

Računarske mreže i mrežne tehnologije

1. Jednostavan Java chat klijent/server

Klijentski deo:

```
import java.net.*;
import java.io.*;
import java.util.*;

public class ChatClient {

    // Port je konstanta, mora da bude isti za klijent i server
    private static final int PORT = 12345;

    public static void main(String[] args) throws IOException {

        PrintWriter out = null;
        BufferedReader in = null;

        if (args.length != 1) {
            System.out.println("Koriscenje: java ChatClient hostname");
            System.exit(0);
        }

        // Konekcija na dati host
        InetAddress address = InetAddress.getByName(args[0]);

        // Pokusaj konekcije na ChatServer
        Socket socket = new Socket(address, PORT);

        try {
            out = new PrintWriter(new BufferedWriter(new OutputStreamWriter(
                socket.getOutputStream())), true);

            in = new BufferedReader(new InputStreamReader(socket
                .getInputStream()));

            Scanner sc = new Scanner(System.in);
            String fromServer;
            String fromUser;

            while ((fromServer = in.readLine()) != null) {
                System.out.println("Server: " + fromServer);
                if (fromServer.equals("Cao."))
```

```
        break;
        fromUser = sc.nextLine();
        if (fromUser != null) {
            System.out.println("Client: " + fromUser);
            out.println(fromUser);
        }
    } finally {
        socket.close();
        in.close();
        out.close();
    }
}
```

Serverski deo:

```
import java.io.*;
import java.net.*;
import java.util.Scanner;

public class ChatServer {
    public static final int PORT = 12345;

    public static void main(String[] args) throws IOException {

        ServerSocket serverSocket = new ServerSocket(PORT);
        BufferedReader in = null;
        PrintWriter out = null;

        while (true) {
            // Blokira se do klijentskog zahteva za konekcijom
            Socket socket = serverSocket.accept();

            try {
                in = new BufferedReader(new InputStreamReader(socket
                    .getInputStream()));

                out = new PrintWriter(new BufferedWriter(
                    new OutputStreamWriter(socket.getOutputStream())), true);

                Scanner sc = new Scanner(System.in);
                String fromClient;
                String fromUser;

                // Inicira komunikaciju slanjem prve poruke
                out.println("Server aktivan");

                while ((fromClient = in.readLine()) != null) {
                    System.out.println("Client: " + fromClient);
                    if (fromClient.equals("Cao."))
                        break;

                    fromUser = sc.nextLine();

                    if (fromUser != null) {
                        System.out.println("Server: " + fromUser);
                        out.println(fromUser);
                    }
                }
            } finally {
                socket.close();
                out.close();
                in.close();
            }
        }
    }
}
```

```
    }  
  }  
} // while true  
}
```

Aplikativni sloj

Aplikacioni sloj je sloj u kome su smeštene sve direktne primene mrežnih tehnologija. Ispod njega se nalazi transportni sloj čiji je zadatak da obezbedi siguran transfer podataka.

Međutim, čak i u aplikacionom sloju postoji potreba za određenim protokolima podrške. Jedan od njih je DNS, sistem za imenovanje na Internetu.

1. DNS - Domain Name System

U samim počecima ARPANET-a, prethodnika Interneta, sistem imenovanja nije postojao, već se adresiranje isključivo svodilo na baratanje IP adresama, što je bilo jako teško za pamćenje. Tu je i drugi problem: Recimo da je nečija e-mail adresa `neko@147.91.209.20`. Šta ako organizacija odluči da migrira e-mail server na drugu našinu sa drugim IP brojem? To bi automatski značilo da se i e-mail adresa menja.

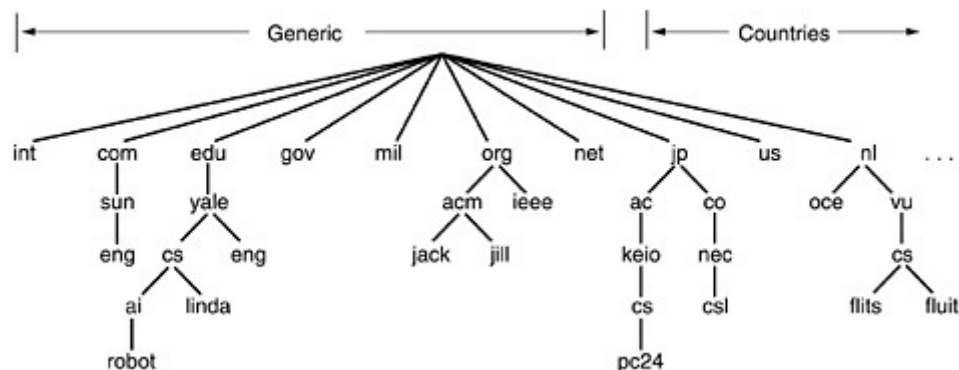
Jedno od prvih rešenja ovog problema je postavljajnje servera koji bi čuvao fajl, npr. `hosts.txt`, a hostovi povremeno (npr. jednom dnevno) skidali taj fajl. Ostaci ove metodologije i danas postoje u vidu fajla `/etc/hosts` na svakom UNIX sistemu.

