Analiza i usporedba rada mrežnih emulatora

Fabrični, Kristian

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:119:617455

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**



Repository / Repozitorij:

Faculty of Transport and Traffic Sciences - Institutional Repository







SVEUČILIŠTE U ZAGREBU

FAKULTET PROMETNIH ZNANOSTI

Kristian Fabrični

ANALIZA I USPOREDBA RADA MREŽNIH EMULATORA

ZAVRŠNI RAD

Zagreb, 2017.

Sveučilište u Zagrebu
Fakultet prometnih znanosti

ZAVRŠNI RAD

ANALIZA I USPOREDBA RADA MREŽNIH EMULATORA

PERFORMANCE ANALYSIS AND COMPARISON OF NETWORK EMULATORS

Mentor: doc. dr. sc. Ivan Grgurević

Student: Kristian Fabrični

JMBAG: 0135231350

Zagreb, rujan 2017.

ANALIZA I USPOREDBA RADA MREŽNIH EMULATORA

SAŽETAK

Pojam emulacija predstavlja tehniku eksperimentiranja čija je glavna funkcija oponašanje funkcioniranja nekog sustava i njegovih komponenti u stvarnom vremenu i kontroliranom okruženju, a sustav na koji se najčešće primjenjuje eksperiment emulacije jesu računalne mreže. Softversko/hardverske komponente koje omogućuju eksperiment emulacije nazivaju se emulatori. U teorijskom dijelu rada prikazane su i opisane osnovne karakteristike i arhitektura računalnih mreža te funkcije, mogućnosti i primjena mrežnih emulatora u funkciji računalnih mreža. U praktičnom dijelu rada izvršen je eksperiment emulacije na tri relevantna mrežna emulatora IMUNES, CORE i GNS3 te su analizirani i uspoređeni načini rada istih.

KLJUČNE RIJEČI**:** računalne mreže; mrežni emulatori; Integrated Multiprotocol Network Emulator/Simulator; Common Open Research Emulator; Graphical Network Simulator 3

SUMMARY

The term emulation is a experiment technique whose main function is to imitate the functioning of a system and its components in a real-time and controlled environment and the system most frequently used by the emulation experiment is a computer network. Software / hardware components that allow emulation experiment are called emulators. Theoretical part presents the basic characteristics and architecture of computer networks as well as the functions, possibilities and application of network emulators. In the practical part, three relevant emulators called IMUNES, CORE and GNS3 were analyzed and compared and emulation experiment was performed by them.

KEYWORDS**:** computer network; network emulator; Integrated Multiprotocol Network Emulator/Simulator; Common Open Research Emulator; Graphical Network Simulator 3

SADRŽAJ

1. UVOD 1

2. KARAKTERISTIKE RAČUNALNIH MREŽA 2

2.1. KLASIFIKACIJA MREŽA PREMA VELIČINI 2
2.2. KLASIFIKACIJA MREŽA PREMA NAMJENI 3
2.3. KLASIFIKACIJA MREŽA PREMA NAČINU OSTVARIVANJA VEZE 3
2.4. KLASIFIKACIJA MREŽA PREMA SKLOPOVSKOJ TEHNOLOGIJI KOJU KORISTE 4
2.5. KLASIFIKACIJA MREŽA PREMA FUNKCIONALNIM POVEZANOSTIMA IZMEĐU POJEDINIH ELEMENATA MREŽE 5

3. ARHITEKTURA RAČUNALNIH MREŽA 7

3.1. ELEMENTI I KOMPONENTE RAČUNALNIH MREŽA 7
3.1.1. AKTIVNA MREŽNA OPREMA 7
3.1.2. PASIVNA MREŽNA OPREMA 9
3.2. TOPOLOGIJE RAČUNALNIH MREŽA 10
3.3. REFERENTNI MODELI RAČUNALNIH MREŽA 13
3.3.1. OSI REFERENTNI MODEL 13
3.3.2. TCP/IP REFERENTNI MODEL 14

4. FUNKCIJE I PRIMJENA MREŽNIH EMULATORA 16

4.1. EKSPERIMENTALNE TEHNIKE 16
4.2. KLASIFIKACIJA MREŽNIH EMULATORA 18
4.2.1. KLASIFIKACIJA PREMA DOSTUPNOSTI MREŽNIH EMULATORA 19
4.2.2. KLASIFIKACIJA PREMA NAČINU IMPLEMENTACIJE I FUNKCIONALNOSTI 19
4.2.3. KLASIFIKACIJA PREMA RAZINI NA KOJOJ SE MREŽA EMULIRA 20
4.2.4. KLASIFIKACIJA PREMA SLOŽENOSTI KORIŠTENOG MREŽNOG MODELA 21

5. ANALIZA MREŽNIH EMULATORA U FUNKCIJI RAČUNALNIH MREŽA 23

5.1. IMUNES 23
5.2. CORE 26
5.3. GNS3 28

6. USPOREDBA RADA MREŽNIH EMULATORA 30

6.1. EKSPERIMENT EMULACIJE UNUTAR IMUNES EMULATORA 30
6.2. EKSPERIMENT EMULACIJE UNUTAR CORE EMULATORA 35
6.3. EKSPERIMENT EMULACIJE UNUTAR GN3 EMULATORA 40

7. ZAKLJUČAK 49

LITERATURA 50

POPIS KRATICA I AKRONIMA 53

POPIS SLIKA 55

POPIS TABLICA 57

1. Uvod

Razvijanjem računalnih sustava pojavila se potreba za povezivanjem istih stoga je razvijen sustav koji će međusobno povezati sva računala i tako kreirati računalnu mrežu pomoću koje će se odvijati proces prijenosa podataka. Računalna mreža predstavlja skup veza koja povezuje sva računala i pomoću nje se odvija prijenos podataka. Postoje nekoliko vrsta računalnih mreža i njihova podjela zavisi od njihove građe, veličine itd. Od vrste mreže, načina prijenosa i ostalih bitnih čimbenika zavisi kako će se brzo i kvalitetno prenositi podaci.

Arhitektura računalne mreže je kompleksna i složena te je potrebno dosta vremena i rada kako bi se moglo kvalitetno izraditi plan iste. Gledano s ekonomske strane, implementacija kompleksne računalne mreže je skup proces u kojem nema mjesta za pogreške. Iz tog razloga razvijene su softversko/hardverske komponente koje omogućuju oponašanje rada računalnih mreža, a te komponente nazivamo mrežni emulatori i simulatori. Cilj završnog rada je analizirati i usporediti rad relevantnih mrežnih emulatora u funkciji računalnih mreža, a svrha završnog rada je prikazati primjenu mrežnih emulatora s obzirom na različitu namjenu, karakteristike i vrste računalnih mreža. Naslov završnog rada je: Analiza i usporedba rada mrežnih emulatora. Rad je podijeljen u sedam cjelina:

1. Uvod
2. Karakteristike računalnih mreža
3. Arhitektura računalnih mreža
4. Funkcije i primjena mrežnih emulatora
5. Analiza mrežnih emulatora u funkciji računalnih mreža
6. Usporedba rada mrežnih emulatora
7. Zaključak

Uvodno poglavlje daje osnovnu sliku o radu te definira svrhu, cilj i strukturu rada. U drugom poglavlju opisane su osnovne značajke računalnih mreža, koja je njihova svrha te klasifikacija istih. Treće poglavlje sadrži opisanu arhitekturu računalnih mreža te njezine glavne elemente i komponente. Prikazane su mrežne topologije kao i podjela mreže na slojeve. Četvrto poglavlje ukratko opisuje eksperimentalne tehnike, koje su namijenjene istraživanju rada računalnih mreža, a kasnije i sam proces emulacije. Objašnjene su i definirane osnovne funkcije i primjena mrežnih emulatora kao i klasifikacija istih. U petom poglavlju su analizirane te opisane funkcije i mogućnosti emulatora koji će se koristiti u ovom radu. Šesto poglavlje obuhvaća praktični rad te analizu dobivenih rezultata i na kraju usporedbu mrežnih emulatora IMUNES, CORE te GNS3. U zaključnom poglavlju napravljena je sinteza svega učinjenog tijekom izrade završnog rada.

2. Karakteristike računalnih mreža

Računalna mreža je veza dva ili više računala putem medija za razmjenu informacija ili podatkovne komunikacije. Također gleda se kao sustav komunikacija koji pruža usluge aplikacijama koje koriste na način da pozivaju funkcije operacijskih sustava. Krajnja svrha računalnih mreža je omogućiti krajnjim korisnicima: [1]

- Dijeljenje datoteka;
- Dijeljenje hardverskih resursa, poput skenera i pisača;
- Korištenje brojnih usluga poput e-pošte, videokonferencije, istovremenih poruka, *World Wide Web-a,* društvenih mreža itd.;
- Lakši pristup i održavanje informacija među umreženim korisnicima;
- Centralizaciju smještaja podataka [2] i
- Distribuiranje obrade podataka na više računala. [3]

S obzirom na to da postoji veliki broj različitih vrsta mreža, potrebno je klasificirati iste. A klasificirane su prema:

- Veličini (Osobna, lokalna, gradska računalna mreža te mreža širokog područja);
- Namjeni (Privatne i javne);
- Načinu ostvarivanja veze (Dijeljena veza i „od točke do točke“ (engl. *Point to Point));*
- Sklopovskoj tehnologiji koju koriste (Žične (Optičke mreže, *ethernet*) i bežične (engl. *wireless*)); [3]
- Funkcionalnim povezanostima između pojedinih elemenata mreže (*Active networking*, klijent – poslužitelj i „od čvora do čvora“ (engl. *Peer to peer* – P2P). [3]

2.1. Klasifikacija mreža prema veličini

Osobna računalna mreža (engl. *Personal Area Network* - PAN) je mreža za povezivanje uređaja (npr. telefon) na računalo koji obično služe jednom korisniku. Prostire se najviše unutar nekoliko metara. [3]

Lokalna računalna mreža (engl. *Local Area Network* - LAN) je računalna mreža u kojoj su računala smještena na manjim udaljenostima (unutar doma, ureda, ili blisko smještenih zgrada). Možemo ju zvati i područna lokalna mreža, a jedna od bitnijih značajki je da su one najčešće u cijelosti u vlasništvu i pod upravljanjem onih koji ih koriste (osobno, vlasništvo tvrtke itd.), tako da je prijenos podataka unutar te mreže besplatan za korisnike. Osim toga, glavna karakteristika je jako velika brzina prijenosa podataka (Gbit/s - Gigabit po sekundi). Također „lokalnost“ mreže je ovdje određena ukupnom dužinom spojnih kablova mreže (ili dosegom elektromagnetskog signala kod bežičnih mreža). [4]

Gradska računalna mreža (engl. *Metropolitan Area Network* - MAN) je mreža gdje je područje pokrivenosti na razini jednog dijela ili cijelog grada (nekoliko desetaka kilometara) te ih možemo svrstati u područje između LAN i WAN mreže. Mogu biti u vlasništvu neke organizacije ili više njih, a brzine prijenosa su obično manje nego u lokalnim mrežama. Neke mreže toga tipa koriste specifične metode prijenosa podataka, koje se razlikuju od metoda prijenosa kod LAN i WAN mreža. [4]

Mreže širokog područja (engl. *Wide Area Network* – WAN) su mreže koje pokrivaju široko geografsko područje i znatno su složenije od lokalnih, a uključuju povezivanje čvorova u različitim gradovima, zemljama i kontinentima. Za razliku od prethodnih mreža, WAN mreža nije u vlasništvu osoba ili organizacija koje ih koriste i prijenos podataka preko njih je ograničen prema brzini, količini i cijeni, što znači da je potrebno platiti za korištenje komunikacijskih veza i određenih resursa.

Iako su brzine dosta ograničene, prednost kod tih mreža je da nema specifičnih ograničenja fizičkog prostora njihovog širenja što ujedno znači da su fizički neograničene veličine. Ali ono što stvara problem kod ograničenja jesu čvorovi (engl. nodes) tj. adrese čvorova. S obzirom na to da postoji ograničen broj adresa čvorova samim time postoji i određen broj računala koji mogu biti uključeni u mrežu. Ali takva vrsta problema se riješila povećanjem broja bitova koje tvore adresu i time je dobiven puno veći broj adresa (nova verzija Internet protokola[1] (engl. *Internet Protocol - IP*[2]) IPv6). Jedan od najboljih primjera WAN mreže je Internet.

2.2. Klasifikacija mreža prema namjeni

Privatne računalne mreže su najčešće mreže izgrađene od strane kompanija koje će kasnije koristiti za svoje potrebe. Obično su to mreže s lokalnim područjem pokrivenosti, ali mogu biti i na većem području.

