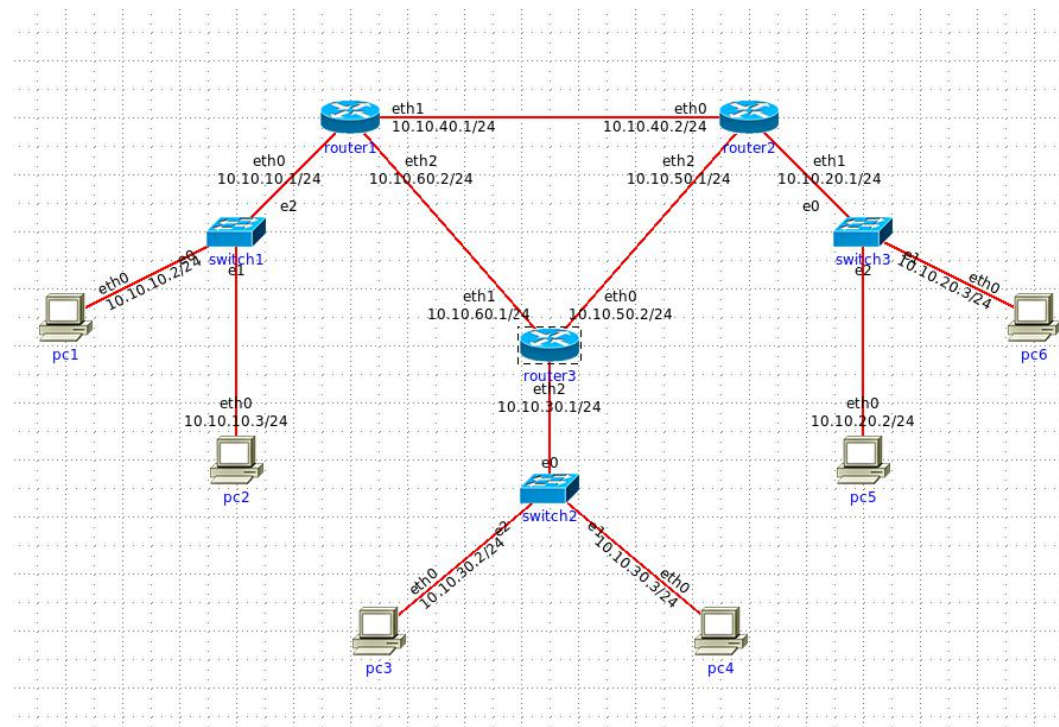


Labos 2

Komunikacijske mreže

22. Netstat -r dat će nam tablicu usmjeravanja.



pc1

Internet:	Destination	Gateway	Flags	Netif	Expire
	10.10.10.0/24	link#2	U	eth0	
	pc1	link#2	UHS	lo0	
	127.0.0.1	link#1	UH	lo0	

pc2

Internet:	Destination	Gateway	Flags	Netif	Expire
	10.10.10.0/24	link#2	U	eth0	
	pc2	link#2	UHS	lo0	
	127.0.0.1	link#1	UH	lo0	

pc3

Internet:	Destination	Gateway	Flags	Netif	Expire
	10.10.30.0/24	link#2	U	eth0	
	pc3	link#2	UHS	lo0	
	127.0.0.1	link#1	UH	lo0	

pc4

Internet:	Destination	Gateway	Flags	Netif	Expire
	10.10.30.0/24	link#2	U	eth0	
	pc4	link#2	UHS	lo0	
	127.0.0.1	link#1	UH	lo0	

```

Internet:
pc5 Destination      Gateway      Flags      Netif Expire
      10.10.20.0/24   link#2      U          eth0
      pc5             link#2      UHS        lo0
      127.0.0.1       link#1      UH         lo0

```

```

Internet:
pc6 Destination      Gateway      Flags      Netif Expire
      10.10.20.0/24   link#2      U          eth0
      pc6             link#2      UHS        lo0
      127.0.0.1       link#1      UH         lo0

```

```

Internet:
rt1 Destination      Gateway      Flags      Netif Expire
      10.10.10.0/24   link#2      U          eth0
      router1         link#2      UHS        lo0
      10.10.40.0/24   link#3      U          eth1
      10.10.40.1      link#3      UHS        lo0
      10.10.60.0/24   link#4      U          eth2
      10.10.60.2      link#4      UHS        lo0
      127.0.0.1       link#1      UH         lo0

```

```

Internet:
rt2 Destination      Gateway      Flags      Netif Expire
      10.10.20.0/24   link#3      U          eth1
      10.10.20.1      link#3      UHS        lo0
      10.10.40.0/24   link#2      U          eth0
      router2         link#2      UHS        lo0
      10.10.50.0/24   link#4      U          eth2
      10.10.50.1      link#4      UHS        lo0
      127.0.0.1       link#1      UH         lo0