Međutim, kako je Internet rastao, moralo se napraviti nešto bolje, i to, naravno, sa većim stepenom distribuiranosti, tako da je patentiran DNS. Suština DNS-a je hijerarhijski sistem imenovanja baziran na domenima.

Da bi aplikacija mapirala ime na IP adresu, ona zove lokalnu bibliotечku proceduru, kao što je `gethostbyname()` koja šalje UDP paket lokalnom DNS serveru, a on vraća IP adresu koju pronade ili u svojoj tabeli ili na nekom drugom DNS serveru.

Na slici je dat deo DNS hijerarhije:

Figure 7-1. A portion of the Internet domain name space.



Primeri adresa su spider.pmf.kg.ac.rs, cs.yale.edu, cs.keio.ac.jp itd. Ograničenje je da puna tekstualna adresa ne sme biti duža od 255 karaktera, sa komponentama koje nisu duže od 63 karaktera.

1.1 Zapisi resursa

Svaki domen, bio to jedan host ili domen najvišeg nivoa može imati skupinu resursa koji su mu pridruženi. Za jedan jedini host, najčešće je to jedna IP adresa i ništa više. Za veliki domen, kao što je npr. kg.ac.rs, može se pridružiti ime podrazumevanog e-mail servera i još mnogo toga.

Format tabela resursa kod DNS-a izgleda ovako:

| Domain_name | Time_to_live | Class | Type | Value |
|-------------|--------------|-------|------|-------|
|-------------|--------------|-------|------|-------|

Domain_name je ime domena na koji se resurs odnosi.

Time_to_live je indikacija stabilnosti datog resursa, što je veće informacija u zapisu je stabilnija.

Class polje je za Internet uvek IN.

Type označava tip resursa i bira se iz sledeće table:

Figure 7-2. . The principal DNS resource record types for IPv4.

| Type | Meaning | Value |
|-------|----------------------|---|
| SOA | Start of Authority | Parameters for this zone |
| A | IP address of a host | 32-Bit integer |
| MX | Mail exchange | Priority, domain willing to accept e-mail |
| NS | Name Server | Name of a server for this domain |
| CNAME | Canonical name | Domain name |
| PTR | Pointer | Alias for an IP address |
| HINFO | Host description | CPU and OS in ASCII |
| TXT | Text | Uninterpreted ASCII text |

- SOA zapis daje osnovne parametre zone za koju je dati DNS server zaduzen.
- Najvazniji zapis je A, odn.veza između imena hosta i IP adrese
- MX daje prioritet servera za transfer e-maila u tom domenu
- NS specificiraju name server-e domena
- CNAME je mehanizam za dodeljivanje alias-a. Recimo ime pravog web servera univerziteta je licej.unic.kg.ac.rs, ali dat mu je alias www.kg.ac.rs
- PTR se uglavnom koristi za reverzne pretrage, tj. da se unosom IP adrese dobije tekstualno ime host-a
- HINFO daje opis hosta
- TXT se koristi za komentare

Evo primera:

Figure 7-3. A portion of a possible DNS database for cs.vu.nl

```
; Authoritative data for cs.vu.nl
cs.vu.nl.      86400  IN  SOA   star boss (9527,7200,7200,241920,86400)
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.
www.cs.vu.nl.   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

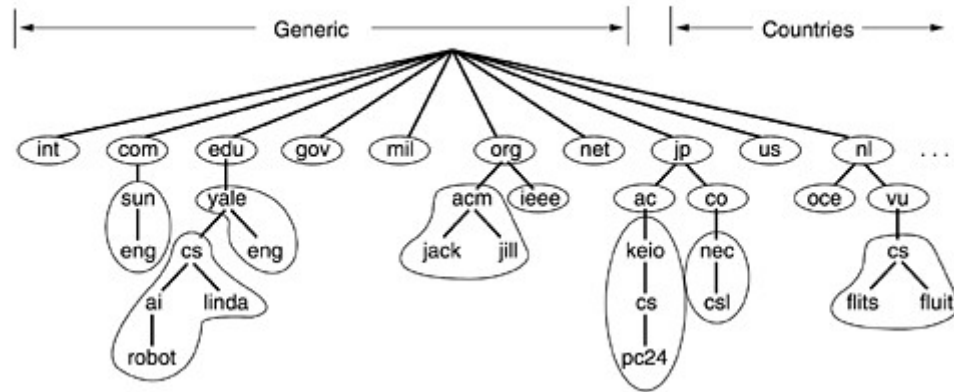
little-sister  IN  A     130.37.62.23
               IN  HINFO Mac MacOS

laserjet       IN  A     192.31.231.216
               IN  HINFO "HP Laserjet III Si" Proprietary
```