Javne računalne mreže (engl. *Public Network*) su pak izgrađene od strane državnih tijela i agencija, a njihova izgradnja financira se iz državnih prihoda. Javnoj mreži može, uz određenu naknadu naravno, pristupiti bilo tko. A kako bi se usluge korištenja javne mreže mogle naplatiti, razvijen je sustav naknada za korištenje usluga računalne mreže koji naziva se mrežnom tarifom.

2.3. Klasifikacija mreža prema načinu ostvarivanja veze

U dijeljenu vezu (engl. *shared link*) pripadaju *broadcast, multicast i unicast* komunikacije. *Broadcast* način komunikacije je namijenjen slanju informacija i podataka od jednog pošiljatelja prema svim mogućim korisnicima/primateljima koji su

[1] **Protokoli** su pravila ili skupovi pravila koja su standardizirana koja omogućuju komunikaciju računala unutar neke mreže.

[2] **Internet protokol (engl. *Internet Protocol – IP*)** je mrežni protokol za prijenos podataka kojeg koriste izvorišna i odredišna računala za uspostavu podatkovne komunikacije preko računalne mreže. IP upravlja i nadzire adrese unutar paketa da bi paketi nesmetano stigli na odredište. [32]

u tom trenutku aktivni na mreži i koji su spojeni na jedan zajednički komunikacijski kanal. Takav način komunikacije u računalnim mrežama se koristi kada se šalju podatci na male udaljenosti. U slučaju da podatke želimo poslati samo određenoj skupini ljudi onda je potrebno koristiti adresiranje, a to se postiže kada korisnici imaju zajednički dio adrese. Ovakav način komunikacije nazivamo *multicast.*

Unicast i *Point to Point* komunikacija su usko povezani jer se radi o vezi između samo dva korisnika. Razlika je u tome što kod *Point to Point* komunikacije korisnici nisu direktno povezani jedan s drugim i adresirani podatci koji se šalju putuju „preko" više povezanih korisnika, a kod *unicast* komunikacije korisnici su direktno spojeni jedan s drugim. [5] Point to point je također jedna od topologije mreža, ali to će biti detaljnije objašnjeno u trećem poglavlju.

2.4. Klasifikacija mreža prema sklopovskoj tehnologiji koju koriste

Glavna karakteristika optičkih mreža je prijenos podataka korištenjem optičkog vlakna kao prijenosnog medija. Korištenjem optičkog vlakna omogućuju se velike brzine prijenosa podataka na jako velike udaljenosti uz male mogućnosti pogrešaka u prijenosu kao i slabog utjecaja vanjskih smetnji. [3]. Također je razvijena arhitektura pod nazivom *Fiber to the x.* Pod tim pojmom podrazumijeva se svaka mrežna arhitektura koja koristi optička vlakna u cilju zamjene klasične bakrene parice.

Razlikuju se četiri tehnologije optičkih mreža: [6]

- *Fiber to the node* (čvora) / *neighborhood* (susjedstva) (FTTN) / *Fiber to the cabinet* (ormara) (FTTCab);
- *Fiber to the curb* (pločnika) (FTTC);
- *Fiber to the building* (zgrade) (FTTB) i
- *Fiber to the home* (doma) (FTTH).

Ethernet predstavlja skup tehnologija za prijenos podataka pakiranjem podataka u okvire. To je protokol sloja veze u TCP/IP (engl. *Transmission Control Protocol / Internet Protocol)* modelu koji pokazuje način kako umreženi uređaji mogu pretvoriti podatke za prijenos na ostale mrežne uređaje na istom segmentu mreže. [7] Na OSI (engl. *Open System Interconnection Model*)[3] referentnom modelu usko povezuje fizički i podatkovni sloj. *Ethernet* na fizičkom sloju definira standarde za ožičenje i signalizaciju, te zajednički format adresiranja dok na podatkovnom sloju definira zajedničku adresu i način na koji se pristupa određenom mediju (koaksijalni kabel, bakrena parica te optičko vlakno) koji se koristi za prijenos podataka. Brzine prijenosa podataka su velike, a udaljenosti koje *Ethernet* može pokrivati su ograničene.

[3] **OSI referentni model (engl. *Open Systems Interconnection Basic Reference Model)*** je apstraktni, slojeviti model podijeljen u sedam slojeva koji služi kao preporuka stručnjacima za razvoj računalnih mreža i protokola. [30]

Bežične (engl. *wireless*) mreže su vrsta računalnih mreža koje povezuju dva ili više računala pomoću elektromagnetskih signala ili valova. Ovakvim načinom prijenosa podataka korisnicima je omogućena pokretljivost unutar dometa pristupne točke (engl. *access point*). [3]

S obzirom da je riječ o tehnologiji koja koristi određene frekvencije za prijenos, potrebno je definirati određene standarde i protokole kao i spektar frekvencija koji će se koristiti. IEEE (engl. *Institute of Electrical and Electronics Engineers*) je uveo set standarda pod nazivom IEEE 802.11 koji definiraju format i strukture signala za bežični prijenos. Daljnjim razvojem tehnologije korištenje ovakvog načina prijenosa raste eksponencijalno te bežične mreže postaju sve dostupnije i popularnije zbog jednostavnosti instalacije, pokretljivosti korisnika i velikoj zastupljenosti bežičnih mrežnih kartica u novim prijenosnim računalima. Za razliku od *Etherneta* brzine prijenosa podataka kod bežičnih mreža su relativno ograničene kao i udaljenosti.

2.5. Klasifikacija mreža prema funkcionalnim povezanostima između pojedinih elemenata mreže

Aktivno umrežavanje je komunikacijski model koji omogućuje paketima koji prolaze telekomunikacijskom mrežom dinamičku promjenu rada mreže. [8] Općenita arhitektura se sastoji od izvršnih okruženja koja se sastoje od čvorova operativnih sustava. Također sastoji se od aktivnog hardvera, koji može usmjeravati jednako i izvršavati kôd unutar aktivnih paketa. [9] Ovo znači da korisnici sami mogu „ubacivati“ određene programe i kodove koji se izvršavaju u čvorovima mreže, a samim time omogućena je prilagođena i brza promjena mrežnih operacija u stvarnom vremenu.

Klijent – poslužitelj (engl. *Client – server*) je naziv metode prema kojoj se odvijaju razni procesi u računalnim mrežama u kojoj su razdvojene uloge klijenta i poslužitelja. Klijent možemo promatrati kao sustav čija je uloga slanje nekog zahtjeva, za određenom uslugom, poslužitelju koji izvršava taj zahtjev i šalje nazad rezultat izvođenja zahtjeva. Ipak, ne mora biti tako. Klijent može zatražiti upis nekog sadržaja u neku bazu podataka i time poslati zahtjev, a poslužitelj će vratiti samo obavijest o izvršenju zahtjeva, a ne nužno i obavijest. [4] Svaki klijent i svaki server u toj arhitekturi predstavlja jednog člana ili čvor (engl. node) te mreže. [3] Primjeri metode klijent-poslužitelj su: sustav razmjene elektroničke pošte, sustav pristupa Internet stranicama, sustav prijenosa datoteka itd.

Kod *peer to peer* - P2P ustrojstva mreže, sva su priključena računala međusobno ravnopravna i nema mrežnog nadzornog računala, što znači da P2P mrežu možemo definirati kao mrežu istovrsnih čvorova (svaki čvor u isto vrijeme ima funkciju klijenta i poslužitelja). Obično su sva računala spojena na koncentrator (engl. hub) preko koga međusobno komuniciraju i razmjenjuju podatke.

P2P mreže se prema strukturi dijele na centralizirane i decentralizirane. Decentralizirane se dalje dijele na strukturirane i nestrukturirane.

Ono što karakterizira centralni P2P sustav je centralni poslužiteljski sustav koji služi da usmjerava promet između korisnika i da održava direktorije u kojima se nalaze dijeljene datoteke koje su spremljene na računalu korisnika. Svaki put kada se korisnik spoji ili odspoji s mreže popis datoteka se osvježava. Tako, svaki put kada korisnik zatraži neku datoteku, centralni poslužitelj uspoređuje zahtjev s popisom datoteka koje su dostupne kod trenutno spojenih korisnika i onda se ta datoteka, ako se nalazi na popisu, isporučuje korisniku.

Decentralizirani P2P sustav je nova generacija P2P mreža i njihova glavna karakteristika je da izbjegavaju strukturu središnjeg sustava poslužitelja. Imamo strukturirane i nestrukturirane sustave. Kod strukturiranih sustava se govori o sustavima s definiranom mrežnom topologijom, što znači da kod ovog sustava svaki čvor ima svoj jedinstveni identifikator „p“ koji se jednoznačno povezuje s ključem „k“. Mrežna topologija nestrukturiranih P2P sustava nema definiranu strukturu, a to znači da su podaci spremljeni na čvorovima koji ih kreiraju i da se pretraživanje sustava izvodi preplavljivanjem ili slučajnim izborom.

3. Arhitektura računalnih mreža

Ono što definira arhitekturu računalnih mreža jesu elementi i komponente, koji omogućuju prijenos podataka, slojevi, koji definiraju strukturu računalne mreže, protokoli, koji definiraju način prijenosa podataka te sam fizički način povezivanja tj. mrežna topologija. Mreže gradimo od kanala, čvorišta i terminala (elementi mreža). [2] Čvorišta imaju zadaću usmjeravanja podataka, a kanali moraju omogućiti prijenos istih do terminala koji predstavljaju uređaje povezane na mrežu.

Kako bi se informacije prenesle „s kraja na kraj“ mreže vrši se postupak komutiranja ili prospajanja. [2] Iako postoje dvije vrste komutacija (komutacija kanala i komutacija paketa) u računalnim mrežama koristimo samo komutaciju paketa. Komutacija paketa prenosi informacije tako da se poruke korisnika dijele na pakete koji se prenose kroz mrežu. [2] Daljnji prijenos paketa ovisi o procesima usmjeravanja te prosljeđivanja paketa u kojem paketi u svom zaglavlju nose globalnu adresu odredišta.

3.1. Elementi i komponente računalnih mreža

Postoje dvije vrste komunikacijske opreme kod računalnih mreža, a to su:

- Aktivna - sastoji se od elektroničkih uređaja koji primaju i šalju promet unutar mreže (preklopnik (engl. *switch*), usmjernik (engl. *router*)).
- Pasivna - služi za povezivanje aktivne opreme i sastoji se od kablova, konektora, razvodnog panela (engl. *patch panel*), komunikacijskih ormara i sustava za napajanje električnom energijom (vodovi, sklopke i naponske letve, sustav za hlađenje).

3.1.1. Aktivna mrežna oprema

Pomoću aktivne komunikacijske opreme možemo aktivno upravljati mrežnim prometom. Oprema se sastoji od uređaja koji se napajaju električnom energijom, a zajednička im je značajka procesor i memorija. Ovi uređaji donose odluke o putanji podataka koji šalju i koji prolazi kroz njih.

Preklopnik je jedan od aktivnih mrežnih uređaja, koji u OSI-referentnom modelu radi na podatkovnoj razini te služi za povezivanje dvaju računala unutar iste mreže (LAN) ili za povezivanje računala i usmjerivača za promet koji je namijenjen drugim mrežama. Njegova osnovna funkcija je prospajati tj. komutirati okvire primljene na jednom priključku (engl. *port*) samo na onaj priključak gdje se nalazi odredište okvira. [10]

Zahvaljujući preklopniku moguće je slanje podataka većem broju korisnika istovremeno bez mogućnosti kolizije istih. Također moguće je povezivanje više *LAN* mreža bez obzira koriste li te mreže iste protokole. [11] Kako bi se omogućilo

prospajanje potrebna je tablica prospajanja (engl. *switching table*) koja sadrži MAC (engl. *Media Access Control*) [4] adrese te brojeve priključaka.

Na slici 1 prikazan je fizički izgled preklopnika.



