikuju. Oni koji koriste POP3, obično svakodnevno prazne svoje poštanske sandučiće. Oni koji koriste IMAP svoju poštu uglavnom čuvaju na serveru. Zamislite da davaocu Internet usluga treba da pomognete da izabere protokol koji c'e da podrži. Šta sve pri tome treba da uzmete u obzir?
22. Da li Webmail koristi POP3, IMAP ili ni jedan ni drugi protokol? Ako koristi jedan od njih, zašto baš taj? Ako ne koristi nijedan, koji je od dva navedena protokola po filozofiji bliži Webmailu?
23. Kada se šalju Web strane, dodaje im se MIME zaglavlje. Zašto?
24. Kada su potrebni spoljni čitači? Kako čitač Weba zna koji da upotrebi?
25. Da li je moguća sledeća situacija: kada korisnik pritisne hipervezu u Netscapeu, pokreće

se određena pomoćna aplikacija, a kada pritisne istu vezu u Internet Explorern, pokreće se sasvim druga pomoćna aplikacija, iako su u oba slučaja primljeni isti MIME tipovi? Objasnite svoj odgovor.

26. Višenitni Web server organizovan je kao na slici 7-21. Treba mu 500 ps da prihvati zahtev i pregleda keš. Šanse da će datoteku naći u kešu i odmah je isporučiti iznose 1:1. Ako je tamo ne nađe, modul se blokira 9 ms dok se njegov zahtev disku ne svrsta u red čekanja i ne obradi. Koliko modula server treba da ima da bi mikroprocesor sve vreme bio potpuno iskorišćen (pretpostavljajući da disk nije usko grlo)?
27. Standardna *http* URL adresa očekuje da Web server osluškuje priključak 80. Međutim, moguće je da Web server osluškuje neki drugi priključak. Smislite neku razumnu sintaksu za URL adresu koja pristupa datoteci na nestandardnom priključku.
28. Iako to u tekstu nije naglašeno, u URL adresi se umesto DNS imena može iskoristiti IP adresa. Takva je, na primer, adresa *http//192.31.231.66Andex.html,* Kako čitač zna da li ime iza šeme *http* predstavlja DNS ime ili IP adresu?
29. Zamislite daje neko s Katedre za računarstvo na Stanfordu upravo napisao nov program koji želi da distribuira protokolom FTP. On smešta program u FTP direktorijum *ftp/pub/freebies/newprog.c.* Kakva će verovatno biti URL adresa za preuzimanje ovog programa?
30. Na slici 7-25, lokacija www.aportal.com pomoću kolačića vodi evidenciju o sklonostima posetilaca. Ta šema ima manu što je veličina kolačića ograničena na 4 KB, pa ako su zahtevi posetioca obimni, na primer, mnogi berzanski izveštaji, sportski rezultati, različite vrste vesti, vremenska prognoza za razne gradove, specijaliteti u brojnim kategorijama proizvoda i tako dalje, može se prekoračiti granica od 4 KB. Smislite alternativan način vođenja evidencije o sklonostima posetilaca, takav da ne pati od ovog problema.
31. Banka NemaProblema želi da svojim lenjim korisnicima olakša pristup preko mreže, pa kada se korisnik prijavi i banka njegov identitet proveri lozinkom, šalje mu kolačić s korisničkim identifikatorom. Na taj način, pri svom budućem pristupanju mrežnoj banci korisnik ne mora da se ponovo identifikuje, niti da upisuje lozinku. Šta mislite o tome? Hoće li raditi? Da li je ideja dobra?
32. Na slici 7-26, u oznaci <img> zadat je parametar *ALT.* Pod kojim okolnostima će ga čitač koristiti i kako?
33. Kako HTML kodom postižete da slika postane osetljiva na pritisak mišem? Ponudite primer.
34. Kako treba da izgleda oznaka <a> da bi tekst „ACM“ postao hiperveza ka adresi http://www. *acm, orgl*
35. Smislite obrazac za novu kompaniju Interburger, koji će omoguc'iti naručivanje hamburgera preko Interneta. Obrazac treba da sadrži ime kupca, njegovu kucnu adresu i grad, veličinu hamburgera (ogroman ili baš ogroman), kao i opciju za dodatak sira. Hamburgeri se plaćaju gotovinom pri isporuci, tako da obrazac ne treba da sadrži podatke o kreditnoj kartici.
36. Smislite obrazac u koji korisnik treba da upiše dva broja. Kada korisnik pritisne dugme Submit, server mu vraća zbir tih brojeva. Napišite serverski deo koda u obliku PHP skripta.
37. Za svaku od sledećih primena odgovorite da li je (1) moguća i (2) dali je za nju bolje primeniti PHP ili JavaScript (i zašto).
    a) Prikazivanje kalendara za svaki zahtevani mesec počev od septembra 1752,

b) Prikazivanje rasporeda letova od Amsterdama do Njujorka.

c) Prikazivanje grafika polinoma kome korisnik zadaje koeficijente.

38. Napišite u JavaScriptu program koji prihvata ceo broj veći od 2 i odgovara da li je taj broj prost. Imajte na umu da JavaScript ima odredbe if i while sa istom sintaksom kao jezici C i Java. Operator za izračunavanje ostatka celobrojnog deljenja je *%*. Ako vam treba kvadratni koren od *x,* upotrebite *Math.sqrt(x)*.
39. Jedna HTML strana izgleda ovako:
<html> <body>
 Pritisnite ovde za informacije
<body> <html>
Kada korisnik pritisne hipervezu, otvara se TCP veza i serveru se šalje niz redova teksta. Napišite sve poslate redove.
40. Zaglavlje *If-Modified-Since* može se iskoristiti da bi se proverila svežina keširane strane. Zahtevi se mogu slati za strane sa slikama, zvukom, videom itd, kao i sa HTML tekstom. Mislite li da je efikasnost ove tehnika veća/manja za JPEG slike, nego za HTML strane? Razmislite dobro šta „efikasnost“ podrazumeva i obrazložite svoj odgovor.
41. Na dan važnog sportskog susreta, npr. Kupa šampiona, zvanična Web lokacija ima mnogo posetilaca. Da li to iznenadno zagušenje ima isti smisao kao i Izbori 2000. na Floridi (vidi primer u tekstu)? Zašto ima ili zašto nema?
42. Ima li smisla da davalac Internet usluga radi kao mreža za isporuku sadržaja? Ako ima, kako to radi? Ako nema, šta ne valja?
43. U kojim okolnostima korišćenje mreže za isporuku sadržaja nije dobra ideja?
44. Bežični Web terminali imaju mali propusni opseg, zbog čega raste važnost efikasnosti kodiranja. Smislite šemu za efikasno prenošenje engleskog teksta preko bežične veze na WAP uređaj. Slobodni ste da pretpostavite da terminal ima više megabajta ROM-a i skroman mikroprocesor. *Pomoć,* razmislite o prenošenju japanskog, gde svaki simbol predstavlja reč.
45. Kompakt disk sadrži 6.50 MB podataka. Da li se za audio CD koristi komprimo- vanje? Objasnite svoj odgovor.
46. Na slici 7~57(c) javlja se šum kvantizacije zbog uzoraka veličine 4 bita koji treba da predstave devet vrednosti signala. Prvi uzorak, u vremenu 0, tačan je, ali sledećih nekoliko nisu. Kolika je procentualna greška za uzorke uzete pri 1/32, 2/32 i 3/32 vremenskog perioda?
47. Može li se za smanjenje propusnog opsega potrebnog za Internet telefoniju iskoristiti psihoakustički model? Ako može, koje uslove (ako postoje) treba prethodno zadovoljiti? Ako ne može, zašto?
48. Audio server za rad u realnom vremenu udaljen je 50 ms (u jednom pravcu) od pro- grama za reprodukovanje multimedije. On šalje podatke brzinom 1 Mb/s. Ako program za reprodukovanje ima bafer veličine 1 MB, šta možete da kažete o položaju njegove gornje, odnosno donje oznake?
49. Algoritam za preplitanje sa slike 7-60 ima tu prednost što može da preživi povremeno gubljenje paketa, a da u reprodukciju ne unese pauzu. Međutim, kada se koristi za Internet telefoniju, ima i jednu sitnu manu. Koja je to mana?
50. Da li Internet telefonija ima isti problem sa zaštitnom barijerom kao i prenos audija u realnom vremenu? Ponudite širi odgovor.
51. Kolika je brzina (b/s) potrebna za prenošenje nekomprimovanih okvira u boji veličine 800 x 600 piksela, uz 8 bitova po pikselu i 40 okvira u sekundi?

52. Može li jednobitna greška u MPEG okviru da pokvari još nešto osim okvira u kome je nastala? Objasnite.
53. Razmotrite video server za 100.000 korisnika, pri čemu svaki korisnik gleda dva filma mesečno. Polovina filmova se emituje u 8 časova uveče. Koliko istovremeno filmova server treba da emituje tokom ovog perioda? Ako je za svaki film potreban propusni opseg 4 Kb/s, koliko treba OC-12 veza između servera i rnreže?
54. Pretpostavimo da Zipfov zakon važi za pristupanje video serveru sa 10.000 filmova. Ako server čuva 1000 najpopularnijih filmova na magnetnom disku, a ostalih 9000 na optičkom disku, napišite izraz za izračunavanje udela svih obraćanja magnetnom disku. Napišite i mali program koji će izračunavati taj izraz.
55. Neki mešetari Web lokacijama registrovali su imena domena koja su (osim namerne pravopisne greške) ista kao imena poznatih lokacija, npr. www.microsfot.com. Napravite spisak barem pet takvih domena.
56. Mnogi su registrovali DNS imena tipa www.rec.com, gde *rec* predstavlja neku običnu reč. Za svaku od sledećih kategorija, navedite pet Web lokacija i opišite šta one predstavljaju (na primer, www.stomach.com je Web lokacija gastroenterologa na Long Ajlendu). Evo liste kategorija: životinje, hrana, kućni aparati, delovi tela. Za poslednju kategoriju, molim vas da se držite delova iznad struka.
57. Napravite nekoliko sopstvenih emoji znakova koristeći bit mapu veličine 12 x 12 tačaka. Neka to budu (moj) dečko, (moja) devojka, profesor i političar.
58. Napišite POP3 server koji prihvata sledeće komande: *USER, PASS, LIST, RETR, DELE i QUIT.*
59. Prepravite server sa slike 6-6 tako da postane pravi Web server, koristeći komandu *GET* za HTTP 1,1. Server treba i da prihvata poruku *Host.* Takođe, treba da održava keš datoteka koje su nedavno preuzimane s diska i da ih na zahtev šalje iz keša kad god je to moguće.

8

BEZBEDNOST *HA* MREŽI

Tokom nekoliko prvih decenija njihovog postojanja računarske mreže su uglavnom koristili istraživači sa univerziteta da bi razmenjivali e-poštu i zaposleni u pre- duzećima da bi zajednički koristili kancelarijske štampače. U tim okolnostima na bezbednost nilco nije obraćao pažnju. Ali danas, kada milioni ljudi koriste mreže za bankarske transakcije, za kupovanje ili za popunjavanje poreskih prijava, bezbednost na mreži počinje da predstavlja veliki problem. U ovom poglavlju ćemo proučiti razne aspekte bezbednosti na mreži, ukazati na neke zamke, i razmotriti brojne algoritme i protokole za obezbeđivanje mreže.

Bezbednost je široka tema koja obuhvata mnoge nedozvoljene radnje. Najjednostavniji oblik bezbednosnih mera jeste sprečavanje radoznalca da tajno čitaju ili - što je još gore - menjaju poruke namenjene dragim osobama. Tu su i osobe koje pokušavaju da neovlašćeno pristupe udaljenim uslugama. Bezbednost treba da odgovori i na pitanje da li poruka „Morate platiti u petak“ stvarno dolazi od Poreske uprave ili od Mafije. Bezbednost se bavi i problemima presretanja i ponovnog slanja legitimnih poruka, kao i osobama koje poriču da su poslale određene poruke.

Većinu bezbednosnih problema namerno izazivaju zlonamerne osobe koje na taj način žele da ostvare neku dobit, da privuku pažnju ili da nekome naude. Nekoliko najčešćih profila takvih osoba navedeno je na slici 8-1. Posle čitanja ove liste trebalo bi da bude jasno da se obezbeđivanje mreže ne svodi samo na bezgrešno programiranje. Obezbeđivanje obuhvata nadmudrivanje često inteligentnog, ostrašćenog i - ponekad - dobrostojećeg protivnika. Trebalo bi da bude jasno i to da mere koje

osujecuju prosečnog protivnika neće biti dovoljne da spreče nekog ozbiljnijeg napadača. Policijski dosijei pokazuju da većinu napada ne izvršavaju osobe izvan firme koje se ilegalno kače na telefonsku liniju, već neki od njenih razočaranih ili poniženih nameštenika. Na sve to treba misliti kada se projektuje sistem bezbednosti.

Profil neprijatelja	Njegov cilj
Student	Da se zabavi čitajući tuđu e-poštu
Haker	Da proveri nečiji bezbednosni sistem; da ukrade podatke
Trgovački predstavnik	Da uveri kupce da predstavlja celu Evropu, a ne samo Andoru
Poslovni čovek	Da otkrije strategiju svog konkurenta
Bivši nameštenik	Da se osveti za otkaz
Računovođa	Da prisvoji novac preduzeća
Berzanski posrednik	Da porekne obećanje koje je klijentu dao e-poštom
Prevarant	Da ukrade brojeve kreditnih kartica i da ih zatim preproda
Špijun	Da sazna vojne i industrijske tajne neprijatelja
Terorista	Da ukrade poverijive podatke o proizvodnji biološkog oružja

Slika 8-1. Profili nekih osoba koje mogu da izazovu bezbednosne probleme.

Bezbednosni problemi na mreži mogu se grubo svrstati u četiri tesno povezane kategorije: tajnost, proveru identiteta, nemogućnost poricanja i kontrolu integriteta. Tajnost, zvana i poverljivost, vodi računa o tome da informacije ne dospeju u ruke neovlašćenih osoba. To je prva stvar na koju pomislimo kad pomenemo bezbednost na mreži. Proverom identiteta treba da utvrdite s kim razgovarate pre nego što otkrijete osetljive podatke ili preduzmete poslovni poduhvat. Nemogućnost poricanja se svodi na potpisivanje: kako ćete dokazati daje vaš kupac stvarno elektronskim putem naručio deset miliona levih kako-li-se-već-zovu, po 89 dinara, ako on kasnije bude tvrdio da je cena bila 69 dinara? Možda tvrdi da ništa nije ni naručio? Najzad, kako možete biti sigurni daje poruku koju ste primili stvarno poslao kupac, a ne neki zlobnik koji ju je presreo i namerno izmenio?

Problematika koju smo naveli (tajnost, provera identiteta, nemogućnost poricanja i kontrola integriteta) postoji i u klasičnim sistemima, ali uz nekoliko važnih razlika. Integritet i tajnost se postižu korišćenjem preporučene pošte i zaključavanjem dokumenata. Pljačku poštanskog voza danas je teže izvesti nego u doba Džesi Džemsa.

Isto tako, ljudi jasno mogu da uoče razliku između originala i fotokopije dokumenta, a ta razlika im je često važna. Probajate sami, fotokopirajte jedan ispravan ček. Original unovčite u ponedeljak, a fotokopiju probajte da unovčite u utorak i uočite razliku u ponašanju bankarskog službenika u jednom i dragom slučaju. Kod elektronskih čekova, original i kopija se ne mogu razlikovati. Možda će malo potrajati dok banke ne nauče kako da ih razlikuju.

Ljudi identifikuju druge ljude prepoznajući njihova lica, glasove i rukopis. Uobičajena je identifikacija dokumenata potpisom, a falsifikati se otkrivaju na osnovu rukopisa, vrste mastila i hartije na kojoj je pisano. Sve to ne postoji u elektronskim dokumentima, pa je potrebno drugačije rešenje.

Pre nego što počnemo da opisujemo sama rešenja, treba reći nešto o rnestu sistema za bezbednost na mreži u skupu protokola. To mesto verovatno nije i jedino, već bez- bednosti treba da pomalo doprinese svaki sloj. U fizičkom sloju, kačenje na žicu se može sprečiti ako se prenosni vodovi smeste u hermetički zatvorenu cev, ispunjenu gasom pod pritiskom. Svaki pokušaj bušenja cevi izazvaće oslobađanje gasa i sniženje pritiska, što će aktivirati signal za

uzbunu. Takva tehnika se koristi u nekim vojnim sistemima.

U sloju veze podataka, paketi se na linijama od tačke do tačke mogu šifrovati na jednom kraju, a dešifrovati na dragom. Sve se to može uraditi na nivou sloja veze, bez znanja viših slojeva. Takvo rešenje, međutim, pada u vodu ako paket treba da prođe više usmerivača jer se na svakom usmerivaču mora dešifrovati, a tada postaje ranjiv na napad iz samog usmerivača. Opisano rešenje talcođe ne dozvoljava da samo neke sesije budu zaštićene (npr. samo one koje se odnose na kupovinu preko mreže pomoću kreditne kartice). Pa ipak, šifrovanje veze (engl. *link encryption),* kako se naziva ova tehnika, može se lako ostvariti u svakoj mreži i često je korisno.

U mrežnom sloju se mogu instalirati zaštitne barijere koje će u mreži čuvati samo „dobre" pakete, a one „loše" zadržavati napolju. IP bezbednost radi na ovaj način.

U transportnom sloju se mogu šifrovati čitave veze od jednog do dragog kraja, tj. od jednog procesa do dragog. Za maksimalnu bezbednost obavezno je ovakvo šifrovanje.

Na kraju, provera identiteta i nemogućnost poricanja mogu se ostvariti samo u sloju aplikacija.

Pošto se bezbednost ne može uklopiti samo u jedan sloj, ne može se ni opisati samo u jednom od dosadašnjih poglavlja ove knjige. Njoj pripada zasebno poglavlje.

Iako će ovo neophodno poglavlje biti dugačko i prepuno tehničke terminologije, ono je za sada možda i irelevantno. Postoje čvrsti dokazi da bezbednosni sistem - na primer, u bankama - uglavnom narušavaju nekompetentni nameštenici, labava bez- bednosna pravila i zlonamerni službenici banke; mnogo manje su za to krivi oštroumni kriminalci koji se kace na telefonske linije da bi ukrali šifrovane poruke. Alto neko može „mrtav hladan" da ušeta u banku s karticom za bankomat koju je našao na ulici i da, izjavivši daje zaboravio svoj PIN broj, na licu mesta dobije nov (u ime poverenja koje banka ukazuje korisnicima), nikakvo šifrovanje neće sprečiti zloupotrebu. U tom smislu, knjiga Rosa Andersona (Anderson, 2001) zaista otvara oči jer na stotinama primera industrijskih firmi dokumentuje da se skoro svako narušavanje bezbednosti događa zbog labavih pravila poslovanja ili zbog toga što se samom obezbeđenju poklanja malo pažnje. Ipak srno puni optimizma da će sa širenjem elektronske trgovine kompanije ispraviti propuste u načinu svog poslovanja i na taj način vratiti tehničkim aspektima obezbeđenja ulogu koja im pripada.

Osim bezbednosti u fizičkom sloju, sve drage bezbednosne mere oslanjaju se na šifrovanje. Zbog toga ćemo naše proučavanje bezbednosti početi kriptografijom. U odeljku 8.1 navešćemo neke osnovne principe. U odeljcima 8.2-8.5 ispitaćemo nekoliko osnovnih algoritama i struktura podataka koji se koriste u lcriptografiji. Zatim ćemo detaljno proučiti način primene ovih koncepata za obezbeđivanje mreže. Zaključićemo s nekoliko kratkih napomena o tehnologiji i društvu.

Pre nego što počnemo, naglasimo i šta ovde nećete naći. Pokušali smo da se u ovom poglavlju usredsredimo na problematiku mreža, iako je često teško povući granicu između tih problema i problematike operativnih sistema i aplikacija. Na primer, ovde ništa ne govorimo o proveri identiteta korisnika biometrijskim metodama, o bezbednosti lozinki, o napadima usmerenim na prelivanje bafera, trojancima, lažnom predstavljanju, logičkim bombama, virusima, crvima i sličnom. Sve te teme opširno su obrađene u 9. poglavlju knjige *Modem Operating Systems* (Tanenbaum, 2001). Zainteresovani će tamo naći sve sistemske aspekte bezbednosti. A sada, na posao.

8.1 KRIPTOGRAFIJA

Izraz **kriptografija** potiče iz grčkog i znači „tajno pisanje“. Kriptografija ima dugu i zanimljivu istoriju koja seže hiljadama godina unazad. U ovom odeljku ćemo samo ocrtati njene glavne principe, kao osnovu za ono što sledi. Za potpunu istoriju kriptografije preporučujemo Kanovu knjigu (Kahn, 1995). Savremeni pogled na bezbednost, kao i algoritme, protokole i aplikacije vezane za nju naći ćete kod Kaufmana i saradnika (2002). Strožiji matematički pristup naći ćete kod Stinsona (2002), a malo popularniji (i razumljiviji) kod Burnetta i Painea (2001).

Profesionalci prave razliku između **šifre** (engl. *cipher)* i **koda** (engl. *code).* Šifra omogućava zamenu znak za znak (bit za bit), bez obzira na jezičku strukturu poruke. S druge strane, kodom se jedna reč zamenjuje drugom rečju ili simbolom. Kodovi se više ne koriste, premda su imali burnu istoriju. Najuspešniji kod koji je ikada smišljen koristile su Američke oružane snage tokom Drugog svetskog rata na Pacifiku. Jednostavno su na vezu postavili dva Navaho Indijanca koji su međusobno razgovarajući na svom jeziku vojne izraze opisivali recima iz Navaho jezika. Na primer, protivten- kovslco oružje zvali su *čaj-da-gahi-nail-caidi,* bukvalno: ubica kornjača. Navaho jezik je izuzetno akcentovan i složen, i ne postoji u pisanom obliku. Ne treba posebno isticati da u Japanu niko ništa nije znao o njemu.

Septembra 194.5, ovaj kod su opisale novine *San Diego Union* sledećim recima: „Tokom tri godine, gde god su se marinci iskrcali, Japanci su mogli hvatati samo čudne grlene poruke, nešto između zapevanja tibetanskog kaluđera i grgotanja iz naglo otvorene pivske flaše“. Japanci nikada nisu provalili ovaj kod, a mnogi Navaho „ši- franti“ dobili su vojna priznanja za izuzetno zalaganje i hrabrost. Činjenica da su Amerikanci uspeli da provale japanski kod, a da Japanci nikada nisu uspeli da provale Navaho kod bila je odlučujuća za američku pobedu na Pacifiku.

8.1.1 Uvod u kriptografiju

Istorijski posmatrano, kriptografiji su svoj doprinos dale četiri grupe ljudi: vojnici, diplomate, letopisci i ljubavnici. Najveći doprinos tokom vekova davali su vojnici jer im je i cilj bio najznačajniji. Unutar vojnih organizacija, poruke koje treba slati šifro- vane, tradicionalno su davane loše plaćenim, nižim činovnicima - šifrantima. Zbog same količine šifrovanih poruka taj posao nije mogao biti poveren nekolicini dobro plaćenih specijalista.

Sve do pojave računara, jedna od glavnih prepreka uspešnom šifrovanju bila je sposobnost šifranta da posao obavi ispravno, često uz minimalnu opremu i usred bitke.

Dodatna prepreka je bila nemogućnost brzog prelaska s jedne kriptografske metode na drugu, jer je bilo potrebno dodatno obučiti veliki broj ljudi. Međutim, zbog rizika da neprijatelj zarobi šifranta, bilo je neophodno obezbediti da se metoda šifro- vanja po potrebi odmah promeni. Takvi međusobno suprotstavljeni zahtevi doveli su do modela prikazanog na slici 8-2.

Slika 8-2. Kriptografski model (za šifrovanje simetričnim ključem).

Poruka koju treba šifrovati, poznata kao **osnovni tekst** (engl. *plaintext*), transfor- miše se pomoću funkcije čiji su parametri zadati **ključem** (engl. *key).* Rezultat šifro- vanja, **šifrovani tekst** (engl. *ciphertext),* prenosi se kurirom ili radio-vezom. Pretpostavljamo da neprijatelj, ili **uljez** (engl. *intruder),* može da čuje i tačno da zapiše ceo šifrovan tekst. Međutim, za razliku od potencijalnog primaoca, on ne zna ključ za dešifrovanje pa ne može lako da dešifruje poruku. Ponekada uljez može ne samo da osluškuje komunikacioni kanal (pasivan uljez), već i da presreće poruke, a zatim da ih ponovo šalje, da u kanal ubacuje sopstvene poruke ili da legitimne poruke menja pre nego što stignu do primaoca (aktivan uljez). Veština razbijanja šifara, **krip- toanaliza,** i veština njihovog smišljanja (kriptografija), zajedno čine disciplinu koja se zove **kriptologija.**

Korisno je da unapred dogovorimo kako ćemo označavati osnovni tekst, odgovarajući šifrovan tekst i same ključeve. Izrazom $C = E_K(P)$ označićemo da osnovni tekst *P,* šifrovan ključem *K,* daje šifrovan tekst C. Slično tome, $P = D_K(C)$ znači da se dešifrovanjem šifrovanog teksta C ponovo dobija osnovni tekst *P.* Odavde sledi daje

$$D_K(E_K(P)) = P$$

Navedenim obeležavanjem ističe se da su *E* i *D* samo matematičke funkcije. Obe su dvoparametarske funkcije, a jedan od parametara (ključ) napisali smo kao indeks, a ne kao argument, da bismo ga razlikovali od same poruke.

Osnovna pretpostavka kriptografije je da kriptoanalitičar zna metode korišcene za šifrovanje i dešifrovanje. Dragim recima, kriptoanalitičar detaljno poznaje metodu šifrovanja *E* i metodu dešifrovanja *D* sa slike 8-2. Trud koji treba uložiti u smišljanje, proveravanje i instaliranje novog algoritma svaki put kada stari algoritam bude otkriven (ili se smatra da je

otkriven), oduvek je praktično onemogućavao da algoritam ostane tajan. Kada mislite da je nešto tajna što u stvari nije, imaćete više štete nego koristi.

Na ovom mestu uskače ključ, Ključ je (srazmerno) kratak tekstualni niz kojim se bira jedan od više mogućih načina šifrovanja. Za razliku od opšte metode, koja se može menjati možda svakih nekoliko godina, ključ se može menjati kad god to zatreba. Prema tome, naš osnovni model obuhvata stabilnu, svakom dostupnu opštu metodu, čiji su parametri, međutim, definisani tajnim ključem koji se može lako menjati. Načelo da kriptoanalitičar poznaje algoritme i da tajnost leži isključivo u ključevima, naziva se **Kerkofov princip,** po flamanskom vojnom kriptografu Augustu Kerckhof- fu koji ga je prvi formulisao 1883. godine (Kerckhoff, 1883). Tako, imamo:

Kerkofov princip: Svi algoritmi moraju biti javni; samo su ključevi tajni.

Ne možemo dovoljno istaći potrebu za javnošću algoritama. Pokušaj da se algoritam zadrži u tajnosti, tzv. **obezbeđivanje kroz prikrivanje** (engl. *security by obscurity),* nilcada ne dovodi do rezultata. Osim toga, kada objavi algoritam, kriptograf dobija „besplatne usluge" kriptologa iz akademskih institucija koji pokušavaju da razbiju sistem i tako pokažu svima koliko su pametni. Ako brojni stručnjaci tokom perioda od pet godina ne uspeju da provale algoritam, on je najverovatnije prilično tvrd orah.

Pošto očuvanje tajnosti stvarno leži u ključu, njegova dužina je od prvenstvene važnosti. Razmotrite najobičniji katanac sa šifrom. Otvorićete ga *ako* pojedine cifre složite u pravu kombinaciju. To svako zna, ali ne i pravu kombinaciju. Ključ dužine dve cifre znači da postoji 100 mogućih kombinacija. Ključ dužine tri cifre povećava broj kombinacija na 1000, onaj sa šest cifara - na milion. Stoje duži ključ, veći je i **uloženi trud** kriptoanalitičara za njegovo razbijanje. Trud neophodan za isprobavanje svih mogućih kombinacija raste eksponencijalno s dužinom ključa. Tajna se može sačuvati ako smislite solidan (ali javan) algoritam i upotrebite dugačak ključ. Da biste mlađeg brata sprečili da čita vašu e-poštu, dovoljan je i 64-bitni ključ. Za normalno ko- rišćenje u poslovne svrhe potreban je barem 128-bitni ključ. Da biste sprečili njuškanje po državnim poslovima, potreban vam je barem 256-bitni ključ, ako ne i duži.

S gledišta kriptoanalitičara, problem kriptoanalize postoji u tri osnovne varijante. Kada pred sobom ima šifrovan tekst, ali ne i odgovarajući osnovni tekst, suočen je s problemom **isključivo šifrovanog teksta.** Kriptogrami koji se pojavljuju na zabavnim stranicama novina spadaju u takve probleme. Kada kriptoanalitičar ima izvesnu količinu uporednog osnovnog i šifrovanog teksta, to je problem s **poznatim osnovnim tekstom.** Konačno, kada kriptoanalitičar može da šifruje osnovni tekst koji sam izabere, imamo problem **izabranog osnovnog teksta.** Lako biste rešili kriptograme iz novina ako biste smeli da pitate: Kako će tekst ABCDEFGHIJKL izgledati kad se šifruje?

Početnici često smatraju da je šifra bezbedna ako šifrovan tekst može da izdrži probu direktnog dešifrovanja. To je prilično naivan stav. Kriptoanalitičar u mnogim slučajevima može da pogodi deliće šifrovanog teksta. Na primer, mnogi računali će kada ih pozovete prvo zahtevati da se prijavite. Kada ima nekoliko uporednih delića običnog i šifrovanog teksta, posao kriptoanalitičara postaje mnogo lakši. Da bi posti gao bezbednost, kriptograf treba da gaji konzervativan stav i da osigura nepovredivost sistema čak i ako napadač može da dešifruje proizvoljnu količinu običnog teksta.

Metode šifrovanja tradicionalno se dele u dve kategorije: u supstitucione i trans-

pozicione šifre. Pozabavićemo se kratico svakom od njih, kao uvod u savremenu kriptografiju.

8.1.2 Supstitucione šifre

Kod supstitucionog šifrovanja svako slovo ili grupa slova šifruju se tako što se zamenjuju drugim slovom ili grupom slova. Jedna od najstarijih poznatih takvih šifara je Cezarova šifra koja se pripisuje Juliju Cezaru. Po toj metodi, *a* postaje *D, b* postaje *E, c* postaje F..., a z postaje C. Na primer, *napad* postaje *QDSDG*. U primeri- ma će osnovni tekst biti pisan malim slovima, a šifrovan velikim.

Ako malo uopštimo Cezarovu šifru, možemo reci da šifrovani tekst, umesto uvek za 3 slova, može biti pomeren za *k* slova abecede u odnosu na osnovni tekst. Tada *k* postaje ključ opšte metode šifrovanja cirkularnim pomeranjem slova po abecedi. Cezarova šifra je možda zavela Pompeja, ali od tada nikog više.

Sledeće poboljšanje šifrovanja postižemo ako svaki simbol osnovnog teksta - neka to bude 26 slova abecede - preslikamo u neko drugo slovo. Na primer,

osnovni tekst:	ab c d e f gh i j k lmn o p q r s t u v w x y z
šifrovan tekst:	**Q W E R T Y U I O P A S D F G H J K L Z X C V B N M**

Opšta šema supstitucije jednog simbola drugim zove se šifrovanje zamenom slova slovom (engl. *monoalphabetic substitution),* a ključ je tekstualni niz od 26 slova abecede (ili odgovarajući broj slova nekog drugog alfabeta). S tim ključem, osnovni tekst *napad* pretvorio bi se šifrovanjem u *FQHQR.*

Taj ključ na prvi pogled izgleda sasvim nemoguće otkriti jer iako kriptoanalitičar zna kako sistem radi (zamena slova slovom), on ne zna koja se od 26! *~4* x IO^{26} mogućih kombinacija koristi kao ključ. Za razliku od situacije pri pogađanju Cezarove šifre, ovde nije preporučljivo isprobavati sve kombinacije. Kada bi svaka pojedinačna prove- ra trajala samo 1 ns, računani bi ipak trebalo IO^{10} godina da proveri sve kombinacije.

Pa ipak, ako imate na raspolaganju i sasvim malu količinu šifrovanog teksta, razbijanje šifre postaje iznenađujuće lako. Napadač u osnovi koristi statistička svojstva govornih jezika. U engleskom, na primer, *e* je najčešće lcorišćeno slovo, a slede ga redom *t, o, a, n, i* itd. U najčešće dvoslovne kombinacije, tzv. digrafe, spadaju *th, in, er, re* i *an*, a u troslovne (trigrafe), *the, ing, and* i *ion.*

Kriptoanalitičar koji pokušava da dešifmje tekst u kome je svako slovo zamenjeno drugim slovom, počeće tako što će izračunati učestalost pojave svakog slova u šifro- vanorn tekstu. Tada bi ono s najvećom učestalošću mogao, probe radi, prevesti sa e, a ono koje se samo nešto rede pojavljuje, sa *t.* Zatim bi tražio trigrafe oblika *tXe,* koji neodoljivo podupiru zaključak da *X* predstavlja *h.* Isto tako, ako se često pojavljuje niz *thYt,* onda *Y* verovatno predstavlja *a.* Uz ove podatke, dalje može da traži trigraf

oblika *aZW.,* koji najverovatnije znači *and.* Pogađajući najčešća slova, digrafe i trigra- fe, i znajući verovatne međusobne odnose suglasnika i samoglasnika, kriptoanalitičar gradi približan osnovni tekst, slovo po slovo.

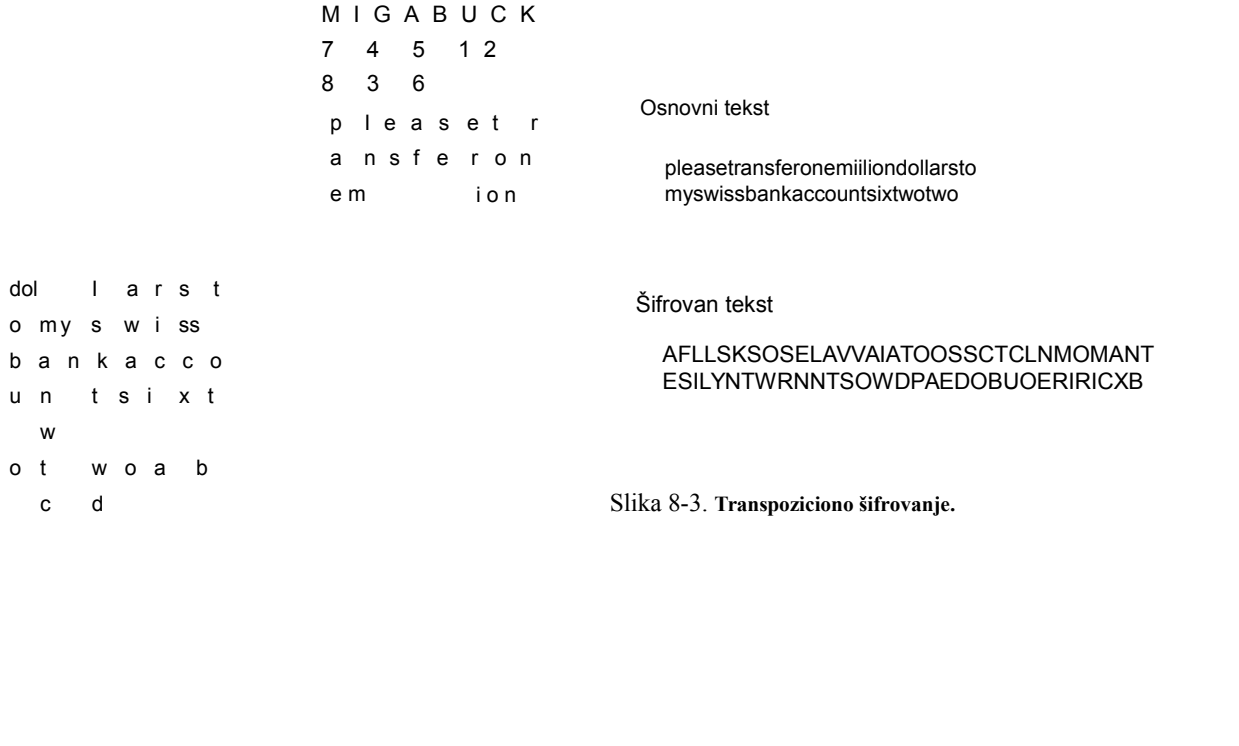
Prema drugom pristupu, pogađa se verovatna reč ili fraza. Na primer, razmotrite sledeci šifrovan tekst koji potiče od jedne računovodstvene firme (podeljen u blokove od po pet znakova):

CTBMN BYCTC BTJDS QXBNS GSTJC BTSWX CTQTZ CQVUJ
QJSGS TJOZZ MNQJS VLNSX VSZJU JDSTS JOUUS JUBXJ
DSKSU JSNTK BGAOJ ZBGYQ TLCTZ BNYBN QJSW

U poruci jedne računovodstvene firme verovatno se pojavljuje *lečfinancicil* (finan- sijski). Imajući u vidu da r*ečfinancial* sadrži dva slova *i* sa četiri draga slova između njih, u šifrovanom tekstu ćemo tražiti takav razmak između dva ista znaka. Nalazimo 12 takvih kombinacija koje počinju na pozicijama 6,15,27, 31,42,48, 56,66,70,71, 76 i 82. Međutim, samo dve (one na pozicijama 31 i 42) imaju isto sledeće slovo (koje bi odgovaralo slovu *n* osnovnog teksta). I opet, samo kombinacija na poziciji 31 ima pravilno raspoređene znalce koji odgovaraju slovu *a,* pa znamo da reč *financial* počinje na poziciji 30. Dešifrovanje je odavde lako, ako iskoristimo statističku učestalost korišćenja slova u engleskom jeziku.

8.1.3 Transpozicione šifre

Zamenjivanje slova slovom ne dira njihov redosled u osnovnom tekstu - samo ih skriva. **Transpoziciono šifrovanje** (engl. *transposition cipher),* s druge strane, ne skriva slova osnovnog teksta, ali im menja redosled. Na slici 8-3 prikazana je jedna transpoziciona šifra za premeštanje po kolonama. Takva šifra kao ključ ima reč ili fra zu u kojima se ne ponavlja nijedno slovo. U prikazanom primeru ključ je MEGA- BUCK. Ključ u stvari numeriše kolone: prva kolona se nalazi ispod slova najbližeg početku abecede i tako redom. Osnovni tekst se upisuje u horizontalne redove i matrica po potrebi dopunjava besmislicama. Šifrovani tekst se očitava iz kolona njihovim rastućim redosledom.



Slika 8-3. **Transpoziciono šifrovanje.**

Da bi razbio prikazanu šifru, kriptoanalitičar mora prvo da bude svestan da se radi o transpozicionoj šifri. Prateći učestalost pojavljivanja slova *E, T, A, O, I, N* itd, lako se utvrđuje da li ona odgovara normalnom pojavljivanju u osnovnom tekstu. Kada se to utvrdi, jasno je da se radi o transpozicionoj šifri jer pri takvom šifrovanju svako slovo predstavlja sebe, pa učestalost njihovog pojavljivanja odgovara učestalosti pojavljivanja u normalnom tekstu.

Sledeći korak je pogađanje broja kolona. Reč ili fraza koji igraju ulogu ključa u mnogim slučajevima se mogu pogoditi iz konteksta. Pretpostavimo, na primer, da kriptoanalitičar očekuje da se u poruci negde pojavi reč *milliondollars* (milion dolara). Obratite pažnju na to da se zbog prelamanja ove fraze u šifrovanom tekstu pojavljuju digrafi *MO, IL, LL, LA, IR* i *OS*. Slovo *O* u šifrovanom tekstu sledi slovo *M* (tj, ona su u koloni 4 susedna po vertikali) jer se u pretpostavljenoj frazi nalaze na odstojanju jednakom dužini ključa. Daje upotrebljen ključ dužine sedam, umesto navedenih pojavili bi se digrafi *MD, IO, LL, LL, IA,OR* i *NS*. U stvari, za svaku dužinu ključa bi se u šifrovanom tekstu pojavili karakteristični digrafi. Ispitujući različite mogućnosti, kriptoanalitičar često može da utvrdi dužinu ključa.

Poslednji korak je pronalaženje redosleda kolona. Kada je njihov broj *k* mali, u svakom od *k(k* - 1) parova kolona mogu se potražiti digrafi čija učestalost odgovara osnovnom engleskom tekstu. Par kod koga je slaganje najveće, smatra se pogođenim. Posle toga se svaka od preostalih kolona dodaje iza pogođenog para i ispituje učestalost pojave digrafa i trigrafa. Kombinacija koja najviše odgovara osnovnom tekstu smatra se ispravnom. Na isti način se pronalazi prava kolona koju treba staviti ispred već „pogođenih". Čitav postupak se nastavlja sve dok se ne pronađe ispravan redo- sled kolona. U tom trenutku postoji velika verovatnoća da će osnovni tekst moći da se prepozna (na primer, ako se dobije reč *milloin,* jasno je gde se nalazi greška).

Neke transpozicione šifre prihvataju blok fiksne dužine osnovnog teksta proizvodeći blok šifrovanog teksta, takođe fiksne dužine. Takve šifre se mogu potpuno opisati listom koja saopštava redosled generisanja šifrovanih znakova. Na primer, šifra sa slike 8-3 može se smatrati blok-šifrom od 64 znalca. Rezultat njene primene je niz 4, 12, 20, 28, 36, 44, 52, 60, 5, 13,..., 62. Drugim recima, četvrti znak koji se šifruje *(a)* u rezultatu šifrovanja je prvi, zatim sledi dvanaesti (/) itd.

8.1.4 Jednokratna zaštita

Neprobojna šifra se u stvari može napraviti sasvim lako; tehnika je poznata de- cenijama. Najpre za ključ izaberite nasumičan niz bitova. Zatim osnovni tekst pretvorite u niz bitova koristeći, na primer, ASCII kodove slova. Izvršite isključivu disjunlcciju (XOR) između dva niza bitova, bit po bit. Rezultujući šifrovan tekst ne može se provaliti jer se u dovoljno velikom uzorku svako slovo, svaki digraf ili trigraf pojavljuju približno isti broj puta. Ova metoda, poznata kao jednokratna zaštita (engl. *one-time pad),* otporna je na sve sadašnje i buduće napade, bez obzira na to kakvim računarom raspolaže napadač. Razlog se može naći u teoriji informacija: u poruci jednostavno nema informacija jer su svi delovi osnovnog teksta iste dužine podjednako verovatni.

Primer korišcenja jednokratne zaštite prikazan je na slici 8-4. Prvo se poruka „1 love you" (volim te) pretvara u 7-bitni ASCII kod. Zatim se bira jednokratni ključ - ključ 1, i podvrgava isključivoj disjunkciji s porukom da bi se dobio šifrovan tekst. Kriptoanalitičar može da isproba sve moguc'e jednokratne ključeve iste dužine da bi proverio kakav će

osnovni tekst dobiti. Na primer, mogao bi isprobati jednokratni ključ 2 sa slike, pri čemu će dobiti osnovni tekst 2, „Elvis lives“ (Elvis je živ), koji ga može, ali i ne mora zadovoljiti (to nije tema ove knjige). U stvari, za svaki osnovni tekst dužine 11 ASCII znakova postoji jednokratni ključ koji ga generiše. To je smisao prethodno iznete tvrdnje da šifrovani tekst ne sadrži informacije: iz njega možete da izvučete bilo koju poruku iste dužine.

Poruka 1:	1001001	0100000	1101100	1101111	1110110	1100101	0100000	1111001	1101111	1110101	0101110
Ključ 1:	1010010	1001011	1110010	1010101	1010010	1100011	0001011	0101010	1010111	1100110	0101011
Šifrovan tekst:	0011011	1101011	0011110	0111010	0100100	0000110	0101011	1010011	0111000	0010011	0000101
Ključ 2:	1011110	0000111	1101000	1010011	1010111	0100110	1000111	0111010	1001110	1110110	1110110
Osnovni tekst 2:	1000101	1101100	1110110	1101001	1110011	0100000	1101100	1101001	1110110	1100101	1110011

Slika 8-4. Jednokratna zaštita poruke i mogućnost dobijanja bilo kakvog teksta ako se za dešifrovanje upotrebi neki drugi ključ.

Jednokratna zaštita je teorijski bez premca, ali u praksi pokazuje mnoge nedostatke. Kao prvo, ključ se ne može zapamtiti, pa ga i pošiljalac i primalac moraju zapisati. Ako neprijatelj može da zarobi bilo kog od njih, nije zgodno da mu pronađe šifru zapisanu na komadiću papira. Osim toga, ukupnu količinu prenesenih podataka ograničava dužina ključa. Ako se špijunu posreći, pa najednom dođe do velike količine vrednih podataka, možda neće moći da ih prosledi poslodavcu jer je ključ prekratak. Drugi problem je osetljivost metode na ispuštene ili umetnute znake. Ako pošiljalac i primalac izgube korak, svi podaci od tog trenutka biće nečitljivi.

Jednokratna zaštita će s dolaskom sve moćnijih računara možda postati praktična za neke primene, Ključ se može smestiti na specijalan DVD disk koji sadrži više gigabajta podataka, a ako se transportuje u odgovarajućem omotu i na početak umetne video sekvenca od nekoliko minuta, čak nec'e pobuditi ni sumnju. Naravno, u mrežama koje rade gigabitnim brzinama može postati mučno da se nov DVD umeće u čitač svakih 30 sekundi. DVD se, takođe, mora lično dostaviti primaocu pre nego što se pošalje bilo kakva poruka, pa i to znatno smanjuje praktičnost takvog pristupa.

Kvantna kriptografija

Zanimljivo je da možda postoji rešenje za prenošenje jednokratnog ključa mrežom, a ono dolazi iz sasvim neočekivane oblasti: kvantne mehanike. Ovo područje je još uvek eksperimentalno, ali početni rezultati obećavaju. Ako se kvantna kriptografija može usavršiti tako da postane efikasna, za sve šifrovanje će se koristiti jednokratna zaštita, postoje ona dokazano bezbedna. U nastavku ćemo ukratko objasniti način funkcionisanja kvantne kriptografije (engl. *quantum cryptography)* i konkretno opisati protokol BB84, koji je dobio ime po svojim autorima i godini objavljivanja (Bennet i Brassard, 1984).

Korisnik Alisa želi da uspostavi jednokratno zaštićenu komunikaciju s dragim korisnikom Bobom. Alisa i Bob su glavne ličnosti ili principali (engl.*principals)* naše priče. Bob je, na prirner, bankar* s kojim Alisa želi da posluje. Alisa i Bob su imena glavnih ličnosti koje se poslednjih desetak godina sreću u skoro svakom radu ili knjizi o kriptografiji. Kriptografi neguju tradiciju. Ako bismo za glavne ličnosti naše priče izabrali Endija i Barbara, niko ne bi poverovao ni u šta iz ovog poglavlja. Onda, neka bude po propisu.

Kada bi Alisa i Bob mogli da ostvare jednokratnu zaštitu, mogli bi i da vode tajne međusobne razgovore. Međutim, kako to da urade a da prethodno ne razmene DVD diskove? Pretpostavićemo da se Alisa i Bob nalaze na dva kraja optičkog kabla kroz koji mogu da šalju svetlosne signale. Međutim, jedan bezočni uljez, Trudi, može da iseče kabl i umetne aktivnu račvu. Tako Trudi može da čita bitove koji dolaze iz bilo kog smera. Takođe, u oba smera može da šalje lažne poruke. Takva situacija Alisi i Bobu izgleda beznadežno, ali tu možda može da pomogne kvantna kriptografija.

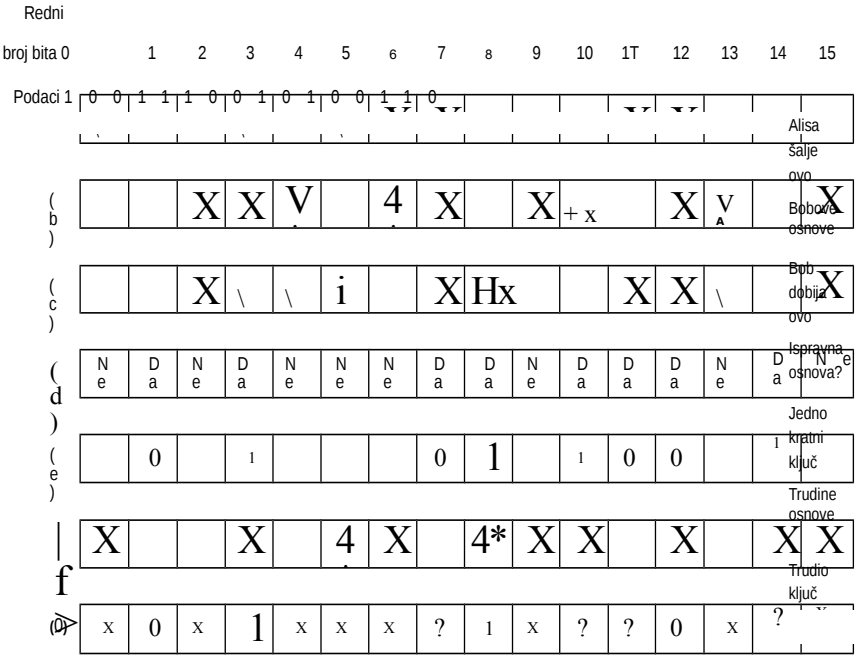
Kvantna kriptografija se zasniva na činjenici da se svetlost prostire u obliku malih paketa - fotona, koji imaju izvesna neobična svojstva. Osim toga, svetlost se propuštanjem kroz odgovarajući filtar može polarizovati, što dobro znaju proizvođači naočara za sunce i fotografi. Kada se snop svetlosti (tok fotona) propusti kroz polari- zacioni filtar, svi propušteni fotoni biče polarizovani u pravcu ose filtra (npr. vertikalno). Ako se snop sada propusti kroz drugi polarizacioni filtar, jačina svetlosti koju propušta drugi filtar biće proporcionalna kvadratu kosinusa ugla koji zaklapaju ose dva filtra. Ako su ose međusobno upravne, neće proći nijedan foton. Apsolutna orijentacija filtara nije važna; bitan je samo ugao između njihovih osa.

Da bi generisala jednokratni ključ, Alisi trebaju dva kompleta polarizacionih filtara. Prvi komplet se sastoji od horizontalnog i vertikalnog filtra. Ta kombinacija se zove pravougaona osnova (engl. *rectilinear basis*). Ta osnova predstavlja jedan ko- ordinatni sistem. Dragi komplet filtara je isti kao prvi, samo zakrenut za 45 stepeni, tako da osa jednog filtra ide iz donjeg levog ugla u gornji desni ugao, a dragog iz gornjeg levog ugla u donji desni ugao. To je dijagonalna osnova (engl. *diagonal basis)*. Tako, Alisa ima dve osnove koje po volji može da stavlja u svetlosni snop koji emitu- je. U stvari, Alisa nema četiri zasebna filtra, već kristal čija se polarizacija u četiri dozvoljena pravca može velikom brzinom menjati električnim. putem. Bob ima istu opremu. To što Alisa i Bob imaju na raspolaganju dve osnove ključno je za kvantnu kriptografiju.

Za svaku osnovu, Alisa jednom pravcu dodeljuje vrednost 0, a dragom vrednost 1. U primera koji prikazujemo u nastavku pretpostavićemo daje vertikalnom pravcu do- delila vrednost 0, a horizontalnom 1. Nezavisno od toga, za dragu osnovu Alisa pravcu iz donjeg levog ugla u gornji desni ugao dodeljuje vrednost 0, a onom drugom vrednost 1. O svemu tome obaveštava Boba šaljući mu nešifrovanu poruku.

Sada Alisa bira jednokratni ključ služeći se, na primer, generatorom slučajnih brojeva (stoje po sebi složena tema). Ona ga šalje Bobu, bit po bit, birajući za svaki bit nasumično jednu od svoje dve osnove, pri čemu njen svetlosni izvor svaki put emituje odgovarajuće polarizovan foton. Ona može, na primer, birati redom osnove: dijagonalnu, pravougaonu, pravougaonu, dijagonalnu, pravougaonu itd. Da bi poslala svoj

jednokratni ključ (1001110010100110) pomoću ovih osnova, ona će slati fotone kao na slici 8-5(a). Kada su zadati jednokratni ključ i redosled osnova, time je jedinstveno određena i polarizacija svakog bita. Bitovi koji se šalju pojedinačnim fotonima zovu se **kubiti** (engl. *gubits)*.



Slika 8-5. Primer kvantne kriptografije.

Bob ne zna koje osnove treba da upotrebi, pa ih nasumično bira za svaki dolazni foton, kao na slici 8-5(b). Kada ispravno izabere osnovu, dobija ispravan bit. Kada pogrešno izabere osnovu, dobija proizvoljan bit jer foton koji prođe kroz polarizacio- ni filtar zakrenut pod uglom od 45 stepeni u odnosu na njegovu sopstvenu ravan polarizacije može s jednakom verovatnoćom da se polarizuje prema filtra ili upravno na njegovu osu. To svojstvo fotona predstavlja temelj kvantne mehanike. Shodno tome, neki primljeni bitovi su ispravni, a neki proizvoljni - Bob ne ume da ih razlikuje. Rezultati koje dobija Bob prikazani su na slici 8-5(c).

Kako će Bob znati koje je osnove primio ispravno, a koje pogrešno? On ce jednostavno, nešifrovanim putem, saopštiti Alisi koju je osnovu koristio za svaki bit, a ona će mu na isti način odgovoriti šta je bilo ispravno, a šta pogrešno, kao na slici 8-5(d). Pomoću toga, svako od njih dvoje može da sastavi niz tačnih bitova, kao na slici 8-5(e). Taj niz će u prošeku biti upola kradi od originalnog niza bitova, ali pošto to znaju obe strane, one ga mogu iskoristiti kao jednokratni ključ. Alisa na početku treba samo da pošalje niz bitova koji je nešto preko dva puta duži od potrebnog ključa i na kraju će dobiti jednokratni ključ željene dužine. Problem je rešen.

Ali, sačekajte. Zaboravili smo Trudi. Pretpostavimo daje radoznala, pa je presekla kabl i umetnula sopstveni detektor i predajnik. Nažalost, ni ona ne zna koju osnovu treba da upotrebi za koji foton. Može jedino da postupi kao i Bob - da za svaki foton na- sumično bira osnovu. Njen izbor je prikazan na slici 8-5(f). Kada Bob kasnije, običnom tekstualnom porokom, saopšti Alisi koje je osnove koristio, a ona mu na isti način odgovori koje su bile tačne, Trudi saznaje gde je pogrešila, a šta je ispravno pogodila. Na slici 8-5 ispravno je pogodila bitove rednih brojeva 0, 1, 2, 3,4, 6, 8, 12 i 13. Međutim, iz Alisinog odgovora sa slike 8-5(d)zna da su samo bitovi 1, 3,7, 8,10,11,12 i 14deo jednokratnog ključa. Četiri takva bita (1, 3, 6 i 12) pogodila je tačno i ulovila njihove ispravne vrednosti. Međutim, za druga četiri (7,10,11 i 14) je pogrešila i ne zna njihove vrednosti. Na taj način, Bob zna da jednokratni ključ, prema slici 8-5(e), počinje sek- vencom 01011001, ali Trudi, prema slici 8-5(g), zna samo 01?1??0?.

Naravno, Alisa i Bob su svesni daje Trudi možda ulovila deo njihovog jednokratnog ključa, pa bi želeli da joj otežaju posao. To mogu ako ključ transformišu na neki način. Mogu ga, na primer, izdeliti u blokove od po 1024 bita, kvadrirati svaki blok da bi dobili 2048-bitne brojeve i te brojeve nadovezati u jednokratan ključ. Uz ograničene informacije o nizu prenesenih bitova, Trudi nikako ne može da ih kvadrira, tako da u stvari nema ništa u rakama. Transformisanje prvobitnog jednokratnog ključa u drugačiji ključ sa ciljem da se umanji vrednost informacija koje ima uljez zove se **pojačanje privatnosti** (engl. *privacy amplification).* U praksi se umesto običnog kva- driranja koristi složeno transformisanje kod koga svaki rezultujući bit zavisi od svakog polaznog bita.

Jadna Trudi. Ne samo da nema pojma kako izgleda ključ, već i sagovornici znaju za nju. U krajnjoj liniji, ona Bobu mora da prenese svaki primljeni bit kako bi ga održala u uverenju da razgovara sa Alisom. Problem je u tome što ona šalje bit koji je primila uz *svoju* nasumičnu, oko 50% pogrešnu polarizaciju, što će izazvati mnoge greške u onome što primi Bob.

Kada Alisa na kraju počne da šalje stvarne podatke, ona ih šalje uz kod koji temeljno ispravlja greške u hodu. S Bobovog stanovišta, jednobitna greška u jednokratnom ključu isto je što i jednobitna greška u prenosu. On uvek dobija pogešan bit, ovako ili onako. Ako je, međutim, kod za ispravljanje grešaka u hodu dovoljno moćan, on uprkos greškama može da rekonstruiše poroku, ali može i da izbroji greške. Ako broj grešaka uveliko prevazilazi očekivanu učestalost grešaka koju izaziva korišćena oprema, on zna da se neka „Trudi“ prikačila na liniju i može da preduzme odgovarajuće mere (npr. da naloži Alisi da pređe na radio-vezu, da pozove policiju itd.). Kada bi Trudi mogla da klonira fotone, pa da original pošalje Bobu dok ispituje kopiju, mogla bi da prikrije svoje prisustvo; međutim, zasada se foton ne može savršeno klonirati. Kada bi i mogla da klonira fotone, to ne bi smanjilo ulogu kvantne kriptografije za uspostavljanje jednokratnih ključeva.

Iako je eksperimentalno utvrđeno da kvantna kriptografija može da radi na optičkom kablu dužine 60 km, odgovarajuća oprema je složena i skupa. Ipak, sama ideja je i dalje privlačna. Više podataka o kvantnoj kriptografiji naći ćete kod Mullinsa (2002).

8.1.5 Dva fundamentalna principa kriptografije

Iako ćemo u nastavku poglavlja proučiti mnoge kriptografske sisteme, u osnovi svakog od njih leže dva principa koja treba unapred razumeti.

Redundansa

Prvi princip je da sve šifrovane poruke moraju sadržati izvestan višak podataka - podatke koji nisu neophodni za razumevanje sadržaja. Možda ćete ovu potrebu bolje razumeti najednom primeru. Razmotrite kompaniju „Fotelja je Moj Dom“ (FMD) koja svojih 60.000 proizvoda na zahtev naručioca šalje poštom. Smatrajući to vrlo efikasnim, FMD programeri su odlučili da narudžbenica treba da sadrži 16-bitno ime naručioca i polje od 3 bajta (1 bajt za količinu i 2 bajta za kod proizvoda). Ta posled- nja 3 bajta treba da se šifruju vrlo dugačkim ključem koji znaju samo kupac i FMD.

Na prvi pogled sve izgleda bezbedno, a u izvesnom smislu i jeste jer pasivni uljezi ne mogu da dešifruju poruke. Nažalost, sistem ima i fatalan propust zbog čega je neupotrebljiv. Pretpostavimo da je nedavno otpuštena službenica kivna na svoju bivšu firmu i da želi da se osveti. U trenutku odlaska iz firme, ona sa sobom nosi i listu kupaca. Zatim provodi noć pišući program za fiktivne narudžbe od strane kupaca stvarnih imena. Pošto nema listu ključeva, u poslednja tri bajta narudžbenice smešta proizvoljne brojeve i firmi šalje stotine takvih narudžbenica.

Kada stignu ove poruke, FMD računar pomoću imena kupaca pronalazi ključeve i dešifruje poruke. Pogubno po firmu, skoro svako 3-bajtno polje odgovara kodu nekog proizvoda, pa računar počinje da štampa otpremnice. Iako može izgledati čudno što neko naručuje 837 klackalica i .540 bazenčića s peskom, računar verovatno smatra da kupac planira da otvori lanac dečjih igrališta. Na taj način, aktivni uljez (bivša službenica) može da izazove ogromne probleme, čak i kada ne razume poruke koje generiše njen računar.

Opisani problem se može resiti ako se u svaku poi'uku doda višak podataka. Na primer, ako se narudžbenice produže na 12 bajtova, od kojih prvih 9 moraju biti nule, tada će napad propasti jer bivši službenik ne može da generiše veliki broj naizgled ispravnih poruka. Iz ove priče treba izvuci pouku da svaka poruka mora da sadrži znatan višak podataka, kako aktivni uljezi ne bi mogli da nasumično šalju lažne poruke koje bi bile protumačene kao važeće.

Međutim, višak podataka istovremeno olakšava posao kriptoanalitičaru. Pretpostavimo da je sistem naručivanja proizvoda iz fotelje visokokonkurentan posao i da firma Više Volim Kauč (VVK), glavni takmac firme FMD, želi da sazna koliko bazenčića s peskom. FMD prodaje. Da bi to saznali, prikačiće se na FMD telefon. Krip- toanaliza prvog sistema s porukama od 3 bajta praktično je nemoguća jer kada kriptoanalitičar dođe do ključa, ne može da kaže da li je ključ ispravan. U krajnjoj liniji, skoro svaka poruka je u tehničkom smislu ispravna. Kod drage, 12-bajtne šeme, kriptoanalitičar lako može da razdvoji ispravne i neispravne poruke. Odavde sledi

Prvi princip kriptografije: Poruke moraju sadržati izvestan višak podataka

Drugim recima, kada dešifruje poruku, primalac mora imati načina da utvrdi da lije poruka ispravna, što čini jednostavnim pregledanjem i možda nekim jednostavnim izračunavanjem. Višak podataka je neophodan da bi se aktivnim uljezima onemogućilo slanje smeća koje će prevariti primaoca i navesti ga, kad takvu poruku dešifruje, da postupi prema njoj. Međutim, taj isti višak podataka olakšava pasivnim uljezima upad u sistem, tako da se mišljenja u pogledu

ovoga ne slažu potpuno. Osim toga, višak podataka nikada ne sme da bude u obliku *n* nula na početku ili na kraju poruke, pošto se obradom takvih poruka pomoću određenih kriptografskih algoritama dobijaju sasvim predvidivi rezultati, što olakšava posao kriptoanalitičara. CRC polinom je mnogo bolji izbor od serije nula jer ga primalac lako može proveriti, a kriptoanalitičaru zadaje više muke. Još bolji izbor je kriptografsko heširanje, o čemu ćemo govoriti kasnije.

Ako se za tenutak vratimo kvantnoj kriptografiji, možemo da utvrdimo i kakvu ulogu kod nje igra višak podataka. Zbog toga što Trudi presreće fotone, neki bitovi jednokratnog ključa koji prima Bob biće pogrešni. Bob može da utvrdi da u porukama postoje greške, ako poruke sadrže izvestan višak podataka. Jedan sasvim prizeman način je da se poruka ponovi dva puta uzastopce. Ako dve pristigle kopije ne budu iste, Bob će znati daje kabl veoma bučan ili da neko ometa prenos. Naravno, previše je sve slati dva puta; Hamingov i Rid-Solomonov kod predstavljaju efikasnije načine otkrivanja i ispravljanja grešaka. Međutim, i dalje ostaje potreba za izvesnim viškom podataka da bi se ispravne poruke mogle razlikovati od neispravnih, naročito u prisustvu aktivnog uljeza.

Svežina

Drugi princip kriptografije predviđa inere kojima se može proveriti da li je primljena poruka zaista sveža, tj. poslata sasvim nedavno. Tim merama sprečavate aktivnog uljeza da reprodukuje stare poruke. Bez takvih mera bi naša bivša službenica mogla da se prikači na FMD telefonsku liniju i da samo u beskonačnost ponavlja ranije pristigle sasvim legitimne narudžbe. Odavde sledi

Dragi princip kriptografije: Potrebna je metoda za sprečavanje napada ponovljenim slanjem poruka

Takva mera je uključivanje u svaku poruku vremenske oznake koja važi, recimo, samo 10 sekundi. Primalac bi približno u tom periodu mogao da čuva poruke, da ih poredi i odbacuje duplikate. Poruke starije od 10 sekundi bile bi odbačene, zajedno s ponovo poslatim porukama. Kasnije ćemo govoriti i o nekim dragim merama.

8.2 ALGORITMI ZA ŠIFROVANJE SIMETRIČNIM KLJUČEM

I u savremenoj kriptografiji koriste se ideje klasične kriptografije (transpozicione i supstitucione), ali je naglasak na drugim stvarima. Ranije su kriptografi uglavnom koristili jednostavne algoritme, a danas vidimo upravo suprotno: cilj je napraviti što složeniji i zapetljaniji algoritam za šifrovanje da kriptoanalitičar, čak i kada sam izabere proizvoljan šifrovani tekst, ne bude u stanju da ga dešifruje bez odgovarajućeg ključa.

Prva klasa algoritama koju ćemu proučiti u ovom poglavlju obuhvata tzv. **algoritme za šifrovanje simetričnim ključem** (engl. *symmetric-key algorithms)* jer se kod njih isti ključ koristi i za šifrovanje i za dešifrovanje. Jedan takav algoritam prikazan je na slici 8-2. Mi ćemo se posebno zadržati na **blok-šiframa** (engl. *block-cipher),* kod kojih se n-bitni blok osnovnog teksta pomoću ključa pretvara u n-bitni blok šifrovanog teksta.

