



SVEUČILIŠTE U ZAGREBU  
Fakultet  
elektrotehnike i  
računarstva

**Preddiplomski studij**

**Računarstvo**

Vedran Podobnik, Ognjen Dobrijević,  
Tomislav Grgić, Krunoslav Ivešić

# **Internetski protokoli u primjeni**

Inačica udžbenika v1.4

**Ak.g. 2020./2021.**

## Predgovor

Udžbenik „Internetski protokoli u primjeni“ bavi se komunikacijskim mrežama, a posebice mrežnim arhitekturama i komunikacijskim protokolima s naglaskom na Internetu, pri čemu obrađuje odabrane praktične primjere koji su usredotočeni na relevantne teme iz domene svakodnevnog korištenja Interneta. Ovaj udžbenik omogućava čitateljima aktivan i praktičan pristup učenju komunikacijskih mreža budući da je pisan jednostavnim stilom te nadopunjen mnoštvom zanimljivih pratećih primjera.

Iako je prvenstvena namjena udžbenika služiti studentima preddiplomskog studija Računarstva na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu kao praktični komplement udžbeniku „Komunikacijske mreže“ (autori: Lovrek, Matijašević, Ježić, Jevtić), on može poslužiti i kao samostalna literatura pojedincima koji žele naučiti osnove o umrežavanju računala te funkcioniranju Interneta. Udžbenik „Internetski protokoli u primjeni“ može, dakle, poslužiti studentima da prouče kako su praktično u stvarnim mrežama izvedeni modeli, koncepti i protokoli o kojima uče na predavanjima, ali isto tako i bilo kojem drugom zainteresiranom čitatelju koji želi postati bolje upoznat korisnik Interneta.

Odabrani praktični primjeri uključuju eksperimentalnu izvedbu određenog realnog scenarija u mrežnom emulatoru, uz istovremeno „snimanje prometa“ koji se generira kao posljedica izvođenja tog scenarija. Udžbenik nadalje daje upute kako u poslije-eksperimentalnoj fazi analizirati „snimljeni promet“ te kroz tu analizu objašnjava temeljne mrežne modele, koncepte i protokole. Na ovaj način čitatelji uče o komunikacijskim mrežama na najbolji mogući način – promatrajući kako one zaista funkcioniraju. Tako će, primjerice, čitateljima biti objašnjeno što se sve događa u mreži ako oni putem web-preglednika na svom računalu pristupaju web-stranici Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu <http://www.fer.unizg.hr> ili pak šalju elektroničku poštu prijatelju koji se nalazi u Australiji.

Bez obzira koje su motivacije čitatelja da uče o komunikacijskim mrežama, ovaj udžbenik će im omogućiti da „uče radeći“, umjesto da samo „uče čitajući“. Dio materijala sadržan u ovom udžbeniku pokazao se kao vrlo koristan sadržaj preddiplomskih i diplomskih sveučilišnih kolegija s tematikom komunikacijskih mreža te internetskih mehanizama, ali isto tako je uspješno korišten prilikom kraćih fokusiranih radionica namijenjenih iskusnijim profesionalcima iz industrije. Stoga se nadamo da ćete u udžbeniku pronaći mnogo korisnog i zanimljivog sadržaja, bez obzira na Vaše godine ili prijašnja iskustva.

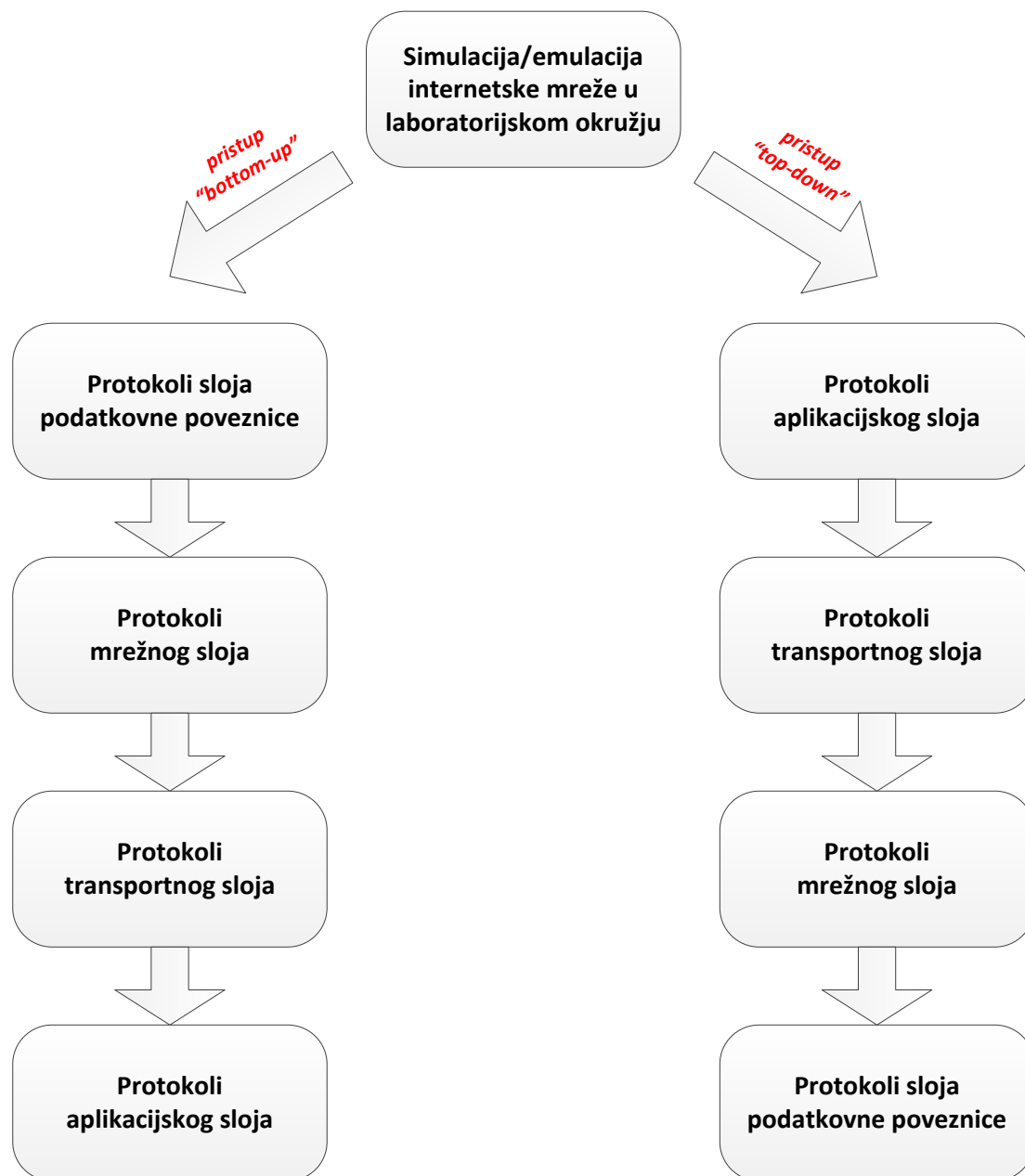
Autori se zahvaljuju dr. sc. Ani Petrić, dr. sc. Vanji Smailoviću i Darianu Škarici, dipl. ing. na recenziji rukopisa, razvojnom timu emulatora/simulatora IMUNES – prof. dr. sc. Miljenku Mikucu, dr. sc. Ani Kukec, dr. sc. Valteru Vasiću, Marku Zecu, dipl. ing. i Denisu Salopeku, mag. ing. – na savjetima i pomoći prilikom izrade primjera koji su korišteni u ovom dijelu radne inačice udžbenika. Nadalje, autori se zahvaljuju studentima Luki Božiću, Nenadu Petehu, Veranu Pokorniću, Igoru Sambolec i Ivanu Šemanjskom na pomoći prilikom izrade poglavlja o simulaciji/emulaciji internetske mreže u laboratorijskom okružju.

*Autori*

## Kako koristiti ovu knjigu (i povezanu programsku podršku)

### Organizacija teksta

Udžbenik „Internetski protokoli u primjeni“ je organiziran u pet glavnih poglavlja: sastoji se od uvodnog poglavlja, koje objašnjava na koji način se kreiraju simulacije/emulacije realnih mrežnih scenarija uz pomoć alata IMUNES, te četiri poglavlja s praktičnim primjerima koja obuhvaćaju različite slojeve protokolnog složaja (sloj podatkovne poveznice, mrežni sloj, transportni sloj te aplikacijski sloj).



Slika: Mogući pristupi u korištenju udžbenika

Odabrani praktični primjeri su organizirani na način da se najprije bave temama vezanim uz niže slojeve te se uspinju po protokolnom složaju (tzv. pristup „bottom-up“), ali je posebna pažnja posvećena činjenici da ih bude moguće koristiti i prilikom alternativne metodologije učenja o komunikacijskim mrežama, kada se najprije započinje s proučavanjem mehanizama na aplikacijskom sloju, a završava na dnu protokolnog složaja slojem podatkovne poveznice (tzv. pristup „top-down“). Udžbenik stoga ima izraženu modularnu strukturu gdje se svako od pet poglavlja može koristiti samostalno i ne ovisi o ostalim poglavljima, ali je čitateljima koji nemaju nikakvog prethodnog iskustva s komunikacijskim mrežama preporučeno da udžbenik koriste kao cjelinu, bilo koristeći metodologiju „bottom-up“ ili „top-down“ (slika „Mogući pristupi u korištenju udžbenika“). Ukoliko čitatelj ne želi samostalno izrađivati eksperimente koji simuliraju određene mrežne scenarije, već se odlučio koristiti samo one odabrane primjere koji su uključeni u ovaj udžbenik, tada nije potrebno proučavati uvodno poglavlje koje tumači na koji način se kreiraju simulacije/emulacije realnih komunikacijskih scenarija uz pomoć alata IMUNES. Ipak, korisno je proučiti početak uvodnog poglavlja koje objašnjava kako se alat IMUNES instalira i pokreće.



*Slika: Jedinstvena struktura poglavlja koja pokrivaju različite slojeve protokolnog složaja*

Poglavlja s praktičnim primjerima koja pokrivaju različite slojeve protokolnog složaja imaju jednaku strukturu (slika „Jedinstvena struktura poglavlja koja pokrivaju različite slojeve protokolnog složaja“): najprije su objašnjeni simulacijski scenariji pomoću kojih će se tumačiti protokoli promatranog sloja, zatim su dane definicije i tumačenja korištenih pojmova, koncepata, protokola i alata, nakon čega su opisani eksperimenti koji se izvršavaju te definirani zadaci koji se rješavaju u kontekstu izvršenih eksperimenata. Svako poglavlje završava s pet pitanja o širem aspektu promatranog sloja te kratkom bibliografijom najvažnijih izvora povezanih sa sadržajem poglavlja.

Udžbenik završava s tri kratka dodatna poglavlja: poglavlje s odgovorima na pitanja koja su postavljena u poglavljima s praktičnim primjerima, poglavlje s indeksom najvažnijih pojmova te prilog u kojem su navedene osnovne naredbe UNIX-ljuske potrebne čitateljima prilikom izrade vlastitih primjera u alatu IMUNES.

### *Alat IMUNES*

IMUNES (*Integrated MUltiprotocol Network Emulator/Simulator*) je alat koji se može koristiti i kao simulator i kao emulator internetskih mreža, a temelji se na operacijskom sustavu FreeBSD. Više detalja o samom alatu, kao i načinu na koji se koristi, dostupno je u dokumentu *Vodič za alat IMUNES* (dostupan za dohvat putem poveznice: <http://www.imunes.net>).

Simulacija/emulacija različitih komunikacijskih scenarija korištenih u ovome udžbeniku izvedena je konfiguracijom, izvršavanjem te analizom eksperimenata u alatu IMUNES. Alat IMUNES, u kojeg su već ugrađeni eksperimenti koji se koriste u uvome udžbeniku, može se dohvatiti putem poveznice <http://www.imunes.net/download.html>. Upute kako instalirati i pokrenuti alat IMUNES nalaze se na početku uvodnog poglavlja. Datoteke s pripremljenim eksperimentima nalaze se u direktoriju `/root/imunes-examples/`, do kojeg se dolazi naredbom „`# cd /root/imunes-examples/`“.

## **Kontaktirajte autore**

Nadamo se da će ovaj udžbenik čitateljima omogućiti praktičan i zabavan uvod u svijet komunikacije među računalima. Željeli bismo čuti Vaša iskustva o korištenju udžbenika i povezanih praktičnih vježbi u sklopu formalnog obrazovnog procesa (primjerice, kao literatura za kolegij na sveučilišnom studiju) ili kao pomagala tijekom samostalnog učenja.

Pozivamo Vas da sve komentare i sugestije kako ovaj udžbenik učiniti još boljim, kao i uočene pogreške, pošaljete autorima na sljedeću adresu:

***Sveučilište u Zagrebu  
Fakultet elektrotehnike i računarstva  
Unska 3  
10000 Zagreb***

ili na adresu elektroničke pošte:

***vedran.podobnik@fer.hr***

## Sadržaj

<b>1</b>	<b>Simulacija/emulacija internetske mreže u laboratorijskom okružju .....</b>	<b>9</b>
1.1	Osnovne upute za korištenje simulatora/emulatora IMUNES .....	9
1.1.1	Upute za instalaciju IMUNES-a.....	9
1.1.2	Mrežno okruženje za konfiguraciju internetskih usluga: korištenje gotovih modela mreže ili izgradnja vlastitih.....	13
1.1.3	Uređivači teksta.....	17
1.1.4	Pristup datotečnom sustavu pojedinog čvora .....	17
1.1.5	Kopiranje datoteka unutar OS-a FreeBSD .....	18
1.1.6	Kopiranje datoteka na pojedini čvor u pokrenutom IMUNES eksperimentu ....	18
1.2	Konfiguriranje DHCP-poslužitelja .....	19
1.2.1	Konfiguriranje i pokretanje DHCP-poslužitelja .....	19
1.2.2	Primjer dodjele IP-adrese pomoću DHCP-poslužitelja.....	22
1.3	Konfiguriranje DNS-poslužitelja.....	23
1.3.1	Korišteni model mreže i koraci konfiguracije DNS-poslužitelja .....	24
1.3.2	Primjer korištenja konfiguriranog sustava DNS .....	29
1.4	Konfiguriranje poslužitelja elektroničke pošte.....	30
1.4.1	Opis mrežnog okružja za konfiguraciju .....	31
1.4.2	Konfiguriranje i pokretanje SMTP-poslužitelja.....	32
1.4.3	Primjer slanja elektroničke pošte .....	39
1.5	Konfiguriranje HTTP-poslužitelja.....	42
1.5.1	Konfiguriranje i pokretanje HTTP-poslužitelja .....	42
1.5.2	Primjer spajanja na HTTP-poslužitelj .....	44
<b>2</b>	<b>Protokoli sloja podatkovne poveznice .....</b>	<b>46</b>
2.1	Konfiguracija eksperimenta.....	46
2.2	Objašnjenja korištenih pojmova, koncepata, protokola i alata .....	47
2.2.1	Standard Ethernet .....	47
2.2.2	Alat Wireshark .....	47
2.3	Eksperimenti i zadaci.....	48
2.4	Pitanja .....	48
2.5	Izvori.....	49
<b>3</b>	<b>Protokoli mrežnog sloja.....</b>	<b>50</b>
3.1	Konfiguracija eksperimenta.....	50

3.2	Objašnjenja korištenih pojmova, koncepata, protokola i alata .....	54
3.2.1	Protokol IP i pomoći protokoli .....	54
3.2.2	Alat ping .....	54
3.2.3	Alat traceroute .....	59
3.2.4	Protokoli usmjeravanja .....	61
3.3	Eksperimenti i zadaci .....	68
3.4	Pitanja .....	73
3.5	Izvori .....	74
<b>4</b>	<b>Protokoli transportnog sloja .....</b>	<b>75</b>
4.1	Konfiguracija eksperimenta .....	75
4.2	Objašnjenja korištenih pojmova, koncepata, protokola i alata .....	75
4.2.1	Protokol UDP .....	76
4.2.2	Protokol TCP .....	76
4.2.3	Alat netcat .....	77
4.3	Eksperimenti i zadaci .....	80
4.4	Pitanja .....	80
4.5	Izvori .....	81
<b>5</b>	<b>Protokoli aplikacijskog sloja .....</b>	<b>82</b>
5.1	Konfiguracija eksperimenta .....	82
5.2	Objašnjenja korištenih pojmova, koncepata, protokola i alata .....	83
5.2.1	Protokol DHCP .....	84
5.2.2	Protokol DNS .....	85
5.2.3	Protokoli elektroničke pošte .....	89
5.2.4	Protokol HTTP .....	90
5.3	Eksperimenti i zadaci .....	92
5.4	Pitanja .....	95
5.5	Izvori .....	95
<b>6</b>	<b>Odgovori na pitanja .....</b>	<b>96</b>
<b>7</b>	<b>Indeks pojmova .....</b>	<b>97</b>
	<b>Prilog A: naredbe UNIX-ljuske .....</b>	<b>98</b>



## 1 Simulacija/emulacija internetske mreže u laboratorijskom okružju

Cilj ovog poglavlja je objasniti postupke konfiguracije aplikacijskih poslužitelja za internetske usluge temeljene na protokolima DHCP, DNS, SMTP, POP i HTTP. Svi postupci realizirani su uz korištenje mrežnog simulatora/emulatora IMUNES.

U poglavlju 1.1 opisane su upute za instalaciju simulatora/emulatora IMUNES te osnovne upute za njegovo korištenje. Nakon toga, opisani su postupci konfiguracije sljedećih internetskih usluga: usluge DHCP (poglavljje 1.2), usluge DNS (poglavljje 1.3), usluge elektroničke pošte koja koristi protokole SMTP i POP (poglavljje 1.4) te usluge pristupa web-sadržaju koja koristi protokol HTTP (poglavljje 1.5).

### 1.1 Osnovne upute za korištenje simulatora/emulatora IMUNES

IMUNES (*Integrated Multiprotocol Network Emulator/Simulator*) je alat koji se može koristiti i kao simulator i kao emulator internetskih mreža, a temelji se na operacijskom sustavu FreeBSD. Ovdje su objašnjene samo one funkcionalnosti alata koje su potrebne za pojedine konfiguracijske postupke. Više detalja o samom alatu, kao i načinu na koji se koristi, dostupno je u dokumentu *Vodič za alat IMUNES*<sup>1</sup>.

#### 1.1.1 Upute za instalaciju IMUNES-a

Simulator/emulator IMUNES nužno mora biti pokrenut unutar operacijskog sustava (OS) *FreeBSD*. Kako bi se olakšalo korištenje IMUNES-a korisnicima OS-a *Windows* ili drugih operacijskih sustava, OS *FreeBSD* može se pokrenuti unutar programa *VirtualBox* (taj program služi za pokretanje virtualnih računala na fizičkom računalu, *domaćinu*). Daljnje upute odnose se na OS *FreeBSD* pokrenut unutar programa *VirtualBox* koristeći OS *Windows* na računalu domaćinu.

##### *Instalacija programa VirtualBox*

1. Preuzeti instalacijsku datoteku programa *VirtualBox* korištenjem poveznice <https://www.virtualbox.org/wiki/Downloads> te sljedeći upute na web-stranici. Moguće je preuzeti 32-bitnu ili 64-bitnu verziju programa za OS *Windows* ili *Linux*, koja je besplatna za osobnu upotrebu. Pritom je važno da se koristi verzija 5.0 ili novija (Svi primjeri u ovom udžbeniku ispitani su na verziji 5.0.4).
2. Pokrenuti *.exe* datoteku i instalirati program *VirtualBox* prema uputama.

##### *Preuzimanje i pokretanje slike operacijskog sustava FreeBSD*

1. Pomoću web-preglednika sa stranice <http://imunes.net/download> (ili s poveznice koja će biti objavljena na stranici predmeta) preuzeti verziju 11.3 OS-a *FreeBSD* prilagođenu pokretanju unutar programa *VirtualBox* (tj. datoteku s ekstenzijom *.ova*). (Napomena: datoteku *.ova* moguće je skinuti i pomoću aplikacije *Torrent*, ali u tom slučaju na računalu treba biti instaliran *Torrent* klijent.)

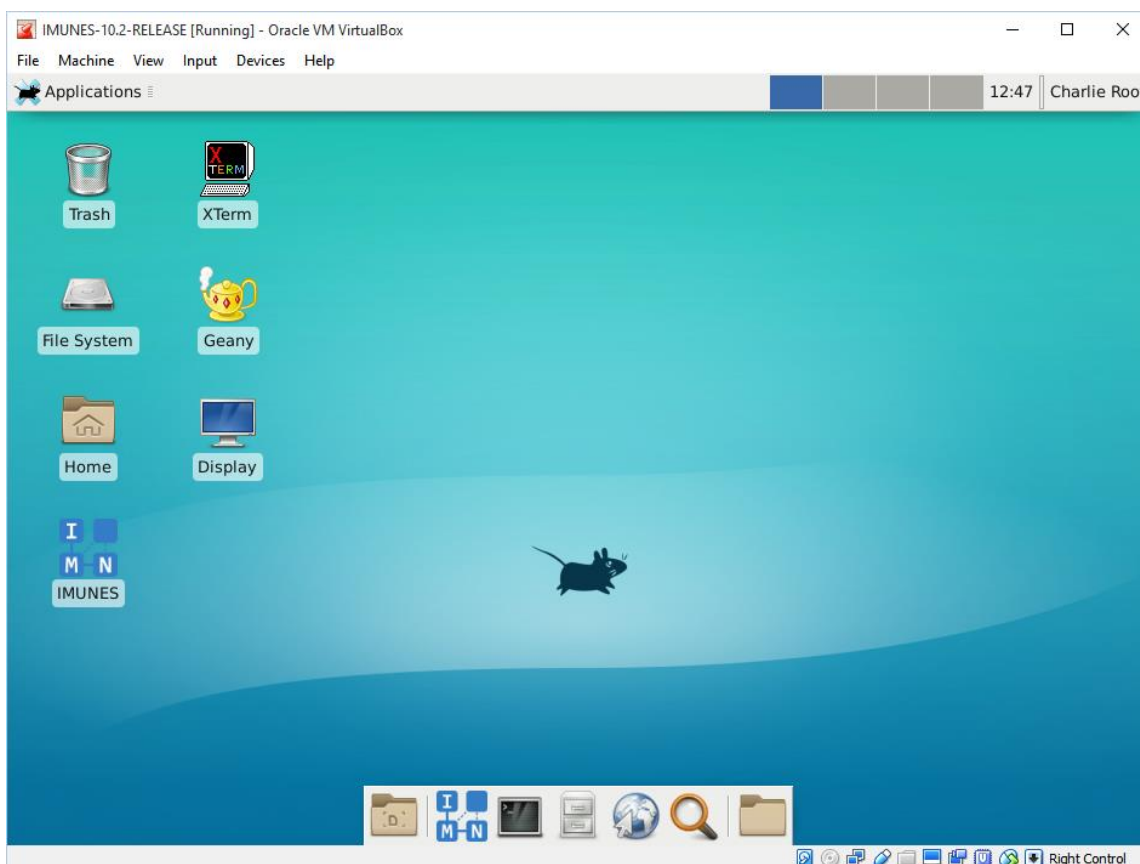
---

<sup>1</sup> Dokument „Vodič za alat IMUNES“ je dohvatljiv preko poveznice <http://www.imunes.net>.

2. Kreirati direktorij na OS-u *Windows* u koji će se pohraniti gostujući OS (*FreeBSD*).
3. Otvoriti datoteku *.ova* u programu *VirtualBox* (dvostrukim klikom na datoteku *.ova*), koja predstavlja sliku OS-a *FreeBSD*. U prozoru koji se pojavi pronađite opciju *Virtual Disk Image*, kliknite na upisanu vrijednost i po potrebi promijenite navedeni direktorij u neki direktorij u kojem imate prava pristupa tako da piše, npr., *D:\moj\_direktorij\IMUNES-11.3-RELEASE\_20190722-disk1.vmdk*. Nakon upisa odgovarajućeg direktorija kliknite na *Import*.
4. Kada se postupak završi, u programu *VirtualBox* s lijeve strane pojavit će se novi virtualni stroj *IMUNES-11.3-RELEASE* s oznakom *Powered Off*. Pokrenite ga dvostrukim klikom na njega.

### *Pokretanje emulatora/simulatora IMUNES*

Rad s programom *VirtualBox* je intuitivan. Klikom miša unutar prozora gdje se nalazi virtualno računalo, miš postaje „dio“ OS-a *FreeBSD*, i njime se dalje može rukovati kao unutar bilo kojeg operacijskog sustava. Kad je unutar OS-a *FreeBSD*, strelica miša je crne boje. Pomicanjem miša van prozora programa *VirtualBox* miš se može koristiti normalno u glavnom operacijskom sustavu računala (ako miš „zapne“ unutar programa *VirtualBox*, vratiti ga se može pritiskom na desnu tipku *Control* na tipkovnici). Aktivacija i deaktivacija prikaza programa *VirtualBox* na cijelom zaslonu provodi se kombinacijom tipki *Ctrf+F* (desni *Control*).

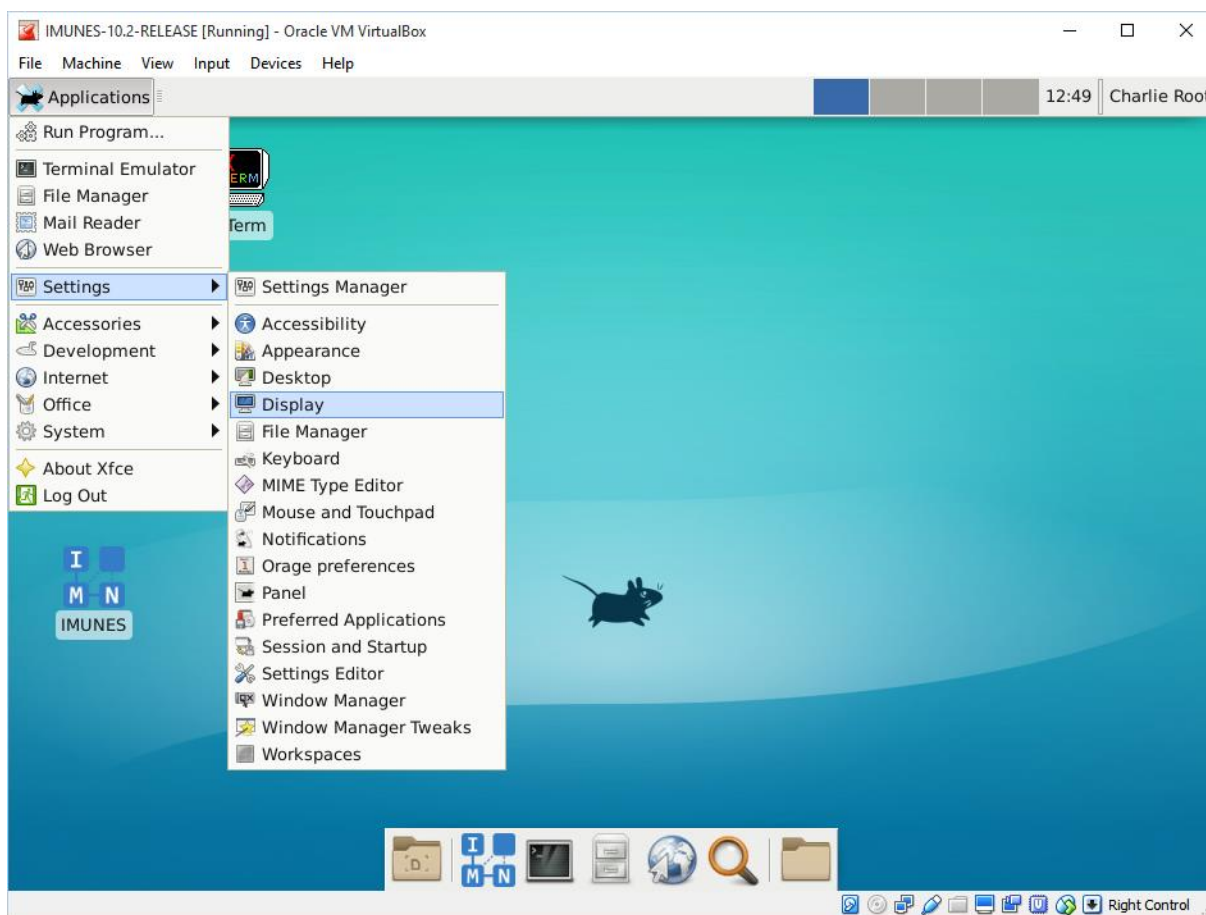


*Slika 1.1: Virtualno okruženje OS-a FreeBSD pokrenuto u programu VirtualBox*

Poveznica na alat IMUNES nalazi se na radnoj površini OS-a *FreeBSD*. Dvostrukim klikom miša on se i pokreće (Slika 1.1). Daljnji rad sa sustavom IMUNES opisan je u dokumentu *Vodič za IMUNES*.

### *Podošavanje rezolucije*

Moguće je podesiti rezoluciju slike virtualnog računala tako da postane razvučenija preko većeg dijela ekrana. To se realizira promjenom opcija u prozoru *Display* unutar grupe *Settings*, a do koje se dođe klikom na ikonu sa slikom crnog miša u gornjem lijevom kutu ekrana (Slika 1.2). Slika 1.3 prikazuje prozor *Display* i opciju *Resolution*, gdje se može odabrati željena rezolucija te kliknuti *Apply*. Napomena: rezolucija se automatski prilagođava i prilikom mijenjanja veličine prozora aplikacije VirtualBox.

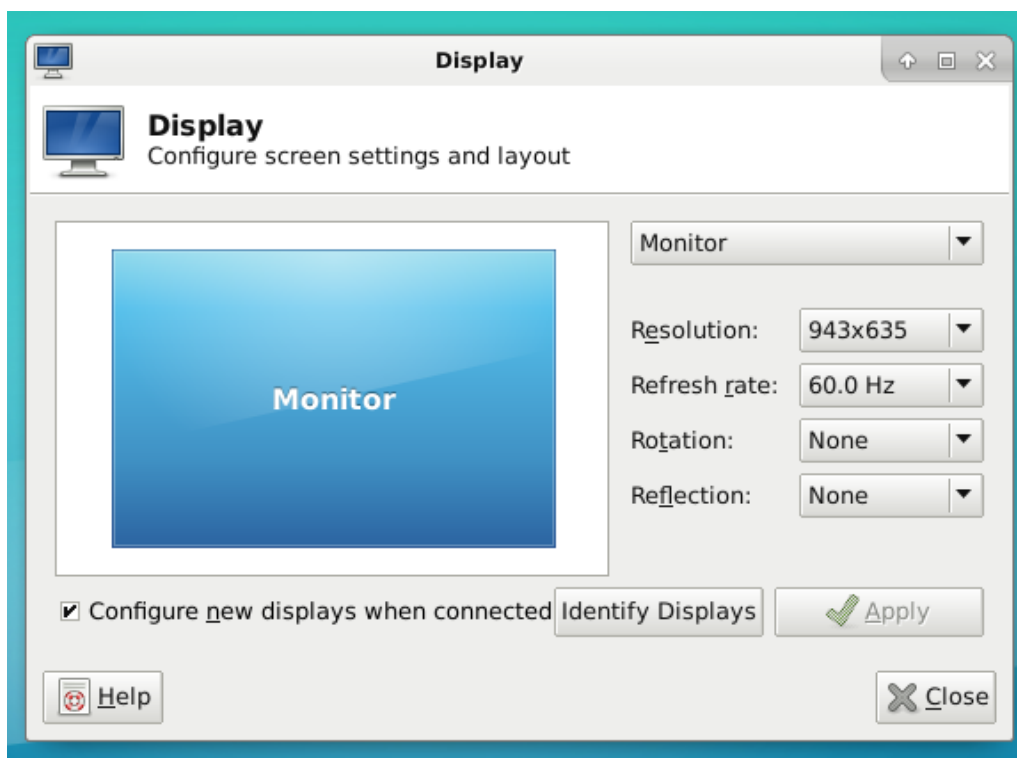


*Slika 1.2: Odabir opcije Display*

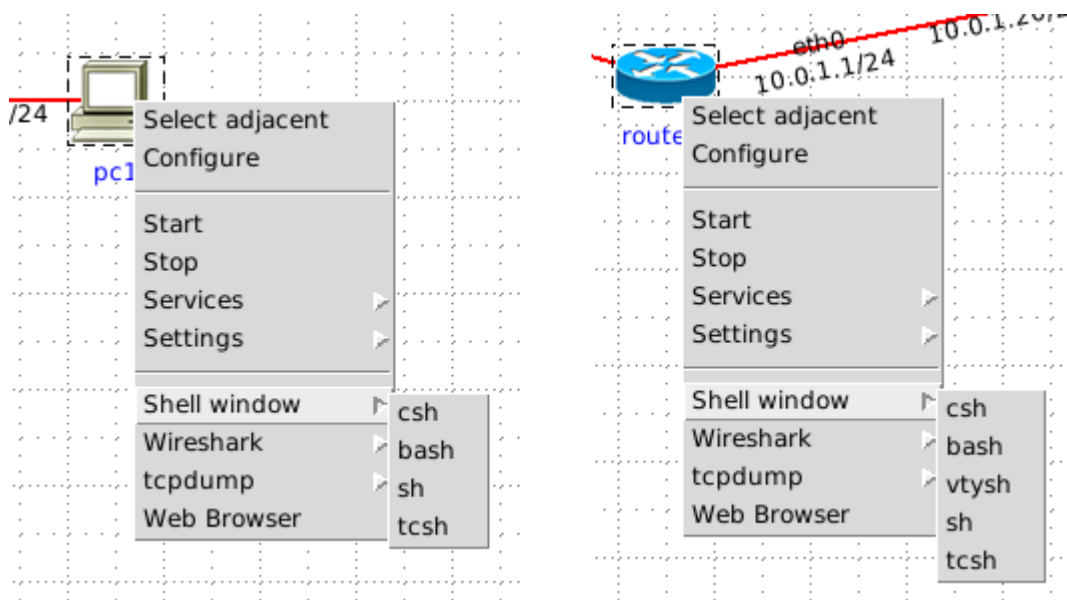
### *Rad u ljuskama (engl. shell)*

Prilikom rada s čvorovima u IMUNES-u, često je potrebno otvoriti konzolu pojedinog čvora i u njoj izvesti neku naredbu (npr. *ping*). U tom se slučaju desnim klikom miša iznad odabranog čvora otvara izbornik u kojem se odabere opcija *Shell window*, a koja nudi otvaranje konzole u nekoliko raspoloživih ljuski (engl. *shell*). Slika 1.4 prikazuje dva slučaja: u lijevom slučaju se nude ljuske *bash*, *csh*, *sh* i *tcsh*, a u desnom *bash*, *csh*, *sh*, *tcsh* i *vtsh*. Odabir ljuske je proizvoljan, ali se preporuča

korištenje ljuske *bash*, jer nudi napredne opcije poput često korištene *autocomplete* (započne se pisanje naredbe i stisne se tipka *Tab*, nakon čega ljuska sama ponudi završetak naredbe).



Slika 1.3: Odabir željene rezolucije

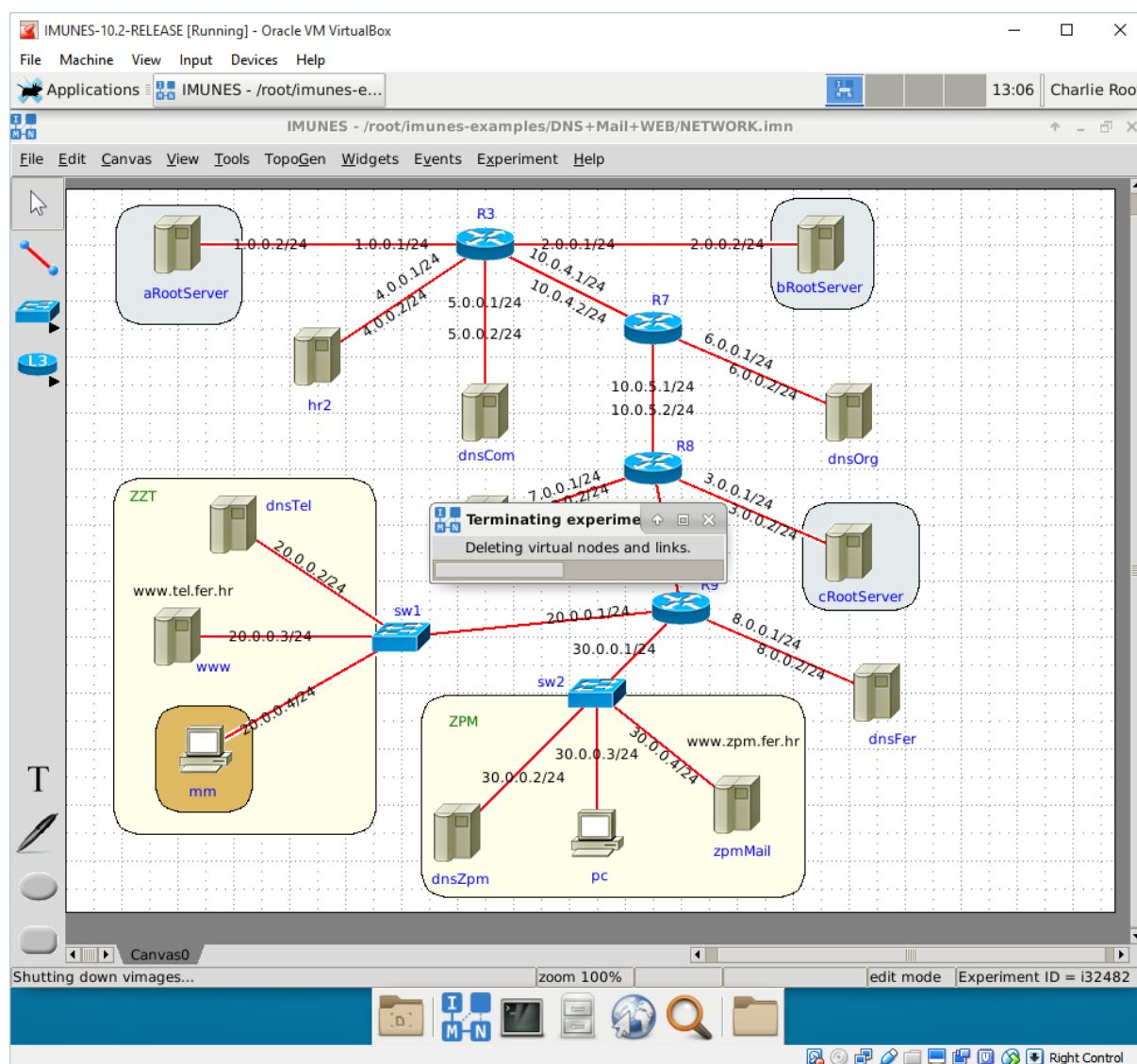


Slika 1.4: Rad u ljuskama

## Pokretanje i zaustavljanje eksperimenata

Pojedini eksperiment se u IMUNES-u pokreće odabirom opcija *Experiment* pa *Execute*.

Prije zatvaranja IMUNES-a ili prije zatvaranja aktivnog eksperimenta koji nam više nije potreban potrebno je aktivni eksperiment prekinuti klikom na opciju *Terminate* u izborniku *Experiment*. Tada se može pojaviti obavijest o prekidanju eksperimenta (pojavi se prozor s natpisom *Terminating experiment...*), kao što prikazuje Slika 1.5. Pri tome će se na dnu prozora u statusnoj traci ispisivati obavijesti o gašenju pojedinih čvorova. **Potrebno je pričekati da se eksperiment prekine do kraja kako se ne bi narušila stabilnost sustava.** Prekidanje eksperimenta je gotovo kada nestane obavijest o prekidanju, a poruke o gašenju pojedinih čvorova se prestanu ispisivati.