1.2 Serveri imena

Čisto teorijski, jedan server imena bi mogao u sebi da sadrži imena svih domena i hostova na Internetu. U praksi se, ipak, pristupa podeli na disjunktne zone čiji su serveri imena zaduženi samo za opsluživanje samo svog domena:

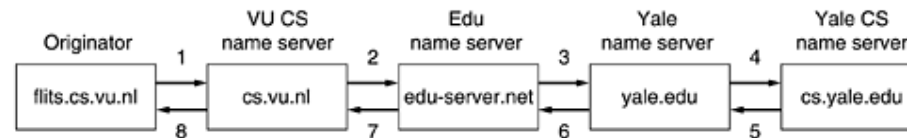
Figure 7-4. Part of the DNS name space showing the division into zones.



Ukoliko je traženi host/domen unutar lokalnog DNS domena, lokalni DNS odmah šalje jedanod svojih lokalnih zapisa onome koji je tražio rezoluciju imena.

Ako je pak traženi host van lokalnog DNS domena, upit se prosleđuje DNS serveru višeg nivoa. Da bi ovo bilo jasnije, data ej sledeća shema upita kao primer:

Figure 7-5. How a resolver looks up a remote name in eight steps.



U datom primeru host flits.cs.vu.nl traži rezoluciju imena cs.yale.edu i dobija je u 8 koraka. Kao što se iz priloženog da videti, proces je rekurzivan.

2. Elektronska pošta

Elektronska pošta kao servis postoji već više od dve decenije. Tokom devedesetih godina broj elektronskih poruka je polako prevazišao broj pismonosnih pošiljaka koje se šalju običnom poštom.

Standardi aplikacionog sloja koji se danas koriste proizilaze iz dokumenata RFC821 (protokol za prenos) i RFC822 (format poruke). Ovi standardi definišu e-mail adresu u obliku:

korisnik@host

npr.

zoki@knez.kg.ac.rs

Ukoliko je u DNS servisu za domen kg.ac.rs definisan e-mail server knez.kg.ac.rs zapisom MX tipa (videti deo o DNS-u), gornja e-mail adresa može da glasi i:

zoki@kg.ac.rs

Svaka poruka (klasična ili elektronska) sastoji se iz 3 dela: omotača (koverta), zaglavlja, i same poruke. Danas se elektronskom poštom barata uz pomoć grafičkih alata kao što su Mozilla Thunderbird, KMail, Outlook... Ti agenti komuniciraju sa e-mail serverom omogućavajući korisniku da poruke čita, obriše, prosledi itd...

2.1 Format poruke

Poruke se sastoje iz primitivnog omotača (definiše ih RFC821), jednog broja linija zaglavlja i prazne linije nakon čega sledi telo poruke. Svako polje zaglavlja se sastoji iz jedne linije ASCII teksta koji sadrži ime polja, dvotačku i vrednost. Osnovna polja zaglavlja data su u tabeli:

Figure 7-9. RFC 822 header fields related to message transport.

| Header | Meaning |
|--------------|---|
| To: | E-mail address(es) of primary recipient(s) |
| Cc: | E-mail address(es) of secondary recipient(s) |
| Bcc: | E-mail address(es) for blind carbon copies |
| From: | Person or people who created the message |
| Sender: | E-mail address of the actual sender |
| Received: | Line added by each transfer agent along the route |
| Return-Path: | Can be used to identify a path back to the sender |

Takođe, tu su i neka dodatna polja koja mogu biti od koristi programima za transfer pošte, kao i ljudima koji na ovaj način međusobno korespondiraju:

Figure 7-10. Some fields used in the RFC 822 message header.

| Header | Meaning |
|--------------|---|
| Date: | The date and time the message was sent |
| Reply-To: | E-mail address to which replies should be sent |
| Message-Id: | Unique number for referencing this message later |
| In-Reply-To: | Message-Id of the message to which this is a reply |
| References: | Other relevant Message-Ids |
| Keywords: | User-chosen keywords |
| Subject: | Short summary of the message for the one-line display |