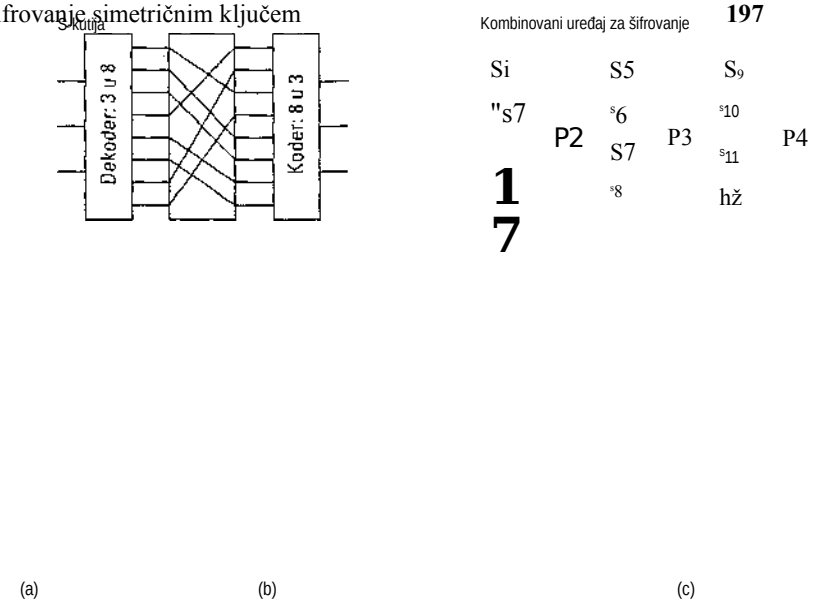
Kriptografski algoritmi se mogu realizovati hardverski (zbog brzine) ili softverski (zbog fleksibilnosti). Iako cemo se pri razmatranju uglavnom držati algoritama i protokola, nezavisno od njihove realizacije, možda će biti zanimljivo da nešto kažemo o kriptografskom hardveru.

Transponovanje i supstituisanje se mogu izvesti pomoću jednostavnih električnih kola. Slika 8-6(a) prikazuje uređaj, tzv. **P-kutiju** (P označava*per*- mutaciju), lcoja transponuje 8-bitni ulazni signal. Ako su 8 bitova raspoređenih na slici odozgo nadole: 01234567, rezultujući bitovi koji napuštaju ovu konkretnu P-kutiju biće: 36071245. P-kutija sa odgovarajućim unutrašnjim ožičenjem može da izvede bilo kakvo transponovanje praktično brzinom svetlosti, pošto ne obrađuje signal, već ga samo propušta. Ovakva konstrukcija sledi Kerkofov princip: napadač zna daje opšti postupak šifrovanja permutovanje bitova. Međutim, ne zna koji bit ide gde, što je ključno.

Supstituisanje se izvodi u **S-kutijama,** slika 8-6(b). U ovom primeru, u kutiju ulazi 3-bitni osnovni tekst, a iz nje izlazi 3-bitni šifrovan tekst. Ulazni 3-bitni signal bira jednu od osam izlaznih linija prvog stupnja i zadaje joj vrednost 1; svim ostalim linijama zadaje vrednost 0. Drugi stupanj je P-kutija. Treći stupanj ponovo kodira izabranu ulaznu liniju u binarni signal. Uz prikazano ožičenje, ako se u sistem jedan za drugim dovede osam oktalnih brojeva 01234567, izlazna sekvenca će biti 24506713. Drugim recima, 0 je zamenjena cifrom 2, 1 cifrom 4 itd. Ponovo ističemo da se odgovarajućim ožičenjem P-kutije unutar S-kutije može postići svaka supstitucija. Šta- više, takav uređaj se može realizovati hardverski da bi se postigla velika brzina rada jer koderi i dekoderi imaju samo jednu ili dve zadrške na logičkim kolima (obe kraće od 1 ns), a vreme prolaska kroz P-kutiju može biti znatno kraće od 1 ps.

Svoju stvarnu moć ovi osnovni elementi pokazuju tek kad se međusobno povežu u **kombinovani uređaj za šifrovanje** (engl. *product cipher),* slika 8-6(c). U našem primeru, u prvom stupnju (Pj) transponuje se (znači, permutuje) 12 ulaznih linija. Teorijski bi bilo moguće da drugi stupanj bude S-kutija koja 12-bitne brojeve preslikava u drugačije 12-bitne brojeve. Međutim, takav uređaj bi morao imati 2^{12} = 4096 ukrštenih žica u svom srednjem stupnju. Ulazni signal se umesto toga razbija na četiri grupe po 3 bita i svaka gmpa se supstituiše nezavisno od ostalih. Iako takvom postupku nedostaje opštost, sasvim je moćan. Kada u kombinovani uređaj za šifrovanje uključite dovoljan broj stupnjeva, izlazni signal će postati izuzetno složena funkcija ulaznog signala.

Kombinovani uređaji za šifrovanje koji rade sa /c-bitnim ulaznim signalom i proizvode k-bitni izlazni signal, veoma su česti, a *k* tipično ima vrednost između 64 i 256. Hardverska realizacija obično ima barem 18 fizičkih stupnjeva, umesto samo sedam, kao na slici 8-6(c). Ovakvo kombinovano šifrovanje softverski se realizuje kao petlja s najmanje 8 iteracija, a u svakoj se vrši supstituisanje tipa S-kutije na podblokovima bloka podataka veličine 64 do 256 bitova, iza čega sledi permutovanje koje meša izlazne signale S-kutija. Cesto se ugrađuje i specijalno početno, odnosno završno permutovanje. U literaturi se ovakve iteracije nazivaju **rundama** (engl. *rounds).*



(a) (b) (c)

Slika 8-6. Osnovni elementi kombinovanih uređaja za šifrovanje. (a) P-kutija. (b) S-kutija. (c) Kombinacija.

8.2.1 DES - standard za šifrovanje podataka

Januara 1977. godine, Američka vlada je prihvatila kombinovano šifrovanje koje je razvio IBM kao zvaničan standard za javne informacije. Taj standard za **šifrovanje** podataka (engl. *Data Encryption Standard, DES)* široko je prihvaćen u industriji za zaštitu osetljivih proizvoda. U svom prvobitnom obliku standard nije više dovoljno bezbedan, ali njegove modifikacije jesu. Sada čemo objasniti kako DES radi.

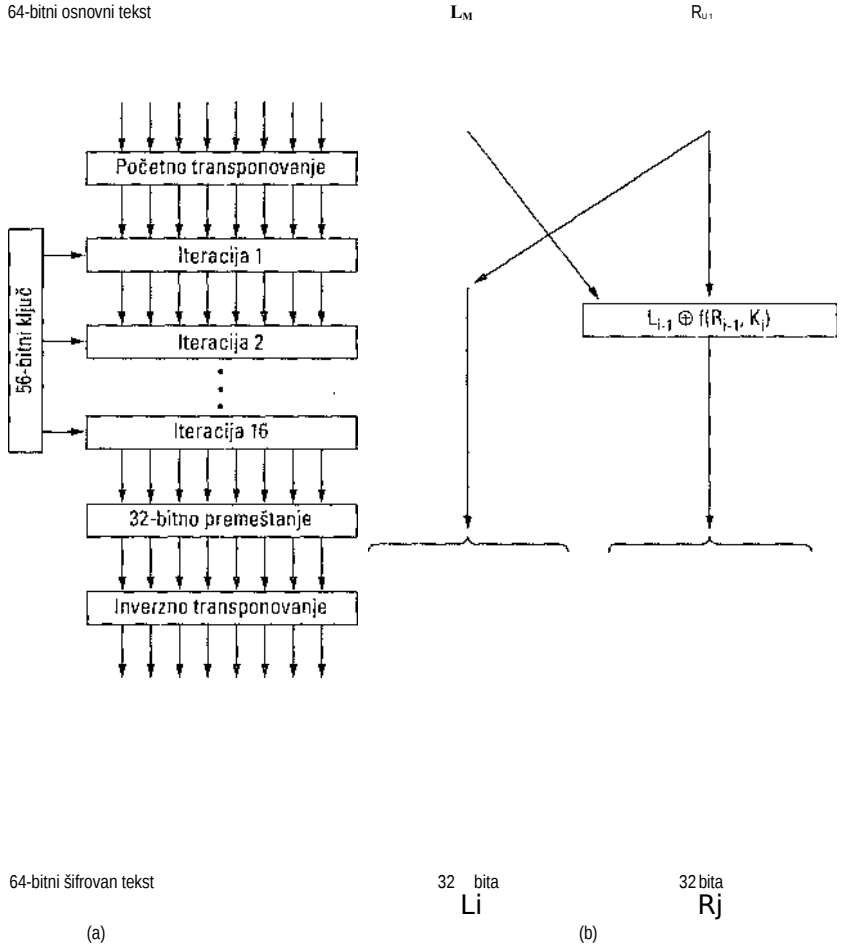
Na slici 8-7(a) DES je prikazan šematski. Osnovni tekst se šifruje u blokovima od po 64 bita, dajući 64-bitne blokove šifrovanog teksta. Algoritam, čiji parametri dobi- jaju konkretne vrednosti pomoću 56-bitnog ključa, sadrži 19 zasebnih stupnjeva. Prvi stupanj je transponovanje 64-bitnog osnovnog teksta bez upotrebe ključa. Poslednji stupanj je tačna inverzija ovog transponovanja. U pretposlednjern stupnju 32 bita na levom kraju zamenjuju mesta s 32 bita na desnom kraju. Ostalih 16 stupnjeva funkcionalno su jednaki, ali vrednosti parametara dobijaju korišćenjem različitih funkcija ključa. Algoritam je tako projektovan da se šifrovanje i dešifrovanje obavljaju istim ključem, što je bitno svojstvo svih algoritama za šifrovanje simetričnim ključem. Pri dešifrovanju se prolaze isti stupnjevi obrnutim redom.

Rad jednog od ovih međustupnjeva prikazanje na slici 8-7(b). U svaki stupanj ulaze dva 32-bitna signala i iz njega izlaze, talcođe, dva 32-bitna signala. Levi izlazni signal je kopija desnog ulaznog signala. Desni izlazni signal je rezultat isključive dis- junkcije po bitovima levog ulaznog signala i funkcije desnog ulaznog signala i ključa za taj stupanj, *Kj.* Sva složenost zavisi od ove funkcije.

Funkcija ima četiri uzastopna koraka. Prvo se konstruiše 48-bitni broj *E* razvijanjem 32-bitnog signala j prema fiksnim pravilima transponovanja i udvajanja. Drugo, *E* i *K,* se podvrgavaju isključivoj disjunkciji. Dobijeni rezultat se razvrstava u 8 grupa po 6 bitova i svaka grupa uvodi u dragu S-kutiju. Svaki od 64 moguća ulaza u S-kutiju preslikava se na 4-bitni izlaz. Konačno se ovih 8x4 bitova propuštaju kroz P-kutiju.

U svakoj od 16 iteracija koristi se dragi ključ. Pre nego što algoritam počne da se

izvršava, na Idjuč se primenjuje 56-bitno transponovanje. Neposredno pre svake ite- racije ključ se deli u dve 28-bitne jedinice i svaka se rotira ulevo za broj bitova koji zavisi od rednog broja iteracije. *Kj* se izvodi iz ovakvog rotiranog ključa još jednim 56-bitnim transponovanjem. U svakoj rundi se izvlači i permutuje drugačiji 48-bitni podskup 56-bitnog broja.



Slika 8-7. Standard za šifrovanje podataka, (a) Opšta šema. (b) Detalj jedne iteracije. Zaokruženi krstić označava isključivu disjunkciju.

Tehnika koja se ponekada koristi za ojačanje DES-a naziva se **„izbeljivanje“** (engl. *whitening).* Ona obuhvata isključivu disjunkciju proizvoljnog 64-bitnog ključa sa svakim blokom osnovnog teksta pre nego što uđe u DES i isključivu disjunkciju drugog 64-bitnog ključa s rezultujućim šifrovanim tekstom pre slanja. Izbeljivanje se lako može poništiti kada se ove operacije izvedu obrnutim redom (ako primalac ima dva ključa za izbeljivanje). Pošto ova tehnika u stvari produžava ključ, njegovo pogađanje duže traje. Imajte na umu da se isti ključ za izbeljivanje koristi za svaki blok (postoji samo jedan ključ za izbeljivanje).

Od kada je lansiran DES, ne prestaje rasprava o njemu. On je zasnovan na algoritmu za šifrovanje Lucifer koji je razvio i patentirao IBM, osim što Lucifer ne koristi 56-bitni, već 128-bitni ključ. Kada je Američka savezna vlada poželela da standardi- zuje jedan algoritam za šifrovanje javnih podataka, ona je „pozvala“ IBM da „raspravi“ stvar sa njenom desnom rukom - Američkom agencijom za bezbednost (NSA) lcoja zapošljava najviše matematičara i kriptologa na svetu. Agencija NSA je tako tajanstvena, da industrijskim krugovima kruži sledeći vic:

Pitanje: Šta znači NSA?

Odgovor: NepoStojeća Agencija.

U stvari, NSA označava Nacionalnu agenciju za bezbednost (engl. *National Security Agency)*.

Pošto su stvar „raspravili“, IBM je skratio ključ sa 128 bitova na 56 i odlučio da ne objavi postupak projektovanja algoritma DES. Mnogi podozrevaju da je ključ skraćen zato da bi samo NSA mogla da razbije DES, ali ne i organizacije s tanjim budžetom. Insistiranje na tajnosti projekta navodno treba da prikrije „mala vrata“ koja agenciji NSA još više olakšavaju razbijanje DES-a. Kada je jedan pripadnik NSA diskretno savetovao institutu IEEE da otkaže planirani skup o kriptografiji, niko se nije osećao prijatno. NSA je sve porekla.

Godine 1977, dva kriptografa sa Stanforda, Diffie i Hellman (1977), projektovali su mašinu za razbijanje DES-a i procenili da bi koštala 20 miliona dolara. Uz komadić osnovnog teksta i odgovarajućeg šifrovanog teksta, mašina bi mogla da pronađe ključ isctpnim pretraživanjem 2^{56} kombinacija u roku od jednog dana. Danas bi takva mašina koštala znatno manje od milion dolara.

Trostruki DES

IBM je još 1979. shvatio daje ključ za algoritam DES prekratak i smislio je način da ga efektivno produži trostrukim šifrovanjem (Tuchman, 1979). Primenjena metoda, koja je u međuvremenu uključena u Međunarodni standard 8732, prikazana je na slici 8-8. Tu se koriste dva ključa i tri stupnja. U prvom stupnju se osnovni tekst šifru- je na uobičajeni način algoritmom DES uz ključ K). U drugom stupnju, DES se izvršava u režimu dešifrovanja, uz ključ K_2. U trećem stupnju se ponavlja DES šifrovanje, uz ključ K\.

(a) (b)

Slika 8-8. (a) Trostruko šifrovanje algoritmom DES. (b) Dešifrovanje.

Ovakav dizajn pokreće dva pitanja. Prvo, zašto se koriste dva ključa, a ne tri? Drago, zašto se koristi **sistem EDE** (engl. *Encrypt Decrypt Encrypt* - šifrovanje dešifrovanje šifrovanje) umesto **sistema** EEE (engl. *Encrypt Encrypt Encrypt* - šifrovanje šifrovanje šifrovanje)? Umesto tri, koriste se dva ključa zato što i najluđi kriptograf veruje daje 112 bitova dovoljno za današnje rutinske komercijalne primene. (Među kriptografima, paranoja je vrlina, a ne bolest.) Proširenje ključa na 168 bitova samo bi otežalo obradu i prenošenje podataka, a korist bi bila zanemarljiva.

Sistem šifrovanja, dešifrovanja i ponovnog šifrovanja izabran je zbog kompatibilnosti s postojećim DES sistemima s jednim ključem. Funkcije za šifrovanje i dešifrovanje su preslikavanja između skupova 64-bitnih brojeva. S gledišta kriptografa, oba preslikavanja su iste vrednosti. Kada umesto sistema EEE koristi sistem EDE, raču- nar koji komunicira uz trostruko šifrovanje može da uspostavi vezu i s računarom koji koristi jednostruko šifrovanje tako što će jednostavno izjednačiti i K_2. To svojstvo omogućava da se trostruko šifrovanje uvodi postepeno, što ne zanima akademske krugove, ali je važno IBM-u i njegovim mušterijama.

8.2.2 AES - napredni standard za šifrovanje

Kako se DES primicao kraju svog korisnog veka, čak i uz trostruko šifrovanje, **Nacionalni institut za standarde i tehnologiju** (engl. *National Institute of Standards and Technology, NIST),* agencija Američkog ministarstva trgovine koja odobrava standarde u ime vlade SAD, odlučila je daje vladi neophodan nov kriptografski standard za javnu upotrebu. NIST je bio svestan nedoumica oko primene DES-a i dobro je znao da će - ako samo objavi nov standard - svako ko se pomalo razume u kripto- grafiju odmah pretpostaviti daje NSA u njega ugradila „mala vrata“ da bi mogla da čita sve što je tim standardom šifrovano. U takvim okolnostima niko ne bi koristio standard i on bi najverovatnije tiho nestao sa scene

Zbog toga se NIST opredelio za pristup, iznenađujuće neobičan za vladine institucije: organizovao je kriptografslco nadmetanje. Januara 1997, upućen je poziv istraživačima širom sveta da podnesu predloge za nov standard koji bi se zvao **Napredni standard za šifrovanje** (engl. *Advanced. Encryption Standard, AES).* Pravila konkursa bila su sledeća:

1. Algoritam mora raditi kao simetrična blok-šifra.
2. Ceo projekat mora biti javan.
3. Moraju se podržati ključevi dužine 128, 192 i 256 bitova.
4. Treba predvideti i softversku i hardversku realizaciju.
5. Algoritam mora biti javan ili se licencirati bez uslovljavanja.

Podneto je petnaest ozbiljnih predloga koji su prikazani na javnim skupovima, a prisutni su ohrabrivani da im traže slabe tačke. Avgusta 1998, NIST je odabrao pet finalista rukovodeći se uglavnom razlozima kao što su bezbednost, efikasnost, jednostavnost, fleksibilnost i memorijski zahtevi (što je važno za ugrađene sisteme). Održano je još konferencija i prikupljeno još kritičkih mišljenja. Na poslednjoj konferenciji organizovano je neobavezujuće glasanje i dobijena je sledeća lista finalista:

1. Rijndael (Joana Daemena i Vincenta Rijmena, 86 glasova).
2. Serpent (Rossa Andersona, Eli Biham i Larsa Knudsena, 59 glasova).
3. Twofish (tima Brucea Schneiera, 31 glas).
4. RC6 (iz laboratorija RSA, 23 glasa).
5. MARS (iz IBM-a, 13 glasova).

Oktobra 2000, NIST je objavio da je i on glasao za Rijndael, a novembra 2001. Rijndael je postao standard Američke vlade pod imenom Federal Information Processing Standard FIPS 197. Zbog izuzetne javnosti nadmetanja, tehničkih osobina Rijn- daela i činjenice da su ga napravila dva mlada Belgijanca (koji baš neće tek tako ugraditi mala vrata samo da bi zadovoljili NSA), očekuje se da će Rijndael dominirati na sceni kriptografskih standarda barem desetak godina. Ime Rijndael (čita se, otprilike: *rajndol)* izvedeno je od imena autora: Rijmen + Daemen.

Rijndael podržava ključeve i blokove veličina 128 do 256 bitova u koracima po 32 bita. Dužina ključa i dužina bloka mogu se birati nezavisno. Međutim, AES nalaže da veličina bloka mora biti 128 bitova, a da dužine ključa moraju biti 128, 192 ili 256 bitova. Nije verovatno da će ilco koristiti ključ dužine 192 bita, pa AES *de facto* postoji u dve varijante: 128-bitni blok sa 128-bitnim ključem i 128-bitni blok sa 256-bitnim ključem.

U našem razmatranju ćemo se ograničiti samo na slučaj 128/128 jer će on najvero- vatnije postati norma za komercijalnu upotrebu. Dužina ključa od 128 bitova omogućava $2^{128} = 3$ x

IO^{38} različitih ključeva. Ako bi NS A izgradila mašinu s milijardom paralelnih procesora, od kojih bi svaki mogao da za jednu pikosekundu proveri jedan ključ, i tada bi joj trebalo oko IO^{10} godina da proveri sve ključeve. Do tada bi se naše Sunce već ugasilo, a preostali ljudi bi rezultate morali da čitaju uz svecu.

Rijnđael

S matematičkog stanovišta, Rijnđael se zasniva na teoriji polja koju je postavio Galoa (Galois), što mu daje izvesna proverljiva bezbednosna svojstva. Međutim, algoritam se može posmatrati i kao C kod, bez zalaženja u matematiku.

Slično DES-u, i Rijnđael koristi supstituisanje i permutovanje, a i veći broj rundi. Broj rundi zavisi od veličina ključa i bloka, počev od 10 za 128-bitne ključeve i 128-bitne blokove, pa do 14 za najduži ključ ili najveći blok. Međutim, za razliku od DES-a, u svim operacijama se radi s celim bajtovima da bi se omogućilo i hardversko i softversko realizovanje algoritma. Pregled koda je prikazan na slici 8-9.

```
#define LENGTH 16                               /* broj bajtova u bloku   podataka ili ključu 7
#define NROWS 4                                 /* broj redova u stanju   7
#define NCOLS 4                                 /* broj kolona u stanju   7
#define ROUNDS 10                               /* broj Iteracija 7
typedef unsigned char byte;                     /*    neoznačen 8-bitni ceo broj7
rijndael(byte plaintextjLENGTH], byte ciphertextjLENGTH], byte keyjLENGTH])
{
int r;                                          /*    brojač petlje 7
byte state[NROWS][NCOLS];                       /*    tekuće stanje 7
struct {byte k[NROWS][NCOLS];} rk[ROUNDS + 1]; /* ključevi za pojedine runde 7
expand_key(key, rk);                            /*    konstruiši ključeve za runde 7
copyjolaintext_to_state(state, plaintext);      /* inicijalizuj tekuće stanje 7
xor_roundkey_into_state(state, rkUGLASTEO); /* izvrši isključivu disjunkciju      ključa i stanja7
tor (r = 1; r<= ROUNDS; r++) {
substitute(state);                              /*    primeni S-kutiju na svaki bajt 7
rotate_rows(state);                             /*    rotiraj red i za i bajtova 7
if (r < ROUNDS) mix_columns(state);             /*    funkcija mešanja 7
xor_joundkeyJnto_state(state, rkjrj);           /*    izvrši isključivu disjunkciju  ključa i stanja7
}
copy_state_to_ciphertext(ciphertext, state); /* vrati rezultat 7
```

Slika 8-9. Pregled koda Rijndaela.

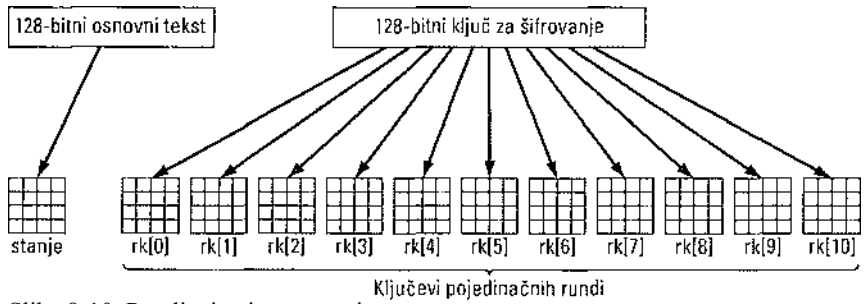
Funkcija *rijnđael* ima tri parametra. To su *plaintext*, niz od 16 bajtova koji sadrži ulazne podatke (osnovni tekst), *ciphertext,* niz od 16 bajtova u koji se vraća šifrovan tekst i *key,* 16-bitni ključ. Tokom izračunavanja, tekuće stanje se čuva u nizu bajtova *state,* čija je veličina *NROWS _ NCOLS.* Za 128-bitne blokove, ovaj niz je veličine 4x4 bajta. U 16 bajtova se može uskladištiti čitav 128-bitni blok podataka.

Niz *state* se inicijalizuje osnovnim tekstom i menja pri svakom koraku izračunavanja. U nekim koracima izvodi se supstituisanje bajt za bajt. U nekim drugim, samo se permutuje niz. Koriste se i drage transformacije. Na kraju se sadržaj niza *state* vrača kao šifrovan tekst.

Kod počinje tako što se ključ razvija uli nizova iste veličine kao i stanje. Oni se čuvaju u *rk*, koji predstavlja niz struktura, od kojih svaka sadrži niz sa stanjem. Jedna od njih se koristi na početku izračunavanja, a ostalih 10 - po jedna u svakoj rundi. Iz- računvanje ključa runde iz ključa za šifrovanje suviše je složeno da bismo ga ovde objašnjavali. Bide dovoljno ako kažemo da se ključ runde dobija ponovljenim rotacijama i isključivim disjunkcijama

različitih grupa bitova u ključu za šifrovanje. Koga zanimaju detalji, neka pogleda rad Daemena i Rijmena (2002).

Sledeci korak je kopiranje osnovnog teksta u niz *state* da bi mogao biti obrađen u rundama. On se kopira u kolone: prva četiri bajta idu u kolonu 0, draga četiri u kolonu 1 itd. Redovi i kolone stanja se numerišu počev od 0, iako se runde numerišu počev od 1. Ovo inicijalizovanje 12 nizova veličine 4x4 bajta prikazano je na slici 8-10.



Slika 8-10. Pravljenje nizova *state* i *rk.*

Postoji još jedan korak pre početka izračunavanja: r/dJGLASTEO se podvrgava isključivoj disjunkciji sa stanjem (bajt po bajt) i rezultat smešta u niz *state.* Dragim recima, svaki od 16 bajtova niza *state* zamenjuje se rezultatom isključive disjunkcije između samog sebe i odgovarajućeg bajta u nizu r/cUGLASTEO.

Sada je vreme za glavnu predstavu. Petlja izvršava 10 iteracija, po jednu u svakoj rundi, pri čemu se niz *state* menja 10 puta. Svaka runda se izvršava u četiri koraka. U koraku 1 stanje se supstituiše bajt za bajt. Svaki bajt redom postaje indeks za S-kutiju koja mu menja vrednost svojom odrednicom tog indeksa. Ovo šifrovanje je potpuno klasična zamena slova slovom. Za razliku od DES-a, koji ima više S-kutija, Rijndael ima samo jednu.

U koraku 2, svaki od četiri reda se rotira ulevo. Red 0 se rotira 0 bajtova (tj. ne po- mera se), red 1 se rotira 1 bajt, red 2 se rotira 2 bajta, a red 3 se rotira 3 bajta. Time se tekući sadržaj rasipa po bloku, slično rezultatu permutovanja sa slike 8-6.

U koraku 3, svaka kolona se meša nezavisno od ostalih. Mešanje se vrši množenjem matrica, pri čemu se nova kolona dobija kao proizvod stare kolone i matrice konstanti, a za množenje se koristi konačno Galoa polje $GF(2^8)$. Iako sve zvuči komplikovano, postoji algoritam koji izračunava svaki element nove kolone samo pomoću dva pretraživanja tabela i tri isključive disjunkcije (Daemen i Rijmen, 2002, Dodatak E).

Na kraju, u koraku 4, ključ runde se posle isključive disjunkcije sa stanjem smešta u niz ***State.***

Pošto je svaki korale reverzibilan, dešifrovanje se može izvesti jednostavnim izvršavanjem algoritma unazad. Međutim, postoji i trik kojim se dešifrovanje može izvesti i direktnim izvršavanjem algoritma za šifrovanje, ali uz drugačije tabele.

Algoritam je projelctovan s namerom da bude ne samo bezbedan, već i brz. Kada se dobro realizuje na računam od 2 GHz, trebalo bi da dostigne brzinu šifrovanja 700 Mb/s, što je dovoljno za šifrovanje preko 100 MPEG-2 video sekvenci u realnom vremenu. Hardverske realizacije rade još brže.

8.2.3 Režimi šifrovanja

Uprkos svoj složenosti, AES, DES, odnosno bilo koja blok-šifra, predstavljaju u osnovi zamenu slova slovom, pri čemu se koriste 128-bitni znakovi za AES i 64-bitni za DES. Kad god naiđe isti blok osnovnog teksta, algoritam daje isti blok šifrovanog teksta. Ako osnovni tekst *abcdefgh* šifrujete 100 puta istim DES ključem, dobićete 100 puta isti šifrovan tekst. Potencijalni uljez može da iskoriti ovo svojstvo da bi lakše upao u sistem.

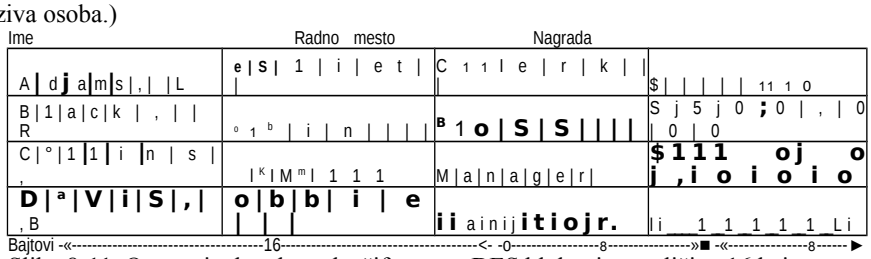
Šifrovanje uz elektronsku knjigu šifara

Da biste videli kako se ovo svojstvo proste zamene slova slovom može iskoristiti za delimično razbijanje šifre, upotrebicemo (trostruki) DES jer je lakše raditi sa 64-bit- nim, nego sa 128-bitnim blokovima, ali je sve suštinski isto i u sistemu AES. Dugačak osnovni tekst normalno se šifruje algoritmom DES tako što se izdeli u 8-bajtne (64-bit- ne) blokove koji se jedan za drugim šifruju istim ključem. Poslednji blok osnovnog teksta se, ako treba, dopuni do 64 bita. Ta tehnika se zove **šifrovanje uz elektronsku knjigu šifara** (engl. *Electronic Code Book mode)* ili jednostavno **ECB režim,** da bi se naglasila sličnost sa nekadašnjim knjigama šifara u kojima je bila popisana svaka reč osnovnog teksta, zajedno s odgovarajućim rezultatima šifrovanja (obično petocifre- nim decimalnim brojevima).

Na slici 8-11 imamo početak datoteke sa spiskom nagrada koje je kompanija odlučila da na kraju godine podeli svojim zaposlenima. Datoteka obuhvata uzastopne 32-bajtne zapise, po jedan za svakog zaposlenog, u prikazanom formatu: 16 bajtova za ime, 8 bajtova za radno mesto i 8 bajtova za nagradu. Svaki od šesnaest 8-bajtnih blokova (numerisanih od 0 do 15) šifruje se trostrukim DES-om.

Leslie je upravo imala nesporazum sa šefom i ne nada se nagradi. Nasuprot tome, Kirnje šefova miljenica i to svi znaju. Lesli može da pristupi datoteci tek pošto bude šifrovana, ali pre nego što bude poslata banci. Može li Lesli da ispravi ovu pomalo ne- fer situaciju, ako ima pristup jedino šifrovanoj datoteci?

Bez ikakvih problema. Lesli treba samo da napravi kopiju 12. bloka šifrovanog teksta (koji sadrži nagradu za Kim) i da njom zameni 4. blok (koji sadrži njenu „nagradu“). Čak i ako ne zna sadržaj 12. bloka, Lesli se ove godine može nadati mnogo veselijem Božiću. (Može se kopirati i 8. blok, ali je tu veća mogućnost da se podvala otkrije; uostalom, Lesli nije gramziva osoba.)



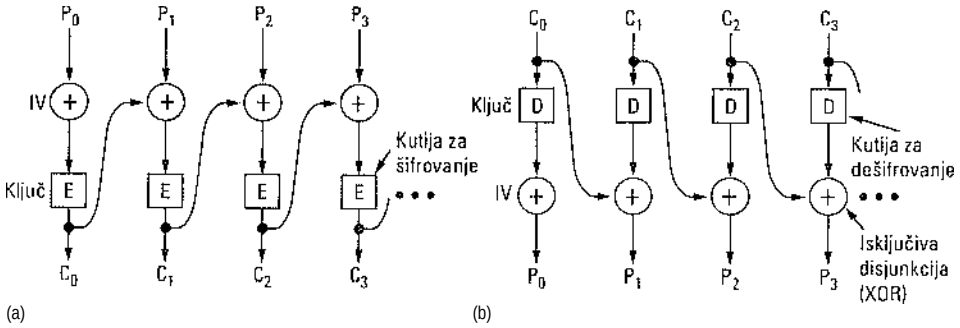
Slika 8-11. Osnovni tekst datoteke šifrovan u DES blokovima veličine 16 bajtova.

Ulančavanje blok-šifara

Za sprečavanje ovakve vrste napada, sve blok-šifre se mogu nadovezati na različite načine tako da se pri dešifrovanju bloka koji je Lesli zamenila pojavi samo neko smeće. Jedan način povezivanja je **ulančavanje blok-šifara** (engl. *cipher block chai- ning).* Prema ovoj metodi, prikazanoj na slici 8-12, svaki blok osnovnog teksta podvrgava se pre šifrovanja isključivoj disjunkciji s prethodnim blokom šifrovanog teksta. Zbog toga se isti blok osnovnog teksta ne preslikava više u isti blok šifrovanog teksta, a šifrovanje prestaje da bude samo velika zamena

slova slovom. Prvi blok osnovnog teksta se podvrgava isključivoj disjunkciji s nasumično odabranim **inicijalizacionim vektorom** (engl. *Initialization Vector, IV),* koji se šalje zajedno s osnovnim tekstom.

Možemo da posmatramo kako radi šifrovanje u režimu ulančanih blok-šifara na primeru, prikazanom na slici 8-12. Počinjemo tako što izračunavamo $C_0 = E(P_0$ XOR $IV)$. Zatim izračunavamo Cj = $E(P$ j XOR $C_0)$ itd. I pri dešifrovanju se koristi isključiva disjunkcija (XOR) da bi se proces obrnuo, uz $P_0 = IVXOR\ D(C_0)$ itd. Obratite pažnju na to da način šifrovanja bloka *i* zavisi od osnovnog teksta sadržanog u svim blokovima od 0 do *i* - 1, tako da se od istog osnovnog teksta, u zavisnosti od toga gde se nalazi, dobija različito šifrovan tekst. Transformacija koju je sprovela Lesli iza- zvaće besmislen sadržaj u dva bloka počev od bloka s njenom nagradom. Nekom brižljivom službeniku zaduženom za bezbednost sistema to ce možda ukazati gde da započne istragu.

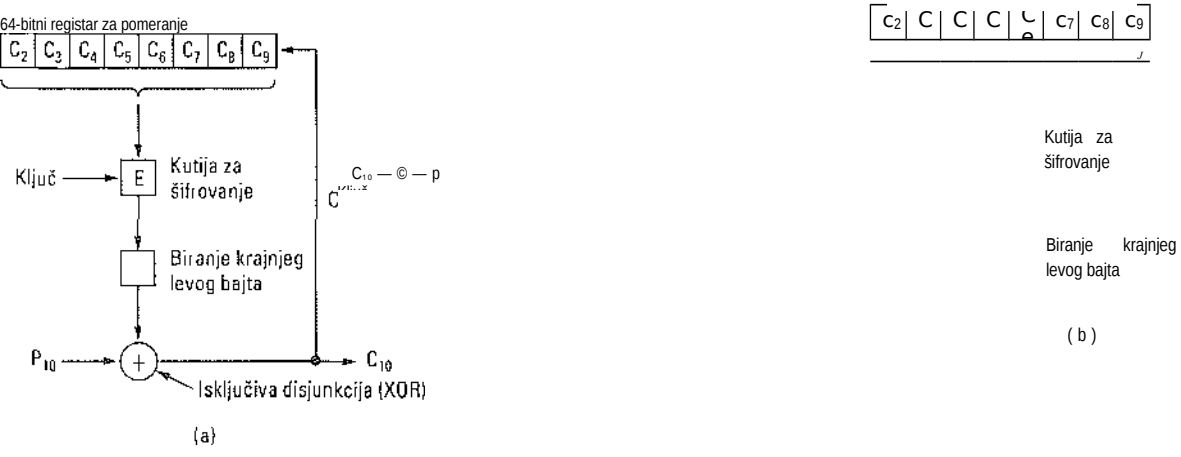


Ulančavanje blok-šifara ima i tu prednost da isti blokovi osnovnog teksta ne daju iste blokove šifrovanog teksta, što otežava kriptoanalizu. To je u stvari glavni razlog zbog koga se koristi.

Šifrovanje s povratnom spregom

Međutim, ulančavanje blok-šifara nije uvek zgodno jer je potrebno da stigne čitav 64-bitni blok podataka da bi se moglo početi s dešifrovanjem. To je, na primer, slučaj na interaktivnim terminalima, gde korisnici unesu red kraci od osam znakova, a zatim čekaju odgovor. Za šifrovanje bajt po bajt koristi se trostruki DES u režimu šifro- vanja s povratnom spregom (engl. *cipher feedback mode),* što je prikazano na slici 8-13. Tehnika za AES je potpuno ista, samo se koristi 128-bitni registar za pomeranje. Na slici je prikazano stanje mašine za šifrovanje nakon što su bajtovi od 0 do 9 šifro- vani i poslati. Kada pristigne bajt 10, slika 8-13(a), algoritam DES koristi 64-bitni registar za pomeranje da bi generisao 64-bitni šifrovan tekst. Krajnji levi bajt tog teksta izvlači se i podvrgava isključivoj disjunkciji sa P_{10}. Taj bajt se upućuje na prenosnu liniju. Osim toga, sadržaj registra se pomera ulevo za 8 bitova, zbog čega bajt C_2 ispada na levom kraju, a Cjq uskače na mesto koje mu je na levom kraju upravo oslobodio C_9. Obratite pažnju na to da sadržaj registra za pomeranje zavisi od cele prethodne istorije osnovnog teksta, pa će isti deo osnovnog teksta koji se ponavlja više puta svaki put biti drugačije šifrovan. Kao kod ulančavanja blok-šifara, i ovde je potreban ini- cijalizacioni vektor.

Dešifrovanje u režimu s povratnom spregom odvija se kao i šifrovanje. Drugim recima, sadržaj registra za pomeranje se ne *dešifruje,* već ponovo *šifruje,* tako da je bajt koji se podvrgava isključivoj disjunkciji s bajtom C_{10} da bi se dobio bajt P_{10} isti onaj bajt koji je ranije bio podvrgnut isključivoj disjunkciji s bajtom P_{10} da bi dao bajt C]0. Sve dok su dva registra za pomeranje jednaka, dešifrovanje ide bez problema. To je prikazano na slici 8-13(b).



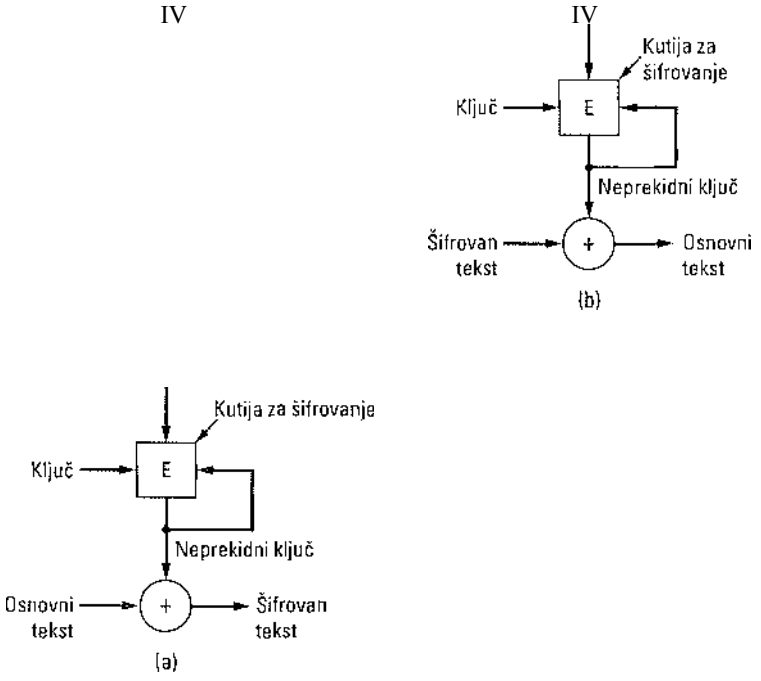
Slika **8-12. Ulančavanje blok-šifara. (a) Šifrovanje. (b) Dešifrovanje.**

Režim šifrovanja s povratnom spregom izuzetno je osetljiv na greške u prenosu. Ako jedan bit šifrovanog teksta u prenosu slučajno promeni vrednost, on đe poremetiti 8 bajtova koji se tokom dešifrovanja nalaze zajedno s njim u registru. Kada neispravan bajt jednom ispadne iz registra, ponovo će se generisati ispravan osnovni tekst. Prema tome, efekat jednog neispravnog bita je na neki način lokalizovan - on ne remeti ostatak poruke, već samo onoliko bitova koliko staje u registar.

Režim uzastopnog šifrovanja

Pa ipak, postoje primene gde se remećenje 64 bita zbog jednog neispravnog bita ne može dopustiti. Za takve primene postoji **režim uzastopnog šifrovanja** (engl. *stream cipher mode),* U njemu se prvo ključem šifruje inicijalizacioni vektor, pri čemu se do- bija jedan izlazni blok. Taj izlazni blok se ponovo šifruje ključem da bi se dobio drugi izlazni blok, koji se ponovo šifruje ključem itd. Sekvenca izlaznih blokova (proizvoljne dužine), tzv. **neprekidni ključ** (engl. *keystream*), koristi se za jednokratnu zaštitu i podvrgava isključivoj disjunkciji sa osnovnim tekstom dajući šifrovan tekst, slika 8-14(a). Imajte na umu da se inicijalizacioni vektor koristi samo u prvom koraku, posle čega se koristi šifrovan blok. Obratite pažnju i na to da neprekidni ključ ne zavisi od podataka, tako da se - ako je potrebno - može izračunati unapred i potpuno je neosetljiv na greške u prenosu. Dešifrovanje je prikazano na slici 8-14(b).

Dešifrovanje se obavlja tako što se kod primaoca generiše isti neprekidni ključ. Pošto neprekidni ključ zavisi samo od inicijalizacionog vektora i ključa, na njega ne utiču greške u prenosu šifrovanog teksta. Na taj način, jednobitna greška u prenosu šifrovanog teksta proizvodi samo jednobitnu gešku u dešifrovanom osnovnom tekstu.



Slika 8-14. Uzastopno šifrovanje, (a) Šifrovanje. (b) Dešifrovanje,

Nikada ne treba dvaput koristiti isti par (ključ, inicijalizacioni vektor) jer će se tako uvek generisati isti neprekidni ključ. Ako to učinite, izlažete se riziku **napada zbog ponavljanja iste šifre** (engl. *keystream reuse attack).* Zamislite da se blok osnovnog teksta P_0 šifruje neprekidnim ključem da bi se dobio blok P_0 XOR Kq. Kasnije se drugi blok osnovnog teksta <2o šifruje istim neprekidnim ključem i dobija se Qq XOR Kq. Uljez koji ulovi oba šifrovana bloka može jednostavno da ih međusobno podvrgne isključivoj disjunkciji i da dobije P_0 XOR Q_0, čime uklanja ključ. Uljez sada ima rezultat isključive disjunkcije dva bloka osnovnog teksta. Ako zna jedan od

njih ili može da ga pogodi, lako može doći i do drugog. U svakom slučaju, rezultat isključive disjunkcije dva toka osnovnog teksta može se napasti korišđenjem statističkih svojstava poruke. Na primer, u engleskom tekstu, najčešći znak u toku verovatno će biti rezultat isključive disjunkcije dva razmaka, zatim razmaka i slova „e“ itd. Ukratko, kada ima rezultat isključive disjunkcije dva osnovna teksta, kriptoanalitičar ima odlične šanse da ih oba dešifraje.

Brojački režim šifrovanja

Nedostatak svih režima šifrovanja, osim šifrovanja uz elektronsku knjigu šifara, jeste nemogućnost nasumičnog pristupanja šifrovanim podacima. Pretpostavimo, na primer, da se datoteka prenosi mrežom, a zatim skladišti na disku u šifrovanom obliku. To može da bude mudar postupak ako je prijemni računar prenosiv, što znači da neko može da ga ukrade. Kada kritične datoteke skladištite u šifrovanom obliku, šteta od curenja poverljivih informacija biće mnogo manja ako računar dođe u pogrešne ruke.

Međutim, datotekama na disku često se pristupa preko reda, naročito datotekama baza podataka. Ako je datoteka šifrovana ulančavanjem blok-šifara, za pristupanje proizvoljnom bloku podataka potrebno je dešifrovanje svih blokova pre njega, što baš nije jeftino rešenje. Zbog toga je smišljen tzv. brojački režim šifrovanja (engl. *coun- termode),* koji je prikazan na slici 8-15. Ovde se osnovni tekst ne šifmje direktno, već se šifruje zbir inicijalizacionog vektora i određene konstante, a rezultat šifrovanja podvrgava isključivoj disjunkciji sa osnovnim tekstom. Ako za šifrovanje svakog sle- dećeg bloka osnovnog teksta uvećamo inicijalizacioni vektor za 1, taj blok je lako dešifrovati ma gde se nalazio u datoteci, pri čemu ne moraju da se dešifruju svi njegovi prethodnici.

C,

Slika 8-15. Brojački režim C_2 šifrovanja.

Iako brojački režim rešava navedeni problem, on ima slabu tačku koju treba istaći. Pretpostavimo da ste ponovo primenili jednom već korišćen ključ K (na drugačiji osnovni tekst, ali uz isti inicijalizacioni vektor) i daje napadač oba puta ulovio šifro- vani tekst. Neprekidni ključ je u oba slučaja isti, što vas, kao što smo već videli, izlaže napadu zbog ponavljanja šifre. Kriptoanalitičar treba samo da dva šifrovana teksta međusobno podvrgne isključivoj disjunkciji i da tako s njih ukloni zaštitu, a zatim da s rezutatom isključive disjunkcije dva osnovna teksta radi šta mu je volja. Navedena slaba tačka ne znači da je šifrovanje u brojačkom režimu loš postupak, već samo da ključeve i inicijalizacione vektore

treba birati nezavisno i nasumično. Čak i ako se slučajno dvaput upotrebi isti ključ, ako su inicijalizacioni vektori različiti, osnovni tekst je bezbedan.

8.2.4 Ostale šifre

DES i Rijndael su najbolji poznati algoritmi za šifrovanje simetričnim ključem. Međutim, treba redi da su smišljeni i brojni drugi simetrični algoritmi za šifrovanje, a neki od njih su ugrađeni u različite proizvode. Neki od poznatijih navedeni su na slici 8-16.

Šifra	Autor	Dužina ključa	Napomena
Blowfish	Bruce Schneier	1—448 bitova	Stara i spora
DES	IBM	56 bitova	Preslaba za današnje uslove
IDEA	Massey i Xuejia	128 bitova	Dobra, ali zaštićena patentom
RC4	Ronald Rivest	1-2048 bitova	Oprez: neke šifre su slabe
RC5	Ronald Rivest	128-256 bitova	Dobra, ali zaštićena patentom
Rijndael	Daemen i Rijmen	128-256 bitova	Najbolji izbor
Serpent	Anderson, Biham, Knudsen	128-256 bitova	Veoma otporna
Trostruki DES	IBM	168 bitova	Sledeći najbolji izbor
Twofish	Bruce Schneier	128-256 bitova	Veoma otporna; široko se koristi

Slika 8-16. Uobičajeni algoritmi za šifrovanje simetričnim ključem,

8.2.5 Kriptoanaliza

Pre nego što zaključimo temu šifrovanja simetričnim ključem, treba barem pome- nuti četiri najvažnija unapređenja kriptoanalize. Prvo je **diferencijalna kriptoanaliza** (Biham i Shamir, 1993). Tom tehnikom se može napasti bilo koja blok-šifra. Počinje se tako što se uoči par blokova osnovnog teksta koji se razlikuju samo u nekoliko bitova i pažljivo se prati šta se s njima događa u svakoj internoj iteraciji tokom šifrovanja. Neke sekvence bitova su u mnogim slučajevima češće od dragih, a to otkriće omogućava napad statističkim metodama.

Drugo unapređenje koje treba pomenuti je **linearna kriptoanaliza** (Matsui, 1994). Njom se može razbiti DES uz samo 2^{43} poznatih bitova osnovnog teksta. Radi se tako što se izvesni bitovi osnovnog i šifrovanog teksta podvrgavaju međusobnoj isključivoj disjunkciji, a zatim se u rezultatu traže njihove karakteristične sekvence. Kada se analiza ponovi više puta, jedna polovina zbirnog rezultata trebalo bi da budu nule, a draga jedinice. Šifrovanjem se često taj odnos pomera na jednu ili dragu stranu, a to odstupanje, ma kako malo, može se iskoristiti za skraćivanje vremena dešifrovanja. Detalje ćete naći u originalnom radu autora.

Treće unapređenje je usmereno na analiziranje potrošnje električne energije. Računali bit 1 obično predstavljaju naponom od 3 V, a bit 0 naponom od 0 V. Na taj način, za obradu jedinice troši se više električne energije nego za obradu nule. Ako kripto- grafski algoritam predstavlja petlju u kojoj se bitovi ključa obrađuju redom, napadač koji zameni glavni sistemski sat brzine n GHz sporijim satom (na primer, brzine 100 Hz) i priključi se na nožice za napajanje mikroprocesora (na napon i uzemljenje), može tačno da prati potrošnju

električne energije pri svakoj mašinslcoj instrukciji. Iz tih podataka se ključ može otkriti iznenađujuće lako. Protiv kriptoanalize ove vrste može se boriti samo ako se pri kodiranju algoritma za šifrovanje u asernbleru dobro povede računa o tome da potrošnja električne energije ne bude ni u kakvoj vezi ni s glavnim ključem, ni s ključevima pojedinačnih rundi.

Četvrto unapređenje je vremenska analiza. Algoritmi za šifrovanje su prepuni naredaba if pomoću kojih se proveravaju bitovi u ključevima pojednačnih rundi. Ako se naredbe then i else izvršavaju različito dugo, usporavanjem sistemskog sata i analiziranjem trajanja svkog koraka može se doći do ključeva pojedinačnih rundi. Kada se saznaju ključevi svih rundi, obično se može izračunati i glavni ključ. Analiza potrošnje električne energije i vremenska analiza mogu se i istovremeno koristiti, pa zadatak postaje lakši. Iako ove dve vrste analize izgledaju pomalo egzotično, to su veoma moćne tehnike i mogu da razbiju svaku šifru koja nema odbranu specijalno protiv njih.

8.3 ALGORITMI ZA ŠIFROVANJE JAVNIM KLJUČEM

Istorijski posmatrano, distribuiranje ključeva je oduvek bilo najslabija tačka kriptosistema. Bez obzira na neprobojnost samog sistema, ako neko uspe da ukrade ključ, sva vrata mu se otvaraju. Kriptolozi su oduvek prećutno pretpostavljali da se za šifrovanje i dešifrovanje koristi isti ključ (ili dva ključa koji se lako izvode jedan iz drugog). Ključ morate podeliti svim korisnicima sistema, pa tako ispada da sistemi šifrovanja pate od urođene unutrašnje slabosti. Iako ključeve treba zaštititi od lopova, morate ih i podeliti korisnicima - dakle, ne možete ih držati u sefu.

Diffie i Hellman (1976), dva istraživača sa Stanforda, predložili 1976. godine potpuno novu vrstu kriptosistema s različitim ključevima za šifrovanje i dešifrovanje koji se ne mogu lako izvesti jedan iz drugog. Prema njihovom predlogu, algoritam za šifrovanje E i algoritam za dešifrovanje D treba da ispune tri sledeća zahteva:

1. D(E(P)) = P.
2. *D* se izuzetno teško može izvesti iz *E*.
3. £ se ne može provaliti napadom zasnovanim na šifrovanju izabranog osnovnog teksta.

Kada se prvi zahtev iskaže recima, on znači da primenom algoritma *D* na šifrovanu poruku *E(P)* treba da dobijemo originalni osnovni tekst poruke *P*. Bez ispunjenja tog zahteva, legitimni primalac ne bi mogao da dešifruje poruku. Drugi zahtev je po sebi jasan. Treći zahtev je neophodan jer, kao što ćemo ubrzo videti, uljez može da ekspe- rimentiše sa algoritmom do mile volje. Uz ispunjenje gornjih uslova, nema razloga da ključ za šifrovanje ne postane javan.

Postupak ide otprilike ovako. Korisnik koji želi da prima tajne poruke, na primer, Alisa, prvo napravi dva algoritma koji zadovoljavaju gornje zahteve. Tada se algoritam za šifrovanje i Alisin ključ objave (odatle ime šifrovanje javnim ključem, engl. *public-key criptography)*. Alisa može, na primer, da svoj javni ključ stavi na svoju Web stranu. Taj Alisin ključ označavaćemo sa E_A i pod njim ćemo podrazume- vati algoritam za šifrovanje čiji su parametri podešeni Alisinim javnim ključem. Slično tome, sa D_A ćemo označavati (tajni) algoritam za dešifrovanje čiji su parametri podešeni Alisinim privatnim ključem. Bob će uraditi isto: objaviće E_B, ali će u tajnosti čuvati D_B.

Pokušajmo sada da rešimo problem uspostavljanja bezbednog kanala između Ali- se i

Boba, smatrajući da nikada ranije nisu stupali u međusobnu vezu. I Alisin i Bobov ključ za šifrovanje, E_A i E_B, mogu se naći u javno dostupnim datotekama. Sada Alisa piše svoju prvu poruku *P,* izračunava $E_B(P)$ i rezultat šalje Bobu. Bob ga dešifruje pri- menjujući na njega svoj tajni ključ D_B [tj, izračunava $D_B(E_B(P)) = P$], Niko dragi ne može da pročita šifrovanu poruku $E_B(P)$ zato što je sistem šifrovanja tvrd orah, a D_B se samo veoma teško može izvesti iz javnog ključa E_B, Bob piše odgovor *R* i šalje Ali- si $E_A(R)$. Alisa i Bob sada mogu da bezbedno komuniciraju.

Ovde treba reći nešto o terminologiji. Za šifrovanje javnim ključem svaki korisnik mora da ima dva ključa: javni ključ koji koristi svako ko želi da tom korisniku šalje šifrovane poruke i privatni ključ koji tom korisniku služi za dešifrovanje primljenih poruka. Te ldjučeve ćemo uvek navoditi kao *javni* i *privatni* ključ da bismo ih razlikovali od *tajnog* ključa koji se koristi za konvencionalno šifrovanje simetričnim ključem.

8.3.1 RSA

Ostaje nam još samo da pronađemo algoritme koji će stvarno zadovoljiti sva tri na vedena zahteva. Zbog potencijalnih prednosti koje bi moglo da pruži šifrovanje javnim ključem na tome vredno rade mnogi istraživači i neki algoritmi su već objavljeni. Jednu dobru metodu otkrila je grupa s Masačusetskog tehničkog instituta (Rivest i sa- radnici, 1978). Ona je poznata pod imenom RSA izvedenim iz inicijala autora (Rivest, Shamir, Adleman). Izdržala je sve pokušaje provaljivanja tokom više od četvrt stoleća i smatra se veoma otpornom. Na njoj se zasnivaju mnogi praktični detalji bezbednosti. Glavna mana joj je potreba za dugačkim ključem (barem 1024 bita, u odnosu na 128 bitova kod šifrovanja simetričnim ključem), zbog čega radi prilično sporo.

Metoda RSA zasnovana je na određenim principima teorije brojeva. U nastavku ćemo opisati kako se metoda koristi, a detalje potražite u originalnom radu. :

1. Izaberite dva velika prosta broja, *p* i *q* (obično od po 1024 bita).
2. *Izračunajte*n = p x q i z = (p - I) x (q - l).
3. Izaberite broj koji je prost u odnosu na *z* i označite ga sa *d.*
4. Pronađite takvo *e* da bude $e \times d = 1 \ mod \ z$.

Pošto unapred izračunamo ove parametre, spremni smo za samo šifrovanje. Pode- lite osnovni tekst (u obliku bitova) u blokove, tako da svaki blok (poruka *P)* padne u interval $0 < P < n$. Učinite to tako što c'ete osnovni tekst grupisati u blokove od *k* bitova, gde *k* predstavlja najveći ceo broj za koji je ispunjen uslov daje $2^k < n$.

Kada želite da šifrujete poruku *P,* izračunajte $C = P^e \ (\text{mod } n)$. Kada želite da dešifrujete C, izračunajte $P = C^l \ (\text{mod } n)$. Za svako *P* u zadatom intervalu može se dokazati da su funkcije za šifrovanje i dešifrovanje međusobno inverzne. Za šifrovanje su vam potrebni *e* i *n,* a za dešifrovanje *cl* i *n.* Prema tome, javni ključ sadrži par *(e, n),* a privatni par (*d*, *n)*.

Bezbednost metode se zasniva na problemima vezanim za razlaganje velikih brojeva na činioce. Kada bi kriptoanalitičar mogao da (javno) *n* razloži na činioce, došao bi do *p* i *q,* a odatle bi izračunao *z*. Kada ima *z* i e, do *d će* doci primenom Euklidovog algoritma. Na sreću, matematičari već preko 300 godina razlažu velike brojeve na činioce bez velikog uspeha jer je taj problem očigledno veoma težak.

Prema proceni Rivesta i saradnika, za razlaganje broja od 500 cifara na činioce primenom grube sile bilo bi potrebno IO^{25} godina, uz pretpostavku da se za to koristi najbolji poznati

algoritam i računar kod koga svaka instrukcija traje samo 1 mikrose- kundu. Čak i ako brzina računara svakih deset godina raste za red veličine, proći će stoleća pre nego što neko broj od 500 cifara rastavi na činioce, a tada će naši potomci izabrati još veće *p* i *q*.

Očigledan didaktički primer rada RSA algoritma prikazan je na slici 8-17. Za pri- mer smo izabrali $p = 3 \backslash q = 11$, što daje $n = 33$ i $z = 20$. Podesna je i vrednost d - 7, pošto 7 i 20 nemaju zajedničkih činilaca. Uz ovakav izbor, e se može naći rešavanjem jednačine $l^e = 1$ (mod 20), što daje $e = 3$. Šifrovani tekst C, koji odgovara osnovnom tekstu *P,* dobija se kao $C = P^3$ (mod 33). Primalac dešifruje šifrovan tekst koristeći pravilo $P=\acute{C}^l$ (mod 33). Na slici je, kao primer, šifrovana reč „SUZANNE“.

Osnovni tekst (P) Šifrovan tekst (C) Posle dešifrovanja

Treba naglasiti da se u našem opisu algoritam RSA koristi slično algoritmu za simetrično šifrovanje u režimu ECB - isti ulazni blok uvek daje isti izlazni blok. Prema tome, za šifrovanje podataka je potrebno nekakvo ulančavanje. U praksi, međutim, sistemi zasnovani na algoritmu RSA većinom koriste šifrovanje javnim ključem da bi distribuirali jednokratne ključeve koji se koriste u simetričnim sistemima za šifrovanje, kao što su AES ili trostruki DES. Algoritam RSA je suviše spor za stvarno šifrovanje veće količine podataka, ali se široko koristi za distribuiranje ključeva.

8.3.2 Ostali algoritmi za šifrovanje javnim ključem

Iako se algoritam RSA široko koristi, to nikako nije i jedini poznati algoritam za šifrovanje javnim ključem. Prvi algoritam za šifrovanje javnim ključem bio je algoritam „ranca“ (Merkle i Hellman, 1978). Tu se pretpostavlja da neko ima veliki broj objekata, svaki drugačije težine. Vlasnik objekata kodira poruku tako što tajno bira podskup objekata i stavlja ih u ranac. Ukupna težina objekata u rancu se objavljuje, kao i spisak svih postojećih objekata. Sadržaj ranca se drži u tajnosti. Smatra se da je - uz još neka ograničenja - otkrivanje liste objekata u rancu poznate težine nerešiv problem, pa je to dalo ideju za ovaj algoritam.

Autor algoritma, Ralph Merkle, bio je potpuno siguran da niko ne može da provali algoritam, pa je ponudio nagradu od 100 dolara onome ko to uspe. Adi Shamir (ono „S“ u algoritmu RSA) odmah je provalio algoritam i osvojio nagradu. Ne ustuknuvši, Merkle je ojačao algoritam i ponovo ponudio nagradu za njegovo razbijanje, ovoga puta od 1000 dolara. Taj algoritam odmah je provalio Ronald Rivest („R“ u RSA) i naravno, uzeo pare. Merkle se nije usudio da za razbijanje sledeće verzije algoritma ponudi nagradu od 10.000 dolara, tako da je Leonard Adleman („A“ u RSA) ostao kratkih rukava. Uprkos tome, algoritam ranca više nije smatran bezbednim pa se više i ne koristi.

Neki drugi sistemi za šifrovanje javnim ključem zasnivaju se na teškoćama vezanim za izračunavanje diskretnih logaritama. Taj princip koriste algoritmi koje su napisali E1 Gamal (1985) i Schnorr (1991).

Postoji još nekoliko drugih sistema, kao onaj koji se zasniva na eliptičkim funkcijama (Menezes i Vanstone, 1993), ali kao glavne ostaju dve kategorije algoritama koji se zasnivaju na teškoćama vezanim za razlaganje velikih brojeva na činioce i izračunavanje ostatka deljenja diskretnih logaritama velikim prostim brojevima. Ta dva problema se smatraju zaista teškim - matematičari ih rešavaju već mnogo godina bez značajnijeg uspeha.

8.4 DIGITALNI POTPISI

Autentičnost mnogih pravnih, finansijskih i drugih dokumenata određuje se na osnovu toga da li je dokument svojeručno potpisala ovlašćena osoba. Fotokopije ne važe. Pošto se u računarskim sistemima prenošenja poiuka ne mogu slati uzorci papira i mastila, neophodno je naći metodu „potpisivanja“ dokumenata koja se ne može falsifikovati.

Nije lako smisliti zamenu za svojeručni potpis. U osnovi je potreban sistem koji će jednoj strani omogućiti da pošalje poruku drugoj uz ispunjenje sledećih uslova:

1. Primalac može da proveri navodni identitet pošiljaoca.
2. Pošiljalac ne može da se ogradi od sadržaja poruke.
3. Primalac ni na koji način ne može da izmeni primljenu poruku.

Prvi uslov je, na primer, neophodan u finansijskim sistemima. Kada korisnikov računar izda bančinom računani nalog za kupovinu tone zlata, bančin računar mora imati načina da proveri da računar s koga je stigao nalog stvarno pripada kompaniji čiji ce račun kupovina teretiti. Drugim recima, banka mora da proveri identitet kupca (a kupac mora da proveri identitet banke).

Drugi uslov štiti banku od prevara. Pretpostavimo daje banka od korisnika primila nalog za kupovinu tone zlata, a neposredno zatim cena zlata naglo pada. Nezadovoljni korisnik može da tuži banku tvrdeći da nikada nije izdao nalog za kupovinu zlata. Kada banka na sudu kao dokaz podnese njegovu poruku, korisnik može da porekne daju je poslao. Svojstvo koje obezbeđuje da nijedna ugovorna strana kasnije ne može da tvrdi da ugovor nije potpisala naziva se nemogućnost poricanja (engl. *nonrepu- diation)*. Sistem digitalnog potpisivanja koji ćemo proučiti u nastavku omogućava obezbeđivanje tog svojstva.

Treći uslov štiti korisnika u situaciji kada cena zlata naglo skoči, pa banka pokuša da falsifikuje njegov potpisan nalog da glasi na šipku, a ne na tonu zlata. U tom pre- varantskom scenariju, banica bi ostatak zlata zadržala za sebe.

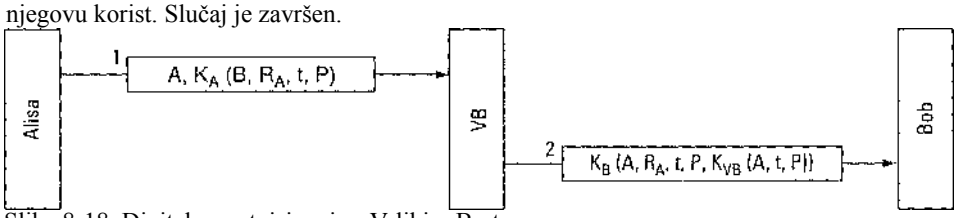
8.4.1 **Potpisivanje** simetričnim ključem

Prema jednom pristupu, digitalno potpisivanje obezbeđuje centralna ovlašcena organizacija koja poznaje svakoga i kojoj svako veruje; neka se zove, recimo, Veliki Brat (VB). Svaki korisnik zatim bira tajni ključ i lično ga nosi u *VB* kancelariju. Na taj način, Alisin tajni ključ K_A znaju samo ona i VB; za ostale korisnike važi slična šema.

Kada Alisa poželi da pošalje potpisanu poruku sa osnovnim tekstom *P* svom bankara Bobu, ona će generisati $K_A(B, R_A, t, P)$, gde je *B* Bobov identitet, R_A je broj koji je Alisa nasumično odabrala, *t* je vremenska oznaka koja treba da garantuje svežinu poruke, a $K_A(JB, R_A, t, P)$ je poruka šifrovana njenim ključem K_A. Ona zatim šalje šifrovanu poruku, kao na slici 8-18. *VB* utvrđuje daje poruka od Alise, dešifraje je i šalje Bobu. Poruka koja stiže Bobu sadrži osnovni tekst Alisine poruke, kao i potpisanu poruku $K_{VB}(A, t, P)$. Bob sada izvršava Alisin nalog.

Sta se događa ako Alisa kasnije porekne daje poslala poruku? Prvi korak je da svako tuži svakoga (barem je tako u SAD). Kada tužba dođe pred sud i Alisa sa indignacijom odbije daje Bobu poslala ikakvu poruku, sudija će pitati Boba kako može da sa sigurnošću tvrdi daje poruku dobio od Alise, a ne od Trudi. Bob će prvo istaći da *VB* ne bi prihvatio poruku od Alise da nije šifrovana ključem K_A, tako da nema mogućnosti da Trudi u Alisino ime pošalje poruku Velikom Bratu a da to odmah ne bude otkriveno.

Bob zatim dramatično podnosi sudu Dokaz A: $K_{VB}(A, t, P)$. Bob kaže da je to poruka s potpisom *VB* koja dokazuje daje Alisa poslala *P* Bobu. Sudija tada traži od *VB* (kome svi veruju) da dešifruje Dokaz A. Kada *VB* potvrdi da Bob govori istinu, sudija presuđuje u njegovu korist. Slučaj je završen.



Slika 8-18. Digitalno potpisivanje s Velikim Bratom.

Moguć problem s protokolom za potpisivanje sa slike 8-18 predstavlja Trudi, koja može da reprodukuje svaku poruku. Da bi se on ublažio, pri razmenjivanju poruka se uvek koriste vremenske oznake. Osim toga, Bob može da pregleda sve skorije poruke i da proveri da li je u bilo kojoj od njih korišćeno R_A. Ako nađe takvu poruku, odmah je odbacuje kao ponovo poslatu. Držeći se jedino vremenskih oznaka, Bob bi odbacivao samo vrlo stare poruke. Da bi se zaštitio od neposrednih napada ponovljenim slanjem poruka, Bob proverava R_A svake dolazne poruke da bi video da lije u proteklom satu primio takvu poruku od Alise. Alco takvu poruku ne nađe, Bob sa sigurnošću može da pretpostavi da je u pitanju nov nalog.