Slika 1.5: Prekidanje eksperimenta u IMUNES-u

Mrežno okruženje za konfiguraciju internetskih usluga: korištenje gotovih modela mreže ili izgradnja vlastitih Slika 1.6 prikazuje model mreže na kojem se mogu konfigurirati odabrane

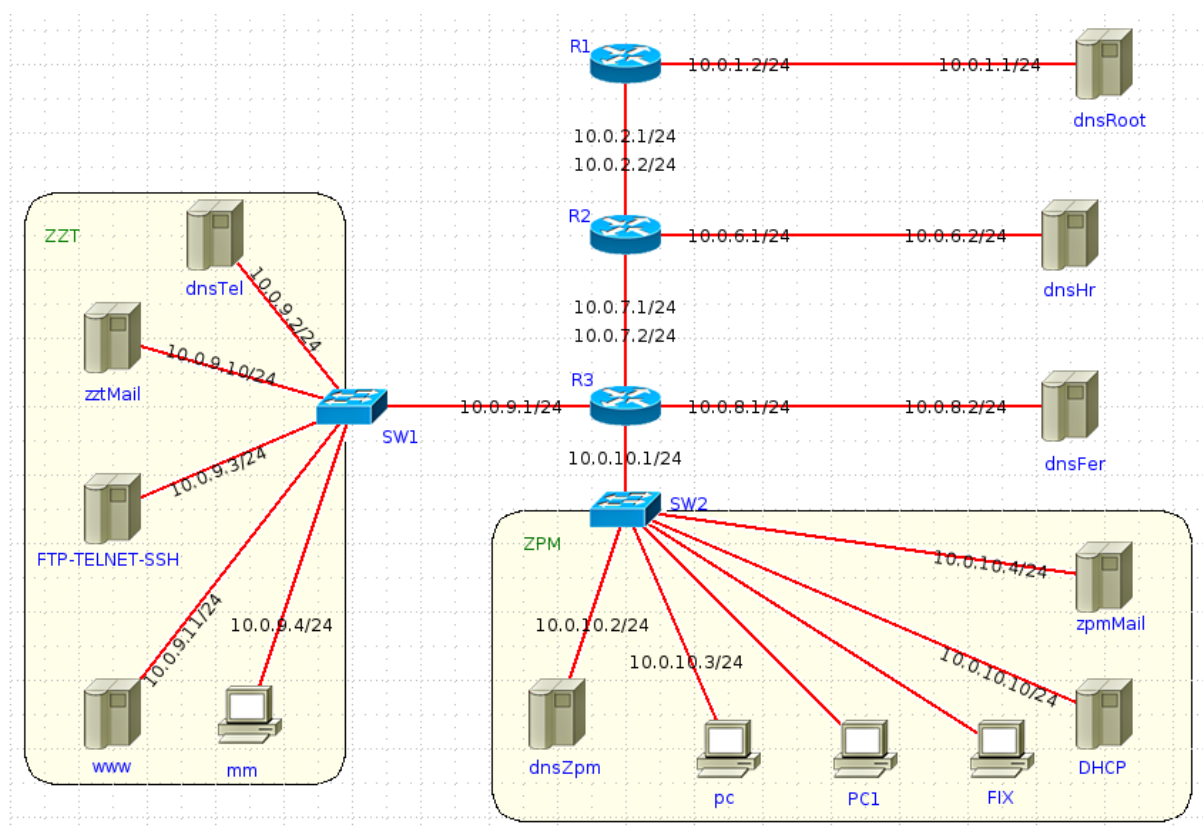
internetske usluge (poglavlja 1.2 - 1.5). Mreža obuhvaća nekoliko podmreža odvojenih usmjeriteljima, poslužitelje pojedinih internetskih usluga, te tri osobna računala kojima može biti dodijeljena statička ili dinamička IP-adresa. Model mreže pohranjen je u datoteci *Network-Config.imn*, koju je potrebno dohvatiti s udaljenog poslužitelja na sljedeći način. Nakon pozicioniranja u direktorij */root/imunes-examples*, u terminalu OS-a FreeBSD potrebno je unijeti sljedeće naredbe:

```
# wget public.tel.fer.hr/km/configScripts.tar

# tar xf configScripts.tar
```

Time se stvara direktorij *Config-scripts* u kojem se nalazi model mreže *Network-Config.imn* i pripadajuće konfiguracijske datoteke.

*Napomena:* Svaki pojedini čvor (virtualno računalo) u mreži sustava IMUNES ima svoj datotečni podsustav koji se kreira prilikom pokretanja svakog eksperimenta. Zbog toga se sve promjene na konfiguracijskim datotekama gube prilikom svakog zaustavljanja eksperimenta, te je potrebno pri njegovom ponovnom pokretanju nanovo podesiti konfiguracijske datoteke. Taj se postupak može pojednostavniti izradom gotovih konfiguracijskih skripti koje će se izvesti svaki put nakon pokretanja eksperimenta.



Slika 1.6: Model mreže na kojem se konfiguriraju odabrane internetske usluge

Osim otvaranja datoteka s gotovom mrežnom topologijom, u alatu IMUNES moguće je postaviti vlastitu proizvoljnu mrežu. Slika 1.7 prikazuje grafičko sučelje alata. U alatnoj traci s lijeve strane sučelja označene su ikone koje se koriste za dodavanje čvorova (elemenata) u topologiju mreže.



Značenja ikona su sljedeća (ikone su objašnjene istim redoslijedom kojim se pojavljuju na grafičkom sučelju):

- *Select Tool* (alat za označavanje) – koristi se za označavanje i premještanje elemenata;
- *Link* (alat za povezivanje) – koristi se za realizaciju fizičke povezanosti dva elementa;
- Izbornik *Link layer node*
  - *Hub* (parični obnavljač) – element lokalne mreže koji prosljeđuje okvire na sve izlazne priključke;
  - *LAN Switch* (ethernetski komutator) – element lokalne mreže koji prosljeđuje okvire na odabrane izlazne priključke koristeći tablicu komutiranja;
  - *Click Switch* (softverski komutator) – **ne koristi se na vježbama**;
  - *External Interface* (vanjsko sučelje) – priključak na vanjsko sučelje kojim se omogućuje emulacija mreže i prolaz stvarnog prometa kroz IMUNES (**ne koristi se na vježbama**);
  - RSTP switch – **ne koristi se na vježbama**
  - Filter node – **ne koristi se na vježbama**
  - Packet generator – **ne koristi se na vježbama**
- Izbornik *Networ layer node*
  - *Router* (usmjeritelj) – aktivni element mreže koji usmjerava datagrame temeljem tablica usmjeravanja, uz dodatnu mogućnost korištenja protokola usmjeravanja;
  - *Click Router* (softverski usmjeritelj) – **ne koristi se na vježbama**;
  - *Host* (poslužitelj) – krajnji čvor na kojem je moguće pokrenuti jednu ili više internetskih usluga i koji posjeduje tablicu usmjeravanja;
  - *PC* (osobno računalo) – krajnji čvor kojim se pristupa internetskim uslugama i koji posjeduje tablicu usmjeravanja.
  - NAT64 – **ne koristi se na vježbama**

Za izgradnju modela mreže potrebno je odabrati željeni čvor klikom na njegovu ikonu te potom kliknuti na radnu površinu emulatora/simulatora IMUNES. Nakon dodavanja elemenata, oni se povezuju alatom za povezivanje tako što se odabere taj alat, te se potom klikne na prvi čvor i povuče veza do drugog čvora bez otpuštanja tipke miša. Preporučuje se najprije povezati usmjeritelje s komutatorima, a tek potom povezati poslužitelje i računala s komutatorima. Poštujući taj redoslijed, automatski će se ispravno postaviti podrazumijevane (engl. *default*) rute u tablicama usmjeravanja krajnjih čvorova. U suprotnom, te će se rute morati naknadno podesiti.

Gore opisanim načinom automatski se postavljaju i IP-adrese i svi ostali parametri mreže. Ako je potrebno dodatno konfigurirati neki čvor, treba odabrati alat za označavanje i dvaput kliknuti na taj čvor. Slika 1.8 pokazuje prozor s postavkama za računalo *mm*. Postavke za ostale čvorove mijenjanju se na sličan način, a detaljnije upute za te postupke mogu se naći u dokumentu Vodič za alat IMUNES<sup>2</sup>.

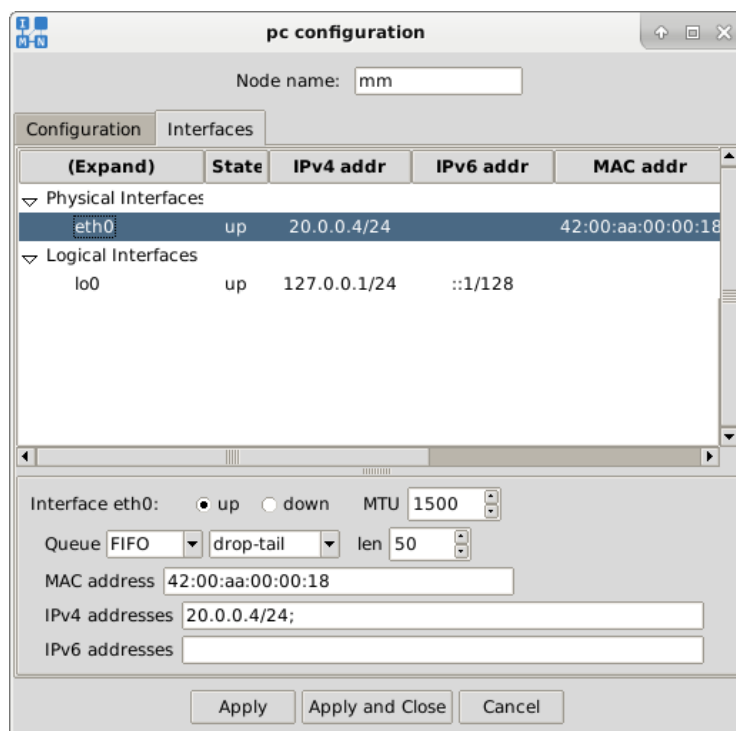
Nakon što je definirana topologija mreže i svi čvorovi su ispravno konfigurirani, eksperiment se može pokrenuti odabirom opcije *Execute* u izborniku *Experiment*.

---

<sup>2</sup> Dokument „Vodič za alat IMUNES“ je dohvatljiv preko poveznice <http://www.imunes.net>.



Slika 1.7: Glavno grafičko korisničko sučelje emulatora/simulatora IMUNES

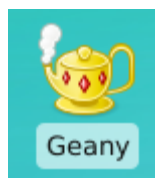


Slika 1.8: Prozor postavki čvora mm



### 1.1.2 Uređivači teksta

Za izradu i uređivanje konfiguracijskih datoteka pojedinih čvorova potrebno je koristiti jedan od uređivača teksta dostupnih u OS-u *FreeBSD*. Preporučuje se koristiti uređivač teksta *Geany*, koji je funkcionalno sličan alatu *Notepad* iz OS-a *Windows*. Taj alat se pokreće dvostrukim klikom na ikonu koja se nalazi na radnoj površini OS-a *FreeBSD* (Slika 1.9).



Slika 1.9: Ikona uređivača teksta *Geany*

Međutim, *Geany* nije moguće pokrenuti iz konzole pojedinog čvora u eksperimentu. (Naime, svaki pokrenuti čvor ponaša se kao zasebno računalo te je u tom smislu na svakome čvoru moguće pokrenuti samo njemu pripadajuću konzolu.) U tu svrhu koristi se uređivač teksta *nano*. On se pokreće naredbom `nano` u konzoli određenog čvora. Datoteke se pomoću ovog uređivača teksta otvaraju naredbom `nano <putanja do datoteke>`. Više informacija o tome kako koristiti uređivač *nano* može se dobiti upisom naredbe `man nano` u konzoli čvora.

U narednim poglavljima ispisi datoteka bit će prikazani u obliku kako to prikazuje *Ispis 1.1*. Bitno je naglasiti da brojevi redova u prvom stupcu nisu dio uređivača teksta, nego su ovdje dodani za lakše referenciranje pojedinih linija ispisa u tekstu knjige.

*Ispis 1.1: Primjer ispisa iz uređivača teksta *Geany* ili *nano**

01	Prva linija
02	Druga linija
03	Treća linija

### 1.1.3 Pristup datotečnom sustavu pojedinog čvora

Datotečni sustavi pojedinih čvorova grupirani su u direktoriju eksperimenta kao zasebni poddirektoriji, te je za pristup nekome od njih ključno doznati naziv direktorija eksperimenta i naziv poddirektorija čvora u eksperimentu. U tu svrhu koristi se naredba `himage` unutar terminala OS-a *FreeBSD* (dakle ne unutar terminala pojedinog čvora). Ovdje će se koristiti dvije opcije te naredbe:

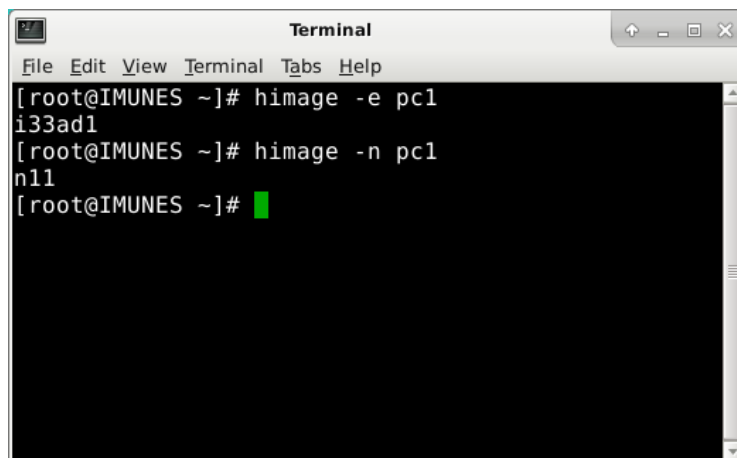
```
# himage -e <naziv_čvora> – dohvaća naziv direktorija u kojem je pohranjen
eksperiment koji sadrži traženi čvor;
```

```
# himage -n <naziv_čvora> – dohvaća naziv poddirektorija čvora.
```

Potrebno je otvoriti terminal OS-a *FreeBSD* klikom na ikonu alatne trake (Slika 1.10) i upisati naredbu `himage` s opcijom `-e` ili `-n` (Slika 1.11).



Slika 1.10: Ikona za otvaranje terminala operacijskog sustava FreeBSD



Slika 1.11: Primjer korištenja naredbe himage

Dobiveni podaci mogu se iskoristiti za pozicioniranje u korijenski direktorij traženog čvora (u ovom slučaju čvora PC) u terminalu OS-a *FreeBSD* na sljedeći način:

```
# cd /var/imunes/i33ad1/n11/root
```

#### 1.1.4 Kopiranje datoteka unutar OS-a FreeBSD

Datoteke u OS-u *FreeBSD* kopiraju se pomoću naredbe `cp`. Postupak kopiranja odvija se u 2 koraka:

1. Pozicioniranje u direktorij u kojem se nalazi datoteka koju želimo kopirati  

```
# cd <putanja>
```
2. Kopiranje datoteke iz direktorija u kojem smo se pozicionirali  

```
# cp <naziv datoteke> <putanja do odredišta>
```

#### 1.1.5 Kopiranje datoteka na pojedini čvor u pokrenutom IMUNES eksperimentu

Datoteke se na pojedini čvor kopiraju s pomoću naredbe `hcp` (pritom pazeći da je eksperiment na kojeg želimo kopirati datoteku pokrenut):

1. Pozicioniranje u direktorij u kojem se nalazi datoteka koju želimo kopirati  

```
# cd <putanja>
```
2. Kopiranje datoteke iz direktorija u kojem smo se pozicionirali  

```
# hcp <naziv datoteke> <ime čvora>:<putanja do odredišta u čvoru>
```

## 1.2 Konfiguriranje DHCP-poslužitelja

DHCP-protokol se koristi za dinamičku dodjelu IP-adresa računalima (tj. DHCP-klijentima) u mreži i obavješćavanje tih računala o ostalim važnim parametrima mreže, poput maske podmreže, adrese pretpostavljenog (engl. *default*) usmjeritelja, ili adrese nadležnog DNS-poslužitelja. DHCP-poslužitelj dodjeljuje adrese iz unaprijed definiranog adresnog raspona, pri čemu može dodijeliti statičke ili dinamičke adrese. Statičke adrese rezervirane su za određena računala i mogu biti dodijeljene samo tim računalima. Dinamičke adrese mogu se dodijeliti bilo kojim računalima u mreži (osim onima za koje već postoje rezervirane adrese), pri čemu neko računalo uvijek dobiva prvu slobodnu adresu iz adresnog raspona.

Ovdje će biti objašnjeno konfiguriranje DHCP-poslužitelja za dodjelu obje vrste adresa. Model mreže u kojoj će se konfigurirati DHCP-poslužitelj prikazuje Slika 1.6. Pri tome će se koristiti sljedeći entiteti u mreži:

- DHCP – DHCP-poslužitelj kojeg treba konfigurirati;
- PC1 – DHCP-klijent kojem poslužitelj dodjeljuje neku od raspoloživih IP-adresa na zahtjev;
- FIX – DHCP-klijent koji na zahtjev od poslužitelja uvijek dobiva istu IP-adresu.

### 1.2.1 Konfiguriranje i pokretanje DHCP-poslužitelja

Konfiguracija poslužitelja i pokretanje procesa za dodjelu adresa na zahtjev odvijaju se u tri koraka:

1. Stvaranje datoteke s konfiguracijskim parametrima za dodjelu adresa;
2. Kopiranje datoteka u ispravne direktorije DHCP-poslužitelja;
3. Pokretanje DHCP-poslužitelja.

#### Korak 1: Stvaranje datoteke s konfiguracijskim parametrima za dodjelu adresa

Prilikom pokretanja DHCP-poslužitelja, potrebno mu je kao parametar dati lokaciju datoteke koja sadrži definirane sve parametre potrebne za rad. Za stvaranje datoteke može se koristiti uređivač teksta *Mousepad*.

Ispis 1.2 prikazuje sadržaj konfiguracijske datoteke DHCP-poslužitelja. Od 7. do 10. linije definirani su parametri koji se tiču vremena važenja (engl. *lease*) dodijeljene adrese. To vrijeme označava koliko dugo (u sekundama) klijent smije koristiti dodijeljenu IP-adresu, a prije njegovog isteka klijent mora pokrenuti novi postupak dodjele adrese. Podrazumijevana vrijednost tog vremena pohranjuje se kao atribut `default-lease-time` (linija 7), a maksimalno dodijeljeno vrijeme pohranjuje se kao atribut `max-lease-time` (linija 8). Nakon pokretanja DHCP-poslužitelja, podaci o dodijeljenim adresama i njihovim *lease* vremenima čuvat će se u datoteci s ekstenzijom *.leases* u direktoriju */var/db* DHCP-poslužitelja. Putanja do *.leases* datoteke, pohranjena je u 9. liniji konfiguracijske datoteke.

Sljedeće je potrebno odrediti podmrežu u kojoj će djelovati DHCP-poslužitelj i raspon adresa koje će dinamički dodjeljivati. Podmreža u primjeru je 10.0.10.0/24, odnosno 10.0.10.0 s maskom podmreže 255.255.255.0 (linija 14). Raspon dinamički dodijeljenih adresa u ovom se primjeru kreće od 10.0.10.10 do 10.0.10.20 (linija 15). Također je navedena i adresa podrazumijevanog usmjeritelja u

podmreži (linija 16). U bloku `host fixed` (linije 19 - 22) unose se DHCP-klijenti kojima će uvijek biti dodijeljena ista IP-adresa. To se radi tako da se unese MAC-adresa klijenta i IP-adresa koja će mu biti dodijeljena. MAC-adresa klijenta može se doznati otvaranjem kartice *Interfaces* na tom klijentu (Slika 1.12). U ovom primjeru konfigurirana je fiksna adresa za čvor FIX. U konfiguracijskoj datoteci se navode i sljedeći atributi:

- Atribut `option domain-name` – ime domene u kojoj se nalazi DHCP-poslužitelj (ova se opcija koristi ako je u mreži konfiguriran DNS, o čemu će više riječi biti kasnije);
- Atribut `option domain-name-servers` – nadležni DNS-poslužitelj u ovoj domeni (opcija se također koristi u slučaju da je u mreži konfiguriran DNS). U ovom primjeru odabrana je IP-adresa poslužitelja *dnsZpm* kao IP-adresa nadležnog DNS-poslužitelja (Slika 1.6);
- Atribut `ddns-update-style` – odabran je način ažuriranja DNS-a *interim*, što znači da se prvo određuje ime domaćina (*hostname*), a zatim ime domene;
- Atribut `authoritative` – njegovo navođenje omogućuje DHCP-poslužitelju slanje DHCP-NAK poruka pogrešno konfiguriranim klijentima i nakon toga dodjeljivanje ispravne IP-adrese klijentima;
- Atribut `log-facility` – određuje način vođenja zapisa.

*Ispis 1.2: Konfiguracijska datoteka DHCP-poslužitelja*

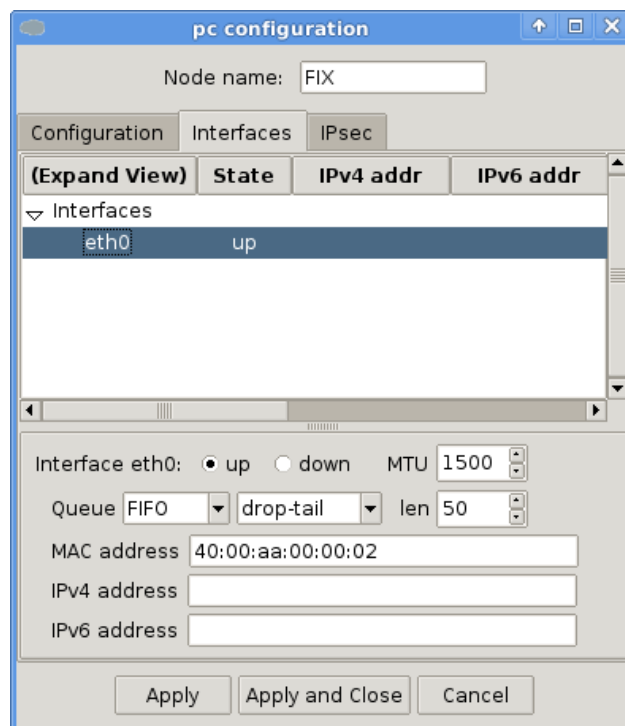
---

```

01  # Datoteka dhcpd.conf na DHCP-poslužitelju
02
03  option domain-name "ZPM";
04  option domain-name-servers 10.0.10.2;
05
06
07  default-lease-time 300;
08  max-lease-time 7200;
09  lease-file-name "/var/db/imunes-dhcpd.leases";
10  ddns-update-style interim;
11  authoritative;
12  log-facility local7;
13
14  subnet 10.0.10.0 netmask 255.255.255.0 {
15      range 10.0.10.10 10.0.10.20;
16      option routers 10.0.10.1;
17  }
18
19  host fixed {
20      hardware ethernet 40:00:aa:00:00:02;
21      fixed-address 10.0.10.30;
22  }

```

---



Slika 1.12: MAC-adresa DHCP-klijenta (računalo FIX)

Više o opcijama konfiguriranja DHCP-poslužitelja moguće je saznati upisom naredbi `man dhcpd` ili `man dhcpd.conf` u konzolu čvora DHCP nakon pokretanja eksperimenta.

#### Korak 2: Dodavanje potrebnih datoteka u datotečni sustav DHCP-poslužitelja

Nakon što je stvorena konfiguracijska datoteka, potrebno ju je trajno pohraniti unutar OS-a *FreeBSD*. U ovom primjeru datoteka se sprema u direktorij `/root/imagenes-examples/Config-scripts/DHCP` pod nazivom `DHCP.dhcpd.conf`. Međutim, ta datoteka mora biti kopirana i u direktorij čvora DHCP kako bi DHCP-poslužitelj na njemu mogao ispravno raditi. To je potrebno napraviti svaki put nakon pokretanja eksperimenta jer se direktoriji svih čvorova brišu prilikom zaustavljanja eksperimenta.

Dakle, nakon pokretanja eksperimenta otvorit ćemo terminal OS-a *FreeBSD* (Slika 1.10). Prvo ćemo stvoriti datoteku `imagenes-dhcpd.leases` u koju će DHCP-poslužitelj pohranjivati podatke o dodijeljenim adresama i njihovim *lease* vremenima. To se radi sljedećim nizom naredbi u otvorenom terminalu:

1. `# himage DHCP touch /var/db/imagenes-dhcpd.leases`  
(naredbom `touch` stvara se navedena datoteka u čvoru *DHCP* u mapi s putanjom `/var/db`)

Potom ćemo kopirati konfiguracijsku datoteku (koju smo osmislili u Koraku 1) u `tmp` direktorij čvora DHCP, koji će imati ulogu DHCP-poslužitelja:

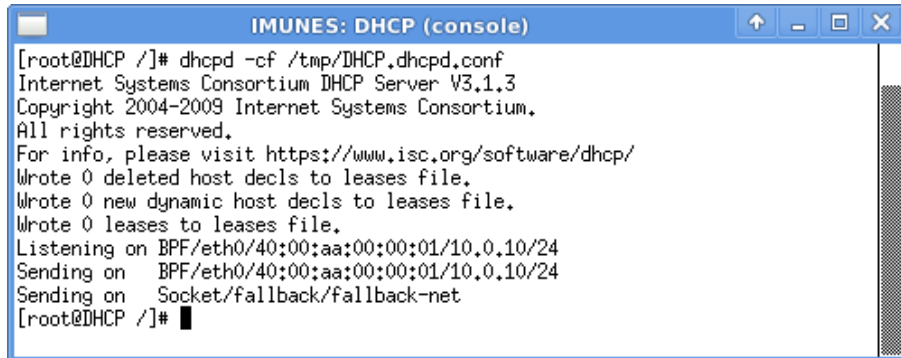
1. `# cd /root/imagenes-examples/Config-scripts/DHCP`  
(pozicioniranje u direktorij u kojem je spremljena konfiguracijska datoteka)
2. `# hcp DHCP.dhcpd.conf DHCP:/tmp`  
(kopiranje konfiguracijske datoteke u direktorij `tmp` na DHCP-poslužitelju)

### Korak 3: Pokretanje DHCP-poslužitelja

DHCP-poslužitelj se pokreće u konzoli čvora DHCP unosom naredbe

```
# dhcpd -cf /tmp/DHCP.dhcpd.conf
```

Nakon toga u konzoli bi se trebao pojaviti ispis kao na slici (Slika 1.13).



Slika 1.13: Pokretanje DHCP-poslužitelja

Ako želimo zaustaviti i ponovo pokrenuti DHCP-poslužitelj, potrebno je zaustaviti sve pokrenute *dhcpd* procese. Te procese je moguće ukloniti naredbom:

```
# killall -9 dhcpd
```

*S obzirom da svaki put prilikom pokretanja eksperimenta treba kopirati konfiguracijske datoteke u direktorij na čvoru i da treba pokrenuti pripadajući DHCP-poslužitelj, taj se postupak može ubrzati izradom skripte i njenim pohranjivanjem u trajni direktorij na OS-u FreeBSD (npr. u direktorij /root/imunes-examples/DHCP). Skripta se pokreće izvršavanjem naredbe ./Ime\_skripte u konzoli OS-a FreeBSD, obavezno tek nakon pokretanja eksperimenta. Primjer jedne takve skripte je skripta /root/imunes-examples/DHCP/start\_dhcp.*

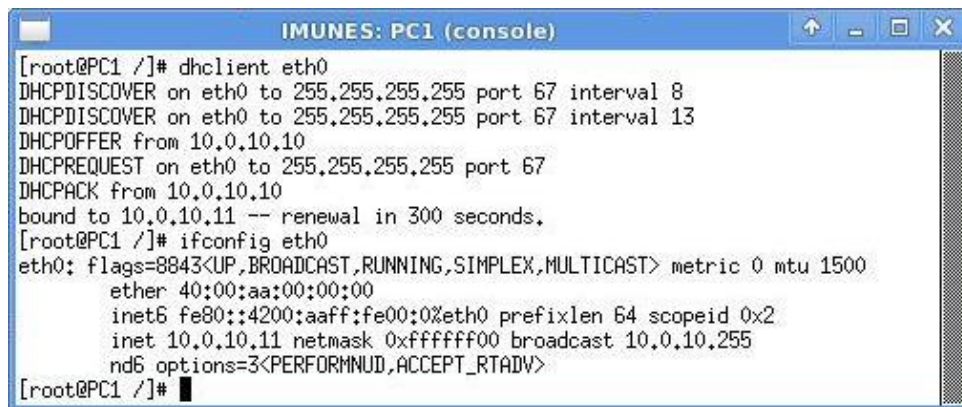
#### 1.2.2 Primjer dodjele IP-adrese pomoću DHCP-poslužitelja

Nakon što je DHCP-poslužitelj pokrenut, moguće je započeti proces traženja dodjele IP-adrese na klijentu. Kao primjer pokrenut ćemo konzolu na čvoru *PCI* (Slika 1.14). Dinamička dodjela adrese za sučelje *eth0* čvora *PCI* inicira se naredbom

```
# dhclient eth0
```

Ako je sve u redu, u konzoli bi se trebale ispisati poruke protokola DHCP, a na kraju obavijest da je čvor vezan uz dodijeljenu adresu te koliko dugo je adresa valjana (Slika 1.14). Provjeru je li adresa ispravno konfigurirana možemo obaviti naredbom

```
# ifconfig eth0
```



```

IMUNES: PC1 (console)
[root@PC1 /]# dhclient eth0
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 8
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 13
DHCPOFFER from 10.0.10.10
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 10.0.10.10
bound to 10.0.10.11 -- renewal in 300 seconds.
[root@PC1 /]# ifconfig eth0
eth0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    ether 40:00:aa:00:00:00
    inet6 fe80::4200:aaff:fe00:0%eth0 prefixlen 64 scopeid 0x2
    inet 10.0.10.11 netmask 0xfffff00 broadcast 10.0.10.255
    nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
[root@PC1 /]#

```

Slika 1.14: Zahtjev za dodjelom adrese (ispis u konzoli čvora PC1)

Ako isti postupak primijenimo na računalu FIX, vidjet ćemo da je tom računalu dodijeljena statička IP-adresa 10.0.10.30, kako je i bilo predviđeno u konfiguracijskoj datoteci DHCP-poslužitelja.

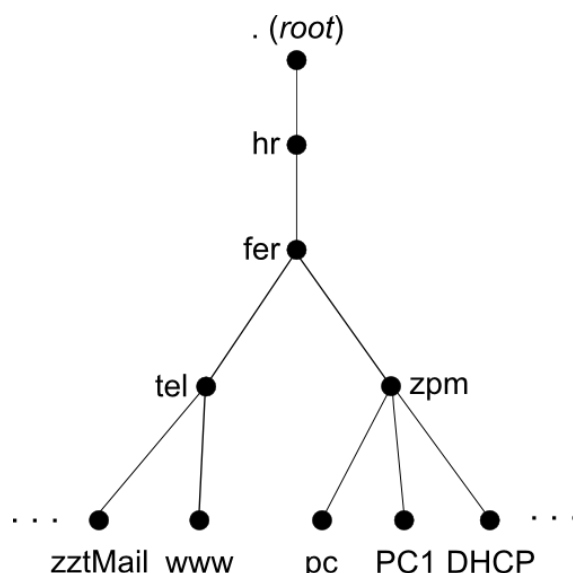
### 1.3 Konfiguriranje DNS-poslužitelja

U ovom poglavlju bit će opisano konfiguriranje DNS-poslužitelja. S obzirom da se sustav DNS sastoji od više DNS-poslužitelja nadležnih za različite domene, postupak konfiguracije je nešto složeniji nego u prošlom poglavlju, s obzirom da moramo osigurati ispravnu interakciju između DNS-klijenata i DNS-poslužitelja, kao i međusobnu interakciju DNS-poslužitelja.

DNS (*Domain Name System*) je distribuirani hijerarhijski sustav poslužitelja u kojem se nalaze informacije o domenskim imenima računala i njima pripadajućim IP-adresama. Najvažnije svojstvo DNS sustava je da prevodi čovjeku lako pamtljiva domenska imena računala u računalu pogodne numeričke zapise, tj. IP-adrese. Entitete u mreži koji od DNS-poslužitelja traže razlučivanje IP-adresa zovemo DNS-klijentima. To mogu biti osobna računala, razni poslužitelji, ili drugi DNS-poslužitelji. Komunikacijski protokol koji se koristi u tu svrhu zove se protokol DNS.

DNS koristi stablasti hijerarhijski sustav imena, gdje hijerarhija odgovara organizacijskoj strukturi internetskih domena. Za odvajanje razina hijerarhije u imenima koristi se znak '.' (točka). Primjer jedne hijerarhijske strukture (prostora imena) prikazuje Slika 1.15. Svaka točka na slici predstavlja jedan čvor u mreži. Listove stabla predstavljaju DNS-klijenti, dok ostale čvorove predstavljaju DNS-poslužitelji zaduženi za pojedinu internetsku domenu. U ovom primjeru za vršnu domenu (*root*) zadužen je DNS-poslužitelj *dnsRoot*, za domenu *hr* zadužen je DNS-poslužitelj *dnsHr*, za domenu *fer* zadužen je DNS-poslužitelj *dnsFer*, itd.

Svaki DNS-poslužitelj sadrži potpune informacije o imenima računala i njihovim IP-adresama smještenih u domeni za koju je zadužen. Takav DNS-poslužitelj zovemo nadležnim poslužiteljem (engl. *authoritative nameserver*) za tu domenu. Na primjer, DNS-poslužitelj *dnsHr* je nadležan za domenu *hr*, a DNS-poslužitelj *dnsTel* za domenu *tel.fer.hr*. Ovakva podjela odgovornosti po domenama omogućava lakšu proširivost sustava i dodavanje novih poddomena po potrebi. Bez toga DNS-sustav ne bi mogao normalno funkcionirati. Primjerice, kad bi jedan poslužitelj bio odgovoran za cijelu domenu *hr* i sve njezine poddomene, morao bi imati potpune informacije o svim računalima u cijeloj domeni, što bi rezultiralo s nekoliko tisuća zapisa u bazi DNS-poslužitelja. Isto tako, takav poslužitelj bi morao odgovarati na veliki broj upita u sekundi te bi zahtijevao veliku procesorsku snagu i izuzetno brzi pristup Internetu.



Slika 1.15: Primjer hijerarhije DNS-sustava

### 1.3.1 Korišteni model mreže i koraci konfiguracije DNS-poslužitelja

Mreža koja će se koristiti za izgradnju sustava DNS opisana je u prvom poglavlju (Slika 1.6). Konfigurirat će se sljedeći DNS-poslužitelji:

- *dnsRoot* (IP-adresa 10.0.1.1, zadužen za domenu *root*),
- *dnsHr* (IP-adresa 10.0.6.2, zadužen za domenu *hr*),
- *dnsFer* (IP-adresa 10.0.0.2, zadužen za domenu *fer*),
- *dnsTel* (IP-adresa 10.0.9.2, zadužen za domenu *tel*),
- *dnsZpm* (IP-adresa 10.0.10.2, zadužen za domenu *zpm*).

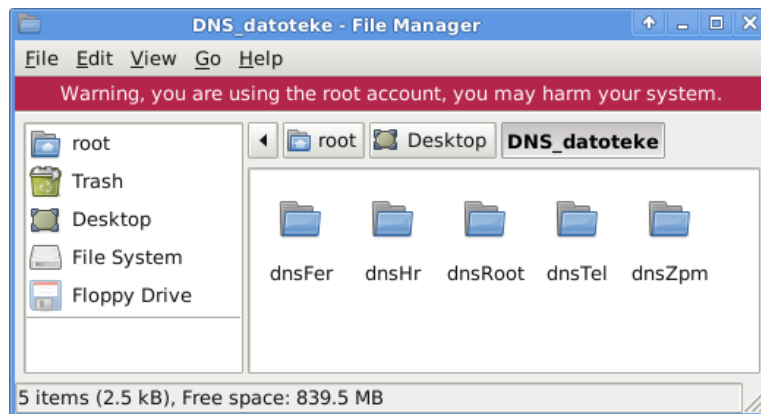
Konfiguriranje i pokretanje sustava DNS-poslužitelja i DNS-klijenata odvija se u četiri koraka:

1. Definiranje konfiguracijskih datoteka DNS-poslužitelja;
2. Definiranje domenskih datoteka DNS-poslužitelja;
3. Definiranje konfiguracijskih datoteka DNS-klijenata;
4. Kopiranje datoteka u pripadajuće direktorije.

#### Korak 1: Definiranje konfiguracijskih datoteka DNS-poslužitelja

Budući da se pri pokretanju eksperimenta na svakom čvoru iznova stvara datotečni podsustav, datoteke je preporučljivo pohraniti na zasebnom mjestu (izvan datotečnog podsustava čvora) kako ih ne bi morali iznova pisati prilikom svakog pokretanja eksperimenta. Ovdje ćemo kreirati novi direktorij na radnoj površini OS-a *FreeBSD* pod nazivom *DNS\_datoteke*. Za svaki DNS-poslužitelj unutar tog direktorija kreirat ćemo direktorij koji nosi naziv DNS-poslužitelja kako je definirano u modelu mreže (npr. za DNS-poslužitelj *dnsRoot* napraviti ćemo istoimeni direktorij *dnsRoot*). Konačni izgled direktorija *DNS\_datoteke* i njegovih poddirektorija prikazuje Slika 1.16.



Slika 1.16: Direktorij *DNS\_datoteke* na radnoj površini OS-a *FreeBSD*

Potom treba kreirati i urediti konfiguracijsku datoteku za svaki DNS-poslužitelj, i potom je pohraniti u pripadajući direktorij na radnoj površini OS-a *FreeBSD*. Za uređivanje datoteke koristit će se uređivač teksta *Mousepad*.

Funkcionalnost pojedinog DNS-poslužitelja definira se u konfiguracijskoj datoteci *named.conf*. Dakle, različiti DNS-poslužitelji imat će datoteku *named.conf* različitog sadržaja, ali istog imena. U slučaju vršnog poslužitelja *dnsRoot*, strukturu njegove konfiguracijske datoteke *named.conf* prikazuje Ispis 1.3.

Ispis 1.3: Konfiguracijska datoteka *named.conf* za vršni DNS-poslužitelj *dnsRoot*


---

```

01 // named.conf
02
03 options {
04     directory "/var/named/etc/namedb";
05 };
06
07 zone "." {
08     type master;
09     file "root";
10 };
11
12 zone "0.0.127.IN-ADDR.ARPA" {
13     type master;
14     file "localhost.rev";
15 };

```

---

Konfiguracijska datoteka sastoji se od sljedećih parametara. Parametrom *options* definiraju se osnovne i dodatne opcije poslužitelja. Jedna takva opcija je direktorij iz kojeg će se čitati datoteke u kojima su definirane domene, tj. zone (linija 4). Potom slijede parametri domene *zone* pomoću kojih se definira tip domene te ime datoteke u kojoj se nalaze opcije domene (linija 9, parametar *file*). Tip domene *master* (linije 8 i 13) označava da DNS-poslužitelj sadrži glavnu (engl. *master*) kopiju podataka za domenu i da je upravo on nadležni poslužitelj. Informaciju o vršnoj domeni potrebno je staviti u svaki DNS-poslužitelj koji se konfigurira, kako bi znao proslijediti upit za razlučivanjem imena vršnom poslužitelju, ako ne zna odgovoriti na upit. Domena *0.0.127.IN-ADDR.ARPA* (linija 12) je posebna domena pomoću koje poslužitelj može razlučiti ime *localhost* iz IP-adrese 127.0.0.1 i ona je također sastavni dio i svih ostalih konfiguracijskih datoteka. Nakon unesenih postavki datoteku

ćemo spremati pod nazivom *named.conf* u direktorij *dnsRoot* na radnoj površini OS-a *FreeBSD*. Podrobniji opis datoteke *named.conf* može se pogledati unosom sljedeće naredbe u konzolu:

```
# man named.conf
```

Za ostale DNS-poslužitelje (*dnsHr*, *dnsFer*, *dnsTel* i *dnsZpm*) potrebno je unutar konfiguracijske datoteke dodati još jedan parametar *zone* koji će opisivati danu domenu. Na primjer, za DNS-poslužitelj domene *fer.hr* treba dodati sljedeće linije (Ispis 1.4):

Ispis 1.4: Dodatni parametar *zone* za DNS-poslužitelj domene *fer.hr*

---

```
01     zone "fer.hr" {
02         type master;
03         file "fer";
04     };
```

---

Također, za konfiguracijske datoteke navedenih DNS-poslužitelja treba u liniji 8 iz ispisa Ispis 1.3 postaviti tip domene *hint* umjesto *master*. Tip domene *hint* označava domenu u kojoj se nalazi popis vršnih poslužitelja.

#### Korak 2: Definiranje domenskih datoteka DNS-poslužitelja

Svaki DNS-poslužitelj sadrži određeni broj domenskih datoteka u kojima su definirana preslikavanja naziva domena u IP-adrese čvorova u toj domeni. Broj i vrsta domenskih datoteka definirani su u prethodnom koraku u konfiguracijskoj datoteci poslužitelja (Ispis 1.3 i Ispis 1.4). Ovdje će biti pojašnjene domenske datoteke *fer*, *localhost.rev* i *named.root*.

Najvažnija domenska datoteka je ona koja opisuje domenu za koju je DNS-poslužitelj nadležan. Ispis 1.5 prikazuje sadržaj domenske datoteke *fer* koja opisuje razlučivanje imena za domenu *fer.hr* DNS-poslužitelja *dnsFer*. Datoteka sadrži zapise o resursima (engl. *resource record*). Parametar TTL (*time to live*) označava vrijeme valjanosti svih zapisa o resursima. Zapis SOA (*start of authority*, linija 4) označava najprihvatljiviji nadležni DNS-poslužitelj za zadanu domenu (u slučaju kad postoji više DNS-poslužitelja za jednu domenu). Zadana domena skraćeno je označena znakom '@', a označava onu domenu za koju je napravljena ova datoteka (u ovom slučaju to je domena *fer.hr*). Nadalje, parametar IN označava klasu Internet (postoje i druge klase, ali ovdje se neće razmatrati). Nakon parametra SOA slijedi adresa DNS-poslužitelja nadležnog za zadanu domenu (u ispisu Ispis 1.5 to je poslužitelj *dnsFer.fer.hr*, linija 4) te adresa elektroničke pošte administratora tog poslužitelja (ovdje je to adresa *root.dnsFer.fer.hr*). Dijelovi adrese elektroničke pošte odijeljeni su samo točkama jer tako nalaže sintaksa, a prava se adresa dobije tako da se prva točka u nazivu zamijeni znakom '@' (dakle, *root@dnsFer.fer.hr*). Značenja parametara *serial*, *refresh*, *retry* i *expire* nisu nužna za razumijevanje ovog primjera i ovdje se neće detaljnije objašnjavati.