Slika 1. Preklopnik

Izvor: [12]

Usmjernik (engl. *router*), čiji je fizički izgled prikazan na slici 2, je mrežni aktivni uređaj koji ima dvije glavne funkcije, a to su funkcije usmjeravanja (engl. *routing*)[5] i prosljeđivanja (engl. *forwarding*) paketa ili jednostavnije rečeno služi za usmjeravanje prometa prema logičkim adresama (IP adrese) računala u mreži. Za razliku od preklopnika, koji koristi MAC adresu za prospajanje, usmjernik na temelju odredišne IP adrese, koja je upisana u zaglavlje paketa i tablice usmjeravanja (engl. *routing table*) odlučuje na koje sučelje prosljeđuje paket. [13]



Slika 2. Usmjernik

Izvor: [14]

Kako bi usmjernik „znao“ gdje treba proslijediti paket koristi određena pravila usmjeravanja tj. protokole. Ti protokoli omogućuju da se mreža sama prilagodi danim uvjetima rada kao i dinamičko oglašavanje usmjernika, a postoje dvije vrste usmjerivačkih protokola: [15]

- Protokoli vektora udaljenosti (engl. *distance-vector*)
- Protokoli stanja veze (engl. *link-state*)

[4] **MAC adresa** je fizička adresa kodirana u ROM (engl. *Read Only Memory*) svakog mrežnog uređaja. Sastoji se od 48 bita i jedinstvena je za svaki uređaj. [31]
[5] **Usmjeravanje (engl. *routing*)** je postupak odabira puta za slanje podataka računalnom mrežom

3.1.2. Pasivna mrežna oprema

Kako je već prije spomenuto, pasivna mrežna oprema služi za povezivanje aktivne mrežne opreme, a s obzirom na to kako imamo različite vrste signala tako imamo različite vrste medija kojima se prenose signali.

Nezaštićena upletena bakrena parica (engl. *Unshielded Twisted Pair - UTP*) je jedna od primjera upletenih bakrenih parica koja se sastoji od 8 upletenih bakrenih žica, a postoje različite kategorije (Cat3, Cat5…) za različite brzine prijenosa.

Zaštićena upletena bakrena parica (engl. *Shielded Twisted Pair - STP*) ima i metalni oklop koji sprječava elektromagnetske smetnje koje bi mogle prouzrokovati oštećenje signala. Sastoji se od dva prepletena vodiča i cilindričnog opleta, koji može biti i folijski.

Koaksijalni kabeli ima bolju oklopljenost od upletene bakrene parice čime je povećana otpornost na smetnje i greške te puno veći frekvencijski pojas (engl. *bandwidth*)[6] i samim time omogućuje veću brzinu prijenosa podataka. Sastoje se od centralnog vodiča i cilindričnog opleta.

Optički kabel je dosta sličan koaksijalnom kabelu, ali bez vanjskog cilindričnog opleta i umjesto bakrenog vodiča u središtu se nalazi staklena jezgra. Zahvaljujući svojstvima svjetlosti, optički prijenos omogućuje veliku brzinu prijenosa podataka uz zanemarivo malo slabljenje signala. Također su imuna na električnu interferenciju, što sprječava transkripciju između signala u različitim kabelima i usputno primanje okolišnog šuma. [16] Podjela je na jednomodno optičko vlakno koje kroz jezgru propušta samo jednu zraku svjetlosti i na višemodno optičko vlakno u koje ulazi više zraka koji totalnom refleksijom putuju kroz vlakno.

Ostala pasivna mrežna oprema se sastoji od:

- Konektora;
- Razvodnih panela (*patch panel, switching panel, punch-down panel*);
- Komunikacijskih ormara koji služe za smještaj pasivne i aktivne mrežne opreme (kablovi, patch paneli, prespojnici, usmjernici, serveri i dr.);
- Sustava za napajanje električnom energijom (vodovi, sklopke i naponske letve, sustav za hlađenje);
- Koncentratora (engl. *Hub*) koji spada u pasivnu mrežnu opremu zato što mu je jedina funkcija pojačavanje primljenih signala i prosljeđivanje istih dalje na sve svoje priključke. Negativna strana koncentratora je upravo to prosljeđivanje paketa na sve priključke jer se time stvara zagušenje toka podatkovnog prometa i samim time dolazi do gubitaka paketa (engl. *Packet loss*). [17]

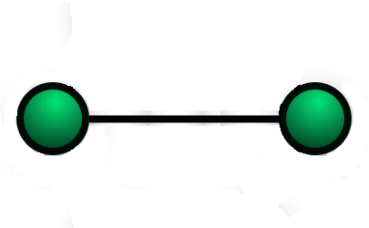
[6] ***Bandwidth*** predstavlja širinu prijenosnog pojasa koja se u podatkovnim mrežama uglavnom odnosi na nominalnu maksimalnu brzinu prijenosa podataka mjerenu u bitima po sekundi (bit/s). [34]

3.2. Topologije računalnih mreža

Kako postoji više različitih kategorija po kojima se definira i utvrđuje rad mreže, na osnovu istih možemo razložiti mrežu na manje dijelove i napraviti raspored elemenata koji čine tu mrežu. Mrežna topologija je naziv tog rasporeda te predstavlja shematski prikaz povezanosti mrežnih uređaja i veza između njih te opisuje njihov međusobni raspored.

Mrežna topologija se dijeli na fizičku topologiju, koja prikazuje fizički način povezanosti mrežnih elemenata i veza te na logičku topologiju koja opisuje način na koji se podaci šalju kroz mrežu. [18] Najčešće fizičke topologije su napisane i opisane u nastavku.

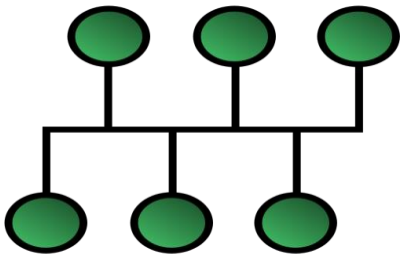
Topologija „od točke do točke“ (engl. *point to point*) je mrežna topologija koja se sastoji od dva čvora (čvorove u ovom slučaju predstavljaju računala, usmjernici, poslužitelji itd.) i komunikacijske veze između njih. Ti čvorovi međusobno komuniciraju, a prikaz topologije je vidljiv na slici 3. Veza između čvorova može biti stalna ili dinamička. Kod dinamičke veze, dijelovi se podataka koji su pakirani u pakete, usmjeravaju preko dijeljenih veza između dva čvora koji međusobno komuniciraju.



Slika 3. *Point-to-point* mrežna topologija

Izvor: [19]

Kod sabirničke mrežne topologije (engl. *bus network topology*) svi čvorovi su povezani na centralnu vezu, što je vidljivo na slici 4 i svi podaci se šalju preko te veze. Nedostatak ovakvog načina povezivanja dolazi do izražaja kada više čvorova šalje podatke u isto vrijeme i tada postoji mogućnost sudara poruka. U slučaju sudara poruka čvor čeka neko slučajno odabrano vrijeme i zatim ponovno šalje poruku. Kada bi došlo do prekida u *bus* mrežnoj topologiji, došlo bi do prekida u komunikaciji između svih čvorova.

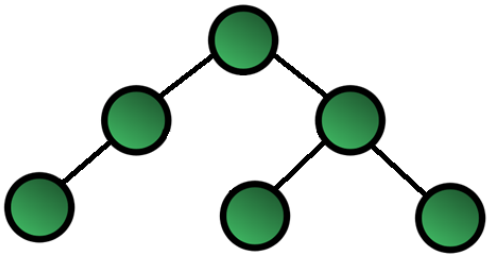


Slika 4. Sabirnička mrežna topologija

Izvor: [19]

Hijerarhijsku mrežnu topologiju (engl. *hierarchical network topology*) čine centralni (engl. *root*) čvor, koji je najviši u hijerarhijskom rasporedu te ostali čvorovi koji se nalaze u nižim slojevima kojih mora biti najmanje tri. Koristi se uglavnom u LAN mreži, a sama topologija je ukorijenjena u tri vrste mrežnih slojeva.

Najniži pristupni sloj u kojem su računala povezana. Srednji sloj koji se još naziva sloj razdjeljivanja i djeluje kao međuprostor između gornjeg i donjeg sloja. Najviši sloj koji je poznat kao jezgra sloja. [20] Terminal unutar ove topologije može komunicirati unutar vlastite podmreže, a tražiti vezu s drugim podmrežama preko nadređenog sloja ili razine. Na slici 5 je prikaz hijerarhijske mrežne topologije.



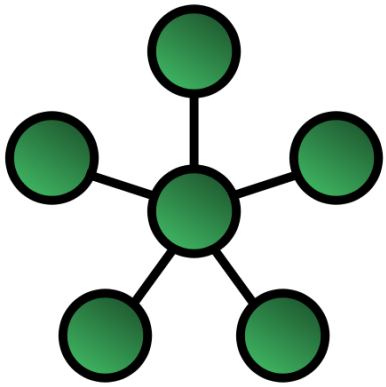
Slika 5. Hijerarhijska mrežna topologija

Izvor: [19]

Kod zvjezdaste mrežne topologije (engl. *star network topology*) svaki čvor mreže je spojen sa zajedničkim središnjim čvorom kroz kojeg prolazi cjelokupni podatkovni promet (izgled topologije prikazan je na slici 6). Središnji čvor vrši nadzor nad mrežom i obično se naziva nadzorno računalo mreže ili poslužitelj.

Ovdje je prikazan centralizirani način rada gdje mreža ne može raditi bez nadzornog računala pa neispravan rad ili isključenje nadzornog računala znači prekid rada cjelokupne mreže te ovakav sustav nije pogodan za primjenu na mreže širokog

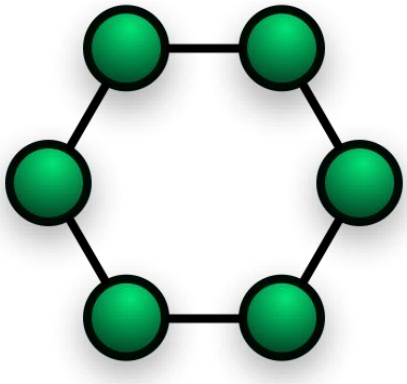
područja. Zato se ova topologija, sa svojim podvrstama, najčešće koristi pri povezivanju unutar lokalnih mreža (LAN).



Slika 6. Zvjezdasta mrežna topologija

Izvor: [19]

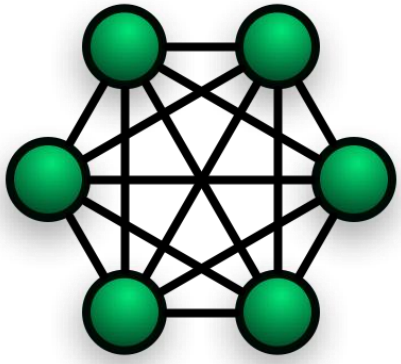
Kada imamo skup čvorova gdje je svaki čvor spojen sa samo susjedna dva čvora, a prvi i zadnji čvorovi su međusobno povezani, onda tu vrstu rasporeda nazivamo prstenasta mrežna topologija (engl. *ring network topology*). Za lakše razumijevanje objašnjenja, raspored prstenaste mrežne topologije je prikazan na slici 7. Prijenos je u ovom slučaju jednosmjeran što znači da podaci putuju od čvora do čvora u samo jednom pravcu. U slučaju kada jedan čvor pokuša komunicirati s čvorom koji mu nije susjedan, a postoji prekid na jednoj vezi, onda podaci putuju obilaznim putem tj. u suprotnom smjeru do odredišta.



Slika 7. Prstenasta mrežna topologija

Izvor: [19]

Isprepletena mrežna topologija (engl. *mesh network topology*) se sastoji od čvorova koji su međusobno direktno povezani što je vidljivo na slici 8. [20]



Slika 8. Isprepletena mrežna topologija

Izvor: [19]

Ovakvim načinom povezivanja osigurana je pouzdanost rada mreže, a samim time i stabilnost što znači da u slučaju kvara jednog čvora neće doći do narušavanja komunikacije između ostalih čvorova. Takva vrsta topologije je skupa i neekonomična te kompleksna za implementaciju na terminalske i druge mreže s malim iskorištenjem pa se najčešće koristi na mjestima poput nuklearne centrale gdje nema velik broj čvorova, ali je potrebna stalna povezanost i komunikacija istih. [2]

3.3. Referentni modeli računalnih mreža

S velikim brojem mreža koje su međusobno povezane, a čije su specifikacije i načini implementacije različiti, postoji potreba za modelom koji bi omogućio funkcionalan rad istih. Ali nije dovoljno samo razviti model, potrebno je razviti i pravila tj. protokole po kojima će se odvijati prijenos podataka. Zato je međunarodna organizacija za standardizaciju (engl. *International Organization for Standardization* - ISO) istraživala različite načine i mrežne sheme, te je stvoren mrežni model koji je osigurao umrežavanje na velikim geografskim područjima bez promjene kodova i oblika poruka tijekom mrežne komunikacije. [21] Tako je nastao OSI referentni model.

Naravno ovakav model se u današnje vrijeme koristi samo u istraživačke svrhe za promatranje razvoja mreža i protokola. Model, koji se zbog svoje jednostavnosti danas koristi je TCP/IP (engl. *Transmission Control Protocol/Internet Protocol*) model. Protokoli, slojevi oba modela te funkcija istih, biti će navedeni i ukratko objašnjeni u nastavku rada.