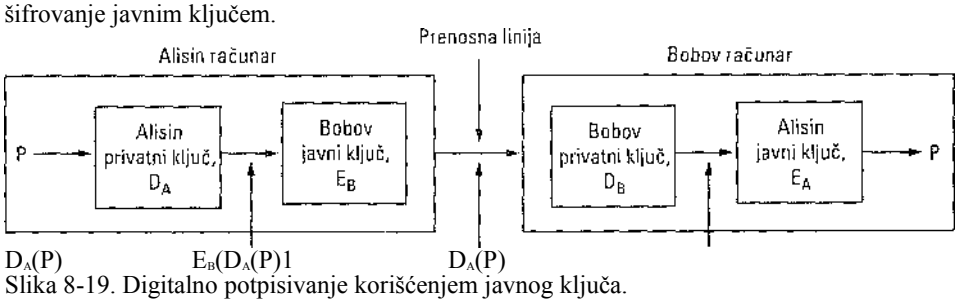
8.4.2 Potpisivanje javnim ključem

Strukturni problem primene šifrovanja simetričnim ključem na digitalno potpisivanje ogleda se u tome što svako mora da veruje Velikom Bratu. Osim toga, Veliki Brat čita sve potpisane poruke. Najlogičniji kandidati za vlasnike servera Veliki Brat bili bi vlada, banke, obračunske firme i advokatske kancelarije. Nažalost, nijedna od ovih organizacija ne pobuđuje totalno poverenje svih građana. Prema tome, bilo bi lepo kada bi se za potpisivanje elektronskih dokumenata mogla izbeći ovlašćena po- verljiva organizacija.

Na sreću, šifrovanje javnim ključem može da pruži značajan doprinos u ovoj oblasti. Pretpostvimo da algoritmi za šifrovanje i dešifrovanje imaju svojstvo da je $E(D(P)) = P$, pored uobičajenog svojstva da je $D(E(P)) = P$. (RSA ima to svojstvo, tako da pretpostavka nije nerazumna). Imajući sve to u vidu, Alisa može da uputi Bobu potpisanu poruku sa osnovnim tekstom P, šaljući mu $E_B(D_A(P))$. Obratite pažnju na to da Alisa ima svoj (privatni) ključ D_A, kao i Bobov javni ključ E_B, tako da može da sastavi navedenu poruku.

Kada Bob primi poruku, transformiše je kao i obično svojim privatnim ključem i dobija $D_A(P)$, kao na slici 8-19. Taj tekst sklanja na bezbedno mesto, a zatim na njega primenjuje E_A da bi došao do osnovnog teksta.

Da biste videli kako radi svojstvo potpisivanja, pretpostavimo da Alisa kasnije porekne da je Bobu poslala poruku P. Kad slučaj dođe pred sud, Bob će pokazati i P i $D_A(P)$. Sudija će se lako uveriti da Bob zaista ima ispravnu poruku šifrovanu ključem D_A tako što će na nju primeniti E_A. Pošto Bob ne zna Alisin privatni ključ, mogao je dobiti poruku koja je njime šifrovana samo ako ju je Alisa zaista poslala. Dok bude se- dela u zatvoru zbog lažnog svedočenja i prevare, Alisa će imati dovoljno vremena da smišlja nove algoritme za šifrovanje javnim ključem.



Slika 8-19. Digitalno potpisivanje korišćenjem javnog ključa.

Premda upotreba javnog ključa za digitalno potpisivanje predstavlja elegantno rešenje, postoje problemi koji ne potiču od samog algoritma, već od okruženja u kome se radi. Kao prvo, Bob može da dokaže da je jedino Alisa mogla poslati poruku samo ako je ključ D_A

tajan. Ako je Alisa nelcome saopštila svoj tajni ključ, taj argument više ne vredi jer je poruku mogao poslati svako, čak i sam Bob.

Problem može da nastane, na primer, ako je Bob Alisin berzanski posrednik i Alisa mu nalaže da kupi određen broj akcija. Neposredno zatim cena tih akcija naglo pada. Da bi porekla daje poslala poruku Bobu, Alisa trči u policiju i prijavljuje da joj je provaljeno u stan iz koga je, između ostalog, odnet računar s njenim tajnim ključem. Njena zakonska odgovornost zavisi od zakona koji vladaju u zemlji, naročito ako izjavi da je otkrila provalu tek kada se posle nekoliko časova vratila s posla.

Drugi problem s šemom potpisivanja nastaje kada Alisa odluči da promeni svoj ključ. S pravnog aspekta, ona to može da uradi kad god poželi, a verovatno je preporučljivo da to periodično i čini. Ako kasnije nastane gore opisani sudski proces, sudija će primeniti tekući ključ E_A na $D_A(P)$ i otkriti da ne dobija P. U tom trenutku, Bob će ispasti glup.

Za digitalno potpisivanje može se u principu iskoristiti svaki algoritam za šifro- vanje javnim ključem. RSA algoritam je, međutim, *de facto* industrijski standard. Njega koriste mnogi proizvodi namenjeni bezbednosti. Međutim, 1991. godine, NIST je predložio varijantu E1 Gamalovog algoritma za šifrovanje javnim ključem za svoj novi Standard za digitalno potpisivanje (engl. *Digital Signature Standard, DSS).* E1 Gamal nije bezbednost algoritma zasnovao na teškoći razlaganja velikih brojeva na činioce, već na teškoći izračunavanja diskretnih logaritama.

Kao i uvek kad vlada pokuša da diktira standard za šifrovanje, podigla se bura protesta. Standard DSS kritikovan je zato što je:

1. Previše tajan (protokol za korišćenje E1 Gamalovog algoritma projektovala je NS A).
2. Previše spor (provera potpisa traje 10 do 40 puta duže nego uz RSA).
3. Previše nov (E1 Gamalov algoritam još nije potpuno ispitan).
4. Nedovoljno bezbedan (fiksni 512-bitni ključ).

U revizijama koje su usledile, četvrta primedba je oslabljena time što su dozvoljeni ključevi dužine do 1024 bita. Pa ipak, prve dve primedbe još uvek važe.

8.4.3 Sažeci poruka

Jedna od kritika upućenih metodama digitalnog potpisivanja je i to da one često spajaju dve jasno razdvojene funkcije: proveru identiteta i tajnost. Često se događa da je provera identiteta neophodna, ali ne i tajnost. Isto tako, dozvola za izvoz se lakše dobija ako predmetni sistem obezbeđuje proveru identiteta bez tajnosti. U nastavku ćemo opisati šemu za proveravanje identiteta kod koje ne treba šifrovati celu poruku.

Sema se zasniva na funkciji za jednosmerno heširanje koja osnovni tekst proizvoljne dužine preračunava u niz bitova fiksne dužine. Ta funkcija za heširanje, zvana i sažetak poruke (engl. *message digest, MD),* ima četiri važna svojstva:

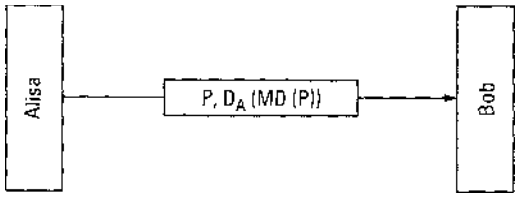
1. Za zadato P, lako se izračunava *MD(P).*
2. Za zadato *MD(P),* praktično je nemoguće naći P.
3. Za zadato P, niko ne može da izračuna takvo P' da važi *MD(P') = MD(P).*
4. Izmena ulaznih podataka čak i za samo 1 bit proizvodi veoma različit rezultat.

Za ispunjenje trećeg kriterijuma, rezultat heširanja mora biti dužine barem 128 bitova. Četvrti kriterijum biće ispunjen ako funkcija za heširanje dobro „protrese“ bitove, slično onome što smo videli kod algoritama za šifrovanje simetričnim ključem.

Dobijanje sažetka poruke iz delića osnovnog teksta mnogo je brže od šifrovanja tog istog teksta javnim ključem, pa se sažeci poruka mogu iskoristiti za ubrzavanje digitalnog potpisivanja. Da bismo videli kako to radi, osvrnimo se ponovo na protokol potpisivanja sa slike 8-18. Umesto da P potpiše sa $K_{BB}(A, t,$ P), *VB* sada izračunava sažetak poruke primenjujući *MD* na P i dobija *MD(P). VB* tada uključuje $K_{BB}\{A, t, MD(P$)) kao petu stavku liste šifrovane pomoću K_B koju šalje Bobu umesto $K_{BB}(A, t,$ P).

Ako nastane spor, Bob može da pokaže i P i $K_{BB}(A, t, MD(P))$. Pošto Veliki Brat dešifruje poruku za sud, Bob ima garantovano autentično *MD(P)* i pretpostavljeno P. Budući da Bob ne može ni na koji način da nađe još jednu poruku sa istim rezultatom heširanja, sudija će se lako uveriti da Bob govori istinu. Kada se sažetak poruke koristi na ovaj način, skraćuje se vreme šifrovanja i smanjuju troškovi prenosa poruke.

Sažimanje poralca se može uvesti i u kriptosisteme koji rade s javnim ključem (slika 8-20). Ovde prvo Alisa pravi sažetak svog osnovnog teksta. Zatim potpisuje sažetak poroke i šalje ga Bobu, zajedno sa osnovnim tekstom. Ako Trudi usput zameni P, Bob će to primetiti kada sam izračuna *MD(P').*



MD5

Predložene su mnoge funkcije za pravljenje sažetaka. Najčešće se koriste funkcije MD5 (Revest, 1992) i SHA-1 (N1ST, 1993). MD5 je peta u nizu funkcija za heširanje sažetaka koje je smislio Ronald Rivest. Ona radi tako što „melje“ bitove na način koji obezbeđuje da svaki izlazni bit zavisi od svakog ulaznog bita. Ukratko, ostatak celo- brojnog deljenja dužine poruke sa 512 prvo se dopunjava do 448 bitova. Zatim se tome pripaja prvobitna dužina poruke kao 64-bitni ceo broj, što ukupnu dužinu ulaznih podataka čini umnoškom od 512. U poslednjem koraku se 128-bitni bafer inicija- lizuje fiksnom vrednošću.

Tek sada počinje stvarno izračunavanje. U svakoj rundi se 512-bitni blok ulaznih podataka energično meša sa sadržajem 128-bitnog bafera. Radi „boljeg ukusa“, u me- šavinu se dodaju sinusne funkcije iz pethodno konstruisane tabele. Sinusne funkcije nisu izabrane zato što možda daju bolji rezultat od generatora slučajnih brojeva, već zato što su svima poznate i otklanjaju sumnju daje projektant ugradio mala vrata kojima samo on ima pristup. Setite se daje IBM-ovo odbijanje da objavi principe pro- jektovanja S-kutija u algoritmu DES izazvalo velike rasprave u pogledu postojanja malih vrata. Rivest je takve rasprave želeo da saseče u korenu. Svaki ulazni blok se obrađuje u četiri runde. Postupak traje sve dok ima ulaznih blokova. Sadržaj 128-bitnog bafera tada predstavlja sažetak poruke.

Algoritam MD5 se koristi već desetak godina i mnogi su za to vreme pokušali da ga provale. U njemu su zaista pronađene neke slabe tačke, ali su ga od provaljivanja sačuvale određene komponente njegove unutrašnje strukture. Međutim, ako te unutrašnje prepreke jednom popuste, MD5 će pasti. Pa ipak, u ovom trenutku on se još uvele čvrsto drži.

SHA-1

Druga poznatija funkcija za sažimanje poruka jeste tzv. bezbedni algoritam za heširanje 1

(engl. *Secure Hash Algorithm 1, SHA-1),* koji je razvila agencija NSA, a blagoslovio NIST u standardu FIPS 180-1. Slično algoritmu MD5, i SHA-1 obrađuje ulazne podatke u blokovima od 512 bajtova, ali za razliku od njega, generiše sažetak poruke dužine 160 bitova. Tipičan način na koji Alisa Bobu šalje javnu, ali potpisanu poruku, prikazanje na slici 8-21. Ovde se njena poruka u obliku osnovnog teksta predaje algoritmu SHA-1 da bi se dobio SHA-1 heširani, 160-bitni rezultat. Taj rezultat Alisa potpisuje svojim privatnim RSA ključem i šalje ga Bobu, zajedno sa osnovnim tekstom poruke.

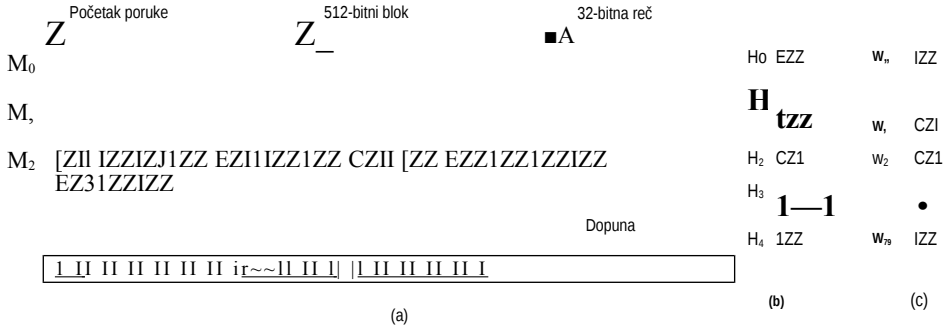
Kada primi poruku, Bob i sam hešira osnovni tekst, i istovremeno primenjuje Ali- sin javni ključ na potpisani rezultat njenog heširanja da bi ga „oslobodio“ potpisa. Ako se heširani rezultat koji je poslala Alisa (*H)* slaže s rezultatom heširanja koji je on izračunao, poruku smatra punovažnom. Pošto Trudi nikako ne može da u prenosu tako izmeni osnovni tekst da on heširanjem ponovo da rezultat *H,* Bob lako otkriva pokušaj falsifikovanja. Šema sa slike 8-21 često se koristi za poruke s javnim sadržajem čiji je integritet važan. Uz vrlo male troškove obrade, ona s visokom verovatno- ćom garantuje otkrivanje svakog pokušaja falsifikovanja poruke u prenosu.

Šalje
se
Bobu

Slika 8-21. Korišcenje algoritama SHA-1 i RSA za potpisivanje otvorenih poruka.

Razmotrimo sada ukratko kako radi SHA-1. Počinje se tako što se poruci na kraj dodaje bit 1 i onoliko nula koliko je potrebno da njena dužina postane umnožak od 512 bitova. Zatim se 64-bitni broj koji predstavlja dužinu poruke pre dopunjavanja podvrgne disjunkciji (OR) s najmanje značajna 64 bita i rezultat smesti na njihovo mesto. Poruka na slici 8-22 dopunjava se zdesna jer tekst i slike u engleskom jeziku teku sleva udesno (donji desni ugao se doživljava kao kraj slike). U računarstvu, takva orijentacija odgovara „big-endian" računalima, kao što je SPARC, ali algoritam SHA-1 uvek dopunjava kraj poruke, bez obzira na to da li je format računara „big-en- dian" ili „little-endian".

SHA-1 tokom izračunavanja održava pet 32-bitnih promenljivih *(H*$_0$ do *H*$_4$*),* u kojima se akumulira rezultat heširanja. One su prikazane na slici 8-22(b). Inicijalizuju se konstantama koje predviđa standard.



Slika 8-22. (a) Poruka dopunjena do umnoška od 512 bitova, (b) Promenljive za prihvatanje izlaznih rezultata, (c) Niz računarskih reci.

Sada se redom obrađuju blokovi *M*$_0$ do *M*$_{n\_}$ j. Za tekući blok se prvo kopira 16 reci na

početak pomoćnog niza *W* (dužine 80 reči), kao na slici 8-22(c). Zatim se niz popunjava sa sledeče 64 reči prema formuli

$$Wj = S^{l}(W_{h\_3} \text{ XOR } VP_{,\_8} \text{ XOR } Wi_{\_u} \text{ XOR } W)_{\_16})(\ 16 < i < 79)$$

gde $S^{b}(W)$ predstavlja cirkularno (povratno) rotiranje ulevo 32-bitne reči *W*za *b* bitova. Sada se četiri privremene promenljive *(A* do *E)* inicijalizuju vrednostima promenljivih *HQ* do 77_4,

Stvarno izračunavanje može se prikazati pseudokodom na jeziku C:

```
for (i = 0; i < 80; i++) {
    temp = S^5(A) + fj (B, C, D) + E + W, + K,;
    E = D; D = C; C = S^30(B); B = A; A = temp;
```

gde su konstante *Kj* definisane standardom. Funkcije mešanja, /•_ definisane su sle- dec'im izrazima;

(0 < f < 19)
(20 < *i* < 39)
(40 < ; < 59)
(60 < *i* <79)

fi (B,C,D) = (B AND C) OR (NOT *B* AND *D) f (B,C,D) = B* XOR *C* XOR *D fl* (*B,C,D) = (B* AND O OR *(B* AND *D)* OR (C AND *D) fj* (*B,C,D*) = *B* XOR C XOR *D*

Kada se završi 80 iteracija petlje, vrednosti promenljivih A do E dodaju se vred- nostima odgovarajućih promenljivih H_Q do H_4.

Postoje tako obrađen prvi 512-bitni blok, počinje obrada sledečeg. Niz *W* se ponovo inicijalizuje novim blokom, ali *H* ostaje kako je i bilo. Po završetku ovog bloka, prelazi se na sledeči i tako sve do kraja poruke. Kada se završi s poslednjim blokom, generiše se 160-bitni kriptografski heširani sažetak na osnovu pet 32-bitnih reči iz niza *H.* Potpun C-kôd algoritma SHA-1 možete da nađete u RFC dokumentu 3174.

U toku je razvijanje novih verzija algoritma SHA-1 koji će poruke sažimati u 256, 384, odnosno 512 bitova.

8.4.4 Rođendanski napad

U svetu šifara ništa nije kao što izgleda. Pomislili biste daje dovoljno oko 2"' operacija da se podrije sažetak poruke od *m* bitova. U stvari, često je dovoljno $2^{m/2}$ operacija ako iskoristite rođendanski napad (engl. *birthday attack)* - pristup koji je objavio Yuval (1979) u svom sada klasičnom radu „Kako prevariti rabina".

Podloga za ovaj napad je pitanje koje profesori teorije verovatnoće često postavljaju studentima na svojim predavanjima: koliko treba da nas se okupi u ovoj sali, da bi se s verovatnoćom 1/2 u njoj našle dve osobe rođene istog dana? Većina studenata odgovara da treba da ih ima barem 100, dok se u stvari može izračunati da ih je potrebno samo 23. Ne ulazeći u detalje, navedimo samo to da se od 23 osobe mogu napraviti (23 x *22)12* = 253 različita para, od kojih svaki ima verovatnoću 1/365 da bude onaj traženi par. Posle ovoga, ne izgleda sve baš tako neverovatno, zar ne?

Problem možemo i da uopštimo. Ako postoji određeno preslikavanje između ulaznih i izlaznih podataka, pri čemu ima *n* ulaznih podataka (osoba, poruka itd.) i *k* mogućih izlaznih podataka (rođendana, sažetih poruka itd.), postoji *n(n* - l)/2 ulaznih parova. Ako je *n(n* — 1)/2 > *k,* šanse zajedno poklapanje su prilično dobre. Na taj način, približno, pogodak je verovatan za [Formula]. Taj rezultat znači da če se između oko

2^{19} " generisanih poruka verovatno pojaviti dve sa istim 64-bitnim sažetkom.

Pogledajmo jedan primer iz prakse. Katedra za računarstvo Državnog univerziteta ima

jedno upražnjeno mesto za stalnog saradnika na koje reflektuju dva kandidata, Tom i Dik. Tom je na univerzitet došao pre Dika, tako da ima više šanse. Ako Tom bude izabran, Dik gubi. Tom zna da Šef katedre Merilin visoko ceni njegov raci i zato

je moli da napiše preporuku Dekanu koji odlučuje o izbora. Kada se pisma jednom pošalju, postaju poverljiva.

Merilin izdaje nalog svojoj sekretarici Elen da napiše pismo Dekanu, objasnivši joj ga u glavnim crtama. Kada Elen napiše pismo, Merilin će ga pregledati, izračunati i potpisati njegov 64-bitni sažetak koji će poslati Dekanu. Samo pismo Elen može kasnije da pošalje e-poštom.

Nažalost, Elen je u romantičnoj vezi s Dikom; zato želi da „udesi" Toma, pa piše sledece pismo koje sadrži 32 opcije u zagradama.

Poštovani gospodine Dekane,

Ovim *[pismom* I *dopisom*] želim da iznesem svoje *[iskreno* I *oboreno]* mišljenje o prof. Tomu Vilsonu, koji se *[sada* I *ove godine][kandidovao* I *prijavio*] za izbor u stalno zvanje. *[Poznajem* I *Znam*] prof. Vilsona *[oko* I *skoro]* šest godina. On je *[izuzetan* I *odličan*] istraživač velikih *[talenata* I *sposobnosti], [širom sveta* I *međunarodno]* priznat po svom *[briljantnom* I *kreativnom*] pristupu *[mnogim* I *brojnim] [teškim* I *izazovnim]* problemima.

On je i [visoko *I* veoma] [cenjeni *I* uvaženi] [nastavnik *1* predavač]. *Studenti njegova* [predavanja *I* kurseve] *ocenjuju kao [*inspirativna *I* fascinantna]. *On je* [najpopularniji *I* najomiljeniji] [nastavnik *I* predavač] [kod nas *I* na našoj Katedri],

[Osim toga I *Štoviše],* prof. Vilson je *[vešt* I *utnešan]* s novcem. *[Donacije* I *Ugovori]* koje je obezbedio doneli su *[velika* I *znatna]* sredstva našoj Katedri. *[Taj novac je* I *Ta sredstva su] [omogućio* I *omogućila]* da *[sprovedemo* I *izvedemo]* mnoge *[specijalne* I *važne]* programe, *[kao* I *na primer,*] vaš program Univerzitet 2000. Bez tih sredstava ne bismo *[mogli* I *uspeli]* da nastavimo taj program, koji je talco *[važan* I *bitan]* za sve nas. Duboko sam ubeđena da njega treba izabrati na upražnjeno mesto.

Na nesreću po Toma, čim je Elen sastavila i otkucala pismo, odmah je napisala i drugo:

Poštovani gospodine Dekane,

Ovim [*pismom* I *dopisom]* želim da iznesem svoje *[iskreno* I *otvoreno]* mišljenje o prof. Tomu Vilsonu, koji se *[sada* I *ove godine][kandidovao* I *prijavio]* za izbor u stalno zvanje. *[Poznajem* I *Znam]* prof. Vilsona *[oko* I *skoro]* šest godina. On je *[loš* I *slab]* istraživač, nepoznat u svojoj *[naučnoj oblasti* I *naučnoj disciplini]*. Njegov istraživački rad *[retko je* I *možda nikada nije]* rezultovao *[objašnjavanjem* I *razumevanjem] [ključnih* I *glavnih]* problema *[sadašnjice* I *današnjice],*

Staviše, on nije [cenjen *I* uvažen] *kao* [nastavnik *I* predavač]. *Studenti njegova* [predavanja *I* kurseve] *ocenjuju kao* [jadna *I* nikakva]. *On je* [najnepopularniji *I* najomraženiji] [nastavnik *I* predavač] [kod nas *1* na našoj Katedri], *poznat* [kod nas *I* na našoj Katedri] [najviše *I* uglavnom] *po svojoj* [težnji *I* sklonosti] *da* [ismeva *I* prepada] *studente koji su dovoljno* [luckasti *I* nevaspitani] *da na njegovim časovima uopšte postavljaju pitanja.*

[Osim toga I *Staviše],* prof. Vilson nije *[vešt* I *umešan]* s novcem. *[Donacije* I *Ugovori]* lcoje je obezbedio doneli su *[mala* I *neznatna]* sredstva našoj Katedri. Ukoliko se *[novac* I *sredstva]* ne obezbede brzo, možda ćemo morati da odustanemo od nekih bitnih programa, kao što je vaš Univerzitet 2000. Nažalost, u ovim *[uslovima* I *okolnostima*], ne mogu *[iskreno* I *čista srca*] da ga preporučim za *[izbor* I *stalno nameštenje*].

Zatim Elen programira svoj računar da preko noći napravi po 2^{23} sažetka svake od poruka. Postoje šanse da će se jedan sažetak prvog pisma poklopiti s jednim sažetkom drugog. Ako do poklapanja ne dođe, Elen može da doda nove opcije i da sve pokuša ponovo

preko vikenda. Pretpostavimo daje otkrila jedno poklapanje između sažetaka „dobrog“ *(A)* i „lošeg“ (B) pisma.

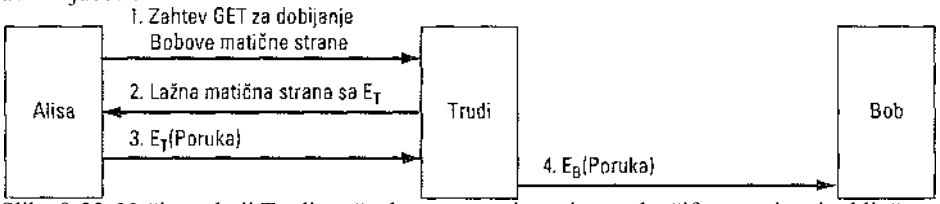
Elen sada e-poštom šalje pismo A Merilin da bi ga ova odobrila. Pismo *B* čuva u potpunoj tajnosti ne pokazujući ga nikome. Merilin, naravno, odobrava pismo, izračunava njegov 64-bitni sažetak, potpisuje ga i potpisani sažetak e-poštom šalje Dekanu. Nezavisno od toga, Elen e-poštom šalje Dekanu pismo *B* (a ne pismo *A* kao što joj je naloženo).

Pošto dobije pismo i potpisani sažetak, Dekan primenjuje algoritam sažimanja na pismo *B* i utvrđuje da se slaže sa onim što mu je poslala Merilin, pa otpušta Toma. Dekan ne zna daje Elen generisala dva pisma sa istim sažetkom i poslala mu ono drugo - koje Merilin niti je videla, niti odobrila. (Moguć epilog: Elen saopštava Diku šta je uradila. Dilc je zapanjen i raskida s njom. Elen pobesni u ispovedi se Merilin. Merilin zove Dekana. Posle svega, Tom dobija postavljenje.) Uz algoritam MD5 teško je izvesti rođendanski napad jer bi čak i generisanjem milijardu sažetaka u sekundi bilo potrebno više od 500 godina za izračunavanje svih 2^{64} sažetaka dva pisma, svakog u 64 varijante, pa i onda poklapanje nije zagarantovano. Naravno, kada 5000 računara radi paralelno, 500 godina se skraćuje na 5 sedmica. Algoritam SHA-1 je još bolji (jer je duži).

8.5 RAD S **JAVNIM** KLJUČEVIMA

Šifrovanje javnim ključem omogućava osobama koje ne dele zajednički ključ da bezbedno komuniciraju. Ono omogućuje i potpisivanje poruka bez mešanja treće strane. I na kraju, potpisani sažeci poruka omogućavaju laku proveru integriteta primljenih poruka.

Međutim, postoji problem koji smo precutno zanemarili: ako se Alisa i Bob ne poznaju, kako će svako od njih doći do javnog ključa onog drugog da bi započeli komunikaciju? Očigledno rešenje - da svako objavi ključ na svojoj Web lokaciji - nije dobro iz više razloga. Pretpostavimo da Alisa želi da pronađe Bobov javni ključ na njegovoj Web lokaciji. Kako ona to radi? Počinje tako što upiše Bobov URL. Njen Web čitač zatim traži DNS adresu Bobove matične strane i na nju šalje zahtev *GET* (slika 8-23). Nažalost, Trudi presreće zahtev i odgovara Alisi lažnom Web stranom, verovatno kopijom Bobove prave Web strane na kojoj je jedino Bobov javni ključ za- menjen Trudinim javnim ključem. Kada Alisa sada šifruje svoju prvu poruku tim ključem (E_r), Trudi će je dešifrovati, pročitati, ponovo šifrovati Bobovim javnim ključem i poslati je Bobu koji pojma nema daje Trudi pročitala poruku. Gore je to što Trudi može da menja poruke pre nego što ih ponovo šifruje i pošalje Bobu. Odavde je očigledno daje neophodan neki mehanizam kojim bi se bezbedno razmenjivali javni ključevi.



Slika 8-23. Način na koji Trudi može da presretne i menja poruku šifrovanu javnim ključem.

8.5.1 Sertifikati

Kao prvi način bezbednog distribuiranja javnih ključeva zamislimo distribucioni centar koji radi 24 sata dnevno šaljući javne ključeve na zahtev. Jedan od mnogih problema ovog rešenja je to što će na centar biti velika navala i on će postati usko grlo na Internetu. Ako se, nedajbože, isključi zbog kvara, bezbednost na Internetu trenutno nestaje.

Iz ovog i dragih razloga, smišljeno je takvo rešenje da distribucioni centar ne mora da bude na mreži sve vreme. U stvari, on uopšte i ne mora da bude na mreži. On treba samo da izdaje sertifikate za javne ključeve koji pripadaju pojedinačnim korisnicima, kompanijama i drugim organizacijama. Takav centar se sada zove **ovlašćena organizacija za izdavanje sertifikata** (engl. *Certification Authority, CA*).

Primera radi, pretpostavimo da Bob želi da omogući Alisi i dragim osobama da s njim bezbedno komuniciraju. On može da ode u CA sa svojim javnim ključem i ličnom kartom ili vozačkom dozvolom, i da traži sertifikat. CA tada izdaje sertifikat sličan onom na slici 8-24 i potpisuje rezultat njegovog SHA-1 heširanja svojim privatnim ključem. Bob CA organizaciji plaća naknadu i dobija disketu sa sertifikatom i potpisanim rezultatom njegovog heširanja.

I hereby certify that the public key
19836A8B03030CF83737E3837837FC3s87092827262643FFA82710382828282A
belongs to
Robert John Smith 12345
University Avenue Berkeley, CA
94702 Birthday: July 4, 1958
Email: bob@superdupernet.com

SHA-1 hash of the above certificate signed with the CA's private key

Slika 8-24. Primer sertifikata i potpisanog rezultata njegovog heširanja.

Osnovni zadatak sertifikata je da poveže javni ključ sa imenom principala (pojedinca, kompanije itd.) Sami sertifikati nisu tajni niti zaštićeni. Bob može, na primer, odlučiti da svoj novi sertifikat postavi na svoju Web lokaciju i da na matičnu stranu stavi vezu: Pritisnite ovde da biste videli sertifikat mog javnog ključa. Kada neko pritisne vezu, dobiće i sertifikat i potpisani blok (potpisani SHA-1 sažetak sertifikata).

Prođimo ponovo kroz scenario sa slike 8-23. Kada Trudi presretne Alisin zahtev za Bobovu matičnu stranu, šta može da uradi? Ona na lažnu stranu može da stavi sopst- veni sertifikat i potpisani blok, ali kada Alisa pročita sertifilcat, odmah će znati da ne razgovara s Bobom jer se u njemu ne nalazi Bobovo ime. Trudi može da Bobovu stranu izmeni u hodu zamenjujući njegov javni ključ svojim. Međutim, kada Alisa na sertifikat primeni algoritam SHA-1, đobiće sažetak koji se razlikuje od onoga koji dobija kada primeni opštepoznati javni ključ organizacije CA na potpisani blok. Pošto Trudi nema privatni ključ organizacije CA, ne može da generiše potpisani blok koji predstavlja rezultat heširanja izmenjene Web strane koja sadrži njen javni ključ. Na taj način, Alisa može da bude sigurna da ima Bobov javni ključ, a ne Trudin ili neki dragi. Kao što smo ranije rekli, za ovakvu šemu nije potrebno da CA bude na mreži tokom verifikovanja, čime se izbegava potencijalno usko grlo.

Iako je glavna funkcija sertifikata da javni ključ poveže s principalom, on može da ga poveže i sa atributom. Sertifikat, na prirner, može da tvrdi: Ovaj javni ključ pripada osobi starijoj od 18 godina. Na taj način, on se može iskoristiti za dokazivanje zrelosti vlasnika, pa neki maloletnik neće moći da pristupi materijalu koji nije podesan za njega, dok vlasnik

ključa pri tome ne otkriva svoj identitet. Osoba koja ima sertifikat obično će ga uputiti Web lokaciji, principalu ili procesu koji vodi računa o starosti korisnika. Ta lokacija, principal ili proces generisaće slučajan broj i šifrovati ga javnim ključem iz sertifikata. Ako vlasnik ključa uspe da ga dešifraje i pošalje natrag, to je dokaz da vlasnik zaista ima atribut koji se pominje u sertifikatu. Alternativno, za generisanje ključa sesije koja sledi može se upotrebiti slučajan broj.

Dragi primer u kome sertifikat može da sadrži atribut nalazimo u objektno orijen- tisanom distribuiranom sistemu. Svaki objekat normalno ima više metoda. Vlasnik objekta može svakog korisnika da snabde sertifikatom s bit mapom metoda koje korisnik sme da poziva, vezujući mapu za javni ključ pomoću potpisanog sertifikata. Ako onaj ko ima sertifikat može da dokaže da poseduje odgovarajući privatni ključ, dozvoljava mu se da koristi metode iz bit mape. Ne mora se znati identitet vlasnika sertifikata, što je zgodno kada je važna privatnost.

8.5.2 X.509

Ako bi svako ko želi da ima nešto potpisano odlazio u CA s drugačijim sertifikatom, obrada svih tih formata ubrzo bi postala problem. Zbog toga su smišljeni standardni sertifikati koje je odobrila organizacija ITU. Standard se zove X.509 i široko se koristi na Internetu. Od standardizacije 1988. godine, doživeo je tri verzije. Govoriće- mo o njegovoj verziji 3.

Na oblikovanje standarda X.509 uveliko su uticali pobornici modela OSI i zato je nasledio neke od njegovih najgorih osobina (npr. imenovanje i kodiranje). Iznenađujuće je da je IETF podržao X.509, iako je skoro u svakoj dragoj oblasti, počev od mašinskih adresa, preko transportnih protokola, do formata poruka e-pošte, uglavnom zanemarivao model OSI, tj. pokušavao da stvar izvede kako treba. IETF verzija standarda X.509 opisana je u RFC dokumentu 3280.

Standard X.509, u suštini treba da opiše sertifikate. Osnovna definisana polja navedena su na slici 8-25. Propratni opisi treba da dočaraju čemu polja služe. Dopunska objašnjenja potražite u samom standardu ili u RFC dokumentu 24.59.

Na primer, ako Bob radi u kreditnom odeljenju banke NovacZaSvakoga, njegova adresa prema standardu X.509 mogla bi biti sledeca:

/C=YU/0=NovacZaSvakoga/OU=Krediti/CN=Bob/

gde *D* označava državu, *O* - organizaciju, *OJ*-organizacionu jedinicu, a *UI*-uobičajeno ime. Ovlaščene (CA) i druge organizacije imenuju se na sličan način. Suštinski problem sa imenima prema standardu X.509 ogleda se u tome što Alisa, kada pozove URL bob@novaczasvakoga.com i dobije sertifikat sa X.500 imenom, neće znati da li se sertifikat odnosi na „njenog“ Boba. Na sreću, počev od verzije 3, umesto X.500 imena dozvoljena su i DNS imena, tako da problem nestaje sam po sebi.

Polje	Značenje
Verzija	Verzija standarda X.509
Serijski broj	Ovaj broj uz ime CA jedinstveno identifikuje sertifikat
Algoritam za potpisivanje	Algoritam iskorišćen za potpisivanje sertifikata
Davalac sertifikata	Ime CA prema standardu X.500
Period važenja	Početak i kraj perioda važenja sertifikata
Ime korisnika sertifikata	Korisnik za čiji se ključ garantuje
Javni ključ	Javni ključ korisnika i identifikator algoritma koji ga koristi
Identifikator davaoca sertifikata	Neobavezni jedinstven identifikator davaoca sertifikata
Identifikator korisnika sertifikata	Neobavezni jedinstven identifikator korisnika sertifikata
Proširenja	Definisana su mnoga proširenja
Potpis	Potpis sertifikata (stavljen pomoću privatnog ključa CA)

Slika 8-25. Osnovna polja sertifikata prema standardu X.509.

Sertifikati se prema modelu OSI kodiraju **apstraktnom sintaksnom notacijom** 1 (engl. *Abstract Syntax Notation l,ASN.l),* što daje rezultat sličan strukturi na jeziku C, samo uz veoma posebno i obilno označavanje. Više informacija o standardu X..509 možete nadi kod Forda i Bauma (2000).

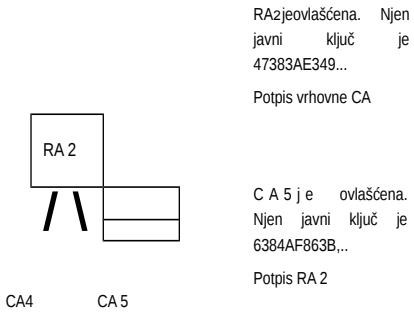
8.5.3 Infrastrukture za sertificiranje javnih ključeva

Jedna organizacija za izdavanje sertifikata za ceo svet očigledno nije realno rešenje. Ona ne bi mogla da izdrži toliko opterećenje, a i bila bi glavni izvor kvarova. Bolje rešenje bi bilo da takva organizacija ima više filijala koje bi sve sertifikate potpisivale istim privatnim ključem. Iako bi se time otklonili problemi opterećenja i kvarova, pojavio bi se nov: „procurivanje“ ključa u javnost. Ako širom sveta postoje desetine servera i svi čuvaju privatni CA ključ, znatno bi se povećale šanse da ga neko ukrade ili da on na neki drugi način iscuri iz organizacije. Pošto bi krađa ovog ključa mogla da ugrozi svetsku bezbednosnu elektronsku infrastruktura, veoma je rizično da ga čuva samo jedna ovlašćena organizacija.

Osim toga, ko bi stvarno upravljao takvom organizacijom? Teško je zamisliti ustanovu koju bi čitav svet smatrao legitimnom i od poverenja. U nekim zemljama se insistira da CA organizacijom upravlja vlada, dok je u drugima trend upravo suprotan.

Zbog opisanih nedoumica, razvijen je nov način sertificiranja javnih ključeva, tzv. **infrastruktura za sertificiranje javnih ključeva** (engl. *Public Key Infrastructure, PKI)*. Na ovom mestu ćemo razmotriti samo principe njenog rada jer se ona i dalje razvija, pa će se detalji vremenom menjati.

PKI ima više komponenata, uključujući korisnike, ovlašćene CA organizacije, ser- tifikate i kataloge. PKI treba da obezbedi način strukturiranja ovih komponenata i da definiše standarde za različite dokumente i protokole. Posebno jednostavnu PKI strukturu predstavlja hijerarhija CA organizacija, prikazana na slici 8-26. U primeru su prikazana tri nivoa, ali ih u praksi može biti i manje i više. CA organizacija najvišeg nivoa, vrh hijerarhije, sertificira CA organizacije dragog nivoa koje ćemo zvati **regionalne** CA **organizacije** (engl. *Regional Authorities, RAs)* jer njihov delokrug obuhvata određenu geografsku oblast, kao što je država ili kontinent. Taj izraz, međutim, nije standardan, kao što uostalom ne postoji standardna terminologija za bilo koji nivo CA hijerarhije. RA organizacije seitificiraju operativne CA organizacije koje izdaju X.509 sertifikate organizacijama i pojedincima. Kada vrhovna CA organizacija ovlašćuje novu RA, ona izdaje X.509 sertifikat sa izjavom da je odobrila rad te RA, u njega uključuje javni ključ nove RA, potpisuje ga i predaje toj RA. Slično tome, kada RA ovlašćuje novu CA, ona joj izdaje i potpisuje sličan sertifikat koji sadrži javni ključ te CA.



(a) (b)

Slika 8-26. (a) Hijerarhijska PKI infrastruktura, (b) Lanac sertifikata.

PKI iz našeg primera radi na sledeći način. Pretpostavimo da Alisi treba Bobov javni ključ da bi s njim uspostavila komunikaciju, pa zato traži i nalazi sertifikat koji ga sadrži, potpisan od strane CA 5. Međutim, Alisa nikada nije čula za organizaciju CA 5. Po njoj, to bi mogla biti i Bobova desetogodišnja ćerkica. Ona može da ode u CA 5 i da kaže: Pokažite vašu legitimaciju. CA 5 će pokazati sertifikat sa svojim javnim ključem koji je dobila od RA 2. Kada ima javni CA 5 ključ, Alisa može da proveri da li je Bobov sertifikat zaista potpisala CA 5, pa je shodno tome legalan.

Osim ako organizacijom RA 2 ne upravlja Bobov dvanaestogodišnji sin. Zato u sledećem koraku Alisa traži od RA 2 da dokaže svoju legitimnost. U odgovor na zah- tev ona đobija sertifikat s javnim RA 2 ključem koji je potpisala vrhovna CA. Tek sada je Alisa uverena da u rukama ima pravi Bobov javni ključ.

Ali, kako će Alisa doći do javnog ključa vrhovne CA? Zanimljivo, ali izgleda da neće morati da radi ništa. Pretpostavlja se da svako zna javni ključ vrhovne CA. Na primer, on može da bude fabrički ugrađen u njen Web čitač.

Bob je ljubazan momak i ne želi da previše opterećuje Alisu. On zna da bi Alisa morala da proverava CA 5 i RA 2, pa da bi joj uštedeo muke, zajedno sa svojim ser- tifikatom šalje i ta dva. Sada Alisa treba da pomoću opštepoznatog javnog ključa vrhovne CA otvori njen sertifikat i da na osnovu javnog ključa sadržanog u njemu proveri RA sledećeg nivoa. Na taj način, Alisa ne mora ni sa kime da stupa u vezu. Pošto su svi sertifikati potpisani, ona lako može da otkrije pokušaj falsifikovanja. Ovakvo uzastopno sertificiranje sve do najvišeg nivoa ponekad se zove **lanac pove- renja** (engl. *chain of trust)* ili **hijerarhijski niz sertifikata** (engl. *certification path).* Opisana tehnika se široko koristi u praksi.

Naravno, i dalje ostaje otvoreno pitanje ko treba da upravlja vrhovnom CA organizacijom. Rešenje je nađeno u razbijanju jedinstvene vrhovne CA organizacije na više vrhovnih CA organizacija, od kojih svaka ima svoje RA i CA organizacije. U stvari, savremeni Web čitači se isporučuju sa javnim ključevima više od 100 vrhovnih CA organizacija, ponekada zvanih **pouzdana polazišta** (engl. *trust anchors).* Na taj način se izbegavaju rizici skopčani s postojanjem jedinstvene ovlašćene organizacije.

Međutim, sada proizvođač Web čitača treba da odluči o tome koja su od ponuđenih „pouzdanih“ polazišta stvarno pouzdana, a koja samo nose taj naziv. I sve se spušta do krajnjeg korisnika koji treba da poveruje proizvođaču Web čitača da će napraviti pravi izbor i da u čitač neće uključiti svako „pouzdano“ polazište koje je spremno da plati odgovarajuću naknadu. Većina čitača omogućavaju korisnicima da provere ključeve vrhovnih CA organizacija (nudeći im obično sertifikate koje je potpisala vrhovna CA) i da odbace one koji im izgledaju sumnjivi.

Katalozi

Svaka PKI mora da vodi računa i o tome da negde skladišti sertifikate (i lance po- verenja koji vode do nekog pouzdanog polazišta). Jedna mogućnost je da svaki korisnik čuva svoj sertifikat. Iako je to bezbedno (korisnik ne može da menja potpisani sertifikat a da se to ne primeti), nije baš zgodno. Predloženo je i da DNS server čuva katalog sertifikata. Pre nego što stupi u vezu s Bobom, Alisa verovatno treba da na DNS serveru potraži njegovu IP adresu, pa zašto kao odgovor, zajedno sa IP adresom, ne bi dobila i čitav Bobov lanac poverenja?

Neki misle da to tako i treba da radi, ali drugi više vole namenslce servere kataloga koji rade isključivo sa X.509 sertifikatima. Takvi katalozi bi se mogli pretraživati prema pojedinim svojstvima X.500 imena. Na primer, takva usluga kataloga mogla bi teorijski da pruži odgovor na zahtev: „Daj mi spisak svih osoba koje se zovu Alisa i koje rade u prodajnim ođeljenjima bilo gde u SAD ili Kanadi“. Sistem LDAP bi se mogao upotrebiti za čuvanje takvih informacija.

Povlačenje sertifikata

I u stvarnom svetu ima mnogo sertifikata, npr. pasoša, vozačkih dozvola itđ. Ponekad takvi sertifikati treba da budu povučeni, npr. vozačka dozvola zbog vožnje pod dejstvom

alkohola i drugih saobraćajnih prekršaja. Slično je i u digitalnom svetu: ustanova koja je izdala sertifikat može odlučiti da ga povuče jer su ga osoba ili organizacija kojima su izdate na neki način zloupotrebili. Sertifikat se može povući i ako je neko provalio privatni ključ korisnika ili još gore, privatni ključ CA organizacije. Prema tome, PKI mora da se pozabavi problemom povlačenja sertifikata.

Kao prvi korak u ovom pravcu, svaka CA povremeno treba da objavljuje listu povučenih sertifikata (engl. *Certificate Revocation List, CRL)* sa serijskim brojevima svih sertifikata koji su povučeni. Pošto sertifikati sadrže rok važenja, CRL treba da sadrži serijske brojeve samo sertifikata kojima rok još nije istekao. Kada mu rok važenja prođe, sertifikat automatski postaje neispravan, tako da treba praviti razliku između sertifikata kojima je rok istekao i onih koji su povučeni. Nijedni ni drugi se više ne mogu koristiti.

Kada uvedete CRL liste, to nažalost znači da korisnik koji želi da upotrebi sertifikat mora prvo da preuzme i pregleda listu da bi se uverio da sertifikat nije povučen. Ako je povučen, ne bi ga trebalo koristiti. Međutim, i kada sertifikat nije na listi, on je ipak možda povučen - neposredno po objavljivanju liste. Prema tome, bićete potpuno sigurni tek kada se direktno obratite CA organizaciji. Štaviše, ako sutra ponovo poželite da upotrebite isti sertifikat, moracete ponovo da se obratite CA organizaciji, jer je u međuvremenu možda povučen.

Dodatna komplikacija je to što se povučenom sertifikatu može ponovo vratiti važnost, na primer, u slučajevima kada je povučen zato što nije na vreme plaćena neka obaveza koja je naknadno ipak plaćena. Potreba za povlačenjem sertifikata (i ponovnim vraćanjem njihove važnosti) kompromituje jedno od najboljih svojstava sertifikata - da ih možete koristiti bez obraćanja CA organizaciji.

Gde bi trebalo čuvati CRL liste? Verovatno na istom mestu gde i sertifikate. Jedna strategija je da CA organizacije povremeno aktivno guraju CRL liste koje katalozi obrađuju tako što jednostavno iz svog sadržaja izbacuju povučene sertifikate,. Ako se katalozi ne koriste i za stvarno čuvanje sertifikata, CRL liste se mogu keširati na razna pogodna mesta na mreži. Pošto je i CRL lista potpisani dokument, svako njeno falsi- fikovanje se lako otkriva.

Kada sertifikati imaju dug period važenja, i CRL liste će biti dugačke. Na primer, ako kreditne kartice važe 5 godina, broj povučenih kartica biće mnogo veći nego kada bi važile samo 3 meseca. Standardan način rada s dugačkim CRL listama obuhvata samo povremeno objavljivanje jedne glavne liste koja se, međutim, često ažurira. Na taj način se pri distribuiranju CRL lista štedi propusni opseg.

8.6 BEZBEDNOST KOMUNICIRANJA

Pretresli smo konačno alatke koje se koriste u ovoj oblasti, objasnivši sve važnije tehnike i protokole. Ostatak poglavlja posvetićeino primeni ovih tehnika u praktičnom obezbeđivanju mreže, a na kraju ćemo nešto natuknuti i o društvenim aspektima bezbednosti.

U naredna četiri odeljka bavićemo se bezbednošću komuniciranja, odnosno pitanjem kako da bitove diskretno i bez menjanja sprovedemo od izvorišta do odredišta i kako da neželjene bitove zadržimo napolju. To nisu i jedini bezbednosni problemi rada u mreži, ali su izvesno među najvažnijima, pa njima i počinjemo.

8.6.1 IPsec

IETF je oduvek znao da Internetu nedostaje bezbednost. Međutim, njeno uvođenje otežavaju sukobi oko toga gde je smestiti. Većina stručnjaka smatra da se stvarna bezbednost

može postići samo šifrovanjem i proverom integriteta od jednog do drugog kraja (tj., u sloju aplikacija). To znači da izvorišni proces treba da šifruje podatke i(ili) da zaštiti njihov integritet, a zatim da ih pošalje odredišnom procesu gde će biti dešifrovani i(ili) provereni. Na taj način se lako može otkriti svaki pokušaj falsifiko- vanja između dva procesa, uključujući i pokušaje na nivou svakog od dva operativna sistema. Ovaj pristup je problematičan zato što bi za njegovo sprovođenje sve aplikacije morale da postanu „bezbeđnosno svesne". Nastavljajući u istom smislu, sledeći, nešto manje dobar pristup bio bi da se podaci šifruju u transportnom sloju ili u novom sloju između sloja aplikacija i transportnog sloja - komunikacija bi i dalje išla od jednog do drugog kraja, a aplikacije se ne bi morale menjati.

Pobornici suprotnog gledišta zagovaraju tezu da korisnici ne shvataju bezbednost u potpunosti i da neće biti sposobni da je koriste na pravi način, da niko ne želi da menja postojeće programe ni za jotu, i da zato mrežni sloj mora da proverava identitet paketa i(ili) da ih šifruje bez učešća korisnika. Posle višegodišnjih žučnih rasprava, to gledište je dovoljno prevagnulo da bude definisan standard za bezbednost u mrežnom sloju. Jedan od argumenata bio je i to da šifrovanje u mrežnom sloju ne sprečava bez- bednosno svesne korisnike da stvar urade kako treba, a da ipak u izvesnoj meri pomaže korisnicima koji bezbednost ne uzimaju ozbiljno.

Rezultat navedenih rasprava bio je projekat IP bezbednost (engl. *IP security, IPsec),* opisan, između ostalog, i u RFC dokumentima 2401, 2402 i 2406. Ne žele svi korisnici šifrovanje (zato što je to računarski zahtevan korak). Umesto da šifrovanje bude neobavezno, odlučeno je da se sprovodi sve vreme, ali da se omogući primena nultog algoritma, koji je opisan u RFC dokumentu 2410 i nahvaljen zbog svoje jednostavnosti, lakoće realizovanja i velike brzine.

Projekat IPsec obuhvata čitavu strukturu sastavljenu od više različitih usluga, algoritama i nivoa granularnosti paralelnih procesa. Usluge su pojedinačno „rasparčane" zato što ne želi svako da plati punu cenu da bi sve usluge imao u svako doba, već se one nude „a la carte". U glavne usluge spadaju bezbednost, integritet podataka i zaštita od napada ponavljanjem poruka (uljez reprodukuje konverzaciju). Za sve usluge se koristi šifrovanje simetričnim ključem jer su ovde primame visoke performanse.

U sistem je ugrađeno više algoritama zato što jedan algoritam koji danas izgleda bezbedan već sutra može biti provaljen. AJco IPsec ne zavisi od konkretnog algoritma, njegova struktura ostaje i kada neko provali neki pojedinačan algoritam.

Različiti nivoi granularnosti paralelnih procesa između ostalog omogućavaju obezbeđivanje samo jedne TCP veze, svog saobraćaja između para računara ili svog saobraćaja između para obezbeđenih usmerivača.

Iznenađujući aspekt sistema IPsec je to što on radi sa uspostavljanjem direktne veze iako se nalazi u IP sloju. U stvari, to i ne iznenađuje, jer se za postizanje bezbed- nosti mora uspostaviti ključ koji se koristi tokom određenog vremena - dakle, jedna vrsta veze. Isto tako, troškovi uspostavljanja veze amortizuju se brojem paketa koji se njom prenose. „Veza" se u kontekstu sistema IPsec zove bezbednosno povezivanje (engl. *security association, SA).* SA je jednosrnerna veza između dve krajnje tačke koja ima svoj bezbednosni identifikator. Ako je saobraćaj potrebno obezbediti u oba smera, koriste se dva bezbednosna povezivanja. Bezbednosni identifikatori putuju ovim obezbeđenim vezama unutar paketa i kada stignu na odredište, koriste se za traženje ključeva i drugih informacija.

U tehničkom smislu, IPsec ima dva glavna dela. Prvi opisuje dva nova zaglavlja koja se

mogu dodati paketu za prenošenje bezbednosnog identifikatora, podataka za proveru integriteta i drugih informacija. Drugi deo, bezbednosni protokol za rad sa šiframa na Internetu (engl. *Internet Security Association and Key Management Protocol, ISAKMP),* služi za uspostavljanje ključeva. Nećemo se dalje baviti protokolom ISAKMP (1) zato što je izuzetno složen i (2) zato što je njegov glavni protokol za razmenu šifara na Internetu (engl. *Internet Key Exchange, IKE*) prepun grešaka i zreo za zamenjivanje (Perlman i Kaufman, 2000).

IPsec se može koristiti u dva režima. U transportnom režimu, IPsec zaglavlje se umeće neposredno iza IP zaglavlja. Polje *Protokol* u IP zaglavlju menja se i ukazuje da iza normalnog IP zaglavlja sledi IPsec zaglavlje (pa tek onda TCP zaglavlje). IPsec zaglavlje sadrži bezbednosne podatke, prvenstveno SA identifikator, nov redni broj paketa i možda proveru integriteta korisnog tereta.

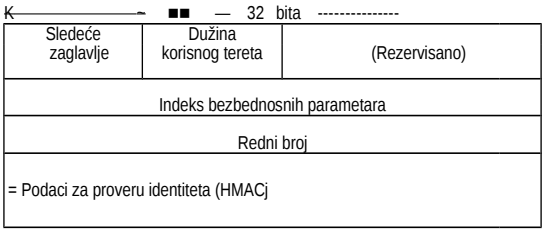
U tunelskom režimu rada, celokupan IP paket, zajedno sa zaglavljima, kapsulira se u telo novog IP paketa s potpuno novim IP zaglavljem. Tunelski režim je zgodan kada se tunel završava u tački koja nije konačno odredište. U nekim slučajevima, tunel se završava u bezbednosnom mrežnom prolazu, na primer, u zaštitnoj barijeri kompanije. U ovom režimu, zaštitna barijera kapsulira i dekapsulira prolazeće pakete. Kada se tunel završava na bezbednosnom računam, računari na lokalnoj mreži kompanije ne moraju da brinu o sistemu IPsec. Za njega zna samo zaštitna barijera.

Tunelski režim je koristan i kada se snop TCP veza udružuje i obrađuje kao jedinstven šifrovani tok jer to sprečava uljeza da utvrdi ko kome šalje koliko paketa. Ponekada je dovoljno znati i samo to koliko se paketa upućuje u kom smeru. Na primer, ako tokom ratne krize saobraćaj između Pentagona i Bele kuće naglo opadne, a naglo se poveća između Pentagona i neke vojne baze u Stenovitim planinama Kolo- rada, uljez bi odatle mogao da izvede zanimljiv zaključak. Praćenje i proučavanje toka čak i šifrovanih paketa zove se analiza saobraćaja. Tunelski režim omogućava da se ona u izvesnoj meri omete. Nedostatak tunelskog režima je dodatno IP zaglavlje koje znatno povećava pakete. Transportni režim mnogo manje menja veličinu paketa.

Prvo novo zaglavlje je zaglavlje za proveru identiteta (engl. *Authentication Header, AH)*. Ono obezbeđuje proveru integriteta i bezbednost od napada ponavljanjem poruka, ali ne i tajnost (nema šifrovanja). Korišćenje zaglavlja AH u transportnom režimu rada, prikazano je na slici 8-27. U protokolu IPv4 ono se postavlja između IP zaglavlja (sa svim eventualnim opcijama) i TCP zaglavlja. U protokolu IPv6 ono predstavlja samo još jedno dodatno zaglavlje i tako se i tumači. Njegov format je u stvari blizak formatu standardnog dodatnog IPv6 zaglavlja. Koristan teret se može dopuniti do neke dužine, već kako zahteva određeni algoritam za provera identiteta.

Obezbeđeni deo

IP zaglavlje	AH	TCP zaglavlje	Koristan teret + dopuna



Slika 8-27. IPsec zaglavlje za proveru identiteta u transportnom režimu za IPv4.

Ispitajmo sada AH zaglavlje. Polje *Sledeće zaglavlje* čuva vrednost koju je imalo IP polje *Protokol* pre nego što je zamenjena vrednošću 51 koja označava da sledi AH zaglavlje. U većini slučajeva tu dolazi kod za TCP zaglavlje (6). *Dužina korisnog tereta* je broj 32-bitnih reči u AH zaglavlju, umanjen za 2.

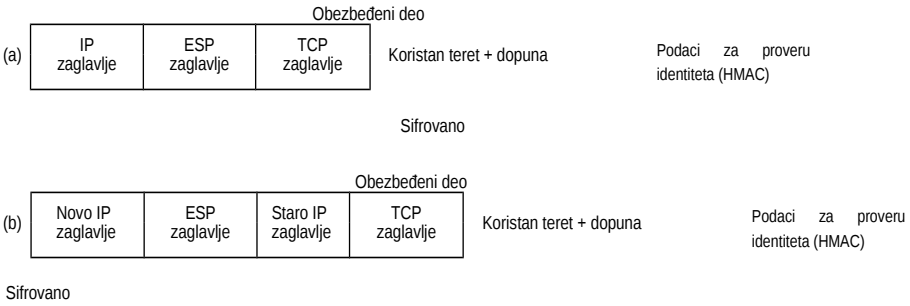
Indeks bezbednosnih parametara je identifikator veze. Njega umeće pošiljalac da bi ukazao na određeni zapis u bazi podataka primaoca. Taj zapis sadrži zajednički ključ koji se koristi na toj vezi i druge informacije o njoj. Da je protokol napravila organizacija ITU, a ne IETF, ovo polje bi se zvalo *Broj virtuelnog kola.*

U polju *Redni broj* numerišu se svi paketi poslati SA vezom. Svaki paket dobija jedinstven broj, čak i oni koji se ponovo šalju. Drugim recima, ponovo poslat paket dobija drugačiji broj od originala (iako su im redni TCP brojevi isti). Ovo polje treba da otkriva napad ponavljanjem poruka i zato se redni brojevi ne smeju reciklovati. Ako se potroši svih 2^{32} rednih brojeva, komunikacija se može nastaviti samo novo- uspostavljenom SA vezom.

Najzad dolazimo do polja *Podaci za proveru identiteta,* koje je promenljive dužine i sadrži digitalni potpis korisnog tereta. Kada se SA veza uspostavi, dve strane dogovaraju algoritam za potpisivanje koji će koristiti. Ovde se normalno ne koristi šifro- vanje javnim ključem jer pakete treba obrađivati izuzetno brzo, a svi poznati algoritmi za šifrovanje javnim ključem previše su spori. Pošto se IPsec zasniva na šifrovanju simetričnim ključem, a pošiljalac i primalac dogovaraju zajednički ključ pre nego što uspostave SA vezu, taj zajednički ključ se koristi za potpisivanje. U jednoj jednostavnoj izvedbi, izačunava se heširani sažetak paketa kome je dodat zajednički ključ. Zajednički ključ se, naravno, ne šalje. Šema slična opisanoj zove se kod za proveru identiteta heširane poruke (engl. *Hashed Message Authentication Code, HMAC).* Ovaj kod se mnogo brže izračunava od uzastopnog izvršavanja, prvo algoritma SHA-1, pa zatim algoritma RSA.

AH zaglavlje ne dozvoljava šifrovanje podataka, tako da je zgodno onda kada je potrebna provera integriteta, ali ne i tajnost. Značajna osobina AH zaglavlja je to što provera integriteta pokriva i neka polja IP zaglavlja - ona koja se ne menjaju od jednog do dragog usmerivača. Polje *Životni vek,* na primer, menja se pri svakom skoku, tako da se ne može uključiti u provera integriteta. Međutim, IP adresa izvorišta je uključena u provera, pa uljez ne može da menja izvorište paketa.

Alternativno IPsec zaglavlje je kapsulirajuće bezbednosno zaglavlje (engl. *Encapsulating Security Header, ESP).* Njegovo korišćenje u transportnom i tunelskom režimu prikazano je na slici 8-28.



Slika 8-28. (a) ESP u transportnom režimu rada. (b) ESP u tunelskom režimu rada.

ESP zaglavlje sadrži dve 32-bitne reči. To su polja *Indeks bezbednosnih parametara* i *Redni broj,* kao i u AH zaglavlju. Treća reč koja ih obično sledi (ali tehnički nije deo zaglavlja) jeste *Inicijalizacioni vektor* koji se koristi za šifrovanje podataka, osim ako se koristi nulto šifrovanje , kada vektora nema.

Kao i AH, šema ESP obezbeđuje i podatke za proveru integriteta (HMAC), ali oni nisu uključeni u zaglavlje, već dolaze posle korisnog tereta (slika 8-28.). Stavljanje HMAC podataka na kraj ima prednost pri hardverskoj realizaciji. HMAC podaci se mogu izračunavati dok koristan teret prolazi kroz mrežni interfejs i zatim im se dodati na kraju. Iz istog razloga Ethernet i druge lokalne mreže stavljaju CRC proveru podataka iza stvarnog zaglavlja, u njegov završni blok. Uz AH, paketi se moraju srneštati u bafer i njihov potpis izračunavati pre nego što se pošalju, što potencijalno smanjuje broj paketa koji se mogu poslati u sekundi.

Ako ESP može sve što i AH, čak i više, a i efikasnije se podiže, šta će nam onda uopšte AH? Razlozi su uglavnom istorijske prirode. Prvobitno je šema AH bila na- menjena samo očuvanju integriteta, a šema ESP samo tajnosti. Kasnije je u šemu ESP dodata i provera integriteta, ali projektanti AH nisu želeli da ta šema, posle toliko uloženog truda, jednostavno nestane. Njihov jedini stvarni argument je to da AH prove- rava delove IP zaglavlja, što ESP ne radi, ali je taj argument tanak. Drugi, isto tako slab argument je i to što proizvodi koji ne podržavaju ESP već AH, mogu lakše da dobiju izvoznu dozvolu jer ne šifruju podatke. Sve u svemu, AH će verovatno u budućnosti nestati.

8.6.2 Zaštitne barijere

Mogućnost povezivanja svakog računara sa svakim drugim računarom bilo gde na svetu istovremeno je i blagoslov i kazna. Za kućne korisnike je tumaranje Internetom velika zabava. Za korporacijske službe bezbednosti to je noćna mora. Većina kompanija drži veliku količinu poverljivih informacija na mreži: poslovnih tajni, planova razvoja novih proizvoda, marketinških strategija, finansijslcih analiza itd. Izručivanje ovih informacija konkurenciji moglo bi imati katastrofalne posledice.

Osim rizika od curenja informacija, postoji i opasnost da nepoželjne informacije uđu u sistem. Naročito virusi, crvi i druga digitalna gamad mogu da naruše bezbednost, da unište dragocene podatke i da opsežno angažuju administratore na otklanjanju posledica. Takve programe često unesu sami zaposleni koji žele da isprobaju neku ig- ricu koja je trenutni hit.

Iz ovoga sledi da je neophodan mehanizam koji će „dobre“ bitove držati unutra, a one „loše“ napolju. Jedan način je da upotrebite IPsec. On štiti podatke dok se prenose s jednog na drugo obezbeđeno mesto. Međutim, IPsec ne sprečava prodiranje digitalne gamadi i uljeza u lokalnu mrežu kompanije. Da bismo videli kako se to može uraditi, treba da razmotrimo zaštitne barijere.

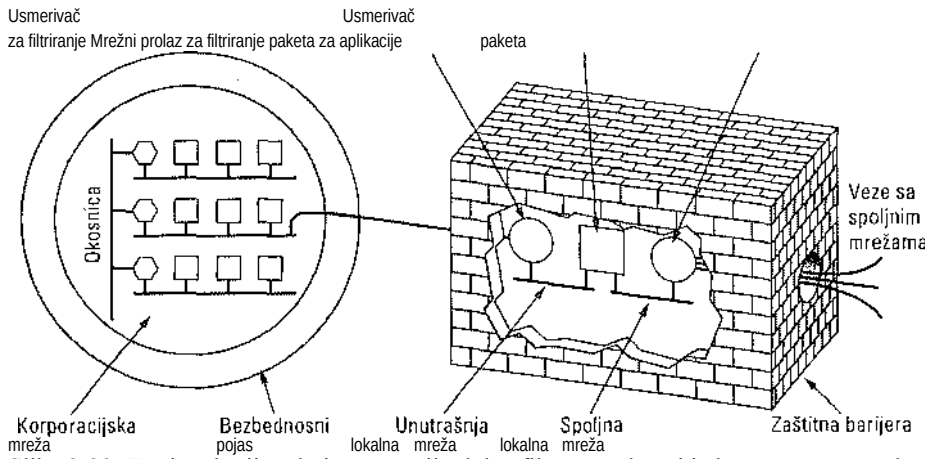
Zaštitne barijere (engl. *firewalls)* samo su savremena varijanta obezbeđivanja srednjevekovnih utvrđenja opkoljavanjem zamka dubokim šancem. Takvo obezbe- đenje je svakoga ko ulazi u zamalc ili izlazi iz njega prisiljavalo da prelazi preko jedinog visećeg mosta gde ga je kontrolisala „ulazno-izlazna“ straža. Isti trik se može primeniti i u mrežama: kompanija može da ima mnogo lokalnih mreža, međusobno povezanih na razne načine, ali se sav saobraćaj ka kompaniji ili iz nje usmerava kroz elektronski „viseći most“ - zaštitnu barijeru (slika 8-29).

Zaštitna barijera u ovoj konfiguraciji ima dve komponente: dva usmerivača koji filtruju pakete i mrežni prolaz za aplikacije. Postoje i jednostavnije konfiguracije, ali ova ima tu prednost da svaki paket koji želi da napusti kompaniju ili da uđe u nju mora prvo da prođe dva filtra i jedan mrežni prolaz za aplikacije. Nema drugog puta. Čitaoci koji misle da je dovoljan jedan kontrolni punkt očigledno davno nisu leteli na nekoj od međunarodnih linija.