Nakon toga slijede zapisi o resursima. Ovdje su dodani zapisi resursa NS (*nameserver*; linije 12, 15 i 18). Ti zapisi povezuju domene s poslužiteljima koji su za njih nadležni. Dodaje se po jedan zapis NS za svaki poslužitelj koji je nadležan za našu domenu (*fer.hr*) ili se nalazi unutar naše domene (*tel.fer.hr* i *zpm.fer.hr*).

Na sličan način treba dodati zapise NS u domenske datoteke ostalih DNS-poslužitelja. U vršnoj domeni postoje dva poslužitelja (*dnsRoot*, i *dnsHr* koji je zadužen za domenu *hr*); u domeni *hr* postoje dva poslužitelja (*dnsHr*, i *dnsFer* koji je zadužen za domenu *fer.hr*), a u domeni *fer.hr* postoje

tri poslužitelja kako je gore objašnjeno (*dnsFer*, *dnsTel* koji je zadužen za domenu *tel.fer.hr* i *dnsZpm* koji je zadužen za domenu *zpm.fer.hr*).

Posljednji zapisi su zapisi resursa A (*address*) (linije 13, 16 i 19). Njima definiramo adrese u koje će se preslikati nazivi poslužitelja i klijenata koji se nalaze u našoj domeni.

*Ispis 1.5: Domenska datoteka fer*

---

```

01 ; fer
02 ;
03 $TTL 60000
04 @ IN SOA dnsFer.fer.hr. root.dnsFer.fer.hr (
05             10 ; serial
06             28 ; refresh
07             14 ; retry
08             3600000 ; expire
09             0 ; default_ttl
10             )
11
12 @ IN NS dnsFer.fer.hr.
13 dnsFer.fer.hr. IN A 10.0.0.2
14
15 tel.fer.hr. IN NS dnsTel.tel.fer.hr.
16 dnsTel.tel.fer.hr. IN A 10.0.9.2
17
18 zpm.fer.hr. IN NS dnsZpm.zpm.fer.hr.
19 dnsZpm.zpm.fer.hr. IN A 10.0.10.2

```

---

Što se tiče domene *localhost*, svaki DNS-poslužitelj sadrži identičnu kopiju datoteke *localhost.rev* koju ilustrira Ispis 1.6.

*Ispis 1.6: Domenska datoteka localhost.rev na svakom DNS-poslužitelju*

---

```

01 ; localhost.rev
02 ;
03 $TTL 86400
04 @ IN SOA localhost.root.localhost (
05             20041128 ; Serial
06             28800 ; Refresh
07             7200 ; Retry
08             3600000 ; Expire
09             86400 ; Minimum
10             )
11 IN NS localhost.
12 1 IN PTR localhost.

```

---

U njoj je definirano preslikavanje adrese 127.0.0.1 u naziv *localhost* pomoću zapisa PTR (*pointer record*, linija 12). Preslikavanje iz IP-adrese u domensko zovemo reverzno DNS preslikavanje, a inicira se slanjem DNS-upita tipa PTR prema DNS-poslužitelju.

Nadalje, svaki DNS-poslužitelj osim vršnog DNS-poslužitelja (*dnsRoot*) sadrži identičnu kopiju datoteke *named.root* u kojoj je definirano preslikavanje vršne domene u adresu vršnog poslužitelja (vršnih poslužitelja može biti više). Sadržaj ove datoteke prikazuje Ispis 1.7.

Ispis 1.7: Domenska datoteka *named.root*

01	; named.root				
02	;				
03	.	3600000	IN	NS	dnsRoot.
04	dnsRoot.	3600000		A	10.0.1.1

**Korak 3: Definiranje konfiguracijskih datoteka DNS-klijenata**

Konfiguracijska datoteka DNS-klijenta omogućava konfiguriranje razlučitelja (engl. *resolver*), programa na klijentu zaduženog za komunikaciju s DNS-poslužiteljima. Datoteka sadrži domenu u kojoj se nalazi klijent te adresu nadležnog poslužitelja za tu domenu. Datoteku je potrebno spremiti u direktorij *DNS\_datoteke* pod nazivom *resolv.x* gdje *x* označava naziv klijenta (npr. *mm*, *www*, *pc*, *DHCP*, itd.). Primjer konfiguracijske datoteke klijenta *zztMail* prikazuje Ispis 1.8.

Ispis 1.8: Konfiguracijska datoteka *resolv.zztMail*

01	domain	tel.fer.hr
02	nameserver	10.0.9.2

**Korak 4: Kopiranje datoteka u pripadajuće direktorije**

Kako bismo nakon pokretanja eksperimenta kopirali sve datoteke na potrebna mjesta, iskoristit ćemo već gotovu skriptu *start\_dns\_new* koja se nalazi u direktoriju s primjerima

```
/root/imagenes-examples/Config-scripts/DNS.
```

Skriptu je potrebno otvoriti pomoću uređivača teksta *Mousepad* i spremiti je pod drugim nazivom (npr. *start\_dns\_my*). U skripti (Ispis 1.9) potrebno je promijeniti sljedeće:

- u varijablu *dns\_servers* pohraniti imena poslužitelja koji se koriste u primjeru (to su *dnsRoot*, *dnsHr*, *dnsFer*, *dnsTel*, *dnsZpm*);
- u varijablu *hosts* pohraniti imena klijenata koji se koriste u primjeru (to su *zztMail*, *FTP-TELENET-SSH*, *www*, *mm*, *pc*, *DHCP*, *zpmMail*)
- umjesto retka *cd DNS\_files* unijeti *cd /root/Desktop/DNS\_datoteke*

Za pokretanje DNS-poslužitelja koristi se program BIND (*Berkeley Internet Name Domain*)<sup>3</sup> koji je u operacijskom sustavu *FreeBSD*, u kojem se izvršava IMUNES, nazvan *named*. Konačni izgled skripte prikazuje Ispis 1.10.

Ispis 1.10: Skripta *start\_dns\_my*

<sup>3</sup> BIND Manual: <http://www.bind9.net/manuals>

---

```

01  #! /bin/sh
02
03  error() {
04      echo $*
05      exit 2
06  }
07
08  dns_servers="dnsRoot dnsHr dnsFer dnsTel dnsZpm"
09  hosts=" zztMail FTP-TELNET-SSH www mm pc DHCP zpmMail"
10
11  vname=`himage -v dnsRoot`
12  if test $? -ne 0; then
13      exit 1
14  fi
15
16  for i in $dns_servers
17  do
18      eid=`himage -e $i`
19      if test $? -ne 0 ;then
20          echo "Cannot find node $i"
21          exit 2
22      fi
23  done
24
25  cd /root/Desktop/DNS_datoteke
26
27  for i in $dns_servers
28  do
29      himage $i hostname \
30          || error "Cannot find node $i. Is simulation started? \
31              Try: Experiment->Execute"
32      himage $i killall -9 named 2> /dev/null
33      dir= /var/named/etc/namedb
34      himage $i mkdir -p $dir
35      hcp $i/* $i:$dir
36      himage $i named -c /var/named/etc/namedb/named.conf
37  done
38
39  echo
40  echo Copy/Create resolv.conf on clients:
41  for i in $hosts
42  do
43      himage $i hostname \
44          || error "Cannot find node $i. Is simulation started? \
45              Try: Experiment->Execute"
46
47      hcp resolv.$i $i: /etc/resolv.conf
48  done
49
50
51
52

```

---

### 1.3.2 Primjer korištenja konfiguriranog sustava DNS

Nakon što pokrenemo eksperiment s modelom mreže, potrebno je pokrenuti skriptu koja će kopirati potrebne datoteke na poslužitelje i klijente. To se radi kroz sljedeće korake:

1. Otvoriti terminal OS-a *FreeBSD*

2. Pozicionirati se u direktorij gdje se nalazi skripta *start\_dns\_my*:

```
# cd /root/imagenes-examples/Config-scripts/DNS
```

3. Dodijeliti skripti dozvolu za izvršavanje:

```
# chmod +x start_dns_my
```

4. Pokrenuti skriptu:

```
# sh start_dns_my
```

Potom ćemo otvoriti konzolu na računalu *pc* i pomoću naredbe *host* saznati:

- a) IP-adresu računala *mm.tel.fer.hr*;
- b) IP-adresu poslužitelja *dnsFer*;
- c) IP-adresu poslužitelja *dnsRoot*;
- d) koji je poslužitelj nadležan za domenu *fer.hr*;
- e) koji je poslužitelj nadležan za domenu *tel.fer.hr*;
- f) koji je poslužitelj nadležan za domenu *hr*;
- g) koji je poslužitelj nadležan za domenu *root*.

Za detaljnije upute kako se koristi naredba *host* treba izvršiti naredbu *man host* ili pogledati primjere korištenja u tablici (Tablica 1.1).

Tablica 1.1: Opcije naredbe *host*

Naredba	Značenje naredbe
# host -t A mm.tel.fer.hr	traži se IP-adresa računala za koje je zadano njegovo ime
# host -t NS fer.hr	traži se nadležni DNS poslužitelj

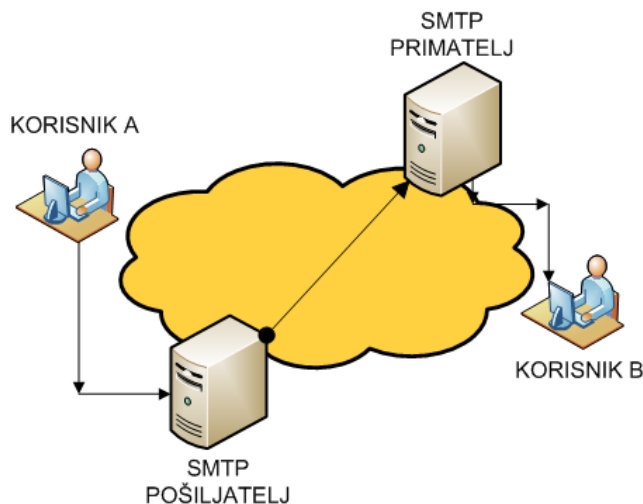
## 1.4 Konfiguriranje poslužitelja elektroničke pošte

U ovom poglavlju bit će objašnjeno konfiguriranje poslužitelja elektroničke temeljnog na protokolima SMTP (*Simple Mail Transfer Protocol*) i POP3 (*Post Office Protocol*). Protokol SMTP omogućava slanje elektroničke pošte od klijenta prema poslužitelju elektroničke pošte te međusobno između različitih poslužitelja elektroničke pošte. Protokolom POP3 klijent pristupa svom sandučiću elektroničke pošte na poslužitelju, dakle može pristupiti dolaznim porukama elektroničke pošte.

Protokol SMTP temelji se na sljedećem komunikacijskom modelu (Slika 1.17):

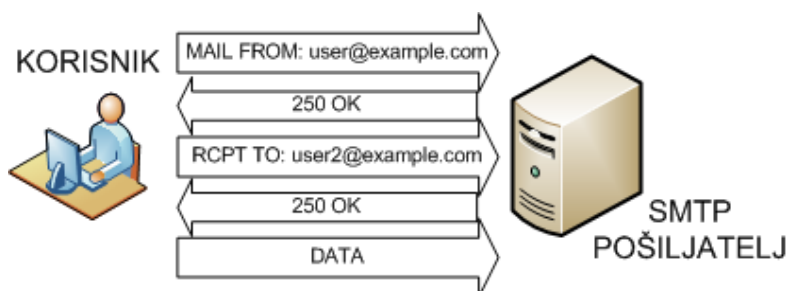
- a) Korisnik A šalje poruku elektroničke pošte korisniku B. Korisnikov MUA (engl. *Mail User Agent*) stvara SMTP-vezu sa SMTP-pošiljateljem i preko te veze šalje poruku;
- b) SMTP-pošiljatelj uspostavlja vezu sa SMTP-primateljem, tj. sa SMTP-poslužiteljem na kojem se nalazi poštanski sadužić korisnika kome je poruka namijenjena.

U oba koraka u komunikaciji pošiljatelj slijedno šalje određene SMTP-naredbe na koje primatelj mora vratiti odgovor. Tek ako je odgovor na neku naredbu pozitivan, može se prijeći na slanje sljedeće naredbe.



Slika 1.17: Model protokola SMTP

Prijenos poruke elektroničke pošte protokolom SMTP, kao što prikazuje Slika 1.18, počinje naredbom *MAIL* koja identificira pošiljatelja. Ukoliko SMTP-pošiljatelj prihvati poruku elektroničke pošte s danom adresom pošiljatelja, vraća korisniku odgovor *OK*. Nakon toga korisnik može prijeći na sljedeću fazu razmjene, slanjem naredbe *RCPT* u kojoj se nalazi odredišna adresa poruke elektroničke pošte. Ako SMTP-pošiljatelj ne prihvati odredišnu adresu, odbaci cijelu poruku. Tek kad primi pozitivan odgovor od SMTP-pošiljatelja na naredbu *RCPT*, korisnik može krenuti sa slanjem podataka koji sačinjavaju tijelo poruke koristeći naredbu *DATA*.



Slika 1.18: Primjer izmjene SMTP-naredbi između korisnika i SMTP-pošiljatelja

Kako mail putujući kroz mrežu prolazi kroz više poslužitelja SMTP, a komunikacija između njih jednaka je prikazanoj komunikaciji između korisnika i SMTP-pošiljatelja.

#### 1.4.1 Opis mrežnog okružja za konfiguraciju

Kao što je prikazano na modelu mreže koji ilustrira Slika 1.6, mreža se sastoji od dvije podmreže nazvane *zst* i *zpm*. U svakoj od tih podmreža potrebno je konfigurirati i pokrenuti SMTP-poslužitelje kako bi računala mogla preko njih razmjenjivati poruke.

Sljedeći čvorovi u mreži koriste se u konfiguraciji te su navedene i njihove uloge:

- *zztMail*, *zpmMail* – SMTP-poslužitelji za pojedinu domenu;
- *mm*, *pc* – osobna računala s kojih korisnici šalju mail i pristupaju svom poštanskom sandučiću;
- *dnsZpm*, *dnsTel* – DNS-poslužitelji koji se koriste za razlučivanje IP-adresa i dohvaćanje informacija o nadležnim poslužiteljima elektroničke pošte za pojedinu domenu.

Prije same konfiguracije i pokretanja poslužitelja u simulator IMUNES potrebno je učitati model mreže koji je objašnjen u uvodnom poglavlju (Slika 1.6) i pokrenuti simulaciju odabirom opcije *Execute* u izborniku *Experiments*.

#### 1.4.2 Konfiguriranje i pokretanje SMTP-poslužitelja

Postupak konfiguracije mreže sastoji se od tri koraka:

1. Provjera postojanja potrebnih paketa na sustavu;
2. Konfiguracija svih poslužitelja elektroničke pošte koji će predstavljati SMTP-poslužitelje u danoj mreži;
3. Konfiguracija svih poslužitelja elektroničke pošte, računala i sustava DNS u mreži.

##### Korak 1: Provjera postojanja potrebnih paketa na sustavu

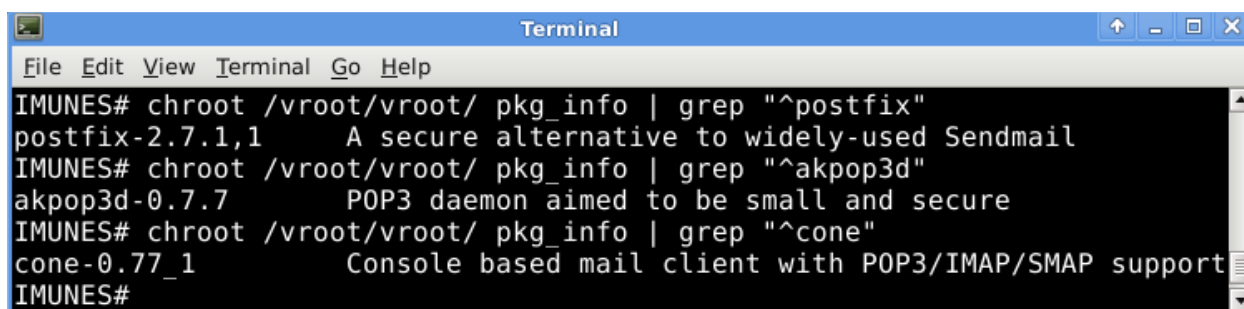
Na početku konfiguriranja potrebno je provjeriti nalaze li se u sustavu svi potrebni paketi za ispravno funkcioniranje SMTP-poslužitelja (*postfix*, *akpop3d* i *cone*). Paketi bi trebali već biti instalirani na novijim inačicama OS-a *FreeBSD*. Ukoliko se navedeni paketi ne nalaze na poslužitelju, potrebno ih je dohvatiti putem Interneta.

Provjera postojanja paketa radi se unosom sljedeće naredbe u OS *FreeBSD*-a:

```
# chroot /var/imunes/vroot pkg info
```

koja mijenja *root* direktorij na zadani i vraća listu paketa na virtualnom sustavu za zadani *root* koji koristi simulator IMUNES. Na listi paketa nakon izvršavanja komande, potrebno je uočiti pakete sustava *postfix*, *akpop3d*, *cone* (Slika 1.19).

Naredba *chroot* se ovdje koristi u kombinaciji s naredbom *grep* koja filtrira listu koja vraća naredba *chroot*, kako ne bi morali tražiti pakete kroz cijelu listu. Prilikom navođenja imena paketa, koristi se znak '^' prije samog imena paketa, kako bi se naglasilo da se traženi izraz nalazi na početku retka.



```
Terminal
File Edit View Terminal Go Help
IMUNES# chroot /vroot/vroot/ pkg_info | grep "^postfix"
postfix-2.7.1,1      A secure alternative to widely-used Sendmail
IMUNES# chroot /vroot/vroot/ pkg_info | grep "^akpop3d"
akpop3d-0.7.7       POP3 daemon aimed to be small and secure
IMUNES# chroot /vroot/vroot/ pkg_info | grep "^cone"
cone-0.77_1         Console based mail client with POP3/IMAP/SMTP support
IMUNES#
```

Slika 1.19: Filtrirana lista instaliranih paketa



**Korak 2: Konfiguracija svih poslužitelja elektroničke pošte koji će predstavljati SMTP-poslužitelje u danoj mreži**

Ukoliko neki od navedenih paketa ne postoji na sustavu, moguće ga je instalirati s naredbom:

```
# pkg_imunes install <ime paketa>
```

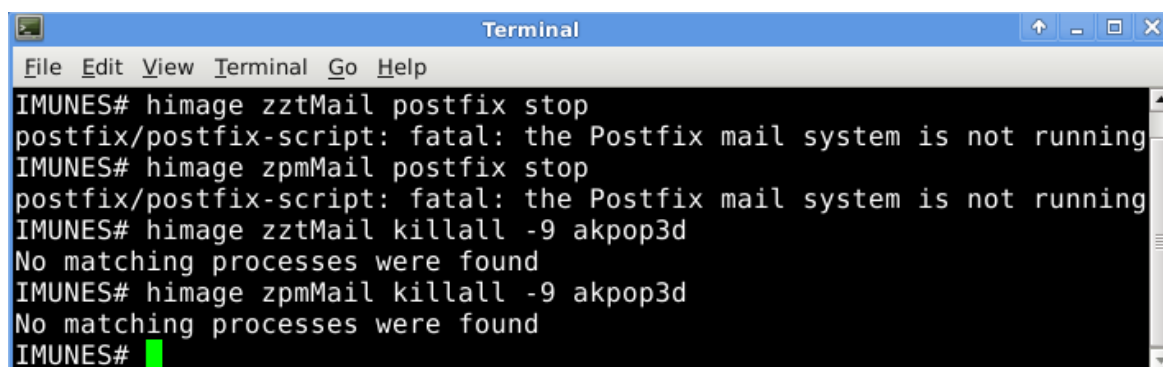
SMTP-poslužitelji u mreži bit će konfigurirani na poslužiteljima *zttMail* i *zpmMail*. Za svaki od odabranih poslužitelja potrebno je izvesti konfiguriranje na način kako je dalje objašnjeno.

Prvo treba pokrenuti eksperiment. Nakon toga treba zaustaviti procese sustava *postfix* i *akpop3d*, a to se radi sljedećim naredbama koje upisujemo u terminal OS-a *FreeBSD*:

```
# himage <ime poslužitelja> postfix stop
```

```
# himage <ime poslužitelja> killall -9 akpop3d
```

Kao što se vidi na sljedećoj slici (Slika 1.20), nijedan od procesa prilikom njihovih gašenja nije bio aktivan.



Slika 1.20: Zaustavljanje procesa

Zatim treba stvoriti direktorij u kojem će se nalaziti konfiguracijske datoteke potrebne aplikaciji *postfix*. U terminalu OS-a *FreeBSD* treba unijeti sljedeću naredbu (za stvaranje direktorija koristi se naredba *mkdir* čije opcije objašnjava Tablica 1.2):

```
# himage <ime poslužitelja> mkdir -p /var/db/postfix
```

Tablica 1.2: Opcije naredbe *mkdir*

Naredba	Značenje naredbe
# mkdir direktorij	stvara novi direktorij sa zadanim imenom
# mkdir -p /path/	stvara novi direktorij

Zatim je potrebno postaviti vlasnika novog direktorija i prava pristupa direktoriju (u tu svrhu koriste se naredbe `chown` i `chmod` čije značenje objašnjava

Tablica 1.3):

```
# himage <ime poslužitelja> chown postfix:wheel /var/db/postfix
# himage <ime poslužitelja> chmod 700 /var/db/postfix
```

Tablica 1.3: Značenje naredbi `chown` i `chmod`

Naredba	Značenje naredbe
# <code>chown vlasnik:grupa direktorij</code>	postavlja vlasnika zadanom direktoriju
# <code>chmod mode direktorij</code>	postavlja određena prava pristupa zadanom direktoriju <i>mode 700</i> - odobrava pisanje i čitanje u direktoriju isključivo njegovom vlasniku

Rezultat izvođenja naredbi za oba SMTP-poslužitelja prikazuje Slika 1.21.

```

Terminal
File Edit View Terminal Go Help
IMUNES# mkdir -p /vroot/i05391/n19/var/db/postfix/
IMUNES# mkdir -p /vroot/i05391/n4/var/db/postfix/
IMUNES# himage zpmMail chown postfix:wheel /var/db/postfix/
IMUNES# chmod 700 /vroot/i05391/n19/var/db/postfix/
IMUNES# himage zztMail chown postfix:wheel /var/db/postfix/
IMUNES# chmod 700 /vroot/i05391/n4/var/db/postfix/
IMUNES#
  
```

Slika 1.21: Stvaranje direktorija i postavljanje prava pristupa

Nakon što se zaustave procesi i stvori novi direktorij, slijedi konfiguriranje sustava postfix. Konfiguriranje započinje prebacivanjem konfiguracijskih datoteka koje dolaze sa simulatorom IMUNES na čvor na kojemu se konfigurira poslužitelj (to su datoteke iz primjera DNS+Mail+WEB koji se nalazi unutar direktorija `imunes-examples`) To se radi pomoću sljedećih naredbi (njihovo značenje objašnjava Tablica 1.4):

```
# cd /root/imunes-examples/DNS+Mail+WEB/Mail_files/postfix.<ime
poslužitelja>
# hcp * <ime poslužitelja>:/usr/local/etc/postfix/
```

Tablica 1.4: Objašnjenje naredbi za kopiranje konfiguracijskih datoteka postfix

Naredba	Značenje naredbe
# cd direktorij	postavljanje u zadani direktorij
# hcp * <ime poslužitelja>:/usr/local/etc/postfix/ destdir	kopiranje cijelog sadržaja direktorija u koji smo se pozicionirali u prethodnom koraku u direktorij na čvoru naveden parametrom <i>destdir</i>

Za potrebe ovog primjera, konfiguracijske datoteke poslužitelja *zpmMail* i *www* iz direktorija *DNS+Mail+Web* kopiraju se u konfiguracijske direktorije naših čvorova *zpmMail* i *zztMail* (Slika 1.22).

```

Terminal
File Edit View Terminal Go Help
IMUNES# cd /root/Examples/DNS+Mail+WEB/Mail_files/postfix.zpmMail/
IMUNES# tar -cf - * | (cd /vroot/i05391/n19/usr/local/etc/postfix/; tar -xf -)
IMUNES# cd /root/Examples/DNS+Mail+WEB/Mail_files/postfix.www/
IMUNES# tar -cf - * | (cd /vroot/i05391/n4/usr/local/etc/postfix/; tar -xf -)
IMUNES#

```

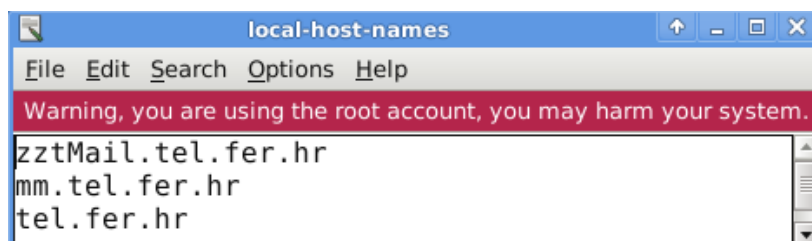
Slika 1.22: Kopiranje konfiguracijskih datoteka postfix u direktorije čvorova *zpmMail* i *zztMail*

Kako je naš poslužitelj *zpmMail* identičan onome iz primjera *DNS+Mail+Web*, njegove konfiguracijske datoteke nije potrebno dodatno podešavati. Međutim, kako su konfiguracijske datoteke poslužitelja *www* kopirane na poslužitelj *zztMail*, treba ih dodatno podesiti kako bi bile primjenjive za ciljani poslužitelj.

U alatu *Mousepad* uredit ćemo datoteke *local-host-names* i *main.cf* koje se nalaze u sljedećem direktoriju:

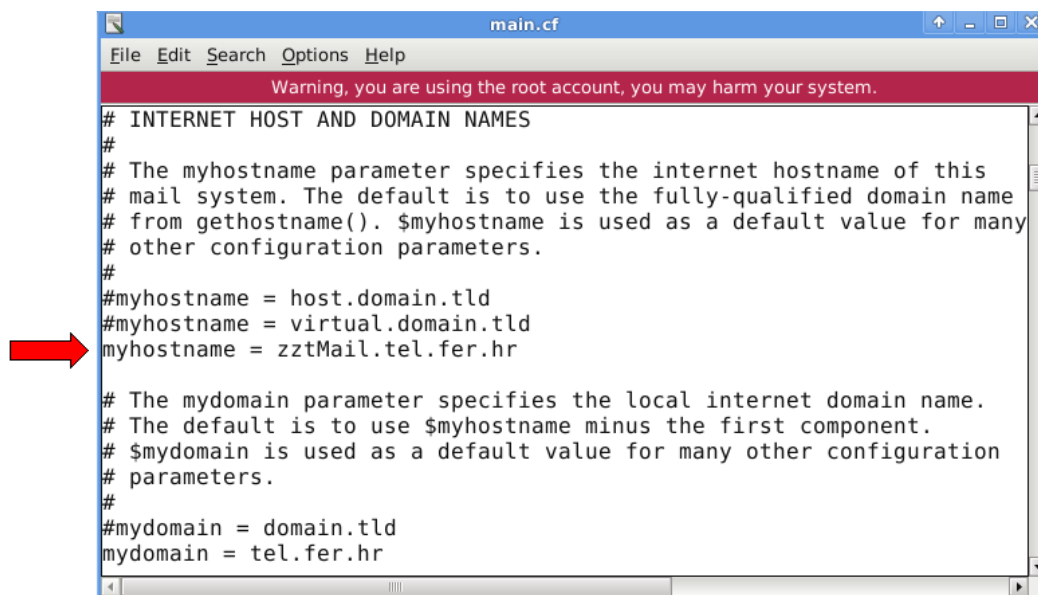
```
# /var/imunes/<eksperiment>/<čvor>/usr/local/etc/postfix/
```

Datoteka *local-host-names* sadrži simboličku adresu samog poslužitelja, računala u pod mreži i simboličku adresu pod mreže. Treba je izmijeniti tako da sadrže ispravna imena iz našeg modela mreže, a nakon izmjene trebala bi izgledati kako je prikazano na slici (Slika 1.23).



Slika 1.23: Datoteka *local-host-names*

U datoteci *main.cf* koja sadrži glavne parametre za funkcioniranje sustava elektroničke pošte, izmjenu treba napraviti u retku gdje je definirana varijabla *myhostname*. Na slici (Slika 1.24) označena je linija u kodu gdje je vrijednost varijable poprimila novu vrijednost (a to je *zztMail.tel.fer.hr* umjesto prijašnjeg *www.tel.fer.hr*).



Slika 1.24: Datoteka *main.cf*

Konfiguracijske datoteke poslužitelja *www* mogu se koristiti i za neke druge poslužitelje u mreži, ali ih je potrebno naknadno dodatno konfigurirati kako bi ih se povezalo s podmrežom u kojoj se nalaze.

U nastavku slijede naredbe koje konfiguriraju sustav *postfix* direktno na poslužitelju preko naredbe *himage* (objašnjene u poglavlju 1.7.3) na samom poslužitelju (naredba *touch* koristi se za kreiranje nove datoteke):

```
# himage <ime poslužitelja> touch /var/log/maillog
#      himage      <ime      poslužitelja>      chown      -R      root
/usr/local/etc/postfix/
```

Zatim se pokreće sustav *postfix*:

```
# himage <ime poslužitelja> postalias
hash:/usr/local/etc/postfix/aliases

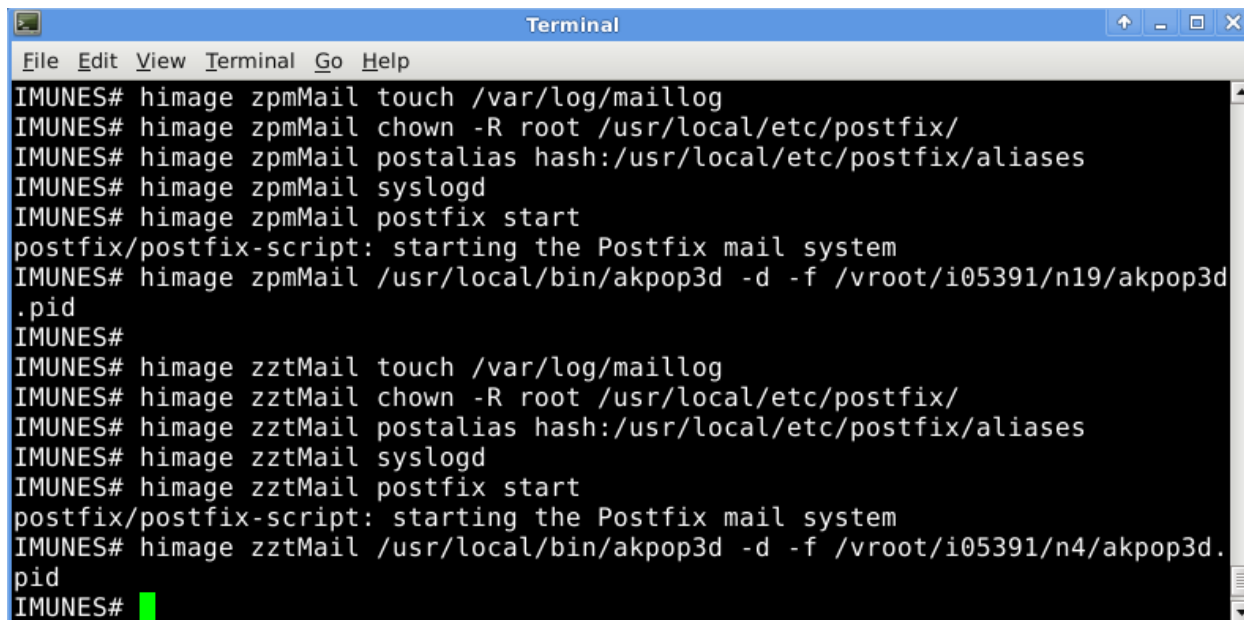
# himage <ime poslužitelja> syslogd

# himage <ime poslužitelja> postfix start
```

Nakon pokretanja sustava *postfix* slijedi pokretanje sustava *POP3*, što radimo sljedećim naredbama:

```
# himage <ime poslužitelja> /usr/local/bin/akpop3d -d -f  
/akpop3d.pid
```

Na slici (Slika 1.25) prikazan je rezultat završnog dijela konfiguriranja sustava *postfix* i njegovo pokretanje, kao i pokretanje sustava *POP3*.



```
Terminal  
File Edit View Terminal Go Help  
IMUNES# himage zpmMail touch /var/log/maillog  
IMUNES# himage zpmMail chown -R root /usr/local/etc/postfix/  
IMUNES# himage zpmMail postalias hash:/usr/local/etc/postfix/aliases  
IMUNES# himage zpmMail syslogd  
IMUNES# himage zpmMail postfix start  
postfix/postfix-script: starting the Postfix mail system  
IMUNES# himage zpmMail /usr/local/bin/akpop3d -d -f /vroot/i05391/n19/akpop3d  
.pid  
IMUNES#  
IMUNES# himage zztMail touch /var/log/maillog  
IMUNES# himage zztMail chown -R root /usr/local/etc/postfix/  
IMUNES# himage zztMail postalias hash:/usr/local/etc/postfix/aliases  
IMUNES# himage zztMail syslogd  
IMUNES# himage zztMail postfix start  
postfix/postfix-script: starting the Postfix mail system  
IMUNES# himage zztMail /usr/local/bin/akpop3d -d -f /vroot/i05391/n4/akpop3d.  
pid  
IMUNES#
```

Slika 1.25: Završno konfiguriranje i pokretanje sustava *postfix* i *POP3*

*Korak 3: Konfiguriranje svih poslužitelja i računala u mreži uključenih u komunikaciju elektroničke pošte*

Kad se postupak konfiguriranja sustava *postfix* i *POP3* ponovi za sve poslužitelje elektroničke pošte, ostaje još kopirati konfiguracijski direktorij sustava *cone* na sva računala i poslužitelje u mreži, kako bi omogućili razmjenu poruka elektroničke pošte između njih.

Poslužitelji i računala koja sudjeluju u komunikaciji elektroničkom poštom, i koje je potrebno konfigurirati, su: *zpmMail*, *pc*, *dnsZpm*, *zztMail*, *mm* i *dnsTel*.

Kako je mreža podijeljena na podmreže ZPM i ZZT, za svaku od njih imamo i odvojene konfiguracijske datoteke koje trebamo dohvatiti iz primjera „DNS+Mail+Web“.

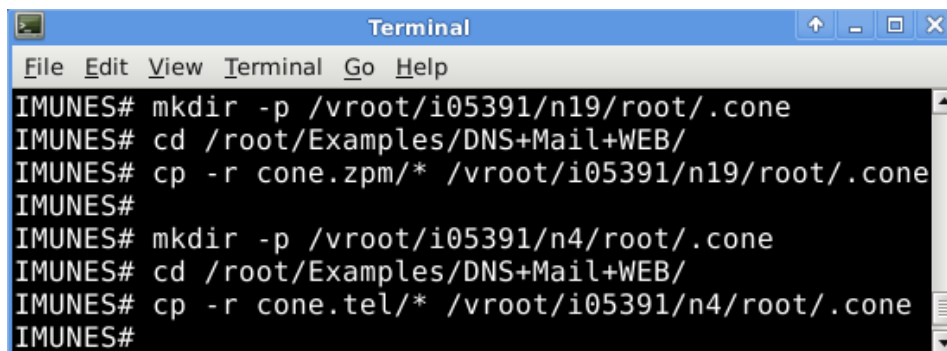
Potrebno je stvoriti direktorij na svakom čvoru i zatim u njega kopirati odgovarajuću konfiguracijske datoteke iz odgovarajućeg direktorija (direktoriji *cone.tel* ili *cone.www*).

*Poslužitelji i računala iz podmreže ZPM našeg primjera i poslužitelji i računala iz podmreže ZPM primjera DNS+Mail+Web jednaki su, stoga će i njihove konfiguracijske datoteke biti jednake. Iz istog razloga koristimo i konfiguracijske datoteke podmreže ZZT iz primjera DNS+Mail+Web.*

*U trećem dijelu konfiguracije mreže, u konfiguraciju poslužitelja ubrajaju se i DNS-poslužitelji objašnjeni u trećem poglavlju, koji se nalaze u podmrežama ZZT i ZPM (dnsTel, dnsZpm).*

```
# himage <čvor> mkdir -p /root/.cone  
# hcp -r cone.<ime pod mreže>/* <čvor>:/root/.cone
```

Izvođenje ovih naredbi za poslužitelje *zpmMail* i *zttMail* prikazano je na slici (Slika 1.26).



```
Terminal  
File Edit View Terminal Go Help  
IMUNES# mkdir -p /vroot/i05391/n19/root/.cone  
IMUNES# cd /root/Examples/DNS+Mail+WEB/  
IMUNES# cp -r cone.zpm/* /vroot/i05391/n19/root/.cone  
IMUNES#  
IMUNES# mkdir -p /vroot/i05391/n4/root/.cone  
IMUNES# cd /root/Examples/DNS+Mail+WEB/  
IMUNES# cp -r cone.tel/* /vroot/i05391/n4/root/.cone  
IMUNES#
```

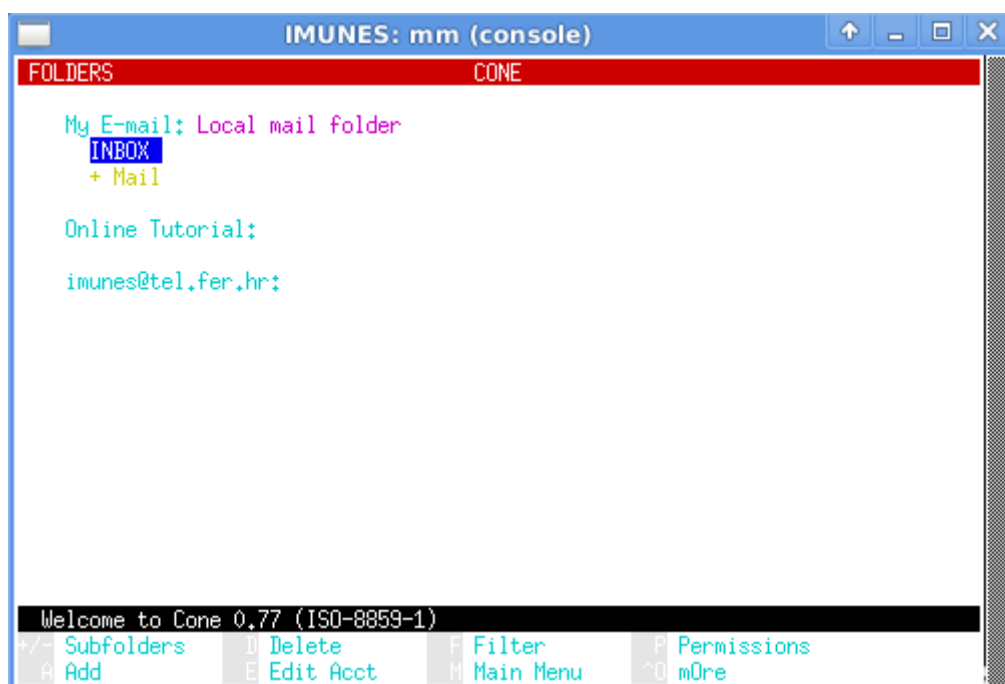
Slika 1.26: Konfiguracija sustava cone

Kopiranje ove konfiguracijske datoteke na ostala računala u mreži (*pc*, *dnsZpm*, *mm*, *dnsTel*) jednak je već gore objašnjenom principu kopiranja (Slika 1.26).

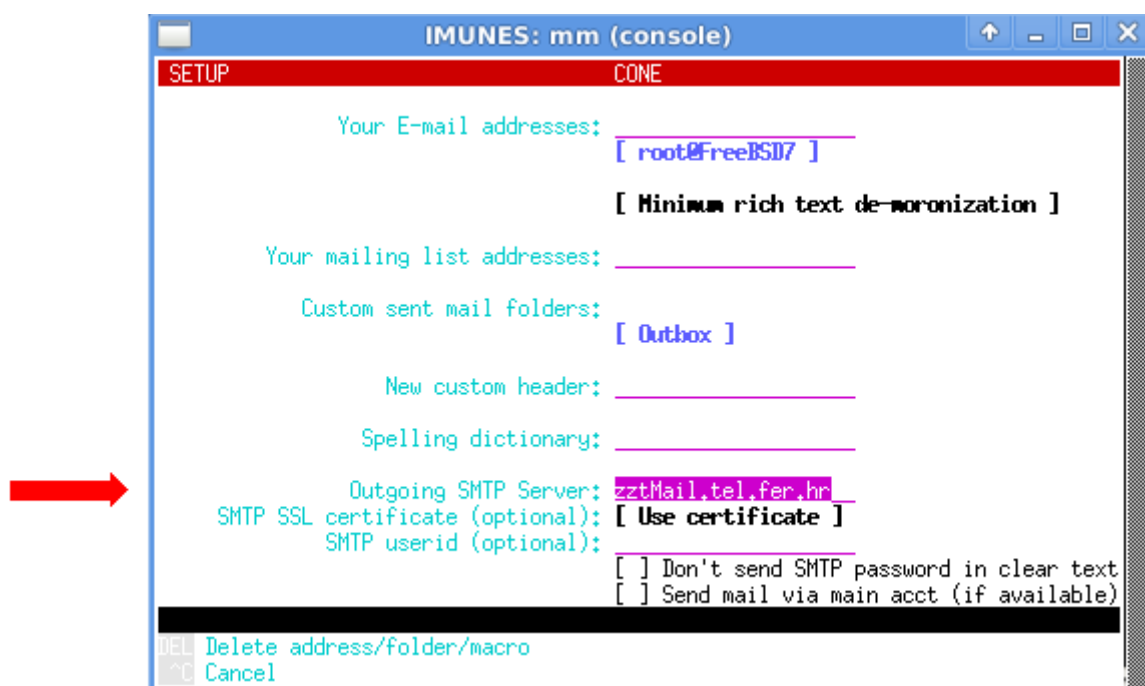
Na samom kraju konfiguriranja ostalo je još podesiti adresu SMTP-poslužitelja za svako računalo u mreži. Adresa se postavlja u programu *cone* koji se pokreće u ljusci na računalu koje se konfigurira, naredbom *cone*. Nakon izvršavanja naredbe na zaslonu se pojavljuje programsko sučelje za razmjenu elektroničke pošte (Slika 1.27).

