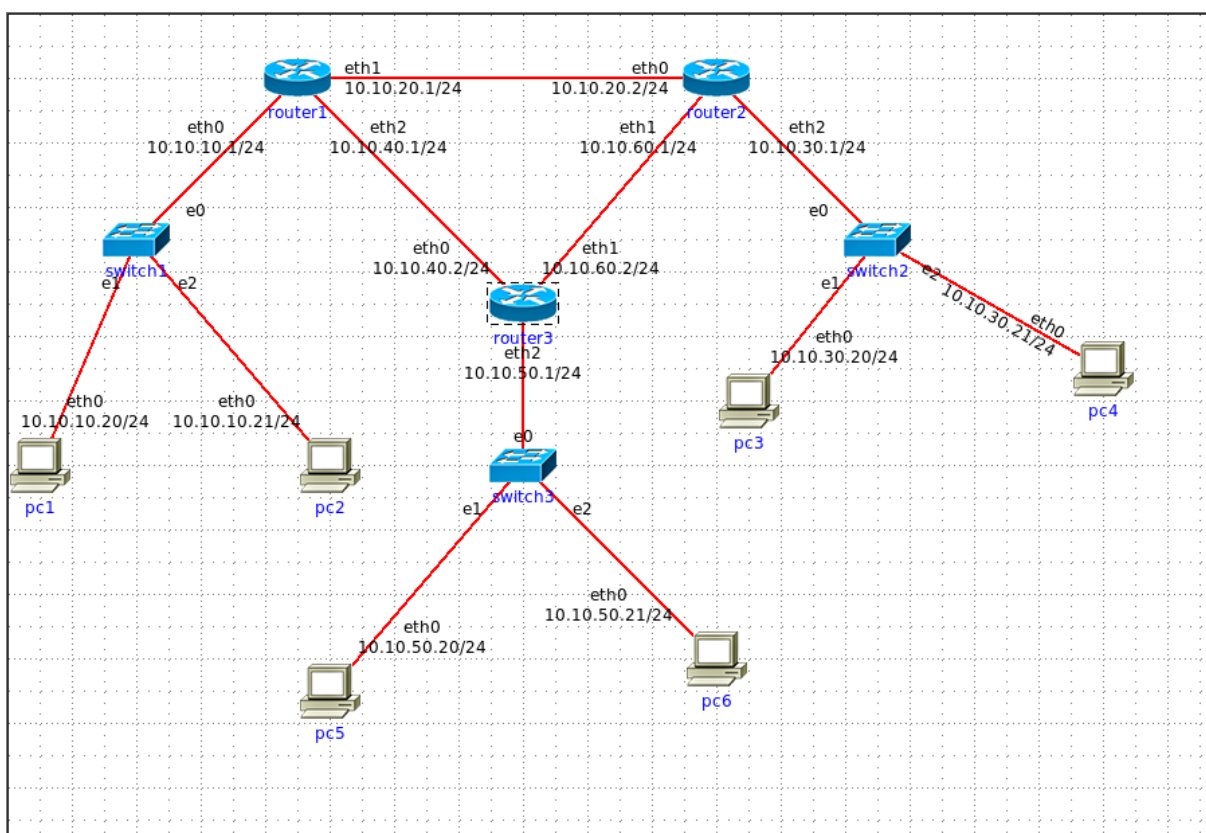


## KOMUNIKACIJSKE MREŽE 2. LABORATORIJSKA VJEŽBA

Zadatak 22. (Vlastito izrađena topologija) U emulatoru/simulatoru IMUNES konstruirajte mrežu koja sadrži tri podmreže povezane usmjeriteljima (Slika 3.16). Konfigurirajte statičko usmjeravanje između svih podmreža (dakle, bez korištenja protokola za usmjeravanje). Sučeljima računala i usmjeritelja dodijelite IP-adrese iz raspona 10.10.10.0/24, 10.10.20.0/24, 10.10.30.0/24, 10.10.40.0/24, 10.10.50.0/24 i 10.10.60.0/24. Ispišite tablice usmjeravanja svih računala i usmjeritelja.



