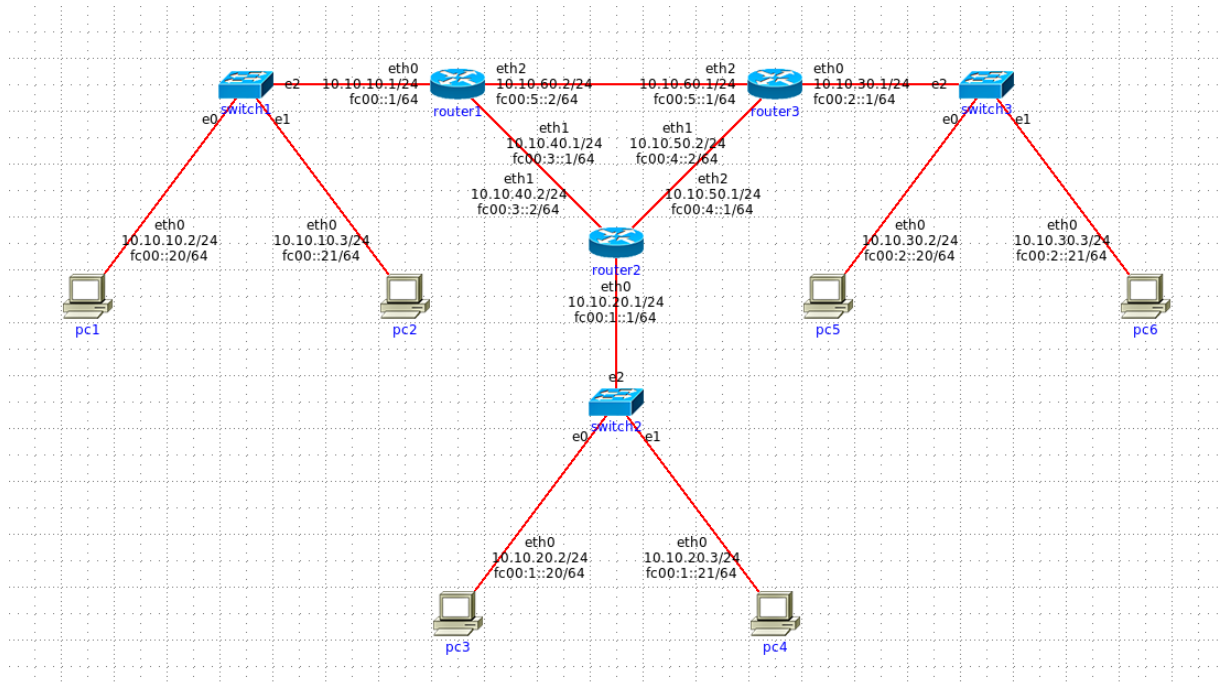


Zadatak 22. (Vlastito izrađena topologija)

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, 10.10.60.0/24. Ispišite tablice usmjeravanja svih računala i usmjeritelja.



Ovo lako doznate tako da stavite sve rutere na static i pokrenete *netstat -r* za svaki entitet.

PC1

Odredište	Sljedeći skok	Mrežno sučelje
10.10.10.0/24	link#2	eth0
10.10.10.2/24	link#2	eth0
localhost	link#1	lo0

PC2

Odredište	Sljedeći skok	Mrežno sučelje
10.10.10.0/24	link#2	eth0
10.10.10.3/24	link#2	eth0
localhost	link#1	lo0

PC3

Odredište	Sljedeći skok	Mrežno sučelje
10.10.20.0/24	link#2	eth0
10.10.20.2/24	link#2	eth0
localhost	link#1	lo0

PC4

Odredište	Sljedeći skok	Mrežno sučelje
10.10.20.0/24	link#2	eth0
10.10.20.3/24	link#2	eth0
localhost	link#1	lo0

PC5

Odredište	Sljedeći skok	Mrežno sučelje
10.10.30.0/24	link#2	eth0
10.10.30.2/24	link#2	eth0
localhost	link#1	lo0

PC6

Odredište	Sljedeći skok	Mrežno sučelje
10.10.30.0/24	link#2	eth0
10.10.30.3/24	link#2	eth0
localhost	link#1	lo0