3.3.1. OSI referentni model

OSI RM predstavlja teoretski model koji prikazuje kako dva različita sustava mogu međusobno komunicirati tj. pokazuje funkcije veze između istih te definira sedam protokolskih razina tj. slojeva.

Fizički sloj koji je definiran fizičkim medijem za prijenos električnih, svjetlosnih ili radio signala kroz mrežu. Razlikuje se od ostalih slojeva u tome što ne postoji razina ili sloj ispod tog sloja, već samo pasivno okruženje.

Sloj podatkovne veze je sloj čija je zadaća slanje podataka do fizičkog sloja te obavljanje kontrole i korekcije pogrešaka. Preklopnik se smatra uređajem podatkovnog sloja zbog mogućnosti kontrole prometa na osnovu adresnih informacija.

Mrežni sloj određuje najbolji put za prijenos podataka između dva računala u mreži, upravlja s adresiranjem poruka i prevođenjem logičkih adresa u fizičke adrese te ima mogućnost segmentacije IP paketa u manje segmente, ako to mreža zahtijeva. Kako bi se podaci pravilno usmjerili kroz mrežu, kreirani su protokoli usmjeravanja.

Protokoli usmjeravanja omogućuju mreži dinamičko prilagođavanje uvjetima, a dijele se na protokole vektora udaljenosti, koji određuju najbolju putanju na osnovu informacije koliko je udaljeno odredište paketa i protokole stanja veze, koji rade na način da usmjernici u mreži međusobno razmjenjuju informacije o stanju u mreži. Najpoznatiji protokoli usmjeravanja su RIP[7] (engl. *Routing Information Protocol*) te OSPF[8] (engl. *Open Shortest Path First*). [22]

Transportni sloj ima zadaću segmentiranja i spajanja podataka u jednu cjelinu. Također omogućuje krajnju komunikaciju između dva računala gdje je svaki paket numeriran pri slanju na odredišno računalo te služi kao provjera kako bi se osiguralo da svi podaci budu primljeni.

Sloj sesije osigurava komunikaciju između aplikacija, koje se izvršavaju na računalima u mreži, izvršava sigurnosne funkcije te vrši koordinaciju zahtjeva usluga i događaja

Prezentacijski sloj je šesti sloj u OSI referentnom modelu i njegova glavna uloga je da brine o tome da informacija koju pošalje aplikacijski sloj jednog sustava bude čitljiva od strane aplikacijskog sloja drugog sustava. Drugim riječima, definira način formatiranja podataka prilikom njihove razmjene između računala. [22]

Aplikacijski sloj je zadnji sloj i omogućuje aplikacijama pristup mrežnim uslugama.

3.3.2. TCP/IP referentni model

TCP/IP model je standard koji ima definirani skup protokola raspoređenih u razine koje odgovaraju grupama kod OSI referentnog modela. Slojevi od kojih se sastoji

[7] **RIP (engl. *Routing Information Protocol*)** je najstariji protokol koji spada u protokole vektora udaljenosti te kao metriku koristi broj koraka**.** [39]

[8] **OSPF (engl. *Open Shortest Path First*)** je hijerarhijski protokol stanja veze koji rute proračunava Dijkstra algoritmom i koristi određeni algoritam za provjeru autentičnosti usmjernika. [39]

TCP/IP model su aplikacijski sloj, transportni sloj, mrežni sloj tj. Internet sloj i sloj mrežnog sučelja te fizički sloj.

Fizički sloj, koji odgovara sloju 1 OSI referentnog modela, obuhvaća fizičke komponente uređaja koje se koriste za prijenos podataka te definira karakteristike prijenosnih medija, brzine prijenosa i sl.

Sloj mrežnog sučelja definira način organizacije podataka unutar okvira te način slanja okvira unutar mreže što je slično načinu rada sloja podatkovne veze unutar OSI referentnog modela.

Internet sloj koristi različite mehanizme za prosljeđivanje i usmjeravanje paket iz izvorišnog prema odredišnom računalu, ali ne brine o redoslijedu datagrama. Predstavnik ovog sloja je Internet protokol koji ima funkciju adresiranja i usmjeravanja paketa kroz mrežu.

TCP protokol korespondira transportnom sloju OSI modela. To je glavni konekcijski orijentirani protokol koji je zadužen za nadzor prijenosa, adresiranje i multipleksiranje te vršenje kontrole toka. Također uspostavlja logičku vezu između procesa u mreži. [22]

Posljednji sloj predstavlja sloj aplikacije i čine ga različiti aplikacijski protokoli koji koriste različite aplikacijske protokole za izvođenje prijenosa informacija.

4. Funkcije i primjena mrežnih emulatora

Kako je već spomenuto u uvodu, razvojem i širenjem računalnih mreža ponestalo je prostora za eksperimentiranje raznih mrežnih topologija u stvarnom svijetu (engl. *Real-world testing*), stoga se javlja potreba za alternativnim načinom „isprobavanja“ i testiranja mrežne konfiguracije. Tehnike testiranja i ispitivanja mreže, koje spadaju u tu skupinu alternativnog rješenja su simulacija te emulacija, a softversko/hardverske komponente koje omogućuju oponašanje rada računalnih mreža su simulatori i emulatori.

Glavna svrha takvih eksperimenata je procjenjivanje performansi mrežnog sustava, a ovisno o tome što se ocjenjuje, može se razlikovati nekoliko vrsta eksperimenata emulacije koje se mogu koristiti za procjenu mrežne opreme, aplikacija te protokola. Kako bi se moglo razraditi i objasniti rad emulatora, potrebno je prvo definirati i objasniti sam proces emulacije te razliku između ostalih eksperimentalnih tehnika.

4.1. Eksperimentalne tehnike

Sam pojam „emulacija“ označava tehniku eksperimentiranja čija je glavna funkcija stvaranje osnovnih funkcionalnosti nekog sustava, tj. oponašanje funkcioniranja sustava, u stvarnom vremenu i kontroliranom okruženju, kako bi se omogućila interakcija i komunikacija s ostalim stvarnim sustavima. Cilj je postizanje okruženja u kojem vanjski promatrač ne vidi razliku između emuliranog i stvarnog sustava.

Prva postavka mrežne emulacije pojavila se zbog potrebe da se kontrolira degradacija[9] kvalitete mreže tako da se obavljaju djelomično ponovljivi testovi s mrežnim protokolima koji su bili u razvoju u to vrijeme. [23] Mrežna emulacija je termin koji još uvijek nije točno definiran i ne postoji potpuna suglasnost o razlikama između simulacije i emulacije, što ne znači da nema razlike. Iako je rečeno da nema točne definicije, većina istraživača se slaže da je mrežna emulacija modeliranje na razinama koje omogućuju ispitivanje stvarnih aplikacija i sustava uz zamjenu većeg ili manjeg dijela fizičkih komponenti sustava sintetskima, pri čemu je uobičajeni cilj da se rad sustava promatra u stvarnom vremenu. [24]

Drugim riječima, emulacija mreže je tehnika eksperimentiranja koja sadrži stvarne[10] mrežne komponente, bilo da se radi o hardveru ili softveru, te komponente koje se emuliraju putem računalnih modela u stvarnom vremenu, a samim time i mogućnost povezivanja s vanjskom stvarnom mrežnom topologijom. Što znači da se prilikom izvođenja mrežne emulacije pokušavaju stvoriti uvjeti mreže koji odgovaraju određenom mrežnom scenariju te se dolazi do pretpostavke da će stvarne komponente, uključene u emulaciju, stupiti u interakciju sa emuliranim komponentama

[9] **Degradacija kvalitete mreže** je pad parametara performansi mreže koji se javljaju dok se vrši promet kroz komunikacijsku mrežu

[10] **Pod stvarne** se misli na stvarne u „fizičkom“ svijetu (usmjernik, preklopnik i sl.)

kao u stvarnom svijetu. Također je bitno napomenuti, kao jednu od važnijih razlika, da je nemoguće imati apsolutno ponovljiv redoslijed događaja u emulaciji. [25]

Kako bi se eksperimentalne tehnike lakše usporedile, u nastavku su navedeni kriteriji po kojima se vidi razlika između istih: [23]

- Troškovi eksperimentalne tehnike (engl. *Experimentation cost*) koji predstavljaju troškove provođenja eksperimenta s određenom tehnikom s ekonomskog i vremenskog gledišta, ali isključujući fiksne troškove koji obuhvaćaju alate koji se mogu iznova koristiti sa svakim novim eksperimentom;
- Izvođenje u stvarnom vremenu (engl. *Real-time execution*) bi definiralo da li se eksperiment, izvršen određenim postupkom, učinkovito izvršavao u stvarnom vremenu ili ne;
- Kontrola nad eksperimentalnim uvjetima (engl. *Control over experimental conditions*) prikazuje u kojoj mjeri se uvjeti eksperimenta mogu kontrolirati za određenu tehniku i koliki je opseg eksperimentalnih uvjeta koji se mogu proučavati;
- Pouzdanost rezultata (engl. *Result reliability*) prikazuje realnost eksperimentalnih rezultata dobivenih određenim postupkom te kolika je pouzdanost istih u svrhu donošenja praktičnih odluka o budućem sustavu;
- Jednostavnost korištenja (engl. *Ease of use*) gdje se vidi u kojoj mjeri je težina izrade eksperimenta.

Temeljem navedenih kriterija napravljena je usporedba eksperimentalnih tehnika koja je vidljiva u tablici 1.

Tablica 1. Usporedba eksperimentalnih tehnika prema navedenim kriterijima

	Simulacija	Testiranje u realnom svijetu	Emulacija
Troškovi eksperimentalne tehnike	**Niski**	**Visoki**	**Srednji**
Izvođenje u stvarnom vremenu	**Ne**	**Da**	**Uglavnom da**
Kontrola nad eksperimentalnim uvjetima	**Visoka**	**Niska**	**Visoka**
Pouzdanost rezulatata	**Niska**	**Visoka**	**Srednja**
Jednostavnost korištenja	**Visoka**	**Niska**	**Srednja**

Izvor: [23]

Vidljivo je kako testiranje u realnom svijetu ima najveću pouzdanost rezultata no zbog troškova eksperimentiranja nije baš ekonomski isplativo. Simuliranje prema navedenim kriterijima ima prednost samo u jednostavnosti korištenja. Stoga eksperiment emulacije ima veliku prednost zbog troškova eksperimenta, izvođenja u realnom vremenu, kontrolom nad eksperimentalnim uvjetima te pouzdanosti rezultata.

4.2. Klasifikacija mrežnih emulatora

Mrežna emulacija je vrsta eksperimentiranja koja se nalazi između simulacije i testiranja u realnom svijetu jer se koristi i modelima i stvarnim elementima. Koliko će modela i stvarnih elemenata koristiti, ovisi o samom mrežnom emulatoru, a i o potrebama samog istraživanja. S obzirom na to da postoji veliki broj mrežnih emulatora koji imaju razne mogućnosti i specifikacije, potrebno ih je klasificirati. Mrežni emulatori klasificirani su prema: [23]

- Dostupnosti mrežnih emulatora;
  - Besplatni mrežni emulatori
  - Istraživački mrežni emulatori
  - Komercijalni mrežni emulatori

- Načinu implementacije i funkcionalnosti;

    - Softverski mrežni emulatori
    - Hardverski mrežni emulatori
    - Network testbed

- Razini na kojoj se mreža emulira;
    - Emulatori prijenosnih kanala
    - Emulatori mrežnih topologija

- Složenosti korištenog mrežnog modela
    - Emulatori modeliranja niske složenosti
    - Emulatori modeliranja srednje složenosti
    - Emulatori modeliranja visoke složenosti

U sljedećim podpoglavljima ukratko su opisane klasifikacije mrežnih emulatora.

4.2.1. Klasifikacija prema dostupnosti mrežnih emulatora

Emulatori otvorenog koda[11] (engl. *open source*), koji su dostupni za širu javnost te se mogu lako instalirati i primjenjivi su na mnogim operativnim sustavima spadaju u kategoriju besplatnih mrežnih emulatora.

Istraživački mrežni emulatori predstavljaju emulatore čija je upotreba ograničena samo na istraživačku zajednicu. Ovo ne mora nužno značiti da ta kategorija emulatora nije besplatna za korištenje, ali je ponekad potrebno imati licence za korištenje istih. Što također znači da su potrebne određene vještine i znanja visoke razine na području računala i računalnih mreža kako bi se moglo koristiti istraživačkim mrežnim emulatorima.