PC1:

```
root@pc1:/ # netstat -4 -rn
Routing tables

Internet:
Destination      Gateway         Flags         Netif Expire
default          10.10.10.1     UGS          eth0
10.10.10.0/24    link#2         U            eth0
10.10.10.20     link#2         UHS          lo0
127.0.0.1       link#1         UH           lo0
root@pc1:/ #
```

PC2:

```
root@pc2:/ # netstat -4 -rn
Routing tables

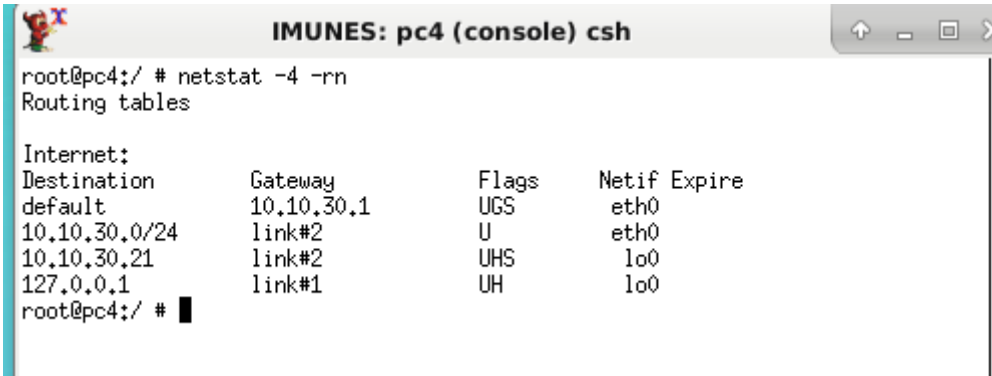
Internet:
Destination      Gateway          Flags    Netif Expire
default          10.10.10.1      UGS      eth0
10.10.10.0/24    link#2          U        eth0
10.10.10.21     link#2          UHS      lo0
127.0.0.1       link#1          UH       lo0
root@pc2:/ #
```

PC3:

```
root@pc3:/ # netstat -4 -rn
Routing tables

Internet:
Destination      Gateway          Flags    Netif Expire
default          10.10.30.1      UGS      eth0
10.10.30.0/24    link#2          U        eth0
10.10.30.20     link#2          UHS      lo0
127.0.0.1       link#1          UH       lo0
root@pc3:/ #
```

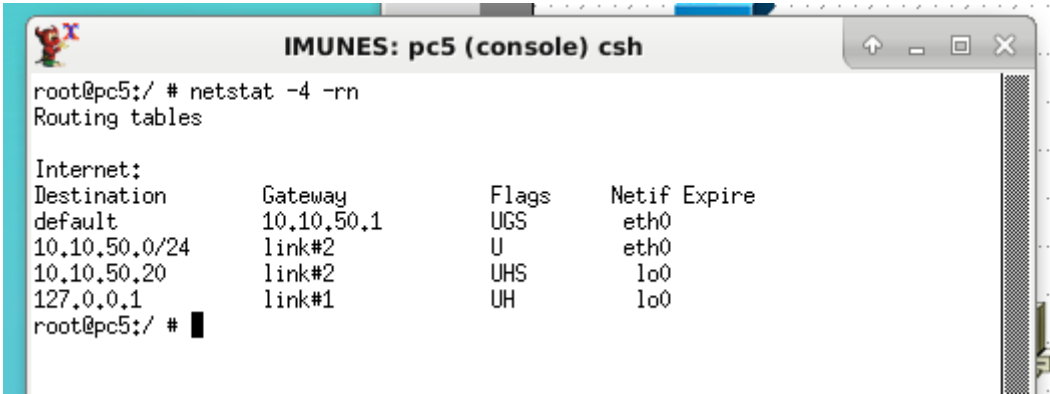
PC4:



```
root@pc4:/ # netstat -4 -rn
Routing tables

Internet:
Destination      Gateway          Flags    Netif Expire
default          10.10.30.1      UGS      eth0
10.10.30.0/24    link#2          U        eth0
10.10.30.21     link#2          UHS      lo0
127.0.0.1       link#1          UH       lo0
root@pc4:/ #
```

PC5:



```
root@pc5:/ # netstat -4 -rn
Routing tables

Internet:
Destination      Gateway          Flags    Netif Expire
default          10.10.50.1      UGS      eth0
10.10.50.0/24    link#2          U        eth0
10.10.50.20     link#2          UHS      lo0
127.0.0.1       link#1          UH       lo0
root@pc5:/ #
```