Router1

Odredište	Sljedeći skok	Mrežno sučelje
10.10.10.0/24	link#2	eth0
10.10.10.1	link#2	eth0
10.10.40.0/24	link#3	eth1
10.10.40.1	link#3	eth1
10.10.60.0/24	link#4	eth2
10.10.60.1	link#4	eth2
localhost	link#1	lo0

Router2

Odredište	Sljedeći skok	Mrežno sučelje
10.10.20.0/24	link#2	eth0
10.10.20.1	link#2	eth0
10.10.40.0/24	link#3	eth1
10.10.40.2	link#3	eth1
10.10.50.0/24	link#4	eth2
10.10.50.1	link#4	eth2
localhost	link#1	lo0

Router3

Odredište	Sljedeći skok	Mrežno sučelje
10.10.30.0/24	link#2	eth0
10.10.30.1	link#2	eth0
10.10.50.0/24	link#3	eth1
10.10.50.2	link#3	eth1
10.10.60.0/24	link#4	eth2
10.10.60.1	link#4	eth2
localhost	link#1	lo0

Zadatak 23. (Vlastito izrađena topologija)

Isto kao i gore, samo treba konfigurirati tako da dođe do petlje u usmjeravanju. 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.

PC tablice ostaju iste. Tablicama usmjeritelja treba dodati parove Odredište-Sljedeći skok:

Router1

Odredište	Sljedeći skok	Mrežno sučelje
10.10.20.0/24	10.10.60.1	eth1
10.10.30.0/24	10.10.40.2	eth2

Router2

Odredište	Sljedeći skok	Mrežno sučelje
10.10.10.0/24	10.10.50.2	eth1
10.10.30.0/24	10.10.40.1	eth2

Router3

Odredište	Sljedeći skok	Mrežno sučelje
10.10.10.0/24	10.10.50.1	eth2
10.10.20.0/24	10.10.60.2	eth1

U alatu *Wireshark* se i datagrami i ICMP poruke uhvate u petlji pa nakon što im istekne TTL budu odbačeni.

Zadatak 24. (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.

Nakon pinganja *servera* s *pc-a*, TTL=60. Nakon pinganja *pc-a* sa *servera*, TTL=61. U prvom slučaju paketi s *router2* krenu prema *router7*, a u drugom slučaju *sw2* prosljeđuje podatke *router6*. U prvom slučaju dobivamo TTL=60 zato što *server pc*-u šalje povratnu poruku duljim putem jer mu je defaultni usmjeritelj *router6*. U drugom slučaju dobivamo TTL=61 zato što *pc serveru* šalje povratnu poruku kraćim putem jer je *router7* defaultni usmjeritelj *routera2*.

Zadatak 25. (RIP/RIP.imn)

U emulatoru/simulatoru IMUNES, pomoću alata Wireshark snimite paket koji pripada protokolu RIP, proučite njegov sadržaj, te ga ukratko komentirajte.

Sadržaj:

Command: Response (govori nam da je ovo RIP odgovor)

Version: RIPv2 (govori o inačici protokola)

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

U ovom odgovoru je za svaku IP adresu, tj. mrežu, jer ovo nisu direktno IP adrese sučelja, priložena metrika, tj. broj skokova do te mreže.

Zadatak 28. (Ping/ping.imn)