Svaki **filtar za pakete** (engl. *packet filter)* predstavlja standardni usmerivač s dodatnom funkcionalnošću koja mu omogućava da pregleda svaki dolazni ili odlazni paket. Paketi koji ispunjavaju određeni kriterijum normalno se prosleđuju. Oni koji ga ne ispunjavaju, jednostavno se odbacuju.

Na slici 8-29, odlazne pakete najverovatnije kontroliše filtar na unutrašnjoj lokalnoj mreži, a dolazne pakete filtar na spoljnoj lokalnoj mreži. Paketi koji prođu prvu proveru upućuju se mrežnom prolazu za aplikacije na dopunsko ispitivanje. Raspoređivanje dva filtra na različite lokalne mreže obezbeđuje da svaki paket koji dolazi ili odlazi mora da prođe kroz mrežni prolaz za aplikacije - drugi put ne postoji.

Filtri za pakete najčešće rade na osnovu tabela koje konfiguriše administrator sistema. Te tabele sadrže prihvatljiva izvorišta i odredišta, blokirana izvorišta i odredišta, kao i načelna pravila rada s paketima koji odlaze drugim računarima ili dolaze od njih.



Slika 8-29. Zaštitna barijera koja se sastoji od dva filtra za pakete i jednog mrežnog prolaza za aplikacije.

U opštem slučaju TCP/IP konfiguracije, izvorište ili odredište su predstavljeni IP adresom i brojem priključka. Priključak određuje zahtevanu uslugu. Na primer, TCP priključak 23 je za telnet, TCP priključak 79 je za finger, a TCP priključak 119 je za dislcusione grupe USENET-a. Kompanija može da blokira dolazne pakete za sve IP adrese u kombinaciji s jednim od ovih priključaka. Na taj način, niko izvan kompanije ne može da se prijavi putem telneta ili da traži osobe sistemskom uslugom Finger. Štaviše, kompanija će efikasnije raditi jer zaposleni neće čitav bogovetni dan čitati poruke diskusionih grupa.

Blokiranje izlaznih paketa je komplikovanije jer kompanije nisu obavezne da se drže standamog numerisanja priključaka. Štaviše, za neke važne usluge, na primer, za FTP, brojevi priključaka se dodeljuju dinamički. Osim toga, iako je blokiranje TCP veza teško, blokiranje UDP paketa je još teže jer se unapred ne zna šta će koji od njih učiniti. Mnogi filtri za pakete se konfigurišu tako da potpuno sprečavaju UDP saobraćaj.

Drugi deo zaštitne barijere je mrežni prolaz za aplikacije (engl. *application gateway)*. Umesto da pregleda sirove pakete, ovaj mežni prolaz radi na nivou aplikacija. Poštanski mrežni prolaz, na primer, može se tako podesiti da pregleda svaku dolaznu ili odlaznu poruku. Za svaku od njih mrežni prolaz odlučuje da li daje propusti ili odbaci na osnovu polja zaglavlja, veličine poruke, čak i na osnovu sadržaja (npr. u vojnim bazama, prisustvo reči „nuklearan“ ili „bomba“ može da izazove preduzimanje posebnih mera).

Svaka instalacija može da ima jedan ili više mrežnih prolaza za različite aplikacije, ali podozrive organizacije često dozvoljavaju samo dvosmerni promet e-pošte i ko- rišćenje Weba, dok sve ostalo zabranjuju kao previše rizično. U kombinaciji sa šifro- vanjem i filtriranjem paketa, takva postavka nudi ograničenu bezbednost po cenu izvesne neudobnosti rada.

Čak i kada se zaštitna barijera savršeno podesi, i dalje ostaju problemi. Na primer, ako se zaštitna barijera podesi da prihvata pakete samo iz određenih mreža (npr. iz drugih pogona kompanije), uljez koji se nalazi izvan barijere može da umetne lažnu izvorišnu adresu da bi zaobišao takvu proveru. Ako neko iznutra želi da napolje pro- krijumčari poverljive dokumente, on može da ih šifraje, čak i da ih fotografiše i da ih pošalje kao JPEG datoteke koje prolaze kroz svaku barijeru koja filtrira reči. Pri tome, nismo još ni pomenuli činjenicu

da 70% svih napada dolazi iznutra, na primer, od nezadovoljnih nameštenika (Schneier, 2000).

Osim toga, postoji čitava klasa napada s kojima zaštitna barijera ne može da se bori. Osnovna svrha zaštitne barijere je da spreči uljeze da uđu u firmu, a tajne daje napuste. Nažalost, postoje osobe koje ne rade ništa drago, već stalno pokušavaju da sruše određene lokacije. Oni na svoju metu šalju legitimne pakete u velikom broju sve dok lokacija ne poklekne pod opterećenjem. Na primer, ako želi da upropasti Web lokciju, napadač joj može poslati TCP paket *SYN* da bi uspostavio vezu. Lokacija će tada re- zervisati mesto u tabeli za vezu i kao odgovor poslati paket *SYN + ACK.* Ako napadač ne odgovori, mesto u tabeli će i dalje tokom nekoliko sekundi ostati rezervisano, sve dok se ne isključi odgo'varajući tajmer. Ako napadač pošalje hiljade zahteva za uspostavljanje veze, rezervisaće sva mesta u tabeli, tako da više neće moći da se uspostavi neka draga - legitimna - veza. Napadi kojima nije cilj krađa podataka, već blokiranje mete zovu se **napadi radi blokiranja usluga** (engl. *Dental of Service, DoS).* Paketi sa zahtevima obično imaju lažne adrese, tako da se napadaču ne može lako ući u trag.

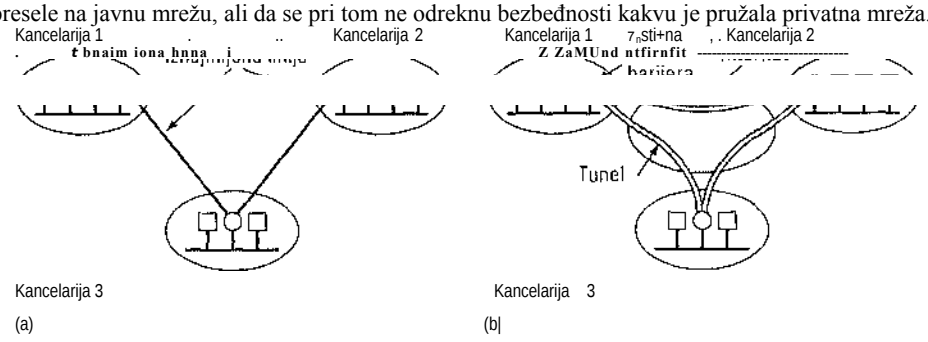
Gora varijanta je ako je napadač već provalio u stotine računara širom sveta i ko- manduje im istovremeni napad na istu metu. U takvoj situaciji napadač ne samo da ima veću moć, već gaje i teže otkriti jer paketi dolaze sa svih strana od računara čiji korisnici ništa ne slute. To je **distribuirani napad radi blokiranja usluga** (engl. *Dis- tributed Dental of Service, DDoS).* Protiv njega gotovo da nema odbrane. Čak i kada napadnuti računar može brzo da prepozna lažni zahtev, treba vremena da se takav zahtev obradi i odbaci; ako u sekundi stiže dovoljan broj zahteva, mikroprocesor će se sve vreme baviti njima.

8.6.3 Virtuelne privatne mreže

U mnogim kompanijama su kancelarije i pogoni raštrkani u više gradova, ponekad i u više zemalja. U starim danima, pre nego što su se pojavile javne mreže za prenos podataka, takve kompanije su često iznajmljivale telefonske linije koje su povezivale samo neke ili sve njene lokacije. Neke kompanije još uvek tako rade. Mreža sastavljena od računara kompanije i iznajmljenih telefonskih linija zove se **privatna mreža.** Na slici 8-30(a) vidite privatnu mrežu koja povezuje tri lokacije.

Privatne mreže rade vrlo dobro i bezbedno. Ako u njima postoje samo iznajmljene linije, nikakav saobraćaj ne može da se probije izvan lokacija kompanije i uljez mora da se fizički zakači za liniju da bi provalio u mrežu, što nije lako. Problem kod privatnih mreža je cena jer mesečna naknadna za liniju TI iznosi na hiljade dolara, a za liniju T3 mnogostruko više. Kada su se pojavile javne mreže za prenos podataka, a

kasnije i Internet, mnoge kompanije su poželele da svoj saobraćaj podataka (možda i govorni) presele na javnu mrežu, ali da se pri tom ne odreknu bezbeđnosti kakvu je pružala privatna mreža.



Slika 8-30. (a) Privatna mreža sa iznajmljenim linijama, (b) Virtuelna privatna mreža.

Takva potreba je ubrzo dovela do virtuelnih privatnih mreža (engl. *Virtual Plivate Networks, VPNs)* koje se uspostavljaju preko javnih mreža, ali imaju svojstva slična svojstvima privatnih mreža. One nose atribut „virtuelne“ jer su samo apstrakcija, kao što ni virtuelna kola nisu prava kola, niti je virtuelna memorija prava memorija.

Iako se VPN mreže mogu realizovati i preko ATM mreža (ili mreža za štafetni pre- nos okvira), sve je popularnija težnja da se one izgrađuju direktno na Internetu. Uobičajeno je da se svaka kancelarija oprema zaštitnom barijerom i kancelarije povezuju peko Interneta tunelima, kao na slici 8-30(b). Ako se za prenos podataka tunelom koristi IPsec, tada se sav saobraćaj između para kancelarija može objediniti u šifrovanu SA vezu s proverom identiteta, čime se postiže integritet, tajnost, čak i znatna otpornost na analizu saobraćaja.

Kada se sistem uspostavi, svaki par zaštitnih barijera mora da dogovori parametre međusobne SA veze, uključujući usluge, radne režime, algoritme i ključeve. Mnoge zaštitne barijere imaju ugrađene VPN mogućnosti, iako to mogu da rade i neki obični usmerivači. Međutim, pošto se zaštitne barijere uglavnom bave bezbednošću, prirodno je da tuneli počinju i završavaju se u njima, ostvarujući jasan prelaz između kompanije i Interneta. Na taj način, virtuelne privatne mreže i IPsec sa šemom ESP u tunelskom režimu rada predstavljaju prirodno sklopljenu kombinaciju koja se široko koristi u praksi.

Kada se uspostavi SA veza, saobraćaj može da krene. Za usmerivač na Internetu, paket koji putuje VPN tunelom predstavlja najobičniji paket. Od običnog paketa ga razlikuje jedino prisustvo IPsec zaglavlja iza IP zaglavlja, ali pošto dodatna zaglavlja ne utiču na proces prosleđivanja, usmerivači o njima ne vode brigu.

Organizovanje VPN na opisani način ima tu ključnu prednost što je VPN mreža potpuno nevidljiva korisničkom softvera. Zaštitne barijere uspostavljaju SA veze i rade s njima. Jedina osoba koja zna da postoji takva organizacija je administrator sistema, koji treba da podesi i održava zaštitne barijere. Svakom drugom korisniku sve izgleda kao povratak na staru dobru iznajmljenu telefonsku liniju. Više detalja o vir- tuelnim privatnim mrežama saznaćete kod Browna (1999) i Izzoa (2000).

8.6.4 Bezbednost bežičnih mreža

Iznenađujuće je lako napraviti logički potpuno bezbedan sistem uz korišćenje VPN mreža i zaštitnih barijera, ali je u praksi takav sistem bušan kao sito. Do takve situacije može doci ako su neki računari povezani bežično i koriste radio-komunika- ciju koja zaobilazi zaštitnu barijeru u oba smera. Mreže 802.11 često imaju domet od nekoliko stotina metara, tako da neko ko poželi da špijunira kompaniju može rano izjutra da na službeni parking kompanije doveze automobil u kome će ostaviti bežično povezan prenosivi računar koji će zabeležiti sve što čuje u toku dana. Predveče će njegov čvrsti disk biti prepun vrednih informacija. Ovakva mogućnost se teorijski ne pretpostavlja. U teoriji se takođe ne pretpostavlja da ljudi obijaju banke.

Veliki deo bezbednosnih problema potiče od baznih stanica za bežični prenos (pristupnih tačaka) jer proizvođači teže da ih napave tako da se što lakše koriste. Čim korisnik raspakuje takav uređaj i uključi ga u električnu mrežu, on obično odmah počne da radi - i skoro uvek bez ikakvih mera bezbednosti, emitujuci tajne svima i svakome u radio dometu. Ako ga korisnik zatim priključi na Ethernet, sav saobraćaj u Ether- netu odjednom će postati dostupan i svima na službenom parkingu. Bežični prenos je nešto o čemu njuškala često sanjare: „Ala volim ovaj režim, tajne stižu dok ja ležim“. Treba li uopšte još naglašavati da je bezbednost čak važnija za bežične sisteme nego za ožičene? U ovom odeljku ćemo razmotriti neke načine pomoću kojih se obezbeđu- ju bežične mreže. Dopunska obaveštenja možete naći kod Nicholsa i Leklcasa (2002).

Bezbednost u mrežama 802.11

Standard 802.11 propisuje bezbednosni protokol zvan privatnost kao u kablovskoj mreži (engl. *Wired. Equivalent Privacy, WEP*) na nivou sloja veze podataka, Kao što mu ime kaže, taj protokol treba da bežičnim lokalnim mrežama ponudi obezbe- đenje jednako obezbeđenju ožičenih lokalnih mreža. Pošto je podrazumevano obez- beđenje ožičenih lokalnih mreža ravno nuli, taj cilj je lako postići, a WEP ga i postiže, kao što ćete se ubrzo i sami uveriti.

Kada se aktivira sistem bezbednosti mreže 802.11, svaka stanica dobija tajni ključ koji deli s baznom stanicom. Način distribuiranja ključeva nije propisan standardom. Njih može fabrički da ugradi i proizvođač. Oni se mogu i unapred razmeniti ožiče- nom mrežom. Najzad, svaka bazna stanica ili korisnički računar mogu da izaberu na- sumičan ključ, da ga šifruju javnim ključem druge strane i da joj ga pošalju bežičnim putem. Kada se uspostave, ključevi se obično ne menjaju mesecima ili godinama.

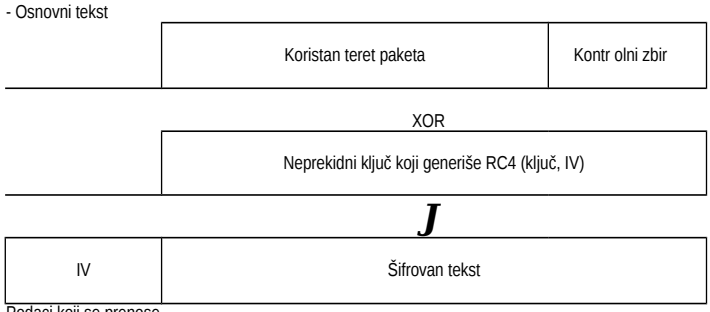
WEP radi u režimu uzastopnog šifrovanja zasnovanom na algoritmu RC4. Taj algoritam je smislio Ronald Rivest i držao ga u tajnosti sve dok nekako nije procurio u javnost i 1994. bio objavljen na Internetu. Kao što smo ranije istakli, tajnost algoritma je skoro nemoguće očuvati, čak i u cilju zaštite intelektualne svojine (što je ovde bio slučaj), a kamoli u cilju obezbeđivanja kroz prikrivanje (što nije bio cilj kod algoritma RC4). U protokolu WEP, algoritam RC4 generiše neprekidni ključ koji se podvrgava isključivoj disjunkciji sa osnovnim tekstom, dajući šifrovan tekst.

Koristan teret svakog paketa šifruje se metodom prikazanom na slici 8-31. Prvo se napravi kontrolni zbir podataka pomoću CRC-32 polinoma i taj zbir doda podacima, što sve zajedno služi kao osnovni tekst za šifrovanje. Takav osnovni tekst se podvrgava isključivoj disjunkciji s delom neprekidnog ključa iste veličine da bi se dobio šifrovan tekst.

Inicijalizacioni vektor (IV) uz koji počinje RC4 šalje se zajedno sa šifrovanim tekstom. Kada primalac dobije paket, vadi iz njega šifrovan korisni teret, generiše neprekidni ključ pomoću deljenog tajnog ključa i IV koji je upravo primio, i podvrgava neprekidni ključ i koristan teret isključivoj disjunkciji da bi dobio osnovni tekst. On tada može da proveri kontrolni zbir i u veri se da paket niko nije čačkao.

Iako opisani postupale obezbeđivanja na prvi pogled izgleda u redu, metod za njegovo provaljivanje već je objavljen (Borisov i sar., 2001). U nastavku ćemo sumirati njihove rezultate. Pre svega, postoji iznenađujuće mnogo instalacija u kojima se isti ključ deli svim korisnicima, što znači da svaki korisnik može da čita saobraćaj svakog drugog korisnika. To sigurno liči na Ethernet, ali nije baš bezbedno.

Međutim, čak i kada bi svaki korisnik imao drugačiji ključ, WEP bi se još uvele mogao uspešno napasti.



Slika 8-31. Šifrovanje paketa pomoću protokola WEP.

Pošto se ključevi u načelu ne menjaju tokom dužeg vremena, WEP standard preporučuje (ali ne obavezuje) da se inicijalizacioni vektor menja pri slanju svakog paketa kako bi se izbegao napad ponavljanjem poruka o kome smo govorili u odeljku
8.2.3. Nažalost, mnoge 802.11 kartice za prenosive računare, kada se utaknu u raču- nar, vraćaju vrednost inicijalizacionog vektora na nulu, a zatim je povećavaju za jedan pri svakom poslatom paketu. Pošto korisnici često vade i vraćaju ove kartice, česti su paketi s niskim vrednostima IV. Ako Trudi od istog korisnika može da sakupi više paketa sa istom IV vrednošću (koja se kao osnovni tekst šalje uz svaki paket), ona može da izvrši isključivu disjunkciju dve takve vrednosti i da verovatno na taj način razbije šifru.

Međutim, čak i kada 802.11 kartice biraju nasumičnu IV vrednost za svaki paket, te vrednosti su 24-bitne, pa se posle 2^{24} poslatih paketa moraju ponovo koristiti. Gore je to što se pri nasumičnom biranju IV vednosti ponavljanje iste IV vrednosti očekuje posle samo 5000 poslatih paketa, što sledi iz opisa rođendanskog napada u ođeljku
8.4.4. Na taj način, ako Trudi osluškuje samo nekoliko minuta, skoro sigurno će uloviti dva paketa sa istim IV i istim ključem. Podvrgavajući šifrovane tekstove isključivoj disjunkciji, ona može da dobije rezultat isključive disjunkcije osnovnih tekstova. Do- bijenu sekvencu bitova može da napadne na različite načine da bi dobila sam osnovni tekst. Uz nešto više truda, može se dobiti i neprekidni ključ za taj IV. Trudi može da nastavi ovako još neko vreme i da sakupi „rečnik“ neprekidnih ključeva za različite IV vrednosti. Kada se jednom provali jedna IV vrednost, mogu se potpuno dešifrovati svi paketi koji će se slati uz taj vektor, ali i oni koji su već uz njega poslati.

Štaviše, pošto se inicijalizacioni vektori biraju na slučajan način, kada Trudi jednom utvrdi važeći par (IV, neprekidni ključ), ona ga može upotrebiti za generisanje svojih paketa i tako aktivno uticati na komuniciranje. Primalac bi teorijski mogao da primeti da odjednom veliki broj pristiglih paketa ima istu IV vrednost, ali (1) WEP to dopušta i (2) niko na to ne obraća pažnju.

Recimo na kraju da CRC nije od velike koristi pošto Trudi može da izmeni koristan teret i istovremeno da odgovarajuće izmeni CRC, a da čak i ne dešifruje paket. Ukratko, sistem bezbednosti mreže 802.11 lako se obara, a imajte u vidu da nismo još ni nabrojali sve napade koje su opisali Borisov i saradnici.

Avgusta 2001, mesec dana postoje prikazan rad Borisova i saradnika, objavljenje drugi katastrofalan napad na WEP (Fluhrer i sar., 2001). On je iskoristio jednu slabost u samom algoritmu RC4. Fluhrer i saradnici su utvrdili da se neki bitovi mnogih ključeva po pravilu mogu izvesti iz neprekidnog ključa. Ako se takav napad uzastopno ponavlja, može se otkriti ceo ključ uz srazmerno malo truda. S obzirom da im je cilj bio akademske prirode, Fluhrer i saradnici nisu stvarno pokušali da provale ni u jednu mrežu 802.11.

Za razliku od njih, kada su jedan student na letnjem kursu i dva istraživača iz AT&T Laboratorija saznali za napad koji su opisali Fluhrer i saradnici, odlučili su da ga isprobaju uživo (Stubblefield i sar., 2002). Za samo sedam dana provalili su prvi 128-bitni ključ jedne industrijske, lokalne 802.11 mreže, pri čemu su najveći deo sedmice proveli u traženju najjeftinije 802.11 kartice, u traženju dozvole da je kupe, u njenom instaliranju i isprobavanju. Za programiranje im je trebalo ciglih dva sata.

Kada su saopštili svoje rezultate, CNN je objavio prilog pod naslovom „Novopečeni hakeri razbijaju bežične šifre“, u kome su neki mudraci iz industrije pokušali da omalovaže njihove rezultate argumentacijom daje to što su uradili bilo sasvim jednostavno posle objavljivanja rezultata Fluhrera i saradnika. Iako je takav argument u tehničkom smislu ispravan, ostaje činjenica da su udruženi napori pomenuta dva tima obelodanili fatalni propust u WEP-u i mreži 802.11.

Sedmog septembra 2001. godine, povodom vesti daje WEP potpuno provaljen, IEEE je dao kratku izjavu u šest tačalca koje se mogu sažeti u sledeće:

1. Upozorili smo vas da bezbednost WEP-a nije bolja od bezbednosti Etherneta.
2. Mnogo je gore ako se sistem bezbednosti uopšte ne aktivira.
3. Pokušajte s drugim sistemima bezbednosti (npr. u transportnom sloju).
4. Sledeća verzija, 802.11 i, imaće bolje obezbeđenje.
5. Za buduće sertificiranje biće obavezna upotreba mreže 802.1 li.
6. Trudićemo se da pronađemo neko rešenje dok se ne završi projekat mreže 802.11 i.

Čitavu priču smo okitili mnogim detaljima samo da bismo istakli činjenicu da postizanje prave bezbednosti nije lako, čak ni za stručnjake.

Bezbednost sistema Bluetooth

Sistem Bluetooth ima znatno manji domet od mreže 802.11, pa se ne može napasti sa parkirališta, ali i on ima svoje bezbednosne probleme. Zamislite, na primer, da je Alisin računar opremljen bežičnom Bluetooth tastaturom. U odsustvu obezbeđenja, kad bi se Trudi nalazila u susednoj kancelariji, ona bi mogla da čita sve što Alisa kuca, uključujući i e-poštu koju šalje. Ona bi mogla da ulovi i sve što Alisin računar šalje Bluetooth štampaču ako bi sedela pokraj njega (npr. dolaznu poštu i poverljive iz- veštaje). Srećom, Bluetooth ima

razrađenu bezbednosnu šemu koja pokušava da omete sve svetske Trudi. Sledi njen opis.

Bluetooth ima tri bezbednosna režima, počev od ničega, pa do potpunog šifrovanja podataka i kontrole integriteta. Kao u mreži 802.11, ako je bezbednosni sistem isključen (podrazumevano), nema obezbeđenja. Većina korisnika koristi podrazumevanu opciju sve dok ne zapadnu u nevolju: zatim uključuju bezbednosni sistem (zatvaraju vrata kad je zec već utekao).

Bluetooth nudi bezbednost u više slojeva. U fizičkom sloju, skokovito menjanje frekvencija nudi izvesnu bezbednost, ali pošto svakom Bluetooth uređaju koji menja elementarnu mrežu (pikonet) mora da se saopšti sekvenca menjanja frekvencija, ta sekvenca očito nije tajna. Stvarna bezbednost počinje od tačke kada nov sporedni čvor zahteva kanal za vezu sa glavnim čvorom. Od dva uređaja se očekuje da dele tajni ključ koji je unapred dogovoren. U nekim slučajevima, ključ je u oba uređaja fabrički ugrađen (npr. za slušalice i mobilni telefon koji se prodaju u kompletu). U drugim slučajevima, jedan uređaj (npr. slušalice) ima ugrađen ključ, a korisnik taj ključ treba da upiše u drugi uređaj (npr. u mobilni telefon) kao decimalan broj. To su **osnovni opšti ključevi** (engl. *passkeys).*

Da bi uspostavili kanal za vezu, sporedni i glavni čvor uzajamno proveravaju da li druga strana zna osnovni opšti ključ. Ako ga oba čvora znaju, dogovaraju se da li će kanal biti šifrovan, da li će se kontrolisati integritet podataka, ili ijedno i drugo. Zatim nasumično biraju 128-bitni ključ sesije, čijih nekoliko bitova mogu biti i javni. Ovakvo oslabljivanje ključa preduzima se radi usaglašavanja s propisima različitih zemalja kojima se ograničava izvoz ili korišćenje ključeva dužih od onoga što vlada dotične zemlje može da dešifruje.

Za šifrovanje se koristi uzastopna šifra E_0; za kontrolu integriteta koristi se SAFER+. Obe su klasične simetrične blok šifre. Šifra SAFER+ je konkurisala za AES, ali je odbačena u prvom krugu jer je bila sporija od drugih kandidata. Sistem Bluetooth je dovršen pre nego stoje izabrana AES šifra; daje bilo drugačije, verovat- no bi se koristio algoritam Rijndael.

Sam postupale uzastopnog šifrovanja prikazanje na slici 8-14, gde se osnovni tekst podvrgava isključivoj disjunkciji s neprekidnim ključem dajući šifrovan tekst. Nažalost, i E_0 (slično algoritmu RC4) možda ima fatalne slabosti (Jackobsson i Wetzel, 2001). Iako šifra još nije razbijena, njene sličnosti sa šifrom A5/1, čiji katastrofalni propusti ugrožavaju sav telefonski GSM saobraćaj, daju povoda za brigu (Biryukov i sar., 2000). Svi se ponekada zapanje (uključujući i autora) kad uvide da u večitoj igri mačke i miša između kriptografa i kriptoanalitičara, češće pobeđuju ovi drugi.

Kod Bluetootha je problem i to što se potvrđuje samo identitet uređaja, a ne i korisnika, tako da lopov, kada ukrade Bluetooth uređaj, ima otvoren pristup finansijskim i dragim dokumentima njegovog legalnog korisnika. Međutim, Bluetooth ugrađuje bezbednost i u više slojeve, pa kada se ona naruši u sloju veze, ipak ostaje nešto očuvane bezbednosti, naročito za aplikacije koje za dovršavanje određene transakcije zahtevaju unošenje PIN koda preko neke vrste tastature.

Bezbednost mreža WAP 2.0

Tvorci WAP-a su na protokolu WAP 1.0 naučili lekciju da ne treba koristiti nestandardan skup protokola. Zbog toga se u WAP-u 2.0 u svim slojevima uglavnom koriste standardni protokoli. Nije izuzeta ni bezbednost. Pošto je zasnovana na protokolu IP, potpuno podržava IPsec u mrežnom sloju. U transportnom sloju se TCP veze mogu zaštititi pomoću IETF standarda TLS koji ćemo proučiti u nastavku ovog poglavlja. U sloju iznad ovog koristi se HTTP provera identiteta klijenta, prema RFC dokumentu 2617. Kriptobiblioteke sloja aplikacija brinu o kontroli integriteta i nemogućnosti poticanja. Sve u svemu, pošto se WAP 2.0 zasniva na opštepoznatim standardima, postoje šanse da njegove bezbednosne usluge, naročito privatnost, provera identiteta, kontrola integriteta i nemogućnost poricanja rade bolje nego u mrežama 802.11 i Bluetooth.

8.7 PROTOKOLI ZA PROVERU IDENTITETA

Provera identiteta (engl. *authentication)* je tehnika pomoću koje proces prove- rava da li je partner s kojim komunicira zaista ono što tvrdi da jeste ili neki prevarant. Proveriti identitet udaljenog procesa u prisustvu zlonamernog, aktivnog uljeza veoma je teško i zahteva složene protokole zasnovane na šifrovanju. U ovom odeljku ćemo proučiti neke od mnogih protokola za proveru identiteta koji se koriste u neobez- beđenim računarskim mrežama.

Pomenimo uzgred da neke osobe mešaju pojmove ovlašćivanja (engl. *authorization)* i provere identiteta. Provera identiteta se odnosi na pitanje da li zaista komunicirate sa određenim procesom. Ovlašćivanje se tiče onoga što je tom procesu dozvoljeno da radi. Na primer, klijentski proces stupa u vezu sa serverom datoteka i kaže: Ja sam Skotov proces i želim da obrišem datoteku *cookbook.old.* S gledišta ser- vera postavljaju se dva pitanja:

1. Da li je to stvarno Skotov proces (provera identiteta)?
2. Da lije Skot ovlašćen da obriše datoteku *cookbook.old* (ovlašćivanje)?

Zahtevana akcija može da se izvrši tek kada server na oba pitanja dobije nesumnjivo potvrdan odgovor. Prvo pitanje je uistinu ključno. Kad server konačno sazna s kim razgovara, provera ovlašćenja je samo stvar pretraživanja odrednica u lokalnim tabelama ili bazama podataka. Zbog toga čemo se u ovom odeljku koncentrisati na proveru identiteta.

Svi protokoli za proveru identiteta rade prema istom opštem modelu. Alisa počinje komunikaciju tako što šalje poruku Bobu ili **Centru za distribuiranje ključeva** (engl. *Key Distribution Center, KDC),* kome se može ukazati poverenje. Sledi razme- njivanje više poruka u raznim pravcima. U toj fazi Trudi može da presretne poruku, daje izmeni ili daje pošalje ponovo, da bi prevarila Alisu i Boba ili samo da bi pravila gužvu.

Bez obzira na sve, kada se protokol završi, Alisa je sigurna da govori s Bobom, a Bob je siguran da govori sa Alisom, Štaviše, oni će u većini protokola uspostaviti i **ključ sesije** (engl. *session key*) koji će koristiti u konverzaciji koja sledi. U praksi se zbog poboljšanja performansi sav saobraćaj podataka šifruje simetričnim ključem (najčešće se koriste algoritmi AES ili trostruki DES), iako se u samim protokolima za provera identiteta i pri uspostavljanju ključa sesije obično koristi šifrovanje javnim ključem.

Korišćenje uvele novog, nasumice izabranog ključa sesije za svaku novu vezu, treba da minimizuje obirn saobraćaja šifrovanog korisničkim tajnim ili javnim ključevima, da smanji količinu šifrovanog teksta koga može da se domogne uljez i da umanji štetu u slučaju kada proces otkaže, a njegova sadržina padne u pogrešne ruke. U takvim slučajevima, napadač se najverovatnije može domoći samo ključa sesije. Kada se jednom uspostavi sesija, sve stalne ključeve treba brižljivo anulirati.

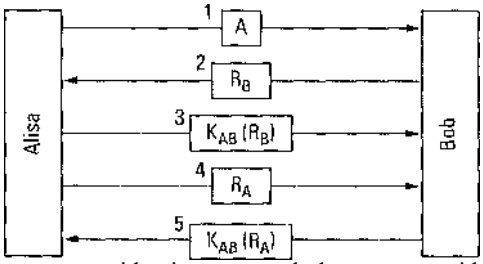
8.7.1 Provera identiteta **zasnovana** na **deljenom** tajnom **ključu**

U prvom protokolu za provera identiteta koji ćemo obraditi, pretpostavićemo da Alisa i Bob već dele tajni ključ K_{AB}. Taj deljeni ključ može da bude unapred dogovoren telefonom ili u ličnom kontaktu, ali nikada preko (neobezbeđene) mreže.

Ovaj protokol se zasniva na principu koji koriste mnogi protokoli za proveru identiteta: jedna strana šalje nasumično odabran broj drugoj strani koja ga transformiše na poseban način i vraća rezultat prvoj strani. To su **protokoli za proveru identiteta testiranjem** (engl. *challenge-response protocols).* U ovom, a i u narednim protokolima za proveru identiteta, koristićemo sledeće označavanje:

A, B su identiteti Alise i Boba.
R_i su pozivne poruke (ponuđene za testiranje), gde indeks označava pozivaoca.
K_i su ključevi, gde *i* označava vlasnika.
K_s je ključ sesije.

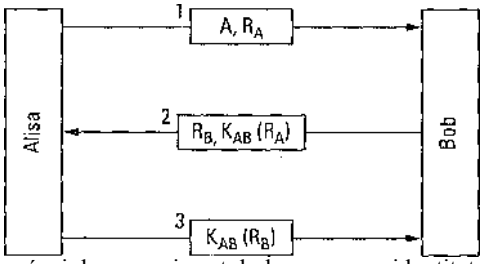
Redosled razmenjivarrja poruka u našem prvom protokolu za provera identiteta prikazan je na slici 8-32. U poruci 1, Alisa šalje Bobu svoj identitet A na način koji Bob može da razume. Bob, naravno, ne može da zna da li ta poruka dolazi od Alise ili od Trudi, pa bira pozivnu poruku - veliki nasumično odabran broj R_B~ i šalje ga „Alisi“ u poruci 2 kao osnovni tekst. Nasumično odabrani brojevi koji se u protokolima za proveru identiteta testiranjem koriste samo jednom, zovu se **jednokratni uzorci** (engl. *nonces).* Alisa tada šifruje pozivnu poruku ključem koji deli s Bobom i šalje mu šifrovan tekst $K_{AB}(R_B)$ kao poruku 3. Kad Bob vidi ovu poruku, on odmah zna da je ona od Alise pošto Trudi ne zna K_{AB} i zato je nije mogla ona generisati. Staviše, pošto je R_B izabran na slučajan način iz velikog skupa (npr. 128-bitnih slučajnih brojeva), mala je verovatnoča da je Trudi ulovila R_B i odgovor na njega u nekoj od prethodnih sesija. Isto je toliko neverovatno da Trudi pogodi ispravan odgovor na bilo koju pozivnu poruku.



Slika 8-32. Dvosmerna provera identiteta u protokolu za proveru identiteta testiranjem.

U ovom trenutku, Bob je siguran da razgovara sa Alisom, ali Alisa nije sigurna ni u šta. Ona misli da je možda Trudi presrela poruku 1 i poslala joj odgovor R_B. Možda se Bob noćas preselio na onaj svet. Da bi utvrdila s kim razgovara, Alisa bira slučajan broj R_A i šalje ga Bobu u obiku osnovnog teksta u poruci 4. Kada Bob odgovori sa $K_{AB}(R_a)$, Alisa zna da razgovara baš s Bobom. Ako njih dvoje sada žele da uspostave ključ sesije, Alisa bira ključ K_s i šalje ga Bobu šifrovanog ključem K_{AB}.

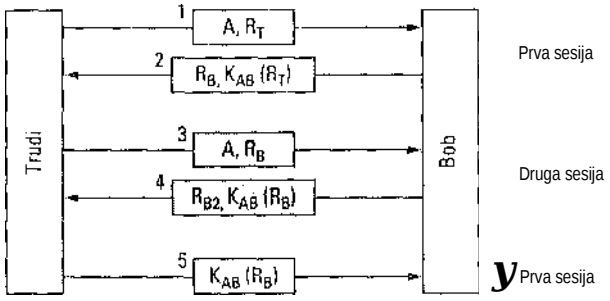
Protokol na slici 8-32 obuhvata čak pet poruka. Uključimo mozak i pokušajmo da taj broj smanjimo. Jedno rešenje je prikazano na slici 8-33. Tu Alisa inicira protokol za proveru identiteta testiranjem ne čekajući da Bob to učini. Isto tako, zajedno sa odgovorom na Alisin poziv, Bob šalje svoj poziv. Čitav protokol se sa pet poruka svodi na tri.



Slika 8-33. Skraćeni dvosmemi protokol za proveru identiteta testiranjem.

Da lije ovaj novi protokol bolji od prvobitnog? U izvesnom smislu jeste: kraći je. Nažalost, takođe je i pogrešan. U određenim slučajevima, Trudi može da ga izigra primenjujući **napad odbijanjem** (engl. *reflection attack).* Takva situacija nastaje ako se s Bobom istovremeno može voditi više sesija, a to će biti moguće ako je Bob, na pri- mer, banka koja je unapred pripremljena da prihvati mnoge istovremene veze s ban- komatima.

Trudin napad odbijanjem prikazan je na slici 8-34. On počinje tako što Trudi šalje ^tvrdeći daje Alisa. Bob, kao i obično, odgovara svojim pozivom R_B. Sada je Trudi na muci jer ne može da izračuna $K_{AB}(R_B)$. Šta može da uradi?



Slika 8-34. Napad odbijanjem.

Ona može da otvori drugu sesiju porukom 3, upućujući kao svoj poziv R_B preuzet iz poruke 2. Bob će to ćutke šifrovati i poslati joj $K_{AB}(R_B)$ u poruci 4. Na slici smo za- senčili poruke druge sesije da bismo ih razlikovali. Sada Trudi ima informaciju koja joj je nedostajala, pa može da završi prvu sesiju i da prekine drugu. Bob je sada ubeđen da je Trudi Alisa, pa kada ga ova pita za stanje Alisinog računa, on joj ga daje bez komentara. A zatim, kada mu ona nalaže da novac s njega prebaci na tajni račun u Švajcarskoj, on i to čini bez trenutka premišljanja.

Pouka:

Projektovanje nepogrešivog protokola za proveru identiteta teže je nego što izgleda.

Cesto pomažu sledeća četiri opšta pravila:

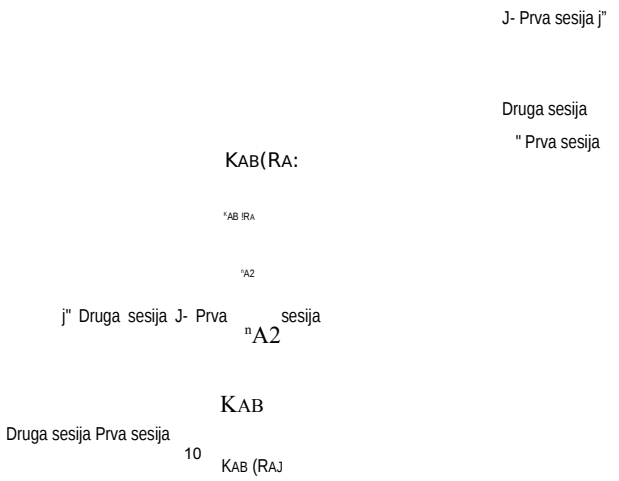
1. Neka svoj identitet dokaže najpre inicijator, pa onda druga strana. U gornjem primeru, Bob je saopštio osetljive informacije pre nego što je od Alise dobio bilo kakav dokaz o njenom identitetu.
2. Neka inicijator i druga strana koriste različite ključeve za dokazivanje identiteta, makar to značilo da treba da imaju dva deljena ključa: K_{AB} i K'_{AB}.
3. Neka inicijator i druga strana izvlače brojeve za svoje pozivne poruke iz različitih skupova slučajnih brojeva. Na primer, neka inicijator koristi skup parnih brojeva, a druga strana - skup neparnih.
4. Ojačajte protokol protiv napada koji uključuju drugu paralelnu sesiju, kod kojih se podaci iz jedne sesije koriste u drugoj.

Ako se naruši bilo koje od pobrojanih pravila, protokol se često može provaliti. U našem primeru su narušena sva četiri pravila, pa su i posledice bile katastrofalne.

Vratimo se sada ponovo na sliku 8-32. Je li sigurno da se on ne može napasti odbijanjem? Na to pitanje uopšte nije jednostavno odgovoriti. Trudi je mogla da porazi

naš protokol napadajući ga odbijanjem zato što je mogla da otvori drugu sesiju s Bobom i da ga navede da sam odgovara na svoja pitanja. Šta bi se desilo kada Alisa ne bi bila jedini korisnik, već je to računar opšte namene koji talcođe prihvata više istovremenih sesija? Razmotrimo šta u tom slučaju Trudi može da uradi.

Pogledajte sliku 8-35. Alisa počinje tako što svoj identitet objavljuje u poruci 1. Trudi presreće ovu poruku i započinje sopstvenu sesiju porokom 2, tvrdeći daje Bob. I ovde smo zasenčili poruke sesije 2. Na poroku 2 Alisa odgovara porokom 3: Tvrdiš da si Bob? Dokaži to. Tu je Trudi uhvaćena jer ne može da dokaže da je Bob.



Slika 8-35. Napad odbijanjem na protokol sa slike 8-32.

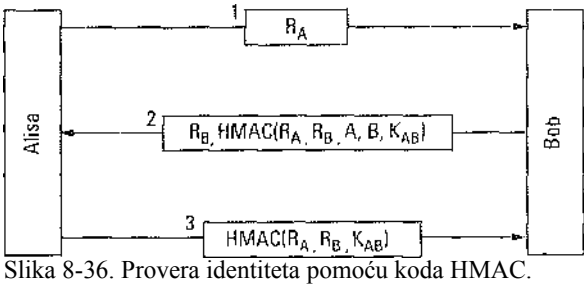
Šta radi Trudi? Vraća se u prvu sesiju, gđe je njen red za slanje poziva, i šalje R_A koji je dobila u poroci 3. Alisa na to ljubazno odgovara porokom .5, snabdevajući Trudi informacijom koja joj treba za slanje poroke 6 u sesiji 2. U ovom trenutku, Trudi je na konju jer je uspešno odgovorila na Alisin poziv u sesiji 2. Ona sada može da poništi sesiju 1, da pošalje bilo koji stari broj za ostatak sesije 2 i imaće potpuno potvrđen identitet u sesiji 2 sa Alisom.

Trudi je, međutim, prava gadura i želi da stvar zamuti do kraja. Umesto da slanjem nekog starog broja dovrši uspostavljanje sesije 2, ona čeka da Alisa pošalje poruku 7 - poziv za sesiju 1. Trudi, naravno, ne zna kako da odgovori, pa ponovo napada odbijanjem, šaljući R_{A2} kao poroku 8. Alisa šifroje R_{A2} i rezultat šalje porokom 9. Trudi se sada vraća u sesiju 1 i Alisi vraća željeni broj porukom 10, kopirajući ga iz Alisine poruke 9. U tom trenutku, Trudi ima dve sesije sa Alisom s potpuno potvrđenim identitetom.

Ovaj napad postiže nešto drugačiji rezultat od napada na protokol s tri poruke, prikazan na slici 8-34. Ovoga puta Trudi ima dve potvrđene veze sa Alisom. U prethodnom primeru imala je jednu takvu vezu s Bobom. Ponovo ističemo sledece: da smo

ovde primenili sva četiri opšta pravila o kojima smo ranije govorili, napad bi se mogao zaustaviti. Detaljno razmatranje napada ovakve vrste i načina borbe protiv njih možete nadi kod Birda i saradnika (1993). Tamo ćete naci i uputstvo za sistematsko projektovanje protokola koji su dokazano otporni na napade. Međutim, i najjednostavniji takav protokol prilično je složen, pa ćemo zato preci na drugačiju klasu protokola koji ipak dobro rade.

Nov protokol za proveru identiteta prikazanje na slici 8-36 (Bird i sar., 1993). U njemu se koristi kod HMAC sličan onom koji smo upoznali pri analiziranju standarda IPsec. Alisa počinje tako što Bobu šalje jednokratni uzorak R_A u poruci 1. Bob odgovara slanjem svog jednokratnog uzorka R_B, zajedno s kodom HMAC. HMAC je tako oblikovan da gradi strukturu podataka sastavljenu od Alisinog jednokratnog uzorka, Bobovog jednokratnog uzorka, njihovih identiteta i deljenog tajnog ključa K_{AB}. Ta struktura podataka se zatim hešira u HMAC, na primer, pomoću algoritma SHA-1. Kada Alisa primi poruku 2, ona tada ima R_A (koji je sama izabrala), R_B koji stiže kao osnovni tekst, dva identiteta i tajni ključ K_{AB} koji je oduvek znala, tako da i sama može da izračuna HMAC. Ako se dva HMAC koda poklope, ona zna da razgovara s Bobom pošto Trudi ne zna K_{AB}, pa ne može da pošalje ispravan HMAC. Alisa odgovara Bobu kodom HMAC koji sadrži samo dva jednokratna uzorka.



Slika 8-36. Provera identiteta pomoću koda HMAC.

Može li Trudi da na neki način prevari ovaj protokol? Ne može, zato što nijednu stranu ne može da natera da šifruje ili hešira vrednost koju je ona izabrala, kao što se desilo na slikama 8-34 i 8-35. Oba HMAC koda sadrže vrednosti koje biraju dve legalne strane veze, a to je nešto na šta Trudi ne može da utiče.

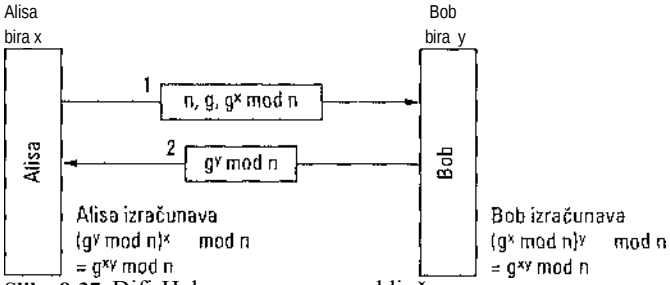
Korišćenje HMAC koda samo je jedna od više realizacija gornje ideje. Umesto da se izračunava HMAC kod niza elemenata, često se elementi šifruju sekvencijalno, ulančavanjem blok-šifara.

8.7.2 Uspostavljanje deljenog ključa: Difi-Helmanova razmena ključa

Sve do sada smo pretpostavljali da Alisa i Bob dele tajni ključ. Pretpostavimo sada da ga ne dele (zato što još uvek nema prihvaćene PKI infrastrukture za potpisivanje i distribuiranje sertifikata). Kako onda njih dvoje mogu da uspostave ključ? Alisa bi mogla da pozove Boba i saopšti inu svoj ključ telefonom, ali bi on verovatno rekao: Kako da znam da si Alisa, a ne Trudi? Mogli bi da ugovore i sastanak, na koji bi poneli svoje lične karte, vozačke dozvole, po tri kreditne kartice, ali, budući da su oboje veoma zauzeti, moglo bi se dogoditi da o vremenu sastanka ne mogu da se dogovore mesecima. Na sreću, ma kako to zvučalo neverovatno, postoji način da totalni stranci uspostave tajni deljeni ključ usred dana, čak i ako Trudi pažljivo beleži svaku razme- njenu poruku.

Protokol za uspostavljanje tajnog ključa između osoba koje se međusobno uopšte ne poznaju zove se **Difi-Helmanova razmena ključa** (engl. *Diffie-Hellman key exchange),* prema njegovim autorima (Diffie i Hellman 1976). Ona radi na sledeći način. Alisa i Bob treba da dogovore dva velika broja, *n* i *g,* pri čemu je *n* prost broj, *(n* — l)/2 takođe, a *g* se bira na sličan način.. Ti brojevi nisu tajna, pa ih jedno drugom mogu javno saopštiti. Sada Alisa bira veliki (recimo, 512-bitni) broj *x* i pažljivo ga skriva. Na isti način, Bob bira veliki tajni broj *y.*

Alisa započinje protokol za razmenu ključa tako što Bobu šalje poruku koja sadrži *n, g* i g^x mod *n,* kao na slici 8-37. Bob odgovara Alisi tako što joj šalje poruku koja sadrži g^y mod *n.* Sada Alisa diže broj koji joj je poslao Bob na x-ti stepen i nalazi njegov ostatak posle celobrojnog deljenja sa *n: (*g^y mod *nf* mod *n.* Bob obavlja sličnu operaciju: *(*g^x mod *n)*^{y} mod *n.* Prema pravilima aritmetike izračunavanja ostatka celobrojnog deljenja, oboje kao rezultat treba da dobiju g^{xy} mod *n.* I, eto ga: Alisa i Bob najednom dele tajni ključ, g^{xy} mod *n.*

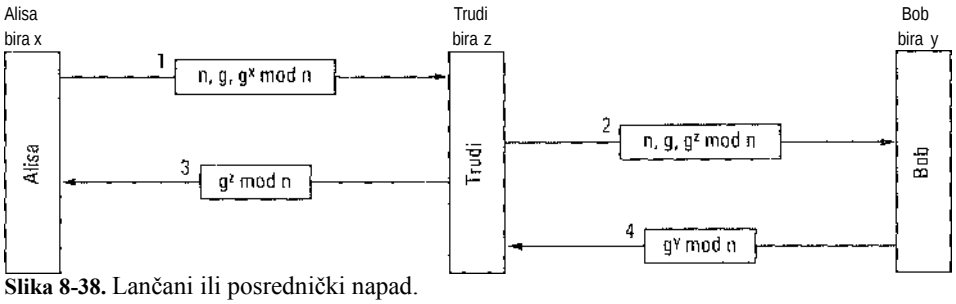


Slika 8-37. Difi-Helmanova razmena ključa.

Trudi je, naravno, ulovila obe poruke. Ona zna *n* i *g* iz poruke 1. Kada bi mogla da izračuna *x* i *y,* znala bi i tajni ključ. Problem je u tome što kada ima g^x mod *n,* ona ne može da izračuna *x.* Ne postoji algoritam za izračunavanje celobrojnog eksponenta *x* broja *g* iz ostatka deljenja g^x vrlo velikim prostim brojem *n.*

Da bismo vam primer približili, izabraćemo (sasvim nerealistične) vrednosti *n* = 47 i *g* = 3. Alisa bira v = 8, a Bob bira *y* = 10. Te dve vrednosti se čuvaju u tajnosti. Alisina poruka Bobu sadrži (47, 3,28) jer 3^8 mod 47 daje 28. Bobova poruka Alisi sadrži (17). Alisa izračunava 17^8 mod 47 i dobija 4. Bob izračunava 28^{10} mod 47 i dobija 4. Alisa i Bob su nezavisno jedno od drugog utvrdili da je tajni ključ 4. Trudi treba da reši jed- načinu 3^X mod 47 = 28, što se može postići sistematskim isprobavanjem malih brojeva upotrebljenih u ovom primeru, ali ne i kada su brojevi dugački nekoliko stotina bitova. Svi poznati algoritmi za rešavanje ovakve jednačine izvršavaju se veoma dugo, čak i na superračunarima.

Uprkos spoljnoj eleganciji Difi-Helmanovog algoritma, postoji i problem: kada Bob dobije triplet (47, 3, 28), kako zna da je stigao od Alise, a ne od Trudi? Nema načina da to sazna. Nažalost, tu činjenicu Trudi može da iskoristi da bi prevarila i Ali- su i Boba (slika 8-38). Dok Alisa i Bob biraju svoje v i *y,* Trudi bira sopstveni slučajan broj z. Alisa šalje poruku 1 namenjenu Bobu. Trudi presreće tu poruku i Bobu šalje pomku 2 u koju stavlja ispravne brojeve g i *n* (koji su ionako javni), ali umesto *x,* stavlja svoj broj z. Takođe odgovara Alisi porukom 3. Bob kasnije Alisi šalje poruku 4 koju Tmdi takođe presreće i zadržava.



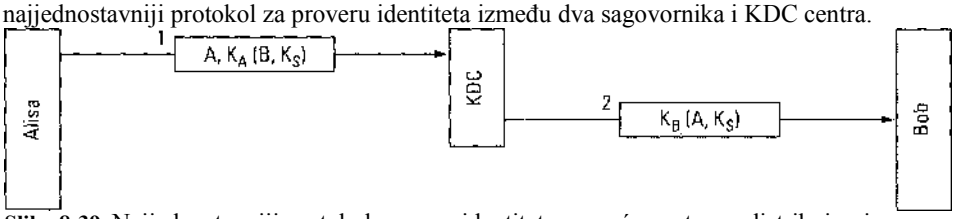
Slika 8-38. Lančani ili posrednički napad.

Sada svi izračunavaju ostatke celobrojnog deljenja. Alisa izračunava tajni ključ kao g^{xz} mod *n,* što radi i Trudi za ponike koje šalje Alisi. Bob izračunava g^{yz} mod *n,* što radi i Tmdi za poruke upućene njemu. Alisa misli da razgovara s Bobom, pa uspostavlja ključ sesije (s Tmdi). Isto tako i Bob. Svaka pomka koju Alisa šalje tokom šifrovane sesije stiže do Tmdi, gde je ova skladišti, po potrebi menja i zatim, opet po potrebi, prosleđuje Bobu. Isto se događa i s porakama iz dragog pravca. Tmdi vidi sve i može sve pomke da menja, dok Bob i Alisa zamišljaju da su povezani bezbednim kanalom. Ovaj napad je poznat kao **lančani napad** (engl. *bucket brigade attack)* jer podseća na lanac vatrogasaca koji jedan drugom dodaju kofe s vodom. Poznat je i kao **posrednički napad** (engl. *man-in-the-middle attack).*

8.7.3 Provera identiteta pomoću centra za distribuiranje ključeva

Kao što i sami vidite, uspostavljanje tajnog ključa između potpuno nepoznatih osoba zamalo je uspelo. S dmge strane, možda od čitave stvari odmah treba odustati (autonapad: „to grožđe je ionako kiselo"). Da biste na opisani način razgovarali sa *n* osoba, treba vam *n* ključeva. Za osobe koje neprestano razgovaraju to postaje stvarno opterećenje, naročito ako svaki ključ treba da spreme na plastičnu karticu.

Rešenje je da se uvede poverljiv centar za distribuiranje ključeva (KDC). Prema ovom modelu, svaki korisnik ima jedan ključ koji deli sa KDC centrom. Provera identiteta i uspostavljanje ključa sesije sada idu preko KDC centra. Na slici 8-39 prikazan je najjednostavniji protokol za proveru identiteta između dva sagovornika i KDC centra.



Slika 8-39. Najjednostavniji protokol provere identiteta pomoću centra za distribuiranje ključeva.

Osnovna zamisao protokola je sledeća: Alisa bira ključ sesije K_s i saopštava KDC centru da želi da razgovara s Bobom koristeći taj ključ. Tu poruku ona šifruje tajnim ključem K_A koji deli samo sa KDC centrom. KDC dešifruje poruku i iz nje izvlači Bobov identitet i ključ sesije. On zatim sastavlja novu poruku sa Alisinim identitetom i ključem sesije i šalje je Bobu, šifrovanu ključem K_B, tajnim ključem koji Bob deli samo s KDC centrom. Kada Bob dešifruje pomku, saznaje da Alisa želi da razgovara s njim, kao i ključ koji ona za taj

razgovor želi da koristi.

Identitet se na ovaj način proverava automatski. KDC zna daje poruka 1 morala stići od Alisejer je niko drugi nije mogao šifrovati Alisinim tajnim ključem. Isto tako, Bob zna da je poruka 2 morala stići od KDC centra kome veruje, pošto niko drugi ne zna njegov tajni ključ.

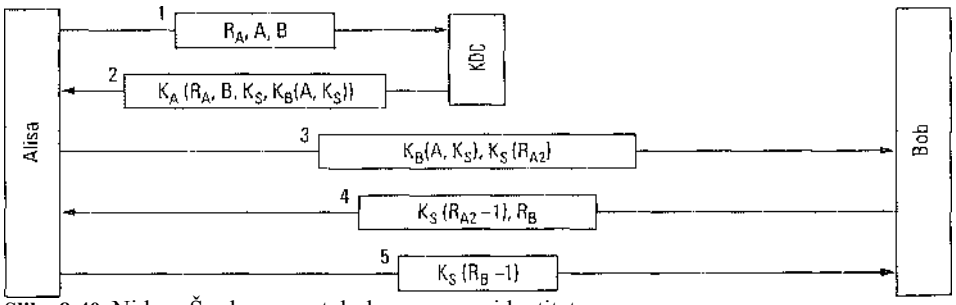
U ovom protokolu, nažalost, postoji veliki propust. Trudi je potreban novac, tako da ona smišlja neku malu legitimnu uslugu koju bi mogla da učini Alisi, pravi privlačnu ponudu i dobija posao. Pošto obavi taj posao, Trudi ljubazno traži od Alise da joj za njega plati preko banke. Zbog toga, Alisa uspostavlja ključ sesije sa svojim bankarom Bobom. Zatim Bobu šalje nalog za prenos novca na Trudin račun.

U međuvremenu, Trudi se vraća svom starom poslu špijuniranja saobraćaja na mreži. Ona kopira obe poruke sa slike 8-39, kao i zahtev za prenos novca koji sledi. Kasnije ih ona ponovo, neizmenjene šalje Bobu. Bob ih dobija i misli: Mora da je Alisa ponovo zaposlila Trudi. Ona očito zna svoj posao. Bob zatim prenosi isti iznos sa Alisinog na Trudin račun. Nakon pedesetak sličnih poruka, Bob žurno istrčava iz kancelarije i traži Trudi da bi joj ponudio veliku pozajmicu kako bi mogla da proširi svoj očigledno uspešan posao. Napad ove vrste zove se **napad ponovljenim slanjem poruka** (engl. *replay attack).*

Protiv takvog napada možete se boriti različitim sredstvima. Prvo je da se u svaku poruku uključi vremenska oznaka. Posle toga, svaka zastarela poruka se odbacuje. Ovakav pristup nije siguran jer satovi na mreži nikada ne rade potpuno sinhrono, pa se važnost vremenske oznake mora smestiti u neki vremenski interval. Taj interval može da iskoristi Trudi da bi unutar njega ponovo slala poruke.

Drugo rešenje je da se u svaku poruku uključi jednokratni uzorak. Svaka strana zatim treba da pamti sve prethodne uzorke i da odbacuje svaku pomku sa uzorkom koji se već pojavio. Ali uzorke treba pamtiti večito jer Trudi može ponovo da pošalje pomku staru 5 godina. Osim toga, ako neki računar doživi havariju i ostane bez liste uzoraka, ponovo postaje podložan napadu ponovljenim slanjem poruka. Vremenske oznake i jednokratni uzorci mogu se međusobno kombinovati da bi se ograničio period obaveznog čuvanja uzoraka, ali to očito prilično komplikuje protokol.

Složeniji pristup je primena višesmernog protokola međusobne provere identiteta testiranjem. Jedan dobro poznat primer takvog protokola je **Nidem-Šrederov protokol za proveru identiteta** (engl. *Needham-Schroeder authentication*) (Needham i Schroeder, 1978), čija je jedna varijanta prikazana na slici 8-40.



Slika 8-40. Nidem-Šrederov protokol za proveru identiteta.

Protokol počinje tako što Alisa saopštava KDC centru da želi da razgovara s Bobom. Ta poruka sadrži veliki slučajan broj R_A kao jednokratni uzorak. KDC joj odgovara porukom 2

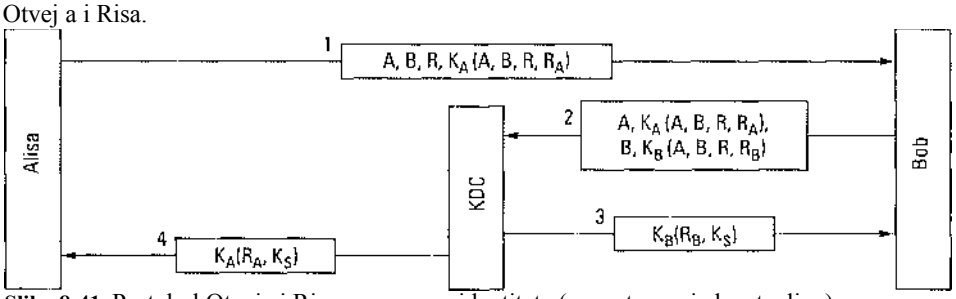
sa Alisinim slučajnim brojem, ključem sesije i tzv. kuponom (engl. *ticket)* koji ona može da pošalje Bobu. Slučajni broj R_A je tu da uveri Alisu u svežinu poruke 2. Tu je i Bobov identitet za slučaj da Trudi pokuša da *B* u poruci 1 zameni svojim identitetom, pa KDC šifruje kupon na kraju poruke 2 ključem K_T, a ne ključem K_B. Kupon šifrovan ključem K_B uključenje u šifro vanu poruku da ga Trudi ne bi za- menila nečim drugim na putu ka Alisi.

Alisa sada šalje kupon Bobu, zajedno s novim slučajnim brojem R_{A2}, šifrovanim ključem sesije K_s. Bob u poruci 4 šalje $K_S(R_{A2} - 1)$ da bi Alisi dokazao da razgovara s pravim Bobom. Odgovaranje sa $K_S(R_{A2})$ ne bi radilo, jer je tu vrednost Trudi mogla jednostavno ukrasti iz poruke 3.

Pošto primi poruku 4, Alisa je uverena da razgovara s Bobom i da do sada nije bilo pokušaja ponovnog slanja poruka. U krajnjoj liniji, ona je generisala R_{A2} tek pre nekoliko milisekundi. Cilj poruke 5 je da ubedi Boba da zaista razgovara sa Alisom i da ni ovde nije bilo ponovljenog slanja poruka. Kada svaka strana proveri identitet one druge testiranjem, potpuno se otklanja mogućnost napada ponovljenim slanjem poruka.

Iako opisani protokol izgleda prilično čvrsto, ipak ima jednu malu slabost. Ako Trudi ikako uspe da ulovi ključ neke stare sesije u obliku osnovnog teksta, ona može započeti novu sesiju s Bobom šaljući mu ponovo poruku 3 sa ukradenim ključem, i tako ga ubediti daje Alisa (Denning i Sacco, 1981). Ovoga puta ona može da isprazni Alisin račun u banci, a da za nju ne uradi baš nikakav legitiman posao.

Nidem i Šreder su kasnije objavili protokol koji ispravlja ovaj problem (Needham i Schroeder, 1987). U istom broju tog časopisa su i Otvej i Ris (Otway i Rees, 1987) objavili protokol koji problem rešava kraćim putem. Slika 8-41 prikazuje neznatno izmenjen protokol Otvej a i Risa.



Slika 8-41. Protokol Otveja i Risa za proveru identiteta (neznatno pojednostavljen).

Prema ovom protokolu, Alisa počinje tako što generiše par slučajnih brojeva: *R* koji će se koristiti kao zajednički identifikator i R_A kojim će Alisa testirati Boba. Kada Bob dobije ovu poruku, on sastavlja novu poruku od šifrovanog dela Alisine poruke i analognog dela svoje poruke. Dva dela, šifrovana ključevima K_A i K_B, identifikuju Alisu i Boba, i sadrže zajednički identifikator i poziv.

KDC proverava da li se u oba dela nalazi isti broj *R,* To može i da ne bude tako ako je Trudi menjala R u poruci 1 ili zamenila deo poruke 2. Ako je u oba dela broj *R* isti, KDC centar je uveren daje Bobov zahtev autentičan. KDC tad generiše ključ sesije i šifruje ga dva puta - po jednom za Alisu i Boba. Svaka poruka sadrži primaočev slučajan broj kao dokaz da ju je generisao KDC centar, a ne Tmdi. U tom trenutku, i Alisa i Bob imaju isti ključ sesije, pa mogu da započnu komuniciranje. Kada prvi put razmene poruke s podacima, svaka strana će utvrditi da ona draga ima identičan K_s, pa je time provera identiteta završena.

8.7.4 Provera identiteta pomoću Kerberosa

Kerberos je sistem za proveru identiteta koji se koristi u mnogim realnim sistemima (uključujući Windows 2000), a zasniva se na varijanti Nidem-Šrederovog protokola. Sistem je nazvan po troglavom psu iz grčke mitologije koji je čuvao ulaz u podzemni svet (da nepoželjni ne bi iz njega utekli). Kerberos je projektovan na Masa- čusetskom tehničkom institutu da bi korisnici radnih stanica mogli bezbedno da pristupaju mrežnim resursima. Od Nidem-Šrederovog protokola uglavnom se razlikuje po pretpostavci da su svi satovi potpuno sinhronizovani. Protokol je prošao kroz mnoge verzije. Verzija 4 se široko koristi, pa ćemo govoriti o njoj, a zatim posvetiti nekoliko reči i njenom nasledniku, verziji 5. Detaljnija objašnjenja potražite kod Steinera i saradnika (1988).

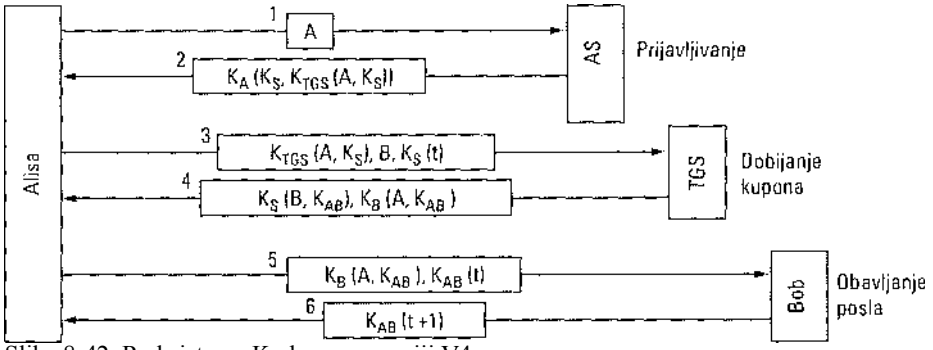
Kerberos, osim Alise (klijentske radne stanice), obuhvata još tri servera:

1. Server za proveru identiteta (engl. *Authentication Server, AS):* proverava korisnike tokom prijavljivanja
2. Server za dodelu kupona (engl. *Ticket-Granting Server, TGS):* izdaje kupone „s dokazom identiteta“
3. Server Bob: radi ono što Alisa od njega zahteva (ispunjava želje)

Server AS liči na KDC centar po tome što deli tajnu lozinku sa svakim korisnikom. TGS server izdaje kupone koji treba da uvere prave servere da je podnosilac TGS kupona stvarno ono što tvrdi da jeste.

Kada želi da započne sesiju, Alisa seda za proizvoljnu javnu radnu stanicu i unosi svoje ime. Radna stanica šalje njeno ime AS serveru u obliku običnog teksta, kao što je prikazano na slici 8-42. Kao odgovor dobija ključ sesije i kupon $K_{TQS}(A, K_s)$, namenjen TGS serveru. Dve stavke su zajedno spakovane i šifrovane Alisinim tajnim ključem, tako da samo ona može da ih dešifruje. Tek kada stigne poruka 2, radna stanica traži od Alise da unese lozinku. Lozinka se tada koristi za generisanje ključa K_A kojim se dešifruje poruka 2 i dobija ključ sesije i TGS kupon. Posle samo nekoliko milisekundi, radna stanica briše iz memorije Alisinu lozinku. Ako Trudi pokuša da se prijavi kao Alisa, njena lozinka će biti pogrešna, a radna stanica će to otkriti jer će ne- tačan biti standardni deo poruke 2.