PC6:

```
IMUNES: pc6 (console) csh
root@pc6:/ # netstat -4 -rn
Routing tables

Internet:
Destination      Gateway          Flags    Netif Expire
default          10.10.50.1       UGS      eth0
10.10.50.0/24    link#2           U         eth0
10.10.50.21     link#2           UHS      lo0
127.0.0.1       link#1           UH       lo0
root@pc6:/ #
```

ROUTER1:

```
IMUNES: router1 (console) csh
root@router1:/ # netstat -4 -rn
Routing tables

Internet:
Destination      Gateway          Flags    Netif Expire
10.10.10.0/24    link#2           U         eth0
10.10.10.1      link#2           UHS      lo0
10.10.20.0/24    link#3           U         eth1
10.10.20.1      link#3           UHS      lo0
10.10.30.0/24    10.10.20.2       UGS      eth1
10.10.40.0/24    link#4           U         eth2
10.10.40.1      link#4           UHS      lo0
10.10.50.0/24    10.10.40.2       UGS      eth2
127.0.0.1       link#1           UH       lo0
root@router1:/ #
```

ROUTER2:

```
IMUNES: router2 (console) csh
root@router2:/ # netstat -4 -rn
Routing tables

Internet:
Destination      Gateway          Flags    Netif Expire
10.10.10.0/24    10.10.20.1       UGS      eth0
10.10.20.0/24    link#2           U         eth0
10.10.20.2      link#2           UHS      lo0
10.10.30.0/24    link#4           U         eth2
10.10.30.1      link#4           UHS      lo0
10.10.50.0/24    10.10.60.2       UGS      eth1
10.10.60.0/24    link#3           U         eth1
10.10.60.1      link#3           UHS      lo0
127.0.0.1       link#1           UH       lo0
root@router2:/ #
```

ROUTER3:

```
root@router3:/ # netstat -4 -rn
Routing tables

Internet:
Destination      Gateway           Flags      Netif Expire
10.10.10.0/24     10.10.40.1       UG1        eth0
10.10.30.0/24     10.10.60.1       UG1        eth1
10.10.40.0/24     link#2           U          eth0
10.10.40.2       link#2           UHS        lo0
10.10.50.0/24     link#4           U          eth2
10.10.50.1       link#4           UHS        lo0
10.10.60.0/24     link#3           U          eth1
10.10.60.2       link#3           UHS        lo0
127.0.0.1         link#1           UH         lo0
root@router3:/ #
```

**Zadatak 23.** (Vlastito izrađena topologija) U emulatoru/simulatoru IMUNES konstruirajte mrežu koja sadrži tri podmreže povezane usmjeriteljima (Slika 3.16). Konfigurirajte statičko usmjeravanje (dakle, bez korištenja protokola za usmjeravanje) tako da dođe do petlje u usmjeravanju. Sučeljima računala i usmjeritelja dodijelite IP-adrese iz raspona 10.10.10.0/24, 10.10.20.0/24, 10.10.30.0/24, 10.10.40.0/24, 10.10.50.0/24 i 10.10.60.0/24. Ispišite tablice usmjeravanja svih računala i usmjeritelja. Pomoću alata Wireshark utvrdite što se tada događa s paketima koji "uđu" u petlju te komentirajte svoja zapažanja.