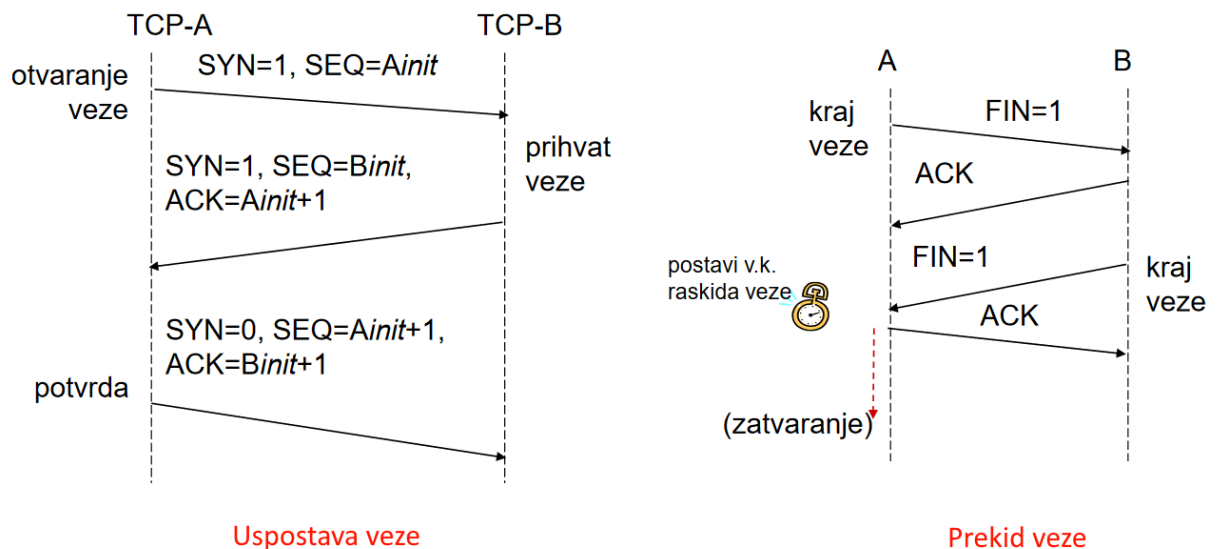
Pomoću alata *Wireshark* snimite proizvoljan TCP-promet koji pripada jednoj vezi te odredite segmente koji se razmjenjuju u fazama uspostave veze i raskida veze (za generiranje TCP-prometa iskoristite alata *netcat*).

- Skicirajte razmjenu segmenata za te dvije faze, uz navođenje korištenih TCP-zastavica
- Za uhvaćeni promet, odredite koje se adrese i vrata koriste na izvorištu 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.
- Utvrdite na koji se način koriste potvrde u TCP-vezi. Komentirajte.
- Snimite proizvoljan TCP-promet (koji pripada jednoj vezi) i utvrdite veličine prozora. Mijenja li se veličina prozora često u tijeku trajanja TCP-veze? Objasnite.

Na host6 pokrećem *nc -l 100*.

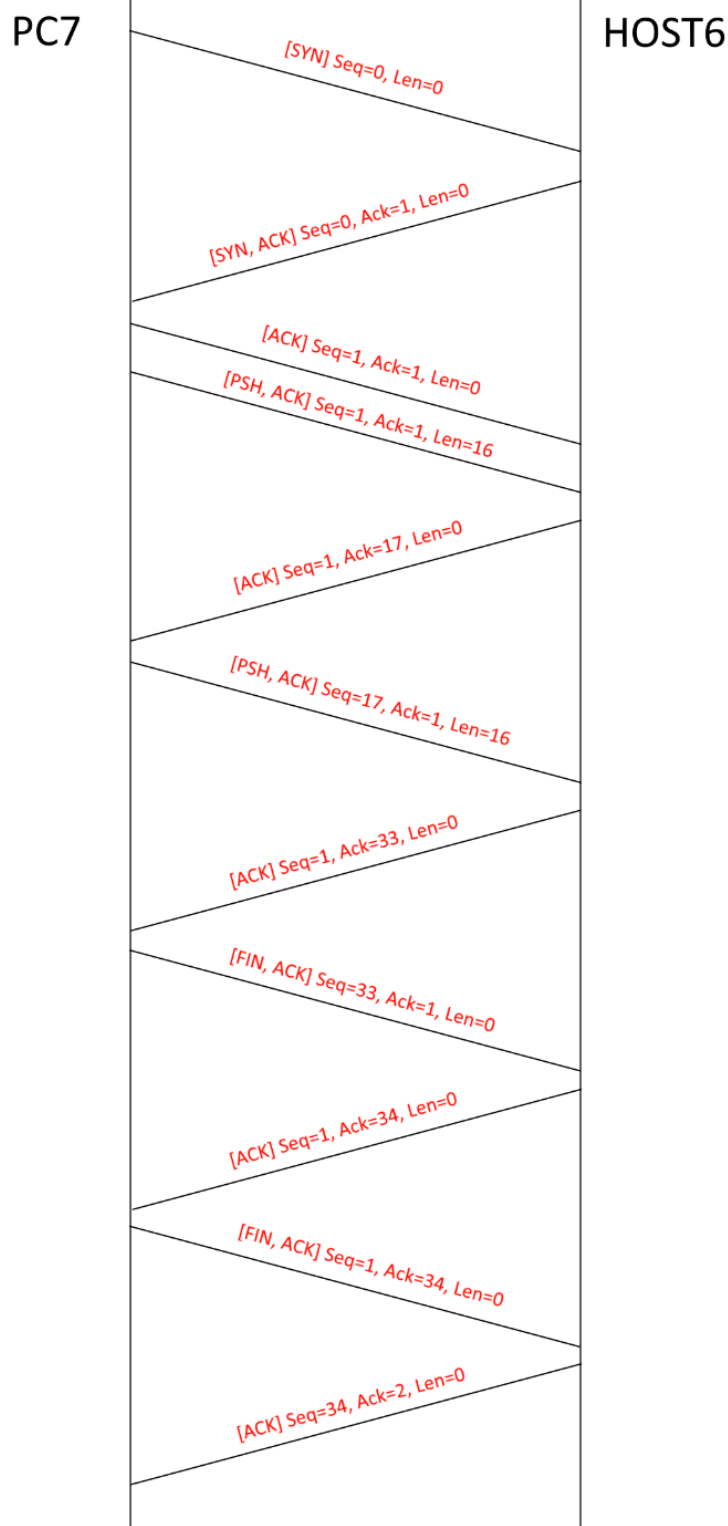
Na pc7 pokrećem *nc 10.0.8.2 100*.