Komercijalni mrežni emulatori, čiji izvorni kod nije javno dostupan, se odnose na sva emulacijska rješenja koja su pružena na komercijalnoj bazi. Iza ove kategorije emulatora stoje profesionalne dokumentacije o radu i specifikacijama istih kao i tehnička podrška. [23]

4.2.2. Klasifikacija prema načinu implementacije i funkcionalnosti

Softverski mrežni emulator je emulator koji postoji samo u elektronskom obliku i zahtijeva dodatni hardver za izvršenje (npr. obično računalo ili neku drugu ekvivalentnu izvršnu platformu). [23] Drugim riječima ovaj emulator predstavlja program koji se implementira i izvršava na nekom računalu.

Ovakva vrsta emulatora je najrasprostranjenija i najčešće korištena zbog same jednostavnosti implementacije i raznih mogućnosti i specifikacija, ali glavna mana je ta

[11] **Otvoreni kod** znači da krajnji korisnik može čitati, izmjenjivati glavni izvorni kod programa, a samim time i sudjelovati u unaprjeđenju istog.

da je izvršavanje emulacije ograničeno karakteristikama platforme na kojoj se izvršava. Ipak, distribuirani pristup emulacije omogućava da se isti emulatori izvršavaju na više mjesta pa je samim time moguća izvedba emulacije kompleksnih mreža.

Hardverski mrežni emulator je emulator koji se isporučuje kao hardverski uređaj koji pruža emulacijsku funkcionalnost na samostalan način. [23] Vrlo je vjerojatno da unutar hardverskog emulatora postoji računalo na koje je instaliran softverski mrežni emulator koji izvršava proces emulacije, no takvo računalo je onda hardverski striktno namijenjeno softverskom dijelu emulatora. Svojstva i mogućnosti hardverskih mrežnih emulatora su daleko veća nego kod softverskih emulatora, ali je time i sama cijena puno veća.

Posebna kategorija raspodijeljenih sustava koja je namijenjena isključivo za ispitivanje rada prototipova drugih raspodijeljenih sustava ili mrežnih protokola naziva se *network testbed*. [24] U konačnici to je izolirani skup računala i povezanih mrežnih komponenata. Razlika između klasične mreže i *network testbed-a* je u tome što se *network testbed* koristi isključivo za istraživačke svrhe.

Network testbeds su pogodno rješenje za testiranje mreže općenito, a posebno emulacija mreže, iz nekoliko razloga: [23]

- Hardver *testbeda* se može ponovno koristiti za nekoliko eksperimenata, bez potrebe za postavljanjem i povezivanjem hardverskih uređaja svaki put.
- Korisnički softver može se napisati za određeni testni sloj kako bi se pojednostavnile zadaće upravljanja eksperimentom na *testbedu*.
- *Testbeds* omogućuju provođenje eksperimenata velikih razmjera po jeftinijem trošku nego što je klasičnom kupovinom potrebnog hardvera ili preraspodjelom postojećeg hardvera u svrhu testiranja.

4.2.3. Klasifikacija prema razini na kojoj se mreža emulira

Najjednostavniji oblik emulacije predstavlja emulacija prijenosnih kanala gdje se jedino gledaju parametri (propusnost[12] (engl. *throughput*), kašnjenje[13] (engl. *delay*) te gubitak paketa[14] (engl. *packet loss*)) koji utječu na performanse mreže. Ono što je bitno napomenuti je da moderni mrežni emulatori izvršavaju stvarnu *end-to-end* konekciju bez obzira na kompleksnost mreže. [23] Emulatori koji izvršavaju emulaciju prijenosnih kanala najčešće se implementiraju u jezgru nekog operativnog sustava i tu se dobiva prednost zbog malog kašnjenja i varijacija kašnjenja te brz rad i visoka propusnost u usporedbi s alatima slične namjene koji se izvode u operacijskom sustavu kao

[12] **Propusnost** je parametar koji izražava efektivnu brzinu prijenosa podataka izraženu brojem prenesenih bita u sekundi. [22]
[13] **Kašnjenje** označuje vrijeme potrebno da se paket (ako se radi o paketnoj mreži) prenese od izvorišta do odredišta. [22]
[14] **Gubitak paketa** nastaje onda kada dođe do prepunjivanja spremnika u čvorovima paketne mreže (ruterima), kao posljedica čekanja paketa u redovima za usmjeravanje, odnosno rutiranje. [22]

korisnički procesi. Također, emulatori prijenosnog kanala se ne oslanjaju na virtualizacijske[15] mehanizme, pa se uobičajeno primjenjuju na način da jedno fizičko računalo emulira svojstva samo jednog prijenosnog kanala. [24]

Emulator mrežne topologije je mrežni emulator koji reproducira u cijelosti ili djelomično topologiju mreže kroz koju se krajnje točke ili terminali mogu međusobno povezati kao i degradaciju kvalitete mreže na prijenosnom kanalu unutar reproducirane topologije. [23] Prvi sustavi koji su omogućili emuliranje mreža kompleksnih topologija s obradom paketa na razini mrežnog sloja u svakom emuliranom čvoru bili su implementirani kao korisnički procesi u kojima se izvodio prilagođeni programski kod mrežnog sloja preuzet iz jezgre nekog operacijskog sustava opće namjene. No problem je nastao kod korisničkih procesa te propusnosti zato što su svi procesi ovisili o procesorskom vremenu koje se trošilo na kopiranje mrežnog prometa iz jezgre OS-a u korisnički proces i obratno. [24] Problem se riješio tako da su se korisnički procesi počeli izvršavati u virtualnim čvorovima što je znatno ubrzalo sam proces emulacije.

Emulacija je u ovom slučaju složenija i kompleksnija s obzirom na to da se žele stvoriti virtualni prikazi mrežnih uređaja, gdje bi svaki uređaj zasebno bio emuliran. Važno je napomenuti da prilikom emulacije topologije, degradacija mreže nije rezultat simulacijskog modela, već emuliranih mrežnih uređaja koji su uključeni u trenutni mrežni promet što dodatno pospješuje uvid u moguće događaje u stvarnoj mreži.

4.2.4. Klasifikacija prema složenosti korištenog mrežnog modela

Emulatori koji imaju mogućnost modeliranja računalne mreže niske složenosti zahtijevaju konfiguraciju s fiksnim vrijednostima (fiksna stopa gubitaka paketa i fiksna stopa vremena kašnjenja), koje utječu na degradaciju mreže na razini prijenosnih kanala. No ova kategorija je rijetko korištena zbog nemogućnosti primjene u distribuiranoj emulaciji.

Emulatori srednje složenosti modeliranja omogućuju određivanje varijabilnih parametara mreže, bilo na statistički način, kao distribucije, ili pomoću pravila varijacija. [23] Što znači da bi rezultati bili što vjerodostojniji tj. što bolje imitirali stvarni mogući događaj, potrebno je odabrati valjanu statističku distribuciju ili pravilo varijacije za koje je dokazano da predstavljaju mrežu koju treba emulirati.

Također, s obzirom na njihovu široko rasprostranjenu upotrebu, može se zaključiti da je razina stvarnosti, koja se emulira od strane emulatora srednje složenosti modeliranja, dovoljna za većinu komercijalnih aplikacija. U ovu kategoriju još spadaju i alati koji omogućuju „reprodukciju“ mrežnog traga koji je prethodno snimljen. [23]

[15] **Pojam „virtualizacija“** podrazumijeva apstraktno predstavljanje pojedinih funkcionalnosti i resursa. Odnosno, izvana za korisnika (čovjek ili program) nema razlike između stvarnog i virtualnog ostvarenja. [35]

Kada nam je potrebno složenije modeliranje mrežnih uređaja i parametara prijenosnih kanala te imitiranje tj. oponašanje stvarnosti, onda koristimo emulatore modeliranja visoke složenosti.

Dodatne značajke koje razlikuju emulatore modeliranja visoke složenosti od niže složenosti su: [23]

- Detaljno modeliranje mrežnih protokola, komunikacijskih kanala i dr. te
- Mogućnost virtualizacije operativnih sustava na emulirana računala.

5. Analiza mrežnih emulatora u funkciji računalnih mreža

U ovom poglavlju analizirat će se tri mrežna emulatora čije će funkcije i rad biti uspoređeni u šestom poglavlju. Nazivi emulatora koji će biti obrađeni u radu su:

- *Integrated Multiprotocol Network Emulator/Simulator* – IMUNES;
- *Common Open Research Emulator* – CORE;
- *Graphical Network Simulator 3* - GNS3.

Prikaz usporedbe na temelju klasifikacija iz prethodnog poglavlja vidljiv je u tablici 2.

Tablica 2. Usporedba mrežnih emulatora

Naziv	Dostupnost	Način implementacije	Razina emulacije	Složenost korištenog mrežnog modela
IMUNES	**Istraživački**	**Softverski**	**Mrežna topologija**	**Srednja složenost**
CORE	**Istraživački**	**Softverski**	**Mrežna topologija**	**Srednja složenost**
GNS3	**Istraživački**	**Softverski**	**Mrežna topologija**	**Visoka složenost**

Izvor: [23]

Može se primijetiti kako su emulatori, čiji će analiza funkcija biti provedena u nastavku, praktički isti po navedenoj klasifikaciji. Jedino se emulator GNS3 razlikuje od ostala dva zbog svoje složenosti korištenog mrežnog modela.

5.1. IMUNES

IMUNES (engl. *Integrated Multiprotocol Network Emulator/Simulator*) je integrirani multi-protokolarni emulator/simulator mreže, čiji je rad temeljen na jezgri (engl. *kernel*) operativnog sustava *FreeBSD*. [26] IMUNES koristi više virtualnih čvorova koji se mogu međusobno povezati formirajući tako kompleksnu mrežu. S obzirom na to da mrežne emulacije kompleksnih mreža uvelike zauzimaju resurse računala na kojem

se izvode, dolazi do ograničenja skalabilnosti[16]. [27] No takav problem je riješen distribuiranim načinom emulacije što znači da se pomoću IMUNES emulatora mogu izvoditi projekti velikih razmjera. Prema podjeli tj. specifikacijama IMUNES spada u istraživački emulator mrežne topologije koji ima mogućnost emulacije mreža srednje složenosti te mogućnost distribuirane izvedbe.

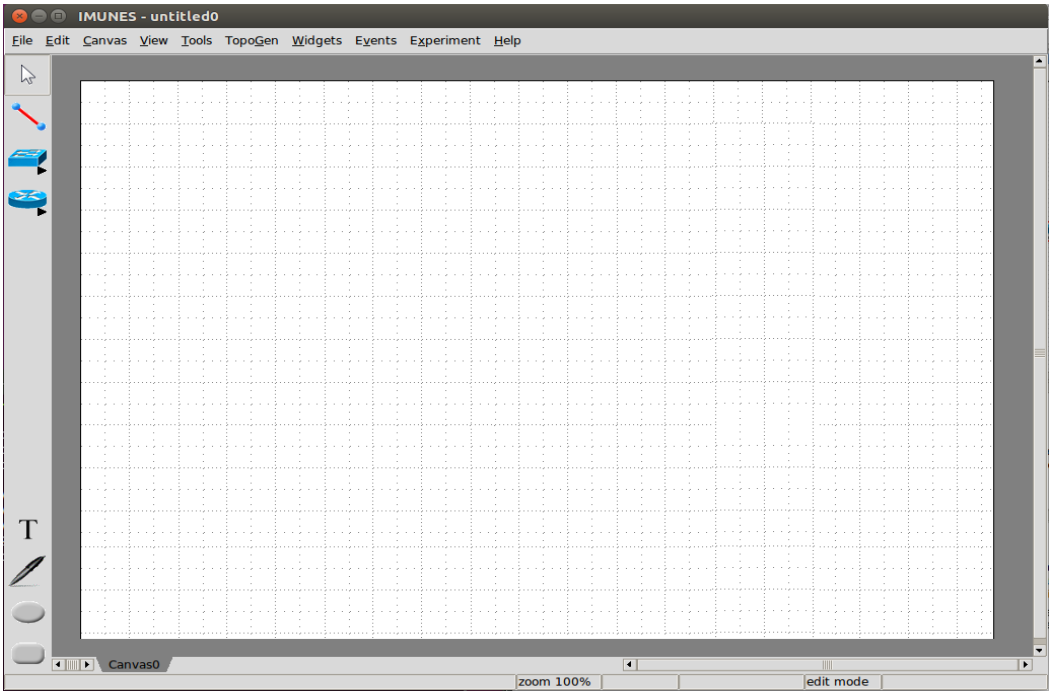
Neke od bitnijih karakteristika IMUNES emulatora su:

- Emulacija/simulacija topologije IP mreže u stvarnom vremenu pri brzinama od nekoliko gigabita po sekundi;
- Mogućnost postavljanja do 1000 virtualnih čvorova na računalu gdje je svaki čvor sposoban pokretati UNIX aplikacije;
- Jednostavnost korištenja koja je omogućena grafičkim korisničkim sučeljem koje je prikazano na slici 9.;
- Mogućnost povezivanja na vanjske stvarne uređaje te mogućnost povezivanja na internet;
- Automatska i ručna konfiguracija mrežnih uređaja te prijenosnih kanala;
- Mogućnost automatskog postavljanja mrežne topologije.