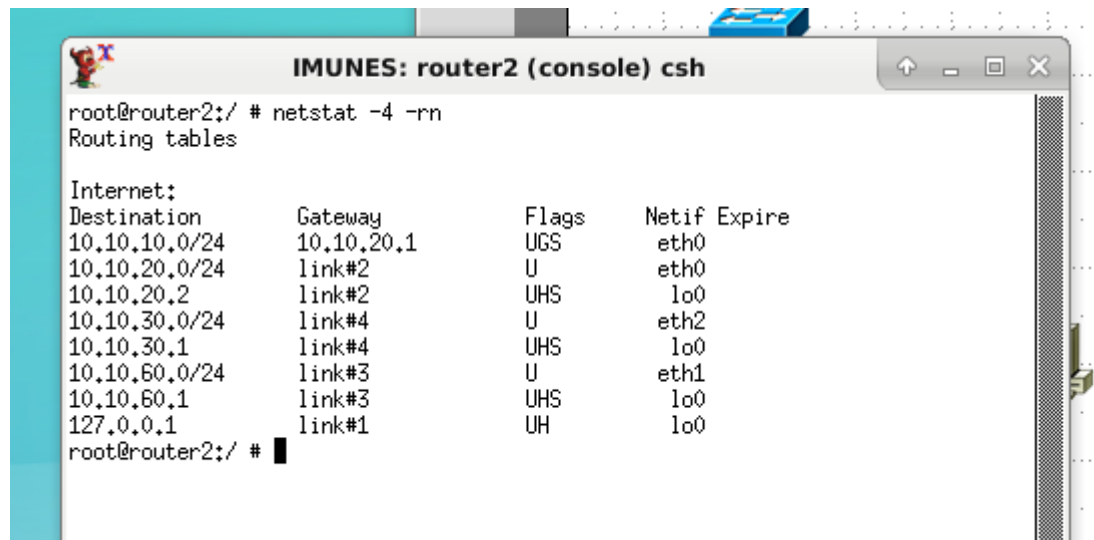
- Pokretanjem alata Wireshark vidimo da se paket samo vrti u petlji dok mu TTL ne padne na 0 te se paket onda odbaci. Tablice usmjerenja PC-eva ostaju iste ali se tablice usmjerenja usmjeritelja mijenjaju.

ROUTER1:

```
IMUNES: router1 (console) csh
root@router1:/ # netstat -4 -rn
Routing tables

Internet:
Destination      Gateway           Flags      Netif Expire
10.10.10.0/24     link#2           U          eth0
10.10.10.1       link#2           UHS        lo0
10.10.20.0/24     link#3           U          eth1
10.10.20.1       link#3           UHS        lo0
10.10.30.0/24     10.10.20.2       UGS        eth1
10.10.40.0/24     link#4           U          eth2
10.10.40.1       link#4           UHS        lo0
127.0.0.1         link#1           UH         lo0
root@router1:/ #
```

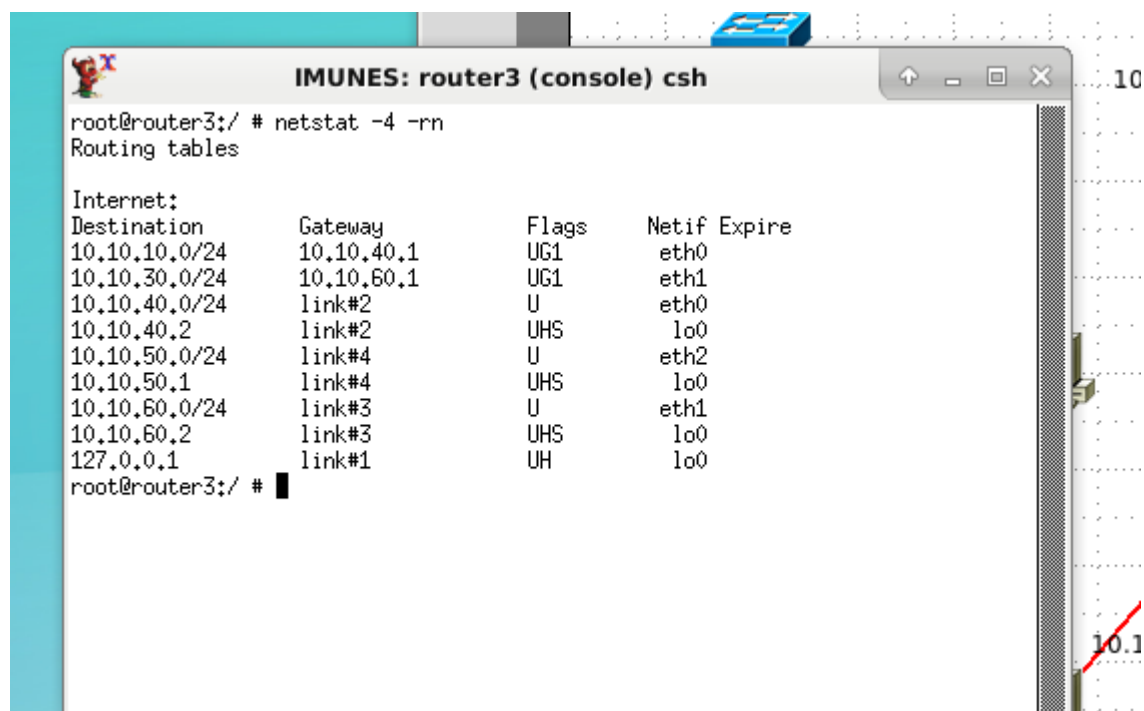
ROUTER2:



```
root@router2:/ # netstat -4 -rn
Routing tables

Internet:
Destination      Gateway          Flags    Netif Expire
10.10.10.0/24     10.10.20.1      UGS      eth0
10.10.20.0/24     link#2          U        eth0
10.10.20.2        link#2          UHS      lo0
10.10.30.0/24     link#4          U        eth2
10.10.30.1        link#4          UHS      lo0
10.10.60.0/24     link#3          U        eth1
10.10.60.1        link#3          UHS      lo0
127.0.0.1         link#1          UH       lo0
root@router2:/ #
```