Pošto se prijavi, Alisa može saopštiti radnoj stanici da želi da stupi u vezu sa serve- rom datoteka Bob. Radna stanica tada TGS serveru šalje poruku 3, zahtevajući kupon za vezu s Bobom. Najvažniji element ovog zahteva je $K_{TGS}(A, K_s)$. On je šifrovan tajnim ključem TGS servera i koristi se kao dokaz daje pošiljalac stvarno Alisa. TGS server odgovara ključem sesije K_{AB} koji se generiše za vezu između Alise i Boba. Na dve strane se šalju dve njegove verzije. Prva se šifruje samo ključem K_s, tako da Alisa može da je dešifruje. Draga se šifruje Bobovim ključem K_B - nju čita Bob.



Slika 8-42. Rad sistema Kerberos u verziji V4.

Trudi može da kopira poruku 3 i da pokuša da je ponovo upotrebi, ali će je u tome sprečiti šifrovana vremenska oznaka *t* koja se šalje zajedno s njom. Trudi ne može vremensku oznaku da zameni svežijom jer ne zna K_s, ključ sesije koji Alisa koristi za komuniciranje sa TGS serverom. Čak i kada bi Trudi uspela da brzo ponovo pošalje poruku 3, đobice samo dragu kopiju poruke 4 koju nije mogla da dešifruje ranije, pa neće moći ni sad.

Sada Alisa može Bobu da pošalje K_{AB} da bi s njim uspostavila sesiju. Ove poruke takođe uključuju vremensku oznaku. Odgovor je potvrda da Alisa stvarno razgovara s Bobom, a ne s Trudi.

Posle ove serije razmenjenih poruka, Alisa može da razgovara s Bobom pod zaštitom ključa K_{AB}. Ako kasnije odluči da treba da razgovara s Kerol - drugim serverom, ona će TGS serveru samo ponoviti poruku 3, zamenjujuci identitet *B* identitetom *K*. TGS server će odmah odgovoriti kuponom šifrovanim ključem K_K koji Alisa može da pošalje Kerol i koji će Kerol prihvatiti kao dokaz da je stvarno stigao od Alise.

Smisao opisanog postupka je da Alisa sada može da bezbedno pristupa serverima na čitavoj mreži, a da njena lozinka nikada ne izlazi na mrežu. U stvari, ona se samo nekoliko milisekundi pojavljuje na radnoj stanici za kojom se Alisa trenutno nalazi. Međutim, imajte na umu da svaki server ima sopstvena ovlašćenja. Kada Alisa dostavi svoj kupon Bobu, to je za Boba samo dokaz da gaje poslala Alisa. Draga je stvar šta Bob dozvoljava Alisi da radi.

Pošto projektanti Kerberosa ne očekuju da ceo svet veruje jednom serveru za pro- veru identiteta, stvorili su uslove za postojanje više područja (engl. *realms),* svakog sa svojim sopstvenim AS i TGS serverima. Da bi dobila kupon za server u udaljenom području, Alisa treba da od sopstvenog TGS servera zatraži kupon koji će prihvatiti TGS server u udaljenom području. Ako je udaljeni TGS server registrovan kod lokalnog TGS servera (kao što je uobičajeno kod lokalnih servera), lokalni TGS server će Alisi dati kupon koji važi za udaljeni TGS. Ona zatim može da obavlja različite poslove, na primer, da dobija kupone za servere u tom području. Međutim, ako dve strane u dva područja žele da međusobno posluju, svaka od njih mora verovati TGS serveru drage strane.

Peta verzija Kerberosa je maštovitija i, takođe, opširnija od njegove četvrte verzije. I u njoj se za opisivanje tipova podataka koristi ASN. 1 (opis apstraktne sintakse 1) prema modelu OSI, ali su protokoli pretrpeli manje izmene. Životni vek kupona je produžen, važnost im se sada može obnavljati, a i mogu se izdavati „na počele“. Osim toga, Kerberos je u ovoj verziji, barem teorijski, postao nezavistan od sistema DES i u njega je uvedena podrška za više područja, tako što je generisanje kupona preneto na više TGS servera.

8.7.5 Provera identiteta pomoću šifrovanja javnim ključem

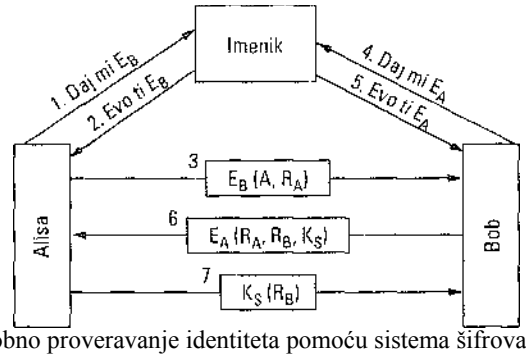
Uzajamna provera identiteta može se izvesti i pomoću šifrovanja javnim ključem. Da bi započela postupak, Alisa mora imati Bobov javni ključ. Ako postoji PKI sa serverom kataloga koji izdaje sertifikate za javne ključeve, Alisa od njega može tražiti Bobov ključ porukom 1 (slika 8-43). Poruka 2 je odgovor koji sadrži sertifilcat X.509 s Bobovim javnim ključem. Kada Alisa utvrdi daje potpis ispravan, ona Bobu šalje poruku s njenim identitetom i jednokratnim uzorkom.

Kada Bob primi ovu poruku, on ne zna da li je stigla od Alise ili od Trudi, ali prihvata igru i od servera imenika traži Alisin javni ključ (poruka 4) koji ubrzo dobija (poruka 5). On tada šalje Alisi poruku sa Alisinim R_A, svojim jednokratnim uzorkom R_B i predloženim ključem sesije K_s (poruka 6).

Kada Alisa dobije poruku 6, dešifruje je svojim privatnim ključem. U njoj nalazi R_a, zbog čega joj je u duši toplije: poruka mora daje stigla od Boba, pošto Trudi nema načina da otkrije R_A. Osim toga, mora biti i daje sveža (a ne ponovljena), postoje
Bobu poslala R_A neposredno pre toga. Alisa potvrđuje sesiju porukom 7. Kada Bob ugleda R_B šifrovano ključem sesije koji je upravo generisao, on zna daje Alisa dobila poruku 6 i da je proverila R_A.

Kako Trudi može da prevari ovaj protokol? Ona može da isfabrikuje poruku 3 i da navuče Boba da proverava Alisu, ali će Alisa videti R_A koji nije poslala i neće ništa odgovarati. Trudi

ne može da falsifikuje poruku 7 upućenu Bobu jer ne zna ni R_B ni K_s, a ne može da ih otkrije bez Alisinog privatnog ključa. Ovoga puta baš nema sreće.



Slika 8-43. Međusobno proveravanje identiteta pomoću sistema šifrovanja javnim ključem.

8.8 BEZBEDNOST E-POŠTE

Kada se između dve udaljene lokacije pošalje poruka e-poštom, ona će na svom putu proći verovatno kroz desetine računara. Svaki od njih može da je pribeleži i sačuva za buduću upotrebu. Privatnost u stvari ne postoji, ma šta ljudi mislili. Pa ipak, mnogi bi želeli da poruku može da pročita samo potencijalni primalac i nilco drugi: ni šef, ni neko iz vlade. Takva želja je podstakla više autora i autorskih grupa da kritpo- grafske principe o kojima smo govorili primene na elektronske poruke i tako ostvare bezbednu e-poštu. U narednim odeljcima ćemo obraditi široko korišćen poštanski sistem PGP i samo pomenuti druga dva: PEM i S/MIME. Dopunske informacije o bez- bednoj e-pošti potražite kod Kaufmana i saradnika (2002), i kod Scheinera (1995).

8.8.1 PGP - prilično dobra privatnost

Naš prvi primer, Prilično dobra privatnost (engl. *Pretty Good Privacy, PGP),* čedo je uglavnom jednog autora, Fila Cimermana (Zimmermann, 1995a, 1995b). Cimerman je pobornik privatnosti, čiji moto glasi: Ako se privatnost izuzme iz zakona, imaće je samo oni izvan zakona. Sistem PGP, objavljen 1991, potpun je paket za e-poštu koji obezbeđuje privatnost, proveru identiteta, digitalno potpisivanje i kompri- movanje - sve u obliku koji se lako koristi. Paket se, zajedno sa izvornim kodom, besplatno distribuira preko Interneta. Zbog svog kvaliteta, cene (besplatan) i lake realizacije u UNIX-u, Linuxu, Windowsu i Mac OS-u, danas je u širokoj upotrebi.

PGP šifruje podatke blok-šifrom zvanom **međunarodni algoritam za šifrovanje podataka** (engl. *International Data Encryption Algorithm, IDEA),* koji koristi 128-bitne ključeve. Algoritam je razvijen u Švajcarskoj u trenutku kada je DES počeo da bije loš glas, a AES još nije postojao. Algoritam IDEA koncepcijski podseća i na DES i na AES: on meša bitove u više rundi, ali mu je funkcija mešanja drugačija. Za rad s ključevima koristi se RSA, a za kontrolu integriteta podataka MD5 - teme koje smo već obradili.

PGP je već prvog dana izazvao gužvu (Levy, 1993). Pošto Cimerman nije sprečavao druge da izlože PGP na Internetu, odakle su ga svi mogli preuzeti, Američka vlada ga je

optužila da je prekršio Zakon o izvozu ratnog materijala. Cimermana su ispitivali 5 godina, a zatim prestali, verovatno iz dva razloga. Prvo, Cimerman nije sam postavio PGP na Internet, pa je njegov advokat tvrdio da on *nikada nije ništa izvezao* (a tada gubi važnost i pitanje da li je pravljenje Web lokacije izvoz). Drugo, Američka vlada je shvatila da bi za đobijanje procesa trebalo ubediti porotu da Web lokacija s programom za privatnost koji svako može preuzeti potpada pod zakon o prometu oružja koji zabranjuje izvoz ratnog materijala, kao što su tenkovi, podmornice, vojni avioni i nuklearno oružje. Višegodišnji negativni publicitet koji je imao ovaj slučaj, takođe nije pomogao.

Izvozni zakoni su, među nama, pomalo čudni, da ne upotrebimo neku težu reč. Vlada je zauzela stav da objavljivanje koda na Web lokaciji predstavlja nelegalan izvoz i zbog toga proganjala Cimermana 5 godina. S druge strane, kada neko objavi potpun izvorni PGP kod na jeziku C kao knjigu (krupnim fontom, s kontrolnim zbirom na svakoj strani, da bi se olakšalo skeniranje), a zatim knjigu izveze, tada vlada ni ne trepne, jer knjige nisu municija. Mač je jači od pera, barem za Ujka Sema.

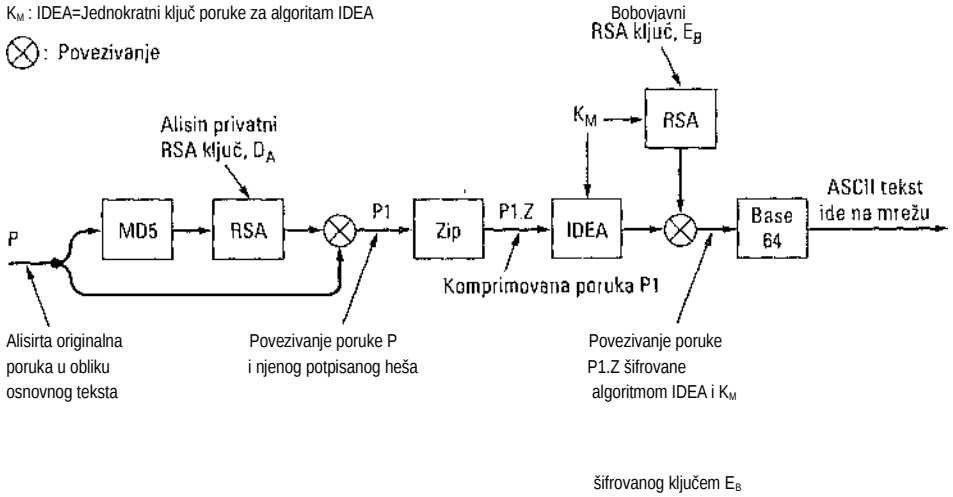
Sistem PGP zadire i u propise koji se tiču zaštite intelektualne svojine. Vlasnik patenta RSA - kompanija RSA Security Inc., najavila je da se korišćenjem algoritma RSA unutar sistema PGP krše njena patentna prava, ali se sve smirilo nakon objavljivanja verzije 2.6. Osim RSA, sistem PGP obuhvata i dragi patentiran algoritam za šifrovanje (IDEA), čije korišćenje je na početku takođe izazvalo pravne probleme.

Pošto je PGP sistem otvorenog koda, mnogi pojedinci i grupe su ga menjali, tako da postoji niz njegovih verzija. Neke od njih su pravljene tako da zaobiđu propise o prometu ratnog materijala, druge - da izbegnu korišćenje patentiranih algoritama, a kod trećih je pokušano da se sistem pretvori u komercijalan proizvod zatvorenog koda. Iako su propisi o prometu ratnog materijala postali nešto blaži (inače se proizvod sa algoritmom AES ne bi mogao izvesti iz SAD), a patent za algoritam RSA je istekao septembra 2000, zbog postojanja ovih problema u prošlosti danas pod različitim imenima kruži više međusobno nekompatibilnih verzija sistema PGP. Razmatranje koje sledi tiče se klasičnog sistema PGP - njegove najstarije i najjednostavnije verzije. Druga popularna verzija, Open PGP, opisana je u RFC dokumentu 2440. Još jedna verzija je GNU Privacy Guard.

Umesto da se izmišljaju novi algoritmi za šifrovanje, u sistemu PGP namemo se koriste postojeći algoritmi. Oni su izabrani nakon što su ih stručnjaci detaljno pregledali i nakon što je utvrđeno da na njihovo projektovanje nije uticala nijedna vladina agencija s namerom da ih oslabi. Oni koji baš nemaju poverenja u vladu, ocenjuju taj potez veoma pozitivno.

Sistem PGP podržava komprimovanje teksta, tajnost i digitalno potpisivanje, a obezbeđuje i dosta alatki za opsežan rad s ključevima, ali, iznenađujuće, ne i za rad sa e-poštom. On više liči na pretprocesor koji ulazni osnovni tekst pretvara u potpisani šifrovan tekst kodiran sistemom base64. Rezultat se tada može poslati kao poruka e-pošte. U nekim realizacijama sistema PGP, u poslednjem koraku se poziva korisnički agent koji stvarno šalje poruku.

Da bismo razumeli kako radi PGP, razmotrimo primer prikazan na slici 8-44. Tu Alisa želi da pošalje Bobu poruku s potpisanim osnovnim tekstom *P* na bezbedan način. I Alisa i Bob imaju svoje privatne (D_x) i javne (E_x) RSA ključeve. Pretpostavimo da svako od njih zna javni ključ onog drugog; ubrzo ćemo nešto reći o radu sa PGP ključevima.



Slika 8-44. Rad sistema PGP pri slanju poruke.

Alisa počinje tako što pokreće program PGP na svom računam. PGP prvo hešira njenu poruku *P* koristeći algoritam MD5, a zatim šifruje rezultujući heš njenim privatnim RSA ključem D_A. Kada Bob na kraju dobije poruku, on može da dešifruje heš Alisinim javnim ključem i da utvrdi njegovu ispravnost. Čak i ako neko drugi (npr. Trudi) uspe da ulovi heš u ovoj fazi i da ga dešifruje Alisinim javnim ključem, algoritam MD5 garantuje da neće moći da napravi dragu poruku sa istim MD5 hešom.

Šifrovani heš i originalna poruka povezuju se sada u jedinstvenu poruku *P1* koja se komprimuje programom ZIP uz korišćenje algoritma Ziva i Lempela (1977). Rezultat ovog koraka označićemo sa *Pl.Z.*

Posle toga, PGP zahteva od Alise da napiše bilo šta. Na osnovu sadržaja i brzine unosa generiše se 128-bitni IDEA ključ poruke K_M (koji se u PGP literaturi zove ključ sesije, ali je to stvarno previd jer ne postoji sesija). K_M se sada koristi za šifrovanje *Pl.Z* algoritmom IDEA u režimu rada s povratnom spregom. Osim toga, K_M se šifruje Bobovim javnim ključem E_B. Ove dve komponente se zatim povezuju i kodiraju sistemom base64, kao što smo opisali u odeljku o MIME kodiranju u 7. poglavlju. Re- zultujuća poruka sadrži samo slova, cifre i simbole +, / i =, što znači da se može smestiti u telo poruke prema RFC dokumentu 822 i očekivati da stigne neizmenjena.

Kada Bob dobije poruku, on dekodira poruku kodiranu sistemom base64 i dešifruje IDEA ključ koristeći svoj privatni RSA ključ. Pomoću IDEA ključa on dešifruje poruku i dobij a *Pl.Z.* Posle dekomprimovanja poruke, Bob odvaja osnovni tekst od šifrovanog heša i dešifruje heš Alisinim javnim ključem. Ako se heš osnovnog teksta slaže s njegovim sopstvenim izračunavanjem pomoću algoritma MD5, on zna da je *P* ispravna poruka i da dolazi od Alise.

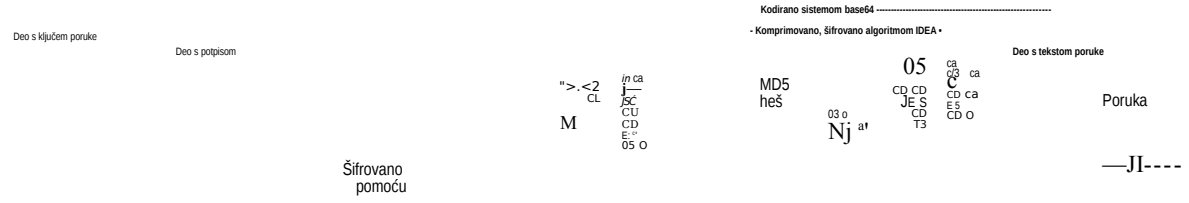
Treba naglasiti da se algoritam RSA ovde koristi samo na dva mesta: za šifrovanje 128-bitnog MD5 heša i za šifrovanje 128-bitnog IDEA ključa. Iako algoritam RSA radi sporo, ovde ne treba da šifruje celu poruku, već samo 256 bitova. Štaviše, svih 2.56 bitova osnovnog teksta izuzetno su dobro izmešani, pa bi se Trudi propisno namučila da utvrdi da li je pogođeni ldjuč uopšte tačan. Opsežno šifrovanje vrši se algoritmom IDEA koji je višestruko brži od algoritma RSA. Tako, sistem PGP nudi bezbednost, komprimovanje i digitalno potpisivanje, i to na mnogo efikasniji način nego šema prikazana na slici 8-19.

PGP podržava četiri dužine RSA ključa. Korisnik bira najpogodniji. Dužine su:

1. Obična (384 bita): danas se može lako provaliti.
2. Komercijalna (512 bitova): mogu je razbiti organizacije s troslovnim imenima.
3. Vojna (1024 bita): ne može je razbiti niko na zemlji.
4. Vanzemaljska (2048 bitova): ne može je razbiti niko u svemiru.

Pošto se algoritam RSA koristi samo za kratka izračunavanja, svi će uvek izabrati vanzemaljski ključ.

Format klasične PGP poruke prikazanje na slici 8-45. Koriste se i različiti drugi formati. Poruka je iz tri dela: IDEA ključa, potpisa i samog teksta poruke. Deo s ključem ne sadrži samo ključ, već i identifikator ključa, pošto je korisnicima dozvoljeno da imaju više javnih ključeva.



Slika 8-45. PGP poruka.

Deo s potpisom sadrži zaglavlje koje nas ovde ne zanima. Iza zaglavlja dolazi vremenska oznaka, identifikator pošiljaočevog javnog ključa kojim se može dešifrovati heš potpisa, neke informacije o tipu koje identifikuju korišćene algoritme (da bi se omogućilo i korišćenje MD6 i RSA2 - kad budu smišljeni), i sam šifrovani heš.

Deo s porukom takođe sadrži zaglavlje, podrazumevano ime datoteke za slučaj da primalac želi da snimi datoteku na disk, vremensku oznaku pisanja poruke i, na kraju, samu poruku.

S ključevima se radi na sledeći način. Svaki korisnik lokalno održava dve strukture podataka: skup privatnih i skup javnih ključeva. Skup privatnih ključeva (engl. private *key ring*) sadrži jedan ili više parova ličnih privatnih i javnih ključeva. Podrška za više parova ključeva po korisniku postoji da bi korisnici mogli da ih periodično menjaju u slučajevima kada sumnjaju da su zloupotrebljeni, a da time ne remete poruke koje su trenutno u pripremi ili u tranzitu. Svakom paru je pridružen iđentifikator tako da pošiljalac poruke može da naznači primaocu koji je javni ključ iskorišcen za njeno šifrovanje. Identifikatori poruke su najmanje značajna 64 bita javnog ključa. Korisnici treba da vode računa o tome da se identifikatori njihovih javnih ključeva ne sukobljavaju. Privatni ključevi-na disku šifruju se specijalnom lozinkom (proizvoljne dužine) da bi se zaštitili od njuškala.

Skup javnih ključeva (engl. *public key ring)* sadrži javne ključeve korisnikovih korespondenata. Oni služe za šifrovanje ključeva poruka koji se pridružuju svakoj poruci. Svaka odrednica u skupu javnih ključeva ne sadrži samo javni ključ, *već* i 64-bit- ni iđentifikator i stepen poverenja koje korisnik ima u taj ključ.

Pitanje poverenja u ključ svodi se na sledeče. Pretpostavimo da se javni ključevi drže na elektronskim oglasnim tablama. Trudi će moći da čita Bobovu tajnu poštu ako napadne oglasnu tablu i Bobov javni ključ zameni svojim. Kada Alisa kasnije preuzme ključ koji važi za Bobov, Trudi protiv Boba može da pokrene posrednički napad.

Da bi takve napade sprečila ili barem umanjila njihove posledice, Alisa treba da zna koliko veruje „Bobovom ključu“ u skupu svojih javnih ključeva. Ako joj je Bob lično predao disketu sa svojim javnim ključem, ona u njega može da ima potpuno po- verenje. Takav decentralizovan rad s javnim ključevima, kojim upravlja korisnik, izdvaja sistem PGP od centralizovanih PKI šema.

Pa ipak, korisnici ponekada dolaze do javnih ključeva pretražujući pouzdane ser- vere ključeva. Zbog toga, kada je prihvaćen standard X.509, PGP je počeo da podržava sertifikate u skladu s njim jednako kao i klasični mehanizam skupa PGP ključeva. Sve aktuelne verzije sistema PGP podržavaju standard X.509.

8.8.2 PEM - pošta s poboljšanom privatnošću

Za razliku od sistema PGP, koji je u početku bio inicijativa jedne osobe, naš drugi primer, Pošta s poboljšanom privatnošću (engl. *Privacy Enhanced Mail, PEM),* koja je razvijena kasnih osamdesetih godina, predstavlja zvaničan standard za Internet i opisana je u četiri RFC dokumenta (1421-1424). Najgrublje rečeno, PEM pokriva isto područje kao i PGP: privatnost i proveru identiteta za sisteme e-pošte zasnovane na RFC dokumentu 822. Ipak, postoje i neke razlike u pristupu i tehnologiji.

Poruke koje se šalju u sistemu PEM prvo se pretvaraju u kanonički oblik tako da za sve važe iste konvencije u pogledu belina (znaka tabulatora, dopunskih razmaka itd.). Zatim se izračunava heš poruke algoritmom MD2 ili MD5. Potom se poruka s nado- vezanim hešom šifruje algoritmom DES. Imajući u vidu slabost 56-bitnog ključa, ovakav izbor izvesno budu sumnju. Šifrovana poruka se tada može kodirati sistemom base64 i poslati primaocu.

Kao u sistemu PGP, i ovde se svaka poruka šifruje jednokratnim ključem koji se uključuje u poruku. Ključ se može zaštititi algoritmom RSA ili trostrukim DES-om u režimu EDE.

Rad s ključevima je strukturiran više nego u sistemu PGP. Ključevi se potvrđuju sertifikatima prema standardu X.509 koje izdaju CA organizacije, uređene u strogu hijerarhiju koja počinje od jedinstvenog vrha. Prednost takve šeme je mogućnost povlačenja sertifikata na osnovu CRL lista koje povremeno objavljuje vrh hijerarhije.

Jedini problem sa sistemom PEM bio je to što ga nikada niko nije upotrebio i što je davno bačen u staro gvožđe. Naravno, problem je bio uglavnom političke prirode: ko bi seo u vrh hijerarhije i pod kojim uslovima? Kandidata je bilo na pretek, ali su mnogi zazirali od toga da bezbednost čitavog sistema povere jednoj kompaniji. Najozbiljniji kandidat, kompanija RSA Security, Inc., želela je da naplaćuje naknadu po svakom izdatom sertifikatu. Međutim, neki učesnici rasprave snažno su se protivili toj ideji. Konkretno, Američka vlada je odobrila besplatno korišcenje svih američkih patenata, a kompanije izvan SAD navikle su da algoritam RSA koriste besplatno (jer je RSA Security zaboravila da ga zaštiti u inostranstvu). Zbog toga niko nije bio oduševljen idejom da kompaniji RSA Security plaća nešto što je oduvek bilo besplatno. Na kraju niko nije izabran za vrhovnu CA organizaciju i sistem PEM je propao.

8.8.3 S/MIME

Sledeći poduhvat organizacije IETF na polju bezbednosti bio je **bezbedni sistem MIME** (engl. *Secure/MIME, S/MIME),* koji je opisan u RFC dokumentima 2632- 2643. Slično PEM-u, i ovaj sistem obezbeđuje proveru identiteta, kontrolu integriteta podataka, tajnost i nemogućnost poricanja. On je i prilično fleksibilan - podržava niz algoritama za šifrovanje. Sudeći po imenu, ne iznenađuje to što se S/MIME dobro slaže sa sistemom MIME, omogućavajući zaštitu svih vrsta poruka. Definisano je više novih MIME zaglavlja, na primer, zaglavlje za čuvanje digitalnih potpisa.

IETF je očigledno izvukao pouku iz iskustva s PEM-om. S/MIME nema strogu hijerarhiju sertificiranja koja počinje od jedinstvenog vrha, već korisnici mogu da biraju između više pouzdanih polazišta. Sve dok se za sertifikat može povratno naći pouzdano polazište kome korisnik poklanja poverenje, sertifikat se smatra važećim. Sistem S/MIME koristi standardne algoritme i protokole o kojima smo već govorili, pa ga nećemo dalje razmatrati. Koga zanimaju detalji, neka pogleda odgovarajuće RFC dokumente.

8.9 BEZBEDNOST WEBA

Upravo smo razmotrili dva važna područja u kojima je neophodna bezbednost: komunikacije i e-poštu. Možemo ih uporediti sa aperitivom i supom, a sada dolazi „glavno jelo“: bezbednost Weba. Web je područje koje danas obilazi skoro svaka Trudi, tražeći žrtve. Naredne ođeljke posvetićemo nekim problemima koji se tiču bezbednosti Weba.

Bezbednost Weba se grubo može podeliti na tri dela. Prvo, kako se objekti i resursi imenuju na bezbedan način? Drago, kako se može uspostaviti bezbedna veza s pro- verenim identitetom? Trec'e, šta se dešava kada Web lokacija pošalje klijentu izvršni kod? Pošto se upoznamo s nekim pretnjama bezbednom radu, vraticemo se na svako od ovih pitanja.

8.9.1 Ugrožavanje Weba

O problemima bezbednosti Weba možete često da čitate u dnevnoj štampi. Situacija nije nimalo ružičasta. Navedimo nekoliko primera onoga što se već dogodilo. Prvo, matične Web strane mnogih organizacija su napadnute i zamenjene stranama po izboru provalnika. (Popularna štampa provalnike u računarske sisteme naziva „hake- rima“, ali programeri tim izrazom označavaju svoje posebno nadarene kolege. Zato ćemo mi zadržati stari dobri izraz „provalnik“.) Među lokacije na koje je provaljeno spadaju Yahoo, Američka vojska, CIA, NASA i New York Times. U većini slučajeva, provalnici su na matičnu stranu samo stavljali neki duhovit tekst, tako da su lokacije mogle biti dovedene u red za nekoliko sati.

Osvrnimo se sada na neke ozbiljnije slučajeve. Brojne lokacije su dovedene do kraha napadima koji blokiraju usluge - provalnik plavi lokaciju saobraćajem, onemogućujući joj da odgovara na legitimne zahteve. Takvi napadi se često preduzimaju s velikog broja računara koje je provalnik već zauzeo (distribuirani DoS napadi). Napadi ove vrste već su tako česti da ne predstavljaju novinsku vest, ali svaki takav napad može žrtvu da košta hiljade dolara zbog neobavljenog posla.

Godine 1999, jedan švedski provalnik je upao na Microsoftovu Web lokaciju Hotmail i napravio njenu kopiju tako da je svako mogao da upiše ime nekog korisnika Hotmaila, a zatim da čita svu njegovu aktuelnu i arhiviranu poštu.

Drugi put, devetnaestogodišnji ruski provalnik Maksim upao je na Web lokaciju za elektronsku trgovinu i tamo ukrao brojeve 300.000 kreditnih kartica. Zatim je ucenio vlasnike lokacije da mu isplate 100.000 dolara, inače će sve brojeve objaviti na Internetu. Vlasnici nisu podlegli učeni i on je stvarno sve brojeve kreditnih kartica objavio na Internetu, zavivši u crno mnoge nedužne korisnike.

Opet jednom, 23-godišnji student iz Kalifornije poslao je e-poštom lažnu izjavu za štampu novinskoj agenciji tvrdeći da se korporacija Emulex sprema da objavi veliki kvartalni gubitak i da direktor daje trenutnu i neopozivu ostavku. Akcije kompanije su za samo nekoliko sati pale na 40% prvobitne vrednosti, zbog čega su njihovi vlasnici izgubili preko dve milijarde dolara. Za razliku od njih, naš mutivoda je zaradio četvrt miliona dolara prodavši akcije neposredno pre objavljivanja izjave. Premda opisani slučaj ne predstavlja upad na Web lokaciju, jasno je da bi objavljivanje takve vesti na matičnoj strani bilo koje velike korporacije izazvalo slične posledice.

Mogli bismo (nažalost) nastaviti ovako do kraja knjige. Ipak, treba da razmotrimo neke tehničke aspekte bezbednosti Weba. Više podataka o bezbednosnim problemima svake vrste nađi ćete kod Andersona (2001), Garflnlcela i Spafforda (2002) i Schnei- era (2000). Do niza specifičnih slučajeva možete doći i pretraživanjem Interneta.

8.9.2 Bezbedno imenovanje

Počnimo od nečeg sasvim jednostavnog: Alisa želi da poseti Bobovu Web lokaciju. Ona upisuje Bobovu URL adresu **u** svoj čitač i posle nekoliko sekundi, Web strana se pojavljuje. Ali, da lije to Bobova strana? Možda jeste, a možda i nije jer se Trudi verovatno vratila svom starom poslu. Ona, na primer, može da presretne sve Alisine odlazite pakete i da ih pregleda. Kada ulovi HTTP zahtev *GET,* upućen Bobovoj Web lokaciji, ona i sama može da ode na Bobovu Web lokaciju, da uzme Web stranu, izmeni je po svom ukusu i vrati je Alisi umesto prave strane. Alisa o tome ništa neće saznati. Gore je to što Trudi može da obori cene artikala u Bobovoj elektronskoj pro- davnici i tako ih učini veoma privlačnim; kada Alisa pošalje „Bobu“ broj svoje kreditne kartice da bi nešto kupila, Trudi će taj broj uloviti bez problema.

Ovakav posrednički napad je nezgodan jer Trudi mora da bude u položaju da presreće

Alisin odlazni saobraćaj i da falsifikuje njen dolazni saobraćaj. U praksi to znači da mora da se „zakači“ ili za Alisin ili za Bobov telefon jer je fizičko kačenje za optički kabl prilično težak poduhvat. Prisluškivanje tuđeg telefona je sigurno moguće uz određen trud, ali naša Trudi - iako pametna - ipak je lenja. Osim toga, Alisa se može prevariti na lakši način.

DNS Iažiranje

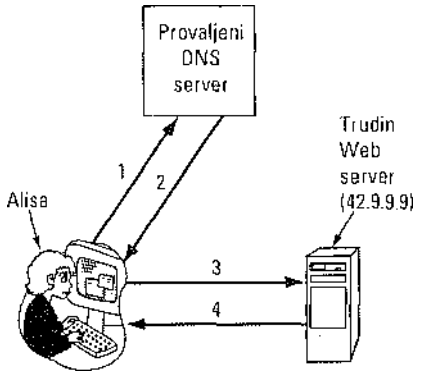
Pretpostavimo, na primer, da Trudi može da uđe u DNS sistem, možda samo u DNS keš Alisinog davaoca Internet usluga, i da zameni Bobovu IP adresu (recimo, 36.1.2.3) svojom (Trudinom) IP adresom (recimo, 42.9.9.9). To omogućava sledeći napad. Alisa normalno zahteva od DNS servera Bobovu IP adresu (1), dobija je (2), traži od Boba njegovu Web stranu (3), pa dobija i nju (4). Pošto Trudi u Bobovom DNS zapisu njegovu IP adresu zameni svojom, dolazimo u situaciju prikazanu slikom 8-46(b). Tu, kada Alisa zatraži Bobovu IP adresu, ona dobija Trudinu, tako da sav njen saobraćaj namenjen Bobu odlazi Trudi. Trudi može sada da organizuje posrednički napad a da se ne kači ni na čiji telefon. Dovoljno je da provali u DNS server i tamo izmeni samo jedan zapis, što je mnogo lakše.

Kako Trudi može da prevari DNS server? Ispada da je to prilično lako. Ukratko, Trudi može da natera DNS server kod Alisinog davaoca Internet usluga da pošalje zahtev za traženje Bobove adrese. Pošto DNS koristi protokol UDP, DNS server nažalost ne može da utvrdi od koga je dobio odgovor. Trudi to može da iskoristi fal- sifikujući očekivani odgovor i da tako ubaci drugu IP adresu u keš DNS servera. Zbog jednostavnosti ćemo pretpostaviti da Alisin davalac Internet usluga na početku nema odrednicu za Bobovu Web lokaciju *bob.com.* Ako, pak, ima odrednicu, Trudi može da sačeka da joj istekne važnost i da pokuša ponovo (ili da primeni druge trikove).

Trudi počinje tako što šalje zahtev za pretraživanje Alisinom davaocu Internet usluga, tražeći IP adresu lokacije *bob.com.* Pošto za ovo ime ne postoji odrednica, server za lceširanje će se obratiti serveru osnovnog domena *com.* Međutim, Trudi pre- tiče server *com* i šalje falsifikovan odgovor: *„bob.com* je 42.9.9.9“, gde je IP adresa njena. Ako njen odgovor stigne prvi do Alisinog davaoca Internet usluga, biće keši- ran, a pravi odgovor će biti odbačen kao netražen odgovor na zahtev koji više ne

postoji. Navođenje DNS servera da instalira pogrešnu IP adresu naziva se DNS laži- ranje (engl. *DNS spoofing).* Keš koji sadrži IP adrese namerno promenjene na ovaj način zove se zatrovani keš (engl. *poisoned cache).*

U stvari se sve ne odvija tako jednostavno. Prvo, Alisin davalac Internet usluga proverava da li odgovor nosi ispravnu izvorišnu IP adresu osnovnog servera. Međutim, pošto Trudi u to IP polje može da stavi bilo šta, ona lako može da prođe proveru jer IP adrese osnovnih servera moraju da budu javne.



1. Daj mi Bobovu IP adresu
2,36.1.2.3 (Bobova IP adresa)
3.
4.
VVeb strane

(a)

1. Daj mi Bobovu IP adresu
2.42.9.9.9 (Trudina IP adresa)
GETindex.html 3.GETindex.html
Bobova Web strana 4. Trudina varijanta Bobove

(b)

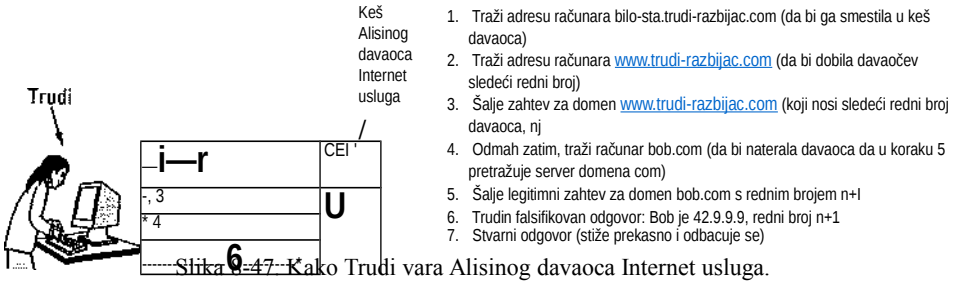
Slika 8-46. (a) Normalna situacija, (b) Napad zasnovan na provaljivanju u DNS server i menjanju Bobovog zapisa.

Drugo, da bi DNS server mogao da poveže odgovore sa zahtevima, svaki zahtev nosi redni broj. Trudi će najlakše saznati tekući redni broj ako i sama registruje do- men, recimo, *trudi-razbijac.com.* Pretpostavimo daje IP adresa ovog domena takođe 42.9.9.9. Ona za svoj novi domen pravi i DNS server *dns.trudi-razbijac.com.* I njegova IP adresa je 42.9.9.9 pošto Trudi ima samo jedan računar. Sada ona mora da privuče pažnju Alisinog davaoca Internet usluga na svoj DNS server. To je lako. Treba samo da od Alisinog davaoca Internet usluga zatraži adresu računara *bilo-sta.trudi- razbijac.com,* što će ga naterati da od osnovnog servera *com* zatraži odgovor ko opslužuje Trudin novi domen.

Kada se *dns.trudi-razbijac.com* jednom nađe u kešu Alisinog davaoca Internet usluga, može da počne pravi napad. Trudi sada od Alisinog davaoca traži adresu www.trudi-razbijac.com. Alisin davalac Internet usluga naravno taj zahtev šalje Tru- dinom DNS serveru. Taj zahtev nosi redni broj koji Trudi očekuje. Sledećeg trenutka, Trudi inu upućuje nov zahtev tražeći Bobovu adresu, a odmah zatim odgovara na sopstveni zahtev šaljući Alisinorn davaocu falsifikovan odgovor u ime servera osnovnog domena *com:* „*bob.com* je 42.9.9.9“. Taj falsifilcovani odgovor nosi redni broj za jedan veći od broja koji je upravo primila. Kada to već ide tako lako, ona može da

pošalje i drugi, treći ili čak desetinu falsifikovanih odgovora čiji redni brojevi uzastopno rastu za jedan. Jedan od njih će sigurno upaliti, a ostali će jednostavno biti odbačeni. Kada Trudin falsifikovan odgovor stigne, kešira se; pravi odgovor koji stiže kasnije odbacuje se jer za njega ne postoji zahtev.

Kada Alisa sada zatraži Bobovu adresu, dobijaTrudinu adresu, 42.9.9.9. Trudi je uspešno organizovala posrednički napad sedeći udobno u svom domu. Pojedinačni koraci ovog napada prikazani su na slici 8-47. Situacija je i crnja nego što izgleda jer se DNS može zavarati na još mnogo načina.



Slika 8-47. Kako Trudi vara Alisinog davaoca Internet usluga.

Bezbedni DNS

Opisana vrsta napada može se izbeći ako DNS serveri, umesto uzastopnih rednih brojeva, koriste nasumično izabrane identifikatore, ali izgleda da se posle svakog „krpljenja“ koda pojavljuje nova „rupa“. Stvarni izvor problema je to što je DNS sistem stvoren u vreme kada je Internet bio alatka za povezivanje nekoliko stotina univerziteta, a tada nisu postojali ni Alisa, ni Bob, ni Trudi. Tada se nije postavljalo pitanje bezbednosti; važnije je bilo postići da Internet uopšte radi. Vremenom se okruženje radikalno promenilo, tako da je IETF 1994. godine oformio radnu grupu za osnovno obezbeđivanje DNS sistema. Taj projekat je poznat kao **DNS bezbednost** (engl. *DNS security, DNSsec),* a njegovi rezultati su prikazani u RFC dokumentu 2535. Nažalost, DNSsec još uvek nije potpuno zamenio stari sistem, tako da su brojni DNS serveri i dalje podložni napadima lažiranja adresa.

Osnovni princip na kome se zasniva sistem DNSsec izuzetno je jednostavan. To je šifrovanje javnim ključem. Za svaku DNS zonu (u smislu slike 7-4) važe dva ključa (javni i privatni). Sve informacije koje šalje DNS server potpisuju se privatnim ključem izvorišne zone, pa primalac može da proveri njihovu autentičnost.

Sistem DNSsec nudi tri osnovne usluge:

1. Potvrđivanje porekla podataka.
2. Distribuiranje javnog ključa.
3. Proveravanje identiteta transakcija i zahteva.

Osnovna je prva usluga: podatke koji se šalju u odgovor na zahtev potvrđuje vlasnik zone. Druga usluga je korisna za bezbedno skladištenje i preuzimanje javnih ključeva. Treća služi kao zaštita od napada ponovljenim slanjem poruka i lažiranjem. Obratite pažnju na to da se ne nudi usluga obezbeđivanja pošto se sve informacije u DNS sistemu smatraju javnim. Pošto se očekuje da uvođenje sistema DNSsec može potrajati godinama, neophodno je omogućiti saradnju između DNSsec i običnih DNS servera, što znači da se sam protokol ne može menjati. Razmotrimo sada neke detalje sistema.

DNS zapisi se grupišu u **skupove zapisa resursa** (engl. *Resource Record Sets, RR - Sets),* pri čemu svaki skup sadrži zapise istog imena, klase i tipa. RRSet može, na primer, da sadrži više A zapisa ako se DNS ime prevodi u primarnu i sekundarnu IP adresu. RRSetovi su proširenі mnogim novim tipovima zapisa (o kojima govorimo u nastavku). Svaki RRSet se kriptografski hešira (npr. algoritmima MD5 ili SHA-1). Heš se potpisuje privatnim ključem zone (npr. koristeći algoritam RSA). Potpisani RRSet je jedinična količina podataka koja se šalje klijentu. Pošto primi potpisani RRSet, klijent može proveriti da li je on potpisan privatnim ključem izvorišne zone. Ako je potpis valjan, podaci se piihvataju. Pošto svaki RRSet sadrži sopstveni potpis, RRSetovi se mogu keširati bilo gde - čak i na nepouzdanim serverima - a da se ne naruši bezbednost.

DNSsec uvodi više novih tipova zapisa. Prvi je zapis *KEY.* U tom zapisu su javni ključ zone, korisnik, računar ili neki drugi principal, kriptografski algoritam koji se koristi za potpisivanje, protokol koji se koristi za prenos i još nekoliko drugih podataka. Javni ključ se u zapisu čuva nešifrovan. Sertifikati prema standardu X.509 ne koriste se jer su preveliki. Polje za algoritam sadrži vrednost 1 za potpise algoritmima MD5/RS A (podrazumevani izbor) ili druge vrednosti za druge kombinacije. Polje za protokol može da sadrži i naznaku da se koristi IPsec ili neki drugi bezbednosni protokol.

Drugi zapis novog tipa je *SIG.* On sadrži heš potpisan algoritmom zadatim u zapisu *KEY.* Potpis važi za sve zapise RRSeta, uključujući i svaki postojeći *KEY* zapis, osim za sam zapis *SIG. On* sadrži i početak i kraj roka važnosti potpisa, ime potpi- sivača i još nekoliko drugih stavki.

Ustrojstvo sistema DNSsec je takvo da se privatni ključ zone ne mora čuvati na mreži. Sadržaj baze podataka za zonu može se jednom ili dvaput na dan ručno prebacivati (npr. na komapkt disku) na samostalni računar na kome se čuva privatni ključ. Tada se mogu potpisati svi RRSetovi i tako dobijeni zapisi *SIG* prebaciti ponovo na osnovni server zone na kompakt disku. Na taj način se privatni ključ na lcompakt disku može čuvati u sefu i iz njega vaditi samo jednom ili dvaput na dan kada treba potpisati nove RRSetove (na računam koji nije umrežen). Kada se završi potpisivanje, sve kopije ključa se brišu iz memorije i čvrsti disk i kompakt disk vraćaju se u sef. Takav postupak svodi elektronsku bezbednost na fizičko obezbeđivanje, nešto što funk- cioniše prilično uspešno.

Prethodno potpisivanje RRSetova umnogome ubrzava odgovaranje na zahteve, pošto ništa ne treba šifrovati u hodu. Međutim, za čuvanje svih ključeva i potpisa iz DNS baza podataka potreban je veliki prostor na disku. Neki zapisi se posle potpisivanja desetostruko povećavaju.

Kada klijentski proces dobije potpisani RRSet, on na njega mora da primeni javni ključ izvorišne zone da bi dešifrovao heš, mora da izračuna heš i da uporedi dve vrednosti. Ako se vrednosti slože, podaci se smatraju ispravnim. Međutim, da bi ovaj postupak radio, klijent

mora imati javni ključ zone. Jedno rešenje je da ga preuzme s pouzdanog servera putem bezbedne veze (npr. uz korišćenje PSec).

U praksi se, međutim, očekuje da su svi klijenti unapred konfigurisani tako da znaju javne ključeve svih osnovnih domena. Ako Alisa sada želi da poseti Bobovu Web lokaciju, ona od DNS servera može da zatraži RRSet za *bob.com* koji će sadržati njegovu IP adresu i zapis *KEY* s Bobovim javnim ključem. Taj RRSet potpisaće server osnovnog domena *com* tako da Alisa lako može da proveri njegovu autentičnost. Na slici 8-48 prikazano je šta ovakav RRSet može da sadrži.

Kada ima proverenu kopiju Bobovog javnog ključa, Alisa od Bobovog DNS servera (koji održava Bob) može da zahteva IP adresu računara www.bob.com. Taj RRSet bide potpisan Bobovim privatnim ključem, tako da Alisa može da proveri potpis RRSeta koji će joj se vratiti od Boba. Ako Trudi nekako uspe da ubaci falsifikovan RRSet u bilo koji keš, Alisa ce lako utvrditi da on nije autentičan jer će zapis *SIG* u njemu biti pogrešan.

Međutim, DNSsec ima i kriptografski mehanizam za pridruživanje odgovora određenom zahtevu kako bi se Trudi onemogućio napad lažiranjem sličan onom koji je prikazan na slici 8-47. Tom (opcionom) merom se odgovora dodaje heš zahteva potpisan privatnim ključem onoga ko šalje odgovor. Pošto Trudi ne zna privatni ključ servera osnovnog domena *com,* ona ne može da falsifikuje odgovor na zahtev koji mu je poslao Alisin davalac Internet usluga. Njen odgovor će sigurno brže stići, ali će biti odbačen zbog pogrešnog potpisa heširanog zahteva.

Ime domena	Životni vek	Klasa	Tip	Vrednost
bob.com.	86400	IN	A	36.1.2.3
bob.com.	86400	IN	KEY	3682793A7B73F731029CE2737D...
bob.com.	86400	IN	SIG	86947503A8B848F5272E53930C...

Slika 8-48. Primer RRSeta za *bob.com.* Zapis *KEY* *je* Bobov javni ključ. Zapis *SIG* je heš zapisa *A* i *KEY* čiju autentičnost svojim potpisom potvrđuje server osnovnog domena *com.*

DNSsec podržava i nekoliko dragih tipova zapisa. Zapis *CERT* se, na primer, može koristiti za čuvanje (npr. X.509) sertifikata. Takav zapis je uveden jer neki žele da DNS sistem pretvore u PKI. Videćemo da li će se to stvarno i desiti. Ovim ćemo završiti objašnjavanje sistema DNSsec. Ko želi da sazna više, neka pročita RFC dokument 2.53.5.

Imena sa sopstvenim sertifikatom

Bezbedni DNS nije jedina mogućnost za obezbeđivanje imena. Za bezbedni sistem datoteka (engl. *Secure File System)* izabran je sasvim drugačiji pristup (Ma- zieres i sar.,

1999). U tom projektu autori su smislili bezbedan, proširiv, svetski sistem datoteka ne dirajući (standardni) sistem DNS i ne oslanjajući se na sertifikate i PKI. U ovom ođeljku ćemo pokazati kako se njihove ideje mogu primeniti na Web. Zbog toga pri opisivanju nećemo koristiti terminologiju samog sistema datoteka upotre- bljenu u radu Mazieresa i saradnika, već Web terminologiju. Međutim, odmah se ograđujemo: iako se šema koju ćemo opisati *može* primeniti na Web da bi se on u visokom stepenu obezbedio, ona se još uvele ne koristi jer za njeno uvođenje treba znatno izmeniti postojeći softver.

Počecemo pretpostavkom da svaki Web server ima dva ključa: javni i privatni. Bez- bedni sistem datoteka suštinski se zasniva na ideji da svaka URL adresa kao svoj deo sadrži kriptografski heš imena servera i javnog ključa. Na primer, na slici 8-49 vidimo URL adresu Bobove fotografije. Ona počinje uobičajenom šemom *http,* iza koje sledi DNS ime servera (www.bob.com). Zatim sledi dvotačka i heš dužine 32 znaka. Na kraju je, kao što je uobičajeno, ime datoteke. Bez heša, to je standardna URL adresa. S hešom, to je URL **adresa sa sopstvenim sertifikatom** (engl. *self-certifying URL).*

Server SHA-1 (Javni ključ servera) Ime datoteke

http://www.bob.com:2g5hd8bfjkc7mf6hg8dgany23xds4pe6/photos/bob.jpg

Slika 8-49. URL adresa sa sopstvenim sertifikatom koja sadrži heš imena servera i javnog ključa.

Čemu heš? Heš se izračunava tako što se na DNS ime servera nadoveže njegov javni ključ i sve propusti kroz algoritam SHA-1 da bi se dobio 160-bitni heš. U ovoj šemi, heš je predstavljen sekvencom od 32 znaka, sastavljenom od cifara i malih slova, osim slova „1“ i „0“ i cifara „1“ i „0“, da bi se izbegla konfuzija. Tako od 26 malih slova (engleske) abecede i 10 cifara, preostaje 32 znaka. Svaki od 32 preostala znaka može da kodira 5-bitni niz. Dakle, sekvenca od 32 znaka može da predstavi 160-bitni SHA-1 heš. U stvari, ne mora se koristiti heš - dovoljan je i sam ključ. Prednost heša je to što skraćuje ime.

U najjednostavnijoj (i najnepogodnijoj) varijanti, ako Alisa želi da pogleda Bobovu fotografiju, samo će u svoj čitač upisati sekvencu sa slike 8-49. Čitač će poslati poruku Bobovoj Web lokaciji tražeći njegov javni ključ. Kada dobije Bobov javni ključ, čitač će ga nadovezati na ime servera i sve heširati. Ako se rezultat složi s hešom dužine 32 znalca u bezbeđnoj URL adresi, čitač je siguran da ima Bobov javni ključ. Na kraju krajeva, kada bi Trudi i uspela da presretne zahtev i da falsifikuje odgovor, ona ne može da dođe do javnog ključa koji bi dao očekivani heš. Svako njeno mešanje biće odmah otkriveno. Bobov javni ključ se može sačuvati u kesu za kasnije korišćenje.

Sada Alisa treba da proveri da li Bob ima odgovarajući privatni ključ. Ona sastavlja poruku koja sadrži predloženi AES ključ sesije, jednokratni uzorak i vremensku oznaku. Tada šifruje poruku Bobovim javnim ključem i šalje mu je. Pošto samo Bob ima odgovarajući privatni ključ, samo Bob može da dešifruje poruku i da kao odgovor pošalje jednokratni uzorak šifrovan AES ključem. Kada primi ispravan jednokratni uzorak šifrovan AES ključem, Alisa zna da razgovara s Bobom. Alisa i Bob sada ta- lcođe imaju AES ključ sesije kojim mogu šifrovati naknadne zahteve *GET* i odgovore.

Kada Alisa dobije Bobovu fotografiju (ili bilo koju njegovu Web stranu), može da je obeleži, tako da sledeći put ne mora da upisuje čitavu njenu URL adresu. Staviše, i URL adrese ugrađene u Web strane mogu da budu sa sopstvenim sertifikatima; one se koriste na uobičajen način, a dodatno ste sigurni da ste dobili upravo stranu koju ste tražili. Početno

upisivanje adrese sa sopstvenim sertifikatom možete izbeći i ako je dobijete preko obezbeđene veze s pouzdanim serverom ili u okviru X.509 sertifilcata koju je potpisala CA organizacija.

Još jedan način (automataskog) dobijanja URL adresa sa sopstvenim sertifikatima jeste povezivanje s pouzdanom mašinom za pretraživanje tako što se upiše njena ser- tificirana URI, adresa (prvi put) i prođe kroz već opisanu proceduru, čime se ostvaraje bezbedna, proverena veza s njom. Mašina za petraživanje se tada može koristiti na uobičajen način, a rezultati će se pojavljivati na potpisanoj strani prepunoj URL adresa sa sopstvenim sertifikatom koje se ne moraju upisivati, već jednostavno pritiskati.

Pogledajmo sada kako ova šema odoleva pokušajima Trudinog DNS lažiranja. Ako Trudi uspe da zatruje keš Alisinog davaoca Internet usluga, Alisin zahtev neće stići Bobu, već će završiti kod nje. Ali protokol sada zahteva da primalac početne poruke (Trudi) vrati javni ključ koji proizvodi ispravan heš. Ako Trudi vrati svoj javni ključ, Alisa će to odmah primetiti jer se SHA-1 heš neće poklopiti sa sertificiranim URL-om. Ako Trudi vrati Bobov javni ključ, Alisa neće otkriti napad, ali će sledeću poruku šifrovati Bobovim ključem. Trudi će dobiti poruku, ali neće moći da je dešifruje da bi iz nje izvukla AES ključ i jednokratni uzorak. Na ovaj ili onaj način, pokušaji DNS lažiranja sada mogu da proizvedu samo napad radi blokiranja usluga.

8.9.3 SSL - sloj bezbednih utičnica

Obezbeđivanje imena je dobar početak, ali je za bezbednost Weba potrebno mnogo više. Sledeći korak je obezbeđivanje veza i sada ćemo razmotriti kako se to može sprovesti.

Kada je Web postao javno dobro, najčešće je korišćen za distribuiranje statičnih strana. Međutim, ubrzo su neke kompanije došle na ideju da ga koriste za finansijske transakcije: kupovinu kreditnom karticom, mrežno bankarstvo i elektronsko učešće na berzi. Za takve aplikacije su bile neophodne bezbedne veze. Godine 1995, korporacija Netscape Communications, tada glavni proizvođač Web čitača, odgovorila je na iskazanu potrebu uvodeći bezbednosni paket pod nazivom **Sloj bezbednih utičnica** (engl. *Secure Socket Layer*, *SSL)*. Taj softver i njegovi protokoli sada se široko koriste, takođe i u Internet Explorera, pa ga vredi malo detaljnije razmotriti.

SSL pravi bezbednu vezu između dve utičnice, što podrazumeva

1. Dogovaranje parametara izumeđu klijenta i servera.
2. Međusobnu proveru identiteta klijenta i servera.
3. Tajno komuniciranje.
4. Zaštitu integriteta podataka.

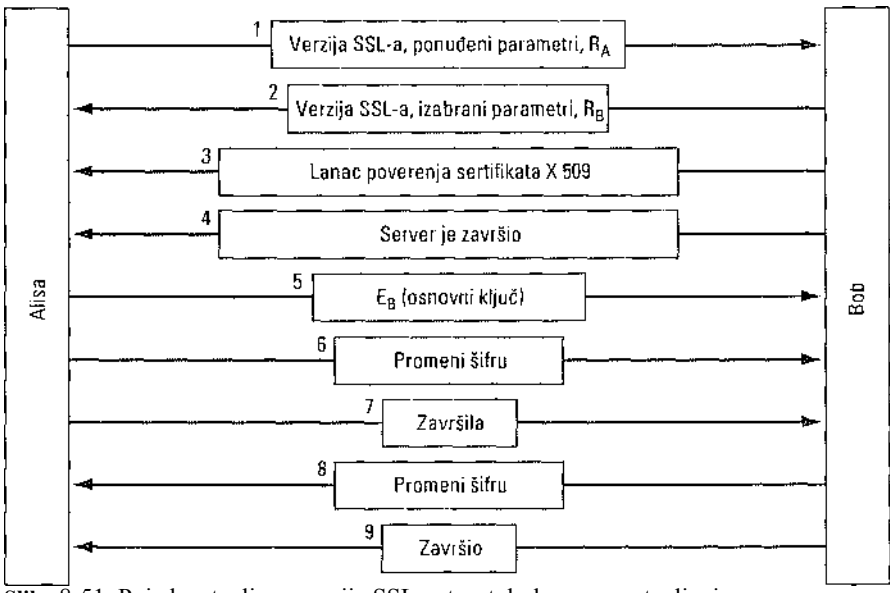
Sve to smo već objašnjavali, pa nema potrebe da ponavljamo.

Mesto SSL-a u uobičajenom skupu protokola prikazano je na slici 8-50. To je praktično nov sloj između sloja aplikacija i transportnog sloja, koji prihvata zahteve od čitača i šalje ih naniže protokolu TCP za slanje servera. Pošto se uspostavi bezbedna veza, SSL uglavnom komprimuje i šifruje podatke. Kada se protokol HTTP koristi preko sistema SSL, to se zove **bezbedni HTTP** (engl. *Secure HTTP*, *HTTPS),* iako se i dalje koristi standardni HTTP protokol. Međutim, on ponekada ne radi preko standardnog priključka (80), već preko novog priključka (443). Recimo uzgred da se SSL ne mora koristiti samo s čitačima Weba, ali je to njegova najčešća primena.

Slika 8-50. Slojevi (i protokoli) za kućni čitač Weba koji koristi SSL.

Protokol SSL pretrpeo je više izmena. Nadalje ćemo govoriti samo o verziji 3 jer se najčešće koristi. SSL podržava niz različitih algoritama i opcija. Opcije obuhvataju prisustvo ili odsustvo kompresije, algoritme za šifrovanje koji će se koristiti i neke detalje koji se tiču izvoznih ograničenja za šifre. Ovo poslednje treba da osigura da će se ozbiljno šifrovanje koristiti samo kada su oba kraja veze u SAD. U ostalim slučajevima, ključ se ograničava na 40 bitova, što kriptografi smatraju lošim vicem. Netscape je bio prisiljen da ugradi ovo ograničenje kako bi dobio izvoznu dozvolu od Vlade SAD.

SSL sadrži dva potprotokola: jedan za uspostavljanje bezbedne veze i drugi za njeno korišćenje. Razmotrimo najpre kako se uspostavlja bezbedna veza. Protokol za uspostavljanje veze prikazan je na slici 8-51. On počinje porukom 1 - Alisinim zahte- vom Bobu za uspostavljanje veze. U zahtevu se navodi verzija sistema SSL koju Ali- sa ima i njene želje u pogledu komprimovanja podataka i algoritama za njihovo šifrovanje. Zahtev sadrži i jednokratni uzorak R_A koji će se koristiti kasnije.



Slika 8-51. Pojednostavljena verzija SSL potprotokola za uspostavljanje veze.

Sada je red na Boba. U poruci 2 Bob bira jedan od ponuđenih algoritama koje Alisa može da podrži i šalje svoj jednokratni uzorak R_B. Zatim, u poruci 3, Bob šalje serti- fikat sa svojim javnim ključem. Ako sertifikat nije potpisala neka opštepoznata ovla- šćena CA, on šalje i lanac poverenja kojim se može stici do takve CA. Svi čitači, uključujući i Alisin, isporučuju

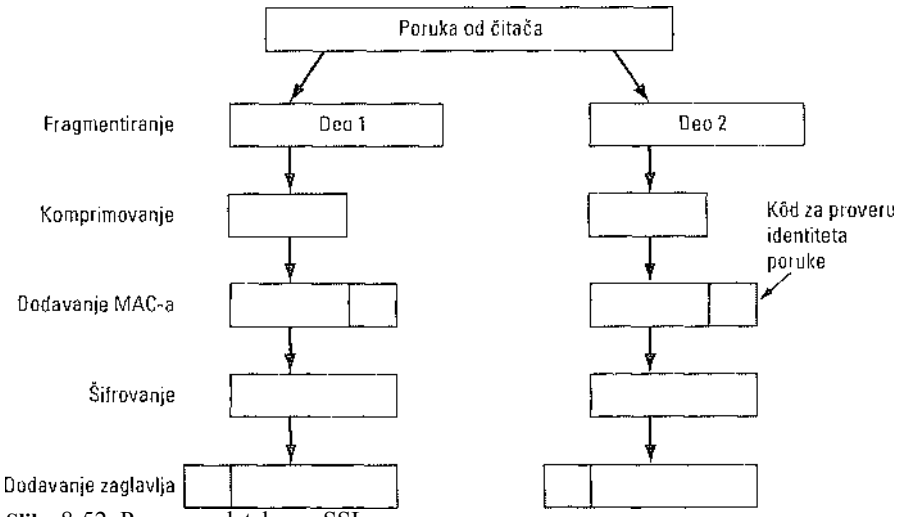
se sa preko 100 javnih ključeva, pa ako Bob može da uspostavi lanac poverenja do jednog takvog pouzdanog polazišta, Alisa će moći da proveri Bobov javni ključ. U ovom trenutku Bob može da pošalje i neke druge poruke (npr, zahtev za sertifikat Alisinog javnog ključa). Kada sve završi, Bob šalje Alisi poruku 4 da bi naznačio da je red na nju.

Alisa odgovara tako što nasumice bira 384-bitni **osnovni ključ** (engl. *premaster key)* i šalje ga Bobu šifrovanog njegovim javnim ključem (poruka 5). Ključ sesije koji se koristi za šifrovanje podataka izvodi se iz osnovnog ključa i oba jednokratna uzorka na složen način. Pošto Bob primi poruku 5, i on i Alisa mogu da izračunaju ključ sesije. Zbog toga Alisa nalaže Bobu da pređe na novu šifru (poruka 6) i istovremeno mu saopštava daje završila s protokolom za uspostavljanje veze (poruka 7). Bob joj obe poruke potvrđuje (poruke 8 i 9).

Međutim, iako Alisa zna ko je Bob, Bob ne zna ko je Alisa (osim ako Alisa ima javni ključ i odgovarajući sertifikat za njega, što baš nema svaki pojedinac). Prema tome, Bobova prva poruka može da bude i zahtev Alisi da se prijavi koristeći prethodno uspostavljeno korisničko ime i lozinku. Protokol za prijavljivanje je, međutim, izvan domašaja sistema SSL. Kada se sve to jednom završi na bilo koji način, može da počne prenos podataka.

Kao što smo već rekli, SSL podržava razne algoritme za šifrovanje. Najmoćniji od njih koristi trostruki DES s tri zasebna ključa za šifrovanje i SHA-1 za integritet poruke. Ta kombinacija je srazmerno spora, pa se najčešće koristi za elektronsko bankarstvo i aplikacije za koje je neophodan najviši stepen bezbednosti. Za obične trgovačke primene koristi se RC4 sa 128-bitnim ključem za šifrovanje i MD5 za pro- veravanje identiteta poruke. RC4 koristi 128-bitni ključ kao klicu (engl. *seed)* i proširuje ga za interno korišćenje u mnogo veći broj. Zatim taj interni broj koristi da bi generisao neprekidni ključ. Neprekidni ključ se podvrgava isključivoj disjunkciji sa osnovnim tekstom u klasičnom režimu uzastopnog šifrovanja, kao što smo videli na slici 8-14. Izvozne verzije takođe koriste RC4 sa 128-bitnim ključevima, ali su njihovih 88 bitova javni, tako da se šifra lako može provaliti.

Za prenos podataka se koristi dragi potprotokol (slika 8-52). Poruke od čitača prvo se razbijaju na jedinice veličine do 16 KB. Ako je uključena kompresija, svaka jedinica se zasebno komprimuje. Posle toga se tajni ključ koji je izveden od dva jednokratna uzorka i osnovnog ključa nadovezuje na komprimovani tekst i rezultat hešira dogovorenim algoritmom za heširanje (obično algoritmom MD5). Ovaj heš se pridružuje svakom fragmentu kao kod za provera identiteta poruke (MAC). Komprimovani fragment i MAC tada se šifruju dogovorenim algoritmom za simetrično šifrovanje (obično podvrgavanjem isključivoj disjunkciji sa neprekidnim RC4 ključem). Na kraju se fragmentu dodaje zaglavlje i sve se prenosi TCP vezom.



Slika 8-52. Prenos podataka uz SSL.

Treba ipak upozoriti na oprez. Postoje pokazano da RC4 ima neke slabe ključeve koji se lako mogu analizirati, bezbednost SSL-a zasnovana na RC4 stoji na klimavim nogama (Fluhrer i sar., 2001). Čitače koji korisnicima omogućavaju da biraju skup šifara treba konfigurisati tako da sve vreme koriste trostruki DES sa 168-bitnim ključevima i SHA-1, iako je ta kombinacija sporija od RC4 i MD5.

Dragi problem je to što principali možda nemaju sertifikate ili to što dok koriste SSL ne proveravaju uvek da li korišceni ključevi odgovaraju sertifilcatima.

Godine 1996, korporacija Netscape Communications ponudila je SSL organizaciji IETF za standardizovanje. Rezultat je bio **Bezbednost transportnog sloja** (engl. *Transport Layer Security, TLS),* protokol opisan u RFC dokumentu 2246.

SSL je srazmerno malo izmenjen, ali dovoljno da njegova 3. verzija ne može da sa- rađuje sa TLS-om. Izmenjen je, na primer, način na koji se ključ sesije izvodi iz osnovnog ključa i jednokratnih uzoraka da bi postao otporniji na kriptoanalizu. TLS je poznat i kao SSL verzije 3.1. Prve realizacije su se pojavile 1999, ali još uvek nije jasno da li će sistem TLS u praksi zameniti SSL, uprkos tome što je malo jači. Međutim, problem sa slabim RC4 ključevima i dalje ostaje.