Iako se izrada mrežne topologije te konfiguracija elemenata može obaviti na bilo kojem operativnom sustavu, postupak emulacije tj. izvršavanje eksperimenta je moguće samo na *FreeBSD OS-u* što bi se moglo reći da je glavni nedostatak IMUNES emulatora.

U nastavku će se pojasniti i prikazati grafičko korisničko sučelje emulatora te mogućnosti alata koji će se koristiti.

[16] **Skalabilnost** je sposobnost sustava da se prilagodi povećanim zahtjevima obrade na predvidiv način, bez da postane previše kompleksan, skup i nepraktičan.



Slika 9. Prikaz korisničkog grafičkog sučelja IMUNES emulatora

Na slici 10 se vidi prikaz alatne trake s elementima računalne mreže koja se koristi pri izradi topologije mreže. Svaki od elemenata ima neku funkciju te su iste, redoslijedom prema slici, navedene te ukratko objašnjene u nastavku.



Slika 10. Prikaz alatne trake s mrežnim elementima u IMUNES emulatoru

Izvor: [26]

- Alat za odabir elemenata (engl. *select tool*) koji služi kako bi se moglo proizvoljno odabrati te postaviti elemente na radnu površinu.
- Veza (engl. *link*) predstavlja element koji spaja čvorove u mreži.
- Koncentrator (engl. *hub*) predstavlja element koji prosljeđuje svaki dolazni paket na sve svoje ulaze i samim time na svaki povezani čvor.
- LAN preklopnik (engl. *LAN switch*) je element koji prosljeđuje dolazne pakete na povezane čvorove koristeći adrese iz tablice odredišta.
- Usmjernik (engl. *router*) je element mrežnog sloja koji je sposoban za prosljeđivanje paketa pomoću rute dobivene dinamičkim protokolima usmjeravanja (dostupno putem mrežnih integriranih programa za usmjeravanje *Quagga* ili *Xorp*).
- *Host* predstavlja element mrežnog sloja koji ne prosljeđuje pakete i ima statičke rute.
- Virtualno računalo – PC je element koji također ne šalje pakete i ima statičku rutu, ali za razliku od *hosta*, ne pokreće nikakve mrežne usluge.

- Fizičko sučelje (RJ45) (engl. *Physical interface*) je alat koji pruža mogućnost povezivanja virtualnog čvora sa stvarnim fizičkim sučeljem.
- Tekst (engl. *text*) predstavlja alat za dodavanje novog teksta na platnu.
- *Oval* predstavlja alat za dodavanje nove boje na radnoj površni.
- Pravokutnik (engl. *Recatngle*) je alat za dodavanje novog pravokutnika na radnu površinu.
- *Freeform* je alat za dodavanje novog slobodnog oblika na radnu površinu.

5.2. CORE

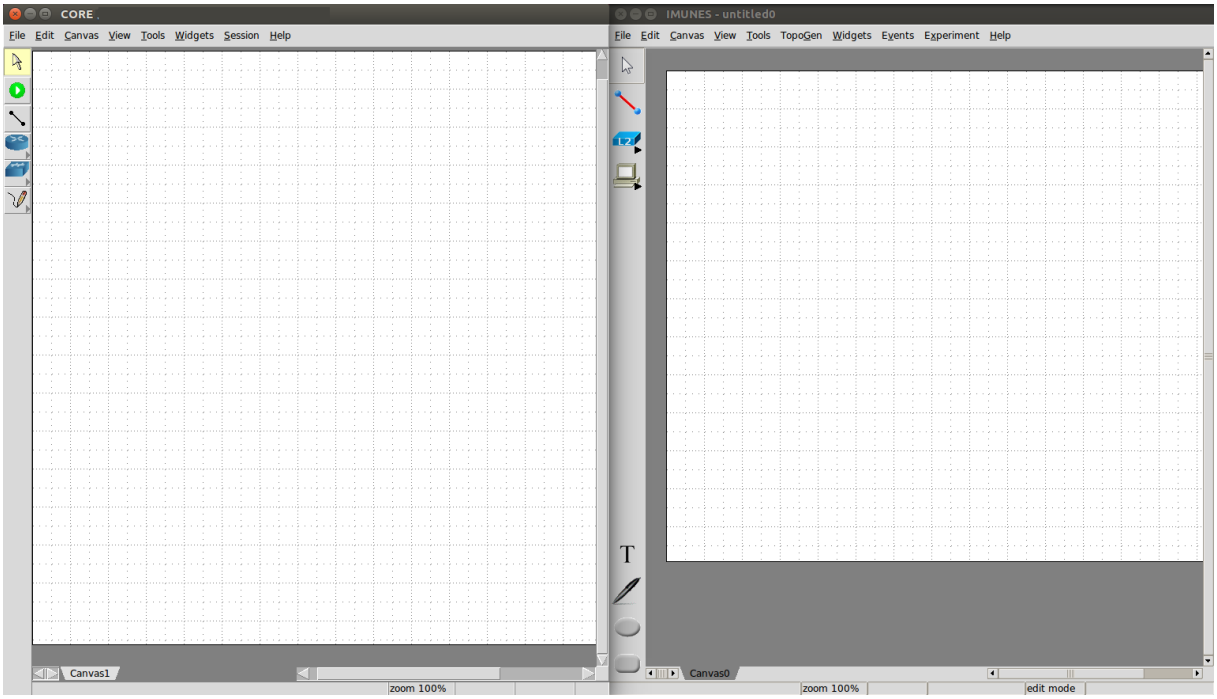
CORE (engl. *Common Open Research Emulator*) je emulator koji služi za izradu virtualnih topologija mreža te omogućuje emulaciju na kompletnih mreža na jednom ili više *hostova*. [28] Funkcija mu je emuliranje usmjernika, računala, preklopnika te veza među njima. Bitno je napomenuti da se bazira na otvorenom kodu IMUNES emulatora stoga su dosta slični po radu, grafičkom sučelju te ostalim funkcijama. No za razliku od IMUNES emulatora, CORE emulator može izvršiti eksperiment emulacije unutar Linux operativnog sustava te sadrži podršku za:

- Bežične mreže;
- Mobilno skriptiranje;
- Distribuiranu emulaciju preko više računala;
- IPsec[17];
- Kontrola vanjskih *linux* usmjernika.

Kao i svaki softverski emulator, CORE također teži prema što jednostavnijem pristupu samom programu od strane korisnika pa su postavljeni ciljevi koji su ujedno i glavne značajke emulatora. Težnja prema jednostavnosti korištenja je primarni cilj kako bi korisnik u nekoliko koraka pokrenuo rad neke mreže. Također su bitni učinkovitost i skalabilnost kako bi se omogućila emulacija što više čvorova na jednom računalu. Jedna od velikih prednosti je umrežavanje s vanjskim stvarnim uređajima ili virtualnim uređajima pokrenutim na drugom računalu te mogućnost čitanja i modificiranja izvornog koda. [28]

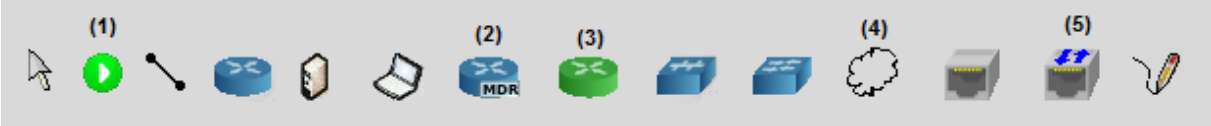
Kako je prije spomenuto, sličnosti sa IMUNES emulatorom su vidljive i u grafičkom sučelju što je predočeno na slici 11.

[17] **IPSec (engl. *IP Security*)** je sigurnosni protokol koji omogućava autentifikaciju i šifriranje na Internet. [36]



Slika 11. Usporedba grafičkih sučelja CORE i IMUNES emulatora

Postoje sitne razlike što se tiče redoslijeda i rasporeda elemenata na alatnoj traci, što je vidljivo na slici 12, stoga će se redom objasniti samo oni elementi koji se ne nalaze u IMUNES emulatoru.



Slika 12. Prikaz alatne trake s mrežnim elementima u CORE emulatoru

- Gumb za pokretanje (1) (engl. *Start button*) – kao što i sam naziv govori, služi za pokretanje procesa emulacije.
- *OSPF MANET Designated Routers* – MDR (2) – predstavlja usmjernik za bežične mreže koji je definiram *OSPFv3* usmjernim protokolom kako bi se postiglo *MANET*[18] optimizacijsko usmjeravanje.
- *PRouter* (3) je fizički usmjernik koji predstavlja stvarni *testbed* stroj.
- Bežična lokalna mreža (4) (engl. *Wireless LAN*) – predstavlja čvorište na koje su povezani mrežni elementi te kontrolira povezanost između ostalih čvorova na temelju udaljenosti između njih
- *Tunnel* (5) – alat koji omogućuje povezivanje više emulacija provedenih od strane CORE emulatora, koristeći *GRE* tuneliranje[19]

[18] **Mobilna ad hoc mreža (engl. *Mobile Ad Hoc Network* - MANET)** je autonomni skup mobilnih uređaja koje komuniciraju putem bežičnih veza relativno male propusnosti. [37]
[19] **Enkapsulacija generičkog usmjeravanja (engl. *Generic Routing Encapsulation* – GRE)** je protokol tuneliranja koji stvara virtualne kanala kojima podaci putuju od izvorišta do odredišta

5.3. GNS3

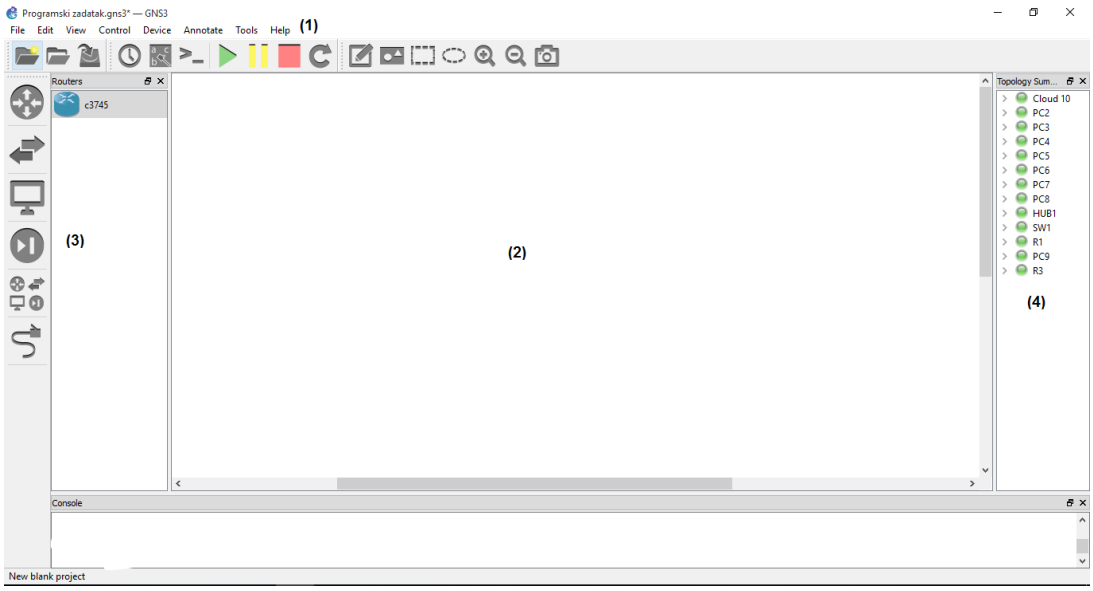
GNS3 (engl. *Graphical Network Simulator 3*) predstavlja grafički emulator mrežnih topologija i prijenosnih kanala. Kao IMUNES i CORE, GNS3 je emulator otvorenog koda. Omogućuje emulaciju kompleksnih mreža u kojoj je moguće detaljno modeliranje mrežnih uređaja i kanala pa samim time, za razliku od IMUNES i CORE emulatora, spada u kategoriju emulatora mreža visoke složenosti. Također je bitno napomenuti da se usmjernici moraju naknadno preuzeti s Interneta te ručno konfigurirati (za razliku od CORE i IMUNES emulatora gdje su usmjernici integrirani i unaprijed konfigurirani) što će biti prikazano u šestom poglavlju.