Na tipkovnici odabrati tipku *M* (koja odgovara pozivu glavnog izbornika programa), a zatim tipku *S* (koja otvara prozor za konfiguriranje programa).

Za svaku od pod mreža (ZZT i ZPM) postoji jedan poslužitelj elektroničke pošte (*zttMail* i *zpmMail*) čije je adrese potrebno upisati u opciju *Outgoing SMTP Server* na svakom računalu u mreži sa kojeg se šalje *mail*. Kao u primjeru prikazanom na slici (Slika 1.28), na računalu *mm* u podešavanjima za *Outgoing SMTP Server* potrebno je upisati adresu *zttMail.tel.fer.hr*, koja predstavlja adresu SMTP-poslužitelja za pod mrežu ZZT i potom pritisnuti tipku *SAVE*.



Slika 1.27: Programsko sučelje cone pokrenuto na računalu mm



Slika 1.28: Podešavanje adrese SMTP-poslužitelja na računalu mm

Nakon upisivanja adrese SMTP poslužitelja na svim računalima, završen je postupak konfiguriranja računala i poslužitelja u mreži za omogućavanje razmjene elektroničke pošte.

#### 1.4.3 Primjer slanja elektroničke pošte

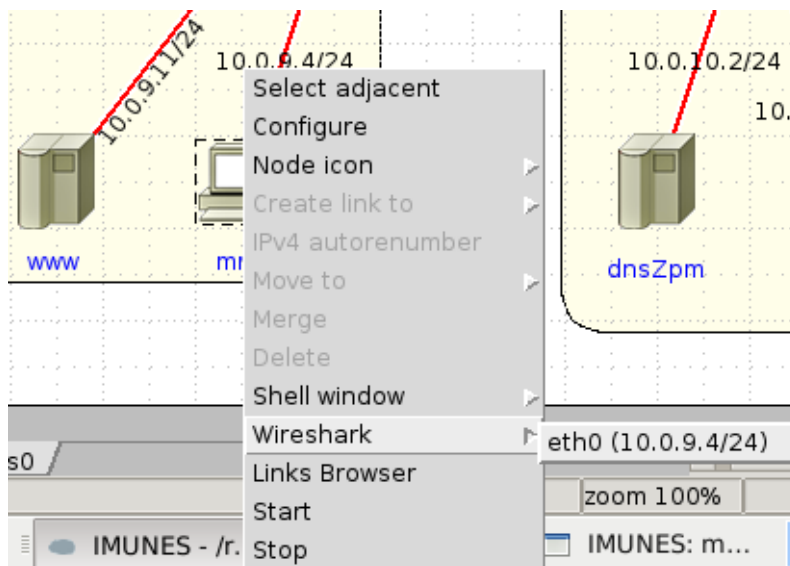
Kako bi pokazali da slanje elektroničke pošte funkcionira u mreži, nakon obavljenih koraka iz uputa za konfiguriranje poslužitelja i računala iz prethodnih poglavlja, elektronička se pošta može



poslati s jednog računala na drugo. Kako bismo omogućili slanje elektroničke pošte, DNS-poslužitelji moraju biti konfigurirani i aktivni prije izvođenja ovog primjera.

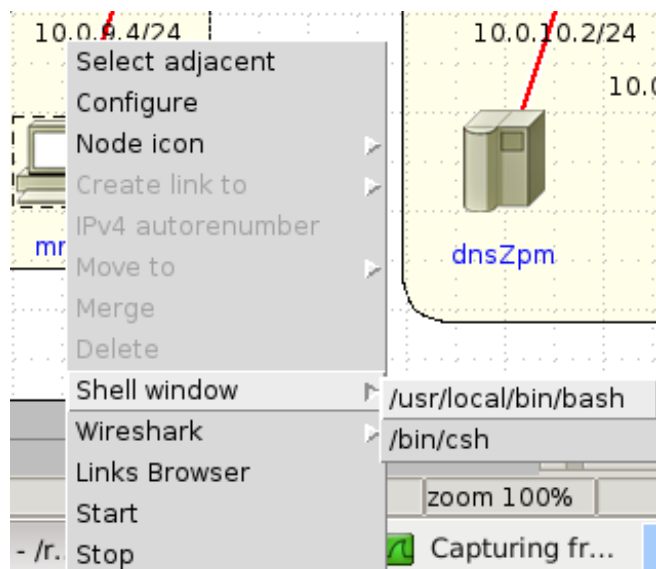
S računala *mm* poslat ćemo poruku elektroničke pošte na adresu *imunes@zpm.fer.hr* s proizvoljnim sadržajem. Pomoću alata *Wireshark* pratit ćemo promet na računalu *mm* i uočiti naredbe koje razmjenjuju SMTP-poslužitelj i klijentsko računalo.

Na početku scenarija pokrenut ćemo alat *Wireshark* na računalu *mm*. Desnim klikom na računalo *mm* u simulatoru *IMUNES* otvara se izbornik u kojem je potrebno odabrati opciju *Wireshark* i zatim sučelje *eth0* (Slika 1.29).



Slika 1.29: Pokretanje programa *Wireshark*

Nakon pokretanja *Wiresharka*, pokrenut ćemo konzolu na računalu *mm* (Slika 1.30).



Slika 1.30: Pokretanje ljuške na računalu *mm*

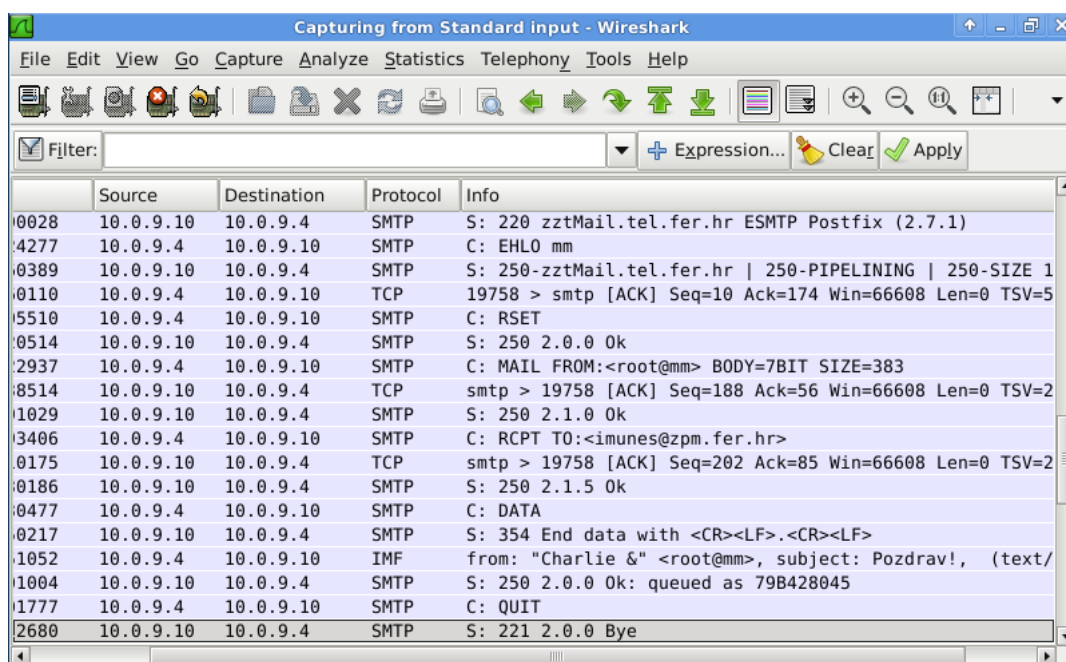


Upisivanjem naredbe `cone` u konzoli, pokreće se program `cone`. Na tipkovnici je potrebno odabrati tipku *M* (koja odgovara pozivu glavnog izbornika), a zatim tipku *W* (koja odgovara pozivu opcije za sastavljanje nove elektroničke poruke). Nakon što se popune potrebna zaglavlja *To* i *Subject* te se poruka napiše (Slika 1.31), ista se šalje odabirom kombinacije tipki *Ctrl+X*.



Slika 1.31: Primjer elektroničke pošte koja se šalje s računala mm

Ako je poruka uspješno poslana na odredište, na dnu prozora programskog sučelja pojavljuje se poruka *250 OK*.



Slika 1.32: Tijek SMTP-poruka zabilježen prilikom slanja elektroničke poruke s računala mm

Na početku zabilježenog protoka u alatu *Wireshark* vidi se slanje DNS zahtjeva za adresu *zztMail.tel.fer.hr* DNS-poslužitelju od strane računala *mm*. Na taj se način saznaje IP-adresa poslužitelja elektroničke pošte *zztMail*, potrebna za ostvarivanje veze između poslužitelja i računala *mm*. Nakon toga dolazi do razmjene niza naredbi između poslužitelja i klijenta.

Kako je objašnjeno u uvodnom dijelu uputa za konfiguraciju poslužitelja elektroničke pošte, prilikom slanja maila dolazi do razmjene SMTP-naredbi između klijenta i poslužitelja. Te se naredbe mogu vidjeti na slici (Slika 1.32) među podacima koje je zabilježio alat *Wireshark*.

Najprije se klijent identificira naredbom *EHLO*. Na svaku klijentovu naredbu poslužitelj šalje odgovor *250 OK* ukoliko ju je poslužitelj odobrio. Nakon toga slijedi naredba *MAIL FROM* koja postavlja adresu pošiljatelja i ujedno služi SMTP-poslužitelju da resetira sva svoja stanja i spremnike. Naredba *RCPT TO* postavlja adresu primatelja i ona može biti ponovljena više puta ako postoji više primatelja iste poruke. Naredba *DATA* započinje razmjenu podataka i završava se određenim rasporedom znakova koji je definirao poslužitelj za signaliziranje kraja razmjene. Kao potvrdu kraja razmjene klijent šalje naredbu *QUIT*, na što dobiva odgovor poslužitelja u poruci *221 Bye*.

## 1.5 Konfiguriranje HTTP-poslužitelja

HTTP (*HyperText Transfer Protocol*) je protokol aplikacijskog sloja koji pruža uslugu prijenosa web sadržaja između računala. Izvorno je HTTP bio namijenjen za prijenos tzv. hiperteksta, ali danas ima gotovo univerzalnu primjenu – koristi se za prijenos datoteka (umjesto protokola FTP), za web usluge (npr., *web-mail*) i sl.

### 1.5.1 Konfiguriranje i pokretanje HTTP-poslužitelja

HTTP-poslužitelj će biti konfiguriran i pokrenut na čvoru *www* s IP-adresom 10.0.9.11 (Slika 1.6). Funkciju HTTP-poslužitelja na tom čvoru obavljat će aplikacija *lighttpd* [<http://www.lighttpd.net/>]. *Lighttpd* je web poslužitelj otvorenog koda optimiziran za izvođenje na slabijim računalima koji imaju malu procesorsku snagu i malo radne memorije. Konfiguracija poslužitelja *lighttpd* odvija se u nekoliko koraka:

- |         |   |
|---------|---|
| Korak 1 | Stvaranje i priprema odgovarajućih direktorija;                                   |
| Korak 2 | Kopiranje i prilagodba konfiguracijske datoteke HTTP-poslužitelja i web sadržaja; |
| Korak 3 | Pokretanje HTTP-poslužitelja.   |

#### Korak 1: Stvaranje i priprema odgovarajućih direktorija

Nakon pokretanja eksperimenta potrebno je kliknuti na čvor *www* desnom tipkom miša i odabrati *Shell window > usr/local/bin/bash*, čime se otvara konzola tog čvora. Odgovarajuće direktorije ćemo na HTTP-poslužitelju stvoriti upisivanjem sljedećih naredbi u konzoli:

```
# mkdir -p /usr/local/etc/lighttpd/
```

(Direktorij za pohranjivanje konfiguracijske datoteke HTTP-poslužitelja.)

```
# mkdir -p /var/log/lighttpd/
```

(Direktorij za spremanje *log* datoteka HTTP-poslužitelja, opcija `-p` stvara direktorij s automatski postavljenim svim dozvolama, tj., za čitanje, pisanje i izvršavanje.)

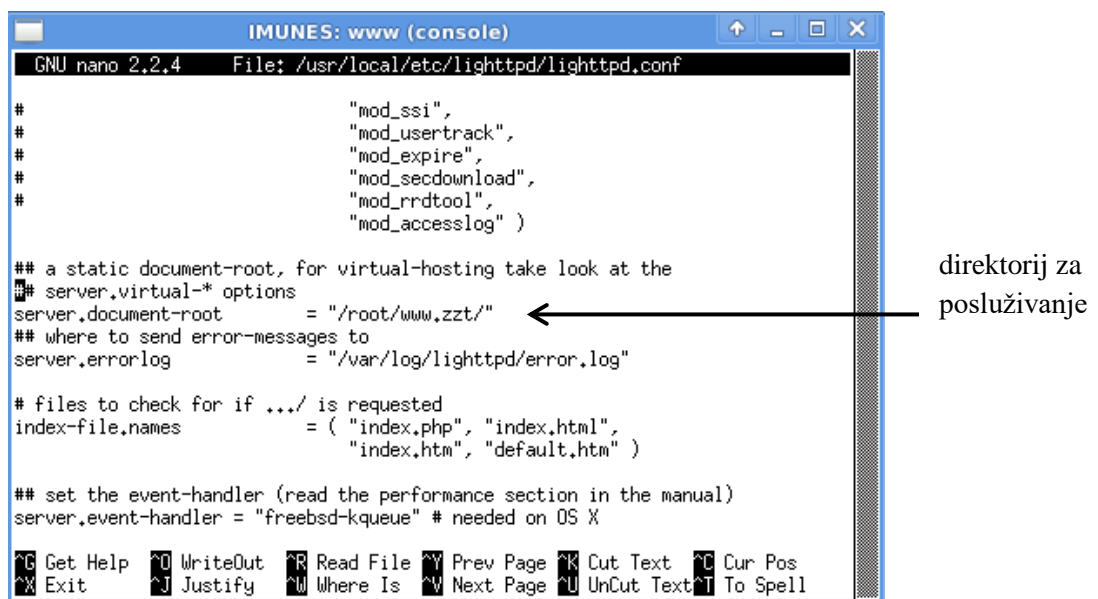
Da bi HTTP-poslužitelj mogao koristiti direktorij `/var/log/lighttpd/` za spremanje *log* datoteka, potrebno je taj direktorij dodijeliti korisniku `www` (`www` je ime korisnika pod kojim HTTP-poslužitelj vrši manipulacije diskovnim prostorom) naredbom:

```
# chown -R www:www /var/log/lighttpd
```

## Korak 2: Kopiranje i priprema konfiguracijske datoteke HTTP-poslužitelja i web sadržaja

U ovom primjeru kopirat ćemo već gotovu konfiguracijsku datoteku u mapu `/usr/local/etc/lighttpd/`. U tu svrhu koristi se konfiguracijska datoteka `www.lighttpd.conf` koja se nalazi u direktoriju `DNS+Mail+WEB/WEB_files`. Kopiranje datoteke na računalo `www` izvest će se pomoću naredbe `hcp` čije je korištenje opisano na početku udžbenika. Kopiranje se provodi unosom sljedeće naredbe u konzolu OS-a *FreeBSD*:

```
# hcp /root/imagenes-
examples/DNS+Mail+WEB/WEB_files/www.lighttpd.conf
www:/usr/local/etc/lighttpd/lighttpd.conf
```



Slika 1.33: Konfiguracijska datoteka HTTP-poslužitelja `zzt.lighttpd.conf`

Nakon kopiranja konfiguracijske datoteke, potrebno je kreirati direktorij u kojoj će se nalaziti HTML-datoteke koje će poslužitelj *lighttpd* staviti na posluživanje. U predefiniranoj konfiguracijskoj datoteci koju ovdje koristimo taj se direktorij zove `www.www`. Ukoliko se želi koristiti neki drugi direktorij, potrebno je promijeniti parametar `server.document-root` u konfiguracijskoj datoteci (Slika 1.33).

Direktorij `www.www` kreirat ćemo unosom sljedeće naredbe u konzolu čvora `www`:

```
# mkdir /root/www.www
```

Ovdje ćemo definirati jednu HTML-datoteku `index.html` koja će se moći dohvatiti s HTTP-poslužitelja. Prvo ćemo kreirati datoteku i otvoriti je:

```
# nano /root/www.www/index.html
```

Nakon otvaranja datoteke unijet ćemo sadržaj HTML-stranice. Ispis 1.11 prikazuje primjer jedne takve stranice.

*Ispis 1.11: Sadržaj datoteke index.html*

---

```
01 <html>
02 <head></head>
03 <body>The web server is running!</body>
04 </html>
```

---

Nakon što su napravljene sve promjene na konfiguracijskoj datoteci, onemogućit ćemo daljnje mijenjanje njezinog sadržaja radi sigurnosnih razloga. To se radi unosom sljedeće naredbe u konzolu čvora `www`:

```
# chmod 755 /usr/local/etc/lighttpd/lighttpd.conf
```

### *Korak 3: Pokretanje HTTP-poslužitelja*

Pokretanje HTTP-poslužitelja *lighttpd* obavlja se unosom sljedeće naredbe u konzolu čvora `www`:

```
# lighttpd -f /usr/local/etc/lighttpd/lighttpd.conf
```

Isto kao i kod konfiguracijskih postupaka opisanih u prethodnim poglavljima, svi se direktoriji čvorova u eksperimentu brišu svaki put prilikom zaustavljanja eksperimenta. Napredniji korisnici mogu skratiti korake konfiguracije izradom skripte čijim se izvođenjem automatski konfiguriraju svi potrebni parametri.

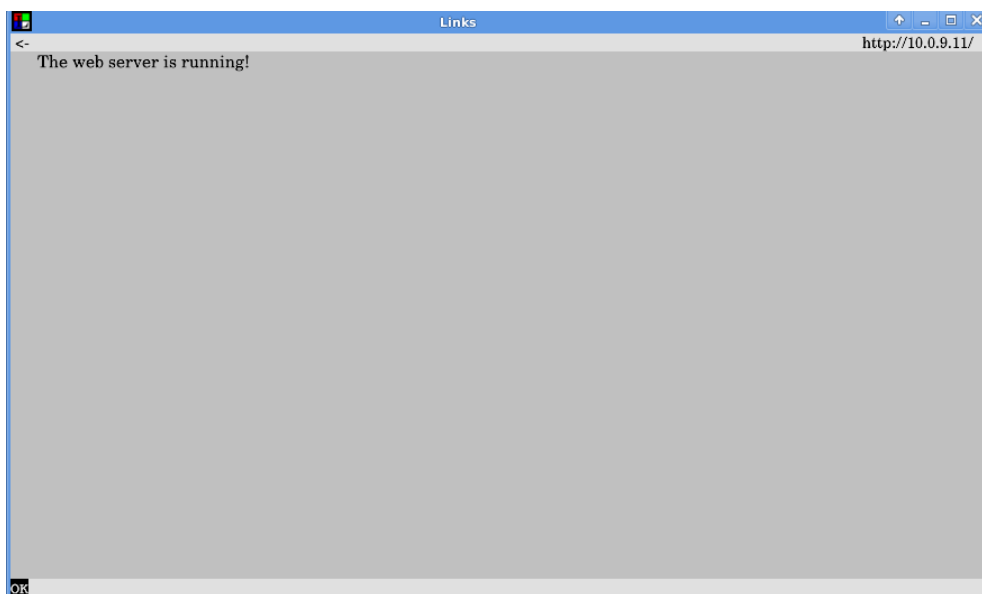
Primjer jedne takve skripte nalazi se u direktoriju `/root/itunes-examples/DNS+Mail+WEB/start_http`. Pokreće se pozicioniranjem u direktorij `DNS+Mail+WEB` u konzoli OS-a *FreeBSD* te unosom naredbe `./start_http`.

## *1.5.2 Primjer spajanja na HTTP-poslužitelj*

Ukoliko je HTTP-poslužitelj uspješno pokrenut, sadržaju se može pristupiti s računala unosom IP-adrese HTTP-poslužitelja (10.0.9.11) ili unosom domenskog imena HTTP-poslužitelja

([www.tel.fer.hr](http://www.tel.fer.hr)). U slučaju da se koristi drugi pristup, u mreži moraju biti ispravno konfigurirani i pokrenuti DNS-poslužitelji.

Poslužitelju ćemo u ovom primjeru pristupiti s računala `pc`. Preglednik weba pokreće se desnim klikom na računalu `pc` i odabirom opcije *Web Browser*. Nakon otvaranja preglednika potrebno upisati URL ili IP-adresu poslužiteljskog računala. Slika 1.34 prikazuje konačan rezultat pristupa web-poslužitelju.



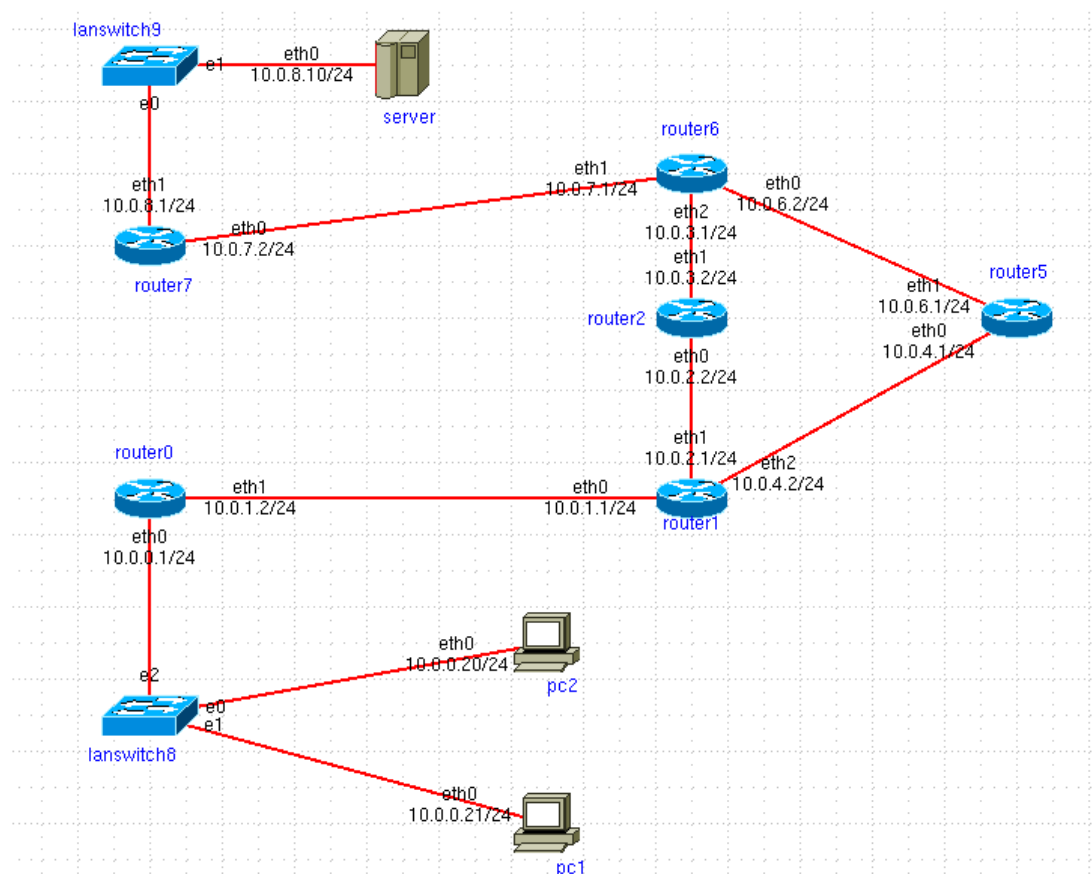
*Slika 1.34: Rezultat pristupa poslužitelju preko web-preglednika*

## 2 Protokoli sloja podatkovne poveznice

Protokoli sloja podatkovne poveznice te posebno protokol Ethernet objašnjeni su u udžbeniku *Komunikacijske mreže*<sup>4</sup>. Tamo je, također, objašnjen pojam i uloga MAC-adrese.

### 2.1 Konfiguracija eksperimenta

Na slici 2.1 prikazana je topologija *Ping/ping.imn*, koja će se u ovom poglavlju koristiti za upoznavanje s alatom *Wireshark*, analizatorom mrežnog prometa, te rješavanje zadataka vezanih uz standard Ethernet i strukturu okvira koji on definira. Topologija *Ping/ping.imn* se sastoji od dva krajnja računala, čvorova *pc1* i *pc2*, jednog poslužitelja (engl. *server*), čvora *server*, dva LAN-komutatora (engl. *LAN switch*), *lanswitch8* i *lanswitch9*, te šest usmjeritelja (engl. *router*), koji su zaduženi za prijenos IP-datagrama između krajnjih čvorova. Topologija *Ping/ping.imn* ne iziskuje dodatnu konfiguraciju eksperimenta niti korištenje pomoćnih alata, već se svi zadaci iz ovog poglavlja mogu riješiti analizom, primjerice, mrežnog prometa koji periodički razmjenjuju usmjeritelji za potrebe procesa usmjeravanja.



Slika 2.1: Topologija *Ping/ping.imn*

<sup>4</sup> Lovrek, Matijašević, Ježić, Jevtić: “Komunikacijske mreže”, Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva (2019) (trenutno dostupna radna inačica udžbenika)

## 2.2 Objašnjenja korištenih pojmova, koncepata, protokola i alata

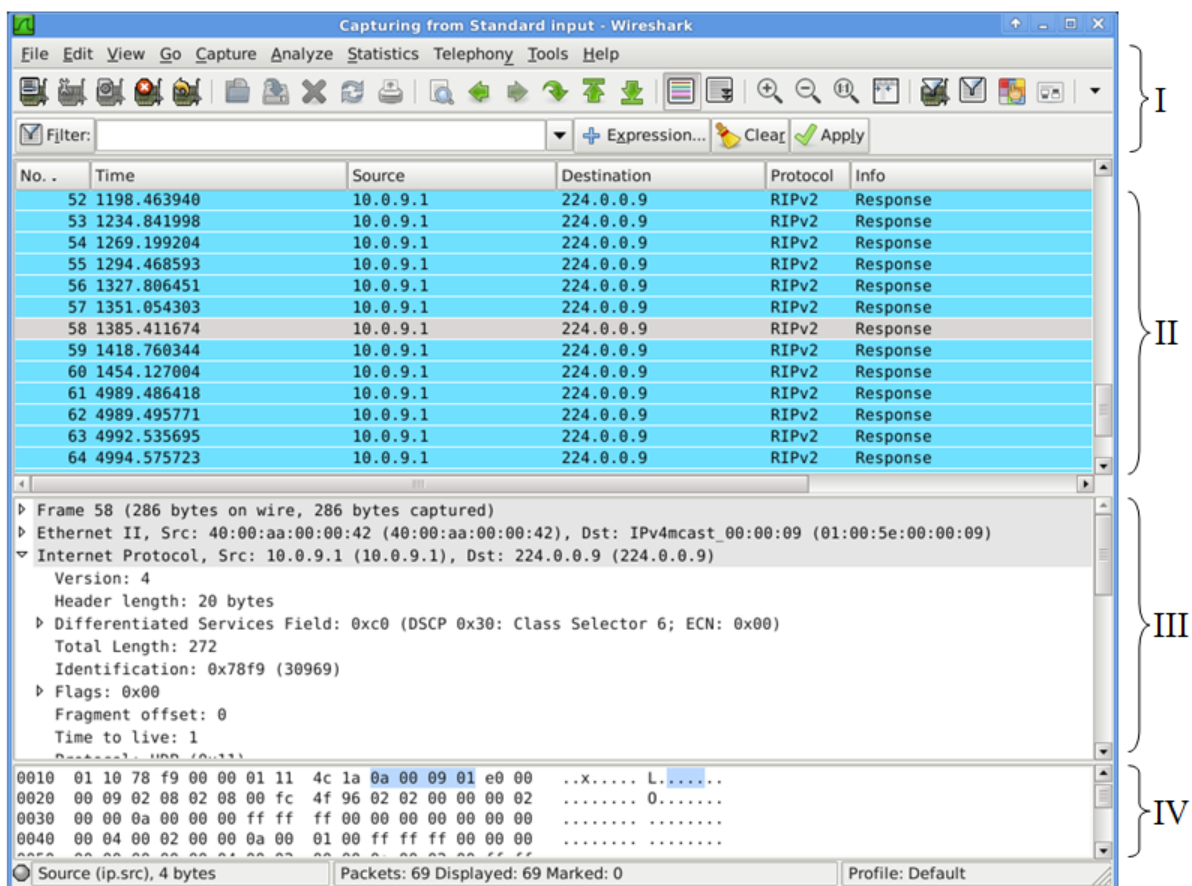
### 2.2.1 Standard Ethernet

Žičane lokalne mreže najčešće su izvedene standardom Ethernet, odnosno IEEE 802.3, koji zapravo obuhvaća niz specifikacija vezanih uz fizički sloj i MAC-podslaj sloja podatkovne poveznice. Ethernet definira posebnu strukturu okvira, koja se sastoji od preambule, oznake početka okvira, Ethernet-zaglavlja, polja podataka i zaštitne sume okvira. Ethernet-zaglavlje, između ostalog, sadrži izvorišnu i odredišnu MAC-adresu okvira.

### 2.2.2 Alat Wireshark

S ciljem detaljnijeg proučavanja načina rada te analize funkcionalnosti protokola i aplikacija karakterističnih za internetske mreže, potreban nam je alat za praćenje i analizu prometa u mreži. Postoji velik broj alata s tom namjenom kao što su *Tcpdump*, *Wireshark* (prijašnji naziv ovog alata bio je *Ethereal*) i *Snoop*. Budući da IMUNES koristi alat *Wireshark*, slijedi nekoliko osnovnih naputaka vezanih uz njegovo korištenje.

Alat *Wireshark* u emulatoru/simulatoru IMUNES pokreće se pritiskanjem desnog gumba miša nad računalom na kojem se želi pratiti mrežni promet, zatim izborom stavke *Wireshark* te izborom sučelja na kojem se želi prikupljati promet. Snimanje mrežnog prometa započinje odmah po pokretanju programa. Prozor alata (Slika 2.2) se sastoji od četiri dijela.



Slika 2.2: Sučelje alata Wireshark

Dio I prikazuje alatnu traku. Dio II prikazuje snimljene protokolne jedinice podataka – njihove redne brojeve, relativna vremena prikupljanja, izvorišne i odredišne adrese te oznake protokola i dodatne informacije specifične za pojedini protokol. Sadržaj odabrane podatkovne jedinice prikazan je u dijelu III, pri čemu se vide nazivi pojedinih protokola, a odabirom strelice pored naziva protokola pojavljuju se vrijednosti pojedinih polja za odabrani protokol (napomena: poredak protokola je obrnut od uobičajenog prikaza OSI-modela tako da se fizički sloj nalazi na vrhu, sloj podatkovne poveznice je ispod njega, itd.). U dijelu IV prikazan je sadržaj pojedine podatkovne jedinice onako kako se „vidi“ na mediju, tj. prikazani su pojedini bitovi zapisani u heksadekaskom obliku.

## 2.3 Eksperimenti i zadaci

Korištenjem alata *Wireshark* u emulatoru/simulatoru IMUNES riješite sljedeće zadatke.

**Zadatak 1.** (Topologija Ping/ping.imn) Analizirajte izvorišnu MAC-adresu iz proizvoljno odabranog Ethernet-okvira te odredite dijelove adrese koji se odnose na organizacijski jednoznačni identifikator (OUI) i identifikator mrežnog sučelja (NIC). Za odabranu MAC-adresu pokušajte utvrditi proizvođača pripadajuće mrežne kartice korištenjem web-tražilice.

**Zadatak 2.** (Topologija Ping/ping.imn) Proizvoljno odaberite jedan Ethernet-okvir i utvrdite veličinu njegovog zaglavlja. Skicirajte strukturu Ethernet-okvira te ju usporedite s prikazom odabranog okvira u alatu *Wireshark*. Koja polja prikazanog okvira prepoznajete? (Za pojašnjenje prikaza okvira u alatu *Wireshark*, koristite web-stranicu: <http://wiki.wireshark.org/Ethernet>)

**Zadatak 3.** (Topologija Ping/ping.imn) Na koji način protokol Ethernet „pamti“ vrstu paketa koji se prenosi u podatkovnom dijelu Ethernet-okvira?

## 2.4 Pitanja

Odgovorite na sljedeća pitanja.

**Pitanje 1.** Kod protokola CSMA/CD, stanica koja se sprema poslati okvir na medij će:

- odmah početi slanje okvira ako ustanovi da je medij slobodan.
- provjeriti je li medij slobodan, pričekati da istekne vrijeme razmaka između okvira (IFG), te početi slati okvir.
- odmah početi slati okvir, bez provjere stanja medija.
- prije slanja odaslati signal zagušenja kako bi se uvjerila da će sve stanice doista primiti poslani okvir, pa tek onda slati okvir.

**Pitanje 2.** Koja je od sljedećih adresa ispravna MAC-adresa pisana u standardnom obliku:

- 00:0c:a4:f2:ff:ff
- 161.53.19.51
- 161.53.19.0
- ff:ff:ff:ff:ff:ff



**Pitanje 3.** S porastom frekvencije signala, gušenje u bakrenom kablju:

- a. raste.
- b. pada.
- c. ne mijenja se.

**Pitanje 4.** Koja od navedenih karakteristika nije karakteristika lokalne mreže (LAN-a)?

- a. Mreža je obično u vlasništvu jedne organizacije.
- b. Koriste se velike prijenosne brzine, veće od 1 Mbit/s.
- c. Moguće je umrežiti neograničen broj računala.
- d. Za komunikaciju se koristi dijeljeni medij.

**Pitanje 5.** Koji sloj referentnog modela OSI obuhvaća mehaničke, električne, funkcijske i proceduralne karakteristike sučelja za pristup fizičkom mediju?

- a. Fizički sloj.
- b. Sloj podatkovne poveznice.
- c. Mrežni sloj.
- d. Prezentacijski sloj.

## 2.5 Izvori

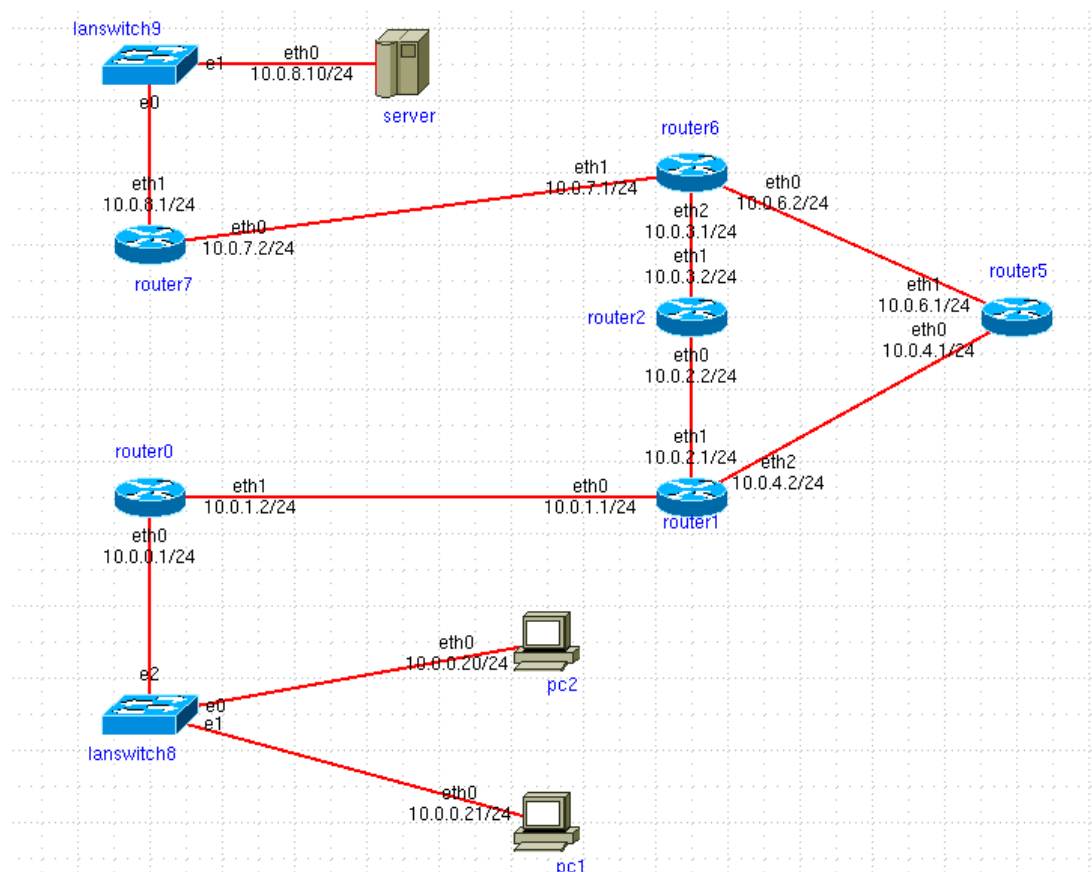
- IEEE 802.3 Ethernet Working Group: <http://www.ieee802.org/3>
- IEEE 802.3 - Standard for Ethernet:  
<http://standards.ieee.org/about/get/802/802.3.html>
- Ethernet Technologies - DocWiki:  
[http://docwiki.cisco.com/wiki/Ethernet Technologies](http://docwiki.cisco.com/wiki/Ethernet_Technologies)
- Tcpdump: <http://www.tcpdump.org>
- Wireshark: <http://www.wireshark.org>
- Snoop: <http://download.oracle.com/docs/cd/E19683-01/817-0675/6mgf7e58t/index.html>

### 3 Protokoli mrežnog sloja

Protokoli mrežnog sloja internetske mreže te posebno protokoli IP i ICMP objašnjeni su u udžbeniku *Komunikacijske mreže*<sup>5</sup>. Tamo je, također, objašnjen pojam i uloga IP-adrese, kao i pomoćnog protokola ARP.

#### 3.1 Konfiguracija eksperimenta

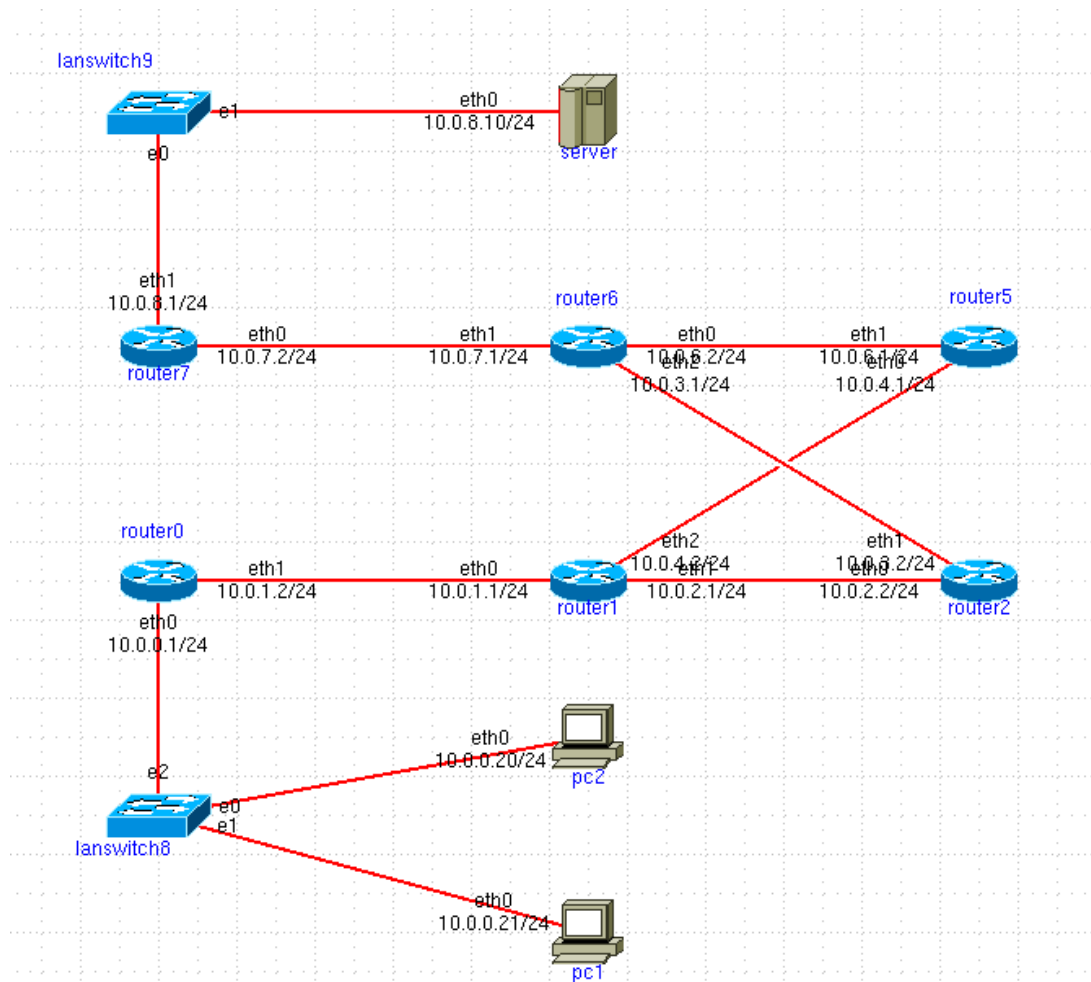
Na slici 3.1 prikazana je topologija *Ping/ping.imn*, koja će se u ovom poglavlju koristiti za ilustraciju funkcija protokola IP (*Internet Protocol*) te primjene pomoćnih protokola ICMP (*Internet Control Message Protocol*) i ARP (*Address Resolution Protocol*), i to kroz korištenje dijagnostičkog alata *ping*. Topologija *Ping/ping.imn* se sastoji od dva krajnja računala, čvorova *pc1* i *pc2*, jednog poslužitelja (engl. *server*), čvora *server*, dva LAN-komutatora (engl. *LAN switch*), *lanswitch8* i *lanswitch9*, te šest usmjeritelja (engl. *router*), koji su zadušeni za prijenos IP-datagrama između krajnjih čvorova. Eksperimenti se uglavnom temelje na provjeri dostupnosti između krajnjih računala i poslužitelja, a dodatno se može provjeriti i dostupnost pojedinih usmjeritelja. Važno je primjetiti da u topologiji na slici 3.1 postoje dva moguća puta između usmjeritelja *router1* i *router6*.