ROUTER3:



```
root@router3:/ # netstat -4 -rn
Routing tables

Internet:
Destination      Gateway          Flags    Netif Expire
10.10.10.0/24     10.10.40.1      UG1      eth0
10.10.30.0/24     10.10.60.1      UG1      eth1
10.10.40.0/24     link#2          U        eth0
10.10.40.2        link#2          UHS      lo0
10.10.50.0/24     link#4          U        eth2
10.10.50.1        link#4          UHS      lo0
10.10.60.0/24     link#3          U        eth1
10.10.60.2        link#3          UHS      lo0
127.0.0.1         link#1          UH       lo0
root@router3:/ #
```

**Zadatak 24.** (Topologija RIP/RIP1.imn) Započnite simulaciju i s računala pc provjerite dostupnost (naredba ping) poslužitelja server i očitajte TTL vrijednost iz ispisa. Zatim, s poslužitelja server provjerite dostupnost računala pc (naredba ping) i očitajte tu TTL vrijednost iz ispisa. Kojim putem idu paketi u jednom, a kojim putem u drugom slučaju? Ukratko objasnite zašto se međusobno razlikuju?

- Pri prvom slanju paketa TTL sa PC-a na Server TTL iznosi 60 , ali nakon prvog ping-a TTL je 61 te ostaje 61. To se dogodi zato što kod prvog slanja paketa defaultni usmjeritelj Servera je router6 ali je to duži put nego preko router7. Nakon prvog pinga dogodi se redirect te ostala komunikacija se odvija preko router7 i TTL ostaje 61.

- Paketi u jednom slučaju idu ovim putem :
  - Echo request: Pc->Router1->Router2->Router7->Server
  - Echo reply: Server->Router6->(redirect)Router7->Router2->Router1->Pc
- A u drugom slučaju :
  - Echo request: Server->Router7->Router2->Router1->Pc
  - Echo reply: Pc->Router1->Router2->Router7->Server

**Zadatak 25.** (*Topologija RIP/RIP.imn*) U emulatoru/simulatoru IMUNES, pomoću alata Wireshark snimite paket koji pripada protokolu RIP, proučite njegov sadržaj, te ga ukratko komentirajte.

```
▶ Frame 5: 166 bytes on wire (1328 bits), 166 bytes captured (1328 bits) on interface 0
▶ Ethernet II, Src: 42:00:aa:00:00:09 (42:00:aa:00:00:09), Dst: IPv4mcast_09 (01:00:5e:00:00:09)
▶ Internet Protocol Version 4, Src: 10.0.4.2, Dst: 224.0.0.9
▶ User Datagram Protocol, Src Port: 520, Dst Port: 520
▼ Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  ▼ IP Address: 0.0.0.0, Metric: 1
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 0.0.0.0
    Netmask: 0.0.0.0
    Next Hop: 10.0.4.1
    Metric: 1
  ▶ IP Address: 10.0.0.0, Metric: 2
  ▶ IP Address: 10.0.1.0, Metric: 2
  ▶ IP Address: 10.0.2.0, Metric: 3
  ▶ IP Address: 10.0.3.0, Metric: 3
  ▶ IP Address: 10.0.7.0, Metric: 1
```

Snimljeno na router7. Iz RIP odgovora možemo vidjeti da se radi o responseu- odgovoru. Možemo vidjeti verziju – RIPv2 te iz ostalih redaka možemo vidjeti broj skokova do određene podmreže. U svakom od tih redaka je vidljiva IP adresa , adresa podmaske mreže i ostalo.