Unutar emulatora integrirani su dodatni alati poput:

- *VirtualBox,* koji omogućuje emuliranom računalu virtualizaciju operativnog sustava;
- *Wireshark,* koji predstavlja alat čija je glavna funkcija snimanje mrežnog prometa pa je samim time omogućena analiza mrežnog prometa koji se generira unutar mreže koja kreirana u emulatoru;
- te *Dynamips* alat koji omogućuje emulaciju usmjernika proizvođača *Cisco* i *Juniper,* kao i emulaciju *Cisco* vatrozida (engl. *firewall*).

Glavne karakteristike GNS3 emulatora su:

- Neograničen broj mrežnih uređaja i čvorova koji se mogu emulirati (jedino ograničenje predstavljaju resursi računala na kojima se emulacija izvodi);
- Moguće pokretanje emulacije na bilo kojem operativnom sustavu bez administracijskih dozvola;
- Unaprijed konfigurirani i optimizirani mrežni uređaji koji su jednostavni za implementaciju [29];
- Izrada kompleksnih mrežnih topologija te povezivanje istih na vanjske mrežne uređaje;
- Izuzetna preglednost korisničkog grafičkog sučelja što je vidljivo na slici 13.



Slika 13. Prikaz korisničkog grafičkog sučelja GNS3 emulatora

- Pri vrhu grafičkog sučelja (1) nalazi se alatna traka koja služi za neke osnovne funkcije kao što su: snimanje projekta, otvaranje projekta, *screenshot* programa, pokretanje emulacije itd.
- Sredina (2) prikazuje radnu površinu na kojoj se postavljaju uređaji tj. elementi mreže koji će tvoriti mrežnu topologiju.
- Na lijevoj strani (3) grafičkog sučelja nalazi se alatna traka s mrežnim uređajima koji se koriste u izradi mrežne topologije, a postavljaju se na radnu površinu pomoću *Drag 'n drop* funkcije.
- Na desnoj strani (4) sučelja prikazan je popis uređaja koji se nalaze na radnoj površini programa gdje se može vidjeti je li uređaj u funkciji ili ne.

6. Usporedba rada mrežnih emulatora

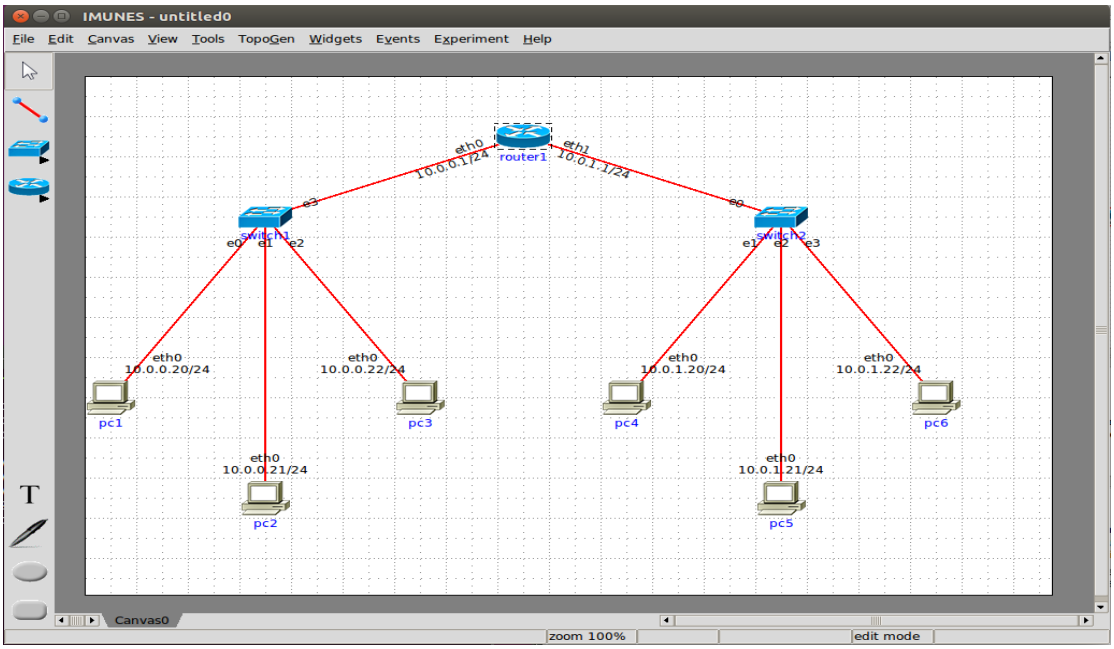
Kako bi se dobio bolji uvid u rad mrežnih emulatora, provedeni su eksperimenti emulacije na istima. No to nije dovoljno za cjelokupnu procjenu pa će se također promatrati jednostavnost korištenja emulatora, zahtjevnost konfiguracije mrežnih uređaja, u kojoj mjeri se zauzimaju određeni resursi računala na kojima se izvodi eksperiment emulacije, parametri koji određuju performanse mreže i sl.

6.1. Eksperiment emulacije unutar IMUNES emulatora

S obzirom na to da je u prethodnom poglavlju prikazan te ukratko objašnjen izgled grafičkog sučelja emulatora IMUNES, započet će se postupak izrade mrežne topologije. Nakon što se razmotre mrežni elementi u alatnoj traci, slijedi odabir te postavljanje istih na radnu površinu emulatora. Postavljene elemente je potrebno povezati kako bi se dobio izgled željene topologije. U ovom radu ta topologija se sastoji od:

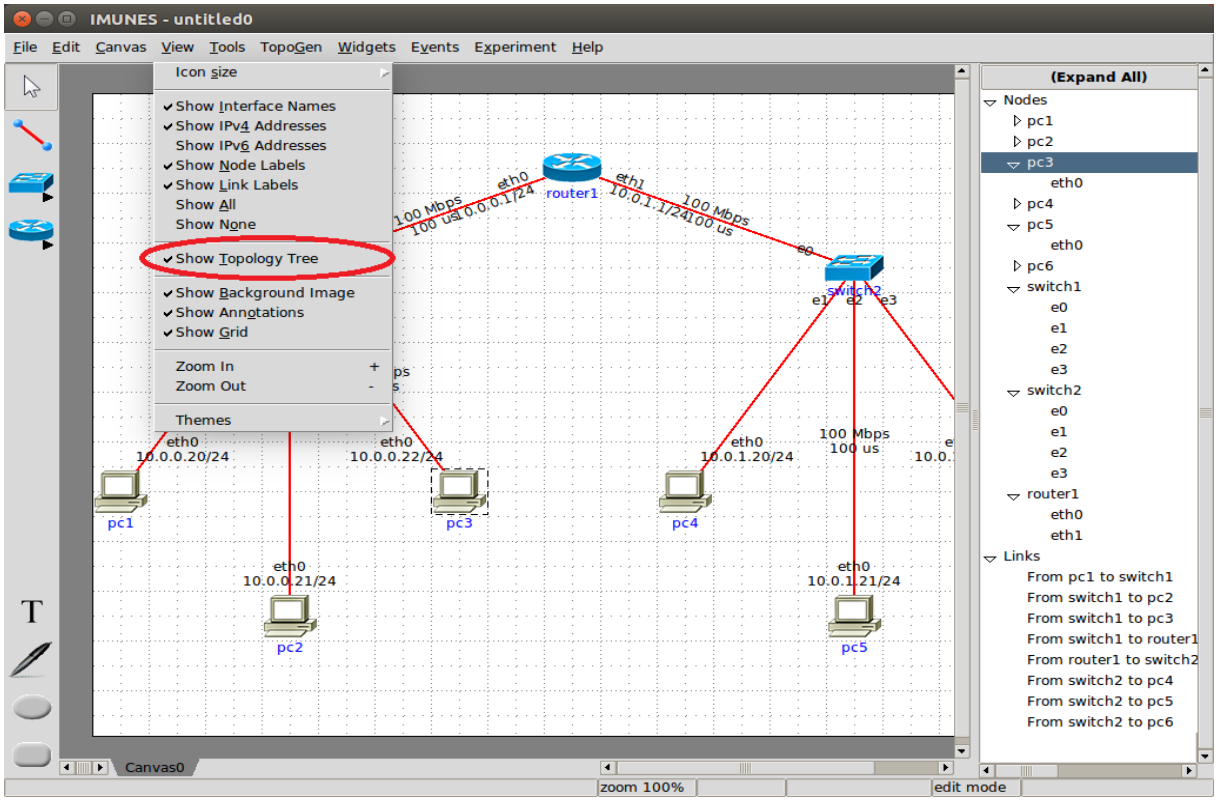
- Šest (6) računala;
- Dva (2) preklopnika te
- Jednog (1) usmjernika.

Projektirana mreža koja je prikazana na slici 14 predstavlja hijerarhijsku mrežnu topologiju gdje je usmjernik glavni centralni čvor i predstavlja najviši sloj, a preklopnik predstavlja srednji sloj koji povezuje usmjernik i računala koja se nalaze u najnižem sloju. Također se može primijetiti kako se radi o dvjema, međusobno povezanim, lokalnim mrežama.



Slika 14. Prikaz složenih i spojenih mrežnih elemenata u IMUNES emulatoru

Odabirom izbornika *View* u alatnoj traci, što je prikazano na slici 15, može se odabrati mogućnost prikaza popisa mrežnih elemenata koji se nalaze na radnoj površini. Također može se vidjeti popis prijenosnih kanala te koje elemente povezuju što dodatno olakšava rad u slučaju da imamo složeniju topologiju.

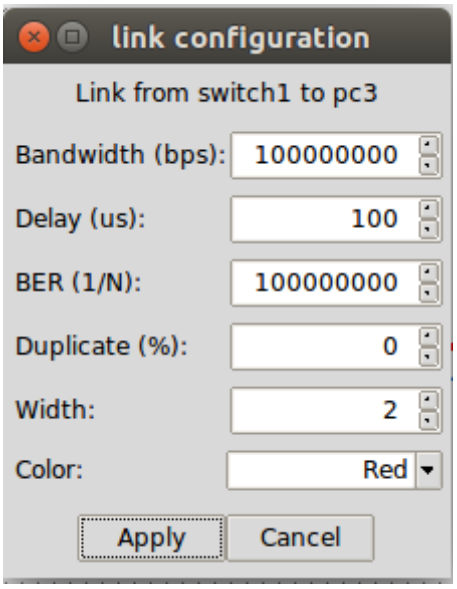


Slika 15. Popis mrežnih uređaja te prijenosnih kanala

Prednost IMUNES emulatora je u tome što su svi uređaji, kao i prijenosni kanali, unaprijed konfigurirani te su automatski spremni za daljnji rad. Naravno postoji i mogućnost za ručnu konfiguraciju i to će u nastavku biti prikazano.