Slika 3.1: Topologija *Ping/ping.imn*

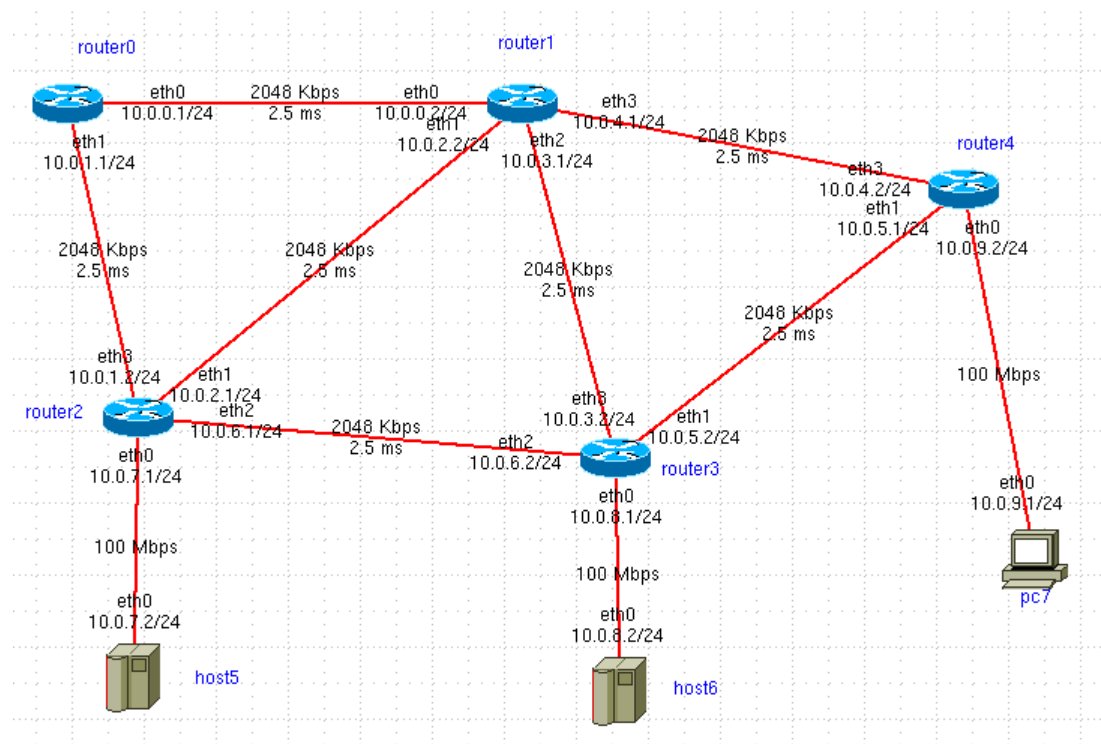
<sup>5</sup> Lovrek, Matijašević, Ježić, Jevtić: “Komunikacijske mreže”, Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva (2019) (trenutno dostupna radna inačica udžbenika)

Na slici 3.2 prikazana je topologija *Traceroute/traceroute.imn*, koja će se u ovom poglavlju koristiti za ilustraciju rada dijagnostičkog alata *traceroute*. Topologija *Traceroute/traceroute.imn* se sastoji od dva krajnja računala, čvorova *pc1* i *pc2*, jednog poslužitelja, čvora *server*, dva LAN-komutatora, *lanswitch8* i *lanswitch9*, te šest usmjeritelja. Eksperimenti se uglavnom temelje na provjeri najvjerojatnijeg puta između krajnjih računala i poslužitelja, a dodatno se mogu provjeriti i najvjerojatniji putevi do pojedinih usmjeritelja. Važno je primjetiti da u topologiji na slici 3.2 postoje dva moguća puta između usmjeritelja *router1* i *router6*.



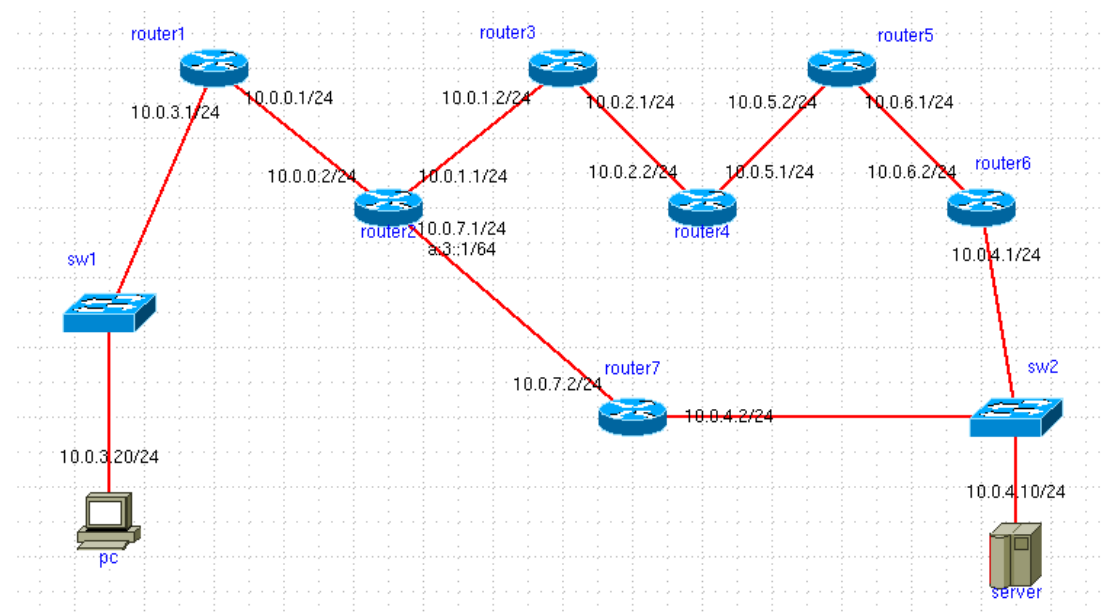
Slika 3.2: Topologija *Traceroute/traceroute.imn*

Na slici 3.3 prikazana je topologija *RIP/RIP.imn*, koja će se koristiti za ilustraciju rada protokola usmjeravanja RIP (*Routing Information Protocol*) u tzv. „mirnom stanju“, odnosno kad ne dolazi do promjena u topologiji mreže dodavanjem usmjeritelja u mrežu ili njihovim uklanjanjem iz mreže. Topologija *RIP/RIP.imn* se sastoji od jednog krajnjeg računala, čvora *pc7*, dva poslužitelja, čvorova *host5* i *host6*, te pet usmjeritelja. Eksperimenti se temelje na analizi poruka protokola RIP koje izmjenjuju usmjeritelji. Važno je primjetiti da u topologiji na slici 3.3 postoji više puteva između bilo koje kombinacije krajnjeg računala i poslužitelja, a zadatak samih usmjeritelja je da odrede puteve koji su najkraći s obzirom na korištenu metriku.



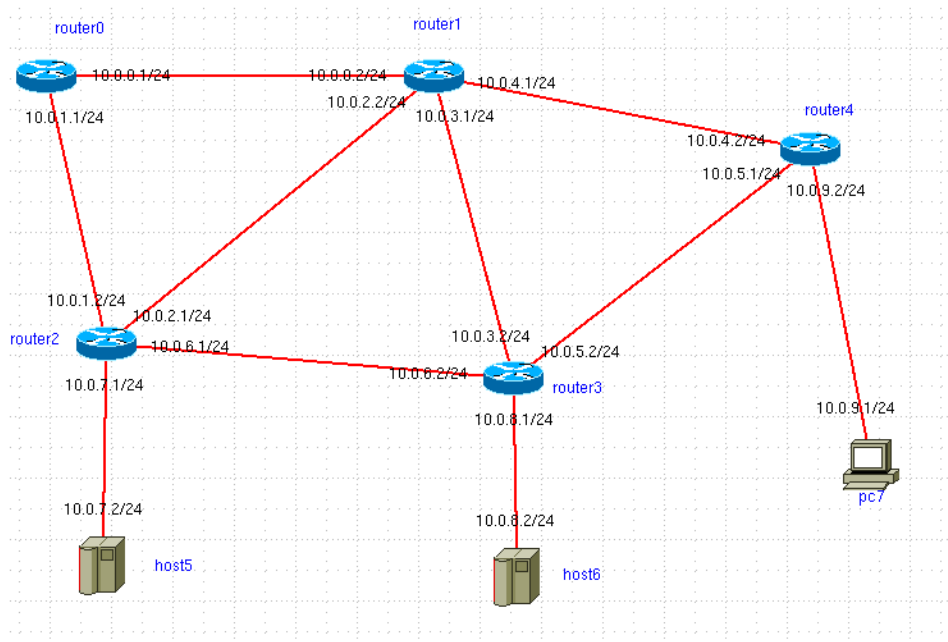
Slika 3.3: Topologija RIP/RIP.imn

Na slici 3.4 prikazana je topologija *RIP/RIP1.imn*, koja će se koristiti za ilustraciju rada protokola usmjeravanja RIP u slučajevima promjene topologije mreže uklanjanjem i dodavanjem čvora (usmjeritelja). Topologija *RIP/RIP1.imn* se sastoji od jednog krajnjeg računala, čvora *pc*, jednog poslužitelja, čvora *server*, te sedam usmjeritelja. Eksperimenti se temelje na analizi poruka protokola RIP koje izmjenjuju usmjeritelji te različitim informacija o mreži koje usmjeritelji pohranjuju u svojim tablicama. Važno je primjetiti da u topologiji na slici 3.4 postoje dva puta između krajnjeg računala i poslužitelja, a zadatak samih usmjeritelja je da odrede najkraći put s obzirom na trenutnu topologiju mreže.



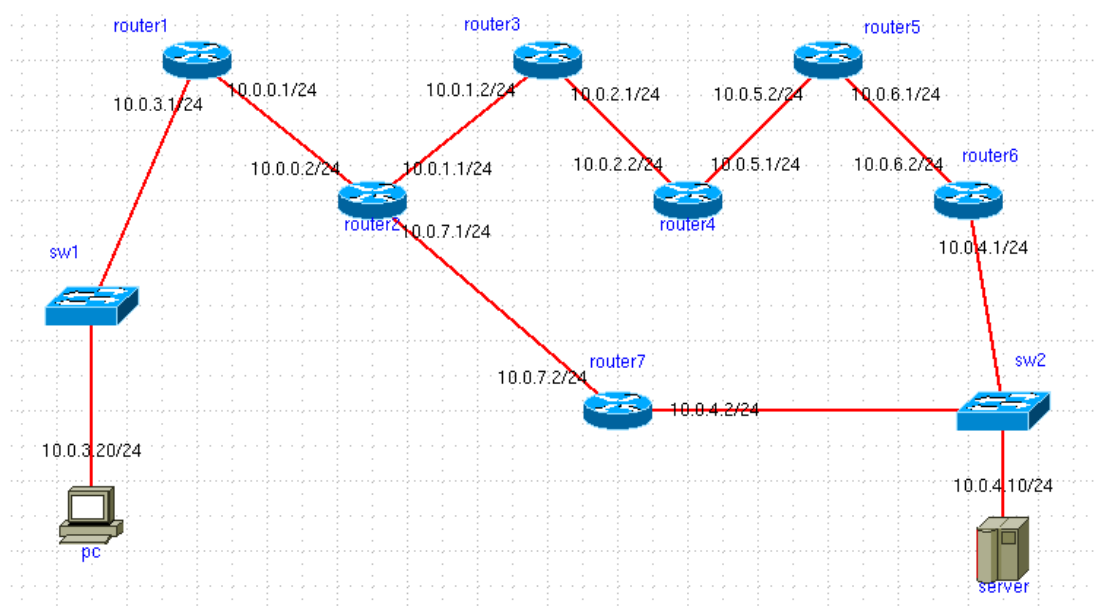
Slika 3.4: Topologija RIP/RIP1.imn

Na slici 3.5 prikazana je topologija *OSPF/OSPF.imn*, koja će se koristiti za ilustraciju rada protokola usmjeravanja OSPF (*Open Shortest Path First*) u tzv. „mirnom stanju“, odnosno kad ne dolazi do promjena u topologiji mreže dodavanjem usmjeritelja u mrežu ili njihovim uklanjanjem iz mreže. Topologija *OSPF/OSPF.imn* se sastoji od jednog krajnjeg računala, čvora *pc7*, dva poslužitelja, čvorova *host5* i *host6*, te pet usmjeritelja. Eksperimenti se temelje na analizi poruka protokola OSPF koje izmjenjuju usmjeritelji. Važno je primjetiti da u topologiji na slici 3.5 postoji više puteva između bilo koje kombinacije krajnjeg računala i poslužitelja, a zadatak samih usmjeritelja je da odrede puteve koji su najkraći s obzirom na korištenu metriku.



Slika 3.5: Topologija *OSPF/OSPF.imn*

Na slici 3.6 prikazana je topologija *OSPF/OSPF1.imn*, koja će se koristiti za ilustraciju rada protokola usmjeravanja OSPF u slučajevima promjene topologije mreže uklanjanjem i dodavanjem čvora (usmjeritelja). Topologija *OSPF/OSPF1.imn* se sastoji od jednog krajnjeg računala, čvora *pc*, jednog poslužitelja, čvora *server*, te sedam usmjeritelja. Eksperimenti se temelje na analizi poruka protokola OSPF koje izmjenjuju usmjeritelji te različitih informacija o stanju mreže koje usmjeritelji pohranjuju u svojim tablicama. Važno je primjetiti da u topologiji na slici 3.6 postoje dva puta između krajnjeg računala i poslužitelja, a zadatak samih usmjeritelja je da odrede najkraći put s obzirom na trenutnu topologiju mreže.



Slika 3.6: Topologija OSPF/OSPF1.imn

## 3.2 Objašnjenja korištenih pojmova, koncepata, protokola i alata

### 3.2.1 Protokol IP i pomoći protokoli

Protokol IP se zasniva na datagramskom načinu rada, a transportnom sloju pruža nespojnu, nepouzdanu mrežnu uslugu. Dvije osnovne funkcije mrežnog sloja koje protokol IP izvodi jesu adresiranje i fragmentacija. Pomoćni protokoli IP-u su protokoli ICMP i ARP. ICMP je, između ostalog, zadužen za dojavu različitih pogrešaka u mreži, a pomoću protokola ARP IP-adresa mrežnog sučelja se „preslikava“ na odgovarajuću MAC-adresu mrežne kartice. Dodatno, protokoli usmjeravanja, kao što su RIP i OSPF, zaduženi su za upravljanje prijenosom IP-datagrama kroz mrežu, jer određuju puteve za prijenos između različitih izvora i odredišta.

### 3.2.2 Alat ping

Kad se pojavi problem u radu neke mrežne aplikacije, prva stvar koju je potrebno provjeriti jest postoji li povezanost na mrežnom sloju. Jednostavno rečeno, potrebno je ustanoviti prolaze li IP-datagrami od jednog do drugog računala između kojih se pojavio problem u komunikaciji. Upravo u tu svrhu koristi se alat *ping*.

Alat *ping* omogućava ispitivanje povezanosti na mrežnom sloju između računala na kojem se pokrene alat i bilo kojeg od ostalih računala i čvorova u mreži. Kroz niz primjera, upoznat ćemo se s ovim alatom i s informacijama o radu mreže koje on pruža.

Alat *ping* se pokreće izvršavanjem sljedeće naredbe na izvorišnom računalu:

```
# ping <adresa ili ime odredišnog računala>
```

Ovaj alat šalje upit (u obliku kontrolne poruke) prema navedenom odredišnom računalu. Na ovaj upit odredišno računalo odgovara drugom kontrolnom porukom. Ukoliko alat *ping* primi

odgovor, ispiše ga, a korisnik ima informaciju da je određeno računalo dostupno. U slučaju da alat ne primi odgovor, postoji problem povezanosti na mrežnom sloju između dotičnih računala.

Na sljedećem primjeru ilustrira se rad alata *ping*. S jednog od računala, koje se nalazi u mreži Zavoda za telekomunikacije na FER-u, a čija adresa je 161.53.19.3, izvršena je naredba *ping* i kao argument joj je dano računalo imena `www.google.com`. Prikazana je naredba i njen ispis (Ispis 3.1) (*napomena*: izvođenje naredbe prekida se pritiskanjem kombinacije tipki `CTRL+C`).

```
# ping www.google.com

Pinging www.google.akadns.net [216.239.59.147] with 32 bytes of data:

Reply from 216.239.59.147: bytes=32 time=91ms TTL=238
Reply from 216.239.59.147: bytes=32 time=91ms TTL=238
Reply from 216.239.59.147: bytes=32 time=73ms TTL=240
Reply from 216.239.59.147: bytes=32 time=90ms TTL=238

Ping statistics for 216.239.59.147:

    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 73ms, Maximum = 91ms, Average = 86ms
```

*Ispis 3.1: Primjer korištenja naredbe ping (prema `www.google.com`)*

Ovaj ispis sadrži mnoštvo korisnih i zanimljivih podataka. Prva linija (`Pinging...`) kazuje da je određeno računalo, čija dostupnost se provjerava, punog naziva `www.google.akadns.net` i da je njegova IP-adresa 216.239.59.147. Također, naznačena je i veličina datagrama koji sadrži upit, a ona u ovom slučaju iznosi 32 okteta.

Svaka od sljedeće četiri linije ispisa (`Reply from...`) predstavlja po jedan odgovor određenog računala 216.239.59.147. Navedena je veličina odgovora (32 okteta), vrijeme koje je proteklo od slanja upita do primanja odgovora, te polje TTL (*Time To Live*) u zaglavlju IP-datagrama koji je primljen kao odgovor.

Može se primijetiti da prva dva odgovora imaju potpuno identične podatke. Treći odgovor je, međutim, zanimljiv. Vidljivo je da je kod njega polje TTL bilo veće nego kod ostalih odgovora te da je polje *time* manje. Prisjetimo se značenja polja TTL. U svaki IP-datagram koji se šalje upisuje se broj TTL koji može poprimiti vrijednost između 0 i 255. Podrazumijevana, tzv. *default*, vrijednost TTL-a podešava se u operacijskom sustavu. Na putu svakog IP-datagrama do odredišta, pri prolasku kroz mrežne čvorove (usmjeritelje), svaki čvor umanjuje vrijednost polja TTL za jedan. Ako u nekom čvoru, nakon umanjivanja polja TTL, njegova vrijednost postane „0“ (nula), čvor odbacuje datagram te prema izvoru datagrama (čija se IP-adresa nalazi u zaglavlju) šalje poruku o pogrešci (poruka *Time to live exceeded*). Posredno, to znači da polje TTL broji „skokove“ na putu datagrama kroz mrežu. Što je datagram prošao kroz više skokova, vrijednost u polju TTL je manja.

Zbog čega je, dakle, polje TTL u trećem odgovoru veće nego kod ostalih odgovora? Razlog je činjenica da je odgovor na treći upit kroz mrežu prošao putem s manjim brojem „skokova“ nego ostali odgovori. Ako je prošao putem s manjim brojem „skokova“, konačna vrijednost polja TTL bit će veća. Osim toga, s obzirom da je put bio kraći (a uz pretpostavku sličnog opterećenja svih mrežnih čvorova), i vrijeme prolaska datagrama je kraće te je stoga i polje *time* manje vrijednosti. Preostale linije ispisa daju pregled konačne statistike za sve poslane upite.

Računalo `www.google.com` nalazi se u SAD-u i do njega datagrami prolaze kroz desetke usmjeritelja. Analizirajmo sada primjer kad se naredba *ping* izvrši „prema“ nekom od računala u mreži FER-a, koje je bliže računalu na kojem se izvodi naredba *ping*.

S istog računala koje se nalazi u mreži Zavoda za telekomunikacije na FER-u, a čija adresa je 161.53.19.3, izvrši se naredba *ping* „prema“ računalu `mail.tel.fer.hr`.

```
# ping mail.tel.fer.hr

Pinging mail.tel.fer.hr [161.53.19.25] with 32 bytes of data:

Reply from 161.53.19.25: bytes=32 time=1ms TTL=255

Reply from 161.53.19.25: bytes=32 time=1ms TTL=255

Reply from 161.53.19.25: bytes=32 time=1ms TTL=255

Reply from 161.53.19.25: bytes=32 time=1ms TTL=255


Ping statistics for 161.53.19.25:

    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

    Approximate round trip times in milli-seconds:

        Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

Ispis 3.2: Primjer korištenja naredbe *ping* (prema `mail.tel.fer.hr`)

Analizirajući prethodni ispis može se primijetiti da je polje TTL u odgovoru na upit jednako 255. To znači da je računalo u istoj lokalnoj mreži kao i računalo s kojeg je poslan upit. Također, može se primijetiti da su IP-adrese oba računala vrlo slične. Štoviše, duljina mrežnog prefiksa podmreže Zavoda za telekomunikacije na FER-u iznosi 24 bita, što znači da te dvije adrese, s obzirom da su u istoj podmreži, moraju imati jednaka prva 24 bita.

Postoje manje razlike između izvedbi alata *ping* na različitim operacijskim sustavima. Pretpostavljene vrijednosti polja TTL i veličine kontrolnih poruka mogu biti različite na različitim operacijskim sustavima, a ispis alata može sadržavati više ili manje detalja. Tako, npr., operacijski sustav *Microsoft Windows* šalje pakete veličine 32 okteta i za vremena odziva manja od 1 ms samo ispisuje „< 1 ms“, dok distribucija operacijskog sustava *Linux Ubuntu 10.04* šalje pakete veličine 64 okteta i daje detaljniji ispis vremena, npr. „0.561 ms“.

Na datagrame koje šalje alat *ping* moguće je utjecati uporabom niza opcija koje naredba prihvaća. Neke od opcija su prikazane u sljedećoj tablici (Tablica 3.1), a detaljniji opis naredbe *ping* može se dobiti izvršavanjem naredbe *man ping*.



Tablica 3.1: Opcije za podešavanje alata ping

Opcija	Značenje opcije
<b>-c</b>	broj <i>ping</i> paketa koji se šalje
<b>-i</b>	interval između slanja <i>ping</i> paketa, u sekundama
<b>-n</b>	prikaz svih adresa računala u brojčanom, a ne simboličkom obliku
<b>-s</b>	veličina paketa koji se šalju izvršavanjem naredbe <i>ping</i>
<b>-m</b>	eksplicitno postavljanje TTL-vrijednosti poslanih paketa na navedeni iznos

Praktični primjer rada alata *ping* bit će demonstriran na mreži definiranoj u datoteci `ping.imn` (Slika 3.1) koja se nalazi u direktoriju `/root/imunes-examples/Ping`.

Nakon otvaranja datoteke u emulatoru/simulatoru IMUNES, pokrene se eksperiment odabirom stavke *Execute* u izborniku *Experiment* te snimanje prometa na računalu *pc1* alatom *Wireshark*, tako što se desnom tipkom miša klikne na računalu *pc1* te u padajućem izborniku odabere *Wireshark* → *eth0*. Otvori se konzola na računalu *pc1* tako što se na njega klikne desnom tipkom miša te u padajućem izborniku odabere stavka *Shell window* → *bash*. Ako se želi, npr., provjeriti dostupnost računala *server*, u konzoli je potrebno izvršiti sljedeću naredbu:

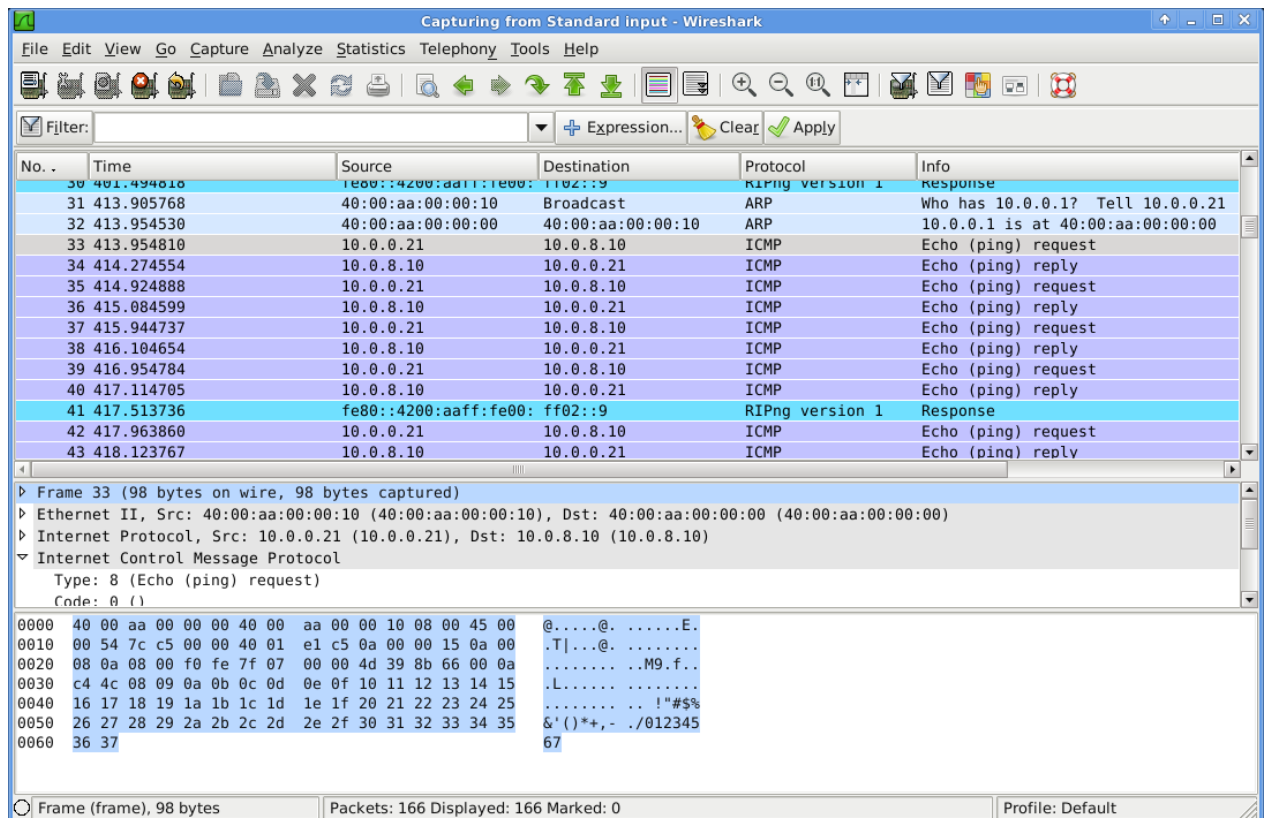
```
# ping 10.0.8.10
```

Nakon što alat ispiše prvih nekoliko redaka, prekine se njegovo izvršavanje kombinacijom tipki `Ctrl+C`. Rezultat je prikazan na sljedećoj slici (Slika 3.7).

```
IMUNES: pc1 (console)
[root@pc1 /]# ping 10.0.8.10
PING 10.0.8.10 (10.0.8.10): 56 data bytes
64 bytes from 10.0.8.10: icmp_seq=0 ttl=59 time=370.579 ms
64 bytes from 10.0.8.10: icmp_seq=1 ttl=59 time=159.978 ms
64 bytes from 10.0.8.10: icmp_seq=2 ttl=59 time=159.974 ms
64 bytes from 10.0.8.10: icmp_seq=3 ttl=59 time=159.954 ms
64 bytes from 10.0.8.10: icmp_seq=4 ttl=59 time=159.971 ms
64 bytes from 10.0.8.10: icmp_seq=5 ttl=59 time=159.973 ms
^C
--- 10.0.8.10 ping statistics ---
6 packets transmitted, 6 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 159.954/195.071/370.579/78.489 ms
[root@pc1 /]#
```

Slika 3.7: Konzola računala *pc1* pri korištenju alata ping

Nakon toga, u alatu *Wireshark* se zaustavi snimanje prometa. Promotrimo sada snimljeni promet u alatu *Wireshark* (Slika 3.8).



Slika 3.8: Snimljeni promet u alatu Wireshark nakon izvođenja naredbe ping

Vidimo da prve dvije snimljene poruke (radi se o porukama s rednim brojevima 31 i 32) pripadaju protokolu ARP. Računalo *pc1* zna da mu duljina mrežnog prefiksa iznosi 24 bita te zaključuje da se računalo *server* koje ima IP-adresu 10.0.8.10 ne nalazi u njegovoj podmreži, jer se IP-adrese računala *pc1* i *server* ne poklapaju u prva 24 bita. Ako pogledamo statičke zapise u tablici usmjeravanja računala *pc1* (klikne se desnom tipkom miša na računalo *pc1* te odabere *Configure*), možemo uočiti redak „0.0.0.0/0 10.0.0.1“ što znači da računalo *pc1* sve IP-datagrame (osim onih kojima je određište podmreža kojoj i sam pripada) treba proslijediti na adresu 10.0.0.1, što je adresa jednog od sučelja usmjeritelja *router0*. Usmjeritelj *router0* je, dakle, pretpostavljeni (engl. *default*) usmjeritelj za računalo *pc1*. Zato će računalo *pc1* IP-datagrame namijenjene računalu *server* proslijediti usmjeritelju *router0*. Budući da računalo *pc1* zna samo IP-adresu usmjeritelja *router0*, treba saznati njegovu MAC-adresu. Ona se saznaje slanjem upita protokolom ARP u kojem se pita tko ima IP-adresu 10.0.0.1. Taj upit računalo *pc1* šalje svim računalima u svojoj lokalnoj mreži. Kao izvorišnu adresu ARP-upita računalo *pc1* postavlja svoju MAC-adresu, a kao odredišnu adresu postavlja adresu ff:ff:ff:ff:ff:ff, koja u lokalnoj mreži predstavlja adresu svih sučelja na sloju podatkovne poveznice (engl. *MAC broadcast address*). To znači da se ARP-upit prosljeđuje svim mrežnim sučeljima u lokalnoj mreži, a u ovom slučaju konkretno računalu *pc2* i usmjeritelju *router0*. Računalo *pc2* po primitku upita ustanovljava da se ne traži njegova adresa i ne šalje nikakav odgovor. Usmjeritelj *router0* po primitku upita prepoznaje da se radi o njegovoj IP-adresi te šalje odgovor protokolom ARP u kojem dojavljuje svoju MAC-adresu računalu *pc1*. Odgovor se šalje samo računalu *pc1*, ostale uređaje u lokalnoj mreži taj odgovor ne zanima, a MAC-adresu računala *pc1* usmjeritelj *router0* zna iz upita koji je prethodno primio.

Nakon što je računalo *pc1* saznalo MAC-adresu sučelja *router0* na koje je spojena promatrana podmreža, može početi slati poruke računalu *server*. Alat *ping* koristi protokol ICMP, odnosno

njegove poruke *Echo request* i *Echo reply*. Računalu čija se dostupnost provjerava šalju se poruke *Echo request* na koje ono, ako je dostupno, odgovara porukama *Echo reply*. Vidimo da se poruke protokola ICMP prenose preko protokola IP (koji se pak u ovom slučaju prenosi preko protokola *Ethernet*). Kad računalu *pc1* pošalje poruku *Echo request* usmjeritelju *router0*, usmjeritelj analizira odredišnu adresu poruke u paketu protokola IP kojim se prenosi dotična poruka *Echo request* te vidi da se radi o odredišnoj adresi 10.0.8.10. Provjerom u svojoj tablici usmjeravanja usmjeritelj zaključuje da poruku treba proslijediti usmjeritelju *router1*. U paketu protokola IP usmjeritelj *router0* smanji vrijednost polja TTL za 1 (pritom ne mijenja izvorišnu niti odredišnu IP-adresu) te stavlja taj paket u novi okvir protokola *Ethernet*. No, za to mu treba odredišna MAC-adresa, u ovom slučaju od sučelja *eth0* usmjeritelja *router1*. Nju usmjeritelj *router0* saznaje na sličan način kako je to učinilo i računalu *pc1* za MAC-adresu usmjeritelja *router0*. Kad poruka *Echo request* konačno stigne do računala *server*, ono na nju odgovara porukom *Echo reply*. Budući da se u ispisu alata *ping* vide odgovori s adrese 10.0.8.10, a u alatu *Wireshark* poruke *Echo request* i *Echo reply*, može se zaključiti da je računalu *server* dostupno (u posebnoj situaciji računalu zapravo može biti dostupno, ali ne odgovarati na upite *Echo request* jer je tako podešeno).

Pogledajmo opet ispis alata *ping* u konzoli (Slika 3.7). Vidimo da je vrijednost polja TTL primljenih paketa jednaka 59. Kako je pretpostavljena vrijednost polja TTL u operacijskom sustavu FreeBSD jednaka 64, to znači da poruke *Echo reply* na putu od računala *server* do računala *pc1* prolaze kroz pet usmjeritelja (*router7*, *router6*, zatim *router2* ili *router5*, pa *router1* i konačno *router0*).

### 3.2.3 Alat *traceroute*

Kako odrediti preko kojih je čvorova određen IP-paket prošao na svom putu do odredišta? U internetskoj mreži odgovor na ovo pitanje je, nažalost, nikako. Mreže temeljene na protokolu IP, od kojih se sastoji Internet, ne pružaju mogućnost ispitivanja puteva koji se koriste za prijenos paketa. Nakon što je paket poslan s izvorišnog računala, ono ne može saznati kojim je putem paket prošao kroz mrežu do odredišta. Iako sam protokol IP, doduše, omogućava zapisivanje skokova kroz koje je paket prošao na svom putu do odredišta, ova funkcionalnost povlači određene probleme te je u internetskoj mreži uglavnom administrativno onemogućena.

Međutim, vrlo domišljatim korištenjem nekih postojećih mehanizama i protokola, koji izvorno imaju potpuno drugu namjenu, Van Jacobson je predložio rješenje koje omogućava saznavanje čvorova kroz koje će paketi poslani prema nekom odredištu najvjerojatnije proći. Njegovo rješenje ugrađeno je u popularni alat *traceroute* (u operacijskom sustavu *Microsoft Windows* koristi se naredba *tracert*). Alat *traceroute* odredištu šalje pakete postupno povećavajući polje TTL. Prvi paket ima vrijednost polja TTL postavljenu na „1“ te ga prvi usmjeritelj na putu odbaci i pošalje izvorištu poruku o pogrešci protokolom ICMP, čime izvorište saznaje prvi „skok“ (usmjeritelj) na putu. Drugom paketu se vrijednost polja TTL postavi na „2“ te ga odbaci drugi usmjeritelj na putu i pošalje novu poruku o pogrešci, čime i on postaje poznat izvorištu. Na ovaj način se paketi šalju sve dok i krajnje odredište ne pošalje odgovor.

Naredba *traceroute* uzima jedan argument, a to je IP-adresa ili ime odredišnog računala. Slijedi primjer njenog korištenja. S jednog od računala, koje se nalazi u mreži Zavoda za telekomunikacije na FER-u, a čija adresa je 161.53.19.3, izvršena je naredba *traceroute* i kao argument joj je dano računalo s imenom *www.google.com*. Detaljnije upute za korištenje naredbe *traceroute* mogu se dobiti naredbom:

```
# man traceroute
```

Ispis izvođenja naredbe *traceroute* (Ispis 3.3) prikazuje niz „skokova“ kroz koje paketi prolaze na putu do odredišta. U prethodnom primjeru, od računala na kojem je pokrenut *traceroute* do računala `www.google.com` postoji 17 međučvorova (zadnji čvor u nizu, onaj oznake „18“, predstavlja odredište `www.google.com`). Prvi stupac predstavlja redni broj detektiranog čvora. Sljedeća tri stupca predstavljaju tri mjerenja, odnosno vrijeme između slanja paketa i primitka poruke o pogrešci. Zadnji stupac označava IP-adresu čvora, a u slučaju da je ova adresa registrirana u DNS-sustavu navedeno je i simboličko ime čvora.