**Zadatak 28.** Učitajte u IMUNES mrežu Ping/ping.imn. Pomoću alata Wireshark snimite proizvoljan TCP-promet koji pripada jednoj vezi te odredite segmente koji se razmjenjuju u fazama uspostave veze i raskida veze (za generiranje TCP-prometa iskoristite alat netcat).

- Skicirajte razmjenu segmenata za te dvije faze, uz navođenje korištenih TCP-zastavica.
- Za uhvaćeni promet, odredite koje se adrese i vrata koriste na izvoru i odredištu. Imaju li svi segmenti istu četvorku {izvorišna IP-adresa, izvorišna vrata, odredišna IP-adresa, odredišna vrata}?
- Skicirajte razmjenu nekoliko TCP-segmenata u fazi trajanja veze, uz navođenje korištenih TCP-zastavica.
- Utvrđite na koji se način koriste potvrde u TCP-vezi. Komentirajte.
- Snimite proizvoljan TCP-promet (koji pripada jednoj vezi) i utvrđite veličine prozora. Mijenja li se veličina prozora često u tijeku trajanja TCP-veze? Objasnite.

**Capturing from eth0 (on server)**

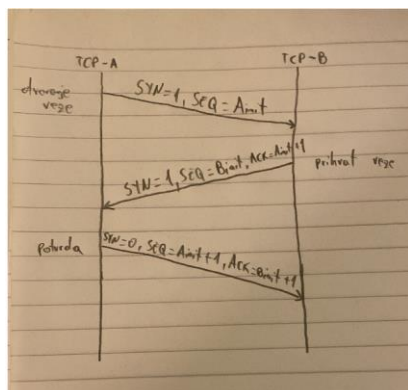
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
4	40.998112620	10.0.8.1	224.0.0.9	RIPv2	186	Response
5	60.939434428	10.0.0.21	10.0.8.10	TCP	74	12533 → 100 [SYN] Seq=
6	60.939511374	10.0.8.10	10.0.0.21	TCP	74	100 → 12533 [SYN, ACK]
7	60.939827652	10.0.0.21	10.0.8.10	TCP	66	12533 → 100 [ACK] Seq=
8	69.006094615	10.0.8.1	224.0.0.9	RIPv2	186	Response
9	72.010954696	fe80::ca4c:75ff:fe0...	ff02::9	RIPng	226	Command Response, Ver

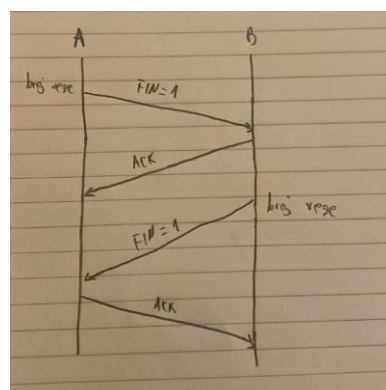
Frame 5: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0  
 Ethernet II, Src: Cisco\_00:00:0d (c8:4c:75:00:00:0d), Dst: IntelCor\_00:00:0e (68:05:ca:00:00:0e)  
 Internet Protocol Version 4, Src: 10.0.0.21, Dst: 10.0.8.10  
 Transmission Control Protocol, Src Port: 12533, Dst Port: 100, Seq: 0, Len: 0

