

22. zadatak

PC1

Destination	Gateway	Flags	Netif	Expire
10.10.10.0/24	link#2	U	eth0	
10.10.10.21	link#2	UHS	lo0	
localhost	link#1	UH	lo0	

PC2

Destination	Gateway	Flags	Netif	Expire
10.10.10.0/24	link#2	U	eth0	
10.10.10.22	link#2	UHS	lo0	
localhost	link#1	UH	lo0	

PC3

Destination	Gateway	Flags	Netif	Expire
10.10.40.0/24	link#2	U	eth0	
10.10.40.21	link#2	UHS	lo0	
localhost	link#1	UH	lo0	

PC4

Destination	Gateway	Flags	Netif	Expire
10.10.40.0/24	link#2	U	eth0	
10.10.40.22	link#2	UHS	lo0	
localhost	link#1	UH	lo0	

PC5

Destination	Gateway	Flags	Netif	Expire
10.10.60.0/24	link#2	U	eth0	
10.10.60.22	link#2	UHS	lo0	
localhost	link#1	UH	lo0	

PC6

Destination	Gateway	Flags	Netif	Expire
10.10.60.0/24	link#2	U	eth0	
10.10.60.21	link#2	UHS	lo0	
localhost	link#1	UH	lo0	

ROUTER1

Destination	Gateway	Flags	Netif	Expire
10.10.10.0	link#2	UHS	lo0	
10.10.10.0/24	link#2	U	eth0	
10.10.20.0/24	link#3	U	eth1	
10.10.20.2	link#3	UHS	lo0	
10.10.30.0/24	link#4	U	eth2	
10.10.30.2	link#4	UHS	lo0	
10.10.40.0/24	10.10.20.1	UG1	eth1	
10.10.50.0/24	10.10.30.1	UG1	eth2	
10.10.60.0/24	10.10.30.1	UG1	eth2	
localhost	link#1	UH	lo0	

ROUTER2

Destination	Gateway	Flags	Netif	Expire
10.10.10.0/24	10.10.30.2	UG1	eth0	
10.10.20.0/24	10.10.30.2	UG1	eth0	
10.10.30.0/24	link#2	U	eth0	
10.10.30.1	link#2	UHS	lo0	
10.10.40.0/24	10.10.50.1	UG1	eth1	
10.10.50.0/24	link#3	U	eth1	
10.10.50.2	link#3	UHS	lo0	
10.10.60.0	link#4	UHS	lo0	
10.10.60.0/24	link#4	U	eth2	
localhost	link#1	UH	lo0	

ROUTER3

Destination	Gateway	Flags	Netif	Expire
10.10.10.0/24	10.10.20.2	UG1	eth0	
10.10.20.0/24	link#2	U	eth0	
10.10.20.1	link#2	UHS	lo0	
10.10.30.0/24	10.10.20.2	UG1	eth0	
10.10.40.0	link#4	UHS	lo0	
10.10.40.0/24	link#4	U	eth2	
10.10.50.0/24	link#3	U	eth1	
10.10.50.1	link#3	UHS	lo0	
10.10.60.0/24	10.10.50.2	UG1	eth1	
localhost	link#1	UH	lo0	

23. zadatak

Tablice za PC1 – PC6 ostaju iste

ROUTER1

Destination	Gateway	Netif
10.10.30.0	10.10.20.1	eth1
10.10.40.0	10.10.30.1	eth2

ROUTER2

Destination	Gateway	Netif
10.10.10.0	10.10.50.1	eth1
10.10.40.0	10.10.30.2	eth0

ROUTER3

Destination	Gateway	Netif
10.10.10.0	10.10.50.2	eth1
10.10.60.0	10.10.20.2	eth0

Kada pošaljemo paket on se vrti u petlji sve dok ne istekne TTL (koji je inicijalno postavljen na 64). Nakon toga se paket odbaci.

24. zadatak

Ping servera sa računala PC1 – Inicijalno je ttl=60 i poruka se šalje preko router6, ali za sve sljedeće zahtjeve iznosi 61 i šalje se preko router7 jer je to kraći put.