2.2 MIME - The Multipurpose Internet Mail Extensions

U ranim danima Interneta, poruke su se uglavnom sastojale isključivo od ASCII teksta napisanog na Engleskom jeziku. Tada je definicija koju daje RFC822 bila potpuno prihvatljiva, ali danas to više nije slučaj, i to zbog sledećih tipova poruka:

- poruke na jezicima sa akcentima (francuski, španski...)
- poruke na jezicima na ne-Latinskim pismima (srpska ćirilica, ruski)
- poruke na jezicima bez alfabetu (kineski, japanski)
- poruke koje uopšte i ne sadrže tekst (audio, slike, video...)

Rešenje je predloženo pod nazivom MIME i danas je u širokoj upotrebi. Ideja je da se transfer i dalje obavlja preko starog standarda zbog očuvanja kompatibilnosti, ali da poruka definiše i način na koji je treba kodirati. MIME standard definiše pet novih zaglavlja:

Figure 7-11. RFC 822 headers added by MIME.

| Header | Meaning |
|----------------------------|--|
| MIME-Version: | Identifies the MIME version |
| Content-Description: | Human-readable string telling what is in the message |
| Content-Id: | Unique identifier |
| Content-Transfer-Encoding: | How the body is wrapped for transmission |
| Content-Type: | Type and format of the content |

Poruka koja ne sadrži *MIME-Version* polje, tretira se kao obična plain-tekst poruka na engleskom jeziku. Polje *Content-Id* ima sličnu funkciju kao *Message-Id* u osnovnom standardu.

Polje *Content-Transfer-Encoding* govori kako se kodira poruka za transfer preko standardnog e-mail protokola. E-mail protokol ima dva ograničenja: (1) poruke se sastoje iz ASCII teksta koji koristi 7 bita i (2) linija ne sme biti duža od 1000 karaktera. Pitanje je kako neki binarni niz podataka kodirati na ovaj način. Danas je u širokoj upotrebi jednostavno **base64 kodiranje** koje se sastoji u tome da se grupe od po 24 bita podele u 4 6-bitne jedinice da se svaka od njih šalje kao običan ASCII znak.

Polje *Content-Type* je najinteresantnije, ono opisuje pravu prirodu sadržaja poruke. Primer vrednosti polja *Content-Type* je:

Content-Type: video/mpeg

dok su još neki primeri dati u tabeli:

Figure 7-12. The MIME types and subtypes defined in RFC 2045.

| Type | Subtype | Description |
|-------------|---------------|---|
| Text | Plain | Unformatted text |
| | Enriched | Text including simple formatting commands |
| Image | Gif | Still picture in GIF format |
| | Jpeg | Still picture in JPEG format |
| Audio | Basic | Audible sound |
| Video | Mpeg | Movie in MPEG format |
| Application | Octet-stream | An uninterpreted byte sequence |
| | Postscript | A printable document in PostScript |
| Message | Rfc822 | A MIME RFC 822 message |
| | Partial | Message has been split for transmission |
| | External-body | Message itself must be fetched over the net |
| Multipart | Mixed | Independent parts in the specified order |
| | Alternative | Same message in different formats |
| | Parallel | Parts must be viewed simultaneously |
| | Digest | Each part is a complete RFC 822 message |

Najinteresantniji tip je Multipart, koji dozvoljava da se u jednu poruku smesti više priloga. Ovo je način na koji e-mail programi “lepe” tzv. atačmente. Primer poruke sa priložima u izvornom obliku dat je na sledećoj slici:

Figure 7-13. A multipart message containing enriched and audio alternatives.

From: elinor@abcd.com
To: carolyn@xyz.com
MIME-Version: 1.0
Message-Id: <0704760941.AA00747@abcd.com>
Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
Subject: Earth orbits sun integral number of times

This is the preamble. The user agent ignores it. Have a nice day.

--qwertyuiopasdfghjklzxcvbnm
Content-Type: text/enriched

Happy birthday to you
Happy birthday to you
Happy birthday dear <bold> Carolyn </bold>
Happy birthday to you

--qwertyuiopasdfghjklzxcvbnm
Content-Type: message/external-body;
 access-type="anon-ftp";
 site="bicycle.abcd.com";
 directory="pub";
 name="birthday.snd"

content-type: audio/basic
content-transfer-encoding: base64
--qwertyuiopasdfghjklzxcvbnm--

External-body označava da se telo poruke nalazi negde drugde, a ne u samoj poruci. U gornjem primeru fajl *birthday.snd* se nalazi na nekom javnom FTP serveru.