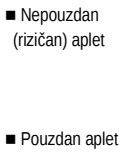
8.9.4 Bezbednost pokretnog koda

Imenovanje i veze su dva područja obezbeđivanja Weba. Međutim, ima još takvih područja. Na početku, kad su sve Web strane bile samo statične HTML datoteke, one nisu sadržale izvršni kod. Danas one često sadrže male programe, uključujući Java aplete, ActiveX kontrole i JavaScriptove. Preuzimanje i izvršavanje takvog **pokretnog koda** (engl. *mobile cocle)* očigledno je skopčano s velikim bezbednosnim rizikom, pa su razvijene mnoge metode da se taj rizik svede na najmanju meni. Ukratko ćemo se osvrnuti na neke probleme s pokretnim kodom i na neka ponuđena rešenja.

Bezbednost Java apleta

Java apleti su mali programi pisani na jeziku Java, koji se prevode u mašinski jezik

zasnovan na korišćenju steka, poznat kao **Javina virtuelna mašina** (engl. *Java Virtual Machine, JVM).* Oni se mogu postaviti na Web stranu i preuzimaju se zajedno s njom. Pošto se strana učita, apleti se smeštaju u JVM interpretator unutar čitača (slika 8-53).



Slika 8-53. Aplete može da interpretira čitač Weba,

Interpretiranje koda ima prednost nad izvršavanjem prevedenog koda jer interpretator svaku instrukciju proveri pre izvršavanja, što mu daje priliku da utvrdi ispravnost adrese na koju ukazuje instrukcija. Pored toga, hvataju se i interpretiraju i sistemski pozivi. Postupak rada s njima zavisi od uspostavljenih bezbednosnih pravila. Na primer, ako je aplet pouzdan (dolazi s lokalnog diska), njegovi sistemski pozivi izvršavaju se bez provere. Međutim, ako je aplet nepouzdan (tj. stigao je sa Interneta), on se može kapsulirati u tzv. **kutiju s peskom** (engl. *sandbox)* da bi se ograničilo njegovo ponašanje i omeli njegovi pokušaji da koristi sistemske resurse.

Kada aplet pokuša da upotrebi sistemski resurs, njegov poziv se prosleđuje na odobrenje programu koji vodi brigu o bezbednosti (monitora). Monitor ispituje poziv s gledišta lokalnih bezbednosnih pravila i donosi odluku da ga odobri ili odbije. Na taj način se apletima može omogućiti pristup nekim, ali ne i svim resursima. U stvarnom životu, nažalost, bezbednosni model loše radi i u njemu se svakodnevno pronalaze novi propusti.

ActiveX

ActiveX kontrole su binarni programi namenjeni izvršavanju na procesorima Pentium i mogu se ugraditi u Web strane. Kada se na strani naiđe na jedan takav pr ogram, prvo se proverava da li bi ga trebalo izvršiti, pa ako program prođe proveru, on se i izvršava. ActiveX kontrola se ne interpretira i ne smešta u kutiju s peskom; ona ima slobodu izvršavanja kao i svaki dragi korisnički program i zato može da nanese veliku štetu. Na taj način, odluka da se ActiveX kontrola izvrši ili ne izvrši predstavlja poslednju (i jedinu) bezbednosnu barijeru.

Za donošenje takve odluke Microsoft je izabrao metodu **potpisivanja koda** (engl. *code signing).* Svakoj ActiveX kontroli pridružuje se digitalni potpis - heš samog koda koji je potpisao njegov autor koristeći javni ključ. Kada naiđe na ActiveX kontrolu, čitač prvo proverava potpis da bi se uverio da program tokom prenosa nije menjan. Ako je potpis ispravan, čitač u svojim internim tabelama proverava da li je autor programa pouzdan ili

postoji lanac poverenja do pouzdanog autora. Ako utvrdi daje autor pouzdan, čitač dozvoljava izvršavanje programa; u suprotnom, zabranjuje ga. Microsoftov sistem za proveru ActiveX kontrola zove se **Authenticode.**

Poučno je uporediti Java aplete i ActiveX kontrole. Kod Java apleta niko ne proverava autora apleta, već interpretator u toku izvršavanja vodi računa o tome da apleti ne rade nešto što im vlasnik računara ne dozvoljava. Za razliku od toga, kod potpisanog pokretnog koda ActiveX kontrola više se ne vodi računa o tome kako se kod tokom izvršavanja ponaša. Ako je stigao iz pouzdanog izvora i nije izmenjen u prenosu, samo se izvršava - ne proverava se njegova eventualna zlonamernost. Ukoliko je prvobitni autor *namerno* napravio kod koji će formatirati čvrsti disk i izbrisati fleš ROM, tako da više nikada ne možete da pokrenete računar, i ako taj programer ima sertifikat da je pouzdan, njegov kod će biti izvršen, a računar uništen (osim ako su ActiveX kontrole globalno deaktivirane u čitaču).

Mnogi misle daje poklanjanje poverenja nepoznatoj softverskoj kompaniji opasno. Da bi to demonstrirao, jedan programer u Sijetlu osnovao je softversku kompaniju i pribavio joj sertifikat o pouzdanosti, što nije bilo teško. Zatim je napravio ActiveX kontrolu koja gasi računar na uobičajen način i razdelio je na sve strane. Kontrola je zaista ugasila mnogo računara, pa su se morali ponovo pokretati, ali bez ikakve štete za sistem. On je na taj način samo pokušao da predoči problem korisnicima. Njegov sertifikat za tu ActiveX kontrolu zvanično je povučen, čime je prekinuta kratka epizoda akutnog nezadovoljstva korisnika, ali osnovni propust nikada nije otklonjen i samo čeka nekog zlonamernika da ga iskoristi (Garfinkel i Spafford, 2002). Pošto je nemoguće uvesti policijski nadzor u hiljade softverskih kompanija koje bi mogle pisati pokretni kod, tehnika potpisivanja koda je katastrofa koja samo što se nije desila.

JavaScript

Za JavaScript ne postoji formalan bezbednosni model, ali realizacije ovih skripto- va vrve od propusta. Svaki proizvođač obezbeđuje sistem na drugi način. Na primer, Netscape Navigator u svojoj verziji 2 koristi nešto slično modelu Jave, ali je u verziji 4 to napušteno u korist modela potpisanog koda.

Dopuštanje stranom kodu da se izvršava na vašem računam ne znači ništa drugo do prizivanje nevolje. S gledišta bezbednosti to je isto kao kada biste pozvali provalnika u kuću, a zatim ga pažljivo pratili da ne bi možda iz kuhinje prešao u sobu. Ako se u međuvremenu desi nešto nepredviđeno (npr. zazvoni telefon), stvar se može završiti rđavo. Nevolju stvara i to što pokretni kod nudi blještavu grafiku i brzu interakciju, pa mnogi Web dizajneri smatraju daje to važnije od bezbednosti, naročito ako se rizik prebaci na korisnikov računar.

Virusi

Virusi su još jedna vrsta pokretnog koda, samo, za razliku od prethodnih primera, niko ne žudi za njima. Virusi se od običnog pokretnog koda razlikuju po tome što su napravljeni s namerom da se umnožavaju. Kada virus stigne u računar s Web stranom, prilogom poruci e-pošte ili na neki drugi način, on svoj „život“ obično započinje infici- ranjem izvršnih datoteka na disku. Kada se neki od tih programa izvrši, virus preuzme upravljanje, obično s namerom da se proširi na druge računare, na primer, umnožavanjem preko e-pošte na računare svih korisnika iz adresara prvobitne žrtve. Neki virusi inficiraju pokretački sektor čvrstog diska, tako da se izvršavaju zajedno s pokretanjem računara. Virusi su postali veliki problem na Internetu i dosad su izazvali štete u vred- nosti više milijardi dolara. Za njih nema brzog rešenja. U tom pogledu možda može da pomogne potpuno nova generacija operativnih sistema zasnovanih na bezbednim mikrojezgrima (engl. *microkernels)* i strogoj kategorizaciji korisnika, procesa i resursa.

8.10 DRUŠTVENI ASPEKTI

Internet i njegova bezbednosna tehnologija predstavljaju područje sukobljavanja društvenih problema, pravila javnog ponašanja i tehnologija - često sa ozbiljnim po-sledicama. U nastavku čemo se samo osvrnuti na tri područja: privatnost, slobodu izražavanja i autorska prava. Ne treba posebno naglašavati da ćemo te probleme samo načeti. Kao dodatno štivo preporučujemo Andersona (2001), Garfinkela i Spafforda (2002), i Schneiera (2000). I na Internetu se može pronaći mnogo materijala. Samo u mašinu za pretraživanje upišite reči: „privacy“, „censorship“ i „copyright“. Neke hiperveze naći ćete i na Web lokaciji posvećenoj ovoj knjizi.

8.10.1 Privatnost

Da li ljudi imaju pravo na privatnost? Dobro pitanje. Četvrti amandman Američkog ustava zabranjuje vladi da pretresa stanove, hartije i račune građana bez jakog razloga, i ograničava situacije u kojima Se može izdati nalog za pretres. Proizlazi da je problem privatnosti na sceni već više od 200 godina, barem u SAD.

Glavne novosti na tom polju poslednje decenije jesu lakoća s kojom vlade mogu da špijuniraju svoje građane i lakoća s kojom građani mogu da spreče takvo špijuniranje. Kada je u 18. veku vlada želela da pregleda dokumentaciju određenog građanina, morala je da na njegovu farmu pošalje policajca na konju sa zahtevom za takvo pregle- danje. Bila je to nezgrapna procedura. Danas telefonske kompanije i davaoci Internet usluga rado instaliraju „prisluškivače“ kada im se predoči zvaničan nalog. Policajcima je tako mnogo lakše, a nema ni opasnosti da padnu s konja (ako zadremaju).

Kriptografija je sve to promenila iz temelja. Svako ko se pomuči da preuzme i instalira PGP i ko koristi dobro čuvan ključ „za vanzemaljce“, može biti prilično siguran da niko u svemiru neće čitati njegovu e-poštu, s nalogom ili bez njega. Ljudi iz vlade to dobro znaju i nisu baš oduševljeni. Stvarna privatnost znači da im je sada mnogo teže da špijuniraju kriminalce svih boja, ali talcođe i novinare ili političke protivnike.

Zbog toga su neke vlade ograničile ili zabranile korišćenje ili izvoz kriptografije. U Francuskoj, na primer, sva kriptografija je do 1999. godine bila zabranjena, osim ako se ključevi predaju vladi.

Francuska nije jedina. Aprila 1993. godine, Američka vlada je najavila nameru da hardverski kriptoprocesor (engl. *clipper chip)* propiše kao standard za sve mrežne komunikacije. Tvrdilo se da će na taj način biti zagarantovana privatnost građana. Pomenuto je takođe da čip omogućava vladi da dešifruje sav saobraćaj pomoću tzv. deponovanja šifara (engl. *key escrow*), čime joj se omogućava pristup svim šiframa. Naravno, sve uz obećanje da će špijuniranja biti samo uz zvaničan sudski nalog. Možete zamisliti kakvu je to raspravu izazvalo između pobornika privatnosti, koji su čitav plan ispljuvali, i policijskih organa koji su ga uzdizali u nebesa. Na kraju je vlada povukla predlog i potpuno odustala od njega.

Mnogo informacija o elektronskoj privatnosti možete naći na Web lokaciji Electronic Frontier Foundation, www.eff.org.

Anonimni serveri za prosleđivanje e-pošte

PGP, SSL i druge tehnologije omogućavaju da dve strane uspostave bezbednu, proverenu komunikaciju, a da ih neko treći ne nadzire niti ometa. Međutim, ponekad se najbolja privatnost postiže kada se *ništa ne proverava,* tj. kada su obe strane anonimne. Anonimnost je poželjna za poruke koje se šalju od tačke do tačke, za disku- sione grupe, ili u oba slučaja.

Razmotrimo neke primere. Prvo, politički disidenti koji žive pod autoritarnim režimima često žele da anonimno komuniciraju iz straha da će biti zatvoreni ili ubijeni. Drago, propuste u radu mnogih korporacija, obrazovnih, državnih i drugih institucija često u javnost iznose mali ljudi koji uglavnom žele da ostanu anonimni da bi izbegli represalije. Treće, osobe s nepopularnim društvenim, političkim ili religioznim stavovima često žele da međusobno komuniciraju e-poštom ili u diskusionim grupama a da se pri tome ne otkrivaju javnosti. Četvrto, mnogi žele da u diskusionim grupama razgovaraju o alkoholizmu, duševnim bolestima, seksualnim problemima, zloupotrebi dece ili svom životu kao člana proganjane manjine, a da ne moraju to da čine javno. Naravno, postoje i brojni drugi primeri.

Razmotrimo jedan konkretan. Devedesetih godina su izvesni kritičari jedne netradicionalne religiozne skupine objavili svoje stavove u diskusionoj grupi USENET-a preko anonimnog servera za prosleđivanje e-pošte (eng. *anonymous remailer*). Taj server je korisnicima omogućavao da naprave pseudonim i da šalju poruke serveru, koji bi ih pod tim pseudonimom prosleđivao ili objavljivao u diskusionim grupama, tako da niko ne zna odakle stvarno dolaze. Neke takve poruke su obelodanile da ono što religiozna grupa zastupa predstavlja poslovnu tajnu i materijal zaštićen autorskim pravima. Religiozna grupa je reagovala tako što je obavestila lokalne vlasti da su njene poslovne tajne objavljene i njena autorska prava ugrožena, a i jedno i drugo su krivična dela u oblasti gde je bio lociran server. Usledio je sudski proces i operater servera je bio prinuđen da otkrije pravi identitet osoba koje su slale poruke. (Uzgred, ovo nije bio prvi slučaj da su religiozni prvaci pobesneli kada je neko otkrio njihove tajne: William Tyndale je 1536. spaljen na lomači kada je preveo Bibliju na engleski.)

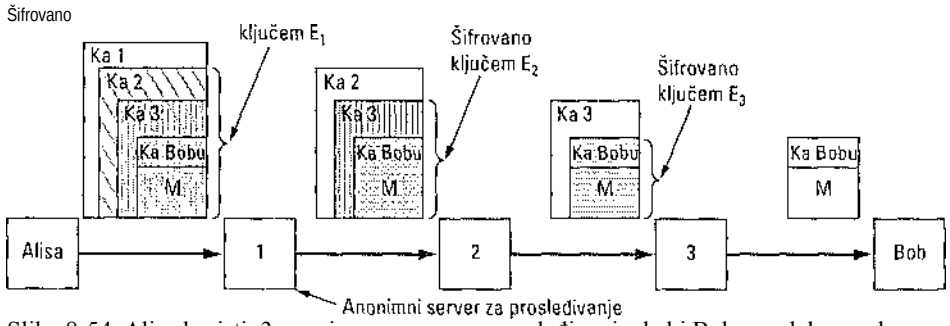
Veliki segment Internet zajednice bio je ogorčen ovim kršenjem prava na privatnost. Zaključak koji se sam nametnuo bio je da anonimni server za prosleđivanje e-pošte koji uporedo čuva stvarne adrese e-pošte i pseudonime (tzv. server za prosleđivanje tipa 1) ne vredi previše. Ovaj slučaj je podstakao mnoge da smisle anonimne servere za prosleđivanje poruka koji bi izdržali napad sudskim nalogom.

Ti novi serveri, često zvani **anonimni serveri za prosleđivanje e-pošte uz šifro- vanje** (engl. *cypherpunk remailers),* rade na sledeci način. Korisnik sastavlja elektronsku poruku,

zajedno sa zaglavljima prema standardu RFC 822 (naravno, osim zaglavlja *From:),* šifruje je javnim ključem anonimnog servera i šalje mu je. Tamo se spoljna RFC 822 zaglavlja uklanjaju, sadržaj se dešifruje i poruka prosleđuje. Anonimni server nema korisničke naloge i ne vodi dnevnike, pa ako kasnije eventualno bude konfiskovan, u njemu se neće naći tragovi poruka koje je prosledio.

Mnogi korisnici koji žele anonimnost ulančavaju svoje zahteve kroz više anonimnih servera, kao na slici 8-54. Tu Alisa želi da pošalje Bobu stvarno, stvarno anonimnu čestitku za Dan Sv. Valentina, pa ulančava tri anonimna servera. Ona sastavlja poruku *M* i dodaje joj zaglavlje s Bobovom adresom e-pošte. Zatim sve šifruje javnim ključem E_3 anonimnog servera 3 (označeno horizontalnom šrafurom). Uz to prilaže adresu e-pošte anonimnog servera 3 u obliku običnog teksta. To je poruka između servera 2 i 3 na slici.

Ona ovu poruku zatim šifruje javnim kl jučem E_2 servera 2 (označeno vertikalnom šrafurom) i dodaje zaglavlje s običnim tekstom koje sadrži adresu e-pošte servera 2. Ta poruka je na slici 8-54 prikazana između servera 1 i 2. Na kraju, ona šifruje čitavu poruku javnim ključem *Ej* servera 1 i dodaje joj zaglavlje sa adresom e-pošte servera 1 u obliku običnog teksta. To je poruka koja je na slici prikazana desno od Alise i to je poruka koja se stvarno šalje.



Slika 8-54. Alisa koristi .3 anonimna servera za prosleđivanje da bi Bobu poslala poruku.

Kada poruka stigne serveru 1, spoljno zaglavlje se uklanja. Telo poruke se dešifruje i zatim šalje serveru 2. Slično se događa i na druga dva servera za prosleđivanje.

Iako će svakome biti izuzetno teško da unazad sledi put poruke sve do Alise, mnogi serveri za prosleđivanje preduzimaju dodatne bezbednosne mere. Na primer, oni mogu čuvati poruke tokom slučajno izabranog perioda vremena, dodavati besmislen tekst na kraj poruke ili takav tekst uklanjati, menjati redosled poruka itd, samo da bi se mogućem špijunu otežalo logičko povezivanje dolaznih i odlaznih poruka na ser- veru i tako omela analiza saobraćaja. Opis sistema koji predstavlja trenutno stanje u razvoju anonimne e-pošte naći ćete kod Mazieresa i Kaashoeka (1998).

Anonimnost se ne iscrpljuje u e-pošti. Postoje i usluge koje omogućavaju anonimno pretraživanje Weba. Korisnik podešava svoj čitač da kao zastupnički sever koristi anonimni server. Tako svi HTTP zahtevi idu anonimnom serveru koji onda stvarno zahteva Web stranu i šalje je korisniku. Za Web lokaciju je anonimni server „stvarni korisnik“. Sve dok anonimni server ne vodi dnevnik, po učinjenom delu niko ne može da utvrdi ko je tražio koju stranu.

8.10.2 Sloboda izražavanja

Privatnost se odnosi na pojedince koji ne žele da Svima pokazuju svoju „ličnu kartu“. Druga zanimljiva društvena tema je sloboda izražavanja i njena suprotnost - cenzura, kojom vlade žele da ograniče ono što njihovi građani čitaju ili objavljuju. Pošto Web sadrži milione strana, to je pravi raj za cenzore. U zavisnosti od prirode i ideologije režima, zabranjeni materijal može da obuhvati Web lokacije sledećeg sadržaja:

1. Materijal nepodesan za decu ili mladež.
2. Govor mržnje usmeren na različite etničke, religiozne, seksualne i druge grupe.
3. Informacije o demokratiji i demokratskim vrednostima.
4. Istorijski materijali koji protivreče zvaničnoj verziji vlade.
5. Priručnici za obijanje brava, pravljenje oružja, šifrovanje poruka itd.

Uobičajen potez režima je zabranjivanje rada takvih lokacija.

Ponekada je rezultat neočekivan. Na primer, da bi omogućili deci da ih koriste, neke javne biblioteke su na svoje računare instalirale Web filtre koji blokiraju pornografske lokacije. Filtri blokiraju lokacije koje se nalaze na njihovim crnim listama, ali i obele- žavaju strane s „nepristojnim“ recima pre nego što ih prikažu. U jednom slučaju, u okrugu Loudoun u Virdžiniji, filtar je blokirao bibliotekarevu pretragu za informacijama o raku dojke jer je ugledao reč „dojka“. Upravnik biblioteke je tužio okrug Loudoun. Međutim, u Livermom, u Kalifomiji, nakon što je jedan dvanaestogodišnjak ulovljen da na računaru gleda pornografiju, njegova majka je tužila javnu biblioteku zato što *nije* instalirala filtar. Šta treba da radi biblioteka?

Mnogima izmiče suština Weba: to je globalna mreža. Ona pokriva čitav svet. Ni dve države se ne slažu u tome šta sme da bude na Webu. Na primer, novembra 2000, francuski sud je naredio kalifornijskoj korporaciji Yahoo da francuskim korisnicima zabrani pristup aukciji nacističkih suvenira koja se održavala na lokaciji Yahoo, budući da se posedovanje takvog materijala kosi s francuskim zakonima. Yahoo se obratio američkom sudu koji ga je podržao, ali je pitanje nadležnosti sudova u takvim situacijama i dalje nejasno.

Zamislite samo šta bi se desilo da neki sud u Juti naredi Francuskoj da blokira Web lokacije koje govore o vinu jer se to ne slaže sa strogim zakonima države Juta. Ili, pretpostavite da Kina predloži zabranjivanje svih Web lokacija koje govore o demokratiji jer to nije u interesu države. Da li iranski religiozni zakoni važe i za mnogo liberalnije Šveđane? Može li Saudijska Arabija da blokira Web lokacije koje se bave zaštitom prava žena? Čitava problematika je jedna velika Pandorina kutija.

Evo jednog primerenog komentara Džona Gilmora: „Mreža tumači cenzuru kao kvar i zaobilazi to područje“. Konkretne realizacije usmerene protiv cenzurisanja poznate su prema Andersonu (1966) kao **usluga večnog trajanja** (engl. *eternity service),* Cilj usluge je da onemogući povlačenje ili prepravljanje onoga što je objavljeno, kao što je bila praksa u Sovjetskom Savezu za vreme Staljina. Kada želi da koristi uslugu večnog trajanja, korisnik zadaje rok trajanja materijala, plača naknadu sra- zmernu roku trajanja i veličini materijala, i objavljuje materijal. Posle toga, materijal niko ne može da povuče ili izmeni, čak ni onaj ko gaje objavio.

Kako se može realizovati takva usluga? Najjednostavniji je model ravnopravnih računara u kome se dokument smešta na desetine servera, od kojih svaki dobija deo uplate kao naknadu za učestvovanje. Serveri treba da su rasuti u što više područja s različitom jurisdikcijom kako bi bili otporniji na različite političke pritiske. Lista od 10 slučajno izabranih servera čuva se bezbedno na više mesta, pa ako neki od servera bude

onesposobljen, njegovo mesto odmah zauzima nov server. Tako predstavnik vlasti zadužen za uništavanje dokumenata nikada ne može biti siguran daje zaplenio sve kopije. Sistem treba da je sposoban i za automatski oporavak: kada se sazna da su neke kopije uništene, ostale lokacije treba da traže nova mesta za njihovo obnavljanje.

Usluga večnog trajanja je bila prvi predlog sistema otpornog na cenzura. Od tada su predloženi još neki sistemi, a neki su i realizovani. Dodate su im razne nove osobine, kao što su šifrovanje, anonimnost i otpornost na greške. Često se datoteke razbijaju na više delova i svaki deo čuva na više servera. Primeri ovih sistema su Freenet (Clarke i sar., 2002), PASIS (Wylie i sar., 2000) i Publius (Waldman i sar., 2000). Ostale možete naći kod Serjantova (2002).

Sve više zemalja danas pokušava da ograniči izvoz nematerijalnih dobara, što uključuje Web lokacije, softver, naučne radove, e-poštu, telefonske servise itd. Čak i u Engleskoj, gde sloboda govora ima viševekovnu tradiciju, ozbiljno se razmatraju restriktivni zakoni koji će, na primer, stručnu raspravu između britanskog profesora i njegovog inostranog studenta na Kembridžu svrstati u izvozne aktivnosti, za šta je neophodna dozvola vlade (Anderson, 2002). Ne treba posebno naglašavati da takva politika vodi mnogim nedoumicama.

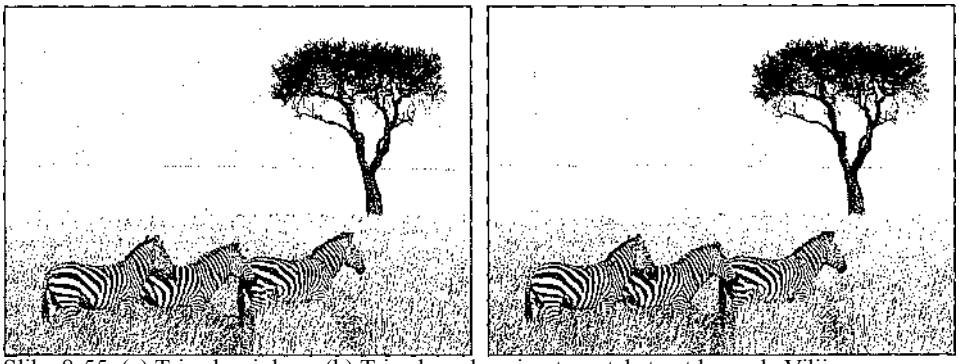
Steganografija

U zemljama u kojima je cenzura obična stvar, disidenti često pribegavaju tehnici da bi je izbegli. Kriptografija omogućava slanje tajnih poruka (iako ni šifrovanje vero- vatno nije legalno), ali ako vlada misli da je Alisa „loš momak“, sama činjenica da ona komunicira s Bobom svrstava Boba u istu kategoriju jer represivna vlada poznaje zakon tranzitivnosti, čak i ako nema pri ruci matematičare. Anonimni serveri za prosle- đivanje poruka u načelu su korisni, ali ako su oni u lokalu zabranjeni, a za poruke preko granice treba pribaviti dozvolu vlade, ne pomažu mnogo. Ali to ne važi za Web.

Osobe koje žele da komuniciraju tajno, često pokušavaju da prikriju činjenicu da uopšte komuniciraju. Veština prikrivanja poruka zove se **steganografija,** prema starogrčkom pojmu „tajno pisanje“. U stvari, stari Grci su je i sami koristili. Herodot je zabeležio daje jedan general ošišao glasnika do glave, na temenu mu utetovirao poruku i poslao ga tek kad mu je kosa ponovo porasla. Savremene tehnike u suštini rade isto, jedino imaju na raspolaganju veći propusni opseg i manje kašnjenje.

Razmotrite konkretan primer steganografije na slici 8-55. Na fotografiji koju je autor snimio u Keniji (a), nalaze se tri zebre pored bagremovog drveta. Fotografija (b) izgleda isto kao i fotografija (a), ali spoljni izgled često vara: ona dodatno sadrži potpun tekst pet Šekspirovih komada: *Hamleta, Kralja Lira, Magbeta*, *Mletačkog trgovca* i *Julija Cezara.* Tekst svih navedenih komada zauzima više od 700 KB.

Kako radi ovaj steganografsld kanal? Originalna slika u boji je veličine 1024 x 768 pilcsela. Svaki pilcsel sadrži tri 8-bitna broja, po jedan za intenzitet crvene, zelene i plave boje. Boja piksela se formira linearnim superponiranjem tri osnovne boje. U steganografskom kodiranju kao tajni kanal se koristi najmanje značajan bit svake od tri RGB vređnosti. Na taj način, svaki piksel ugošćava 3 bita tajnih informacija, po jedan u crvenom, zelenom i plavom području. U slici navedene veličine može se uskla- dištiti do 1024 x 768 x 3 bita ili 294.912 bajtova tajnih informacija.



Slika 8-55. (a) Tri zebre i drvo. (b) Tri zebre, drvo i potpun tekst pet komada Vilijama Šekspira,

Potpun tekst pet Šekspirovih komada i kratka napomena zauzimaju 734.891 bajt. Taj tekst je prvo komprimovan standardnim algoritmom na 274 KB. Komprimovani rezultat je tada šifrovan algoritmom IDEA i unesen redom u najmanje značajne bitove vrednosti boja. Kao što se može videti (u stvari, kao što se ne može videti), informacije su potpuno skrivene. One ostaju skrivene i na velikoj fotografiji u boji. Oko ne može lako da razlikuje 21-bitnu od 24-bitne boje.

Upoređivanje dve crno-bele slike u niskoj rezoluciji ne daje pravi utisak o moći opisane tehnike. Da bi vas u nju ubedio, autor je pripremio demonstraciju koja obuhvata sliku 8-55(b) s pet Šekspirovih komada u punoj boji i visokoj rezoluciji. Ta demonstracija, zajedno sa alatkama kojima se tekst može umetnuti u slike i izvaditi iz njih može se naći na Web lokaciji posvećenoj ovoj knjizi.

Kada požele da iskoriste steganografiju za tajno komuniciranje, disidenti mogu da naprave Web lokaciju koja obiluje „politički podobnim“ fotografijama, npr. Velikog Vođe, lokalnih sportskih događaja, filmskih iTV zvezda. Naravno, slike bi bile „preparirane“ steganografskim porukama. Ako se poruke prvo komprimuju, pa onda šifruju, čak i onaj lco pogađa da one postoje imaće velike muke da ih razlikuje od belog šuma. Naravno, slike moraju biti sveže skenirane; kada ih kopkate sa Interneta i zatim izme- nite neke njihove bitove, to je ravno dobrovoljnoj predaji. Slike nisu i jedini moguć nosilac steganografskih poruka. Zvučne datoteke su za to jednako pogodne. Video datoteke imaju ogroman propusni opseg za steganografske poruke. Informacije se mogu preneti čak i oznakama za formatiranje strane unutar HTML datoteke.

Iako smo steganografiju razmatrali u okviru slobode izražavanja, ona ima i drage primene. Često je koriste vlasnici slika da bi u samu sliku „ugravirali“ svoja vlasnička prava. Ako neko ukrade tako obeleženu sliku i postavi je na Web lokaciju, legalni vlasnik može da otkrije postojanje steganografske poruke na njoj i da je ponudi sudu kao dokaz vlasništva. Ta tehnika je poznata kao **označavanje vodenim žigom** (engl. *watermarking).* Opisali su je Piva i saradnici (2002).

Ko se više zanima za steganografiju, neka potraži radove Artza (2001), Johnsona i Jajoda (1998), Katzenbeissera i Petitcolasa (2000), i Waynera (2002).

8.10.3 Autorska prava

Privatnost i cenzura samo su dve oblasti u kojima se tehnologija sukobljava s društvenim pravilima ponašanja. Treća oblast su autorska prava. **Autorsko pravo** (engl. *copyright)* obezbeđuje da se autorima onih dela koja spadaju u **intelektualnu svojinu** (engl. *Intellectual Property*, *IP)* što znači piscima, slikarima, kompozitorima, muzičarima, fotografima, filmskim stvaraocima, koreografima i drugim autorima, daju isključiva prava korišćenja njihovog dela za određeni vremenski period, najčešće za života autora plus 50 godina ili plus 75 godina u slučaju korporacijskog vlasništva. Pošto isteknu autorska prava na određeno delo, ono prelazi u javno vlasništvo kada ga svako može koristiti i preprodavati po svojoj želji. U okviru projekta Gutenberg (*www.pro- mo.net/pg),* na primer, postavljeno je na Web na hiljade dela u javnom vlasništvu (npr. Sekspirovih, Tvenovih i Dikensovih). Godine 1998, Američki kongres je produžio rok važnosti autorskih prava u SAD za još 20 godina zato što je Holivud ultimativno tvrdio da bez toga niko više ništa neće snimati. S druge strane, patentna prava traju samo 20 godina, ali ljudi se i dalje trude da pronalaze razne stvari.

Problem autorskih prava zaoštrio se u SAD kada je Napster, servis za razmenu muzičkih dela, upisao svog 50-milionitog člana. Iako Napster u stvari uopšte nije kopirao muziku, sud je smatrao da je držanje centralne baze s podacima o tome ko ima koje muzičko delo, saučesništvo koje je dragima omogućilo da krše zakon. Mada niko ozbiljno ne tvrdi da je zaštita autorskih prava loš potez (iako mnogi tvrde da je period njihovog važenja predugačak i da su u tom pogledu velike korporacije favori- zovane), sledeća generacija sistema za razmenjivanje muzičkih dela već se susreće s glavnim etičkim problemima.

Razmotrite, na primer, mrežu ravnopravnih računara u kojoj korisnici dele legalne datoteke (javno dostupnu muziku, kućne video-zapise, religiozne emisije koje ne otkrivaju poslovne tajne itd.) i možda nekoliko datoteka čija su autorska prava zaštićena. Pretpostavimo da su svi sve vreme na mreži, bilo da su povezani ADSL linijama, bilo preko kablovske mreže. Svaki računar ima indeks sadržaja na svom čvrstom disku, kao i listu ostalih članova. Neko ko traži određenu stvar može da nasumice odabere jednog člana da bi video da li je on ima, Ako je ne nađe kod njega, on može da proved sve članove koje nađe na njegovoj listi, zatim članove na njihovim listama itd. Računari vrlo efikasno rade takve stvari. Kada pronađe šta je tražio, korisnik to jednostavno iskopira.

Ako je delo koje je kopirao zaštićeno autorskim pravima, postoje šanse daje prekršio zakon (iako u međunarodnom prenosu datoteka nije jasno da li će se primeniti zakon jedne ili druge države). Staje, međutim, s korisnikom kod koga je našao datoteku? Da lije zločin imati muziku koju ste platili i legalno preuzeli na svoj čvrsti disk gde je mogu naći i drugi? Ako imate vikendicu koju ne zaključavate, pa se kradljivac intelektualne svojine u nju uvuče i pomoću skenera i prenosivog računara iskopira knjigu koja je zaštićena autorskim pravima, da li ste *vi* krivi zbog toga što niste uspeli da zaštitite tuđa autorska prava?

Ali, na frontu zaštite autorskih prava bujaju novi problemi. Trenutno se vodi velika bitka između Holivuda i računarske industrije. Holivud zahteva strogu zaštitu sve intelektualne svojine, a industrija ne želi da bude Holivudski policajac. Oktobra 1998, Kongres je izglasao **Zakon o autorsldm pravima u digitalnom milenij umu** (engl. *Digital Millennium Copyright Act, DMCA)* koji je krivičnim delom proglasio svako zaobilaženje zaštitnog mehanizma autorskog dela ili saopštavanje drugom kako da to učini. Sličan zakon je donet i u okviru Evropske unije. Premda niko ne pravda kopiranje autorskih dela kakvo se masovno događa na Dalekom Istoku, mnogi misle daje DMCA potpuno poremetio ravnotežu između interesa vlasnika autorskih prava i opšteg interesa.

Evo jednog slučaja. Septembra 2000, industrijski konzorcijum zadužen za izgradnju

neprobojnog sistema prodaje muzike putem mreže pozvao je najširu publiku na takmičenje u provaljivanju sistema (što je test kome treba podvrći svaki nov bezbed- nosni sistem). Tim sastavljen od istraživača s više univer ziteta, koji je predvodio prof. Edvard Felten s Prinstona, prihvatio je bačenu rukavicu i provalio u sistem. Zatim su napisali saopštenje o tome kako su to uradili i prijavili ga za konferenciju USENIX posvećenu bezbednosti, gde je ono recenzirano i prihvaćeno. Pre nego što je njihov rad saopšten, Felten je primio pismo od Američkog udruženja izdavača muzičkih me- dijuma (engl. Recording Industry Association of America) s pretnjom da će autori biti tuženi prema zakonu DMCA ukoliko objave rad.

Autori rada su se obratili saveznom sudu za mišljenje da li je objavljivanje naučnog rada proisteklog iz ispitivanja bezbednosnog sistema još uvele legalno. Pribojavajući se da će vrhovni sud presuditi u korist autora, industrija je povukla svoju pretnju i sud je odbacio razmatranje Feltonovog zahteva. Nema sumnje daje industrija tako postupila jer je u stvari sama sebi izvukla tepih ispod nogu: prvo su javno pozvali ljude da provale u njihov sistem, a zatim su počeli da prete onima koji su prihvatili ponuđeni izazov. Pošto je pretnja povučena, rad je objavljen (Carver i sar., 2001). Gotovo je izvesno da će doći do nove konfrontacije.

Bliska opisanom problemu je i **doktrina časnog korišćenja** (engl .*fair use doctrine)* koja je uspostavljena sudskom praksom u mnogim zemljama. Ova doktrina kaže da kupci dela zaštićenih autorskim pravima imaju ograničeno pravo da ih kopiraju, uključujući i pravo da ih citiraju u naučne svrhe, da ih koriste kao didaktički materijal u školama i na fakultetima, a u nekim slučajevima i da prave rezervne kopije za ličnu

upotrebu kako bi se obezbedili ako originalni medijum zakaže. Kada se proverava časnost korišćenja, ispituje se (1) da li se delo koristi komercijalno, (2) koji procenat dela je kopiran i (3) kakav je efekat kopiranja na prodaju dela. Pošto DMCA i odgovarajući propisi u Evropskoj uniji zabranjuju zaobilaženje sistema za zaštitu od kopiranja, oni zabranjuju i legalno časno korišćenje dela. U stvari, DMCA oduzima korisnicima istorijslci stečena prava da bi ih uvećao prodavcima sadržaja. Neizbežan je trenutak kada će svako morati da pokaže karte.

Jednu drugu inovaciju koja baca u zasenak čak i DMCA razvili su Intel i Microsoft. To je **alijansa za pouzdanu računarsku platformu** (engl. *Tmsted Computing Plat- formAlliance, TCPA*). Ideja je u osnovi da se izmisle procesorski čip i operativni sistem koji će pažljivo motriti korisnikovo ponašanje (npr. da li sluša piratizovanu muziku) i takvo ponašanje sprečavati. Sistem čak treba da omogući vlasnicima sadržaja da daljinski manipulišu PC računarima korisnika, menjajući pravila kada je to potrebno. Čini se da bi društvene posledice takve šeme bile nezamislive. Lepo je kada industrija konačno obrati pažn ju na bezbednost, ali je tužno da se to događa samo u oblasti zaštite autorskih prava, a zanemaruju se virusi, uljezi, razbijači i sve ostalo što tišti većinu korisnika.

Jednom reči, u godinama koje su pred nama, oni koji kroje zakone i oni koji ih sprovode imaće mnogo muke da uravnoteže ekonomske interese vlasnika autorskih prava s javnim interesima. Virtuelni svet se ne razlikuje od realnog sveta po neprestanom sukobljavanju suprotstavljenih grupacija, što prvo rezultuje u odmeravanju snaga, zatim u sudskom procesu i (možda) konačno u nekoj vrsti rezolucije koja traje barem dok na scenu ne stupi neka nova tehnologija.

8.11 SAŽETAK

Kriptografija je alatka kojom se može očuvati tajnost, integritet i autentičnost in- formacija. Svi savremeni kriptografski sistemi zasnivaju se na Kerkofovom principu da se koristi javno poznat algoritam i tajni ključ. U mnogim kriptografskim algoritmima osnovni tekst se pretvara u šifrovan tekst složenim transformisanjem koje obuhvata supstituisanje i permutovanje. Međutim, ako kvantna kriptografija teži da bude prihvaćena u praksi, onda samo jednokratna zaštita može da obezbedi stvarno neprobojne sisteme šifrovanja.

Kriptografski algoritmi se mogu podeliti u algoritme za šifrovan je simetričnim ključem i one za šifrovanje javnim ključem. Algoritmi sa simetričnim ključem pretvaraju osnovni tekst u šifrovan tako što mešaju bitove u nizu rundi na način koji je određen ključem. Trostruki DES i Rijndael (AES) zasad su najpoznatiji algoritmi za šifrovanje simetričnim ključem. Oni se mogu koristiti za šifrovanje uz elektronsku knjigu šifara, za ulančavanje blok-šifara, za uzastopno šifrovanje, u brojačkom režimu šifrovanja i na druge načine.

Algoritmi za šifrovanje javnim ključem odlikuju se time što se za šifrovanje i dešifrovanje koriste različiti ključevi i što se ključ za dešifrovanje ne može izvesti iz ključa za šifrovanje. Ta svojstva omogućavaju da ključ bude javan. Glavni algoritam za šifrovanje javnim ključem je RSA, koji svoju snagu zasniva na teškoći razlaganja velikih brojeva na činioce.

Pravni, trgovački i drugi dokumenti moraju biti potpisani. Zbog toga su smišljene različite šeme digitalnog potpisivanja u kojima se koriste algoritmi kako sa simetričnim, tako i s javnim ključem. Umesto da se potpisuju originalne poruke, one se obično prvo heširaju

algoritmom MD5 ili SHA-1, a zatim se heš potpisuje.

Za javne ključeve se mogu koristiti sertifikati - dokumenti koji povezuju principa- la s javnim ključem. Sertifikate potpisuje pouzdana ovlaščena organizacija ili neko ko je (rekurzivno) ovlaščen od nje. Vrh lanca poverenja mora se znati unapred, ali se čitači Weba danas obično isporučuju s više ugrađenih sertifikata najvišeg stepena.

Sledeće kriptografske alatke mogu se koristiti za obezbeđivanje saobraćaja na mreži. IPsec radi u mrežnom sloju, šifrujući tokove paketa od jednog do drugog računara. Zaštitne barijere prosejavaju saobraćaj koji ulazi u organizaciju ili izlazi iz nje, često na osnovu korišćenog protokola i priključka. Virtuelne privatne mreže mogu da simuliraju klasične mreže zasnovane na iznajmljenim linijama i tako ponude neka poželjna be- zbednosna svojstva. Na kraju, bežičnim mrežama je potrebno dobro obezbeđenje, a WEP u mreži 802.11 to ne nudi; mreža 802.1 li treba da znatno popravi stanje.

Kada dve strane uspostave sesiju, one moraju međusobno da provere identitete i, ako treba, da uspostave deljeni ključ sesije. Za proveru identiteta dve strane postoje različiti protokoli, uključujući i takve koji koriste treću stranu, zatim Difi-Helmanova razmena ključa, Kerberos i kriptografija javnim ključem.

Bezbednost pri razmeni e-pošte može se postići kombinovanjem tehnika koje smo razmatrali u ovom poglavlju. Sistem PGP, na primer, komprimuje poruke, a zatim ih šifruje algoritmom IDEA. On šalje IDEA ključ šifrovan primaočevim javnim ključem. On dodatno hešira poruku i šalje potpisan heš kao potvrdu njenog integriteta.

I bezbednost Weba je važna tema, počev od bezbednog imenovanja. DNSsec pruža način za sprečavanje DNS lažiranja, a isto nude i imena sa sopstvenim sertifikatom. Većina Web lokacija za e-trgovinu koristi SSL za uspostavljanje bezbedne, proverene sesije između klijenta i servera. Za pokretni kod se koriste različite tehnike, naročito smeštanje u kutiju s peskom i potpisivanje koda.

Internet pokreće mnoga pitanja kod kojih se tehnika žestoko sukobljava s pravilima javnog ponašanja. Neka od njih su privatnost, sloboda izražavanja i zaštita autorskih prava.

ZADACI

1. Otkrijte šifru kojom je osnovni tekst šifrovan prostom zamenom slova slovom. Osnovni tekst (na engleskom) sastoji se od samih slova i predstavlja opštepoznati odlomak iz poeme Luisa Kerola.

   kfd ktbd fzm eubd kfd pzyiom mztx ku kzyg ur bzha kfthcm ur
   mftnm zhx mfudm zhx mdzythc pzq ur ezsszcdm zhx gthcm zhx
   pfa kfd mdz tm sutythc fulc zhx pfdkfdi ntcm fzld pthcm sok pztk
   z stk kfd uamkdim eitdx sdruid pd fzld uoi efzk rui mubd ur om
   zid uok ur sidzkf zhx zyy ur om zid rzk hu foiia mztx kfd
   ezindhkdi kfda kfzhgdx ftb boef rui kfzk

2. Provalite sledecu šifru za transponovanje po kolonama. Osnovni tekst (na engleskom) preuzet je iz popularnog udžbenika o računarima, tako da se u njemu vero- vatno pojavljuje reč „Computer“. Osnovni tekst sadrži samo slova (bez razmaka). Sifrovan tekst je razbijen u blokove od po pet znakova radi bolje preglednosti, aauan cvlre rurnn dltme aeepb ytust iceat npmey iicgo gorch srsoc
   nntii imiha oofpa gsivt tpsit lbolr otoex

3. Pronađite 77-bitnu jednokratnu zaštitu koja generiše tekst „Donald Duck“ (Paja Patak)

od šifrovanog teksta na slici 8-4.

4. Za kvantnu kriptografiju je neophodan izvor fotona koji će na zahtev emitovati samo jedan foton koji nosi 1 bit informacija. U ovoj vežbi izračunajte koliko bitova mogu fotoni da prenesu kroz optičku vezu brzine 100 Gb/s. Pretpostavite da je dužina fotona jednaka njegovoj talasnoj dužini, koja je u ovom slučaju 1 mikrometar. Brzina svetlosti u optičkom vlaknu iznosi 20 cm/ns.
5. Ako Trudi lovi i regeneriše fotone dok se koristi kvantna kriptografija, ona ce neke dobiti pogrešno, pa će se pojaviti greške u Bobovoj jednokratnoj zaštiti. Koji ce deo bitova Bobove jednokratne zaštite u prošeku biti pogrešan?
6. Osnovni princip kriptografije nalaže da sve poruke moraju imati izvesan višak podataka (redundansu). Međutim, znamo i to da višak podataka pomaže uljezu da utvrdi da li je pretpostavljeni ključ ispravan. Razmotrite dve vrste redundanse. Kod prve, početnih *n* bitova osnovnog teksta sadrže poznatu sekvencu. Kod druge, poslednjih *n* bitova poruke sadrže heš cele poruke. Da li su sa stanovišta bezbednosti ove dve redundanse ekvivalentne? Obrazložite odgovor.
7. Na slici 8-6 smenjuju se P-kutije i S-kutije. Iako takav raspored lepše izgleda, da li je on s gledišta bezbednosti povoljniji od situacije u kojoj prvo idu sve P-kutije, a za njima sve S-kutije?
8. Smislite napad na DES znajući da u osnovnom tekstu učestvuju samo velika ASCII slova, razmak, zarez, tačka, tačka i zarez, znak za povratak na početak reda i znak za prelazak u sledeći red. Ništa se ne zna o bitovima parnosti osnovnog teksta.
9. U tekstu smo izračunali da bi računani za razbijanje šifara koji ima milijardu paralelnih procesora i može da analizira šifru u 1 pikosekundi, bilo potrebno samo IO^{10} godina da razbije 128-bitnu verziju algoritma AES. Međutim, savremeni računali imaju i 1024 paralelna procesora i treba im 1 ms za analiziranje šifre, što čini napredak od 10^{15} puta u smislu ostvarivanja računara za razbijanje algoritma AES. Ako Murov zakon (udvajanje snage računara svakih 18 meseci) nastavi da važi, koliko ćemo još čekati na izgradnju takvog računara?
10. AES podržava 256-bitni ključ. Koliko ključeva ima AES-256? Potražite u fizici, herniji ili astronomiji broj približno te veličine. U traženju velikih brojeva pomozite se i Internetom. Izvucite zaključke iz pretraživanja.
11. Pretpostavite da je poruka šifrovana algoritmom DES ulančavanjem blok-šifara. Jedan bit šifrovanog teksta u bloku C_i slučajno je tokom prenosa promenjen iz 0 u 1. Koliko osnovnog teksta će ta greška pokvariti?
12. Sada ponovo razmotrite ulančavanje blok-šifara. Umesto zamene nule jedinicom, sada je u tok šifrovanog teksta posle bloka Q greškom unet dodatni bit 0. Koliko će sad osnovnog teksta biti pokvareno?
13. Uporedite ulančavanje blok-šifara i šifrovanje s povratnom spregom u pogledu broja operacija šifrovanja neophodnih za prenos velike datoteke. Koji je režim efikasniji i koliko?
14. Neka je dat kriptosistem s javnim RSA ključem, pri čemu je $a = 1,6 = 2$ itd.
    a) Ako je $p = 7$ i $q = 11$, navedite pet legalnih vrednosti d.
    b) Ako je $p = 13$, $q = 31$, $d = 7$, nađite e.
    c) Uz p - 5, $q = 11$ i $d = 27$, nađite e i šifrujte „abcdefghij“.
15. Pretpostavimo da Marija otkrije daje njen privatni RSA ključ *{dl, ni)* isti kao i javni RSA ključ (<?2, *>i2)* drugog korisnika, Franje. Drugim recima, *cl* 1 = e2 i ni = *n2*. Da li Marija treba da menja svoj javni i svoj privatni ključ? Obrazložite odgovor.
16. Razmotrite upotrebu brojačkog režima sa slike 8-15, ali *uz IV* = 0. Da li korišćenje nule u načelu ugrožava šifru?
17. Protokol za potpisivanje sa slike 8-18 ima sledeću slabu tačku: ako Bobov računar

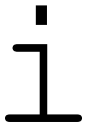
padne, izgubiće se sadržaj RAM memorije. Kakve probleme to može da izazove i šta Bob može da uradi da ih spreči?

18. Na slici 8-20 vidimo kako Alisa može da Bobu pošalje potpisanu poruku. Ako Trudi zameni *P,* Bob će to otkriti. Ali šta se dešava ako Trudi zameni i *P* i potpis?
19. Digitalno potpisivanje ima potencijalnu slabu tačku - lenjost korisnika. U elektronskim trgovačkim transakcijama može se napraviti ugovor i poslati korisniku s molbom da potpiše njegov SHA-1 heš. Ako korisnik ne proveri da li se ugovor i heš slažu, može mu se desiti da potpiše neki drugi ugovor. Pretpostavimo da Mafija želi da zloupotrebi ovu slabu tačku i tako se domogne novca. Oni formiraju naplatnu Web lokaciju (npr. pornografsku, kockarsku itd.) i od korisnika zahtevaju broj kreditne kartice. Zatim mu šalju ugovor u kome stoji da korisnik želi da koristi njihove usluge koje plaća kreditnom karticom (bla, bla...) i traže od njega da ga potpiše, znajući da će većina to uraditi ne proveravajući da li se ugovor i heš slažu. Pokažite kako Mafija može da kupi dijamante od legitimnog Internet prodavca i da ih zaračuna korisnicima koji tako nešto i ne sanjaju.
20. Času matematike prisustvuje 20 studenata. Kolika je verovatnoća da barem dvoje slave rođendan istog dana? Pretpostavite da niko nije rođen 29. februara prestupne godine, tako da ukupno ima 365 mogućih rođendana.
21. Postoje Elen priznala Merilin svoje podvale u okviru Tomovog postavljenja, Meri- lin je resila da takve probleme ubuduće izbegne tako što će pisma snimati na diktafon, odakle će ih u računar prekucavati njena nova sekretarica. Zatim će poruke pregledati na terminalu kako bi bila sigurna da sadrže tačno ono što je rekla. Da li nova sekretarica još uvek može da preduzme rođendanski napad i falsifikuje poruke, a ako može, kako? *Pomoć;* Može.
22. Razmotrite bezuspešan Alisin pokušaj da se domogne Bobovog javnog ključa sa slike 8-23. Pretpostavite da Bob i Alisa već dele tajni ključ, ali Alisa i dalje želi Bobov javni ključ. Postoji li način da ga dobije bezbedno? Ako postoji, navedite ga.
23. Alisa želi da komunicira s Bobom šifrujući poruke javnim ključem. Ona uspostavlja vezu s nekim za koga se nada daje Bob. Ona od njega traži njegov javni ključ i on joj ga šalje u obliku običnog teksta, zajedno sa sertifikatom X.509 koji je potpisala vrhovna CA organizacija. Alisa već ima javni ključ te organizacije. Koje korake treba da preduzme Alisa kako bi proverila da li razgovara s pravim Bobom? Pretpostavite da Bobu nije važno s kim razgovara (tj, Bob je neka vrsta javne službe).
24. Pretpostavimo da se u sistemu koristi PKI zasnovan na hijerarhijskoj strukturi CA organizacija. Alisa želi da komunicira s Bobom i od njega dobija sertifikat koji je potpisala CA organizacija *X* po uspostavljanju komunikacionog kanala s Bobom. Pretpostavite da Alisa nikada nije čula za organizaciju *X.* Koje korake treba da preduzme Alisa da bi se uverila da razgovara s pravim Bobom?
25. Može li se IPsec iskoristiti uz AH u transportnom režimu ako je jedan od računara iza NAT kutije? Obrazložite odgovor.
26. Navedite jednu prednost algoritma HMAC nad algoritmom RSA za potpisivanje SHA-1 heševa.
27. Navedite jedan razlog zbog koga zaštitna barijera treba da ispituje dolazni saobraćaj. Navedite i jedan razlog zašto bi ona trebalo da ispituje odlazni saobraćaj. Mislite li da su te provere uspešne?
28. Format WEP paketa je prikazan na slici 8-.31. Pretpostavimo da je kontrolni zbir dugačak 32 bita jer je izračunat isključivom disjunkcijom svih 32-bitnih reči korisnog tereta. Pretpostavite takođe da su problemi sa algoritmom RC4 izbegnuti tako stoje on zamenjen neprekidnim ključem koji nema slabih tačaka i da su inicijaliza- cioni vektori (IV) proširenі na 128 bitova. Može li sada uljez da špijunira ili da ometa prenos podataka a da ne bude otkriven?

29. Pretpostavimo da određena organizacija koristi VPN da bi se bezbedno povezivala s lokacijama na Internetu. Da li korisnik iz ove organizcije (Džim) treba da koristi šifrovanje ili neki drugi bezbednosni mehanizam da bi razgovarao s drugim korisnikom iz iste organizacije (Meri)?
30. Izmenite neznatno jednu poruku u protokolu sa slike 8-34 tako da postane otporna na napad odbijanjem. Objasnite zašto će uvedena izmena raditi.
31. Difi-Helmanova razmena se koristi za uspostavljanje tajnog ključa između Alise i Boba. Alisa šalje Bobu (719, 3, 191). Bob odgovara sa (.543). Alisin tajni broj *x* je 16. Kako glasi tajni ključ?
32. Ako se Alisa i Bob nisu nikada sreli, ne dele nikakve tajne i nemaju sertifikate, oni ipak mogu da uspostave deljeni tajni ključ pomoću Difi-Helmanovog algoritma. Objasnite zastoje tako teško odbraniti se od posredničkog napada.
33. Zašto se - u protokolu sa slike 8-32 - A šalje u obliku osnovnog teksta, zajedno sa šifrovanim ključem sesije?
34. U protokolu na slici 8-39 istakli smo daje s bezbednosnog stanovišta opasno početi poruku pisanu osnovnim tekstom sa 32 bita 0. Pretpostavite da svaka poruka počinje slučajnim brojem koji bira korisnik - u stvari drugim tajnim ključem koji znaju samo korisnik i KDC. Da li to uklanja opasnost od poznatog napada na osnovni tekst? Zašto?
35. U Nidem-Šrederovom protokolu Alisa generiše dve pozivne poruke: R_A i R_{A2}. To izgleda kao preterivanje. Zar ne bi samo jedna bila dovoljna?
36. Pretpostavimo da određena organizacija za proveru identiteta koristi Kerberos. Šta se događa s gledišta bezbednosti i raspoloživosti usluge ako se server za proveru identiteta (AS) ili server za dodelu kupona (TGS) isključe?
37. U protokolu za proveru identiteta javnim ključem (slika 8-43), R_B se šifruje ključem K_s. Da li je ovo šifrovanje neophodno ili bi odgovor mogao da bude u obliku osnovnog teksta? Obrazložite odgovor.
38. Terminali na prodajnim rnestima gde se koriste magnetne kartice i PIN kodovi imaju fatalan propust: vešt trgovac može da podesi čitač kartica tako da lovi i skladišti sve informacije s kartice, zajedno s PIN kodovima, da bi kasnije mogao da organi- zuje lažne narudžbine. U sledećoj generaciji terminala na prodajnim rnestima, kori- stice se kartice sa sopstvenim procesorom, tastaturom i malim „monitorom“. Smislite protokol za takav sistem koji zlonarnerni trgovac ne može da pobedi.
39. Navedite dva razloga zbog čega PGP komprimuje poruke.
40. Pretpostavljajući da svi na Internetu koriste PGP, odgovorite da li PGP poruku posla- tu na proizvoljnu Internet adresu može da dešifruje svako? Obrazložite odgovor.
41. U napadu prikazanom na slici 8-47 nedostaje jedan korak. Lažiranje će i bez njega raditi, ali će uz njega napad biti prikriveniji. Koji je taj korak?
42. Da bi se osujetilo DNS lažiranje, predloženo je da server ne bira identifikatore redom, već nasumično. Opišite bezbednosne aspekte takvog pristupa.
43. Protokol za prenos podataka SSL obuhvata dva jednokratna uzorka i osnovni ključ. Da li korišćenje jednokratnih uzoraka ima smisla, a ako ima, kakvog?
44. Slika 8-55(b), između ostalog, sadrži ASCII tekst pet Šekspirovih komada. Da li bi umesto teksta u ovu sliku bilo moguće sakriti muziku? Ako je moguće, kako bi to radilo i koliko muzike bi stalo u sliku? Obrazložite odgovor i ako mislite suprotno.
45. Alisa intenzivno koristi anonimni server za prosleđivanje tipa 1. Ona objavljuje mnoge poruke u svojoj omiljenoj diskusionoj grupi *cilt.fanclub.alice* i svi znaju da ih je poslala Alisa jer sve poruke nose isti pseudonim. Ako pretpostavimo da server za prosleđivanje radi ispravno, Trudi ne može da se prikaže kao Alisa. Nakon što su svi serveri tipa 1 ugašeni, Alisa se prebacila na server za prosleđivanje pošte uz šifro- vanje i pokrenula nov lanac poruka u svojoj diskusionoj grupi. Smislite način pomoću koga će Alisa moći

da spreči Trudi da se u diskusionoj grupi predstavlja njenim imenom.

46. Potražite na Internetu neki zanimljiv sudski slučaj o privatnosti i napišite o tome izveštaj na jednoj strani.
47. Potražite na Internetu neki sudski slučaj koji se tiče dvojstva: autorska prava - časno korišćenje i napišite o tome izveštaj na jednoj strani.
48. Napišite program koji ulazne podatke šiffuje tako što ih podvrgava isključivoj disjunkciji s neprekidnim ključem. Pronađite ili napravite što bolji generator slučajnih brojeva da biste generisali neprekidni ključ. Program treba da dejstvuje kao filtar, preuzimajući osnovni tekst na standardnom ulazu i generišući šifrovan tekst na standardnom izlazu (i obrnuto). Program se pokreće uz jedan parametar - ključ koji predstavlja klicu generatora slučajnih brojeva.
49. Napišite potprogram za izračunavanje SHA-1 heša bloka podataka. Potprogram treba da se pokreće uz dva parametra: pokazivač na ulazni bafer i pokazivač na 20-bajtni izlazni bafer. Tačnu specifikaciju algoritma SHA-1 potražite na Internetu koristeći ključnu frazu „FIPS 180-1 “. To je potpuna specifikacija.



DODAJMO ŠTIVO I KORIŠĆEMA LITERATURA

Završili smo s proučavanjem računarskih mreža, ali je to tek početak. Mnoge zanimljive teme nisu obrađene onoliko koliko zaslužuju, a druge su sasvim izostavljene zbog nedostatka prostora. U ovom poglavlju dacemo vam neke predloge za dalje čitanje, kao i spisak literature koju smo koristili - sve namenjeno onim čitaocima koji žele da prodube svoje znanje o računarskim mrežama.

9.1 PREDLOŽI ZA DALJE ČITANJE

Postoji obimna literatura o svim aspektima računarskih mreža. Tri časopisa u kojima se često objavljuju radovi iz ove oblasti su: *IEEE Transactions on Communications*, *IEEE*

Journal on Selected Areas in Communications i *Computer Communication Review.* I mnogi drugi časopisi povremeno objavljuju radove iz ove oblasti.

IEEE objavljuje još tri časopisa: *IEEE Internet Computing, IEEE Network Magazine* i *IEEE Communications Magazine,* koji sadrže pregledne radove, uputstva i realne primere rada na mreži. U prva dva je naglasak na arhitekturi, standardima i softveru, a trec'i naginje komunikacionim tehnologijama (optičkim vlaknima, satelitima itd.).

Osim toga, svake godine ili svake dve godine održava se i niz skupova koji privlače brojna saopštenja o mrežama i distribuiranim sistemima, a naročito su poseceni *Godišnja konferencija SIGCOMM, Međunarodna konferencija o distribuiranim računarskim sistemima* i *Simpozijum o principima operativnih sistema.*

U nastavku nabrajamo neke predloge za dodatno čitanje, sređene prema poglavljima ove knjige. To su većinom priručnici i radovi koji daju pregled aktuelnog stanja u pojedinim oblastima. Nekoliko predloga su poglavlja iz udžbenika.

9.1.1 Uvod i opšte teme

Bi i saradnici, „Wireless Mobile Communications at the Start of the 21st Century“ Novo stolece, nova tehnologija. Zvuči dobro. Posle nešto istorije o bežičnim komunikacijama, ovde su obrađene sve glavne teme, uključujući standarde, aplikacije, Internet i tehnologije.

Comer, The Inernet Book

Svako ko traži pristupačan uvod u Internet, treba da pogleda ovu knjigu. Comer opisuje istoriju, razvoj, tehnologiju, protokole i usluge Interneta na način koji mogu da razumeju početnici, ali je obuhvaćeno toliko materijala da je knjiga interesantna i za iskusnije čitaoce.

IEEE Internet Computing, jan./feb. 2000

U prvom broju časopisa *IEEE Internet Computing* u poslednjoj godini prethodnog milenijuma, učinjeno je ono što ste i očekivali: osobama koje su pomogle razvoj Interneta u prethodnom milenijumu postavljeno je pitanje šta će se s njim dogoditi u sledećem. Za časopis govore Paul Baran, Lawrence Roberts, Leonard Kleinrock, Stephen Crocker, Danny Cohen, Bob Metcalfe, Bill Gates, Bill Joz i drugi stručnjaci. Najbolje će biti da sačekate jedno 500 godina, a *zatim* proverite njihova predviđanja,

Kipnis, „Beating the System: Abuses of the Standards Adoption Process“

Komiteti koji odlučuju o standardima pokušavaju da zadrže neutralnost u svom poslu, ali nažalost postoje i kompanije koje žele da zloupotrebe sistem. Na primer, stalno se iznova događa da neka kompanija pomogne da se napravi standard, a zatim, kada se on usvoji, odjednom objavi da se on zasniva na njenom patentu i da će ga li- cencirati drugim kompanijama po svojoj želji i po ceni koju će sama odrediti. Da biste ušli u mutne vode standardizacije, počnite od ovog članka.

Kyas i Crawford, *ATM Networks*

ATM je jednom trebalo da postane mrežni protokol budućnosti, a još uvek je važan za telefonski sistem. Ova knjiga je savremeni vodič kroz sadašnje stanje ATM mreža, s detaljnim informacijama o ATM protokolima i načinima njihovog integrisanja sa IP mrežama.

Kwok, „A Vision for Residential Broadband Service“

Ako želite da saznate šta je Microsoft 1995. mislio o isporuci videa na zahtev, pročitajte ovaj članak. Tu viziju je nakon pet godina potpuno pregazilo vrerne. Članak je koristan jer pokazuje da čak i dobro potkovani i visokomotivisani ljudi nisu u stanju da predvide

budućnost s nekakvom preciznoišću čak ni za narednih pet godina. Svi treba da iz toga izvučemo pouku.

Naughton, A Brief History of the Future

Ko je u stvari izmislio Internet? Zasluge se pripisuju mnogima, i s pravom, jer su u njegovom stvaranju učestvovali mnogi ljudi, svako na svoj način. Ova istorija Interneta opisuje sva ta događanja na duhovit i šarmantan način, i prepuna je anegdota poput one kada korporacija AT&T više puta uzastopno izjavljuje da digitalne komunikacije nemaju budućnost.

Perkins, „Mobile Networking in the Internet“

Odličan pregled protokola za mrežni rad pokretnih korisnika, sloj po sloj. Obrađeni su slojevi od fizičkog do transportnog, kao i posrednički softver, bezbednost i ad hoc rad u mreži.

Teger i Waks, „End-User Perspectives on Home Networking“

Kućne mreže ne liče na korporacijske mreže. Aplikacije su drugačije (više naginju multimediji), oprema potiče od više proizvođača, a korisnici su s malo iskustva i nimalo strpljenja za eventualne probleme. Pročitajte više o tome u ovom dokumentu.

Varshney i Vetter, „Emerging Mobile and Wireless Networks“

Još jedan uvod u bežične komunikacije. Obuhvata bežične lokalne mreže, bežične lokalne linije i satelite, a delimično pokriva softver i aplikacije.

Wetteroth, OSI Reference Model for Telecommunications

Iako se OSI protokoli više ne koriste, ovaj sedmoslojni model je postao veoma poznat. Osim što detaljnije objašnjava model OSI, ova knjiga opisuje i njegovu pri- menu na telekomunikacione mreže (za razliku od računarskih), ukazujući na tačke u kojima se pri radu u mreži susreću obična telefonija i drugi protokoli za prenos glasa.

9.1.2 Fizički sloj

Abramson, „Internet Access Using VSATs“

Male zemaljske stanice postaju sve popularnije kako za telefoniju izvan gradskog područja, tako i za korporacijski pristup Internetu u razvijenim zemljama. Međutim, saobraćaj se u ova dva slučaja odvija veoma različito, pa su za njih potrebni i različiti protokoli. U ovom članicu autor sistema ALOHA razmatra niz metoda za dodeljivanje kanala koje se mogu iskoristiti u sistemima VSAT.

Alkhatib i saradnici, „Wireles Data Networks: Reaching the Extra Mile“

Ovaj rad je odličan kratak uvod u terminologiju i tehnologiju bežičnog umrežavanja, uključujući i širenje spektra.

Azzam i Ransom, Broadband Access Technologies

Ovde su kao tehnologije pristupanja mreži opisani telefonski sistem, optičko vlakno, ADSL linija, kablovske mreže, sateliti, čak i električna mreža. Od ostalih tema, tu su: kućne mreže, usluge, performanse mreže i standardi. Knjiga se završava biografijama glavnih kompanija važnih za telekomunikacije i umrežavanje, ali s obzirom na brzinu promena u industriji, to poglavlje će verovatno pre zastareti nego poglavlja o tehnologiji.

Bellamy, Digital Telephony

U ovoj autoritativnoj knjizi naći ćete sve što ste ikada želeli da saznate o telefonskom sistemu, a i više od toga. Posebno su zanimljiva poglavlja o prenosu i multiplek- siranju, digitalnom komutiranju, optičkim vlaknima, mobilnoj telefoniji i DSL liniji.