a)



- Pc7 šalje host6 preko svog porta 58256 na port 100 TCP promet. U svim fazama, dakle sinkronizacija, slanje prometa i prekid veze, sve uređene četvorke su iste.

c)



- d) Potvrde govore primatelju koliko je okteta zaprimljeno. Slična funkcionalnost kao i broj okvira, gotovo analogna offsetu.
- e) Snimljen promet imao je veličine prozora 1026. Tijekom slanja veličina prozora se nije mijenjala. To je iz razloga što nakon što je primatelj definirao koliko brzo može primiti podatke, ukoliko nema zagušenja u mreži, nema razloga da se veličina prozora mijenja jer je na optimalnoj vrijednosti.

Zadatak 29. (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.

To je moguće. Smisao vrata je da različiti servisi na jednom računalu mogu dobiti različite podatke. Ne postoji razlog zašto više servisa ne bi moglo slušati na istim vratima.

Zadatak 30. (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.

Moguće je slušati i u slučaju UDP-a. Isti razlog kao i gore.

Zadatak 31. (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.

U ulozi transportnog sloja ne nalazi se zahtjev na koji način će se segmenti u transportnom sloju prenositi. Razlika TCP-a i UDP-a nalazi se u ciljevima koje oni ostvaruju, i donekle u sadržaju, no princip odašiljanja na vrata je isti, stoga su oboje po OSI u transportnom sloju.

Zadatak 32. (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?

Da je došlo do gubitka segmenta možemo zaključiti da se potpuni isti ponovno šalje. Ovo ne govori izričito da se segment izgubio, ali je tako moguće detektirati gubljenje segmenta.

Mogu, smanjivanjem veličine prozora primatelja.

Zadatak 33. (Ping/ping.imn)

Pokušajte identificirati promet koji pripada jednoj TCP-vezi za vrijeme u kojem dolazi do gubitaka segmenata. Utvrdite što se tada događa s potvrdama i veličinom prozora.

Poslani segment bude prevelik da ga prozor primatelja može pohraniti. Tada se segment odbacuje, te se javlja pogreška pošiljatelju. Pošiljatelj će zatim ponovo poslati segment (retransmisija TCP-a), no kako će segment zbog neusklađenosti veličine prozora ponovo biti prevelik, segment se ponovo neće pohraniti.