```

```

Internet:
rt3 Destination      Gateway      Flags      Netif Expire
      10.10.30.0/24   link#4      U          eth2
      10.10.30.1      link#4      UHS        lo0
      10.10.50.0/24   link#2      U          eth0
      router3         link#2      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

```

23. U tablice usmjeritelja treba dodati nove vrijednosti koje će pakete poslati u petlju, tj. poslat će ih na krivi usmjeritelj te će usmjeritelji razmjenjivati pakete dok im TTL ne dođe do 0. Također, pc tablice moraju imati default rutu postavljenu na neposredno spojen usmjeritelj.

```

Static routes:
rt1 10.10.20.0/24 10.10.60.1
    10.10.30.0/24 10.10.40.2

```

```

Static routes:
rt2 10.10.10.0/24 10.10.50.2
    10.10.30.0/24 10.10.40.1

```

```

Static routes:
rt3 10.10.10.0/24 10.10.50.1
    10.10.20.0/24 10.10.60.2

```

Prilikom pinganja 10.10.20.2 (pc5) s pc1, echo request odlazi na beskonačnu rutu: rt1 -> rt3 -> rt1 -> rt3 itd. Naravno, svaka poruka ima svoj TTL, a nakon što istekne bit će odbačena. Ovdje je prikazana zadnja primljena poruka na rt1.

```

▼ Internet Protocol Version 4, Src: 10.10.10.2, Dst: 10.10.20.2
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 84
  Identification: 0x2e41 (11841)
  ▶ Flags: 0x0000
  ▶ Time to live: 1

```

24. Ping od pc do server vraća TTL 60, a obratno 61. Server kao default rutu ima 10.0.4.1 te stoga echo reply poruka ide prema **router6**. Međutim, router6 zna da paket za pc treba poslati na router7 (jer je to najkraća ruta) te to i čini. Dakle, ping od pc daje **TTL 60** jer echo reply poruka putuje preko 4 usmjeritelja (zadani TTL je 64):

server -> router6 -> router7 -> router2 -> router1 -> pc

Nakon prve poruke, router6 javlja do server da je optimalna ruta do pc preko router7, te će server dodati tu informaciju u svoju tablicu usmjeravanja i ubuduće slati direktno na router7.

Kad server šalje ping, pc šalje echo reply na svoju default rutu do 10.0.3.1. Router1 zatim šalje paket najkraćim putem do server preko 3 usmjeritelja, pa je **TTL 61**.

pc -> router1 -> router2 -> router7 -> pc

25. Snimka paketa RIP-a i jedne njegove stavke na eth0@router4:

```

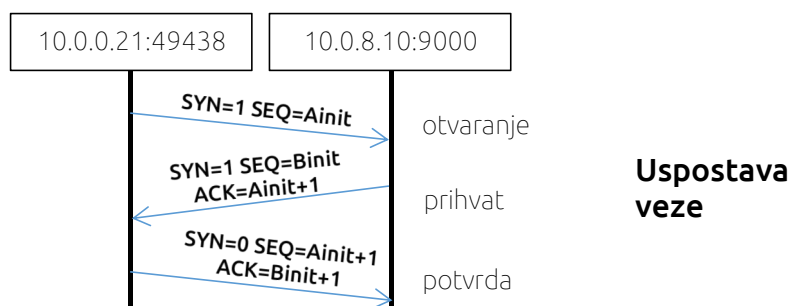
▼ Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  ▶ IP Address: 10.0.0.0, Metric: 2
  ▶ IP Address: 10.0.1.0, Metric: 3
  ▶ IP Address: 10.0.2.0, Metric: 2
  ▶ IP Address: 10.0.3.0, Metric: 2
  ▶ IP Address: 10.0.4.0, Metric: 1
  ▶ IP Address: 10.0.5.0, Metric: 1
  ▶ IP Address: 10.0.6.0, Metric: 2
  ▶ IP Address: 10.0.7.0, Metric: 3
  ▶ IP Address: 10.0.8.0, Metric: 2
  ▼ IP Address: 10.0.0.0, Metric: 2
    Address Family: IP (2)
    Route Tag: 0
    IP Address: 10.0.0.0
    Netmask: 255.255.255.0
    Next Hop: 0.0.0.0
    Metric: 2

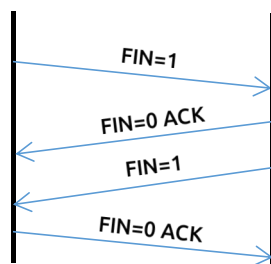
```

RIP paket sadrži informaciju o tome koliko su udaljene pojedine pod mreže. **Metric** govori o tome kolika je **udaljenost izražena u skokovima**. Tako na primjer od eth0@router4 do eth0@router0 (pod mreža 10.0.0.0) treba 2 skoka (router4 i router1). **Adress Family** govori o adresnom protokolu, u ovom slučaju 2, odnosno IP. **Route Tag** govori o tom da je RIP odredio udaljenost pod mreže, a **Next Hop** ukazuje na sljedeći skok.