```
# traceroute www.google.com

Tracing route to www.google.akadns.net [216.239.59.147] over a maximum of 30 hops:

 1  12 ms   12 ms   11 ms  161.53.19.1
 2   1 ms    1 ms    2 ms  161.53.16.9
 3   1 ms    2 ms    2 ms  193.198.229.9
 4   4 ms    4 ms    5 ms  193.198.228.5
 5   2 ms    3 ms    2 ms  carnet.hrl.hr.geant.net [62.40.103.217]
 6   9 ms   10 ms   10 ms  hr.hu1.hu.geant.net [62.40.96.146]
 7  17 ms   17 ms   18 ms  hu.at1.at.geant.net [62.40.96.177]
 8  34 ms   36 ms   35 ms  at.ch1.ch.geant.net [62.40.96.2]
 9  44 ms   44 ms   43 ms  so-6-0-0.ar2.cdg2.gblx.net [208.48.23.161]
10  51 ms   50 ms   51 ms  so6-0-0-2488m.ar2.lon3.gblx.net [67.17.66.2]
11  51 ms   51 ms   51 ms  level-3public-peering.ge-5-0-0.ar2.lon3.gblx.net [208.51.239.162]
12  51 ms   54 ms   52 ms  ae-0-17.gar1.london1.level3.net [212.187.131.169]
13  52 ms   51 ms   51 ms  so-6-0.metro2-londencyh00.london1.level3.net [212.113.3.26]
14  50 ms   50 ms   50 ms  195.50.116.70
15  58 ms   49 ms   50 ms  216.239.46.173
16  77 ms   80 ms   78 ms  216.239.49.254
17  80 ms   79 ms   80 ms  216.239.49.121
18  78 ms   80 ms   79 ms  216.239.59.147

Trace complete.
```

Ispis 3.3: Primjer korištenja naredbe *traceroute* (prema računalu `www.google.com`)

Primijetimo još jednu interesantnu činjenicu vezanu uz navedeni primjer. Zašto su vremena za prvi čvor veća od vremena za čvorove 2, 3, 4, 5 i 6? Ukratko, kad usmjeritelj dobije paket kojem je TTL-polje jednako 1, a odredište tog paketa nije sam usmjeritelj, on će paket odbaciti te pošiljatelju odgovoriti posebnom ICMP-porukom. Ova obrada zahtijeva određene procesorske resurse u usmjeritelju i traje duže nego samo prosljeđivanje paketa. Ovisno o kapacitetu procesora u

usmjeritelju, ovo vrijeme može biti duže ili kraće. Izgleda da je procesor u prvom usmjeritelju sporiji u odnosu na nekoliko ostalih koji se nalaze na putu do računala `www.google.com`.

### 3.2.4 Protokoli usmjeravanja

Tablica usmjeravanja u svakom IP-čvoru, bio on usmjeritelj ili „obično“ računalo, pohranjuje informaciju o dostupnim odredištima te o „smjeru“ kojim treba prosljeđivati IP-datagrame do tih odredišta (tzv. rute usmjeravanja). Prosljeđivanje IP-datagrama obavlja se isključivo na temelju informacija u tablici usmjeravanja. Tablica može sadržavati jednu ili više ruta, a sadrži najmanje 3 vrste parametara: oznaku odredišta (odredišne mreže), metriku („cijenu“ puta prema odredištu) i oznaku „sljedećeg skoka“ (koja najčešće odgovara IP-adresi sljedećeg usmjeritelja na putu do odredišta). Dodatno, tablica usmjeravanja sadrži i oznaku lokalnog mrežnog sučelja koje „vodi“ do sljedećeg „skoka“ na putu. Generički primjer tablice usmjeravanja s 3 rute dan je sljedećom tablicom:

Odredište	Mrežna maska	Sljedeći skok	Mrežno sučelje	Metrika
0.0.0.0	0.0.0.0	192.168.1.1	eth1	100
127.0.0.0	255.0.0.0	127.0.0.1	lo0	1
192.168.1.0	255.255.255.0	0.0.0.0	eth0	10

Parametri *Odredište* i *Mrežna maska* zajedno predstavljaju oznaku odredišne mreže. Prvi redak se odnosi na podrazumijevanu (engl. *default*) rutu, koja se koristi u procesu prosljeđivanja paketa ako niti jedna druga ruta iz tablice usmjeravanja nije primjenjiva za promatrani IP-datagram. Drugi redak je vezan uz rutu tzv. povratne petlje (engl. *loopback*), koja, u ovom primjeru, određuje kako usmjeriti datagram s odredišnom IP-adresom 127.x.y.z (jer se primjenjuje mrežna maska 255.0.0.0), odnosno datagram namijenjen lokalnom čvoru. Primjer sadržaja tablice usmjeravanja za operacijski sustav FreeBSD dan je sljedećom tablicom (izbačeni su neki parametri koji nisu važni za ovo razmatranje):

Odredište	Sljedeći skok	Zastavice	Mrežno sučelje
Default	10.0.0.1	UGS	eth0
127.0.0.1	127.0.0.1	UH	lo0
10.0.0.0/24	link#1	UC	eth0

Ono što je specifično za simulator/emulator IMUNES, zbog njegove zasnovanosti na operacijskom sustavu FreeBSD, jest da polje *Sljedeći skok*, osim IP-adrese, može sadržavati i internu oznaku poveznice (u primjeru, „link#1“). Potonja najčešće označava rutu koja se odnosi na skup računala/mrežnih sučelja iz lokalne mreže, dostupna bez posredovanja usmjeritelja. (U starijim inačicama sustava FreeBSD mogu se pronaći i „rute“ za koje je pod parametrom *Sljedeći skok* navedena MAC-adresa, a koje povezuju IP-adresu odredišta s pripadajućom MAC-adresom, što je rezultat rada protokola ARP. Zbog jednoznačnosti namjene tablice usmjeravanja, takav zapis je napušten).

Informacije se u tablicu usmjeravanja čvora mogu upisivati od strane administratora sustava ili pak od strane protokola za usmjeravanje. Protokoli za usmjeravanje u IP-čvorovima rade na način da s ostalim čvorovima u mreži razmjenjuju informacije o odredištima te na temelju tih informacija donose zaključke o usmjeravanju IP-datagrama. Kada protokol za usmjeravanje donese odluku o najkraćem putu kojim treba prosljeđivati IP-datagrame prema nekom odredištu, on u tablicu usmjeravanja upisuje odredište i sljedeći „skok“ na tom putu.

Bitno je napomenuti da se sama odluka o tome na koje sučelje treba proslijediti IP-datagram donosi isključivo na temelju onoga što piše u tablici usmjeravanja. Nadalje, protokoli za usmjeravanje

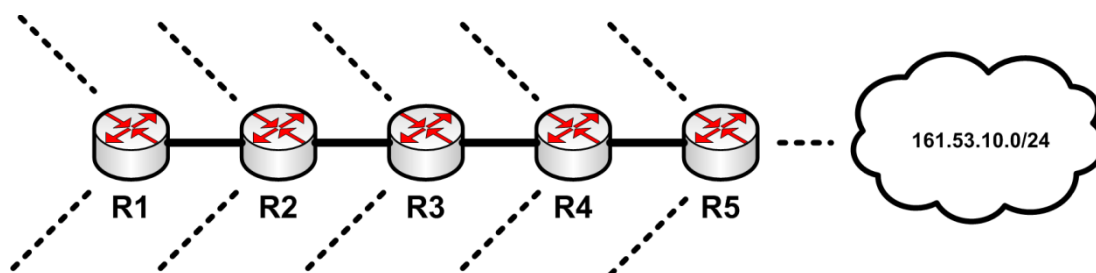
na samo usmjeravanje utječu isključivo na način da mijenjaju tablicu usmjeravanja. Protokoli za usmjeravanje koriste protokol IP za komunikaciju među čvorovima koji obavljaju usmjeravanje.

Protokoli za usmjeravanje u Internetu dijele se u sljedeće kategorije: „unutarnji“ i „vanjski“. Unutarnji protokoli koriste se unutar autonomnih sustava. Autonomni sustav (AS) je IP-mreža koja se najčešće nalazi pod administrativnom nadležnošću jednog tijela, a protokol usmjeravanja koji se koristi unutar AS-a nije „vidljiv“ izvan tog AS-a. Unutar autonomnih sustava nadležno tijelo može koristiti bilo koji protokol za unutarnje usmjeravanje, u čemu se, između ostalog, i odražava njegova autonomija. Autonomni sustav zadužen je za ostvarivanje povezanosti među čvorovima unutar sebe. Primjeri protokola za unutarnje usmjeravanje su *Routing Information Protocol* (RIP), protokol *Open Shortest Path First* (OSPF) i protokol *Intermediate System-Intermediate System* (IS-IS).

Za razmjenu informacija o usmjeravanju između autonomnih sustava koriste se vanjski protokoli usmjeravanja. S obzirom da svi autonomni sustavi moraju moći razmjenjivati informacije s ostalim sustavima, izbor protokola za vanjsko usmjeravanje u Internetu nije proizvoljan, već se mora koristiti protokol *Border Gateway Protocol verzije 4* (BGP4).

### 3.2.4.1 Protokol RIP

Jedan od najstarijih i najjednostavnijih protokola za unutarnje usmjeravanje jest protokol RIP, koji koristi transportni protokol UDP (*User Datagram Protocol*) za prijenos poruka. Protokol RIP spada u kategoriju tzv. protokola temeljenih na vektorima udaljenosti. Vektor udaljenosti je lista odredišta i pripadajućih udaljenosti do tih odredišta. Razmjenom ovakvih vektora udaljenosti čvorovi mogu jednostavno pronaći najkraće putove do svih ostalih odredišta. Način na koji se interpretiraju vektori udaljenosti u čvorovima objašnjen je na sljedećoj slici (Slika 3.9):



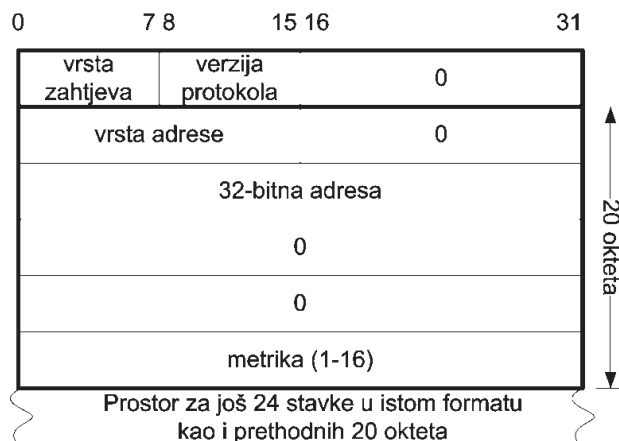
Slika 3.9: Primjer dijela mrežne topologije s rutom do podmreže 161.53.10.0/24

Neka, na primjer, čvor R1 primi vektor udaljenosti od susjednog čvora R2, kojim R2 kaže da se do podmreže 161.53.10.0/24 može doći preko njega u 4 skoka. Ukoliko čvor R1 prije primitka tog vektora nije znao da uopće postoji podmreža 161.53.10.0/24 (tj., takav zapis nije postojao u njegovoj tablici usmjeravanja), on u svoju tablicu usmjeravanja jednostavno dodaje redak koji izgleda ovako:

Odredište	Sljedeći skok	Udaljenost
161.53.10.0/24	R2	5

Međutim, ako je čvor R1 u svojoj tablici usmjeravanja već imao odredište 161.53.10.0/24 te ako mu je udaljenost do tog odredišta bila, na primjer, 3, onda se postojeći redak u tablici ne mijenja. To je zato što čvor R1 već zna „put“ do navedenog odredišta, i to u manje skokova nego preko čvora R2. Dakle, redak u tablici se ažurira samo ako je novi put do odredišta kraći od onoga koji se već koristi (tj., koji je već zapisan).

Slika 3.10 prikazuje format RIP-paketa. Polje vrsta zahtjeva u zaglavlju RIP-paketa označava predstavlja li paket zahtjev za informacijom (vrijednost 1) ili odgovor na zahtjev (vrijednost 2). Polje verzija protokola označava verziju protokola RIP koja se koristi. Polje vrsta adrese najčešće ima vrijednost „2“, što označava da se radi o 32-bitnim IP-adresama. Nakon vrste adrese navodi se sama IP-adresa i pripadajuća metrika, čiji je smisao broj „skokova“ do odredišta koje dana adresa predstavlja.



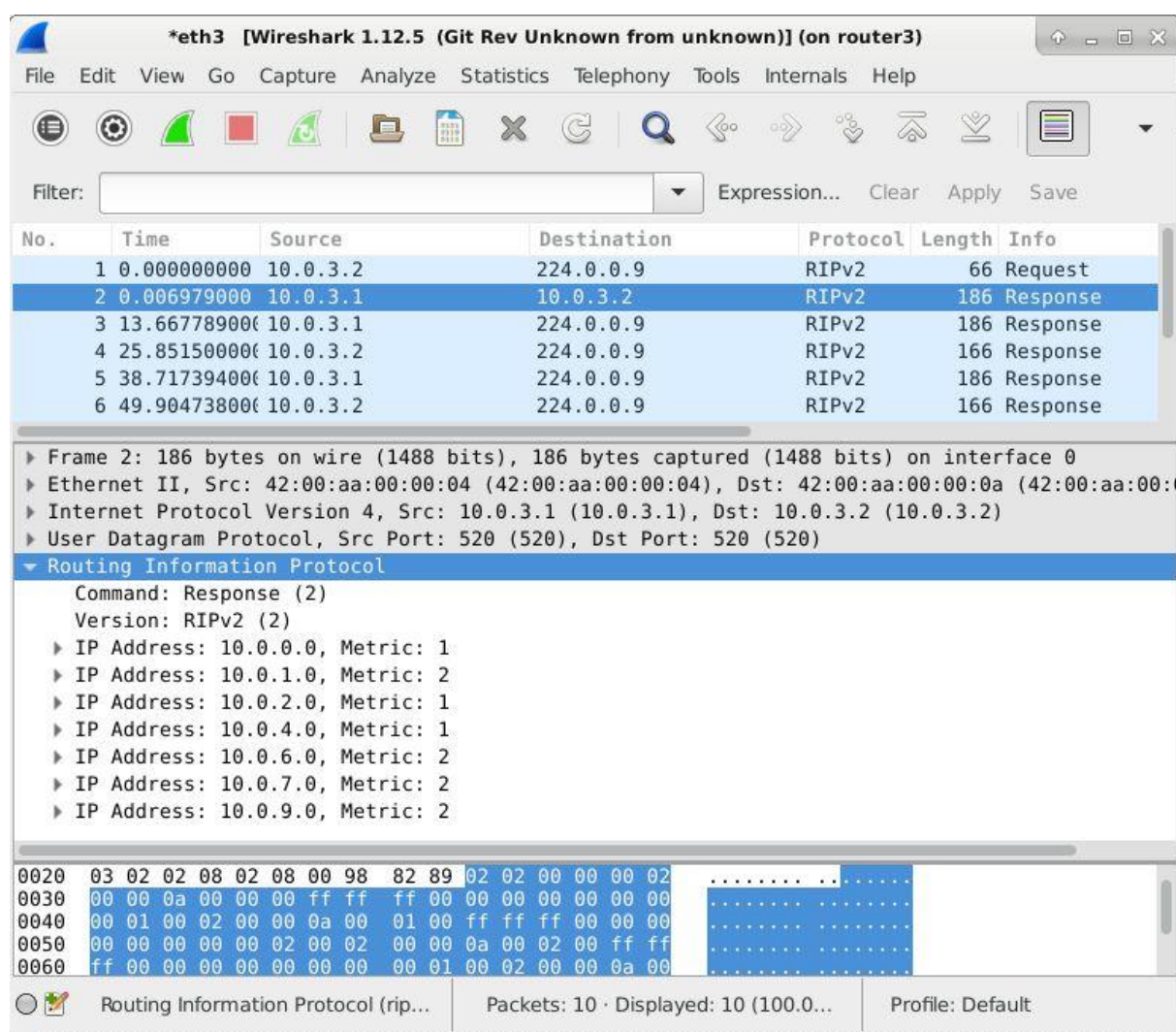
Slika 3.10: Format RIP-paketa

Otvorimo primjer `RIP.imn` (Slika 3.3) iz direktorija `/root/imagenes-examples/RIP`, pokrenimo eksperiment te započnimo snimanje prometa na sučelju `eth3` usmjeritelja `router3`. Snimanje prometa zaustavimo nakon otprilike jedne minute, odnosno nakon što se u alatu *Wireshark* pojave barem dva UDP-datagrama (od čega barem jedan zahtjev, *Request*, i barem jedan odgovor, *Response*, što se vidi u koloni *Info* u popisu snimljenih paketa alata *Wireshark*).

Odaberimo prvo jedan zahtjev protokola RIP i proučimo njegov sadržaj. Obratimo pažnju na adrese u pripadnom paketu protokola IP. Vidimo da je kao izvorišna adresa postavljena adresa 10.0.3.2, što odgovara IP-adresi mrežnog sučelja `eth3` usmjeritelja `router3`, te da je kao odredišna adresa postavljena adresa 224.0.0.9, višedredišna adresa kojom se adresiraju svi usmjeritelji koji koriste protokol RIP verzije 2. Po primitku ove poruke usmjeritelji šalju svoje odgovore. Odaberimo sada jedan odgovor od susjednog usmjeritelja na ovoj poveznici, npr. od usmjeritelja `router1`. Vidimo da je kao izvorišna IP-adresa postavljena adresa sučelja `eth2` usmjeritelja `router1` (10.0.3.1) koje je izravno spojeno s usmjeriteljem `router3`, a kao odredišna adresa je postavljena adresa sučelja `eth3` usmjeritelja `router3` (10.0.3.2) s kojeg je zahtjev i poslan. Odgovor (Slika 3.11) se sastoji od zaglavlja u kojem vidimo vrstu naredbe (*Response*) i verziju protokola (2) te tijela poruke koje sadrži adrese podmreža za koje usmjeritelj „zna“ i pripadne metrike. Pogledajmo, npr., treći podatak u tijelu poruke koji glasi:

▷ IP Address: 10.0.2.0, Metric 1.





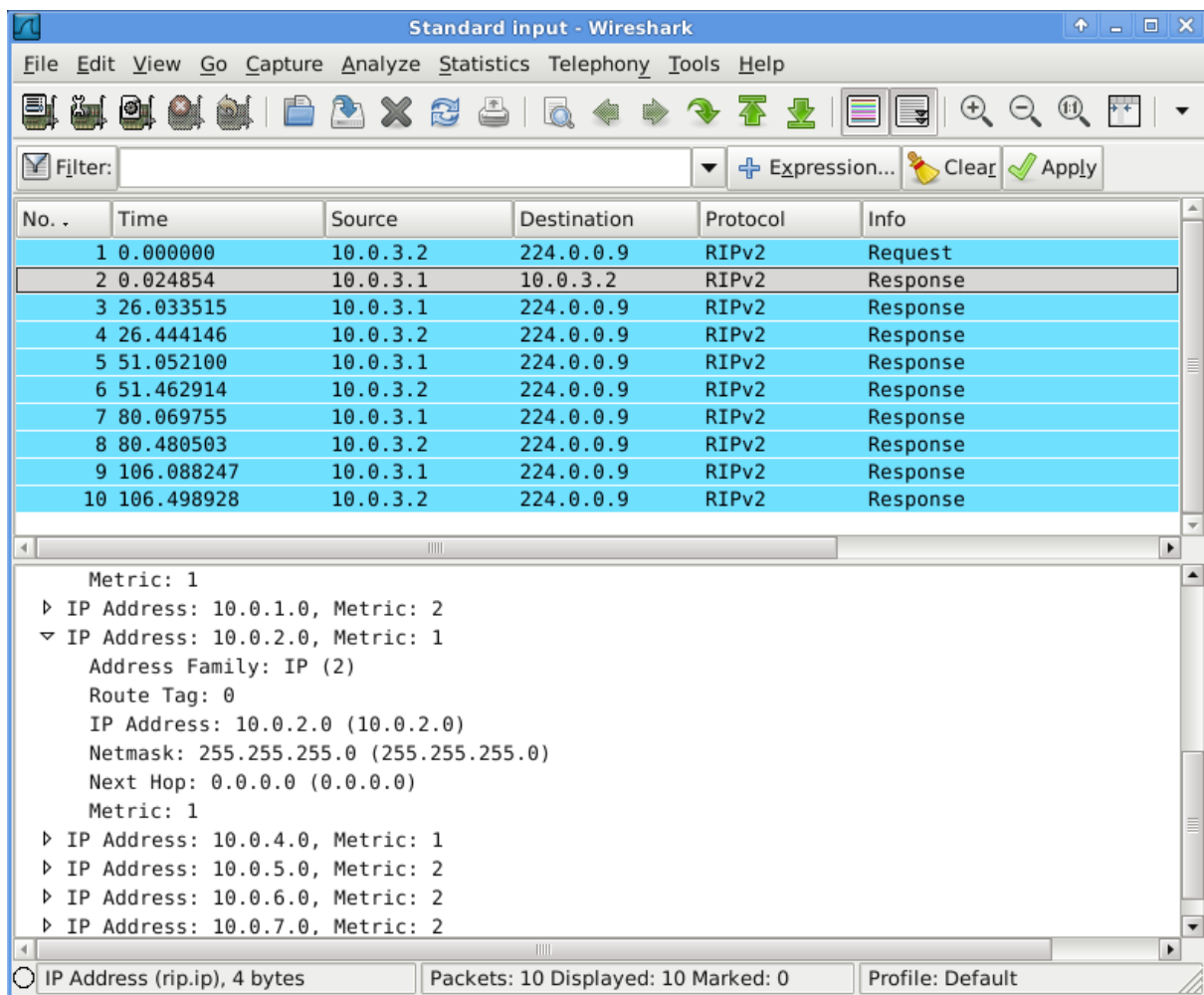
Slika 3.11: Poruke protokola RIP u alatu Wireshark: RIP-odgovor

Taj je dio poruke detaljnije prikazan na sljedećoj slici (Slika 3.12). Vidimo da se sastoji od sljedećih dijelova:

- Vrsta adrese (*address family*) – postavljeno na 2, što označava da se radi o 32-bitnim IP-adresama,
- Oznaka rute (*route tag*) – koristi se za razlikovanje ruta određenih protokolom RIP od ruta određenih drugim protokolima usmjeravanja,
- IP-adresa (*IP address*) – adresa pod mreže za koju se određuje ruta, u ovom slučaju radi se o mreži s IP-adresom 10.0.2.0,
- Maska pod mreže (*subnet mask*) – označava mrežnu masku te pod mreže, a budući da se radi o masci 255.255.255.0, zaključujemo da se radi o mreži 10.0.2.0/24,
- Sljedeći skok (*next hop*) – oznaka sljedećeg skoka, postavljeno na „0.0.0.0“ što znači da je sljedeći skok čvor koji je poslao poruku – u ovom slučaju usmjeritelj *router1*.

Dakle, na primjeru ovog dijela poruke zaključujemo da usmjeritelj *router1* poručuje usmjeritelju *router3* da udaljenost usmjeritelja *router1* od mreže 10.0.2.0/24 iznosi jedan, što znači da je usmjeritelj *router1* spojen izravno na mrežu 10.0.2.0/24. Ovime usmjeritelj *router3* ustanovljuje da može doći do mreže 10.0.2.0/24 udaljene dva „skoka“ (metrika 2) preko usmjeritelja *router1*.





Slika 3.12: Poruke protokola RIP u alatu Wireshark: prikaz metrika

Pogledajmo sada što se događa u mreži kad neki usmjeritelj prestane s radom, a koristi se protokol RIP. Otvorimo u IMUNES-u topologiju `RIP1.imn` (Slika 3.4). Pokrenimo snimanje prometa na sučelju `eth2` usmjeritelja `router2`, otvorimo konzolu usmjeritelja `router2` (desni klik na usmjeritelj `router2` → *Shell window* → `vtysh`) i izvršimo naredbu `show ip rip`, kojom ćemo vidjeti koje su rute zapisane u tablici usmjeravanja usmjeritelja `router2` (Slika 3.13).

Vidimo, primjerice, da se do mreže `10.0.2.0/24` može doći s metrikom 2 preko adrese `10.0.1.2` (adresa sučelja `eth0` usmjeritelja `router3`). Provjerimo sada dostupnost računala `server` (`10.0.4.10`) s računala `pc`. Primjećujemo da računalo `server` odgovara, što znači da je između njih ostvarena povezanost na mrežnom sloju.

Zaustavimo sada usmjeritelj `router7` (desni klik na usmjeritelj `router7` → *Stop*). Ako sada pokušamo provjeriti povezanost između računala `pc` i poslužitelja `server` pomoću naredbe `ping`, vidjet ćemo da nisu povezani. Možemo pratiti što se događa u usmjeritelju `router2` pomoću naredbi navedenih u sljedećoj tablici (Tablica 3.2).

```

IMUNES: router2 (console)
Hello, this is Quagga (version 0.99.15).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

router2# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

      Network          Next Hop          Metric From          Tag Time
R(n) 0.0.0.0/0         10.0.7.2          2 10.0.7.2           0 02:47
C(i) 10.0.0.0/24       0.0.0.0           1 self               0
C(i) 10.0.1.0/24       0.0.0.0           1 self               0
R(n) 10.0.2.0/24       10.0.1.2          2 10.0.1.2           0 02:56
R(n) 10.0.3.0/24       10.0.0.1          2 10.0.0.1           0 02:58
R(n) 10.0.4.0/24       10.0.7.2          2 10.0.7.2           0 02:47
R(n) 10.0.5.0/24       10.0.1.2          3 10.0.1.2           0 02:56
R(n) 10.0.6.0/24       10.0.7.2          3 10.0.7.2           0 02:47
C(i) 10.0.7.0/24       0.0.0.0           1 self               0
router2#
  
```

Slika 3.13: Rezultat izvršavanja naredbe `show ip rip` na usmjeritelju `router2`

Tablica 3.2: Naredbe za korištenje u konzoli usmjeritelja (protokol RIP)

Naredba	Značenje naredbe
<code>show ip rip</code>	prikazuje sve rute protokola RIP
<code>show ip rip status</code>	prikazuje trenutni status ruta protokola RIP

Odmah nakon zaustavljanja usmjeritelja `router7` naredbe u konzoli usmjeritelja `router2` daju ispis iz kojeg se ne može zaključiti da se nešto dogodilo. Naime, iz rezultata izvršavanja naredbe `show ip rip` iščitavamo sljedeći redak:

```
R(n) 10.0.4.0/24      10.0.7.2      2      10.0.7.2      0      0      02:50
```

Redak označava da se radi o ruti protokola RIP (oznaka `R(n)`) i da je pod mreža `10.0.4.0/24` dostupna s metrikom 2 preko sljedećeg „skoka“ `10.0.7.2` (usmjeritelj `router7`), a ta je informacija stigla s adrese `10.0.7.2` prije 10 sekundi (primitkom informacije, vremenski brojač se inicijalizira na 3 min, a u trenutku izvršavanja naredbe `show ip rip` njegova vrijednost se smanjila na 2:50 min). Iz rezultata izvršavanja naredbe `show ip rip status` iščitavamo redak:

```

Routing information sources:
10.0.7.2      0      0      120      00:00:22
  
```

Iz ovoga zaključujemo da je s adrese `10.0.7.2` stigla poruka prije 22 s (broj 120 označava administrativnu distancu protokola RIP). Nakon isteka tri minute usmjeritelj `router2` pretpostavlja da usmjeritelj `router7` nije dostupan te naredba `show ip rip` ispiše metriku 16 za pod mrežu `10.0.4.0/24`, a naredba `show ip rip status` uopće ne navodi adresu `10.0.7.2` u ispisu. Nakon

nekog vremena mreža se „oporavi“ – usmjeritelj *router2* u svojoj tablici usmjeravanja (`show ip rip`) sada ima redak:

```
R(n) 10.0.4.0/24      10.0.1.2    5      10.0.1.2    0      0      02:50
```

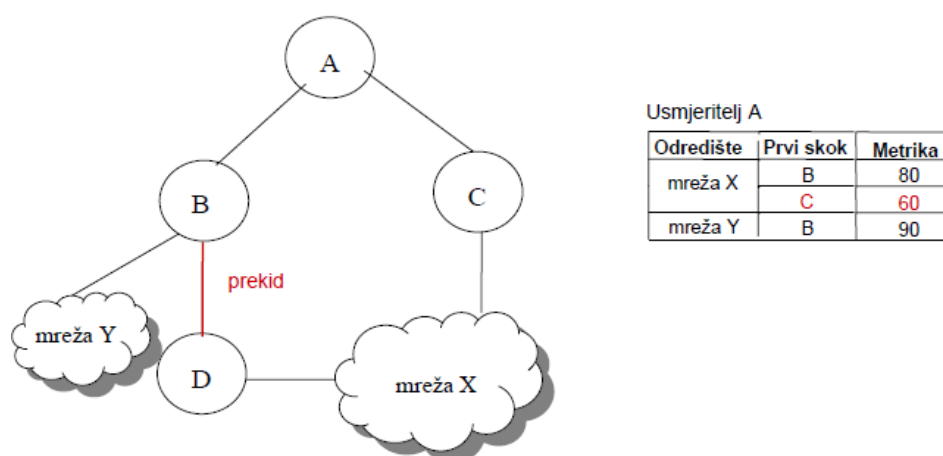
To znači da je mreža 10.0.4.0/24 ponovno dostupna, ovaj puta s metrikom 5 preko sljedećeg „skoka“ 10.0.1.2 (sučelje *eth0* usmjeritelja *router3*). Ako sada pokušamo provjeriti dostupnost poslužitelja *server* s računala *pc1* pomoću naredbe *ping*, vidjet ćemo da je poslužitelj dostupan.

Ako ponovno aktiviramo usmjeritelj *router7* (desni klik na usmjeritelj *router7* → *Start*) i odmah nakon toga u konzoli usmjeritelja *router2* provjerimo rute, vidimo da se odmah vratila „stara“ (kraća) ruta prema mreži 10.0.4.0/24 koja ide preko usmjeritelja *router7*.

### 3.2.4.2 Protokol OSPF

Sljedeći poznati protokol za unutarnje usmjeravanje je OSPF. Protokol se temelji na algoritmu stanja poveznice koji ne uzima u obzir samo topologiju mreže, nego i propusnost poveznica. Rad protokola se zasniva na podatkovnoj strukturi koja se naziva baza podataka stanja poveznice (engl. *Link State Database*, LSDB). Svaki usmjeritelj AS-a održava vlastitu kopiju te baze, koja sadrži informacije o trenutnom stanju mreže u vidu njezine topologije i propusnosti poveznica. Svaka poveznica prema odredištu ili drugom usmjeritelju određena je zasebnim unosom u toj bazi te karakterizirana pripadajućom vrijednosti metrike („cijene“).

Informacije o stanju mreže usmjeritelji razmjenjuju slanjem poruka, gdje svaki usmjeritelj ostalima javlja njemu poznate informacije o stanju mreže, kako bi svi usmjeritelji u konačnici imali jednak „pogled“ na cijelu mrežu, odnosno identičnu tablicu usmjeravanja. Prilikom komunikacije s drugim usmjeriteljima šalju se samo stanja pojedinih poveznica, a ne cijele tablice usmjeravanja, što rezultira manjim zauzećem mrežnih resursa. Ako se dogodi promjena u mreži, primjerice dodavanjem ili ispadom pojedinog čvora, susjedni usmjeritelji uočavaju nastalu promjenu i šalju poruke s novom informacijom kako bi svi ostali usmjeritelji bili obaviješteni. Kod OSPF-a se takve poruke razmjenjuju po potrebi, tj., samo kod promjena u topologiji mreže, a ne periodički kao kod RIP-a.



Slika 3.14: Primjer rada protokola OSPF

Kako bi se ustanovile rute usmjeravanja, svaki usmjeritelj koristi svoj LSDB s ciljem izgradnje stabla najkraćeg puta. To stablo opisuje poveznice prema svim ostalim usmjeriteljima te određuje puteve najmanje cijene prema bilo kojem odredištu. Ako usmjeritelj primi novu informaciju o stanju mreže, stablo najkraćeg puta se ponovno izračunava, čime se proces usmjeravanja dinamički prilagođava mrežnim uvjetima. Slika 3.14 prikazuje primjer rada protokola OSPF.

Za danu mrežnu topologiju prikazana je tablica usmjeravanja usmjeritelja A. Vidimo da od usmjeritelja A do mreže X promet može ići preko usmjeritelja B ili preko usmjeritelja C. Ako dođe, primjerice, do prekida veze između usmjeritelja B i D, sav promet od usmjeritelja A do mreže X se može preusmjeriti na usmjeritelj C. Slika 3.15 prikazuje format zaglavlja protokola OSPF.

0	8	16	24	31
version	packet type	packet length		
router ID				
area ID				
checksum		autype		
authentication (64 bits)				

Slika 3.15: Format zaglavlja protokola OSPF

Protokol OSPF definira više vrsta poruka koje se razlikuju po vrijednosti polja *Packet type*:

- 1 – *Hello*: omogućuju svakom usmjeritelju da otkrije druge susjedne usmjeritelje na svojim lokalnim poveznicama, čime susjedni uređaji mogu razmijeniti informacije važne za rad protokola,
- 2 – *Database Description* (DD): služi za razmjenu sadržaja LSDB-a između usmjeritelja,
- 3 – *Link State Request* (LSR): omogućuje usmjeritelju da zatraži osvježavanje informacije o stanju pojedinih poveznica od drugog usmjeritelja,
- 4 – *Link State Update* (LSU): šalje se periodički, ili kao odgovor na poruku LSR, a nosi osvježenu informaciju o stanju pojedinih poveznica,
- 5 – *Link State Acknowledgement* (LSA): služi kao potvrda primitka poruke LSU, a s ciljem povećanja pouzdanosti rada protokola.

### 3.3 Eksperimenti i zadaci

**Zadatak 4.** U emulatoru/simulatoru IMUNES, ispitajte način rada alata *traceroute* na mreži iz primjera *Traceroute/traceroute.imn* (Slika 3.2).

1. Započnite simulaciju.
2. Pokrenite alat *Wireshark* na sučelju *eth0* računala *pc1* i započnite snimanje mrežnog prometa.
3. Otvorite konzolu na računalu *pc1*.
4. Provjerite najvjerojatniji put (naredba *traceroute*) od računala *pc1* do poslužitelja *server* (10.0.8.10). Analizirajte odgovor koji je dobilo računalo *pc1* u sklopu izvršavanja naredbe *traceroute*. Provjerite IP-adrese sučelja na usmjeriteljima uključenim u usmjeravanje paketa generiranih alatom *traceroute*.

5. Otvorite konzolu na poslužitelju *server*.
6. Provjerite najvjerojatniji put (naredba *tracert*) od poslužitelja *server* do računala *pc1* (10.0.0.21). Analizirajte odgovor koji je dobio poslužitelj *server* u sklopu izvršavanja naredbe *tracert*. Provjerite IP-adrese sučelja na usmjeriteljima uključenim u usmjeravanje paketa generiranih alatom *tracert*.
7. Usporedite rezultate dobivene izvršavanjem naredbe *tracert* u koracima 4 i 6 te ih komentirajte.

**Zadatak 5.** (Topologija *Tracert/tracert.imn*) Kojim protokolom se IP-paketi prenose između računala smještenih unutar jedne lokalne mreže? Čemu, pri tome, služi protokol ARP?

**Zadatak 6.** (Topologija *Ping/ping.imn*) Ponovite pokazni eksperiment s početka poglavlja (*Ping/ping.imn*) te snimite promet koji pripada protokolu ARP. Skicirajte i objasnite način rada protokola ARP, a posebnu pozornost obratite na IP-adresu koja se navodi u ARP-zahtjevu. Kojem čvoru odgovara ta IP-adresa? Objasnite. Čemu služe višedrežne adrese u protokolu Ethernet? Koristi li ih protokol ARP?

**Zadatak 7.** (Topologija *Ping/ping.imn*) Proučite utjecaj raznih parametara, koje je moguće prosljediti naredbi *ping*, na sadržaj paketa koji se šalju. Parametri se mogu dobiti izvođenjem naredbe *ping* bez argumenata ili na stranici s uputama koja se dobiva izvršavanjem naredbe *man ping*. Komentirajte parametre ukratko.

**Zadatak 8.** (Topologija *Ping/ping.imn*) Utvrdite i objasnite što se događa pri slanju paketa alatom *ping* koji u polju TTL imaju vrijednost 3, a određeno računalo je neko računalo udaljeno više od 3 „skoka“.

**Zadatak 9.** (Topologija *Ping/ping.imn*) Utvrdite i objasnite što se događa kad je *ping* paket koji se šalje velik 10000 okteta. Kolika je maksimalna moguća veličina paketa koji se može postaviti prilikom izvršavanja naredbe *ping*? O čemu ona ovisi?

**Zadatak 10.** (Topologija *Tracert/tracert.imn*) Utvrdite i objasnite kako veličina paketa koji se šalje utječe na vrijeme koje prijavljuje alat *ping* (tzv. *ping time*). Ispitajte kako se mijenjaju vrijednosti koje vraća alat *ping*, ako se u mreži izravno spoje dva usmjeritelja koja prije nisu bila izravno povezana (npr. računalo *pc1* provjerava dostupnost poslužitelja *server* bez i uz postojanje izravne veze između usmjeritelja *router0* i *router7*)?

**Zadatak 11.** (Topologija *Tracert/tracert.imn*) Utvrdite i objasnite kako propagacijsko kašnjenje utječe na vrijeme koje prijavljuje alat *ping* (tzv. *ping time*). Ispitajte kako se mijenjaju vrijednosti vremena koje vraća alat *ping*, ako se u mreži promijeni propagacijsko kašnjenje između računala i ethernetskog komutatora (npr. računalo *pc1* provjerava dostupnost poslužitelja *server* uz različito podešeno propagacijsko kašnjenje između računala *pc1* i ethernetskog komutatora *lanswitch8*)?

**Zadatak 12.** (Topologija *Ping/ping.imn*) U emulatoru/simulatoru IMUNES proučite i detaljno analizirajte uhvaćeni slijed paketa koji je generirao alat *ping* između različitih računala u mreži. Utvrdite koji su sve protokoli iskorišteni kao posljedica izvođenja naredbe *ping* i koji je odnos među njima (tj., koje druge protokole svaki pojedini protokol koristi). Navedite kojem sloju TCP/IP-modela svaki od tih protokola pripada.

**Zadatak 13.** (Topologija Ping/ping.imn) Utvrdite što se sve mijenja u okviru protokola Ethernet kad se koristi naredba *ping* s različitim veličinama paketa koji se šalju.

**Zadatak 14.** (Topologija Ping/ping.imn) Utvrdite kakav se promet generira na ethernetskom sučelju računala kad se provjerava dostupnost (naredba *ping*) adrese 127.0.0.1. Komentirajte rezultat.

**Zadatak 15.** (Topologija Ping/ping.imn) Utvrdite kolike su minimalna i maksimalna vrijednost MTU-a (*Maximum Transfer Unit*) na ethernetskom sučelju. Pokušajte podesiti MTU i veličinu *ping* paketa tako da ostvarite što veći broj fragmenata. Način podešavanja MTU-a pronađite u uputama naredbe *ifconfig(8)*, dakle, izvršenjem naredbe `man ifconfig`.

**Zadatak 16.** (Topologija Traceroute/traceroute.imn) Utvrdite neke od mogućih situacija u kojima alat *traceroute* može proizvesti rezultat koji nije ispravan (*naputak*: pogledajte što piše u uputama alata – izvršite naredbu `man traceroute`).

**Zadatak 17.** (Topologija Traceroute/traceroute.imn) U emulatoru/simulatoru IMUNES, ispitajte način rada alata *traceroute* na mreži iz primjera *Traceroute/traceroute.imn*. Potrebno je snimati mrežni promet na pojedinim sučeljima i utvrditi mehanizam na kojem se temelji rad alata. Kako se koristi TTL-polje i protokol ICMP?

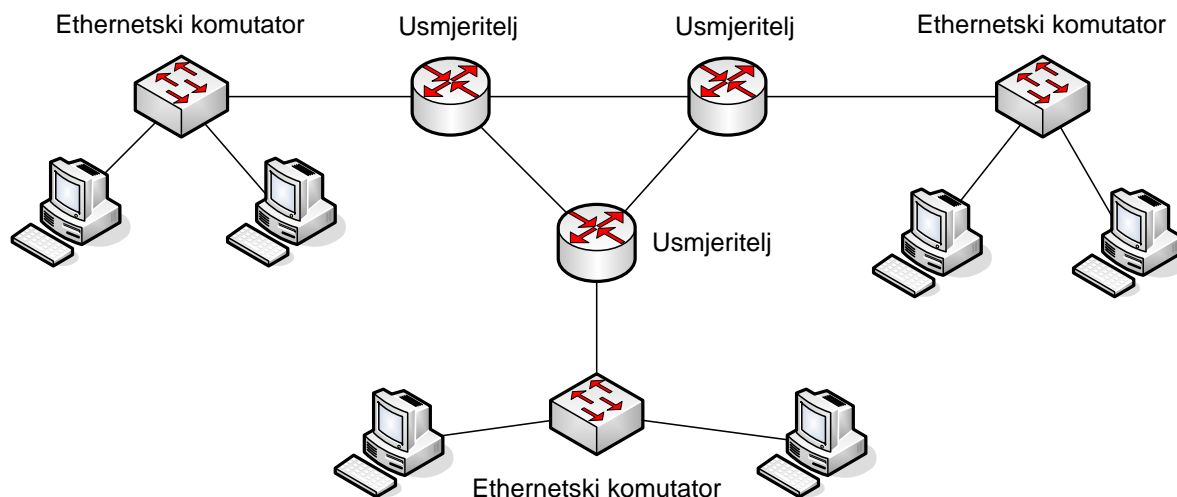
**Zadatak 18.** (Topologija Traceroute/traceroute.imn) Na koji način protokol IP „pamti“ vrstu paketa koji se prenosi u podatkovnom dijelu IP-datagrama? Navedite primjere različitih paketa iz podatkovnog dijela IP-datagrama.

**Zadatak 19.** (Topologija Traceroute/traceroute.imn) Utvrdite postoji li način da se iz primljenog IP-paketa očita put kojim je paket prošao kroz mrežu.

**Zadatak 20.** (Topologija Traceroute/traceroute.imn) Utvrdite postoji li način kojim protokol IP može ustanoviti da je poslani paket stvarno i primljen na odredištu.

**Zadatak 21.** (Topologija Ping/ping.imn) Utvrdite utječe li fragmentacija na propusnost i kašnjenje te komentirajte dobivene rezultate.

**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.



Slika 3.16: Arhitektura mreže koju je potrebno konstruirati

**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.

**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?

**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.

**Zadatak 26.** (Topologija OSPF/OSPF.imn) Proučite primjer *OSPF/OSPF.imn* (Slika 3.5). Svrha ovog primjera je pokazati što se događa u „tihoj“ mreži – kako usmjeritelji razmjenjuju informacije o svojim susjedima. Scenarij vježbe je sljedeći:

1. Započnite simulaciju.
2. Pokrenite alat *Wireshark* na sučelju *eth2* usmjeritelja *router3*.
3. Započnite snimanje mrežnog prometa alatom *Wireshark*.
4. Zaustavite snimanje prometa nakon otprilike 1 minute.
5. U alatu *Wireshark* otvorite jedan OSPF-paket te navedite o kojem je paketu riječ.
  - a. Navedite njegove izvorišnu i odredišnu IP-adresu.
  - b. Navedite koliko često se šalje.

6. Prikažite sadržaj odabranog paketa, provjerite OSPF-podatke i ukratko objasnite njihovu namjenu.

7. Da li OSPF, kao i RIP, koristi UDP kao transportni protokol?

8. Objasnite kako usmjeritelji pomoću protokola OSPF razmjenjuju informacije o metrici, te napravite usporedbu s protokolom RIP u tom pogledu.

9. Komentirajte značenje metrike kod protokola OSPF te ju usporedite s metrikom kod protokola RIP.

**Zadatak 27.** (Topologija *OSPF/OSPF1.imn*) Proučite primjer *OSPF/OSPF1.imn* (Slika 3.6). Svrha ovog primjera je pokazati što se događa kad neki usmjeritelj prestane raditi, pa zatim nakon nekog vremena opet započne s radom. U konzoli usmjeritelja možete koristiti naredbe navedene u tablici (Tablica 3.3).

Tablica 3.3: Naredbe za korištenje u konzoli usmjeritelja (protokol OSPF)

Naredba	Značenje naredbe
<b>show ip route</b>	prikazuje sve rute
<b>show ip ospf route</b>	prikazuje OSPF-rute
<b>show ip ospf interface</b>	prikazuje informacije o sučeljima usmjeritelja
<b>show ip ospf neighbor</b>	prikazuje informacije o susjedima usmjeritelja

Scenarij vježbe je sljedeći:

1. Započnite simulaciju.
2. Pokrenite alat *Wireshark* na sučelju *eth2* usmjeritelja *router2* i započnite snimanje mrežnog prometa.
3. Otvorite konzolu za upravljanje usmjeriteljem *router2* (desni klik mišem → *Shell window* → *vttysh*) i izvršite naredbu `show ip route`. Ova naredba će prikazati rute koje su zapisane u tablici usmjeravanja usmjeritelja *router2*. Analizirajte tablicu usmjeravanja, odnosno dostupne mreže, sljedeće „skokove“ (*hops*) i pripadajuću metriku.
4. Provjerite dostupnost (naredba *ping*) poslužitelja *server* (10.0.4.10) s računala *pc* (10.0.3.20), odnosno provjerite put do njega (naredba *traceroute*). Što zapažate? Komentirajte.
5. Zaustavite usmjeritelj *router7* (desni klik mišem → *stop*).
6. Promatrajte na konzoli usmjeritelja *router2* što se događa (koristeći naredbe navedene u tablici, Tablica 3.3). Primjetite vrijednost *Dead Time* prilikom izvršavanja naredbe `show ip ospf neighbor` i pratite što se s njom događa nakon što usmjeritelj *router7* prestane s radom.
7. Provjerite dostupnost (naredba *ping*) poslužitelja *server* (10.0.4.10) s računala *pc* (10.0.3.20), odnosno provjerite put do njega (naredba *traceroute*). Što zapažate? Komentirajte.
8. Promatrajte što se događa u vremenu nakon 3 minute. Da li se mreža rekonfigurira (ako je odgovor potvrđan - objasnite na koji način; ako je odgovor negativan - objasnite zašto).



9. Ponovno provjerite dostupnost (naredba *ping*) poslužitelja *server* (10.0.4.10) s računala *pc* (10.0.3.20), odnosno provjerite put do njega (naredba *traceroute*). Kojim putem sada putuju paketi?

10. Pokrenite opet usmjeritelj *router7* (desni klik mišem → *start*). Promatrajte što se događa s rutama. Komentirajte.

11. Ponovno provjerite dostupnost (naredba *ping*) poslužitelja *server* (10.0.4.10) s računala *pc* (10.0.3.20), odnosno provjerite put do njega (naredba *traceroute*). Komentirajte.

### 3.4 Pitanja

**Pitanje 6.** Adresa pod mreže u kojoj se nalazi računalo s adresom 121.63.91.181/26 glasi:

- a. 121.63.91.128
- b. 121.63.0.0
- c. 121.63.91.192
- d. 121.63.91.0

**Pitanje 7.** Zadano je računalo s IP-adresom 105.185.78.193/26. Koja od navedenih adresa može biti adresa podrazumijevanog (engl. *default*) usmjeritelja?

- a. 105.185.78.233
- b. 255.255.255.192
- c. 105.185.78.255
- d. 105.185.78.192

**Pitanje 8.** Ako usmjeritelj dobije paket u kojem je vrijednost polja TTL postavljena na 1, a on nije krajnje odredište, on:

- a. vrati taj paket pošiljatelju.
- b. izbací paket iz mreže i pošalje odgovarajuću poruku protokola ICMP pošiljatelju.
- c. usmjeri taj paket koristeći podrazumijevanu (engl. *default*) rutu.
- d. usmjeri taj paket koristeći prvi zapis iz tablice usmjeravanja.

**Pitanje 9.** Prilikom čitanja tablice usmjeravanja na usmjeritelju, podrazumijevana (engl. *default*) ruta:

- a. se uvijek razmatra prva.
- b. se uvijek razmatra posljednja.
- c. se može razmatrati u bilo kojem trenutku.
- d. nikad se ne razmatra.

**Pitanje 10.** Tablica usmjeravanja na razini protokola IP koristi se:

- a. u računalima, usmjeriteljima i komutatorima.
- b. u računalima i usmjeriteljima.
- c. samo u usmjeriteljima.
- d. samo u komutatorima.

### 3.5 Izvori

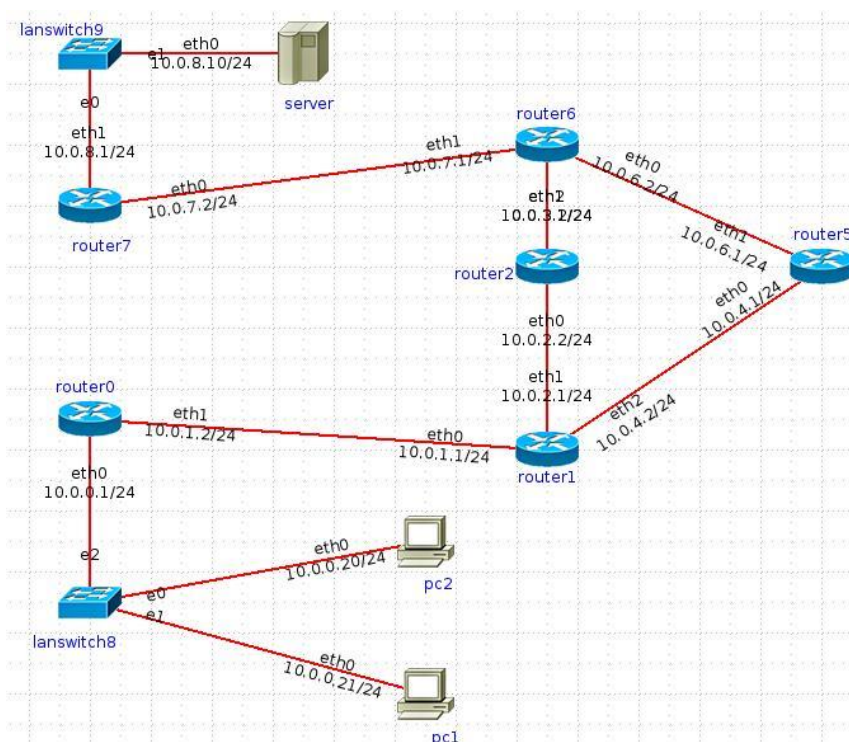
- IETF RFC 791 „Internet Protocol“: <http://tools.ietf.org/html/rfc791>
- IETF RFC 1518 „An Architecture for IP Address Allocation with CIDR“: <http://tools.ietf.org/html/rfc1518>
- IETF RFC 4632 „Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan“: <http://tools.ietf.org/html/rfc4632>
- IETF RFC 826 „An Ethernet Address Resolution Protocol“: <http://tools.ietf.org/html/rfc826>
- IETF RFC 792 „Internet Control Message Protocol“: <http://tools.ietf.org/html/rfc792>
- IETF RFC 2453 „RIP Version 2“: <http://tools.ietf.org/html/rfc2453>
- IETF RFC 2328 „OSPF Version 2“: <http://tools.ietf.org/html/rfc2328>
- Ping: <http://ftp.arl.mil/mike/ping.html>
- Traceroute: <http://kb.pert.geant.net/PERTKB/VanJacobsonTraceroute>

## 4 Protokoli transportnog sloja

Protokoli transportnog sloja u internetskim mrežama te posebno protokoli UDP i TCP objašnjeni su u udžbeniku *Komunikacijske mreže*<sup>6</sup>.

### 4.1 Konfiguracija eksperimenta

Na slici *Slika 4.1* prikazana je topologija mreže *Ping/ping.imn* koja će se koristiti u ovom poglavlju za primjere s protokolima transportnog sloja. Mreža se sastoji od dva računala, jednog poslužitelja, dva komutatora i šest usmjeritelja.



Slika 4.1: Topologija mreže *Ping/ping.imn*

### 4.2 Objašnjenja korištenih pojmova, koncepata, protokola i alata

U ovom poglavlju opisane su osnove protokola transportnog sloja te je na primjeru korištenja alata *netcat* pokazano koja je uloga istih kod prijenosa podataka u mrežama.

Protokoli koji su na raspolaganju za obavljanje funkcije transportnog sloja u Internetu su TCP (*Transmission Control Protocol*) i UDP (*User Datagram Protocol*). U nastavku su, kroz niz primjera, ilustrirana svojstva dvaju navedenih protokola. Za analizu njihova rada koristit će se analizator

<sup>6</sup> Lovrek, Matijašević, Ježić, Jevtić: “Komunikacijske mreže”, Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva (2019) (*trenutno dostupna radna inačica udžbenika*)

mrežnog prometa *Wireshark* i alat *netcat*, koji će poslužiti kao jednostavan generator prometa protokola TCP i UDP.

#### 4.2.1 Protokol UDP

Protokol UDP omogućava prijenos paketa između procesa koji se odvijaju na različitim računalima. Karakteristika protokola UDP je da on ni na koji način ne garantira dostavu paketa na odredište. Također, UDP ne garantira da redoslijed kojim su paketi primljeni odgovara redoslijedu kojim su oni poslani. Umjesto toga, UDP jednostavno šalje sve što se od njega zatraži, ne očekujući potvrde niti ispravljajući pogrešan redoslijed isporuke paketa. S obzirom na ova svojstva, UDP je pogodniji za aplikacije kod kojih se mrežom prenosi višemedijski sadržaj (video, audio i sl.) u stvarnom vremenu, jer kod ovih aplikacija postoji izvjesna tolerancija na ispuštene pakete, odnosno pakete isporučene van redoslijeda.

Već je utvrđeno da protokol IP omogućava razmjenu paketa između računala u mreži. Ukoliko neko računalo, npr., računalo A, želi poslati podatke na drugo računalo, npr., na računalo B, formira se IP-datagram, kao izvorišna adresa paketa upisuje se IP-adresa računala A, za odredišnu adresu stavlja se IP-adresa računala B i paket se proslijeđuje prvom čvoru na putu prema B.

Međutim, s obzirom da se na jednom računalu može odvijati više međusobno neovisnih procesa (npr., različite korisničke aplikacije) koji imaju potrebu komunicirati s odgovarajućim procesima na drugom računalu, sama IP-adresa u paketima koji pristižu na računalo nije dovoljna da bi se odredilo kojem procesu je potrebno dostaviti sadržaj primljenih paketa. IP-adresa je paket „dovela“ do računala, ali unutar računala ima više potencijalnih primatelja, odnosno, procesa. Stoga je, osim IP-adrese, u komunikaciju između procesa potrebno uvesti dodatnu oznaku koja određuje proces na računalu kojem je potrebno dostaviti sadržaj IP-paketa. Ova oznaka naziva se vrata (engl. *port*).

#### 4.2.2 Protokol TCP

Protokol TCP omogućava pouzdan i slijedni prijenos niza okteta između udaljenih procesa. Sa stajališta procesa koji razmjenjuju niz okteta, TCP predstavlja pouzdan „tunel“ kroz koji je oktete moguće prenijeti bez gubitaka i u očuvanom redoslijedu. Da bi omogućio ovakvu vrstu usluge procesima koji ga koriste, TCP interno koristi niz složenih mehanizama. Iako su ovi mehanizmi u potpunosti transparentni za procese koji koriste TCP-usluge, korisno ih je poznavati jer oni uvelike utječu na brzinu i kašnjenje pri prijenosu podataka između komunicirajućih procesa. Tako je, na primjer, zbog ponovljenog slanja izgubljenih paketa i oscilacije kašnjenja u isporuci paketa, nepovoljno koristiti TCP za prijenos podataka osjetljivih na kašnjenje. To su, primjerice, višemedijske aplikacije za prijenos govora i videa u stvarnom vremenu.

Za razliku od protokola UDP, TCP pruža bitno složeniju uslugu. Prije svega, TCP osigurava pouzdan prijenos niza okteta čuvajući njihov redoslijed. Pojam pouzdan prijenos znači da će, u slučaju gubitka paketa u mreži, TCP ponavljati slanje izgubljenih paketa sve dok se svi ne prenesu na odredište. Pojam prijenos niza okteta odnosi se na činjenicu da TCP procesima isporučuje jedan po jedan oktet, ne pružajući informaciju o načinu na koji su skupine okteta, prilikom prijenosa kroz mrežu, grupirane u IP-datagrame. Nadalje, TCP osigurava isporuku okteta u istom redoslijedu kojim su oni i poslani. Bez obzira na činjenicu da se u IP-mrežama relativno često događa da paketi na odredište stižu različitim redoslijedom od onoga u kojem su poslani, TCP „ispravlja“ redoslijed IP-paketa i procesu isporučuje „izvorni“ slijed okteta.

Da bi osigurao pouzdan prijenos okteta između procesa, TCP koristi mehanizam potvrđivanja. Ispravan primitak svakog poslanog okteta mora biti potvrđen od strane primatelja. Ukoliko pošiljatelj ne dobije potvrdu o primitku okteta koje je već poslao, ponavlja se slanje istih okteta sve dok se ne dobije potvrda da ih je odredište primilo. U zaglavlju TCP-a za potvrđivanje se koristi posebno polje ACK. Prije slanja okteta između udaljenih procesa, TCP mora uspostaviti vezu. Kao i kod protokola UDP, TCP-veza je jednoznačno određena četvorkom:

*{ izvorišna IP-adresa, izvorišna vrata, odredišna IP-adresa, odredišna vrata }.*

Jedna od bitnih funkcija koju obavlja TCP je i tzv. kontrola toka. Radi se o sljedećem. Pretpostavimo da jedan proces šalje određenu količinu podataka nekom drugom procesu, te da se proces koji šalje podatke (pošiljatelj) nalazi na računalu koje je u mrežu spojeno vezom kapaciteta 100 Mbit/s, dok se proces koji prima podatke (primatelj) nalazi na računalu koje je spojeno vezom kapaciteta 10 Mbit/s. S obzirom da je pošiljatelj spojen vezom od 100 Mbit/s, on bi mogao slati podatke tom brzinom, ali ti podaci ne bi mogli biti isporučeni primatelju istom brzinom, jer je pristup do primatelja brzine 10 Mbit/s. Kako pošiljatelj zna kojom brzinom smije slati podatke, a da ih primatelj stigne obraditi? Kontrola toka koju obavlja TCP upravo rješava ovaj problem.

Protokol TCP koristi mehanizam klizećeg prozora (engl. *sliding window*) za kontrolu toka. Primatelj je zadužen za kontinuirano obavješćavanje pošiljatelja o količini podataka koju je on trenutno u stanju obraditi. Pošiljatelj nikad neće poslati više podataka od ove količine, bez da mu primatelj potvrdi primitak podataka. Na primjer, pri uspostavi TCP-veze obje strane objavljuju jedna drugoj koliko su podataka spremne primiti u danom trenutku. Ova veličina naziva se prozor. Svaka strana smije odjednom poslati samo toliko podataka koliko je druga strana objavila u prozoru. Nakon što je poslala tu količinu podataka, strana pošiljatelja mora čekati potvrdu da je barem jedan dio podataka primljen na odredištu. Kad dobije potvrdu, pošiljatelj može poslati dodatnu količinu podataka, ali samo onoliko novih koliko je okteta potvrđeno. Na taj se način u mreži, u svakom trenutku, nalazi najviše onoliko nepotvrđenih podataka kolika je veličina prozora. S obzirom da pošiljatelj ne smije slati nove podatke dok primatelj ne potvrdi primitak starih, primatelj određuje brzinu kojom mu pošiljatelj šalje podatke.

Osim pouzdanog i slijednog prijenosa, protokol TCP vodi računa i o kontroli zagušenja u mreži izazvanog TCP-prometom te o ravnopravnosti podjele mrežnih kapaciteta između konkurentnih TCP-veza. Tematika kontrole zagušenja je složena, te neće ovdje biti detaljnije razmatrana.

#### 4.2.3 Alat netcat

Alat *netcat* implementira funkcije klijenta i poslužitelja u Internetu te omogućava prijenos proizvoljnih podataka između ta dva entiteta. Alat *netcat* može koristiti protokol TCP ili UDP za prijenos podataka između klijenta i poslužitelja. Osnovne opcije alata su navedene u tablici (Tablica 4.1).

Tablica 4.1: Osnovne opcije alata netcat

Opcija	Značenje opcije
<b>-l</b>	pokreće alat u poslužiteljskom ( <i>listen</i> ) načinu rada
<b>-u</b>	koristi UDP za prijenos podataka (umjesto podrazumijevanog TCP-a)

Alat *netcat* uobičajeno se pokreće izvršavanjem naredbe `nc`. Na jednom računalu prvo treba pokrenuti alat *netcat* u tzv. *listen* načinu rada, uz navođenje odgovarajućih vrata (engl. *port*) na kojima *netcat* „sluša“ zahtjeve za uspostavom veze, a na drugom računalu treba pokrenuti alat *netcat* te od njega zahtijevati uspostavu veze prema prvom računalu i vratima na kojima alat *netcat* sluša. Uspostavljena veza se prekida korištenjem kombinacije tipki `Ctrl+C`. Detaljan opis korištenja alata *netcat* dobiva se izvršavanjem naredbe `man nc`.

Pretpostavimo da želimo prenijeti neke podatke između računala *PCI* i računala *server* koji se nalaze u mreži prikazanoj slikom . Mreža se učitava u IMUNES iz datoteke *imunes-examples/Ping/ping.imn*. Nakon pokretanja simulacije, u konzoli računala *server* se izvrši naredba

```
# nc -l 100
```

čime se pokrene „slušanje“ na TCP-vratima 100. Nakon toga, izvođenjem naredbe u konzoli računala *PCI*

```
# nc 10.0.8.10 100
```

uspostavlja se veza od računala *PCI* prema računalu s danom IP-adresom (u ovom slučaju radi se o računalu *server*), na dana TCP-vrata (u ovom slučaju 100).

Alat *netcat* očekuje podatke na svom standardnom ulazu, i onda te podatke šalje po uspostavljenoj vezi. Nadalje, sve što alat primi putem uspostavljene veze, ispisuje na standardni izlaz. Standardni ulaz i izlaz se obično poistovjećuju s tipkovnicom i zaslonom, te se čitanje sa standardnog ulaza svodi na upisivanje podataka putem tipkovnice, dok se ispisivanje na standardni izlaz svodi na ispisivanje na zaslon. Međutim, korištenjem operatora „<“ i „>“ moguće je ulaz i izlaz preusmjeriti iz datoteke, odnosno u datoteku. Na primjer, izvršavanje naredbe

```
# nc 10.0.8.10 100 < COPYRIGHT
```

spaja se na računalu s IP-adresom 10.0.8.10, i to na TCP-vrata 100, te se sadržaj datoteke *COPYRIGHT* (koju možete ispisati izvršavanjem naredbe: `cat COPYRIGHT`) šalje računalu kao da je upisan s tipkovnice. Nadalje, izvođenjem naredbe

```
# nc -l 100 > file.txt
```

sve što klijent pošalje, umjesto na zaslon, upisat će se u datoteku *file.txt*. Za stvaranje datoteke proizvoljnog sadržaja može se koristiti naredba `echo` koja jednostavno ispisuje svoje argumente na standardni izlaz. Preusmjeravanjem standardnog izlaza može se ostvariti pisanje u datoteku. Na primjer, za stvaranje datoteke *README* čiji je sadržaj tekst "neki tekst", može se koristiti sljedeća naredba:

```
# echo "neki tekst" > README
```

Koristeći opisane naredbe, moguće je kopirati neku datoteku, primjerice datoteku *boot.conf* s računala *PCI* na računalu *server*. Prvo treba na računalu *server* kreirati datoteku s proizvoljnim imenom u koju će se kopirati sadržaj originalne datoteke. To se radi izvođenjem naredbe

```
# touch boot1.conf
```

Datoteka se kopira pokretanjem alata *netcat* pomoću sljedećih naredbi:

```
- na računalu server: # nc -l 100 > boot1.conf
```

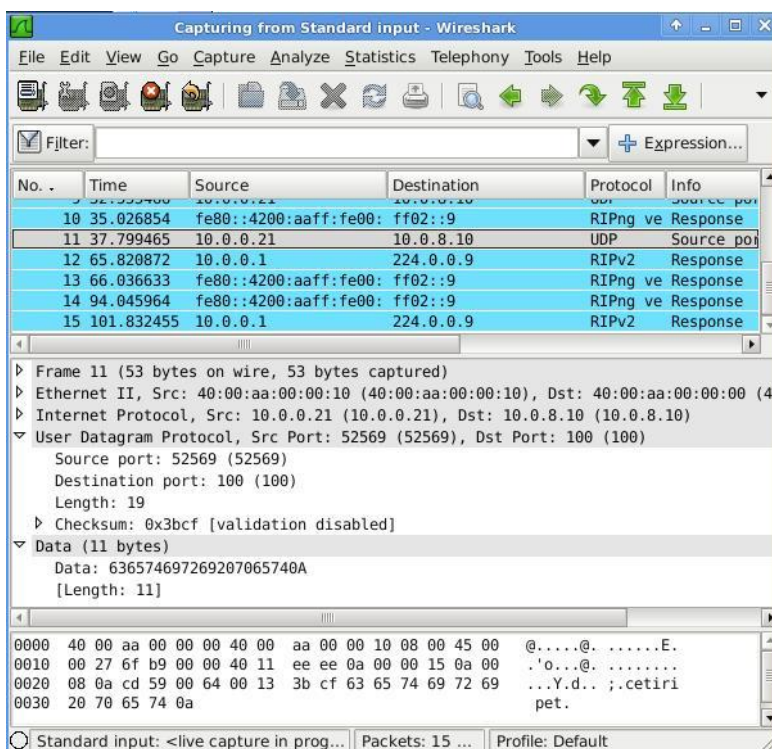
- na računalu *PCI*: `# nc 10.0.8.10 100 < boot.conf`

Otvaranjem datoteka na oba računala uvjerite se da je datoteka doista kopirana (za više detalja o dostupnim uređivačima teksta pogledajte poglavlje 1.1.2 ovog udžbenika). U ovom primjeru alat *netcat* za prijenos podataka je koristio podrazumijevani protokol TCP. Ako se želi podatke prenijeti protokolom UDP, to je potrebno eksplicitno navesti dodavanjem naredbe u kako slijedi:

- na računalu *server*: `# nc -lu 100 > boot1.conf`
- na računalu *PCI*: `# nc -u 10.0.8.10 100 < boot.conf`

Ukoliko se na nekom od usmjeritelja snimi generirani promet, dobije se ispis kao na slici Slika 4.2. Podrobnijom analizom jednog UDP-paketa, može se primijetiti da protokol UDP dodaje svoje zaglavlje te tijelo poruke u koje smješta tekst koji se prenosi. Zaglavlje i tijelo UDP-poruke smješta se unutar tijela IP-datagrama, te se na to dodaje IP-zaglavlje. Samo UDP zaglavlje sastoji se od nekoliko polja, od kojih su najvažnija polja s izvorišnom i odredišnom adresom. S obzirom da UDP adresira procese na krajnjim računalima, ovdje se koriste UDP-vrata kao adrese. Odredišna adresa su ona UDP-vrata na kojima računalu *server* očekuje zahtjeve. U ovom primjeru radi se o UDP-vratima 100. Kao izvorišna adresa koriste se bilo koja UDP-vrata koja su na računalu *PCI* slobodna. U ovom primjeru radi se o UDP-vratima 52569. Možemo primijetiti da se unutar poruke protokola UDP ne zapisuje niti izvorišna niti odredišna IP-adresa. Ti podaci smještaju se u zaglavlje protokola mrežnog sloja, u ovom slučaju protokola IP.

Osim polja s adresama, postoje još i polje s podatkom o duljini UDP-paketa te polje sa zaštitnom sumom zaglavlja. S obzirom da je UDP nespojno orijentiran protokol, zaglavlje ne sadrži slijedni broj paketa, potvrdu o prethodno primljenoj poruci te ostale funkcionalnosti karakteristične protokolu TCP.



Slika 4.2: Analiza sadržaja UDP-paketa alatom Wireshark

### 4.3 Eksperimenti i zadaci

**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*).

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

**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.

**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.

**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.

**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?

**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 potvrdom i veličinom prozora.

### 4.4 Pitanja

**Pitanje 11.** Brojevi vrata (engl. *port*), koji zapravo predstavljaju transportnu adresu asociranu s procesom na računalu, mogu biti u rasponu:

- a. od 0 do 1023
- b. od 1 do 254
- c. od 0 do 65535
- d. od 0 do 255



**Pitanje 12.** Kojem sloju protokolnog složaja internetske mreže (TCP/IP) pripada protokol UDP (*User Datagram Protocol*)?

- a. Protokoli se ne povezuju sa slojevima TCP/IP-složaja.
- b. Sloju prezentacije.
- c. Transportnom sloju.
- d. Mrežnom sloju.

**Pitanje 13.** Koja je od sljedećih tvrdnji točna?

- a. Protokol TCP ne mora uspostavljati vezu ako se zahtijeva prijenos manje od 2 okteta.
- b. Protokol TCP mora uspostavljati vezu čak i u slučaju da se prenosi samo jedan oktet korisničkih podataka.
- c. TCP-veza se uspostavlja posebno za svaki oktet koji je potrebno prenijeti.
- d. TCP-veza se uspostavlja posebno za svaki TCP-segment koji je potrebno prenijeti.

**Pitanje 14.** Mehanizam klizećeg prozora (engl. *sliding window*), koji koristi protokol TCP, služi za:

- a. uspostavu veze.
- b. raskid veze.
- c. upravljanje retransmisijom segmenata.
- d. upravljanje tokom.

**Pitanje 15.** Korištenje protokola UDP za prijenos datoteka:

- a. je bolje na "duljim" putevima kroz mrežu, jer se ne gubi vrijeme na potvrđivanje.
- b. je bolje od korištenja protokola TCP, jer se ne gubi vrijeme na uspostavu veze.
- c. nije moguće ostvariti, jer protokol IP to ne dopušta.
- d. nije prikladno.

## 4.5 Izvori

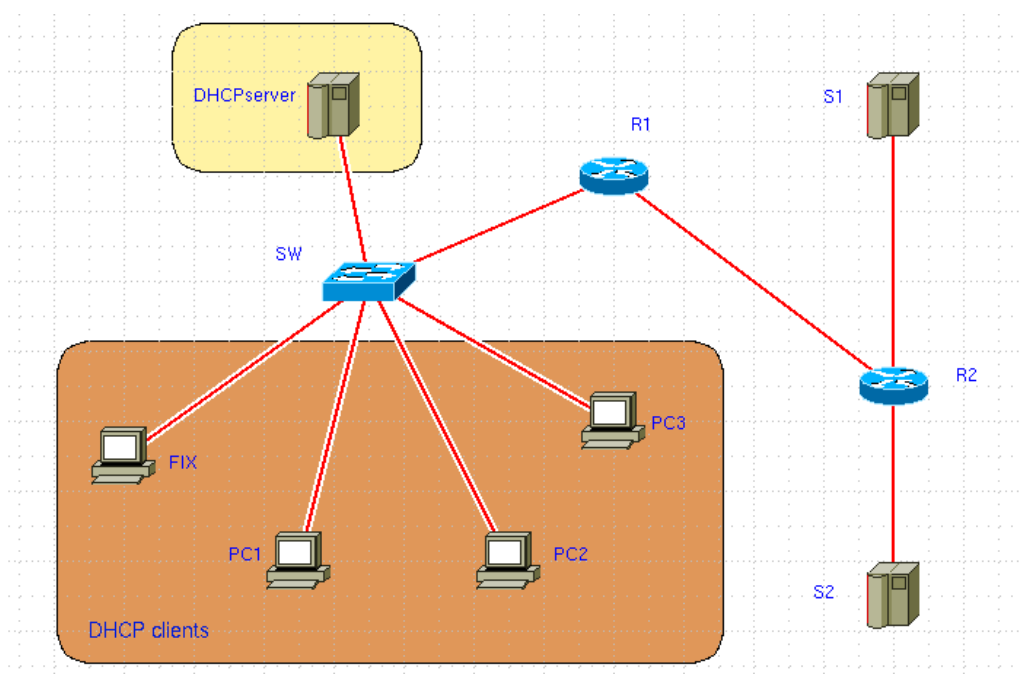
- IETF RFC 768 „User Datagram Protocol“: <http://tools.ietf.org/html/rfc768>
- IETF RFC 793 „Transmission Control Protocol“: <http://tools.ietf.org/html/rfc793>
- Tcpcat: <http://www.tcpcat.org>
- Netcat: <http://netcat.sourceforge.net>

## 5 Protokoli aplikacijskog sloja

Protokoli aplikacijskog sloja u internetskim mrežama objašnjeni su u udžbeniku *Komunikacijske mreže*<sup>7</sup>.

### 5.1 Konfiguracija eksperimenta

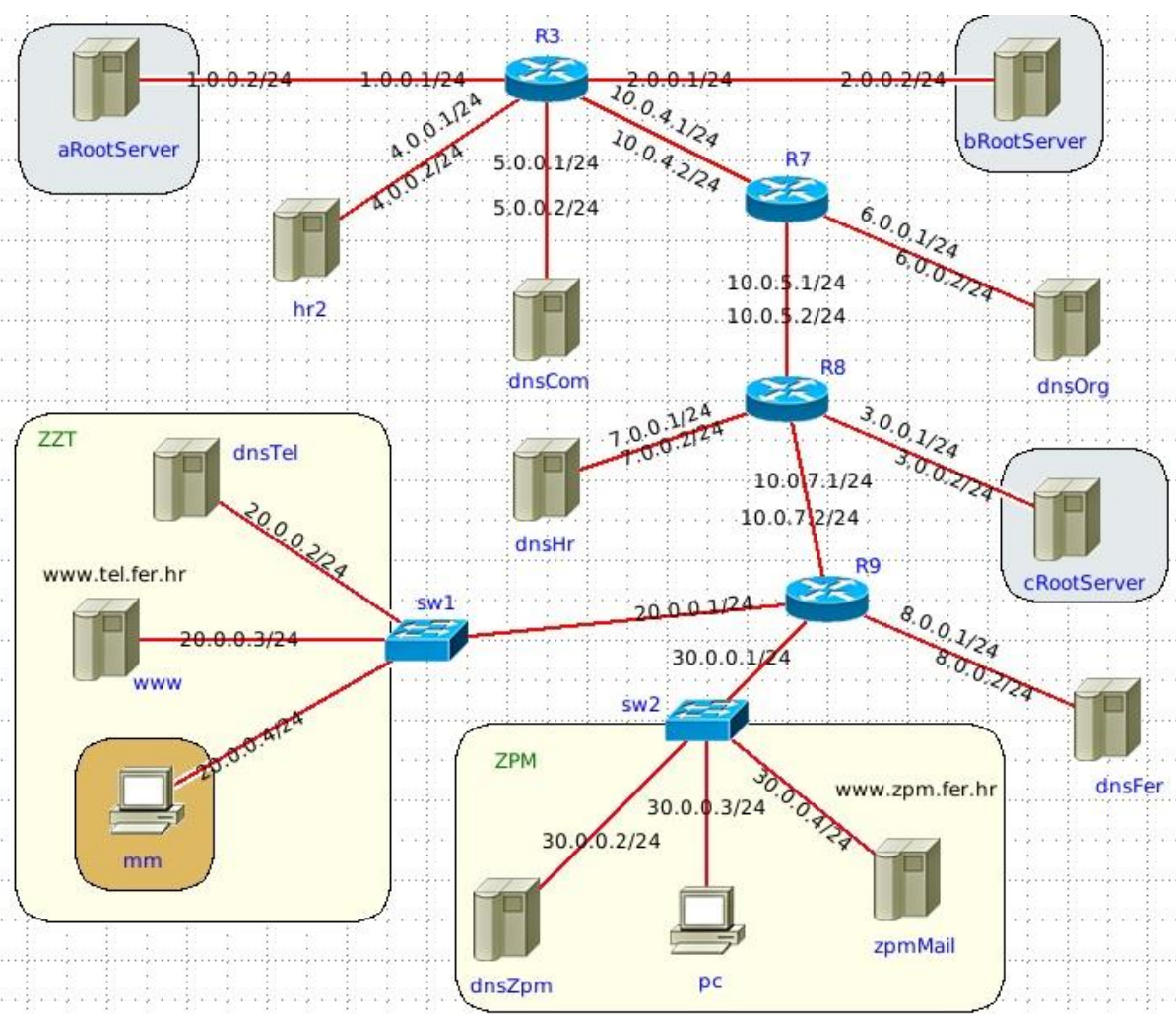
Na slici Slika 5.1 prikazana je topologija *DHCP/DHCP.imn* u kojoj je unaprijed podešen poslužitelj protokola DHCP te se omogućuje praćenje procesa zahtijevanja i dodjele IP-adrese protokolom DHCP. Na slici Slika 5.2 prikazana je topologija *DNS+Mail+Web/Network.imn* koja sadrži određeni broj DNS-poslužitelja, jedan HTTP-poslužitelj i nekoliko poslužitelja elektroničke pošte, a uz nju dolaze i pripadne skripte *start\_dns*, *start\_http* i *start\_mail* koje je potrebno izvršiti prije pojedinih eksperimenata, prema uputama koje slijede.



Slika 5.1: Topologija *DHCP/DHCP.imn*

---

<sup>7</sup> Lovrek, Matijašević, Ježić, Jevtić: “Komunikacijske mreže”, Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva (2019) (*trenutno dostupna radna inačica udžbenika*)



Slika 5.2: Topologija DNS+Mail+Web/Network.imn

## 5.2 Objašnjenja korištenih pojmova, koncepata, protokola i alata

U ovom poglavlju opisani su protokoli DHCP, DNS i HTTP te protokoli za slanje i primanje elektroničke pošte.

Protokoli TCP i UDP omogućavaju prijenos informacija između dva udaljena procesa. Na toj podlozi aplikacijski sloj definira komunikaciju potrebnu za realizaciju različitih usluga u mreži. Na primjer, usluga elektroničke pošte se temelji na protokolu SMTP (*Simple Mail Transfer Protocol*) koji definira slanje elektroničke pošte od strane klijenta i njezino primanje od strane SMTP-poslužitelja. S druge strane, protokol HTTP (*HyperText Transfer Protocol*) definira uslugu prijenosa web sadržaja između računala. Ta dva protokola za prijenos vlastitih informacija koriste protokol TCP.

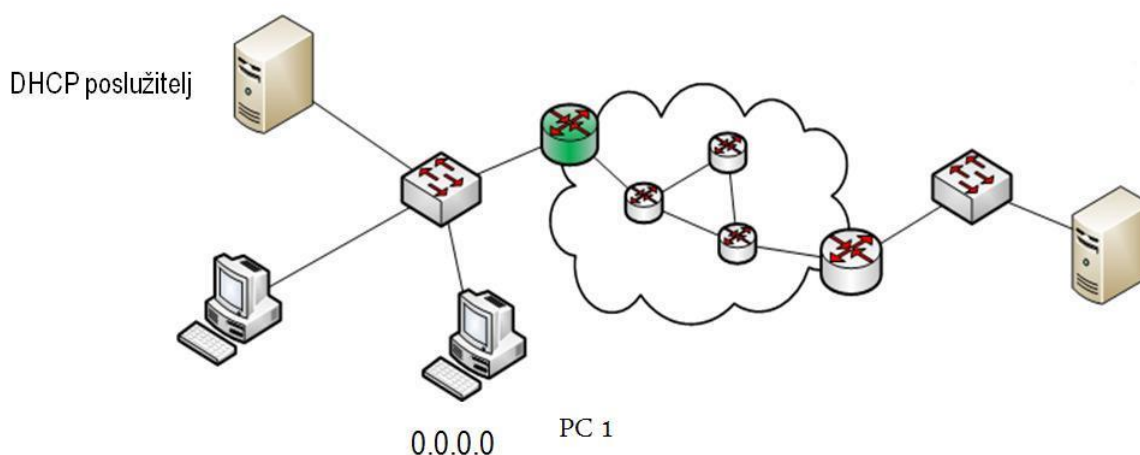
Cilj ovog poglavlja je upoznavanje s osnovnim načelima rada protokola DHCP (*Dynamic Host Configuration Protocol*), DNS (*Domain Name System*), SMTP, POP (*Post Office Protocol*) i HTTP. Mreža, pomoću koje se rješavaju zadaci u emulatoru/simulatoru IMUNES vezani uz aplikacijske protokole, nalazi se u datotekama *DHCP/DHCP.imn* i *DNS+Mail+WEB/NETWORK.imn*. Kao korisničko ime, gdje je to potrebno, koristite *root*, a kao lozinku *imunes*.

### 5.2.1 Protokol DHCP

U dosadašnjim primjerima smo pretpostavljali da su IP adrese mrežnim sučeljima pridijeljene trajno (statičke IP adrese), npr., od strane mrežnog administratora. U praksi se računalima adrese dodjeljuju dinamički, pomoću protokola DHCP (*Dynamic Host Configuration Protocol*). Računala prilikom svakog novog priključivanja u lokalnu mrežu, dobivaju nove IP adrese (dinamičke IP adrese). Nadalje, protokol DHCP služi i za dinamičko podešavanje ostalih mrežnih parametara računala u internetskoj mreži: maske podmreže, podrazumijevanog odlaznog sučelja (engl. *gateway*), adrese DNS-poslužitelja, itd.

Bitno je naglasiti da protokol DHCP računalima omogućava najam (engl. *lease*) određene IP adrese neki period vremena. Ako računalo nije spojeno u podmrežu dulje od vremena određenog najmom (engl. *lease time*), najam adrese ističe i ona može biti pridijeljena nekom drugom mrežnom sučelju. Vrijeme najma ovisi o vrsti i namjeni podmreže: od razine sata (npr., bežična mreža u zračnoj luci) do razine mjesec dana (npr., računala u istraživačkom laboratoriju).

Budući da u svom radu koristi i mehanizme transportnog sloja internetskog (TCP/IP) složaja, DHCP se logički smješta u aplikacijski sloj. Za ispravan rad protokola potreban je DHCP-poslužitelj koji se smješta u podmrežu za koju je nadležan (Slika 5.3. Poslužitelju je na raspolaganju određen broj IP-adresa (obično iz zadanog raspona) koje prema potrebi dodjeljuje računalima. Pritom se koristi dvama spremnicima: spremnikom za pohranu nedodijeljenih adresa i spremnikom za pohranu dodijeljenih adresa, kako bi u svakom trenutku znao koja adresa je dodijeljena kojem računalu, te koje adrese su slobodne.



Slika 5.3: Smještaj DHCP-poslužitelja u lokalnoj mreži

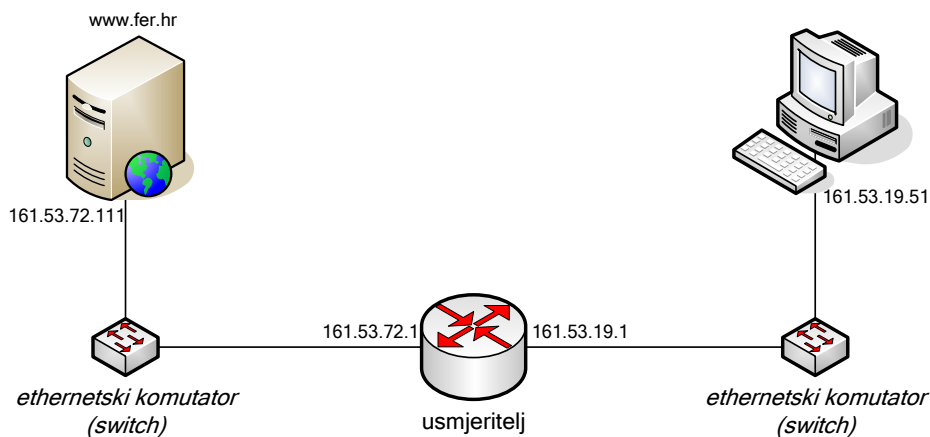
Pretpostavimo da se računalo *PC1* uključuje u mrežu prikazanu slikom Slika 5.3. U procesu dodjele IP-adresa izmjenjuje se 4 vrste poruka:

- *DHCP Discover* - Poslan od strane računala *PC1* s izvorišnom IP-adresom 0.0.0.0 (jer računalo tek treba dobiti adresu) i odredišnom adresom postavljenom na *broadcast* adresu 255.255.255.255 (računalo *PC1* ne zna adresu DHCP-poslužitelja, pa mora adresirati sva računala),

- *DHCP Offer* - DHCP poslužitelj šalje ponuđenu IP-adresu, podatke o nadležnim DNS-poslužiteljima, podrazumijevanom usmjeritelju i mrežnoj masci,
- *DHCP Request* - Računalo *PC1* šalje na *broadcast* adresu zahtjev za ponuđenom adresom,
- *DHCP ACK* - DHCP-poslužitelj odobrava zahtjev za adresom, definira vrijeme „najma“ adrese, te potvrđuje podatke o podrazumijevanom usmjeritelju i DNS-poslužiteljima.

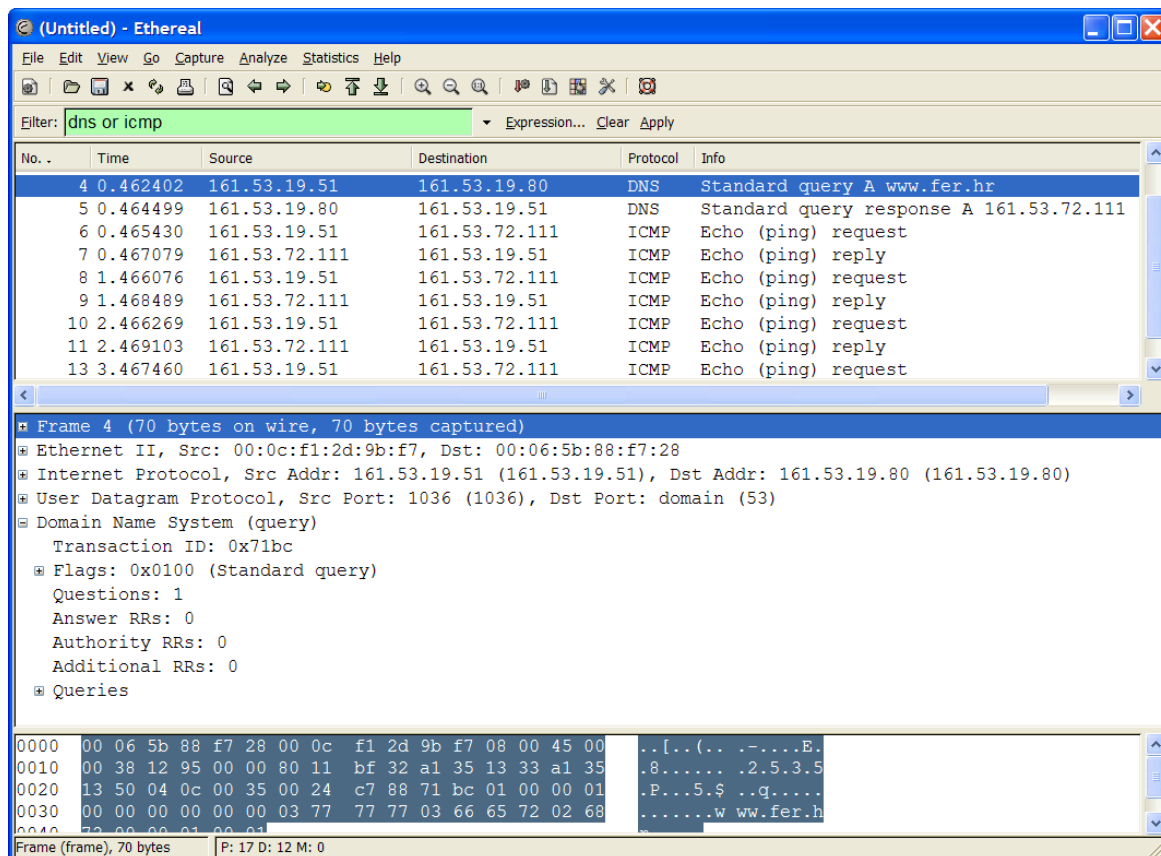
### 5.2.2 Protokol DNS

Pokrenimo alat *Wireshark* na računalu 161.53.19.51 (Slika 5.4) i u komandnoj liniji nakon toga pokrenimo izvršavanje naredbe `ping www.fer.hr`. Nakon što je naredba *ping* završila svoj rad (tj., nakon što smo ju prekinuli s kombinacijom tipki `CTRL+C`), zaustavimo snimanje mrežnog prometa u alatu *Wireshark*. Otvara se prozor s prikazom svih snimljenih paketa. S obzirom da se u mreži odvijaju razne aktivnosti, vrlo je vjerojatno da će osim prometa koji je u mreži generirala naredba *ping* biti uhvaćen i drugi mrežni promet. Da bismo lakše pratili onaj dio koji nas zanima, filtrirat ćemo uhvaćeni promet i to tako da u polje *filter*, koje se nalazi neposredno ispod glavnog izbornika alata *Wireshark*, upišemo *icmp or dns*. Zašto baš ova dva protokola, vidjet ćemo u nastavku.



Slika 5.4: Topologija ispitivane mreže

Prikaz snimljenih paketa u alatu *Wireshark* dan je na sljedećoj slici (Slika 5.5). Prva dva uhvaćena paketa pripadaju sustavu DNS. Protokol IP radi isključivo s IP-adresama. Za njega niz znakova *www.fer.hr* ne predstavlja adresu i paket ne može biti poslan na adresu čija je vrijednost *www.fer.hr*. IP-adresa računala koje se naziva *www.fer.hr* je 161.53.72.111. Kada se u komandnoj liniji napiše `ping www.fer.hr`, ono što alat *ping* prvo napravi je da za ime *www.fer.hr* pokuša saznati IP-adresu. Prva dva paketa snimljena alatom *Wireshark* predstavljaju ovaj događaj. Na računalu na kojem je izvedena naredba *ping* administrativno je podešeno da se za sva pitanja u vezi imena računala i njihovih IP-adresa treba obratiti na adresu 161.53.19.80. Stoga je prvi paket upućen na adresu 161.53.19.80, a pitanje je glasil: „Kuju IP-adresu ima računalo čije ime je *www.fer.hr*?” U sljedećem retku prikazan je odgovor koji je stigao s adrese 161.53.19.80, a on glasi 161.53.72.111.



Slika 5.5: Promet generiran prilikom izvršavanja naredbe  
# ping www.fer.hr

Nakon što je alat *ping* uspio saznati IP-adresu odredišnog računala, na tu je odredišnu adresu upućena ICMP poruka *Echo Request*.

Topologija mreže podešena za analizu protokola DNS nalazi se u datoteci *imunes-examples/DNS+Mail+WEB/NETWORK.imn* (Slika 5.2). Za ispravno generiranje DNS-prometa potrebno je na odgovarajućim računalima pokrenuti DNS-poslužitelje. To se radi u sljedećim koracima:

1. Pokrenuti simulaciju nakon učitavanja mreže
2. U konzoli (*Applications Menu -> Terminal Emulator*) računala na kojem je pokrenut IMUNES, pozicionirati se u direktorij *DNS+Mail+WEB*:

```
# cd /root/imunes-examples/DNS+Mail+WEB/
```

3. Pokrenuti DNS-poslužitelje naredbom:

```
# ./start_dns
```

Naredbom

```
# ./start_dns
```

konfigurirat će se odgovarajući poslužitelji i klijenti u mreži za potrebe ove analize. Na taj način svako računalo je u mogućnosti poslati neki DNS-upit te potom primiti DNS-odgovor od svog nadležnog DNS-poslužitelja. Na klijentskim računalima postoji alat koji omogućuje saznavanje informacija o imenima i IP-adresama ostalih računala, zatim nadležnim poslužiteljima elektroničke pošte te nadležnim DNS-poslužiteljima. Alat se izvodi pokretanjem naredbe *host*, koja potom generira određeni DNS-upit i šalje ga u mrežu. Naredba *host* može se pokrenuti korištenjem više opcija, od kojih su najvažnije ispisane u tablici (Tablica 5.1). Ostale opcije naredbe *host* mogu se saznati pokretanjem naredbe

```
# man host
```

Tablica 5.1: Opcije naredbe *host* za slanje DNS-upita

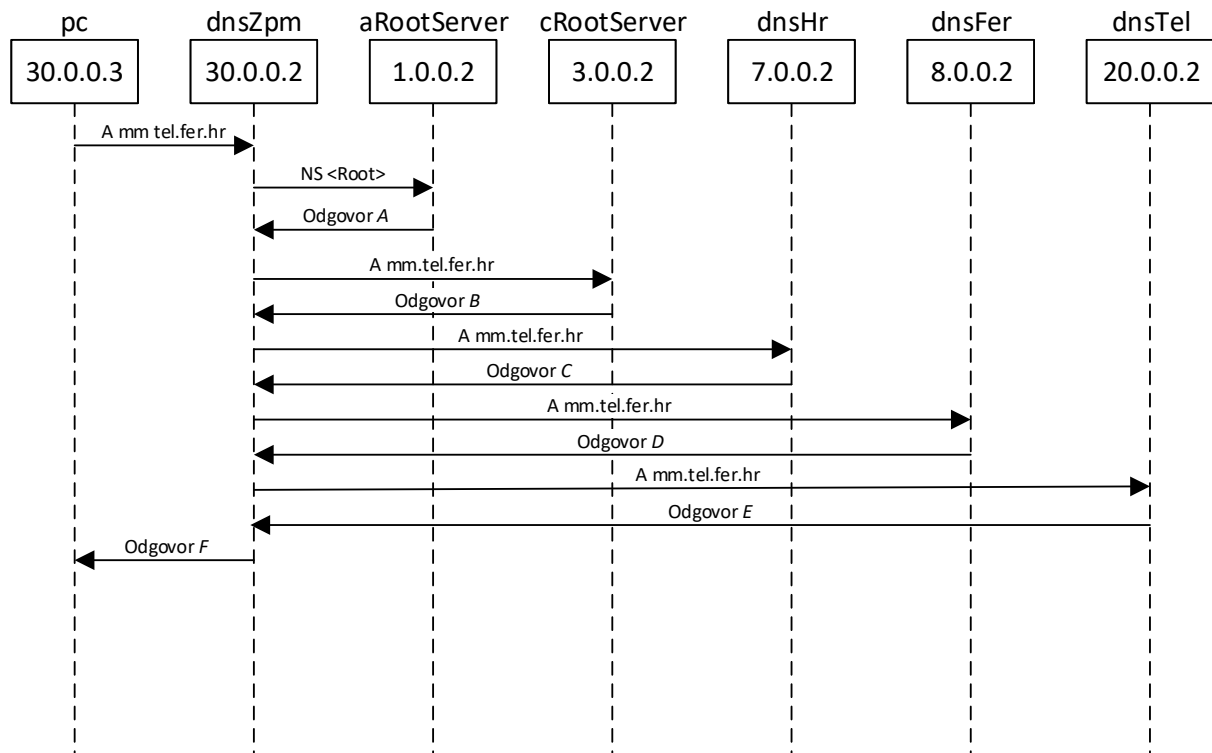
Naredba	Značenje naredbe
<b>host -t A mm.tel.fer.hr</b>	traži se IP-adresa računala sa zadanim imenom
<b>host -t MX tel.fer.hr</b>	traži se računalo nadležno za primanje pošte
<b>host -t NS fer.hr</b>	traži se nadležni DNS-poslužitelj
<b>host -t PTR 10.0.0.1</b>	traži se ime računala sa zadanom IP-adresom

Analizirajmo rad protokola DNS pomoću primjera. Uključimo snimanje prometa alatom Wireshark na računalo *dnsZpm.zpm.fer.hr*. Otvorimo konzolu računala *pc.zpm.fer.hr* te saznajmo IP-adresu računala *mm.tel.fer.hr* izvođenjem naredbe

```
# host -t A mm.tel.fer.hr
```

U konzoli se ispisuje tražena IP-adresa: 20.0.0.4. Analizirajući snimljeni promet može se identificirati slijed poruka protokola DNS izmijenjenih u mreži od trenutka slanja upita s računala *pc* do primitka odgovora (Slika 5.6). Računalo *pc* šalje DNS-upit svom nadležnom DNS-poslužitelju: računalo *dnsZpm*. S obzirom da *dnsZpm* u svom priručnom spremniku nema traženu adresu, šalje dodatne upite u mrežu kako bi je saznao. Prvo se obraća jednom od vršnih (korijenskih) DNS-poslužitelja: računalo *aRootServer*, od kojega traži popis vršnih poslužitelja kako bi ažurirao vlastiti popis koji ima pohranjen. Nakon što od vršnog poslužitelja dobije popis dostupnih vršnih poslužitelja (Odgovor A), šalje upit jednom od njih, u ovom slučaju vršnom poslužitelju *cRootServer*, tražeći IP-adresu računala *mm.tel.fer.hr*. S obzirom da vršni DNS-poslužitelj mora imati pohranjene zapise samo o DNS-poslužiteljima vršnih domena, niti on ne zna traženu adresu. Stoga šalje poruku *Odgovor B* računalo *dnsZpm*.





Slika 5.6: Slijedni dijagram DNS-upita i DNS-odgovora za upit  
# host –t A mm.tel.fer.hr

Analiziramo li sadržaj poruke *Odgovor B* (Slika 5.7), vidimo karakteristična polja protokola DNS. Primjerice, polje *Queries* sadrži inicijalni upit za adresom računala *mm.tel.fer.hr*. Polje *Authoritative Nameservers* sadrži podatke o sljedećem DNS-poslužitelju (ili nekolicini njih) kojem je potrebno proslijediti upit. U ovom slučaju radi se o vršnim poslužiteljima domene *hr*: računalima *dnsHr* i *hr2*. Nakon što primi *Odgovor B*, računalo *dnsZpm* ponovno šalje isti upit, ali ovaj put nekom od „predloženih“ DNS-poslužitelja. U ovom slučaju upit se šalje računalu *dnsHr*. Na jednak način, porukom *Odgovor C* računalo *dnsHr* vraća adresu DNS-poslužitelja nadležnog za domenu *fer.hr*, a porukom *Odgovor D* računalo *dnsFer* vraća adresu DNS-poslužitelja nadležnog za domenu *tel.fer.hr*. S obzirom da ime traženog računala pripada domeni za koju je zadužen DNS-poslužitelj *dnsTel*, poruka *Odgovor E* će vratiti traženu IP-adresu. Konačno, *dnsZpm* vraća *Odgovor F* s traženom IP-adresom računalu *pc*. Navedeni primjer opisao je proces dohvaćanja adrese računala u slučaju da su sva priručna spremišta DNS-poslužitelja prazna. Ukoliko bi postojao zapis o traženoj IP-adresi u bilo kojem od DNS-poslužitelja na putu, ona bi se vratila u odgovoru, umjesto podatka o sljedećem poslužitelju kojeg „treba pitati“.



```

▶ Frame 11: 157 bytes on wire (1256 bits), 157 bytes captured (1256 bits) on interface 0
▶ Ethernet II, Src: 42:00:aa:00:00:15 (42:00:aa:00:00:15), Dst: 42:00:aa:00:00:1b (42:00:aa:00:00:1b)
▶ Internet Protocol Version 4, Src: 3.0.0.2 (3.0.0.2), Dst: 30.0.0.2 (30.0.0.2)
▶ User Datagram Protocol, Src Port: 53 (53), Dst Port: 56349 (56349)
▼ Domain Name System (response)
    [Request In: 10]
    [Time: 0.019621000 seconds]
    Transaction ID: 0xd0e2
    ▶ Flags: 0x8000 Standard query response, No error
    Questions: 1
    Answer RRs: 0
    Authority RRs: 2
    Additional RRs: 3
    ▼ Queries
        ▶ mm.tel.fer.hr: type A, class IN
    ▼ Authoritative nameservers
        ▶ hr: type NS, class IN, ns hr2.com
        ▶ hr: type NS, class IN, ns dnsHr.hr
    ▼ Additional records
        ▶ hr2.com: type A, class IN, addr 4.0.0.2
        ▶ dnsHr.hr: type A, class IN, addr 7.0.0.2
        ▶ <Root>: type OPT

```

Slika 5.7: Sadržaj poruke Odgovor B

S obzirom na način na koji vraćaju odgovore, DNS-poslužitelji možemo svrstati u dvije kategorije. U prvu kategoriju spada DNS-poslužitelj *dnsZpm*, koji, nakon što primi upit za koji ne zna odgovor, šalje dodatne upite u mrežu ostalim DNS-poslužiteljima. To znači da će kao odgovor na originalni upit uvijek vratiti ili traženi podatak (nakon što ga sazna), ili poruku o pogrešci (npr., ako traženo računalo ne postoji). Za takav DNS-poslužitelj kažemo da rekurzivno razlučuje adrese.

U drugu kategoriju spadaju svi ostali DNS-poslužitelji iz opisanog scenarija. Oni na traženi upit uvijek odmah vraćaju odgovor bez generiranja daljnjih upita. Odgovor može biti traženi odgovor (ako je poznat) ili adresa sljedećeg DNS-poslužitelja kojeg treba pitati. Za takve DNS-poslužitelje kažemo da iterativno razlučuju adrese.

Transportni protokol koji se koristi za slanje DNS-upita i DNS-odgovora je UDP. S obzirom da se radi o kratkim upitima i odgovorima koji moraju biti što brže preneseni kroz mrežu, praktičnije je koristiti UDP za koji ne mora biti uspostavljena i raskinuta veza prilikom prijenosa svakog upita/odgovora. Naravno, upotreba protokola TCP za tu svrhu nije zabranjena, ali se ne preporuča i nije česta u praksi.

### 5.2.3 Protokoli elektroničke pošte

Elektronička pošta je sustav koji omogućuje primanje i slanje elektroničkih poruka. Za primanje poruka koriste se protokoli POP i IMAP, a za njihovo slanje protokol SMTP. U zadacima koji slijede ukratko će se proučiti protokoli SMTP i POP.

Prije nego što se započne s proučavanjem protokola za slanje (SMTP) odnosno primanje (POP) elektroničke pošte u IMUNES-u, potrebno je na odgovarajućim računalima konfigurirati SMTP- i POP-poslužitelje. To se obavlja u sljedećim koracima:

1. Pokrenuti primjer `imunes-examples/DNS+Mail+WEB/NETWORK.imn`
2. U konzoli (*Applications Menu* -> *Terminal Emulator*) računala na kojem je pokrenut emulator/simulator IMUNES, potrebno je pozicionirati se u direktorij *DNS+Mail+WEB*:

```
# cd imunes-examples/DNS+Mail+WEB
```

3. Pokrenuti DNS-poslužitelje.
4. Iz direktorija *DNS+Mail+WEB* pokrenuti SMTP- i POP-poslužitelje naredbom:

```
# ./start_mail
```

Protokol SMTP (*Simple Mail Transfer Protocol*) definira postupak razmjene poruka elektroničke pošte između dva udaljena računala. Svi podaci se, u toku prijenosa elektroničke pošte s jednog na drugo računalo, prenose u ASCII-obliku, a sastoje se od niza naredbi te samog sadržaja poruke.

#### 5.2.4 Protokol HTTP

HTTP (*HyperText Transfer Protocol*) je aplikacijski protokol koji definira uslugu prijenosa web-sadržaja između računala. Izvorno je HTTP bio namijenjen za prijenos tzv. hiperteksta, ali danas ima gotovo univerzalnu primjenu – koristi se za prijenos datoteka (umjesto protokola FTP), kod usluga prilagođenih webu (npr., *web-mail*) i sl.

Topologija mreže podešena za analizu protokola HTTP nalazi se u datoteci *imunes-examples/DNS+Mail+WEB/NETWORK.imn*. Za ispravno generiranje HTTP-prometa potrebno je na odgovarajućim računalima pokrenuti DNS-poslužitelje i HTTP-poslužitelj. To se radi u sljedećim koracima:

1. Pokrenuti simulaciju nakon učitavanja mreže
2. U konzoli (*Applications Menu -> Terminal Emulator*) računala na kojem je pokrenut IMUNES, pozicionirati se u direktorij *DNS+Mail+WEB*:

```
# cd /root/imunes-examples/DNS+Mail+WEB/
```

3. Konfigurirati poslužitelje naredbama:

```
# ./start_dns
```

```
# ./start_http
```

Webu se pristupa s računala *pc*, te je na njemu potrebno uključiti snimanje prometa alatom *Wireshark*. Nakon toga na računalo *pc* treba kliknuti desnom tipkom miša i u padajućem izborniku odabrati opciju *Web Browser* čime se pokreće preglednik weba *Firefox*. Uvidom u promet snimljen alatom *Wireshark* mogu se primijetiti DNS-upiti i DNS-odgovori te protokoli TCP i HTTP. Odmah po pokretanju programa *Firefox* generira se veliki broj DNS-upita i odgovora koji služe podešavanju programa *Firefox* te ih se može zanemariti.

Početna stranica web-sjedišta na poslužitelju *www.tel.fer.hr* otvara se tako da se u polje za unos URI-a unese *http://www.tel.fer.hr* i pritisne Enter.

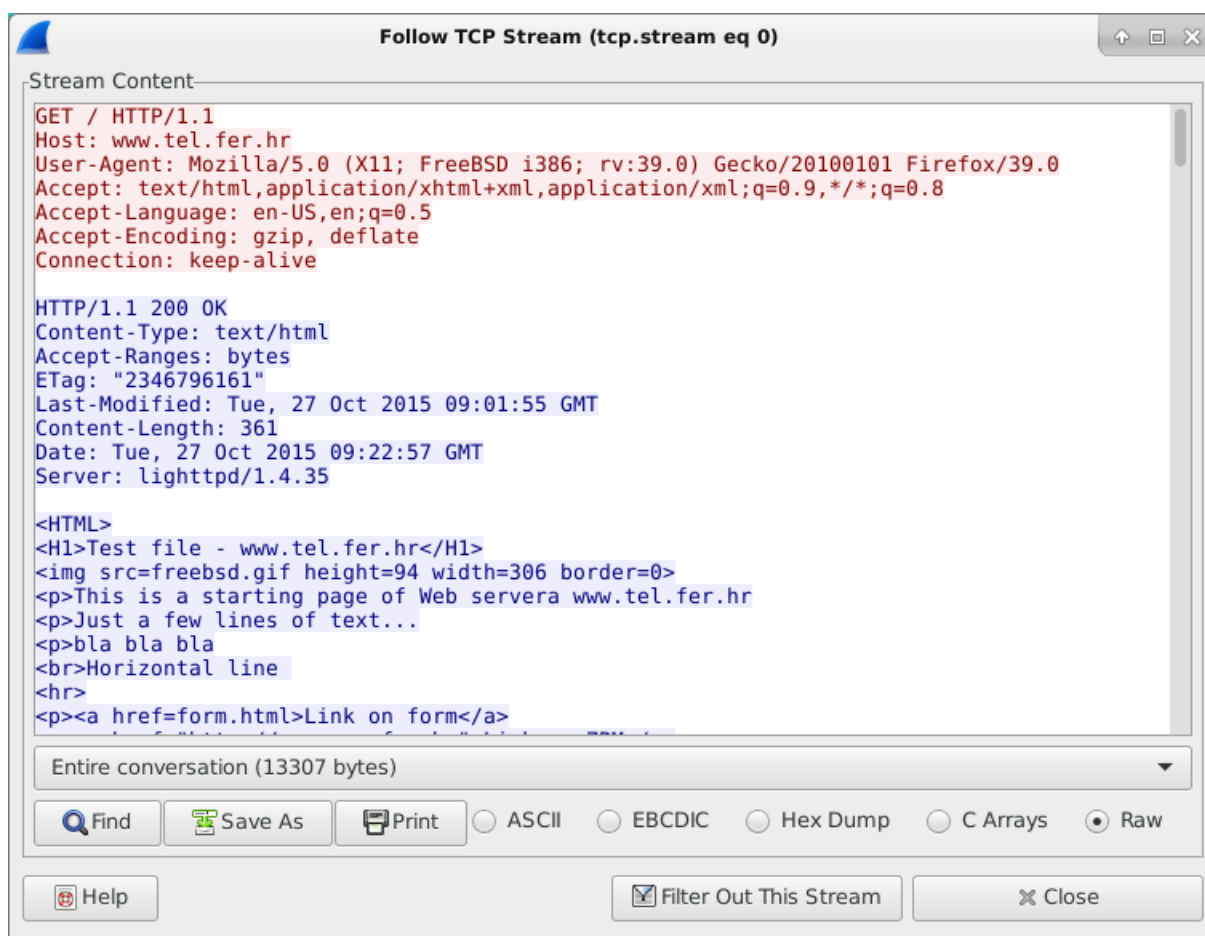
Uvidom u promet snimljen alatom *Wireshark*, može se vidjeti DNS-upit koji šalje računalo *pc* svom nadležnom DNS-poslužitelju *dnsZpm* kako bi saznalo IP-adresu računala *www.tel.fer.hr*, s obzirom da se u polje za unos URI-a upisalo FQDN (*Full Qualified Domain Name*) web-poslužitelja. Računalo *dnsZpm* potom vraća traženu IP-adresu računalu *pc*. Znanje o IP-adresi web-poslužitelja nužno je kako bi se HTTP-poruke uspješno usmjerile od izvorišta do odredišta korištenjem mrežnog protokola IP.

Transportni protokol TCP koristi se kao spojna usluga protokolu HTTP. Pritom se na računalu *pc* koriste proizvoljna TCP-vrata, dok se na web-poslužitelju koriste dobro znana vrata 80. Nakon uspostave TCP-sjednice između klijenta i poslužitelja, klijent može započeti sa slanjem HTTP-zahtjeva. Označi li se prva TCP-poruka koja se pojavljuje u ispisu, a zatim pritisnite desna tipka miša te odabere opcija *Follow TCP stream*, dobije se ispis kao na slici (Slika 5.8) koji omogućava lakše praćenje svih poruka protokola HTTP koje su se izmijenile u toj TCP-konekciji.

Prva poruka koja se javlja je poruka GET, koju šalje računalu *pc*. Tom porukom traži se dohvat resursa s web-poslužitelja. U ispisu su vidljiva dva karakteristična dijela poruke protokola HTTP:

- prvi redak - sadrži metodu zahtjeva, lokaciju traženog resursa (u ovom slučaju lokacija je '/' jer se traži početna stranica web-sjedišta) te verziju protokola;
- zaglavlje – sadrži ime poslužitelja na kojem se nalazi traženi resurs (*www.tel.fer.hr*), ime korištenog web-preglednika te ostale parametre.

Kao odgovor web-poslužitelja (tj., računala *www.tel.fer.hr*) na zahtjev GET šalje se odgovor 200 OK. Tim odgovorom signalizira se računalu *pc* da je traženi resurs dostupan te se on istovremeno i dostavlja. U zaglavlju odgovora nalaze se dodatni podaci o resursu, poput vremena zadnje promjene, duljine resursa, itd. Također je vidljiv treći karakterističan dio poruke protokola HTTP – tijelo poruke. U tijelu poruke nalazi se traženi resurs: u ovom slučaju radi se o dokumentu pisanom jezikom HTML.



Slika 5.8: Poruke protokola HTTP izmijenjene u jednoj TCP-konekciji

### 5.3 Eksperimenti i zadaci

**Zadatak 34.** Proučite primjer *DHCP/DHCP.imn* (Slika 5.1). Svrha ovog primjera je pokazati kako se računalima dinamički dodjeljuje IP-adresa. Scenarij vježbe je sljedeći:

1. Započnite simulaciju.
2. Kroz konzolu (*Applications Menu* -> *Terminal Emulator*) **računala na kojem je pokrenut IMUNES** (dakle, konzolu operacijskog sustava FreeBSD) pozicionirajte se u direktorij */root/imunes-examples/DHCP/* te izvršite skriptu *start\_dhcp*:

```
# ./start_dhcp
```

Skripta podešava odgovarajuće klijente i poslužitelje za ovaj primjer.

3. Otvorite konzolu na računalu *pc3* te pokrenite alat *Wireshark* tako da snima mrežni promet na mrežnom sučelju računala.
4. U konzoli na računalu *pc3* izvršite naredbu:

```
# dhclient eth0
```

Ovom naredbom se za mrežno sučelje *eth0* računala *pc3* zahtijeva od DHCP-poslužitelja dodjela IP-adrese.

5. Provjerite je li računalu *pc3* dodijeljena IP-adresa (naredba *ifconfig eth0*).
6. Zaustavite snimanje mrežnog prometa te, pomoću prikaza snimljenog u alatu *Wireshark*, proučite proces dobivanja IP-adrese od DHCP-poslužitelja i identificirajte pripadajuće DHCP-poruke. Skicirajte tijekom razmjene DHCP-poruka, uz navođenje pripadajućih izvorišnih i odredišnih MAC-adresa i IP-adresa.

**Zadatak 35.** (*Topologija DNS+Mail+Web/Network.imn*) Koristeći naredbu *host* u konzoli računala *pc.zpm.fer.hr* saznajte:

- a. IP-adresu računala *dnsHr.hr*,
- b. koje računalo je nadležno za primanje pošte u domeni *zpm.fer.hr*,
- c. koje računalo je nadležno za primanje pošte u domeni *tel.fer.hr*,
- d. koji su DNS-poslužitelji nadležni za domenu *hr*,
- e. koji su DNS-poslužitelji nadležni za domenu *fer.hr*,
- f. koji su DNS-poslužitelji nadležni za domenu *tel.fer.hr*,
- g. koji su DNS-poslužitelji nadležni za vršnu domenu („.“) i
- h. ime računala s IP-adresom 20.0.0.4.

**Zadatak 36.** (*Topologija DNS+Mail+Web/Network.imn*) Kakav se mrežni promet generira prilikom izvršavanja prethodnih naredbi? Skicirajte i obrazložite slijed DNS-upita i DNS-odgovora za zadatak 35.c. Izmjenjuju li se DNS-poruke s vršnim DNS-poslužiteljem? Objasnite.

**Zadatak 37.** (*Topologija DNS+Mail+Web/Network.imn*) Analizirajte korištena UDP-vrata za scenarij iz zadatka 35.g.

**Zadatak 38.** (Topologija DNS+Mail+Web/Network.imn) Analizirajte rad protokola SMTP kroz sljedeći scenarij:

1. Pokrenite snimanje mrežnog prometa na mrežnim sučeljima *eth0* računala *mm* i *www*.
2. Na računalu *mm* konfigurirajte klijentsku aplikaciju elektroničke pošte. Desnim klikom miša nad računalom *mm* odaberite opciju *Mail client*, čime se automatski pokreće klijentska aplikacija *Sylpheed*. Potom, u dijalogu *Mailbox setting* odaberite opciju *Create mailbox at the following default location* i pritisnite tipku *OK*. U sljedećem koraku odaberite opciju *POP3* i pritisnite tipku *Forward*. Nakon toga, u polje *Display name* upišite svoje ime, a u polje *E-mail address* adresu *root@tel.fer.hr*, te pritisnite tipku *Forward*. Nakon toga, u polje *User ID* upišite *root*, te u polja *POP3 server* i *SMTP server* upišite adresu pretpostavljenog mail poslužitelja (u ovom slučaju to je poslužitelj *www.tel.fer.hr*). Ostale postavke ostavite nepromijenjene te pritisnite tipku *Forward*, a potom i *Close*.
3. U grafičkom sučelju aplikacije *Sylpheed* sastavite novu poruku elektroničke pošte proizvoljnog sadržaja (opcija *Compose*), te poruku pošaljite na adresu *imunes@zpm.fer.hr*.
4. Zaustavite snimanje mrežnog prometa, te pomoću prikaza snimljenog u alatu *Wireshark* odgovorite na sljedeća pitanja:
  - a. Koji su se protokoli pojavili kao rezultat slanja elektroničke poruke? Koja je njihova osnovna uloga? Navedite kojem sloju TCP/IP-modela pripada svaki od tih protokola.
  - b. Koji su koraci prilikom slanja elektroničke poruke s računala *mm* na adresu *imunes@zpm.fer.hr*? Obratite pozornost na primjenu sustava DNS i protokola SMTP.
  - c. Gdje se sve koristi protokol DNS u ovom slučaju?  
Koje računalo je nadležni poslužitelj elektroničke pošte za domenu *tel.fer.hr*? Kako računalo *mm* saznaje IP-adresu tog poslužitelja?  
Koje računalo je nadležni poslužitelj elektroničke pošte za domenu *zpm.fer.hr*? Kako računalo *www* saznaje IP-adresu tog poslužitelja?
  - d. Koliko se TCP-konekcija uspostavlja u ovom primjeru i koja se vrata pritom koriste? Odaberite opciju *Statistics* → *Conversations* → *TCP* u izornoj traci alata *Wireshark*. Čemu služe te konekcije?
  - e. Pronađite TCP-segment kojim započinje uspostava konekcije s računala *mm* prema njegovom nadležnom SMTP-poslužitelju. Označite ga, zatim pritisnite opciju *Follow Stream*. Kako teče komunikacija protokolom SMTP između tog računala i njegovog nadležnog poslužitelja? Skicirajte dobiveni tijek poruka. Pronađite segment u kojem se prenosi sadržaj elektroničke poruke.

Protokol POP (*Post Office Protocol*) definira postupak pristupa elektroničkoj pošti koja je pohranjena u poštanskom sandučiću na POP-poslužitelju. Svi podaci se prenose u ASCII-obliku, a sastoje se od niza naredbi te samog sadržaja poruke.

**Zadatak 39.** (Topologija DNS+Mail+Web/Network.imn) Analizirajte rad protokola POP kroz sljedeći scenarij:

1. Pokrenite snimanje mrežnog prometa na mrežnom sučelju *eth0* računala *pc*.
2. Na računalu *pc* konfigurirajte klijentsku aplikaciju elektroničke pošte. Desnim klikom miša nad računalom *pc* odaberite opciju *Mail client*, čime se automatski pokreće klijentska aplikacija *Sylpheed*. Potom, u dijalogu *Mailbox setting* odaberite opciju *Create mailbox at the following default location* i pritisnite tipku *OK*. U sljedećem koraku odaberite opciju *POP3* i pritisnite tipku *Forward*. Nakon toga, u polje *Display*

*name* upišite ime svog asistenta, a u polje *E-mail address* adresu *imunes@zpm.fer.hr*, te pritisnite tipku *Forward*. Nakon toga, u polje *User ID* upišite *imunes*, te u polja POP3 server i SMTP server upišite adresu pretpostavljenog mail poslužitelja (u ovom slučaju to je poslužitelj *zpmMail.zpm.fer.hr*). Ostale postavke ostavite nepromijenjene te pritisnite tipku *Forward*, a potom i *Close*.

3. U grafičkom sučelju aplikacije *Sylpheed* pristupite POP-poslužitelju na računalu *zpmMail* kako bi pročitali pristiglu poruku. To napravite tako da pritisnete opciju *Get all* u grafičkom sučelju, u ponuđenom izborniku upišete lozinku *imunes*, te pritisnete tipku *OK*. Otvorite pristiglu poruku.
4. Zaustavite snimanje mrežnog prometa te pomoću prikaza snimljenog u alatu *Wireshark* odgovorite na sljedeća pitanja:
  - a. Koji su se protokoli pojavili kao rezultat pristupanja elektroničkoj pošti na POP-poslužitelju računala *zpmMail*? Navedite kojem sloju TCP/IP-modela pripada svaki od tih protokola.
  - b. Koliko se TCP-konekcija uspostavlja u ovom primjeru i koja se vrata pritom koriste?
  - c. Za što se koristi protokol DNS u ovom slučaju?
  - d. Pronađite TCP-segment kojim započinje uspostava konekcije s računala *pc* prema njegovom nadležnom POP-poslužitelju. Označite ga i odaberite opciju *Follow Stream*. Kako teče razmjena POP-poruka između računala *pc* i njegovog nadležnog poslužitelja? Skicirajte dobiveni tijek poruka. Čemu služe POP-poruke *LIST* i *RETR*? Analizirajte njihove odgovore.
  - e. Pogledajte sadržaj elektroničke poruke i analizirajte polja *Received*. Kojim „putem“ je poruka stigla na poslužitelj *zpmMail*?
  - f. Je li komunikacija između ovih računala šifrirana? Analizirajte slanje segmenata koji se odnose na prijenos lozinke.

**Zadatak 40.** (Topologija *DNS+Mail+Web/Network.imn*) Otvorite početnu stranicu web-poslužitelja *www.zpm.fer.hr* (*zpmMail*) korištenjem URI-a *http://www.zpm.fer.hr*. Koliko se HTTP-konekcija uspostavi u ovom primjeru? Čemu služe te konekcije? Odredite transportne adrese za svaku od tih konekcija.

**Zadatak 41.** (Topologija *DNS+Mail+Web/Network.imn*) Proizvoljno odaberite jedan HTTP-zahtjev i jedan HTTP-odgovor, skicirajte ih te utvrdite njihove karakteristične dijelove.

**Zadatak 42.** (Topologija *DNS+Mail+Web/Network.imn*) U programu *Firefox* pokrenutom na računalu *pc* pristupite URI-ju *http://www.tel.fer.hr* te odaberite poveznicu „*Link on ZPM*“. Kada se učita nova stranica, odaberite poveznicu „*Link on ZZT*“ čime ćete se vratiti na stranicu Zavoda za telekomunikacije (*http://www.tel.fer.hr*). Proučite poruke protokola HTTP koje će se pojaviti u alatu *Wireshark* nakon odabira poveznice „*Link on ZZT*“, te skicirajte njihov tijek. Koja je uloga parametra *If-Modified-Since* u zaglavlju prvog HTTP-zahtjeva koji se pojavio nakon odabira poveznice „*Link on ZZT*“? Zašto je tijelo poruke prvog HTTP-odgovora koji se pojavio nakon odabira iste poveznice prazno?

## 5.4 Pitanja

**Pitanje 16.** Koja je uloga DNS-a?

- a. pridruživanje numeričke IP adrese na osnovu poznate MAC adrese.
- b. pridruživanje MAC adrese na osnovu poznate IP adrese.
- c. pridruživanje simboličke adrese na osnovu poznate numeričke IP adrese.
- d. pridruživanje simboličke adrese na osnovu poznate MAC adrese.

**Pitanje 17.** Koja od navedenih tvrdnji je istinita?

- a. u javnom Internetu može postojati više računala s istom IP adresom, ali u tom slučaju moraju imati različite simboličke adrese.
- b. u javnom Internetu svako računalo ima jedinstvenu IP adresu, ali simboličke adrese ne moraju biti jedinstvene.
- c. u javnom Internetu može postojati više računala s istom IP adresom.
- d. u javnom internetu ne mogu postojati dvije identične simboličke adrese

**Pitanje 18.** Na koji transportni protokol se oslanjaju protokoli za čitanje pošte?

- a. ICMP.
- b. IP.
- c. UDP.
- d. TCP.

**Pitanje 19.** Prilikom slanja elektroničke pošte putem protokola SMTP, poslužitelj na zahtjeve klijenta odgovara:

- a. DATA zahtjevima.
- b. ACK zahtjevima.
- c. SUCCESS ili FAILURE zahtjevima.
- d. statusnim kodom.

**Pitanje 20.** Shema URI-ja:

- a. identificira virtualni poslužitelj.
- b. može određivati protokol koji se koristi.
- c. određuje aplikaciju kojom se pristupa resursu.
- d. određuje pojedini resurs na poslužitelju.

## 5.5 Izvori

- IETF RFC 2131 „Dynamic Host Configuration Protocol“: <http://tools.ietf.org/html/rfc2131>
- IETF RFC 1034 „Domain Names – Concepts and Facilities“: <http://tools.ietf.org/html/rfc1034>
- IETF RFC 1035 „Domain Names – Implementation and Specification“: <http://tools.ietf.org/html/rfc1035>
- IETF RFC 5321 „Simple Mail Transfer Protocol“: <http://tools.ietf.org/html/rfc5321>
- IETF RFC 1939 „Post Office Protocol – Version 3“: <http://tools.ietf.org/html/rfc1939>
- IETF RFC 3501 „Internet Message Access Protocol – Version 4rev1“: <http://tools.ietf.org/html/rfc3501>
- IETF RFC 2616 „Hypertext Transfer Protocol – HTTP/1.1“: <http://tools.ietf.org/html/rfc2616>

## **6      Odgovori na pitanja**

1-b, 2-a, 3-a, 4-c, 5-a, 6-a, 7-a, 8-b, 9-b, 10-b, 11-c, 12-c, 13-b, 14-d, 15-d, 16-c, 17-d, 18-d, 19-d, 20-b.



## 7 Indeks pojmova

\*\*\* Indeks pojmova će biti generiran nakon finalnog prijeloma teksta. \*\*\*

## Prilog A: naredbe UNIX-ljuske

Postoji niz naredbi koje su standardno dostupne na operacijskom sustavu UNIX i njemu srodnim operacijskim sustavima (Linux, Solaris, FreeBSD). Tablica „Popis UNIX-naredbi korištenih u ovom udžbeniku“ sadrži popis naredbi korištenih u ovom udžbeniku uz kratka objašnjenja i primjere korištenja. Puno više detalja o naredbama može se pronaći na *man*-stranici pojedine naredbe, izvršavanjem:

```
# man <naziv naredbe>
```

Tablica: Popis UNIX-naredbi korištenih u ovom udžbeniku

Naziv	Objašnjenje	Primjer korištenja	
		Naredba	Rezultat
pwd	Ispisuje trenutno aktivni direktorij	# pwd	Ispis: /root/imagenes-examples
cd	Pozicionira se u zadani direktorij	# cd /root	Pozicioniranje u direktorij /root.
ls	Ispis sadržaja direktorija	# ls	Ispis svih datoteka i poddirektorija u trenutnom direktoriju.
cp	Kopira datoteke	# cp f.txt /root/	Kopira datoteku f.txt iz trenutnog direktorija u direktorij /root/
touch	Mijenja zapis o zadnjem vremenu pregleda i izmjene datoteke (u slučaju nepostojanja datoteke kreira praznu datoteku sa zadanim imenom).	# touch file.txt	Ako u trenutnom direktoriju postoji datoteka file.txt, postavlja vrijeme zadnjeg pristupa i izmjene na trenutno ili zadano vrijeme; ako ne postoji, kreira praznu datoteku file.txt.
mkdir	Kreira direktorij	# mkdir noviDir	Kreira direktorij noviDir u trenutnom direktoriju.
chmod	Mijenja prava pristupa datoteci	# chmod +x file.sh	Daje datoteci file.sh dozvolu za izvršavanje.
chown	Mijenja vlasništvo nad datotekom.	# chown root file.txt	Korisnik root postaje vlasnik datoteke file.txt.
tar	Kreira i raspakirava arhive	# tar cf archive.tar f1.txt f2.txt	Kreira arhivu archive.tar s datotekama f1.txt i f2.txt.
echo	Ispisuje tekst.	# echo 'Lab. vježba' > dat.txt	Ispisuje tekst „Lab. vježba“ u datoteku dat.txt.