Ping računala PC1 sa servera – Inicijalno paket ide preko router6 i tada ttl iznosi 40, ali za sve naknadne zahtjeve ide preko router7 i ttl iznosi 61

Traceroute od PC1 do server:

```
1 10.0.3.1 (10.0.3.1) 39.046 ms 41.143 ms 38.242 ms
2 10.0.0.2 (10.0.0.2) 60.204 ms 60.596 ms 59.034 ms
3 10.0.7.2 (10.0.7.2) 79.713 ms 80.938 ms 79.722 ms
4 10.0.4.10 (10.0.4.10) 120.835 ms 119.157 ms 119.599 ms
```

Traceroute od server do PC1:

```
1 10.0.4.2 (10.0.4.2) 37.631 ms 39.993 ms 39.952 ms
2 10.0.7.1 (10.0.7.1) 59.967 ms 58.960 ms 60.112 ms
3 10.0.0.1 (10.0.0.1) 81.652 ms 77.900 ms 80.655 ms
4 10.0.3.20 (10.0.3.20) 120.468 ms 119.869 ms 120.297 ms
```

25. zadatak

RIP Request:

```
▼ Routing Information Protocol
  Command: Request (1)
  Version: RIPv2 (2)
  ▶ Address not specified, Metric: 16
```

RIP Response:

```
▼ Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  ▶ IP Address: 10.0.0.0, Metric: 1
  ▶ IP Address: 10.0.1.0, Metric: 2
  ▶ IP Address: 10.0.2.0, Metric: 1
  ▶ IP Address: 10.0.4.0, Metric: 1
  ▶ IP Address: 10.0.6.0, Metric: 2
  ▶ IP Address: 10.0.7.0, Metric: 2
  ▶ IP Address: 10.0.9.0, Metric: 2
```

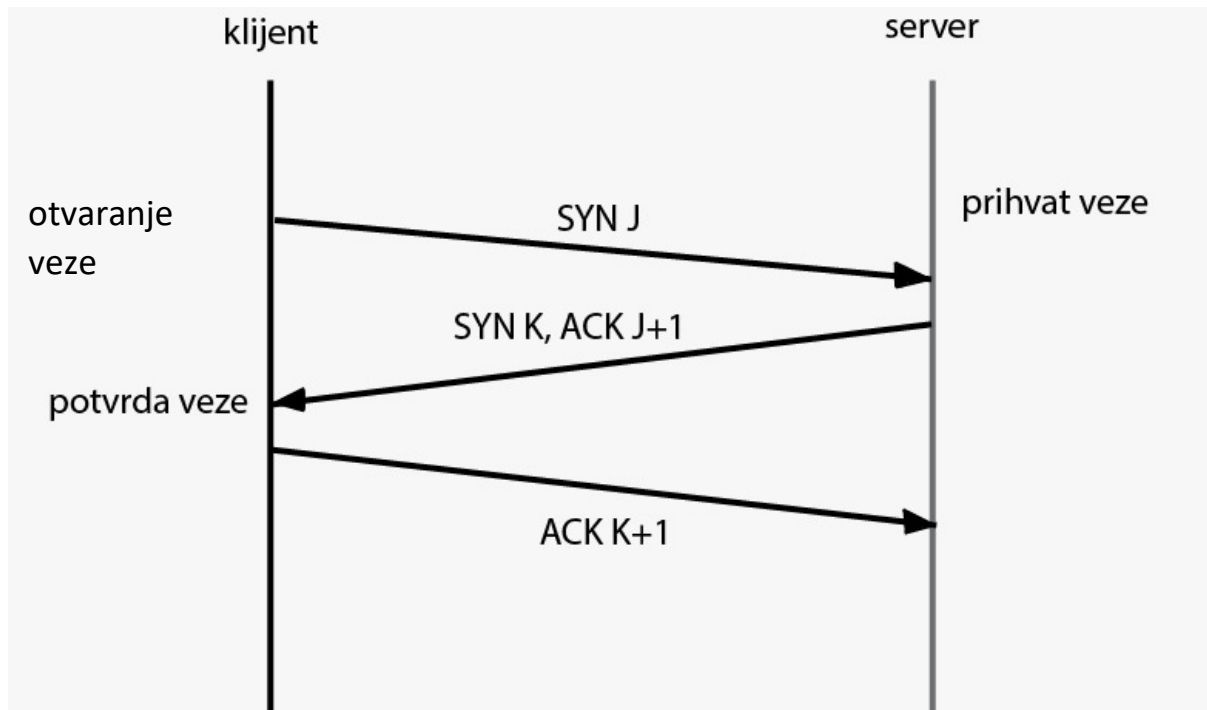
RIP protokol u sebi sadrži popis IP adresa različitih pod mreža, a 'Metric' atribut nam govori koliko je skokova potrebno do tih mreža.