28.

a)





**Prekid
veze**

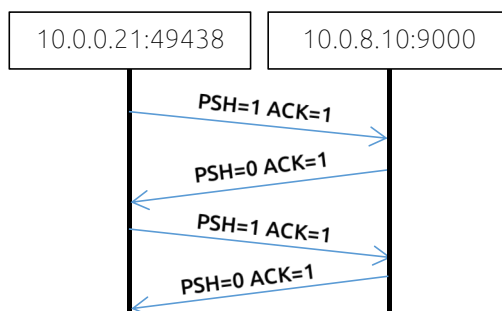
b) netcat od pc1 do server

```

> Internet Protocol Version 4, Src: 10.0.0.21, Dst: 10.0.8.10
  Transmission Control Protocol, Src Port: 49438, Dst Port: 9000, Seq: 0, Len: 0
    Source Port: 49438
    Destination Port: 9000
  
```

Kako se komunikacija odvija između dva uređaja na mreži, logično je da izvorišna IP adresa i vrata te odredišna IP adresa i vrata budu **isti** za sve pakete. Paket na ruti mijenja MAC adrese, no ne i IP adrese. Naravno, paketi koji se šalju od server do pc1 imaju obratno izvorište i odredište.

c)



**Uspostavljena
veza**

d) Potvrde služe za komunikaciju između pošiljatelja i primatelja koja se tiče broja **primljenih okteta**. Pošiljatelj na taj način zna da je točno određen broj okteta zaprimljen te može nastaviti slati nove blokove podataka ako su svi dosad primljeni.

e) Početni **Window size** bio je 65535. Prilikom slanja podatak taj broj se smanjuje jer pristižu novi okteti koji još nisu potvrđeni. Vidi se da se veličina smanjuje, no onda opet počne rasti jer je pošiljatelj završio sa slanjem. Pošiljatelj i primatelj stalno usklađuju veličinu prozora.

```

TCP      66 9000 → 49438 [ACK] Seq=1 Ack=2897 Win=62784
TCP      66 9000 → 49438 [ACK] Seq=1 Ack=5793 Win=59904
TCP      66 9000 → 49438 [ACK] Seq=1 Ack=6182 Win=65664
  
```

29. Dva procesa **ne mogu istovremeno** slušati na istim vratima. Oba slušanja se **moгу pokrenuti**, ali samo jedan proces prima podatke. Za slušanje sam koristio naredbu `nc -l 3000`, a za spajanje `nc <IP-adresa> 3000`. Smisao vrata jest da dva procesa na istom računalu mogu slušati različite tokove informacija, no svaki proces mora koristiti svoja vrata.

30. Dodavanjem zastavice `-u` koristi se UDP. Tim protokolom se također **ne može istovremeno** slušati na istim vratima, tj. samo jedan proces prima podatke. Razlika koju sam primijetio je da prilikom spajanja UDP ne provjerava je li itko sluša željeni port, već će uvijek otvoriti tok. Također, UDP ne provjerava je li slušanje na vratima prestalo.

31. Transmission Control Protocol (TCP) i User Datagram Protocol (UDP) pripadaju transportnom sloju modela OSI. Sama izvedba ovih protokola je različita, kao i njihove karakteristike, pa i primjene. No, njihova funkcionalnost je ista: **transport informacija** s kraja na kraj. Također, oba protokola šalju podatke na vrata.

32. Gubitak TCP segmenta najlakše se detektira **retransmisijom** tog segmenta. No moramo biti pažljivi jer se možda radi o zakašnjelom segmentu. Moguće je da potvrda o primitku stigne nakon što je istekao **timeout** pa je stoga segment opet poslan. Gubitak segmenta možemo izazvati smanjivanjem veličine prozora primatelja tako da prozor bude manji od segmenta.

33. Veličina prozora premalena je za segment i ostaje takva. Primatelj će pošiljatelju poslati poruku o grešci, a pošiljatelj neće znati drugo nego **ponovno poslati segment** iste veličine. Segment nikad neće stići do željenog primatelja.