Berezdivin i saradnici, „Next-Generation Wireless Communications Concepts and Technologies“

Autori su jedan korak ispred svega. „Sledeća generacija“ iz naslova odnosi se na četvrtu generaciju bežičnih mreža. Od njih se očekuje da će IP uslugu obezbediti na svakom mestu, kao i potpuno neprimetno povezivanje sa Internetom uz širok propusni opseg i odličan kvalitet usluga. Ti ciljevi bi trebalo da se postignu mudrim korišćenjem spektra, dinamičkim radom s resursima i prilagodljivom uslugom. Sve to trenutno zvuči previše avangardno, ali i mobilni telefoni su 1995. izgledali pusta mašta.

Dutta-Roy, „An Overview of Cable Modem Technology and Market Perspectives“
Kablovska televizija se od jednostavnog sistema CATV razvila u složen sistem za distribuciju TV signala, pristup Internetu i telefoniju. Te promene su znatno uticale na kablovsku infrastrukturu. Članak treba pročitati ako vas zanimaju kablovske mreže, standardi i marketinški aspekti, a naročito sistem DOCSIS.

Farserotu i Prasad, „A Survey of Future Broadband Multimedia Satellite Systems, Issues, and Trends“

Na nebu ili na crtežima nalaze se brojni sateliti za prenos podataka, uključujući sisteme Astrolink, Cyberstar, Spaceway, Skybridge, Teledesic i iSky. U njima se koriste različite tehnike - od savijene cevi do komutiranja. Ovaj rad je dobar početak za svakoga ko želi da sazna više o različitim satelitskim sistemima i tehnikama koje se u njima koriste.

Hu i Li, „Satellite-Based Internet: A Tutorial“

Pristup Internetu preko satelita drugačiji je od pristupa pomoću zemaljskih veza. To nije samo pitanje kašnjenja, već i usmeravanja i komutiranja. U ovom radu autori razmatraju neke probleme korišćenja satelita za pristup Internetu.

Joel, „Telecommunications and the IEEE Communications Society“

Članak je sažeta, ali začuđujuće iscrpna istorija telekomunikacija, počev od telegrafa, pa do mreže 802.11. Tu su i radio, telefoni, analogno i digitalno komutiranje, podmorski kablovi, digitalni prenos, ATM, TV-difuzija, sateliti, kablovska TV, optičke komunikacije, mobilni telefoni, komutiranje paketa, ARPANET i Internet.
Metcalfe, „Computer/Network Interface Design: Lessons from Arpanet & Ethernet“ Iako inženjeri već decenijama prave mrežne interfejse, često se pitamo da li išta uče na iskustvu. U ovom radu autor Etherneta daje uputstva za pravljenje mrežnog in- terfejsa i njegovog korišćenja. On ne beži od sopstvene odgovornosti i tačno navodi šta je uradio dobro, a šta loše.

Palais, Fiber Optic Communications, *3. izdanje*

Knjiga o tehnologiji optičkih vlakana obično je namenjena stručnjacima, ali je ova mnogo pristupačnija. Obuhvata talasovode, svetlosne izvore, detektore, spojnice, modulaciju, šum i mnoge druge teme.

Pandya, „Emerging Mobile and Personal Communications Systems“

Kratak i zgodan uvod u ručne sisteme za ličnu komunikaciju. Jedna od devet strana ovog

članka puna je skraćenica koje se koriste na preostalih osam strana.

Sarikaya, „Packet Mode in Wireless Networks: Overview of Transition to Third Generation“

Glavni zadatak treće generacije mobilnih telefona je da prenose podatke bežičnim putem. Pročitajte ovaj članak da biste razumeli kako to radi draga generacija mobilnih telefona i šta treba izmeniti u trećoj. Među temama su GPRS, IS-95B, WCDMA i CDMA2000.

9.1.3 Sloj veze podataka

Carlson, PPP Design, Implementation and Debugging, *2. izdanje*

Ako vas zanimaju detaljne informacije o svim protokolima koji sačinjavaju skup PPP protokola, uključujući CCP (komprimovanje) i ECP (šifrovanje), pročitajte ovu knjigu. U njoj se posebno govori o ANU PPP-2.3, popularnoj realizaciji sistema PPP.

Gravano, Introduction to Error Control Codes

Greške se provlače kroz skoro sve digitalne komunikacije, a za njihovo otkrivanje i otklanjanje smišljeni su mnogi kodovi. U knjizi su opisani najvažniji takvi kodovi, počev od jednostavnog, linearnog Hamingovog koda, pa do polja Galoa i konvolu- cionih kodova. Autor pokušava da pri objašnjavanju matematiku svede na minimum, ali je i dalje ima dosta.

Holzman, Design and Validation of Computer Protocols

Čitaoci koje zanimaju formalni aspekti protokola sloja veze (i sličnih protokola), treba da pročitaju ovu knjigu u kojoj su objašnjeni specifikacije, modelovanje, ispravnost i proveravanje takvih protokola.

Peterson i Davie, Computer Networks: A Systems Approach

Drago poglavlje sadrži materijal o mnogim pitanjima veze podataka, uključujući uokvirivanje, otkrivanje grešaka, protokole „stani i čekaj“, protokole kliznih prozora i lokalne IEEE 802 mreže.

Stallings, Data and Computer Communications

U sedmom poglavlju se govori o sloju veze podataka, kao i o kontroli toka, otkrivanju grešaka i osnovnim protokolima sloja veze („stani i čekaj“ i „vrati se n“). Opisani su i protokoli tipa HDLC.

9.1.4 Podsloj za upravljanje pristupom medijumima

Bhagwat, „Bluetooth: Technology for Short-Range Wireless Apps“

Dobro početno štivo za direktno uvođenje u sistem Bluetooth. Razmatraju se ključni protokoli, radio, elementarne mreže (pikoneti) i veze, a zatim sledi uvod u razne protokole.

Bisdikian, „An Overview of the Bluetooth Wireless Technology“

Kao prethodni rad Bhagawata, i ovo je dobar uvod u sistem Bluetooth. Između ostalog, u radu se razmatraju pikoneti, skup protokola i profili.

Crow i saradnici, „IEEE 802.11 Wireless Local Area Networks“

Dobar i jednostavan uvod u tehnologiju i protokole mreže 802.11. Naglasak je na podsloju MAC. Opisano je i distribuirano i centralizovano upravljanje. Rad se završava primerima simulacije performansi mreže 802.11 u različitim uslovima.

Eklund i saradnici, „IEEE Standard 802.16: A Technical Overview of the Wireless MAN Air Interface for Broadband Wireless Access“

Bežična lokalna linija koju je IEEE standardizovao 2002. kao mrežu 802.16, može da napravi revoluciju u telefonskom sistemu omogućujući kućnim korisnicima širok propusni opseg. U ovom radu autor opisuje glavna tehnička pitanja koja se odnose na ovaj standard.

Kapp, „802.11: Leaving the Wire Behind“

Kratak uvod u mrežu 802.11; obuhvata osnove, protokole i odgovarajuće standarde.

Kleinrock, „On Some Principles of Nomadic Computing and Multi-Access Communications“

Bežični pristup deljenom kanalu složeniji je od situacije kada kanal deli više kablom povezanih stanica. Ovde se, između ostalog, razmatraju dinamičke topologije, usmeravanje i upravljanje napajanjem. Tu su i druge teme koje se odnose na pristupanje mobilnog bežičnog uređaja kanalu.

Miller i Cummins, LAN Technologies Explained

Želite li da saznate nešto više o tehnologijama koje se koriste u lokalnim mrežama? U ovoj knjizi ćete ih naći sve: FDDI, token ring, kao i popularni Ethernet. Iako se prve dve danas retko instaliraju, koriste se u mnogim postojećim mrežama, a prstenaste mreže su još uvek česte (npr. SONET).

Perlman, *Interconnections,* 2. izdanje

Knjiga je autoritativan, ali zabavan pregled tehnologije mrežnih mostova, usmeri- vača i usmeravanja uopšte. Autor je projektovao algoritme za mostove u razgranatom stablu mreža IEEE 802 i jedan je od vodećih svetskih autoriteta za različite aspekte umrežavanja.

Webb, „Broadband Fixed Wireless Access“

U radu se razmatraju i „zašto“ i „kako“ fiksnih bežičnih mreža širokog opsega. U delu „zašto“ razmatra se zašto ljudi ne žele posebno kućnu adresu e-pošte, adresu e-pošte na poslu, posebne telefonske brojeve za kuću, posao i mobilni telefon, nalog za neposredno slanje poruka i možda jedan ili dva posebna broja za faks. Oni žele jedinstven integrisan sistem koji radi svuda. U delu o tehnologiji naglasak je na fizičkom sloju, gde se razmatraju teme kao što su: TDD u odnosu na FDD, adaptivna modulacija u odnosu na fiksnu, i broj nosilaca podataka.

9.1.5 Mrežni sloj

Bhatti i Crowcroft, „QoS Sensitive Flows: Issues in IP Packet Handling“

Jedan način postizanja višeg kvaliteta usluge u mreži svodi se na pažljivo određivanje trenutka u kom paket napušta svaki usmerivač. U ovom radu su uz izvesne detalje razmotreni različiti algoritmi za vremensko raspoređivanje paketa, kao i druge srodne teme.

Chakrabarti, „QoS Issues in Ad Hoc Wireless Networks“

Kada se prenosivi računari u ad hoc mrežama slučajno nađu jedan do drugog, usmeravanje se sreće s toliko problema da kvalitet ostvarene usluge postaje sekundaran. Bez obzira na to, korisnici zahtevaju kvalitet usluge, pa toj temi treba posvetiti pažnju. U ovom članku se govori o prirodi ad hoc mreža i nekim problemima u vezi sa usmeravanjem i kvalitetom usluge.

Comer, Internetworking with TCP/IP, *Vol. 1, 4. izdanje*

Comer je napisao neprevaziđeno delo o skupu protokola TCP/IP. Poglavlja 4 do 11 bave se protokolom IP i srodnim protokolima mrežnog sloja. Druga poglavlja se uglavnom odnose

na više slojeve i takođe ih treba pročitati.

Huitema, Routing in the Internet

Ako želite da saznate sve što se može znati o usmeravanju na Internetu, ovo je knjiga za vas. Detaljno su obrađeni algoritmi čije ime možete da izgovorite (npr. RIP, CIDR i MBONE), kao i algoritmi kod kojih ćete polomiti jezik (npr. OSPF, IGRP, EGP i BGP). Tu su i nove mogućnosti, kao što su višestruko usmeravanje, protokol za IP komuniciranje s pokretnim korisnicima i rezervisanje resursa.

Malhotra, IP Routing

Knjiga sadrži mnogo detaljnog materijala o IP usmeravanju. Među opisanim protokolima su i RIP, RIP-2, IGRP, EIGRP, OSPG i BGP-4.

Metz, „Differentiated Services“

Za mnoge multimedijske aplikacije neophodno je garantovati kvalitet usluge. To se može postići ili integrisanjem usluga ili njihovim diferenciranjem. Opisane su obe vrste, a naglasak je na diferenciranim uslugama.

Metz, „IP Routers: New Tool for Gigabit Networking“

IJ većini drugih referenci koje se tiču 5. poglavlja govori se o usmeravanju. Ova se razlikuje: ona razmatra stvarni rad usmerivača. Usmerivači su evoluirali od računara opšte namene do vrlo specijalizovanih uređaja. Saznajte sve o tome iz ovog članka.

Nemeth i saradnici, UNIX System Administration Handbook

Za promenu, trinaesto poglavlje ove knjige tretira umrežavanje na mnogo praktičniji način od većine ostalih referenci. Umesto da se bavi opštim temama, ono sadrži puno praktičnih saveta za održavanje stvarne mreže.

Perkins, „Mobile Networking through Mobile IP“

Pošto sve češće viđamo pokretne računarske uređaje, sve je važniji i protokol IP namenjen takvim korisnicima. Ovaj priručnik je dobar uvod u pokretni IP protokol i srodne teme.

Perlman, Interconnections: Bridges and Routers, *2. izdanje*

Perlmanova od 12. do 15. poglavlja opisuje mnoge probleme s kojima se sreću projektanti algoritama za jednosmerno i višesmerno usmeravanje, kako u regionalnim mrežama, tako i u mrežama sastavljenim od više lokalnih mreža, i njihovo praktično rešavanje u različitim protokolima. Najbolje je, međutim, 18. poglavlje, u kome autor- ka na poučan i zabavan način sažima godine svog iskustva s mrežnim protokolima.

Puzmanova, Routing and Switching: Time of Convergence ?

Krajem devedesetih godina, izvesni prodavci mrežne opreme počeli su sve da nazivaju skretnicama, dok su mnogi administratori velikih mreža govorili da se prebacuju sa usmerivača na skretnice. Ova knjiga, u skladu s naslovom, predviđa budući razvoj usmerivača i skretnica i postavlja pitanje da li se oni zaista približavaju jedni dragima?

Royer i Roh, „A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks“

Algoritam AODV za ad hoc usmeravanje o kome smo govorili u 5. poglavlju nije i jedini. U ovom članku se razmatraju i mnogi dragi slični algoritmi, kao što su DSDV, CGSR, WRP, DSR, TORA, ABR, DRP i SRP, i međusobno upoređuju. Jedno je jasno: ako planirate da

napišete nov protokol za ad hoc usmeravanje, najpre treba da smislite pogodnu troslovnu ili četvoroslovnu skraćenicu.

Stevens, TCP/IP Illustrated, *Vol. 1*

U poglavljima 3-10 iscrpno se objašnjava protokol IP i srodni protokoli (ARP, RARP i ICMP), a sve je ilustrovano primerima.

Striegel i Manimaran, „A Survey of QoS Multicasting Issues“

S razvojem usluga kao što su Internet radio i televizija, sve značajniji postaju višesmemo usmeravanje i kvalitet usluge. U ovom preglednom radu autori pokazuju kako algoritmi za usmeravanje mogu da obuhvate istovremeno oba pitanja.

Yang i Reddy, „ATaxonomy for Congestion Control Algorithms in Packet Switching Networks“

Autori su smislili taksonomiju algoritama za kontrolu zagušenja. Glavne kategorije su algoritmi koji *sprečavaju* da zagušenje nastane (1) utičući na izvorište i (2) utičući na odredište, i algoritmi koji raščišćavaju zagušenje *pošto* nastane na osnovu (3) eksplicitnih i (4) implicitnih povratnih informacija. Svoju taksonomiju autori koriste da bi prema njoj razvrstali 23 postojeća algoritma.

9.1.6 Transportni sloj

Comer, Internetworking with TCP/IP, *Vol. 1, 4. izdanje*

Već smo pomenuli da je Corner napisao neprevaziđenu knjigu o skupu protokola TCP/IP. Poglavlje 12 je o protokolu UDP; poglavlje 13 je o protokolu TCP.

Hali i Cerf, Internet Core Protocols: The Definitive Guide

Ako informacije volite da skupljate na samom izvora, ovo je rnesto gde možete da saznate više o protokolu TCP. Kako i ne bi, kada je Cert jedan od njegovih tvoraca. Sedmo poglavlje je dobro štivo o protokolu TCP gde ćete naći kako da protumačite informacije dobijene analizom protokola i alatkama za upravljanje mrežom. Ostala poglavlja govore o protokolima UDP, IGMP, ICMP i ARP.

Kurose i Ross, *Computer Networking: A Top-Down Approach Featuring the Internet* Treće poglavlje govori o transportnom sloju i sadrži dosta materijala o protokolima UDP i TCP. U njemu se govori i o protokolima „stani i čekaj“ i „vrati se n“, koje smo i mi opisali u 3. poglavlju.

Mogul, „IP Network Performance“

Uprkos naslovu, članak više govori o protokolu TCP i mrežnim performansama u načelu, nego konkretno o IP performansama. Prepun je korisnih uputstava i iskustvenih pravila.

Peterson i Davie, Computer Networks: A Systems Approach

Peto poglavlje govori o protokolima UDP, TCP i nekim srodnim protokolima. Ukratko se govori i o mrežnim performansama.

Stevens, TCP/IP Illustrated, *Vol. 1*

U poglavljima 17-24 iscrpno se objašnjava protokol TCP uz brojne primere.

9.1.7 Sloj aplikacija

Bergholz, „Extending Your Markup: An XML Tutorial“

Kratak i neposredan uvod u XML za početnike.

Cardellini i saradnici, *The State-of-the-Art in Locally Distributed Web-Server Systems* S

porastom popularnosti Weba, neke Web lokacije moraju da održavaju velike farme servera da bi uspele da obrade sav saobraćaj. Tu je jedan od glavnih problema raspodeljivanje opterećenja na servere, a on se u ovom radu detaljno razmatra.

Berners-Lee i saradnici, „The World Wide Web“

Tvorac Weba i njegove kolege iz CERN-a govore o perspektivama i budućnosti Weba. Članak razmatra arhitekturu Weba, URL adrese, protokol HTTP i jezik HTML, kao i buduće pravce razvoja Weba, i poredi ga s drugim distribuiranim informacionim sistemima.

Choudbury i saradnici, „Copyright Protection for Electronic Publishing on Computer Networks“

Iako kriptografske algoritme možete naći u brojnim knjigama i člancima, malo ih je koji govore o tome kako se ti algoritmi mogu upotrebiti za sprečavanje korisnika da dalje distribuiraju dokumente koje smeju da dešifruju. U radu se opisuju brojni mehanizmi zaštite autorskih prava u elektronskoj eri.

Collins, „Carrier Grade Voice over IP“

Ako ste pročitali rad Varshneyja i saradnika, i sada želite da znate sve o prenosu glasa Internetom prema specifikaciji H.323, ovo je dobro štivo. Iako je knjiga velika i prepuna detalja, pisana je kao priručnik za koji nije neophodno predznanje iz telefonije.

Davison, „A Web Caching Primer“

S rastom Weba, keširanje postaje sve važniji faktor postizanja dobrih performansi. Ovo je dobar, kratak uvod u Web keširanje.

Krishnamurthy i Rexford, Web Protocols and Practice

Teško je naći iscrpniju knjigu o svim aspektima Weba. Kao što očekujete, ona obuhvata klijente, servere, zastupničke servere i keširanje. Ali, u njoj su i poglavlja o saobraćaju na Webu i njegovom merenju, kao i poglavlja u aktuelnim istraživanjima i poboljšanju Weba.

Rabinovich i Spatscheck, Web Caching and Replication

Pravi izbor ako želite iscrpan tekst o Web keširanju i repliciranju. U knjizi su detaljno obrađeni zastupnički serveri, keš, prethodno skladištenje, mreže za isporuku sadržaja, izbor servera i još mnogo toga.

Shahabi i saradnici, „Yima: A Second-Generation Continuous Media Server“

Serveri multimedije su složeni sistemi koji moraju da upravljaju raspodelom procesorskog vremena, da raspoređuju datoteke na disku, da sinhronizuju tokove, i još mnogo toga. Vremenom su ljudi naučili da ih bolje projektuju. U radu je dat pregled arhitekture jednog modernijeg sistema.

Tittel i saradnici, *Mastering XHTML*

Dvotomna knjiga koja opisuje nov standardni jezik za označavanje Web strana. Najpre ide tekst koji opisuje XHTML, ističući uglavnom njegove razlike u odnosu na HTML. Ostatak je referentni priručnik za oznake, kodove i specijalne znake koji se koriste u jeziku XHTML verzije 1.0.

Varshney i saradnici, „Voice over IP“

Kako funlccioniše prenos glasa Internetom i da li će zameniti javnu komutiranu telefonsku mrežu? Pročitajte i sami izvedite zaključak.

9.1.8 **Bezbednost** na mreži

Anderson, „Why Cryptosystems Fail“

Bezbednost bankarskog sistema je prema Andersonu loša, ali ne zato što inteligentni provalnici pomoću svojih računara razbijaju algoritam DES. Stvarni problemi se kreću od nepoštenih nameštenika (službenik koji klijentovu kućnu adresu zameni sopstvenom da bi dobio njegovu kreditnu karticu i PIN broj), pa do programskih grešaka (davanje istog PIN broja svim klijentima). Posebno je upečatljiv arogantan stav koji banke zauzimaju kada im se predoči problem: „Naš sistem je savršen, a sve greške potiču od klijenata i prevaranata.

Anderson, Security Engineering

U izvesnom smislu, ova knjiga je verzija prethodno navedene, ali na 600 strana. Više je tehnički usmerena od knjige *Secrets and Lies,* a manje od knjige *Network Security* (pogledajte u nastavku). Posle uvoda u osnove bezbednosnih tehnika, čitava poglavlja su posvećena raznim oblastima primene, uključujući bankarstvo, upravljanje i komandovanje nuklearnim postrojenjima, obezbeđivanje štampanog materijala, biometriju, fizičko obezbeđivanje, elektronski rat, bezbednost telekomunikacija, e-trgovinu i zaštitu autorskih prava. Treći deo knjige je o bezbednosnim pravilima, sprovođenju mera i oceni sistema.

Artz, „Digital Steganography“

Steganografija vuče korene iz stare Grčke, gde su tajne poruke na voštanim tablicama pisane na drvenoj podlozi pre provlačenja voskom. Danas se koriste druge tehnike, ali je cilj ostao isti. Ovde su opisane savremene tehnike skrivanja informacija u slike, audio i drage nosioce podataka.

Brands, Rethinking Public Key Infrastructures and Digiatal Certificates

Osim što je opširan uvod u digitalne sertifikate, ova knjiga je i snažan propagandni materijal. Autor je ubeđen da su današnji sistemi provere identiteta koji su zasnovani na papirologiji zastareli i neefikasni, i predlaže da se digitalni sertifikati primene na takve aktivnosti kao što su elektronsko glasanje, digitalno uređivanje ljudskih prava, čak i kao zamena za gotov novac. On takođe upozorava daje Internet bez PKI infrastrukture i šifrovanja samo ogroman sistem za nadziranje građana.

Kaufman i saradnici, *Network Security,* 2. izdanje

Ova autoritativna i pronicljiva knjiga treba da bude prvo što ćete pročitati ako tražite detaljnije tehničke informacije o mrežnim bezbednosnim algoritmima i protokolima. Algoritmi i protokoli s tajnim i javnim ključem, sažeci poruka, provera identiteta, Kerberos, PKI, IPsec, SSL/TLS i bezbednost e-pošte, sve je to pažljivo i detaljno objašnjeno, uz mnoge primere. Poglavlje 26 je o bezbednosnom folkloru i pravi je biser. Kod bezbednosti, đavo se krije u detaljima. Svako ko planira da pro- jektuje bezbednosni sistem koji će stvarno biti primenjen, treba da dobro zapamti sa- vete iz ovog poglavlja koji su izvedeni na osnovu stvarnih situacija.

Pohlman, Firewall Systems

Zaštitne barijere su za mnoge mreže prva (i poslednja) linija odbrane od napadača. Ova knjiga objašnjava kako i šta one rade, počev od najjednostavnijih softverskih barijera namenjenih za obezbeđivanje jednog računara, pa do naprednih bezbednosnih uređaja u vidu samostalnih računara preko kojih se privatna mreža priključuje na Internet.

Schneier, *Applied Cryptography,* 2. izdanje

Ovaj monumentalan, iscrpan priručnik predstavlja noćnu moru agencije NSA: to je knjiga koja opisuje svaki poznat kriptografski algoritam. Još gore je to (ili je još bolje, zavisno na čijoj ste strani) što knjiga sadrži sve algoritme u obliku potpuno funkcionalnih programa napisanih na jeziku C. Tu je i preko 1600 referenci na krip- tografsku literaturu. Ova knjiga nije za početnike, ali ako *stvarno* želite da svoje tajne zadržite za sebe, pročitajte je.

Scheiner, Secrets and Lies

Ako pročitate *Applied Cryptography* od korica do korica, znaćete sve što se može znati o kriptografskim algoritmima. Ako zatim od korica do korica pročitate *Secrets and Lies* (što je mnogo brže), saznaćete da kriptografski algoritmi predstavljaju samo deo priče. Većina bezbednosnih slabosti ne krije se u lošim algoritmima i prekratkim ključevima, već u propustima samog okruženja. Prikazani su brojni primeri pretnji, napada, odbrana, protivnapada itd. Ako želite neformalan i fascinirajući prikaz računarske bezbednosti u najširem smislu, pročitajte ovu knjigu.

Skoudis, Counter Hack

Hakera ćete najlakše osujetiti ako razmišljate kao on. Ova knjiga objašnjava kako hakeri vide mrežu i ističe da bezbednost treba da bude ugrađena u nju tokom pro- jektovanja, a ne naknadno biranjem određene tehnologije. Knjiga opisuje sve uobičajene napade, uključujući i one iz domena lažnog predstavljanja, na šta često „padaju“ osobe nedovoljno upoznate s bezbednosnim merama u računarstvu.

9.2 ABECEDNI SPISAK KORISCENE LITERATURE

ABRAMSON, N.: „Internet Access Using VSATs,“ *IEEE Commun. Magazine,* vol. 38, str. 60-68, jul 2000.

ABRAMSON, N.: „Development of the ALOHANET,“ *IEEE Trans, on Information Theory,* vol. IT-31, str. 119-123, mart 1985.

ADAMS, M. i DULCHINOS, D.: „OpenCable,“ *IEEE Commun. Magazine,* vol. 39, str. 98-105, jun 2001.

ALKHATIB, H.S., BAILEY, C., GERLA, M. i McRAE, J.: „Wireless Data Networks:

Reaching the Extra Mile," *Computer,* vol. 30, str. 59-62, dec. 1997.

ANDERSON, R.J.: „Free Speech Online and Office," *Computer,* vol. 25, str. 28-30, jun 2002.

ANDERSON, **R.J.:** *Security Engineering,* New York: Wiley, 2001.

ANDERSON, R.J.: „The Eternity Service," *Proc. First Int'l Conf. on Theory and Appl. of Cryptology,* CTU Publishing House, 1996.

ANDERSON, R.J.: „Why Cryptosystems Fail," *Commun. of the ACM,* vol. 37, str. 32^10, nov. 1994.

ARTZ, D.: „Digital Steganography," *IEEE Internet Computing,* vol. 5, str. 75-80, 2001.

AZZAM, A.A. i RANSOM, N.: *Broadband Access Technologies,* New York: McGraw- Hill, 1999.

BAKNE, A. i BADRINATH, B.R.: „I-TCP: Indirect TCP for Mobile Hosts," *Proc. 15th Int'l Conf. onDistr. Computer Systems,* IEEE, str. 136-143, 1995.

BALAKRISHNAN, H., SESHAN, S. i KATZ, R.H.: „Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," Proc. ACM *Mobile Computing and Networking Conf,* ACM, str. 2—11, 1995.

BALLARDIE, T., FRANCIS, P. i CROWCROFT, **J.:** „Core Based Trees (CBT)," *Proc. SIGCOMM '93 Conf,* ACM, str. 85-95, 1993.

BARAKAT, C., ALTMAN, E. i DABBOUS, W.: „On TCP Performance in a Heterogeneous Network: A Survey," *IEEE Commun. Magazine,* vol. 38, str. 40^16, jan. 2000.

BELLAMY, J.: *Digital Telephony,* 3. izd., New York: Wiley, 2000.

BELLMAN, R.E.: *Dynamic Programming,* Princeton, NJ: Princeton University Press, 1957.

BELSNES, D.: „Flow Control in the Packet Switching Networks," *Communications Networks,* Uxbridge, England: Online, str. 349-361, 1975.

BENNET, C.H. i BRASSARD, G.: „Quantum Cryptography: Public Key Distribution and Coin Tossing," *Int'l Conf. on Computer Systems and Signal Processing,* str. 175-179, 1984.

BEREZDIVIN, R., BREINIG, R. i TOPP, R.: „Next-Generation Wireless Communication Concepts and Technologies," *IEEE Commun. Magazine,* vol. 40, str. 108-116, mart 2002.

BERGHEL, H.L.: „Cyber Privacy in the New Millennium," *Computer,* vol. 34, str. 132-134, jan. 2001.

BERGHOLZ, A.: „Extending Your Markup: An XML Tutorial," *IEEE Internet Computing,* vol. 4, str. 74-79, jul/avg. 2000.

BERNERS-LEE, T., CAILLIAU, A., LOUTONEN, A., NIELSEN, H.F. i SECRET, A.: „The World Wide Web," *Commun, of the ACM,* vol. 37, str. 76-82, avg. 1994.

BERTSEKAS, D. i GALLAGER, R.: *Data Networks,* 2. izd., Englewood Cliffs, NJ: Prentice Hall, 1992.

BHAGWAT, P.: „Bluetooth: Technology for Short-Range Wireless Apps," *IEEE Internet Computing,* vol. 5, str. 96-103, maj/jun 2001.

BHARGHAVAN, V., DEMERS, A., SHENKER, S. i ZHANG, L.: „MACAW: A Media Access Protocol for Wireless LANs," *Proc. SIGCOMM '94 Conf.*, ACM, str. 212-225, 1994.

BHATTI, S.N. i CROWCROFT, J.: „QoS Sensitive Flows: Issues in IP Packet Handling," *IEEE Internet Computing,* vol. 4, str. 48-57, jul/avg. 2000.

BI, Q., ZYSMAN, G.I. i MENKES, H.: „Wireless Mobile Communications at the Start of the 21st Century," *IEEE Commun. Magazine,* vol. 39, str. 110-116, jan. 2001.

BIHAM, E. **i SHAMIR,** A.: „Differential Cryptanalysis of the Data Encryption Standard," *Proc. 17th Ann. Int'I Cryptology Conf,* Berlin: Springer-Verlag LNCS 1294, str. 513-525, 1997.

BIRD, R., GOPAL, I., HERZBERG, A., JANSON, P.A., KUTTEN, S., MOLVA, R i YUNG, M.: „Systematic Design of a Family of Attack-Resistant Authentication Protocols," *IEEE J. on Selected Areas in Commun.,* vol. 11, str. 679-693, jun 1993.

BIRRELL, A.D. i NELSON, B.J.: „Implementing Remote Procedure Calls," *ACM Trans, on Computer Systems,* vol. 2, str. 39-59, feb. 1984.

BIRYUKOV, A., SHAMIR, A. i WAGNER, D.: „Real Time Cryptanalysis of A5/1 on a PC," *Proc. Seventh lnt'I Workshop on Fast Software Encryption,* Berlin: Springer-Verlag LNCS 1978, str. 1, 2000.

BISDIKIAN, C.: „An Overview of the Bluetooth Wireless Technology," *IEEE Commun. Magazine,* vol. 39, str. 86-94, dec. 2001.

BLAZE, M.: „Protocol Failure in the Escrowed Encryption Standard," *Proc, Second ACM Conf. on Computer and Commun. Security,* ACM, str. 59-67, 1994.

BLAZE, M. i BELLO VIN, S.: „Tapping on My Network Door," *Commun, of the ACM,* vol. 43, str. 136, okt. 2000.

BOGINENI, K., SIVALINGAM, K.M. i DOWD, P.W.: „Low-Complexity Multiple Access Protocols for Wavelength-Division Multiplexed Photonic Networks," *IEEE Journal on Selected Areas in Commun.,* vol. 11, str. 590-604, maj 1993.

BOLCSKEI, H., PAULRAJ, A.J., HAM, K.V.S. i NABAR, R.U.: „Fixed Broadband Wireless Access: State of the Art, Challenges, and Future Directions," *IEEE Qommun. Magazine,* vol. 39, str. 100-108, jan. 2001.

BORISOV, N., GOLDBERG, I. i WAGNER, D.: „Intercepting Mobile Communications: The Insecurity of 802.11*Seventh Int'l Conf. on Mobile Computing and Networking,* ACM, str. 180-188, 2001.

BRANDS, S.: Rethinking Public Key Infrastructures and Digital Certificates, *Cambridge, MA: M.I.T. Press, 2000.*

BRAY, J. i STURMAN, C.F.: *Bluetooth 1.1: Connect without Cables,* 2. izd., Upper Saddle River, N1: Prentice Hall, 2002.

BREYER, R. i RILEY, S.: *Switched, Fast, and Gigabit Ethernet,* Indianapolis, IN: New Riders, 1999.

BROWN, S.: *Implementing Virtual Private Networks,* New York: McGraw-Hill, 1999.

BROWN, L., KWAN, M., **PIEPRZYK,** J. i SEBERRY, J.: „Improving Resistance to Differential Cryptanalysis and the Redesign of LOKI,*"ASIACRYPT '91 Abstracts,* str. 2.5-30, 1991.

BURNETT, S. **i** PAINE, S.: *RSA Security's Official Guide to Cryptography,* Berkeley, CA: Osborne/McGraw-Hill, 2001.

CAPETANAKIS, J.I.: „Tree Algorithms for Packet Broadcast Channels," *IEEE Trans, on Information Theory,* vol. IT-25, str. 505-515, sept. 1979.

CARDELLINI, V., **CASALICCHIO, E., COLAJANNI, M. i** YU, **P.S.:** „The State-of--the-Art in Locally Distributed Web-Server Systems," *ACM Computing Surveys,* vol. 34,

str. 263-311, jun 2002.

CARLSON, J.: *PPP Design, Implementation and Debugging,* 2. izd., Boston: Addison-Wesley, 2001.

CERF, V. i KAHN, R.: „A Protocol for Packet Network Interconnection," *IEEE Trans, on Commun.,* vol. COM-22, str. 637-648, maj 1974.

CHAKRABARTI, S.: „QoS Issues in Ad Hoc Wireless Networks," *IEEE Commun. Magazine,* vol. 39, str. 142-148, feb. 2001.

CHASE, J.S., GALLATIN, A.J. i YOCUM, K.G.: „End System Optimizations for High-Speed TCP," *IEEE Commun. Magazine,* vol. 39, str. 68-75, april 2001.

CHEN, B., JAMIESON, K., BALAKMSHNAN, H. i MORRIS, *R.* „Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *ACM Wireless Networks,* vol. 8, sept. 2002.

CHEN, **K.-C.:** „Medium Access Control of Wireless LANs for Mobile Computing," *IEEE Network Magazine,* vol. 8, str. 50-63, sept./okt. 1994.

CHOUDBURY, A.K., MAXEMCHUK, N.F., PAUL, S. i SCHULZRINNE, H.G.: „Copyright Protection for Electronic Publishing on Computer Networks," *IEEE Network Magazine,* vol. 9, str. 12-20, maj/jun, 1995.

***CHU, Y., RAO, S.G. i ZHANG, H.: „A** Case for End System Multicast,"* Proc. Int'l Conf. on Measurements and Modeling of Computer Syst., *ACM, str. 1-12, 2000.*

CLARK, D.D.: „The Design Philosophy of the DARPA Internet Protocols," *Proc. SIGCOMM '88 Conf,* ACM, str. 106-114, 1988.

CLARK, D.D.: „Window and Acknowledgement Strategy in TCP," RFC 813, jul 1982.

CLARK, D.D., DAVIE, B.S., FÄRBER, D.J., GOPAL, I.S., KADABA, B.K., SINCOSKIE, W.D., SMITH, J.M. i TENNENHOUSE, D.L.: „The Aurora Gigabit Testbed," ***Computer Networks and ISDN Systems,*** **vol. 25, str. 599-621, Jan. 1993.**

CLARK, D.D., JACOBSON, V., ROMKEY, J. i SALWEN, H.: „An Analysis of TCP Processing Overhead," *IEEE Commun. Magazine,* vol. 27, str. 23-29, jun 1989.

CLARK, D.D., LAMBERT, M. i ZHANG, L.: „NETBLT: A High Throughput Transport Protocol," *Proc. SIGCOMM '87 Conf,* ACM, str. 353-359, 1987.

CLARKE, A.C.: „Extra-Terrestrial Relays," *Wireless World,* 1945.

CLARKE, I., MILLER, S.G., HONG, T.W., SANDBERG, O. i WILEY, B.: „Protecting Free Expression Online with Freenet," *IEEE Internet Computing,* vol. 6, str. 40^49, jan./feb. 2002.

COLLINS, D.: *Carrier Grade Voice over IP,* New York: McGraw-Hill, 2001.

COLLINS, D. i SMITH, C.: *3G Wireless Networks,* New York: McGraw-Hill, 2001.

COMER, D.E.: *The Internet Book,* Englewood Cliffs, NJ: Prentice Hall, 1995.

COMER, D.E.: *Internetworking with TCP/IP,* vol. 1,4. izd., Englewood Cliffs, NJ: Prentice Hall, 2000.

***COSTA, L.H.M.K., FDIDA, S. i DUARTE, O.C.M.B.:** „Hop by Hop Multicast Routing Protocol,"* Proc. 2001 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Commun., *ACM, str. 249-259, 2001.*

CRAVER, S.A., WIJ, M., LIU, B., STUBBLEFIELD, A., SWARTZLANDER, B., WALLACH, D.W., DEAN, D. i FELTEN, E.W.: „Reading Between the Lines: Lessons from the SDMI Challenge," ***Proc. 10th USENIX Security Symp.,*** **USENIX, 2001.**

CRESPO, P.M., HONIG, M.L. i SALEHI, J.A.: „Spread-Time Code-Division Multiple

Access," *IEEE Trans, on Commun.,* vol. 43, str. 2139-2148, jun 1995.

CROW, B.P., WIDJAJA, I, KIM, J.G. i SAKAI, P.T.: „IEEE 802.11 Wireless Local Area Networks," *IEEE Commun. Magazine,* vol. 3.5, str. 116-126, sept. 1997.

CROWCROFT, J., WANG, Z., SMITH, A. i ADAMS, J.: „A Rough Comparison of the IETF and ATM Service Models," *IEEE Network Magazine,* vol. 9, str. 12-16, nov./dec. 1995.

DABEK, F., BRUNSKILL, E., KAASHOEK, M.F., KARGER, D., MORRIS, R., STOICA, R. i BALAKRISHNAN, H.: „Building Peer-to-Peer Systems With Chord, a Distributed Lookup Service," *Proc. 8th Workshop on Hot Topics in Operating Systems,* IEEE, str. 71-76, 2001a.

DABEK, F., **KAASHOEK,** M.F., KARGER, D., MORRIS, R. i **STOICA,** I.: „Wide- Area Cooperative Storage with CFS," *Proc. 18th Symp. on Operating Systems Prin.,* ACM, str. 202-215, 2001b.

DAEMEN, J. i **RIJMEN, V.:** *The Design ofRijndael,* Berlin: Springer-Verlag, 2002.

DANTHINE, A.A.S.: „Protocol Representation with Finite-State Models," *IEEE Trans, on Commun.,* vol. COM-28, str. 632-643, april 1980.

DAVIDSON, J. **i** PETERS, **J.:** *Voice over IP Fundamentals,* Indianapolis, IN: Cisco Press, 2000.

DAVIE, B. **i REKHTER,** *MPLS Technology and Applications,* San Francisco: Morgan Kaufmann, 2000.

DAVIS, P.T. i **McGUFFIN, C.R.:** *Wireless Local Area Networks,* New York: McGraw- -Hill, 1995.

DAVISON, B.D.; „A Web Caching Primer," *IEEE Internet Computing,* vol. 5, str. 38—45, jul-avg. 2001.

DAY, J.D.: „The (Un)Revised OSI Reference Model," *Computer Commun. Rev.,* vol. 25, str. 39-55, okt. 1995.

DAY, J.D. **i** ZIMMERMANN, H.: „The OSI Reference Model," *Proc. of the IEEE,* vol. 71, str. 1334-1340, dec. 1983.

DE VRIENDT, J., LAINE, P., **LEROUGE,** C **i XU, X.:** „Mobile Network Evolution: A Revolution on the Move," *IEEE Commun. Magazine,* vol. 40, str. 104—111, april 2002.

DEERING, S.E.: „SIP: Simple Internet Protocol," *IEEE Network Magazine,* vol. 7, str. 16-28, maj/jun 1993.

DEMERS, A., KESHAV, S. i SHENKER, S.: „Analysis and Simulation of a Fair Queueing Algorithm," *Internetwork: Research and Experience,* vol. 1, str. 3-26, sept. 1990.

DENNING, D.E. i SACCO, **G.M.:** „Timestamps in Key Distribution Protocols," *Commun. of the ACM,* vol. 24, str. 533-536, avg. 1981.

DIFFIE, W. i HELLMAN, M.E.: „Exhaustive Cryptanalysis of the NBS Data Encryption Standard," *Computer,* vol. 10, str. 74-84, jun 1977,

DIFFIE, W. i HELLMAN, M.E.: „New Directions in Cryptography," *IEEE Trans, on Information Theory,* vol. IT-22, str. 644-654, nov. 1976.

DIJKSTRA, E.W.: „A Note on Two Problems in Connexion with Graphs," *Numer. Math.,* vol. 1, str. 269-271, okt. 1959.

DOBROWSKI, G. i GRISE, D.: *ATM and SONET Basics,* Fuquay-Varina, NC: APDG Telecom Books, 2001.

DONALDSON, G. i JONES, D.: „Cable Television Broadband Network Architectures," *IEEE Commun. Magazine,* vol. 39, str. 122-126, jun 2001.

DORFMAN, R.: „Detection of Defective Members of a Large Population*," Annals Math. Statistics,* vol. 14, str. 436^140, 1943.

DOUFEXI, A., ARMOUR, S., BUTLER, M., NIX, A., BULL, D., McGEEHAN, J. i KARLSSON, P.: „A Comparison of the HIPERLAN/2 and IEEE 802.11 A Wireless LAN Standards," IEEE *Commun. Magazine,* vol. 40, str. 172-180, maj 2002.

DURAND, A.: „Deploying IPv6," *IEEE Internet Computing,* vol. 5, str. 79-81, jan./feb. 2001.

DUTCHER, B.: *The NAT Handbook,* New York: Wiley, 2001.

DUTTA-ROY, A.: „An Overview of Cable Modem Technology and Market Perspectives," *IEEE Commun. Magazine,* vol. 39, str. 81-88, jun 2001.

EASTTOM, C.: *Learn JavaScript,* Ashburton, U.K.: Wordware Publishing, 2001.

EL GAMAL, T.: „A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Trans, on Information Theory,* vol. IT-31, str. 469-472, jul 1985.

ELHANANY, I., KAHANE, M. i SADOT, D.: „Packet Scheduling in Next-Generation Multiterabit Networks," *Computer,* vol. 34, str. 104-106, april 2001.

ELMIRGHANI, J.M.H. i MOUFTAH, H.T.: „Technologies and Architectures for Scalable Dynamic Dense WDM Networks," *IEEE Commun. Magazine,* vol. 38, str. 58-66, feb. 2000.

FARSEROTU, J. i PRASAD, R.: „A Survey of Future Broadband Multimedia Satellite Systems, Issues, and Trends," *IEEE Commun. Magazine,* vol. 38, str. 128-13.3, jun 2000.

FIORINI, D., CHIANI, M., TRALLI, V. i SALATI., C.: „Can we Trust HDLC?," *Computer Commun. Rev.,* vol. 24, str. 61-80, okt. 1994.

FLOYD, S. i JACOBSON, Y.: „Random Early Detection for Congestion Avoidance," *IEEE/ACM Trans, on Networking,* vol. 1, str. 397-413, avg. 1993.

FLUHRER, S., MANTIN, I. i SHAMIR, A.: *„Weakness in the Key Scheduling Algorithm of RC4,"* Proc. Eighth Ann. Workshop on Selected Areas in Cryptography, *2001.*

FORD, L.R., Jr. i FULKERSON, D.R.: *Flows in Networks,* Princeton, NJ: Princeton University Press, 1962.

FORD, W. i BAUM, M.S.: *Secure Electronic Commerce,* Upper Saddle River, NJ: Prentice Hall, 2000.

FORMAN, G.H. i ZAHORJAN, J.: „The Challenges of Mobile Computing," *Computer,* vol. 27, str. 38-47, april 1994.

FRANCIS, P.: „A Near-Term Architecture for Deploying Pip," *IEEE Network Magazine,* vol. 7, str. .30-37, maj/jun 1993.

FRASER, A.G.: „Towards a Universal Data Transport System," in *Advances in Local Area Networks,* Kummerle, K., Tobagi, F., and Limb, J.O. (Red.), New York: IEEE Press, 1987.

FRENGLE, N.: *1-Mode: A Primer,* New York: Hungry Minds, 2002.

GADECKI, C. i HECKERT, €.: *ATM for Dummies,* New York: Hungry Minds, 1997.

GARBER, L.: „Will 3G Really Be the Next Big Wireless Technology?," *Computer,* vol. .35, str. 26-32, jan. 2002.

GARFINKEL, S. **i** SPAFFORD, G.: *Web Security, Privacy, and Commerce,* Sebastopol, CA: O'Reilly, 2002.

GEIER, J.: *Wireless LANs,* 2. izd., Indianapolis, IN: Sams, 2002.

GEVROS, P., CROWCROFT, **J., KIRSTEIN,** P. **i BHATTI, S.:** „Congestion Control Mechanisms and the Best Effort Service Model," *IEEE Network Magazine,* vol. 15, str. 16-25, maj/jun 2001.

GHANI, N. i DIXIT, S.: „TCP/IP Enhancements for Satellite Networks," *IEEE Commun. Magazine,* vol. 37, str. 64-72, 1999.

*GINSBURG, **D.:*** ATM: Solutions for Enterprise Networking, *Boston: Addison-Wesley,* 1996.

GOODMAN, D.J.: „The Wireless Internet: Promises and Challenges," *Computer,* vol. 33, str. 36-41, jul 2000.

GORALSKI, W.J.: *Optical Networking and WDM,* New York: McGraw-Hill, 2001.

GORALSKI, W.J.: *SONET,* 2. izd., New York: McGraw-Hill, 2000.

GORALSKI, W.J.: Introduction to ATM Networking, *New York: McGraw-Hill, 1995.*

GOSSAIN, H., DE **MORAIS CORDEIRO i AGRAWAL, D.P.:** „Multicast: Wired to Wireless," IEEE *Commun. Mag.,* vol. 40, str. 116-123, jun 2002.

GRAVANO, S.: *Introduction to Error Control Codes,* Oxford, U.K.: Oxford University Press, 2001.

GUO, Y. i CHASKAR, **H.:** „Class-Based Quality of Service over Air Interfaces in 4G Mobile Networks," *IEEE Commun. Magazine,* vol. 40, str. 132-137, mart 2002.

HAARTSEN, J.: „The Bluetooth Radio System," *IEEE Personal Commun. Magazine,* vol. 7, str. 28-36, feb. 2000.

HAC, A.: „Wireless and Cellular Architecture and Services," *IEEE Commun. Magazine,* vol. 33, str. 98-104, nov. 1995.

HAC, A. i **GUO, L.:** „A Scalable Mobile Host Protocol for the Internet," *hit'I J. of Network Mgmt,* vol. 10, str. 115-134, maj/jun, 2000.

***HALL, E. i CERF,** Y.:* Internet Core Protocols: The Definitive Guide, *Sebastopol, CA: O'Reilly, 2000.*

HAMMING, R.W.: „Error Detecting and Error Correcting Codes," *Bell System Tech. J.,* vol. 29, str. 147-160, april 1950.

HANEGAN, K.: Custom CGI Scripting with Perl, *New **York:** Wiley, **2001.***

HARRIS, A.: JavaScript Programming for the Absolute Beginner, *Premier Press, 2001.*

HARTE, L., KELLOGG, S., DREHER, R. i SCHAFFNIT, T.. *The Comprehensive Guide to Wireless Technology,* Fuquay-Varina, NC: APDG Publishing, 2000. HARTE, L., LEVINE, R. i KIKTA, R.: *3G Wireless Demystified,* New York: McGraw- -Hill, 2002.

HAWLEY, G.T.: „Historical Perspectives on the U.S. Telephone System," *IEEE Commun, Magazine,* vol. 29, str. 24-28, mart 1991.

HECHT, J.: „Understanding Fiber Optics," Upper Saddle River, N.I: Prentice Hall, 2001.

HEEGARD, C., COFFEY, J.T., GUMMADI, S., MURPHY, P.A., PROVENCIO, R., ROSSIN, E.J., SCHRUM, S. i SHOEMAKER, M.B.: „High-Performance Wireless Ethernet," *IEEE Commun. Magazine,* vol. 39, str. 64-73, nov. 2001.

HELD, G.: *The Complete Modem Reference,* 2. izd., New York: Wiley, 1994.

HELLMAN, M.E.: „A Cryptanalytic Time-Memory Tradeoff," *IEEE Trans, on Information Theory,* vol. IT-26, str. 401—406, jul 1980.

HILLS, A.: „Large-Scale Wireless LAN Design," *IEEE Commun. Magazine,* vol. 39, str. 98-

104, nov. 2001.

HOLZMANN, G.J.: *Design and Validation of Computer Protocols,* Englewood Cliffs, NJ: Prentice Hall, 1991.

HU, Y. i LI, **V.O.K.:** „Satellite-Based Internet Access," *IEEE Commun. Magazine,* vol. 39, str. 155-162, mart 2001.

HU, Y.-C. i JOHNSON, D.B.: „Implicit Source Routes for On-Demand Ad Hoc Network Routing," *Proc. ACM Int'l Symp. on Mobile Acl Hoc Networking & Computing,* ACM, str. 1-10, 2001.

HUANG, V. i ZHUANG, W.: „QoS-Oriented Access Control for 4G Mobile Multimedia CDMA Communications," *IEEE Commun. Magazine,* vol. 40, str. 118-125, mart 2002.

HUBER, J.F., WEILER, D. i BRAND, H.: „UMTS, the Mobile Multimedia Vision for IMT-2000: A Focus on Standardization," *IEEE Commun. Magazine,* vol. 38, str. 129-136, sept. 2000.

HUI, J.: „A Broadband Packet Switch for Multi-rate Services," *Proc. Int'l Conf. on Commun.,* IEEE, str. 782-788, 1987.

HUITEMA, C.: *Routing in the Internet,* Englewood Cliffs, NJ: Prentice Hall, 1995.

HULL, S.: *Content Delivery Networks,* Berkeley, CA: Osborne/McGraw-Hill, 2002.

HUMBLET, P.A., RAMASWAMI, R. i SIVARAJAN, K.N.: „An Efficient Communication Protocol for High-Speed Packet-Switched Multichannel Networks," *Proc. SIGCOMM '92 Conf,* ACM, str. 2-13, 1992.

HUNTER, D.K. i ANDONOVIC, I.: „Approaches to Optical Internet Packet Switching," *IEEE Commun. Magazine,* vol. 38, str. 116-122, sept. 2000.

HUSTON, G.: „TCP in a Wireless World," *IEEE Internet Computing,* vol. *5,* str. 82-84, mart-april, 2001.

***IBE,** O.C.:* Essentials of ATM Networks and Services, *Boston: Addison-Wesley, 1997.*

IRMER, T.: „Shaping Future Telecommunications: The Challenge of Global Standardization," *IEEE Commun. Magazine,* vol. 32, str. 20-28, jan. 1994.

IZZO, P.: *Gigabit Networks,* New York: Wiley, 2000.

JACOBSON, V.: „Congestion Avoidance and Control," *Proc. SIGCOMM '88 Conf,* ACM, str. 314-329, 1988.

JAIN, **R.:** „Congestion Control and Traffic Management in ATM Networks: Recent Advances and a Survey," *Computer Networks and ISDN Systems,* vol. 27, nov. 1995.

*JAIN, **R.:*** FDDI Handbook - High-Speed Networking Using Fiber and Other Media, *Boston: Addison-Wesley, 1994.*

JAIN, R.: „Congestion Control in Computer Networks: Issues and Trends," *IEEE Network Magazine,* vol. 4, str. 24-30, maj/jun 1990.

JAKOBSSON, M. i WETZEL, S.: „Security Weaknesses in Bluetooth," *Topics in Cryptology: CT-RSA 2001,* Berlin: Springer-Verlag LNCS 2020, str. 176-191, 2001.

JOEL, A.: „Telecommunications and the IEEE Communications Society," *IEEE Com- mun. Magazine,* 50th Anniversary Issue, str. 6-14 i 162-167, maj 2002.

JOHANSSON, P., KAZANTZIDIS, M., KAPOOR, R. i GERLA, M.: „Bluetooth: An Enabler for Personal Area Networking," *IEEE Network Magazine,* vol. 15, str. 28-37, sept./okt. 2001.

JOHNSON, D.B.: „Scalable Support for Transparent Mobile Host Internetworking," *Wireless Networks,* vol. 1, str. 311-321, okt. 1995.

JOHNSON, H.W.: *Fast Ethernet - Dawn of a New Network,* Englewood Cliffs, NJ: Prentice Hall, 1996.

JOHNSON, N.F. i JAJODA, S.: „Exploring Steganography: Seeing the Unseen," *Computer,* vol. 31, str. 26-34, feb. 1998.

KAHN, D.: „Cryptology Goes Public," *IEEE Commun. Magazine,* vol. 18, str. 19-28, mart 1980.

KAHN, D.: *The Codebreakers,* 2. izd., New York: Macmillan, 1995.

KAMOUN, F. **i KLEINROCK, L.:** „Stochastic Performance Evaluation of Hierarchical Routing for Large Networks," *Computer Networks,* vol. 3, str. 337-353, nov. 1979.

KAPP, S.: „802.11: Leaving the Wire Behind," *IEEE Internet Computing,* vol. 6, str. 82-85, jan./feb. 2002.

KARN, P.: „MACA - A New Channel Access Protocol for Packet Radio," *ARRL/CRRL Amateur Radio Ninth Computer Networking Conf,* str. 134—140, 1990.

KARTALOPOULOS, S.: Introduction to DWDM Technology: Data in a Rainbow, *New York, NY: IEEE Communications Society, 1999.*

KASERA, S.K., HJALMTYSSON, G., TOWLSEY, D.F. i KUROSE, J.F.: „Scalable Reliable Multicast Using Multiple Multicast Channels," *IEEE/ACM Trans, on Networking,* vol. 8, str. 294-310, 2000.

KATZ, D. i FORD, P.S.: „TUBA: Replacing IP with CLNP," *IEEE Network Magazine,* vol. 7, str. 38-47, maj/jun 1993.

KATZENBEISSER, S. i PETITCOLAS, F.A.P.: Information Hiding Techniques for Steganography and Digital Watermarking, *London, Artech House, 2000.*

KAUFMAN, €., PERLMAN, R. i SPECINER, M.: *Network Security,* 2. izd., Englewood Cliffs, NJ: Prentice Hall, 2002.

KELLERER, W., VOGEL, H.-J. i STEINBERG, K.E.: „A Communication Gateway for Infrastructure-Independent 4G Wireless Access," *IEEE Commun. Magazine*, vol. 40, str. 126-131, mart 2002.

KERCKHOFF, A.: „La Cryptographie Militaire," *J. des Sciences Militaires,* vol. 9, str. 5-38, jan. 1883 i str. 161-191, feb. 1883.

KIM, J.B., SUDA, T. i YOSHIMURA, M.: „International Standardization of B-ISDN," *Computer Networks and ISDN Systems,* vol. 27, str. 5-27, okt. 1994.

KIPNIS, J.: „Beating the System: Abuses of the Standards Adoptions Process," *IEEE Commun. Magazine,* vol. 38, str. 102-105, jul 2000.

KLEINROCK, L.: „On Some Principles of Nomadic Computing and Multi-Access Communications," *IEEE Commun. Magazine,* vol. 38, str. 46-50, jul 2000.

KLEINROCK, L. i TOBAGI, F.: „Random Access Techniques for Data Transmission over Packet-Switched Radio Channels," *Proc. Nat. Computer Conf.,* str. 187-201,1975.

KRISHNAMURTHY, B. i REXFORD, J.: *Web Protocols and Practice,* Boston: Addison-Wesley, 2001. KUMAR, V., KORPI, M. i SENGODAN, S.: *IP Telephony with H.323,* New York: Wiley, 2001. KUROSE, J.F. i ROSS, K.W.: *Computer Networking: A Top-Down Approach Featuring the Internet,* Boston: Addison-Wesley, 2001.

KWOK, T.: „A Vision for Residential Broadband Service: ATM to the Home," *IEEE*

Network Magazine, vol. 9, str. 14-28, sept./okt. 1995.

KYAS, O. i CRAWFORD, G.: *ATM Networks,* Upper Saddle River, NJ: Prentice Hall, 2002.

LAM, C.K.M. i TAN, B.C.Y.: „The Internet Is Changing the Music Industry," *Commun, of the ACM,* vol. 44, str. 62-66, avg. 2001.

LANSFORD, J., STEPHENS, A. i NEVO, R.: „Wi-Fi (802.11b) and Bluetooth: Enabling Coexistence," *IEEE Network Magazine,* vol. 15, str. 20-27, sept./okt. 2001.

LASH, D.A.: *The Web Wizard's Guide to Perl and CGI,* Boston: Addison-Wesley, 2002.
LAUBACH, M.E., FÄRBER, D.J. i DUKES, S.D.: *Delivering Internet Connections over Cable,* New York: Wiley, 2001. LEE, J.S. i MILLER, L.E.: *CDMA Systems Engineering Handbook,* London: Artech House, 1998.

IEEPER, D.G.: „A Long-Term View of Short-Range Wireless," *Computer,* vol. 34, str. 39-44, jun 2001.

LEINER, B.M., COLE, R., POSTEI., J. i MILLS, D.: „The DARPA Internet Protocol Suite," *IEEE Commun. Magazine,* vol. 23, str. 29-34, mart 1985.

LEVINE, D.A. i AKYILDIZ, I.A.: „PROTON: A Media Access Control Protocol for Optical Networks with Star Topology," *1EEE/ACM Trans, on Networking,* vol. .3, str, 158-168, april 1995.

LEVY, S.: „Crypto Rebels," *Wired,* str. 54-61, maj/jun 199.3.

LI, J., BLAKE, C., DE COUTO, D.S.J., LEE, H.I. i MORRIS, R.: „Capacity of Ad Hoc Wireless Networks," *Proc. 7th Int'I Conf. on Mobile Computing and Networking,* ACM, str. 61-69, 2001.

LIN, F., CHU, P. i LIU, M.: „Protocol Verification Using Reachability Analysis: The State Space Explosion Problem and Relief Strategies," *Proc. SIGCOMM '87 Conf.,* ACM, str. 126-135, 1987.

LIN, Y.D., HSU, N.-B. i HWANG, R.H.: „QoS Routing Granularity in MPLS Networks", *IEEE Commun. Magazine,* vol. 40, str. 58-65, jun 2002.

LISTANI, M., ERAMO, V. i SABELLA, R.: „Architectural and Technological Issues for Future Optical Internet Networks," *IEEE Commun. Magazine,* vol. 38, str. 82-92, sept. 2.000.

LIU, C.L. i LAYLAND, J.W.: „Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment," *Journal of the ACM,* vol. 20, str. 46—61, jan. 1973.

LIVINGSTON, D.: *Essential XML for Web Professionals,* Upper Saddle River, NJ: Prentice Hall, 2002.

LOSHIN, P.: *IPv6 Clearly Explained,* San Francisco: Morgan Kaufmann, 1999.

LOUIS, P.J.: *Broadband Crash Course,* New York: McGraw-Hill, 2002.

LU, W.: Broadband Wireless Mobile: 3G and Beyond, *New York: Wiley, 2002.*

MACEDONIA, M.R.: „Distributed File Sharing," *Computer,* vol. 33, str. 99-101,2000.

MADRUGA, E.L. i GARCIA-LUNA-ACEVES, J.J.: „Scalable Multicasting: the Core-Assisted Mesh Protocol," *Mobile Networks and Applications,* vol. 6, str. 151-165, april 2001.

MALHOTRA, R.; *IP Routing,* Sebastopol, CA: O'Reilly, 2002.

MATSUI, M.: „Linear Cryptanalysis Method for DES Cipher," *Advances in Cryptology - Eurocrypt '93 Proceedings,* Berlin: Springer-Verlag LNCS 765, str. 386-397, 1994.

MAUFER, T.A.: *IP Fundamentals,* Upper Saddle River, NJ: Prentice Hall, 1999.

MAZIERES, D. i KAASHOEK, M.F.: „The Design, Implementation, and Operation of an Email Pseudonym Server," *Proc. Fifth Conf. on Computer and Commun. Security,* ACM, str. 27-36, 1998.

MAZIERES, D., KAMINSKY, M., KAASHOEK, M.F. i WITCHEL, E.: „Separating Key Management from File System Security," *Proc. 17th Symp. on Operating Systems Prin.,* ACM, str. 124-139, dec. 1999.

McFEDRIES, P.: *Using JavaScript,* Indianapolis, IN: Que, **2001.**

McKENNEY, P.E. i DOVE, K.F.: „Efficient Demultiplexing of Incoming TCP Packets," *Proc. SIGCOMM '92 Conf, ACM,* str. 269-279, 1992.

MELTZER, K. i **MICHALSKI,** B.: *Writing CGI Applications with Perl,* Boston: Addison-Wesley, 2001.

MENEZES, A.J. **i** VANSTONE, S.A.: „Elliptic Curve Cryptosystems and Their Implementation,'"*Journal of Cryptology,* vol. 6, str. 209-224, 1993.

MERKLE, R.C.: „Fast Software Encryption Functions," *Advances in Cryptology - CRYPTO '90 Proceedings,* Berlin: Springer-Verlag LNCS 473, str. 476-501, 1991.

MERKLÉ, R.C. **i** HELLMAN, M.: „On the Security of Multiple Encryption," *Commun, of the ACM,* vol. 24, str. 465-467, jul 1981.

MERKLE, R.C. **i** HELLMAN, M.: „Hiding and Signatures in Trapdoor Knapsacks," *IEEE Trans, on Information Theory,* vol. IT-24, str. .525-530, sept. 1978.

METCALFE, R.M.: „On Mobile Computing," *Byte,* vol. 20, str. 110, sept. 1995.

METCALFE, R.M.: „Computer/Network Interface Design: Lessons from Arpanet and Ethernet," *IEEE Journal on Selected Areas in Commun.,* vol. 11, str. 173-179, feb. 1993.

METCALFE, R.M. **i** BOGGS, **D.R.:** „Ethernet: Distributed Packet Switching for Local Computer Networks," *Commun, of the ACM,* vol. 19, str. .395-404, jul 1976.

METZ, C: „Interconnecting ISP Networks," *IEEE Internet Computing,* vol. 5, str, 74-80, mart/april 2001.

METZ, C.: „Differentiated Services," *IEEE Multimedia Magazine,* vol. 7, str. 84-90, jul/ sept. 2000.

METZ, C.: „IP Routers: New Tool for Gigabit Networking," *IEEE Internet Computing,* vol. 2, str. 14-18, nov./dec. 1998.

MILLER, B.A. **i BISDIKIAN,** C.: *Bluetooth Revealed,* Upper Saddle River, NJ: Prentice Hall, 2001.

MILLER, **P. i** CUMMINS, M.: *LAN Technologies Explained,* Woburn, MA: Butter- worth-Heinemann, 2000.

MINOLI, D.: *Video Dialtone Technology,* New York: McGraw-Hill, 1995.

MINOLI, D. **i** VITELLA, M.: *ATM & Cell Relay for Corporate Environments,* New York: McGraw-Hill, 1994.

MISHRA, **P.P. i** KANAKIA, H.: „A Hop by Hop Rate-Based Congestion Control Scheme," *Proc. SIGCOMM '92 Conf,* ACM, str. 112-123, 1992.

MISRA, A., DAS, S., DUTTA, A., MCAULEY, A. **i** DAS, S.: „IDMP-Based Fast Han- doffs and Paging in IP-Based 4G Mobile Networks," *IEEE Commun. Magazine*, vol. 40, str.

138-145, mart 2002.

MOGUL, J.C.: „IP Network Performance," in *Internet System Handbook,* Lynch, D.C. i Rose, M.T. (Red.), Boston: Addison-Wesley, str. 575-675, 1993.

MOK, A.K. **i** WARD, S.A.: „Distributed Broadcast Channel Access," *Computer Networks,* vol. 3, str.. 327-335, nov. 1979.

MOY, **J.:** „Multicast Routing Extensions," *Commun, of the ACM,* vol. 37, str. 61-66, avg. 1994.

MULLINS, J.: „Making Unbreakable Code," *IEEE Spectrum,* str. 40^15, maj 2002.

NAGLE, J.: „On Packet Switches with Infinite Storage," *IEEE Trans, on Commun.,* vol. COM-35, str. 435—438, april 1987.

NAGLE, J.: „Congestion Control in TCP/IP Internetworks," *Computer Commun. Rev.,* vol. 14, str. 11-17, okt, 1984.

NARAYANASWAMI, C., KAMIJOH, N., RAGHUNATH, M., INOUE, T., CIPOL- LA, T., SANFORD, J., SCHLIG, E., VENTKITESWARAN, S., GUNIGUNTALA, D., KULKARNI, V. i YAMAZAKI, K.: „IBM's Linux Watch: The Challenge of Miniaturization," *Computer,* vol. 35, str. 33-41, jan. 2002.

NAUGHTON, J.: „A Brief History of the Future," Woodstock, NY: Overlook Press, 2000.

NEEDHAM, R.M. i SCHROEDER, M.D.: „Authentication Revisited," *Operating Systems Rev.,* vol. 21, str. 7, jan. 1987.

NEEDHAM, R.M. i SCHROEDER, M.D.: „Using Encryption for Authentication in Large Networks of Computers," *Commun, of the ACM,* vol. 21, str. 993-999, dec. 1978.

NELAKUDITI, S. i ZHANG, Z.-L.: „A Localized Adaptive Proportioning Approach to QoS Routing," *IEEE Commun. Magazine,* vol. 40, str. 66-71, jun 2002.

NEMETH, E., SNYDER, G., SEEBASS, S. i HEIN, T.R.: *UNIX System Administration Handbook,* 3. izd., Englewood Cliffs, NJ: Prentice Hall, 2000.

NICHOLS, R.K. i LEKKAS, P.C.: *Wireless Security,* New York: McGraw-Hill, 2002.

NIST: „Secure Hash Algorithm," U.S. Government Federal Information Processing Standard 180, 1993.

O'HARA, B. i PETRICK, A.: *802.11 Handbook: A Designer's Companion,* New York: IEEE Press, 1999.

OTWAY, D. i REES, O.: „Efficient and Timely Mutual Authentication," *Operating Systems Rev.,* str. 8-10, jan. 1987.

OVADIA, S.: Broadband Cable TV Access Networks: from Technologies to Applications, *Upper Saddle River, NJ: Prentice Hall, 2001.*

PALAIS, J.C.: *Fiber Optic Commun.,* 3. izd., Englewood Cliffs, NJ: Prentice Hall, 1992.