### Uspostava veze

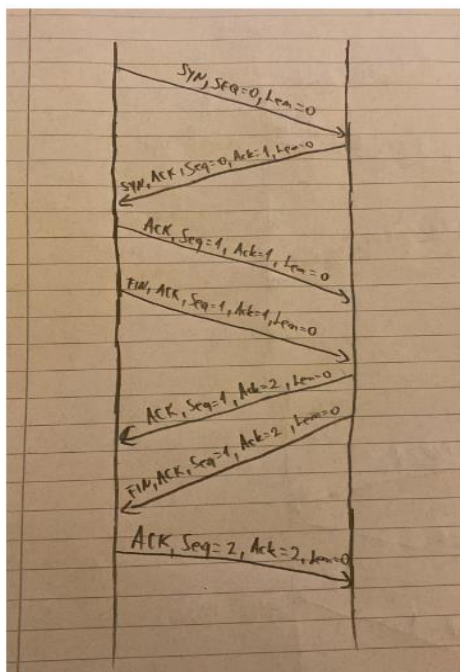


a)

### prekid veze



b) PC1 (10.0.0.21) sa vrata 12533 šalje na server(10.0.8.10) na vrata 100. Iste koristi i server samo u drugom smjeru.



c)

- d) Zastavica ACK(acknowledgment) je potvrda jeli paket uspješno primljen. Vrijednost ACK prilikom početnog uspostavljanja baze iznosi  $init+1$ . Kasnije ovisno o paketima te njihovim primitcima se ACK mijenja.
- e) Veličina prozora se nije puno mijenjala. Većinom je ona bila 1026. U par trenutaka se promijenila u slučaju da dođe do smetnji u komunikaciji onda se veličina prozora smanji te se kasnije postepeno vrati .

**Zadatak 29.** (Topologija Ping/ping.imn) Utvrdite mogu li se na jednom računalu pokrenuti dva procesa koji slušaju na istim vratima (npr., pokušajte dvaput pokrenuti alat netcat, istovremeno iz dvije konzole istog računala). Komentirajte.

- Nakon otvaranja Ping i pokretanja alata netcat spajao sam dva računala preko TCPa no samo jedno računalo se uspjelo spojiti te se koristilo. Nakon što sam ugasio proces koji se spojio te ponovno poslao spojio se drugi, što znači da se samo jedan proces može koristiti , ne mogu dva istovremeno.

**Zadatak 30.** (Topologija Ping/ping.imn) Ukoliko se alatu netcat ne zada protokol koji će koristiti, on podrazumijeva protokol TCP. Ponovite pokus iz prethodnog zadatka uz korištenje protokola UDP te komentirajte.

- Ovog puta su se oba računala spojila na netcat jer koristimo UDP, što nam govori da se korištenjem UDP-a može istovremeno slušati više procesa na istom portu.



**Zadatak 31.** (*Topologija Ping/ping.imn*) Primijetite da, iako su bitno različiti po svojstvima, protokoli UDP i TCP po funkcionalnosti spadaju u transportni sloj referentnog modela OSI. Objasnite zašto.

- Protokoli spadaju u transportni sloj jer prenose transportne jedinice od izvorišta na odredište. Razlike između dva protokola je što TCP daje potvrdu primitka paketa, a UDP ne. TCP se koristi kada nam je bitan prijenos bez gubitka informacija, a UDP kada nam bitna brzina prijenosa.

**Zadatak 32.** (*Topologija Ping/ping.imn*) Pokušajte prouzročiti gubitak TCP-segmenata. Na koji način možete utvrditi da je došlo do gubitaka? Možete li izazvati gubitke paketa bez mijenjanja karakteristika poveznica mreže?

- Gubitak TCP-segmenta se može dobiti na dva načina. Odspajanjem dijela mreže koji služi za komunikaciju između dva uređaja ili ako ne mijenjamo karakteristike poveznica mreže: smanjenjem prozora primatelja -I kako bi se izazvala retransmisija. Možemo utvrditi da je došlo do gubitka slanjem dvostruke potvrde(DUPACK).

**Zadatak 33.** (*Topologija Ping/ping.imn*) Pokušajte identificirati promet koji pripada jednoj TCP-vezi za vrijeme u kojem dolazi do gubitka segmenata. Utvrdite što se tada događa s potvrdama i veličinom prozora.

- Poslani segment se odbacuje te se javlja pogreška pošiljatelju. Pošiljatelj zatim ponovno šalje segment, retransmisija TCP-a. Primatelju se veličina prozora svakim primitkom segmenta smanjuje, I javlja pošiljatelju kada veličina dođe na 0. Kada se to dogodi pošiljatelj čeka na potvrdu od primatelja da može slati ponovno.