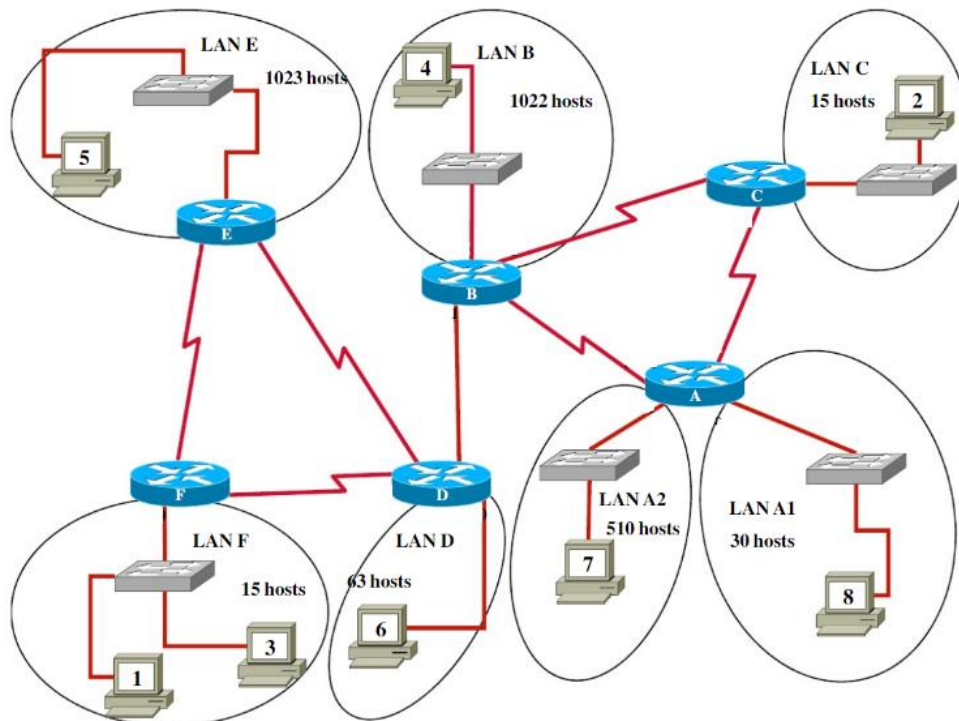


Računarske mreže i mrežne tehnologije Prvi kolokvijum

1. Dat je sledeći signal:

$$x(t) = 1.5 \sin(2\pi 100t) + \sin(2\pi 25t) + 0.5 \sin(2\pi 75t)$$

- Koliko komponentata ima amplitudski spektar složenog signala $x(t)$?
 - Nacrtati spektar ovog signala. Kolika je širina spektra?
 - Da li je signal koji se dobija zbirom konačnog broja sinusoida uvek periodičan?
 - Kolika je osnovna učestanost složenog signala $x(t)$?
2. Pošiljalac treba da pošalje niz bitova vrednosti 0xCF0.
- Za dati niz bitova odrediti koeficijente Furijeove funkcije a_n , b_n i c .
 - Koja je vrednost koeficijenta a_n trećeg harmonika?
 - Koristeći Hamingov kod, napisati tok bitova koji se stvarno šalje.
 - Ukoliko se greškom invertuje 2 bit sa desne strane, koji će kontrolni bitovi ukazivati na grešku?
3. Pošiljalac treba da pošalje sledeći niz bitova: 1001100
Ako se za prenos koristi standardna CRC metoda sa generatorskim polinomom $G(x) = x^3 + x^2 + 1$, napisati tok bitova koji se stvarno šalje.
4. Za mrežu na sledećoj slici obezbeđen je skup adresa 50.2.192.0/19.



- Koristeći dati adresni blok, odrediti adrese LAN mreža i point-to-point segmenata. Adrese dodeljivati redom, od većih ka manjim mrežama.
 - Adresirati sve interfejsa na ruterima. Na point-to-point vezama postaviti manju adresu na strani rutera sa manjim indeksom. Na LAN mrežama koristi prvu raspoloživu IP adresu.
 - Odrediti celu ruting tabelu na ruteru D.
 - Kojim putem će ići saobraćaj od računara pod rednim brojem 3 do računara pod rednim brojem 7?
5. Sažeto odgovoriti na sledeća pitanja:
- Koliko najmanje (umreženih) računara čini računarsku mrežu?
 - Šta određuje da li je računarska mreža LAN, MAN ili WAN?
 - Koja je razlika između protokola i interfejsa?

6. Od ponuđenih zaokružiti ispravne IP adrese:
 - a. 173.15.18.3.3
 - b. 173.15.18.3
 - c. 173.15.1
 - d. 173.15.288.12
 - e. 120.10.10.280
 - f. 120.0.0.1
 - g. 120.1010.10.10
7. Adresa host-a je 10.10.0.13/20
 - a. Koje je klase mreža?
 - b. Koja je mrežna maska?
 - c. Broadcast adresa?
 - d. IP adresa mreže kojoj pripada dati host?
 - e. Ukupan broj uređaja po mreži?
8. Modifikovati kod *simplex* protokola za slanje podataka bučnim kanalom tako da *sender* prihvata dva uzastopna paketa od mrežnog sloja, pa tek onda šalje ta dva paketa odjednom. *Sender* uzima nova dva paketa od mrežnog sloja tek onda kada je stigla potvrda za oba prethodno poslata paketa. Paketi koji ne stignu na određite ili na kojima se desi greška, šalju se ponovo.

```
/* Protocol 3 (par) allows unidirectional data flow over an unreliable channel. */
#define MAX_SEQ 1 /* must be 1 for protocol 3 */
typedef enum {frame_arrival, cksum_err, timeout} event_type;
#include "protocol.h"

void sender3(void)
{
    seq_nr next_frame_to_send; /* seq number of next outgoing frame */
    frame s; /* scratch variable */
    packet buffer; /* buffer for an outbound packet */
    event_type event;

    next_frame_to_send = 0; /* initialize outbound sequence numbers */
    from_network_layer(&buffer); /* fetch first packet */
    while (true) {
        s.info = buffer; /* construct a frame for transmission */
        s.seq = next_frame_to_send; /* insert sequence number in frame */
        to_physical_layer(&s); /* send it on its way */
        start_timer(s.seq); /* if answer takes too long, time out */
        wait_for_event(&event); /* frame_arrival, cksum_err, timeout */
        if (event == frame_arrival) {
            from_physical_layer(&s); /* get the acknowledgement */
            if (s.ack == next_frame_to_send) {
                stop_timer(s.ack); /* turn the timer off */
                from_network_layer(&buffer); /* get the next one to send */
                inc(next_frame_to_send); /* invert next_frame_to_send */
            }
        }
    }
}

void receiver3(void)
{
    seq_nr frame_expected;
    frame r, s;
    event_type event;

    frame_expected = 0;
    while (true) {
        wait_for_event(&event); /* possibilities: frame_arrival, cksum_err */
        if (event == frame_arrival) { /* a valid frame has arrived. */
            from_physical_layer(&r); /* go get the newly arrived frame */
            if (r.seq == frame_expected) { /* this is what we have been waiting for. */
                to_network_layer(&r.info); /* pass the data to the network layer */
                inc(frame_expected); /* next time expect the other sequence nr */
            }
            s.ack = 1 - frame_expected; /* tell which frame is being acked */
            to_physical_layer(&s); /* send acknowledgement */
        }
    }
}
```