28. zadatak

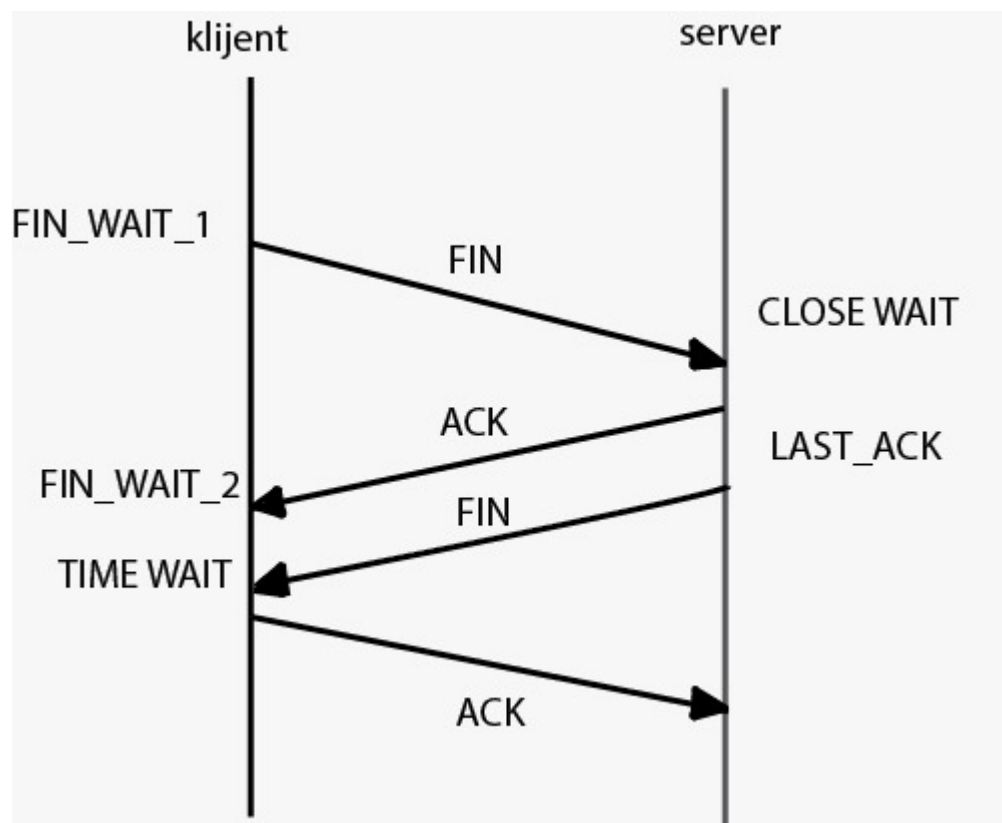
TCP promet između pc1 i server.

a)

Uspostava veze:



Prekid veze:

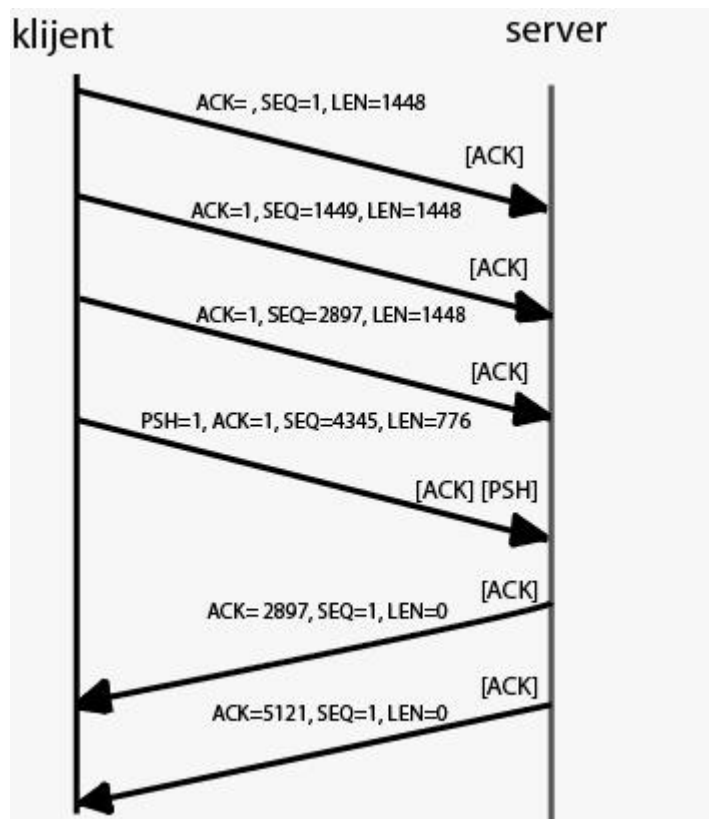


b) Svi segmenti nemaju iste četvorke jer se mijenja koji par IP adrese i vrata su pošiljalac, a koja primatelj.

{10.0.0.21, 59480, 10.0.8.10, 100} i {10.0.8.10, 100, 10.0.0.21, 59480} su četvorke koje se pojavljuju u segmentima. Tijekom prekida veze umjesto vrata 59480 koriste se vrata 47439.

c)

TCP promet:



d) Potvrde nam pokazuju koliko je okteta zaprimljeno. Ako pošiljalac šalje segment sa Len=16, primatelj će odgovoriti sa Ack=16 + 1=17 za uspostavu veze kako bi potvrdio da je primio 16 okteta te da je veza uspostavljena. Ako se nakon toga opet šalje segment sa Len=16, primatelj ovaj put odgovara za Ack=33.

e) Veličina prozora TCP segmenta je iznosila 1026. Ta veličina se ne mijenja tijekom trajanja TCP veze zato jer je primatelj unaprijed odredio optimalnu brzinu kojom može primiti podatke.

29. zadatak

Mogu se pokrenuti dvije instance slušanja preko netcat komande na jednom računalu te na istim vratima. To je moguće jer vrata služe kako bi različiti servisi na jednom računalu mogli dobiti različite podatke.

30. zadatak

Isti rezultat dobijemo neovisno o tome jel koristimo TCP ili UDP protokol premda oba protkola pripadaju transportnom sloju te na isti način koriste vrata kako bi znali kojem procesu trebaju dostaviti podatke.

31. zadatak

Oba protokola pripadaju transportnoj sloju jer im je svrha ista te na isti način koriste vrata za dostavu podataka. Unatoč tome postoji neke razlike u načinu prijenosa podataka. Npr. UDP ne čuva redoslijed dok TCP čuva.

32. zadatak

Možemo zaključiti da je došlo do gubitka TCP segmenata kada uočimo ponovno slanje tih segmenata. To možemo prouzročiti bez mijenjanja karakteristika poveznica mreže tako da smanjimo veličinu pomičnog prozora.

33. zadatak

Segment koji je poslan je prevelik za veličinu prozora. Taj segment se odbacuje te primatelj javlja grešku pošiljatelju. Nakon toga pošiljatelj ponovno šalje isti segment, ali će ponovno biti odbačen zbog neusklađenosti veličine prozora.