PAN, D.: „A Tutorial on MPEG/Audio Compression," *IEEE Multimedia Magazine,* vol. 2, str, 60-74, leto 1995.

PANDYA, R.: „Emerging Mobile and Personal Communication Systems," *IEEE Commun. Magazine,* vol. 33, str. 44-52, jun 1995.

PARAMESWARAN, M., SUSARLA, A. i WHINSTON, A.B.: „P2P Networking: An Information-Sharing Alternative," *Computer,* vol. 34, str. 31—38, jul 2001.

PARK, J.S. i SANDHU, R.: „Secure Cookies on the Web," *IEEE Internet Computing,* vol. 4, str. 36-44, jul/avg. 2000.

PARTRIDGE, C., HUGHES, J. i STONE, J.: „Performance of Checksums and CRCs over Real Data," *Proc. SIGCOMM '95 Conf,* ACM, str. 68-76, 1995.

PAXSON, V.: „Growth Trends in Wide-Area TCP Connections," *IEEE Network Magazine,* vol. 8, str. 8-17, jul/avg. 1994.

PAXSON, V. **i FLOYD,** S.: „Wide-Area Traffic: The Failure of Poisson Modeling," *Proc, S1GCOMM '94 Conf.,* ACM, str. 257-268, 1995.

PEPELNJAK, I. i GUICHARD, J.: *MPLS and VPN Architectures,* Indianapolis, IN: Cisco Press, 2001.

PERKINS, C.E.: *RTP: Audio and Video for the Internet,* Boston: Addison-Wesley, 2002.

PERKINS, G.E. **(Red.):** *Ad Hoc Networking,* Boston: Addison-Wesley, 2001.

PERKINS, C.E.: *Mobile IP Design Principles and Practices,* Upper Saddle River, NJ: Prentice Hall, 1998a.

PERKINS, C.E.: „Mobile Networking in the Internet," *Mobile Networks and Applications,* vol. .3, str. 319-334, 1998b.

PERKINS, G.E.: „Mobile Networking through Mobile IP," *IEEE Internet Computing,* vol. 2, str. 58-69, jan./feb. 1998c.

PERKINS, G.E. **i ROYER, E.:** „The Ad Hoc On-Demand Distance-Vector Protocol," in *Ad Hoc Networking,* u redakciji C. Perkinsa, Boston: Addison-Wesley, 2001.

PERKINS, G.E. **i ROYER,** E.: „Ad-hoc On-Demand Distance Vector Routing," *Proc. Second Ann. IEEE Workshop on Mobile Computing Systems and Applications,* IEEE, str. 90-100, 1999.

PERLMAN, R.: *Interconnections,* 2. izd., Boston: Addison-Wesley, 2000.

PERLMAN, R.: Network Layer Protocols with Byzantine Robustness, *Ph.D. thesis, M.I.T., 1988.*

PERLMAN, R. i KAUFMAN, C.: „Key Exchange in IPsec," *IEEE Internet Computing,* vol. 4, str. 50-56, nov./dec. 2000.

PETERSON, L.L. i DAVIE, B.S.: *Computer Networks: A Systems Approach,* San Francisco: Morgan Kaufmann, 2000.

PETERSON, W.W. i BROWN, D.T.: „Cyclic Codes for Error Detection," *Proc. IRE,* vol. 49, str. 228-235, jan. 1961.

PICKHOLTZ, R.L., SCHILLING, D.L. i MILSTEIN, L.B.: „Theory of Spread Spectrum Communication - A Tutorial," *IEEE Trans, on Commutt,* vol. COM-30, str. 855-884, maj 1982.

PIERRE, G., KUZ, I., VAN STEEN, M. i TANENBAUM, A.S.: „Differentiated Strategies for Replicating Web Documents," *Computer Commun.,* vol. 24, str. 2.32-240, feb. 2001.

PIERRE, G., VAN STEEN, M. i TANENBAUM, A.S.: „Dynamically Selecting Optimal Distribution Strategies for Web Documents," *IEEE Trans, on Computers,* vol. 51, str. 637-65l.jun 2002.

PISCITELLO, D.M. i CHAPIN, A.L.: Open Systems Networking: TCP/IP and OSI, *Boston: Addison-Wesley, 1993.*

PITT, D.A.: „Bridging - The Double Standard," *IEEE Network Magazine,* vol. 2, str. 94-95, jan. 1988.

PIVA, A., **BARTOLINI,** F. **i BARNI,** M.: „Managing Copyrights in Open Networks," *IEEE Internet Computing,* vol. 6, str. 18-26, maj-jun 2002.

POHLMANN, N.: *Firewall Systems,* Bonn, Germany: MITP-Verlag, 2001.

PUZMANOVA, **R.:** *Routing and Switching: Time of Convergence?,* London: Addison- - Wesley, 2002.

RABINOVICH, M. **i SPATSCHECK, O.:** *Web Caching and Replication,* Boston: Addison-Wesley, 2002.

RAJU, J. **i GARCIA-LUNA-ACEVES, J.J.:** „Scenario-based Comparison of Source-Tracing and Dynamic Source Routing Protocols for Ad-Hoc Networks," *ACM Computer Communications Review,* vol. 31, oktobar2001.

RAMANATHAN, R. **i** REDI, J.: „A Brief Overview of Ad Hoc Networks: Challenges and Directions," *IEEE Commun. Magazine,* 50th Anniversary Issue, str. 20-22, maj 2002.

RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R. i SHENKER, S.: „A Scalable Content-Addressable Network," *Proc. SIGCOMM '01 Conf,* ACM, str. 161-172,2001.

RIVEST, R.L.: „The MD5 Message-Digest Algorithm," RFC 1320, april 1992.

REVEST, R.L. **i** SHAMIR, A.: „How to Expose an Eavesdropper," *Commun. of the ACM,* vol. 27, str. 393-395, april 1984.

RIVEST, R.L., **SHAMIR, A. i ADLEMAN,** L.: „On a Method for Obtaining Digital Signatures and Public Key Cryptosystems," *Commun. of the ACM,* vol. 21, str. 120-126, feb. 1978.

ROBERTS, L.G.: „Dynamic Allocation of Satellite Capacity through Packet Reservation," *Proc. NCC,* AFIPS, str. 711-716, 1973.

ROBERTS, L.G.: „Extensions of Packet Communication Technology to a Hand Held Personal Terminal," *Proc. Spring Joint Computer Conference,* AFIPS, str. 295-298, 1972.

ROBERTS, L.G.: „Multiple Computer Networks and Intercomputer Communication," *Proc. First Symp. on Operating Systems Ptin.,* ACM, 1967.

ROSE, M.T.: *The Simple Book,* Englewood Cliffs, NJ: Prentice Hall, 1994.

ROSE, M.T.: *The Internet Message,* Englewood Cliffs, NJ: Prentice Hall, 1993.

ROSE, M.T. i **McCLOGHRIE, K.:** *How to Manage Your Network Using SNMP,* Englewood Cliffs, NJ: Prentice Hall, 1995.

ROWSTRON, A. **i** DRUSCHEL, P.: „Storage Management and Caching in PAST, a Large-Scale, Persistent Peer-to-Peer Storage Utility," *Proc. 18th Symp. on Operating Systems Prin., ACM,* str. 188-201, 2001a.

ROWSTRON, A. i DRUSCHEL, P.: „Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Storage Utility," *Proc. 18th Inti Conf. on Distributed Systems Platforms,* ACM/IFIP, 2001b.

ROYER, E.M. i TOH, C.-K.: „A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks," *IEEE Personal Commun. Magazine,* vol. 6, str. 46-55, april 1999.

RUIZ-SANCHEZ, M.A., BIERSACK, E.W. i DABBOUS, W.: „Survey and Taxonomy of IP Address Lookup Algorithms," *IEEE Network Magazine,* vol. 15, str. 8-23, mart/ april 2001.

SAIRAM, K.V.S.S.S.S., GUNASEKARAN, N. i REDDY, S.R.: „Bluetooth in Wireless Communication," *IEEE Commun. Mag.,* vol. 40, str. 90-96, jun 2002.

SALTZER, J.H., REED, D.P. i CLARK, D.D.: „End-to-End Arguments in System Design," *ACM Trans, on Computer Systems,* vol. 2, str. 277-288, nov. 1984.

SANDERSON, D.W. i DOUGHERTY, D.: *Smileys,* Sebastopol, CA: O'Reilly, 1993.

SARI, H., VANHAVERBEKE, F. i MOENECLAEY, M.: „Extending the Capacity of Multiple Access Channels," *IEEE Commun. Magazine,* vol. 38, str. 74-82, Jan. 2000.

SARIKAYA, B.: „Packet Mode in Wireless Networks: Overview of Transition to Third Generation," *IEEE Commun. Magazine,* vol. 38, str. 164-172, sept. 2000,

SCHNEIER, B.: *Secrets and Lies,* New York: Wiley, 2000.

SCHNEIER, B.: *Applied Cryptography, 2.* izd., New York: Wiley, 1996.

SCHNEIER, B.: *E-Mail Security,* New York: Wiley, 1995.

SCHNEIER, B.: „Description of a New Variable-Length Key, 64-Bit Block Cipher [Blowfish]," *Proc. of the Cambridge Security Workshop,* Berlin: Springer-Verlag LNCS 809, str. 191-204, 1994.

SCHNORR, C.P.: „Efficient Signature Generation for Smart Cards," *Journal of Cryptology,* vol. 4, str. 161-174, 1991.

SCHOLTZ, R.A.: „The Origins of Spread-Spectrum Communications," *IEEE Trans, on Commun.,* vol. COM-30, str. 822-854, maj 1982.

SCOTT, R.: „Wide Open Encryption Design Offers Flexible Implementations," *Cryptologia,* vol. 9, str. 75-90, jan. 1985.

SEIFERT, R.: *The Switch Book,* Boston: Addison-Wesley, 2000.

SEIFERT, R.: *Gigabit Ethernet,* Boston: Addison-Wesley, 1998.

SENN, J.A.: „The Emergence of M-Commerce," *Computer,* vol. 33, str. 148-150, dec. 2000.

SERJANTOV, A.: „Anonymizing Censorship Resistant Systems," *Proc. First Int'l Workshop on Peer-to-Peer Systems,* Berlin: Springer-Verlag LNCS, 2002.

SEVERANCE, C.: „IEEE 802.11: Wireless Is Corning Home," *Computer,* vol. 32, str. 126-127, nov. 1999.

SHAHABI,.C., ZIMMERMANN, R., FU, K. i YAO, S.-Y.D.: „YIMA: A Second-Generation Continuous Media Server," *Computer,* vol. 35, str. 56-64, jun 2002.

SHANNON, C.: „A Mathematical Theory of Communication," *Bell System Tech. J.,* vol. 27, str. 379-423, jul 1948; i str. 623-656, okt. 1948.

SHEPARD, S.: *SONET/SDH Demystified,* New York; McGraw-Hill, 2001.

SHREEDHAR, M. **i** VARGHESE, G.; „Efficient Fair Queueing Using Deficit Round Robin," *Proc. SIGCOMM '95 Conf.,* ACM, str. 231-243, 1995.

SKOUDIS, E.: *Counter Hack,* Upper Saddle River, NJ: Prentice Hall, 2002.

SMITH, **D.K. i** ALEXANDER, R.C.: *Fumbling the Future,* New York: William Morrow, 1988.

SMITH, R.W.: *Broadband Internet Connections,* Boston: Addison Wesley, 2002.

SNOEREN, A.C. i BALAKRISHNAN, H.: „An End-to-End Approach to Host Mobility," *Int'I Conf. on Mobile Computing and Networking,* ACM, str. 155-166, 2000.

SOBEL, D.L.: „Will Carnivore Devour Online Privacy," *Computer,* vol. 34, str. 87-88, maj 2001.

SOLOMON, J.D.: *Mobile IP: The Internet Unplugged,* Upper Saddle River, NJ: Prentice Hall, 1998.

SPOHN, M. i GARCIA-LUNA-ACEVES, J.J.: „Neighborhood Aware Source Routing," *Proc. ACMMobiHoc 2001, ACM,* str. 2001.

SPURGEON, C.E.: *Ethernet: The Definitive Guide,* Sebastopol, CA: O'Reilly, 2000.

STALLINGS, W.: *Data and Computer Communications,* 6. izd., Upper Saddle River, NJ: Prentice Hall, 2000.

STEINMETZ, R. i NAHRSTEDT, K.: Multimedia Fundamentals. Vol. 1: Media Coding and Content Processing, *Upper Saddle River, NJ: Prentice Hall, 2002.*

STEINMETZ, R. i NAHRSTEDT, K.. Multimedia Fundamentals. Vol. 2: Media Processing and Communications, *Upper Saddle River, NJ: Prentice Hall, 2003a.*

STEINMETZ, R. i NAHRSTEDT, K.: Multimedia Fundamentals. Vol. 3: Documents, Security, and Applications, *Upper Saddle River, NJ: Prentice Hall, 2003b.*

STEINER, J.G., NEUMAN, B.C. i SCHILLER, J.I.: „Kerberos: An Authentication Service for Open Network Systems," *Proc. Winter USENIX Conf,* USENIX, str. 191-201, 1988.

STEVENS, *W.R.:* UNIX Network Programming, Volume 1: Networking APIs - Sockets and XTI, *Upper Saddle River, NJ: Prentice Hall, 1997.*

STEVENS, W.R.: *TCP/IP Illustrated, Vol. I,* Boston: Addison-Wesley, 1994.

STEWART, R. i METZ, C.: „SCTP: New Transport Protocol for TCP/IP," *IEEE Internet Computing,* vol. 5, str. 64-69, nov./dec. 2001.

STINSON, D.R.: *Cryptography Theory and Practice,* 2. izd., Boca Raton, FL: CRC Press, 2002.

STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, M.F. i BALAKRISHNAN, **H.:** „Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," *Proc. SIGCOMM '01 Conf,* ACM, str. 149-160, 2001.

STRIEGEL, A. i MANIMARAN, G.: „A Survey of QoS Multicasting Issues," *IEEE Commun. Mag,,* vol. 40, str. 82-87, jun 2002.

STUBBLEFIELD, A., IOANNIDIS, J. i RUBIN, A.D.: „Using the Fluhrer, Mantin, and Shamir Attack to Break WEP," *Proc. Network and Distributed Systems Security Symp.,* ISOC, str. 1-11,2002.

SUMMERS, U.K.: ADSL: Standards, Implementation, and Architecture, *Boca Raton, FL: CRC Press, 1999.*

SUNSHINE, C.A. i DALAL, Y.K.: „Connection Management in Transport Protocols," *Computer Networks,* vol. 2, str. 454^473, 1978.

TANENBAUM, A.S.: *Modern Operating Systems,* Upper Saddle River, NJ: Prentice Hall, 2001.

TANENBAUM, A.S. i VAN STEEN, M.i *Distributed Systems: Principles and Paradigms,* Upper Saddle River, NJ: Prentice Hall, 2002.

TEGER, S. i WARS, D.J.: „End-User Perspectives on Home Networking," *IEEE Commun. Magazine,* vol. 40, str. 114-119, april 2002.

THYAGARAJAN, A.S. i DEERING, S.E.: „Hierarchical Distance-Vector Multicast Routing for the MBone," *Proc. SIGCOMM '95 Conf,* ACM, str. 60-66, 1995.

TITTEL, E., VALENTINE, C., BURMEISTER, M. i DYKES, L.: *Mastering XHTML,* Alameda, CA: Sybex, 2001.

TOKORO, M. i TAMARU, K.: „Acknowledging Ethernet," *Compcon,* IEEE, str. 320-325, jesen 1977.

TOMLINSON, R.S.: „Selecting Sequence Numbers," *Proc. SIGCOMM/S1GOPS Inter-process Commun. Workshop,* ACM, str. 11-23, 1975.

TSENG, Y.-C., WU, S.-L., LIAO, W.-H. i CHAO, C.-M.: „Location Awareness in Ad Hoc Wireless Mobile Networks," *Computer,* vol. 34, str. 46-51, 2001.

TUCHMAN, W.: „Heilman Presents No Shortcut Solutions to DES," *IEEE Spectrum,* vol. 16, str. 40-41, jul 1979.

TURNER, J.S .: „New Directions in Communications (or Which Way to the Information Age)," *IEEE Commun. Magazine,* vol. 24, str. 8-15, okt. 1986.

VACCA, J.R.: *I-Mode Crash Course,* New York: McGraw-Hill, 2002.

VALADE, J.: *PUP & MySQL for Dummies,* New York: Hungry Minds, 2002.

VARGHESE, G. i LAUCK, T.: „Hashed and Hierarchical Timing Wheels: Data Structures for the Efficient Implementation of a Timer Facility," *Proc. 11th Symp. on Operating Systems Prin,,* ACM, str. 25-38, 1987.

VARSHNEY, U., SNOW, A., McGIVERN, M. i HOWARD, C.: „Voice over IP," *Commun. of the ACM,* vol. 45, str. 89-96, 2002.

VARSHNEY, U. i VETTER, R.: „Emerging Mobile and Wireless Networks," *Commun. of the ACM,* vol. 43, str. 73-81, jun 2000.

VETTER, P., GODERIS, D., VERPOOTEN, L. **i** GRANGER, A.: „Systems Aspects of APON/VDSL Deployment," *IEEE Commun. Magazine*, vol. 38, str. 66-72, maj 2000.

WADDINGTON, D.G. **i** CHANG, F.i „Realizing the Transition to IPv6," *IEEE Commun. Mag.,* vol. 40, str. 138-148, jun 2002.

WALDMAN, M., RUBIN, A.D. **i CRANOR, L.F.:** „Publius: A Robust, Tamper-Evident, Censorship-Resistant, Web Publishing System," *Proc. Ninth USENIX Security Symp.,* USENIX, str. 59-72, 2000.

WANG, Y. i CHEN, W.: „Supporting IP Multicast for Mobile Hosts," *Mobile Networks and Applications,* vol. 6, str. 57-66, jan./feb. 2001.

WANG, Z.: *Internet QoS,* San Francisco: Morgan Kaufmann, 2001.

WARNEKE, B., LAST, M., LIEBOWITZ, B. **i** PISTER, **K.S.J.:** „Smart Dust: Communicating with a Cubic Millimeter Computer," *Computer,* vol. 34, str. 44—51, jan. 2001.

WAYNER, *P.:* Disappearing Cryptography: Information Hiding, Steganography, and Watermarking, *2. izd., San Francisco: Morgan Kaufmann, 2002.*

WEBB, W.: „Broadband Fixed Wireless Access as a Key Component of the Future Integrated Communications Environment," *IEEE Commun. Magazine,* vol. 39, str. 115-121, sept. 2001.

WEISER, M.: „Whatever Happened to the Next Generation Internet?," *Commun, of the ACM,* vol. 44, str. 61-68, sept. 2001.

WELTMAN, R. **i** DAHBURA, T.: *LDAP Programming with Java,* Boston: Addison- Wesley, 2000.

WESSELS, D.: *Web Caching,* Sebastopol, CA: O'Reilly, 2001.

WETTEROTH, D.: *OSI Reference Model for Telecommunications,* New York: McGraw-Hill, 2001.

WILJAKKA, J.: „Transition to Ipv6 in GPRS and WCDMA Mobile Networks," *IEEE Commun. Magazine,* vol. 40, str. 134-140, april 2002.

WILLIAMSON, H.: *XML: The Complete Reference,* New York: McGraw-Hill, 2001.

WILLINGER, W., TAQQU, M.S., SHERMAN, R. i WILSON, D.V.: „Self-Similarity

through High Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level," *Proc. SIGCOMM '95 Conf,* ACM, str. 100-113, 1995.

WRIGHT, D.J.: Voice over Packet Networks, *New York: Wiley, 2001.*

WYLIE, J., BIGRIGG, M.W., STRUNK, J.D., GANGER, G.R., **KILICCOTE, H. i KHOSLA, P.K.:** „Survivable Information Storage Systems," *Computer,* vol. 33, str. 61-68, avg. 2000.

XYLOMENOS, G., POLYZOS, G.C., MAHONEN, P. i SAARANEN, M.: „TCP Performance Issues over Wireless Links" , *IEEE Commun. Magazine,* vol. 39, str. 52-58, april 2001.

YANG, C.-Q. **i** REDDY, A.V.S.: „A Taxonomy for Congestion Control Algorithms in Packet Switching Networks," *IEEE Network Magazine,* vol. 9, str. 34-4.5, jul/avg. 1995.

YUYAL, G.: „How to Swindle Rabin," *Cryptologia,* vol. 3, str. 187-190, jul 1979.

ZACKS, M.: „Antiterrorist Legislation Expands Electronic Snooping," *IEEE Internet Computing,* vol. 5, str. 8-9, nov./dec. 2001.

ZADEH, A.N., JABBARI, B., PICKHOLTZ, R. i VOJCIC, B.: „Self-Organizing Packet Radio Ad Hoc Networks with Overlay (SOPRANO)," *IEEE Commun. Mag.,* vol. 40, str. 149-157, jun 2002.

ZHANG, L.: „Comparison of Two Bridge Routing Approaches," *IEEE Network Magazine,* vol. 2, str. 44^48, jan./feb. 1988.

ZHANG, L.: „RSVP: A New Resource ReSerVation Protocol," *IEEE Network Magazine,* vol. 7, str. 8-18, sept./okt. 199.3.

ZHANG, Y. **i** RYU, **B.:** „Mobile and Multicast IP Services in PACS: System Architecture, Prototype, and Performance," *Mobile Networks and Applications,* vol. 6, str. 81-94, jan./feb. 2001.

ZIMMERMANN, P.R.: *The Official PGP User's Guide,* Cambridge, MA: M.I.T. Press, 1995a.

ZIMMERMANN, P.R.: *PGP: Source Code and Internals,* Cambridge, MA: M.I.T. Press, 1995b.

ZIPF, G.K.: Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology, *Boston: Addison-Wesley, 1949.*

ZIV, J. i LEMPEL, Z.: „A Universal Algorithm for Sequential Data Compression," *IEEE Trans, on Information Theory,* vol. IT-23, str. 337-34.3, maj 1977.

Spisak termina korišćenih u knjizi

ActiveX
kontrole ad hoc
mreža ad hoc
umrežavanje
adresiranje
agent za domaće računate, domaći
agent agent za prenos poruka agent za strane računate, agent za strance
agent za osmatranje
aktivan repetitor
aktivni susedi
algoritam bušne kofe algoritam kofe sa žetonima
algoritam SHA-1, bezbedni algoritam za heširanje 1
algoritam za šifrovanje simetričnim ključem
algoritam za usmeravanje
algoritam zasnovan na toku podataka
amplitudna modulacija analiza dostupnosti
analogno-digitalni pretvarač aplet
apstraktna sintaksna notacija 1
arhitektura mreže, mrežna arhitektura
ASCII oklop
asimetrična digitalna pretplatnička linija, ADSL linija
asinhroni režim prenosa

ASP strane, aktivne serverske strane
ATM sloj za adaptaciju

audio koji se reprodukuje u realnom vremenu / tokom preuzimanja
audio sloj MPEG 3
automatsko ponavljanje zahteva
autonoman sistem

ActiveX controls ad hoc network ad hoc networking addressing home agent

message transfer agent foreign agent

snooping agent active repeater active neighbors leaky bucket algorithm token bukket algorithm

Secure Hash Algorithm 1, SHA-1

symmetric-key algorithm routing algorithm flow-based algorithm

amplitude modulation reachability analysis Analog to Digital Converter, ADC applet Abstract Syntax Notation 1, ASN.l network architecture

ASCII armor Asymmetric Digital Subscriber Line, ADSL Asynchronous Transfer Mode, ATM Active Server Pages, ASP
ATM Adaptation Layer, AAL streaming audio

MPEG audio layer 3

autonomno MPLS control-driven MPLS
komutiranje autorsko pravo *copyright*
ažurirati *update*
baf'erovanje, privremeno *buffering* skladištenje
barijera, zaštitna barijera *firewall* besklasno međudomensko *Classless InterDomaiti* usmeravanje *Routing, CIDR*
bez gubitaka *lossless*
bezbedni sistem datoteka *Secure File System*
bezbedni sistem MIME *Secure/MIME, S/MIME*
bezbednosno povezivanje *security association, SA*
bezbednost transportnog *Transport Layer*

Automatic Repeat reQuest, ARQ
Autonomous System, AS
sloja
bezkonfliktni protokol, protokol u kome nema sukobljavanja
bežična gradska mreža bežična lokalna linija

bežična lokalna mreža binarno eksponencijalno odustajanje binarno odbrojavanje
bipolarno označavanje, bipolarna notacija
bit parnosti blok-šifra bod
broj veze
broj za potvrđivanje

brojački režim šifrovanja brzi Ethernet brzina usluživanja bujica
fragmenata bujica neusmerenih paketa, neusmerena bujica
bujica okvira CDN mreža, mreža za isporuku sadržaja
centar za distribuiranje ključeva centrala sistema mobilne telefonije
Security, TLS
collision-free protocol

wireless MAN Wireless Local Loop, WLL wireless LAN binary exponential backoff binary countdown
bipolar notation

parity bit
block-cipher
baud
connection number
acknowledgement number counter mode
fast Ethernet service rate
fragment burst broadcast storm

frame bursting Content Delivery Network, CDN Key Distribution Center, KDC Mobile Telephone Switching Office, MTSO; Mobile Switching Center, MSC

Kompletan spisak termina koji se koriste u izdanjima Mikro knjige nalazi se na adresi www.mk.co.yu/recni k.

centralni razvodnile cev, kanal
CGI interfejs, opšti interfejs za mrežni prolaz ciklična provera ređundanse često postavljana pitanja

čisti sistem
ALOHA čitač Weba ćelija daljinski most daljinsko pozivanje procedure daljinsko prijavljivanje datagrafska usluga, usluga datagrama datagrami datagramska podmreža davalac Internet usluga

davalac mrežnih usluga davalac usluga prenosa

DCF razmak između okvira debeli Ethernet (10Base5) decimalna notacija s tačkom deljenje resursa demultipleksiranje deponovanje šifara DES standard, DES šifrovanje, standard za šifrovanje podataka DHCP agent za prenos diferencijalna impulsno- -kodna modulacija diferencijalno Mančester kodiranje diferencirane usluge

difuzno emitovanje, neusmereno emitovanje digitalna pretplatnička linija, DSL linija digraf dijagonalna osnova dinamički HTML direktno

sekvencijalno širenje spektra diskretan višetonski sistem
head end
pipe
Common Gateway Interface, CGI Cyclic Redundancy Check, CRC Frequently Asked Questions, FAQs pure ALOHA Web browser cell remote bridge Remote Procedure Call, RPC remote login datagram service

datagrams datagram subnet Internet Service Provider, ISP network provider transport service provider DCF InterFrame Spacing, DIFS thick Ethernet dotted decimal notation

resource sharing demultiplexing key escrow Data Encryption Standard, DES

DHCP relay agent differential pulse code modulation differential Manchester encoding differentiated services, DS broadcast

Digital Subscriber Line, xDSL digram diagonal basis dynamic HTML Direct Sequence Spread Spectrum, DSSS Discrete MultiTone, DMT
diskretna kosinusna transformacija diskusione grupe distribuirana koordinativna funkcija distribuirani napad radi blokiranja usluga, DDoS napad

Discrete Cosine Transformation, DCT
newsgroups Distributed Coordination Function, DCF Distributed Denial of Service, DDoS

distributed system DNS security, DNSsec DNS spoofing extension headers

distribuirani sistem DNS bezbednost DNS lažiranje dodatna zaglavlja
dogovaranje nivoa usluge *service level agreement* doktrina časnog korišćenja *fair use*
top-level domain

collision domain
pad
gratuitous ARP Denial of Service, DoS

Clear to Send, CTS
Internet header length
Time Division Duplexing, TDD jumbogram
expedited forwarding

e-trgovina
elementarna jedinica
elementarna mreža, pikonet
emotikon
ESP zaglavlje, kapsulirajuće bezbednosno zaglavlje farma diskova farma servera fazna modulacija FDD prenos, dvosmerni prenos podelom frekvencije FDDI interfejs, interfejs za podatke distribuirane optičkim kablom FEC klasa, klasa ekvivalentnog prosleđivanja fiksni bežični sistem fiksni bežični telefon filtar za pakete

doctrine domaći agent, agent za *home agent*

domaće računare
domen najvišeg nivoa, osnovni domen
domen sukobljavanja
dopuna
dopunska usluga
ARP-a DoS napad, napad radi blokiranja
usluga dozvola za slanje dužina Internet zaglavlja dvosmerni prenos podelom vremena
džambogram
ekspresno prosleđivanje
elektronska pošta, e-pošta *e-mail*
elektronska trgovina, *electronic commerce, e-commerce chunk piconet*

emoticon
Encapsulating Security Header, ESP

disk farm server farm
phase modulation
Frequency Division Duplexing, FDD

Fiber Distributed Data Interface, FDDI

Forwarding Equivalence Class, FEC fixed wireless
cordless phone packet filter

fizički medijum *physical medium*
fizički sloj *physical layer*
frekventna modulacija *frequency modulation*
FTTC sistem, optički kabl u susedstvu *Fiber To The Curb, FTTC*
FTTH sistem, optički kabl do kuc'e *Fiber To The Home, FTTH*
Furijeov niz *Fourier series*
garantovano *assured forwarding*
generatorski polinom *generator polynomial*
geostacionarna orbita *Geostationary Earth Orbit, GEO*
gigabitni Ethernet *gigabit Ethernet*
glavne ličnosti, principali *principals*
globalni sistem komunikacija, GSM *Global System for Mobile Communications, GSM*
globalni sistem za cioniranje, sistem *Global Positioning System, GPS*
gradska mreža *Metropolitan Area Network, MAN*
grupa diskova *disk array*
grupna odrednica *aggregate entry*
gurajuc'i server *push server*
gusti WDM *Dense WDM, DWDM*
hardverski kriptoprocesor *clipper chip*
HD televizija, visoke rezolucije *High Definition Television, HDTV*
heširana poruka *hashed message*
hibridni optičko-jalni kabl/sistem *Hybrid Fiber Coax, HFC*
hijerarhijski niz sertifikata *certification path*
hijerarhijsko Web keširanje *hierarchical caching*
hipertekst *hypertext*
hiperveza, veza *hyperlink, link*
hitno slanje podataka *urgent data transfer*
hromatska disperzija *chromatic dispersion*
HTML oznaka *HTML tag*
identifikator čvora *node identifier*
implementacija, realizacija *implementation*
impulsno-kođna modulacija *Pulse Code Modulation, PCM*
Inženjerske snage Interneta *Internet Engineering Task Force, IETF*
indigo kopija, polje *Carbon copy, Cc flag*
indikator
indikatorski bajt *flag byte*
indirektni TCP *indirect TCP*
informacioni režim, i-režim *information mode, i-mode*
infrastruktura za kovanje javnih ključeva *Public Key Infrastructure, PKI*
inicijalizacioni vektor, IV vektor *Initialization Vector, IV*
Institut inženjera elek trotehnike i *Institute of Electrical and Electronic Engineers, IEEE*
integrisane usluge *integrated services*
intelektualna svojina *Intellectual Property, IP*
interfejs *interface*
Interfejs za fiksne pri stupne bežične sisteme *Air Interface for Fixed Broadband Wireless Access Systems*
interfejs za podatke buirane optičkim blom, FDDI *Fiber Distributed Data Interface, FDDI*
Internet društvo *Internet Society*
Internet protokol, protokol IP *Internet Protocol, IP*
intranet, interna mreža *intranet*
IP bezbednost *IP security, IPsec*
ispravljanje grešaka u hodu *forward error correction*
Istraživačke snage Interneta *Internet Research Task Force, IRTF*
izbeljivanje *whitening*
izlazni luk *output arc*
iznajmljivanje *leasing*
iznenadna zagušenja *flash crowds*
izobličen signal *distorted signal*
izveštavanje *reporting*
izvor išni priključak *source port*
Javina virtuelna *Java Virtual Machine, JVM*
javna komutirana telefonska mreža *Public Switched Telephone Network, PSTN*
javne službe *common carriers*
jedinstvena adresa resursa, URL adresa *Uniform Resource Locator, URL*
jedinstveno ime *Universal Resource Name, URN*
jednokanalni sistemi mobilne telefonije *push-to-talk systems*
jednokratna zaštita *one-time pad*
jednokratni uzorci *nonces*
jednorežimsko monomodno vlakno *single-mode fiber*
jednosmerna veza *simplex*
jednosmerno *unicasting*
JSP strane, strane Java servera *JavaServer Pages, JSP*
kabl primopredajnika *transceiver cable*
kabl s više priključaka *multidrop cable*

kadar, okvir	*frame*
kanal za dodelu	*access grant channel*
kanal za objavljivanje	*paging subchannel*
kanal za registrovanje, propuštanje i status	*Registration/Admission/Stalus, RAS*
kanal za slobodan	*random access channel*
kanal za upravljanje neusmerenim/difuzn emitovanjem	*broadcast control channel*
kanal, cev	*pipe*
kapsulirajuće bezbednosno zaglavlje, ESP	*Encapsulating Header, ESP*
kapsuliran	*encapsulated*
lutajući računati	*roaming hosts*
m-trgovina, mobilna vina, pokretna	*m-commerce, mobile commerce*
mašina konačnih	*finite state machine*
magistrala	*bus*
Mančester kodiranje	*Manchester encoding*
markiranje,	*markup*
maska	*skin*
maska podmreže	*subnet mask*
maskiranje	*frequency masking*
matična grupa	*mastergroup*
matična lokacija	*home location*
matična strana, strana	*home page*
medij za kontinualno reprodukovanje	*continuous media*
meko predavanje upravljanja	*soft handoff*
međumesna centrala	*IntereXchange Carrier, IXC*
međumreža, kombinovana mreža	*internet, internetwork*
međumrežni sloj	*internet layer*
Međunarodna cija za	*International Standards Organization, ISO*
Međunarodni savez za telekomunikacije	*International Tele communication Union, ITU*
međunarodni standard	*International Standard, IS*
meritoran zapis	*authoritative record*
migrirajući računari	*migratory hosts*
mikroćelija	*microcell*
mikrojezgro	*microkernel*
mikrojezgro za rad u realnom vremenu	*real-time microkernel*
miniintervali	*minislots*
mobilna ad hoc mreža	*Mobile Ad hoc NETwork, MANET*
mobilni bežični	*mobile wireless*
mobilni telefon	*mobile phone, cell phone*
model guranja	*push*
modul za obradu	*processing module*
modulacija s frekventnim pomeranjem	*frequency shift keying*
monomodno vlakno, jednorežimsko vlakno	*single-mode fiber*
most, mrežni most	*bridge*
MPLS komutiranje, višeprotokolarno komutiranje oznaka	*Multiprotocol Label Switching, MPLS*
MPLS komutiranje vođeno podacima	*data-driven MPLS*
konkurentski LEC	*Competitive LEC, CLEC*
konstelacioni dijagram	*constellation diagram*
kontrola grešaka	*error control*
kontrola pristupa	*admission control*
kontrola toka	*flow control*
kontrola zagušenja	*congestion control*
kontrolni usmerivač	*adjacent router*
kontrolni zbir	*checksum*
konvergencija prenosa	*Transmission Convergence, TC*
Konzorcijum W3C,	*World Wide Web Consortium, W3C*
Konzorcijum za upra vljanje Webom	
korenski server	*root server*
korisnički agent	*user agent*
korisnički profil	*user profile*
korisnički segment	*user plane*
korisnik usluga	*transport service user*
koristan teret, podaci	*payload*
kratak razmak između okvira	*Short InterFrame Spacing, SIFS*
kreditna poruka	*credit message*
kružna blokada	*deadlock*
kubiti (bitovi koji se pojedinačnim fotonima)	*qubits*
kupon	*ticket*
kutija s peskom	*sandbox*
kvadraturna amplitudna modulacija	*Quadrature Amplitude Modulation*
kvadraturna modulacija faznim pomakom	*Quadrature Phase Shift Keying, QPSK*
kvalitet usluge	*Quality of Service, QoS*
kvantizacija	*quantization*
kvantna kriptografija	*quantum cryptography*
labava mreža	*scatternet*
lanac poverenja	*chain of trust*
lančani napad	*bucket brigade attack*
LED dioda, svetleća dioda, svetlosna	*Light Emitting Diode, LED*
lične mreže	*personal area networks*
linije prenosa	*transmission lines*
lista povučenih	*Certificate Revocation List, CRL*
lista slanja	*mailing list*
lokalna centrala	*local central office*
lokalna linija	*local loop*
lokalna mreža	*Local Area Network, LAN*
lokalna telefonska centrala	*end office, Local Exchange Office, LEC*
lokalni TV pretvarač	*set-top box*

mreža Petri	*Petri net*
mreža širokog regionalna mreža	*Wide Area Network, WAN*
mreža za isporuku sadržaja, CDN	*Content Delivery Network, CDN*
mreže od tačke do tačke	*point-to-point networks*
mreže povezane u tački	*stub networks*
mreže povezane u više tačaka	*multiconnected networks*
mreže ravnopravnih računara	*peer-to-peer networks*
mreže sa difuznim emitovanjem	*broadcast networks*
mrežna arhitektura, tektura mreže	*network architecture*
mrežna okosnica, okosnica	*backbone*
mrežna rasprodaja, mrežna aukcija	*on-line auction*
mrežna skretnica, skretnica	*switch*
mrežni interfejs	*Network Interface Device, NID*
mrežni prolaz	*gateway*
mrežni prolaz za aplikacije	*application gateway*
mrežni sloj	*network layer*
MTU jedinica, jedinica prenosa	*Maximum Transmission Unit, MTU*
multimodno vlakno, višerežimsko vlakno	*multimode fiber*
multipleksiranje	*multiplexing*
multipleksiranje podelom	*Frequency Division Multiplexing, FDM*
multipleksiranje talasne dužine	*Wavelength Division Multiplexing, WDM*
multipleksiranje podelom vremena	*Time Division Multiplexing, TDM*
multipleksiranje sa čestom podelom talasnih dužina	*Dense Wavelength Division Multiplexing, DWDM*
multipleksor pristupa digitalnoj koj liniji	*Digital Subscriber Line Access Multiplexer, DSLAM*
Nacionalna agencija bezbednost	*National Security Agency, NSA*
Nacionalni institut za standarde i tehnologiju	*National Institute of Standards and Technology, NIST*
Nacionalni komitet za televizijske standarde	*National Television Standards Committee, NTSC*

nacrt međunarodnog standarda	keystream reuse attack
nacrt standarda	replay attack
nađovezivanje	Denial of Service, DoS
nadzorni najkraća putanja	Advanced Mobile Phone System, AMPS
namenski upravljački kanal	NAT box
namenski usmerivač	parked overhead
napad odbijanjem	nonrepudiation
napad ponavljanjem iste šifre	unnumbered
napad ponovljenim slanjem poruka	Unshielded Twisted Pair, UTP
napad radi blokiranja usluga, DoS	odd keystream
napad	nonadaptive algorithms
napredni sistem mobilne telefonije	jitter
NAT kutija	broadcast storm
neaktivan	broadcast, broadcasting
nekoristan teret, sistemski podaci	transparent bridge
nemogućnost poricanja	Blind carbon copy, Bcc
nenumerisan	spam
neoklopljene upređene parice, UTP	low frequency, LF
kabl	thread
neparan	level
neprekidni ključ	TI carrier
neprilagodljivi algoritmi	Incumbent LEC, ILEC
neravnomernost	security by obscurity
neusmerena bujica, bujica neusmerenih paketa	voice-grade line
neusmereno emitovanje, difuzno emitovanje	snail mail
nevidljiv most	backbone area
nevidljiva kopija, polje Bcc	colored threads
neželjena pošta	Interface Message Processors, IMPs
niska frekvencija	form
nit	reverse lookup
nivo	
nosilac TI	
obavezni LEC	
obezbeđivanje kroz prikrivanje	
obična govorna telefonska linija	
obična zemaljska pošta	
oblast okosnice	
obojene niti	
obrađivači poruka na interfejsu	
obrazac	
obrnuto pretraživanje	
Draft International Standard, DIS	
draft standard	
concatenation	
supervisory	
dedicated control channel	
designated router	
reflection attack	

odbacivanje paketa	*load shedding*
odnos signala i šuma	*signal-to-noise ratio*
odredišni priključak	*destination port*
određivanje rastojanja	*ranging*
oglas	*advertisement*
okosnica, mrežna okosnica	*backbone*
okvir podataka, okvir s podacima	*data frame*
okvir za potvrdu	*acknowledgement frame*
okvir, kadar	*frame*
omotnica poruke	*envelope*
opisi stilova	*style sheets*
opšta paketna radio usluga, usluga	*General Packet Radio Service, GPRS*
opštepoznati	*well-known ports*
opšti ključevi	*passkeys*
opšti upravljački	*common control channel*
optički čvor	*fiber node*
optički kabl do kuće, FTTH sistem	*Fiber To The Home, FTTH*
optički kabl u FTTC sistem	*Fiber To The Curb, FTTC*
optički kanal	*Fibre Channel*
organizacija za sertifikata	*Certification Authority, CA*
osnovna ploča	*backplane*
osnovne operacije	*primitives*
osnovni domen, najvišeg nivoa	*top-level domain*
osnovni ključ	*premaster key*
osnovni protokol zasnovan na bit mapi	*basic bit-map protocol*
osnovni tekst	*plaintext*
oštro predavanje upravljanja	*hard handojf*
otisak	*footprint*
ovlašćivanje	*authorization*
označavanje vodenim žigom	*watermarking*
označavanje, oznaka	*markup label, tag*
p-trajni CSMA	*p-persistent CSMA*
paket	*packet*
paket za podešavanje	*setup packet*
paran	*even*
PCF funkcija, jedinstvena koordinativna	*Point Coordination Function, PCF*
PCF razmak između okvira	*PCF InterFrame Spacing, PIFS*
perceptivno kodiranje	*perceptual coding*
pikonet, elementarna mreža	*piconet*
piksel	*pixel*

plavljenje poboljšani sistem mobilne telefonije, sistem IMTS početna strana, početno stanje podinterval podmreža podmreža s komutiranjem paketa podmreža s virtuelnim *vir* kolima podsloj konvergencije

matična strana

podsloj za upravljanje pristupom medijumima, MAC podsloj pojačanje privatnosti pokazivač na hitne podatke pokretni kod pokretni računan policijski nadzor saobraćaja polinomski kod polje polje Bcc, nevidljiva kopija polje Cc, indigo kopija polje za potvrdu polucrni ispis poludupleks poludupleksni režim pomoćna aplikacija posrednički, zastupnički posrednički napad

pošta s poboljšanom privatnošću poštansko sanduče potkomitet potpisivanje koda potpomognuto preuzimanje upravljanja potpuni dupleks potvrđivanje i ponovno slanje

pouzdano polazište povratne informacije povremeni CSMA protokol prag pravougaona osnova

flooding
Improved Mobile Telephone System, IMTS
home page

initial state
chip
subnet
pakket-switched subnet

Convergence Sublayer, CS
medium access control sublayer, MAC

privacy amplification
urgent pointer

mobile code
mobile hosts
traffic policing

polynomial code field
Blind carbon copy, Bcc

Carbon copy, Cc
checkbox
bold
half duplex half-duplex
mode helper application
proxy
man-in-the-middle attack
Privacy Enhanced Mail, PEM mailbox
subcommittee, SC code signing Mobile Assisted Han- dOff, MAHO full duplex
Positive A cknowledgement with Retransmission, PAR trust anchor
feedback
nonpersistent CSMA

threshold rectilinear basis

predavanje upravljanja prediktivno kodiranje predlog standarda prednacrt standarda predviđanje zaglavlja pregovarati prekidački elementi prekoredni paket

prelaz prenos
prenos datoteka
prenos glasa preko Interneta
prenosilac podataka
preplitanje
preslikavanje (servera)
preslušavanje
prevođenje mrežnih adresa pričaonica
prigušni paket
prijemni prozor
prikazivanje
priključak
priključak na optičku mrežu priključna tačka
prilagodljivi algoritmi prilično dobra privatnost
handoff
predictive encoding
Proposed Standard
Committee Draft, CD
header prediction
negotiate
switching elements out-of-order packet transition
transfer file transfer voice

carrier interlacing
mirroring crosstalk
Network Address Translation, NAT chat
room choke packet
receiving window
displaying port
Optical Network Unit, ONU
Point of Presence, POP
adaptive algorithms Pretty Good Privacy, PGP
transceiver optimality principle

two-army problem
exposed station problem
count-to-infinity problem
hidden station problem

primopredajnik princip optimalnosti principali, glavne ličnosti *principals* pristup uz podelu talasne *Wavelength Division* dužine

pr istupna tačka pritoke privatno povezivanje ravnopravnih uređaja privatnost kao u ožičenoj *Wired Equivalent*

mrezt privremena adresa privremeni kodić privremeno maskiranje problem skladištenje, baferovanje problem vojske problem izložene stanice problem približavanja beskonačnosti problem skrivene stanice procedura LAP, procedura *Link Access Procedure,* za pristupanje vezi*LAP*

Multiple Access, WDMA
access point
tributaries private peering

Privacy, WEP dodatak
address nonpersistent
problem
problem buffering

over IP

produženi razmak između Extended InterFrame *okvira* Spacing, EIFS
programski dodatak *plug-in*
proizvod opsega i kašnjenja *bandwidth-delay product*
prolazna skretnica *cut-through switch*
promiskuitetni režim *promiscuous mode*
propusni opseg *bandwidth*
prosejavanje poziva *call screening*
prosleđivanje *forwarding*
prosleđivanje paketa reverse path forwarding
ispitivanjem izvorišta
proširenje nosioca podataka *carrier extension*
protočno slanje podataka *pipelining*
protokol ARP, protokol za razrešavanje adresa *Address Resolution Protocol, ARP*
protokol BGP, protokol graničnog mrežnog prolaza *Border Gateway Protocol, BGP*
protokol CSMA/CD, protokol CSMA uz CSMA with Collision Detection, CSMA/CD otkrivanje sukoba
protokol DHCP, protokol za dinamičko podeša- *Dynamic Host Configu- ration Protocol,* vanje računara *DHCP*
protokol ESMTP, prošireni *extended SMTP,* protokol SMTP *ESMTP*
protokol FTP, protokol File Transfer Protocol, za prenos datoteka *FTP*
protokol HTTP, protokol za prenos hiperteksta *HyperText Transfer Protocol, HTTP*
protokol HTTPS, Secure HTTP, HTTPS
bezbedni HTTP protokol
ICMP, protokol za upravljanjeporukama na Internetu *Internet Control Message Protocol, ICMP*
protokol IGMP, protokol za rad s grupama na Internetu *Internet Group Management Protocol, IGMP*
protokol IMAP, protokol za pristupanje porukama na Internetu *Internet Message Access Protocol, IMAP*
protokol IP, Internet protokol *Internet Protocol, IP*
protokol IS-IS, protokol za međusistemske veze *Intermediate System-Intermediate System, IS-IS*
protokol ISAKMP, bezbednosni protokol *za rad sa šiframa Internetu* Internet SecurityAssociation and Key Management Protocol, ISAKMP
protokol LCP, protokol za upravljanje vezom *Link Control Protocol, LCP*
protokol LDAP, Light-weight Directory jednostavan protokol za pristup imenicima *Access Protocol, LDAP*
protokol LTP, jednostavan transportni protokol *Lightweight Transport Protocol, LTP*
protokol NCP, protokol za upravljanje mrežom *Network Control Protocol, NCP*
protokol NNTP, protokol za prenos poruka *Network News Transfer Protocol, NNTP*
diskusionih grupa
protokol OSPF, otvoren najkraće putanje *Open Shortest Path First, OSPF*
protokol POP3, poštanski protokol verzije 3 *Post Office Protocol Version 3, POP3*
protokol PPP, protokol od tačke do tačke *Point-to-Point Protocol, PPP*
protokol RARP, obrnuti *ARP protokol* *Reverse Address Resolution* Protocol, RARP
protokol RSVP, protokol za rezervisanje resursa *Resource reSerVation Protocol, RSVP*
protokol RTCP, protokol za upravljanjeprenosom u realnom vremenu *Real-time Transport Control Protocol, RTCP*
protokol RTP, protokol za prenos u realnom vremenu *Real-time Transport Protocol, RTP*
protokol RTSP, protokol za preuzimanje podataka u realnom vremenu *Real Time Streaming Protocol, RTSP*
protokol SCTP, protokol za upravljanje tokom podataka *Stream Control Transmission Protocol, SCTP*
protokol SIP, protokol otvaranje sesije *Session Initiation Protocol, SIP*
protokol SIPP, dopunjeni jednostavan protokol za Internet *Simple Internet Protocol Plus, SIPP*
protokol SMTP, jednostavan protokol za prenos elektronske pošte *Simple Mail Transfer Protocol, SMTP*
protokol SOAP, jednostavan protokol za pristupanje objektima *Simple Object Access Protocol, SOAP*
protokol T/TCP, transakcioni TCP protokol *Transactional TCP, T/TCP*
protokol TCP, protokol za upravljanje prenosom *Transmission Control Protocol, TCP*
protokol tipa čuvaj i prosledi *store-and-forward protocol*
protokol tipa stani i čekaj *stop-and-wait*

protocol protokol u vlasništvu *proprietary protocol*
protokol WAP, za bežične aplikacije *Wireless Application Protocol, WAP*
protokol WDP, za bežične *Wireless Datagram Protocol, WDP*
protokol za korisničke datagrame *User Datagram Protocol, UDP*
protokol za početno povezivanje *initial connection protocol*
protokol za podizanje sistema *BOOTstraP, BOOTP*
protokol za osluškivanje na nosiocu podataka *carrier sense protocol*
protokol za usmeravanje na vektoru *Distance Vector Multi cast Routing Proto col, DVMRP*
protokoli s rezervisanjem vremena emitovanja *reservation protocols*
protokoli sa ograničenom konkurencijom *limited-contention protocols*
protokoli za proveru identiteta *challenge-response protocols*
provera identiteta *authentication*
prozor za slanje *sending window*
prozor zagušenja *congestion window*
računar na Internetu *Internet host*
računar povezan protokolom *protocol machine*
računarske mreže *computer networks*
rad sa žetonima *token management*
radio-dugmad *radio buttons*
rafalne greške *burst errors*
RAID diskovi, redundantne grupe jeftinih diskova *Redundant Array of Inexpensive Disks, RAID*
rano otkrivanje zagušenja *Random Early Detection, RED*
rasparčavanje *striping*
ravnopravna obrada redova čekanja *fair queueing, weighted fair queueing*
ravnopravni *peers*
razdelnik *splitter*
razgranato stablo *spanning tree*
razmena ključa *key exchange*
razmena šifara na Internetu *Internet Key Exchange, IKE*
razrešivač *resolver*
razvodni orman *wiring closet*
razvodnik *hub*
realizacija, implementacija *implementation*
redni brojevi *sequence numbers*
redovna usluga, regular na usluga *regular service*
server za dodelu kupona proizvođača
referentni model *TCP/IP Reference Model*
referentni sistem ISO OSI *International Standards Organization Open Systems Interconnection (ISO OSI)*
reflektometrij vremenskog *time domain reflectometiy*
regionalna centrala *RBOC*
regionalna mreža, širokog područja *Wide Area Network, WAN*
regionalna telefonska centrala *toll office*
regionalne organizacije *Regional Authorities, RAs*
regionalni vodovi *toll connecting trunks*
rekurzivno ispitivanje, rekurzivni upit *recursive query*
repetitor, pojačivač *repeaters*
reprodukovanje tokom preuzimanja (u realnom *streaming media*
rešetkasto kodirana modulacija *Trellis Coded Modulation, T'CM*
režim *mode*
režim šifrovanja s povratnom *cipher feedback mode*
režim uzastopnog šifrovanja *stream cipher mode*
RF komunikacija, komunikacija na -frekvencijama *radio frequency communication*
rođendanski napad *birthday attack*
run-length kodiranje *run-length encoding*
runda *round*
s gubicima *lossy*
sastavljanje (poruke) *composition*
satelit niske orbite *Low-Earth Orbit (LEO) satellite*
satelit s orbitom *Geostationary Earth Orbit (GEO) satellite*
satelit srednje orbite *Medium-Earth Orbit (MEO) satellite*
savijena cev *bent pipe*
sažetak *message digest, MD*
sekvenca podintervala *chip sequence*
selektivno plavljenje *selective flooding*
selektivno ponavljanje *selective repeat*
sertifikat *certificate*
server imena *name server*
server *directory server*
server osnovnih *top-level domain server*
server procesa *process server*
server za proveru identiteta serveri za

prosleđivanje e-pošte uz šifrovanje serverska vezivna funkcija servis, usluga sesija signaliziranje za pojedinačne kanale signalni okvir silazno multipleksiranje sindrom luckastog prozora sinhrona digitalna hijerarhija sinhrona optička mreža, SONET mreža sinhroni korisnički podaci

sinhronizovanje sinhrono povezivanje sa uspostavljanjem direktne veze sinhrono upravljanje povezivanjem podataka sinusni noseći talas sistem GPS, globalni sistem za pozicioniranje sistem GSM, globalni sistem mobilnih komunikacija sistem imenovanja domena sistem IMTS, poboljšani sistem mobilne telefonije sistem kodiranja base64 sistem ubrzanog prenosa podataka za GSM

sistemska usluga sistemska usluga „na odmoru sam do“ sistemski podaci, nekoristan teret sistemski podaci odeljaka skokovito frekventno širenje spektra

skoro video na zahtev skretnica, mrežna skretnica
Ticket-Granting Server, TGS
Authentication Server, AS
cypherpunk remailers

server stub
service
session
channel-associated signaling
beacon frame
downward multiplexing
silly window syndrome
Synchronous Digital Hierarchy, SDH
Synchronous Optical NETwork, SONET
Synchronous Payload Envelope, SPE
synchronization
Synchronous Connection Oriented, SCO

Synchronous Data Link Control, SDLC
sine wave carrier
Global Positioning System, GPS
Global System for Mobile Communications, GSM
Domain Name System, DNS
Improved Mobile Telephone System, IMTS

base64 encoding
Enhanced Data rates for GSM Evolution, EDGE
daemon
vacation daemon

overhead

section overhead
Frequency Hopping Spread Spectrum, FHSS
near video on demand
switch
skup javnih ključeva
skup privatnih ključeva
skup protokola
skupovi zapisa resursa

slabljenje slabljenje zbog različitih putanja sloj sloj aplikacija sloj prezentacije sloj sesije sloj veze podataka smeško solitoni (vrsta impulsa) specifikacija DOCSIS, specifikacija interfejsa za kablovski prenos podataka specifikacija toka spoj ni kabl spoljni modul spoljni protokol za mrežni prolaz spori algoritam srednje odstupanje srednja frekvencija sredstvo komunikacije SSL sloj, sloj bezbednih utičnica stablo optimalnih putanja stablo zasnovano na korenu stacionarni računari standard za digitalno potpisivanje standard za šifrovanje podataka, DES standard, DES šifrovanje standardno odstupanje stara dobra telefonska usluga statičko usmeravanje stavljanje pozivaoca na čekanje strane Java servera, JSP strane sukobljavanje, sukob svetleća dioda, svetlosna dioda, LED dioda šifra šifrovanje
public key ring
private key ring
protocol stack
Resource Record Sets, RRSets
attenuation
multipath fading

layer
application layer
presentation layer
session layer
data link layer
smiley solitons
Data Over Cable Service Interface Specification, DOCSIS

flow specification
drop cable
front-end-module
exterior gateway protocol
slow start
mean deviation
medium frequency, MF
communication medium
Secure Socket layer, SSL
sink tree

core-based tree

stationary hosts
Digital Signature Standard, DSS
Data Encryption Standard, DES

standard deviation
Plain Old Telephone Service, POTS
static routing
call waiting

JavaServer Pages, JSP

collision
Light Emitting Diode, LED
cipher
encryption

šifrovanje uz knjigu šifara — *Electronic Code Book mode*
šifrovanje veze — *link encryption*
šifrovanje zamenom slova slovom — *monoalphabetic substitution*
šifrovan tekst — *ciphertext*
širokopojasni CDMA — *Wideband CDMA, W-CDMA*
širokopojasno — *broadband*
šlepovanje — *piggybacking*
štafetni prenos okvira — *frame relay*
šum — *noise*
šum kvantizacije — *quantization noise*
tabela prstiju — *finger table*
tačka pristupa mreži — *Network Access Points: NAP*
tajmer za ograničenje čekanja — *persistence timer*
tajmer za ponovno — *retransmission timer*
tajmer za proveru veze — *keepalive timer*
tandem centrala — *tandem office*
tanki Ethernet — *thin Ethernet*
tapl, dublet — *tuple*
tarifa — *tariff*
telegrafske stanice s čitačem/bušačem trake — *torn tape office*
telekomunikacioni — *carrier hotels*
televizija sa antenom — *Community Antenna Television*
televizija visoke cije, HD televizija — *High Definition Tele vision, HDTV*
telo (poruke) — *body*
teorija svrstavanja u redove čekanja — *queueing theory*
tok — *flow*
topologija pasivne zvezde — *passive star topology*
topologija prstena — *ring topology*
trajan kolačić — *persistent cookie*
trajna virtuelna kola — *permanent virtual circuits*
trajne veze — *pen istent connections*
trajni — *persistent*
transakcija — *transaction*
tiansponder — *transponder*
transportna jedinica — *transport entity*
transportni mrežni — *transport gateways*
transportni protokol — *transport protocol*
transportni sloj — *transport layer*
transpoziciono — *transposition cipher*
tranzitna mreža — *transit network*
trenutno poruka — *instant messaging*
trigraf — *trigram*
trostepeno — *three-way handshake*
ubodna račva — *vampire taps*
učenje na iskustvu — *backward learning*
ujednačavanje saobraćaja — *traffic shaping*
ulančavanje blok-šifara — *cipher block chaining*
ulazni luk — *input arc*
uljez — *intruder*
umetanje bajtova — *byte stuffing*
umetanje bitova — *bit stuffing*
umetanje znakova — *character stuffing*
umrežen računar — *host*
univerzalni sistem mobilnih tele komunikacija — *Universal Mobile Telecommunication System, UMTS*
unutrašnji protokol za mrežni prolaz — *interior gateway protocol*
uokviravanje — *framing*
upotreba tunela — *tunneling*
upravljački segment — *control plane*
upravljanje dijalogom — *dialog control*
upravljanje logičkom vezom — *Logical Link Control, LLC*
upravljanje pristupom medijumima — *Medium Access Control, MAC*
upravljanju tokom na osnovu povratnih informacija — *feedback-based flow control*
upravljanje tokom zasnovano na ograničenju brzine — *rate-based flow control*
upredena parica — *twisted pair*
URL adresa, adresa resursa — *Uniform Resource Locator, URL*
URL adresa sa sopstvenim sertifikatom — *self-certifying URL*
usaglašavanje — *handshake*
usluga, servis — *service*
usluga bez uspostavljanja direktne veze — *connectionless*
usluga datagrama, datagrafska usluga — *datagram service*
usluga datagrama s potvrdom o — *acknowledged data gram service*
usluga GPRS, opšta paketna radio — *General Packet Radio Service, GPRS*
usluga LMDS, distributivna usluga za više korisnika — *Local Multipoint Distribution Service, LMDS*
usluga MMDS, višekanalna distributivna usluga za više korisnika — *Multichannel Multi point Distribution Service, MMDS*

usluga odgovaranja na zahteve — *request-reply service*
usluga sa uspostavljanjem direktne veze — *connection-oriented service*
usluga SMS, usluga prenosa kratkih — *Short Message Service, SMS*
usluga večnog trajanja — *eternity service*
usluge PCS, usluge ličnih komunikacija — *Personal Communications Services, PCS*
usluge s više klasa kvaliteta — *class-based quality of setyice*
usmeravanje — *routing*
usmeravanje na vektora razdaljine — *distance vector routing*
usmeravanje na više odredišta — *multidestination routing*
usmeravanje na zahtev zasnovano na razdaljine — *Ad hoc On-demand Distance Vector, AODV*
usmeravanje za sesiju — *session routing*
usmeravanje na stanju veze — *link state routing*
usmereni snop — *spot beam*
usmerivač — *router*
usmerivački softver — *routing software*
ustrojavanje — *marshaling*
utičnica — *socket*
UTP kabl, upredene parice — *Unshielded Twisted Pair, UTP*
uzlazno — *upward multiplexing*
vektor za dodelu NAV vektor — *Network Allocation Vector, NAV*
veza, hiperveza — *link, hyperlink*
vezivna funkcija — *stub*
video na zahtev — *video on demand*
virtuelna lokalna — *Virtual LAN, VLAN*
virtuelna privatna — *Virtual Private Network, VPN*
virtuelno kolo — *virtual circuit, VC*
visoka frekvencija — *high frequency, HF*
visokobrzinsko sekvencijalno spektra — *High Rate Direct Sequence Spread Spectrum, HR-DSSS*
višekorisnički pristup — *multiaccess*
višekorisnički pristup uz izbegavanje — *Miltiple Access with Collision Avoidance, MACA*
višeprotokolarni usmerivač — *multiprotocol router*
višerežimsko vlakno, muldmodno vlakno — *multimode fiber*
višesmerna okosnica — *Multicast Backbone, MBone*
višesmerni usmerivač — *multicast router, mrouter*
višesmerno — *multicast, multicasting*
višesmerno — *multicast routing*
višesmerno nezavisno od — *Protocol Independent Multicast, PIM*
višestruki okviri — *multiframes*
vod, kabl — *trunk*
vokoder — *vocoder*
Volšov kod — *Walsh codes*
vratar — *gatekeeper*
vrati se n — *go back n*
vreme boravka — *dwell time*
vremenski ALOHA — *slotted ALOHA*
vremenski točak — *timing wheel*
vremensko — *timestamping*
VSAT terminal, s vrlo uskim snopom — *Very Small Aperture Terminal, VSAT*
vučni server — *pull server*
Web pošta — *Webmail*
Web strane — *Web pages*
zabranjena oblast — *forbidden region*
zaglavlje — *header*
zaglavlje odgovora — *response headers*
zaglavlje okvira — *frame header*
zaglavlje tabele — *table header*
zaglavlje zahteva — *request header*
zaglavlje za proveru identiteta — *Authentication Header, AH*
zagušenje — *congestion*
zahtev za slanje — *Request to Send, RTS*
zahtevi za komentare — *Request For Comments, RFCs*
zajedničko za sve kanale — *common-channel signaling*
zapisi resursa — *resource records*
zastupnički, — *proxy*
zaštitna barijera, — *firewall*
zatrovani keš — *poisoned cache*
zavisan od fizičkog medijuma? — *Physical Medium Dependent, PMD*
završni blok okvira — *frame trailer*
završni sistem — *Cable Modem Termination System, CMTS*
kablovskog modema — *tion System, CMTS*
znak za novi red — *carriage return*
zona — *zone*
žeton — *token*

INDEKS

A

B

DŽ

E

F

G

L

M

N

O

P

Q

R

S

Š

T

U

OAUTORU

Endru S. Tanenbaum (Andrew S. Tanenbaum) diplomirao je i na Masačusetskom tehničkom institutu i na Kalifornijskom univerzitetu u Berkliju. Trenutno je profesor računarstva na univerzitetu Vrije u Amsterdamu, Holandija, gde vodi Grupu za računarske sisteme. Takođe je i dekan Više škole za računarstvo i digitalnu grafiku - međuuniverzitetske institucije koja istražuje napredne paralelne, distribuirane digitalne grafičke sisteme. Bez obzira na takve svoje obaveze, čvrsto je rešen da se ne pretvori u činovnika.

Tokom svog radnog veka istraživao je programske prevodioce, operativne sisteme, umrežavanje i lokalno distribuirane sisteme. Njegova današnja istraživanja pretežno su usmerena na projektovanje i realizaciju široko distribuiranih sistema koji obuhvataju i do milijardu korisnika. Ta istraživanja, koja je sproveo zajedno sa prof. Maarten van Steenom, opisana su na Web lokaciji www.cs.vu.nl/globe. Iz njih je proisteklo preko sto radova izloženih na naučnim skupovima i objavljenih u časopisima, kao i pet knjiga.

Prof. Tanenbaum je i plodan autor softvera. Bio je glavni tvorac popularne alatke za pisanje prenosivih programskih prevodilaca - Amsterdam Compiler Kit - kao i MINX-a, malog klona sistema UNIX namenjenog studentskim laboratorijskim vež- bama. Taj sistem je podstakao Torvalda Linusa da napravi Linux. Zajedno sa svojim diplomcima i programerima, učestvovao je u projektovanju distribuiranog operativnog sistema visokih performansi - Amoeba, koji se zasniva na mikrojezgru. MINX i Amoeba se sada mogu besplatno preuzeti sa Interneta.

Njegovi studenti su se po diplomiranju proslavili u raznim oblastima, čime se prof. Tanenbaum ponosi - baš kao kvočka novoizleglim pilićima.

Prof. Tanenbaum je član asocijacije ACM, instituta IEEE i Holandske kraljevske akademije nauka i umetnosti. Dobitnik je brojnih nagrada: ACM nagrade (Karl V. Karlstrom) za izuzetne uspehe u nastavi 1994. godine, slične ACM/SIGCSE nagrade 1997. godine i, 2002. godine, nagrade Texty za izvanredne udžbenike. Tanenbaum se nalazi na listi *Ko je ko u svetu.* Njegova adresa na Webu je http://www.cs.vu.nl/~ast/.

I

Brzina prenosa podataka (b/s)

Simbolički Brojčano P^3 P^3(mod 33)

C^7 (mod 33) Simbolički

S	19	6859	28
	13492928512	19	S
U	21	9261	21
	1801088541	21	U
Z	26	17576	20
	1280000000	26	Z
A	01	1	1
01	A		
N	14	2744	5
14	N		
N	14	2744	5
14	N		
E	05	125	26
	8031810176	05	E

>. „ , /

v

Izračunavanja kod pošiljaoca

kod primaoca

Slika 8-17. Primer korišćenja RSA algoritma.

Pošto su za ovaj primer izabrani tako mali prosti brojevi, svaka poruka P mora da bude kraća od 33 bita, pa svaki blok osnovnog teksta može da obuhvati samo jedno slovo. Kao rezultat se dobija prosta zamena slova slovom, što baš ne ostavlja utišale. Da smo $z\,a\,p\,i\,q$ izabrali brojeve reda veličine 2^{512}, n bi bilo reda 2^{1024}, pa bi blokovi mogli biti veličine 1024 bita i sadržati 128 osmobitnih slova, što je neuporedivo više nego kod DES-a (8 slova) ili AES-a (16 slova).