2.3 Transfer poruka - SMTP protokol

SMTP (*Simple Mail Transfer Protocol*) je odgovoran za transport poruka od mašine do mašine. Veoma je jednostavan i zasniva se na prostoj razmeni ASCII poruka, bez ikakve binarne veze. Koristi TCP protokol (jer je pouzdanost bitna) na portu 25 koji je za njega i rezervisan na svakom sistemu. *Daemon* prihvata konekcije na pomenutom portu i poruke koje prima šaljeu odgovarajuće *mailbox*-ove, a ako naiđe na grešku, poruku vraća pošiljaocu.

Na slici je dat primer otvorene SMTP sesije preko telnet-a na portu 25:

Figure 7-14. Transferring a message from elinor@abcd.com to carolyn@xyz.com.

```
S: 220 xyz.com SMTP service ready
C: HELO abcd.com
S: 250 xyz.com says hello to abcd.com
C: MAIL FROM: <elinor@abcd.com>
S: 250 sender ok
C: RCPT TO: <carolyn@xyz.com>
S: 250 recipient ok
C: DATA
S: 354 Send mail; end with "." on a line by itself
C: From: elinor@abcd.com
C: To: carolyn@xyz.com
C: MIME-Version: 1.0
C: Message-Id: <0704760941.AA00747@abcd.com>
C: Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
C: Subject: Earth orbits sun integral number of times
C:
C: This is the preamble. The user agent ignores it. Have a nice day.
C:
C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: text/enriched
C:
C: Happy birthday to you
C: Happy birthday to you
C: Happy birthday dear <bold> Carolyn </bold>
C: Happy birthday to you
C:
C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: message/external-body;
C:   access-type="anon-ftp";
C:   site="bicycle.abcd.com";
C:   directory="pub";
C:   name="birthday.snd"
C:
C: content-type: audio/basic
C: content-transfer-encoding: base64
C: --qwertyuiopasdfghjklzxcvbnm
C: .
S: 250 message accepted
C: QUIT
S: 221 xyz.com closing connection
```

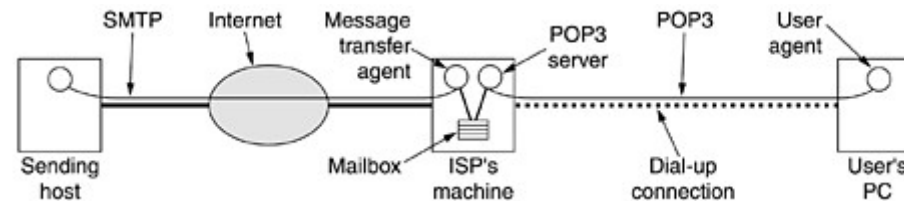
Naravno, ovu ASCII komunikaciju danas umesto korisnika obavlja grafički mail klijent. Ukoliko se poruka šalje na više adresa, treba ubaciti višestruko RCPT TO polje. Na svakom UNIX ili Windows sistemu prisutan je program *telnet* koji služi jednostavnoj ASCII komunikaciji sa bilo kojim servisom. Komanda za otvaranje SMTP sesije na serveru smtp.kg.ac.rs bila bi:

```
telnet smtp.kg.ac.rs 25
```

Pomoć se dobija komandom HELP.

2.4 Transfer poruka - POP3 protokol

Uobičajeno je da korisnički desktop računari ne učestvuju u transferu e-mail-a putem SMTP protokola jer nisu uvek uključeni i on-line i spremni da primaju poruke. Tzv. mailbox-ovi se obično nalaze na serverima davaoca internet usluga (provajdera) i po potrebi se skidaju sa njega na desktop računar korisnika. Proces je prikazan na slici:



Najrašireniji protokol koji služi tzv. konačnom transferu pošte je POP3 - *Post Office Protocol*. Standardno radi na TCP portu 110, a najbolji opis komunikacije sa njim daje sledeći primer:

Figure 7-16. Using POP3 to fetch three messages.

```

S: +OK POP3 server ready
C: USER carolyn
S: +OK
C: PASS vegetables
S: +OK login successful
C: LIST
S: 1 2505
S: 2 14302
S: 3 8122
S: .
C: RETR 1
S: (sends message 1)
C: DELE 1
C: RETR 2
S: (sends message 2)
C: DELE 2
C: RETR 3
S: (sends message 3)
C: DELE 3
C: QUIT
S: +OK POP3 server disconnecting
```

Konekcija sa servisom se postiže na sličan način kao sa SMTP-om:

```
telnet pop3.kg.ac.rs 110
```