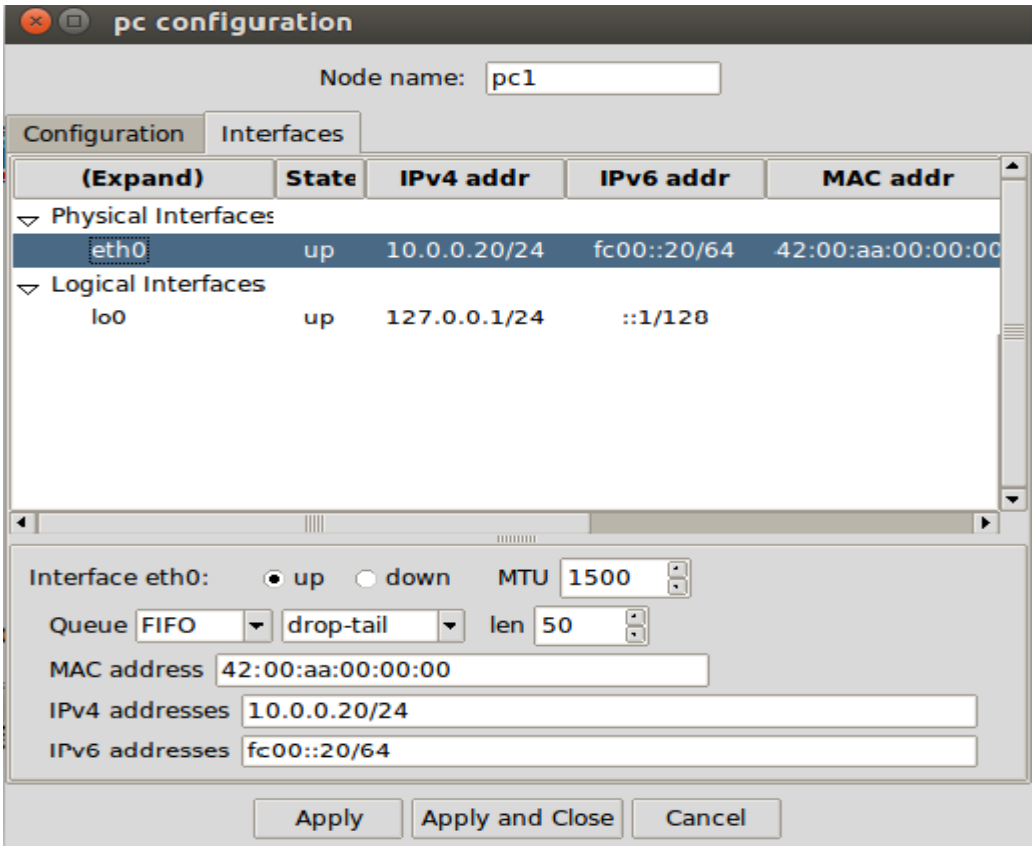
Dvostrukim klikom na prijenosni kanal otvara se dodatni prozor, koji je prikazan na slici 16, sa postavkama konfiguracije prijenosnog kanala. Sve početne vrijednosti su postavljene na nulu i kada bi se izvršio eksperiment emulacije, kašnjenje te gubici poslanih paketa bi iznosili također nula. S obzirom na to da u emulatoru GNS3 nije moguća konfiguracija prijenosnog kanala, a u istom će se koristiti *FastEthernet*[20] *100BASE-T*[21] prijenosni standard koji spada u prijenosne kanale, vrijednosti širine prijenosnog pojasa, kašnjenja te *BER-a*[22] će biti ekvivalentne tom standardu [46] što je također vidljivo na slici 16. Bitno je napomenuti da kod *BER-a* „N“ predstavlja broj prenesenih bita.

[20] **FastEthernet** je naziv za *ethernet* standarde koji omogućuju prijenos podataka unutar lokalne mreže (LAN) pri brzini prijenosa od 100 Mbit/s.
[21] **100BASE-T** je jedan od standarda *fastEtherneta.*
[22] **BER (engl. *Bit error rate*)** je broj pogrešno prenesenih bita u odnosu na ukupno prenesenih bita u nekom periodu. [22]



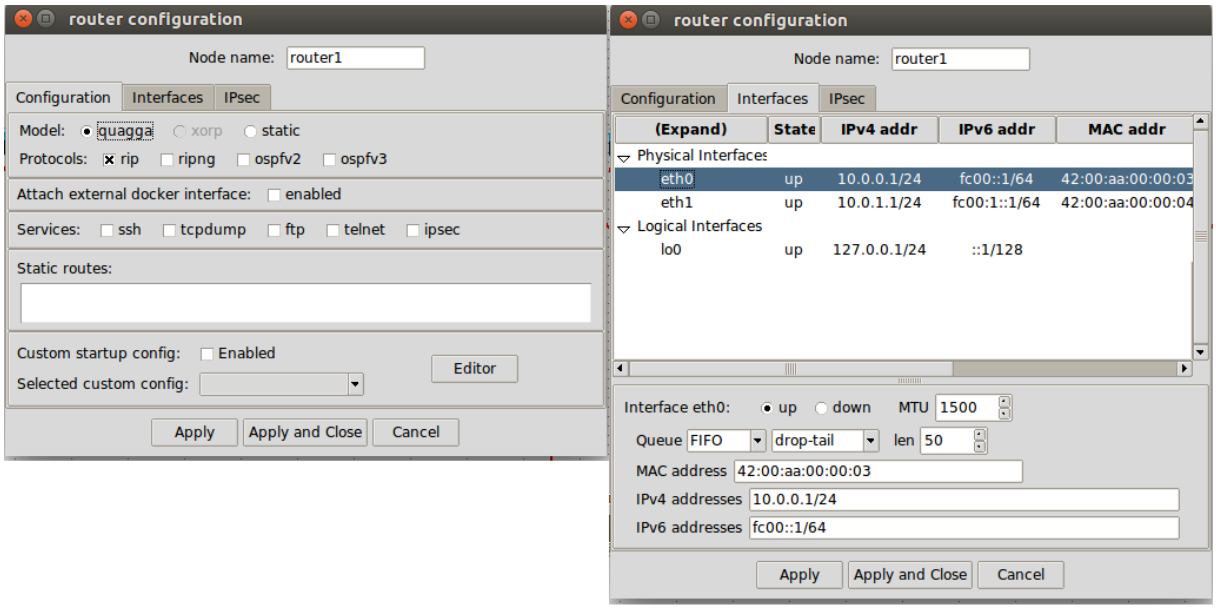
Slika 16. Konfiguracija prijenosnog kanala

Nakon konfiguracije prijenosnog kanala, prelazi se na konfiguraciju računala. No kako je već prije spomenuto, računala su predkonfigurirana pa će se samo izvršiti kratak pregled mogućnosti konfiguracije. Na slici 17 vidimo da su dodijeljene IPv4, IPv6 adrese računala te fizička *MAC* adresa unutar fizičkog sučelja te stanje računala, a na dnu prozora konfiguracije moguća je ručna izmjena istih (naravno, bitno je pripaziti da se ne upiše postojeća IPv4, IPv6 ili MAC adresa nekog drugog računala unutar mreže).



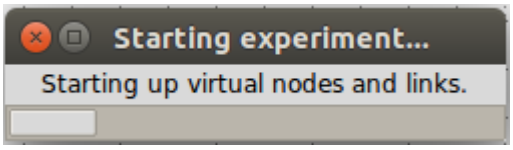
Slika 17. Konfiguracija računala

Nakon računala slijedi pregled te konfiguracija usmjernika. Na izborniku *Configuration* se proizvoljno odabire model usmjeravanja te slijedi odabir protokola koji je integriran u tom modelu. U ovom radu se će se koristiti *RIP* protokol što je i vidljivo na slici 18. Nadalje u izborniku *Interfaces* može se primijetiti dva fizička sučelja s različitim adresama, a ta sučelja predstavljaju ulaze na koje su spojene dvije podmreže te kao i kod računala moguća ručna izmjena adresa.



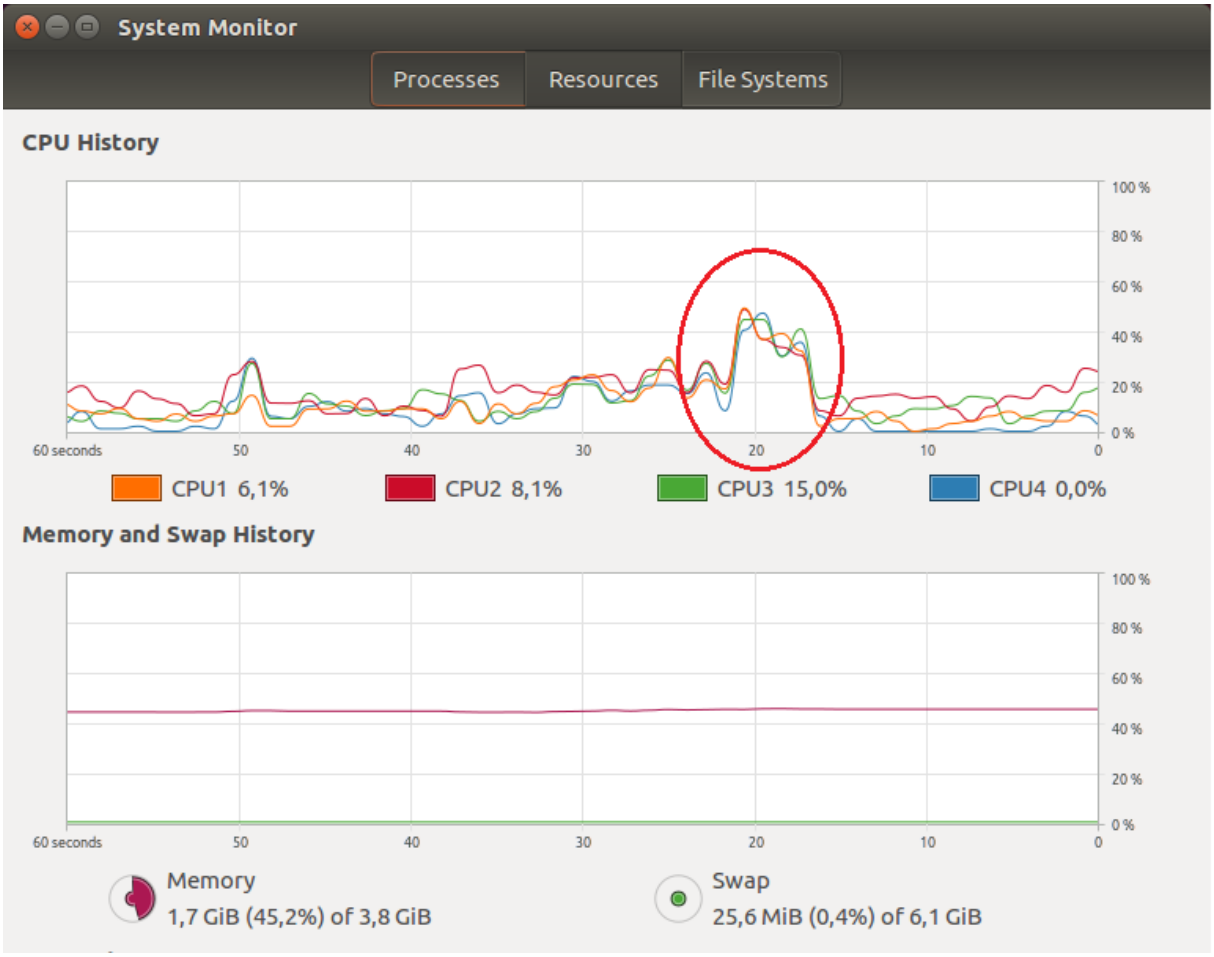
Slika 18. Konfiguracija usmjernika

S obzirom na to da su svi mrežni elementi konfigurirani može se započeti sa eksperimentom emulacije. Odabirom akcije *Execute,* koja se nalazi u izborniku *Experiment*, pokreće se emulacija svih mrežnih elemenata što je vidljivo na slici 19.



Slika 19. Pokretanje eksperimenta emulacije

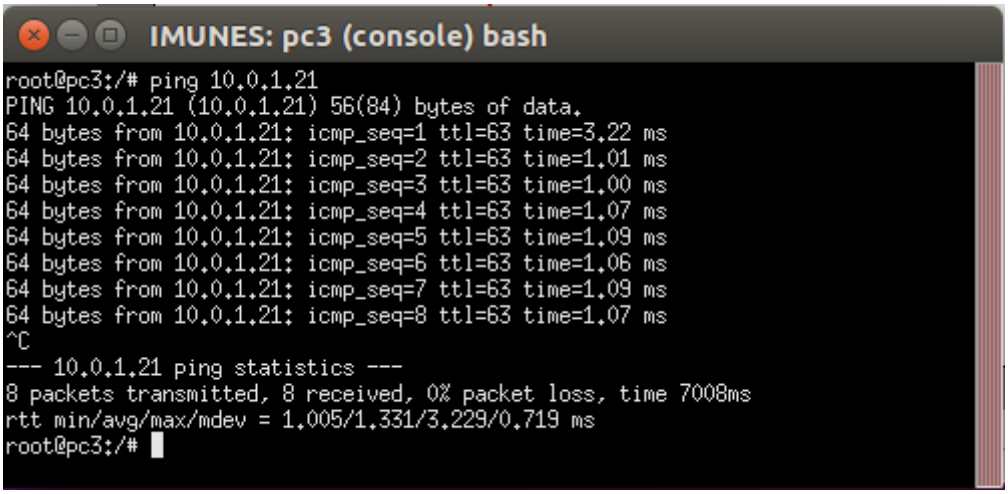
Kao što je već spomenuto, emulacija zauzima određene resurse računala pa je samim time povećan intenzitet rada procesora računala na kojem se izvodi eksperiment emulacije. No zahvaljujući postavkama unutar emulatora, intenzitet rada jezgre procesora u kratkom roku padne i zadržava se na optimalnoj razini. Nagli „skok“ intenziteta rada procesora prilikom pokretanja eksperimenta emulacije vidljiv je na slici 20. Također, može se primijetiti kako korištenje radne memorije ostaje nepromijenjeno.



Slika 20. „Skok“ intenziteta rada procesora računala prilikom pokretanja eksperimenta emulacije

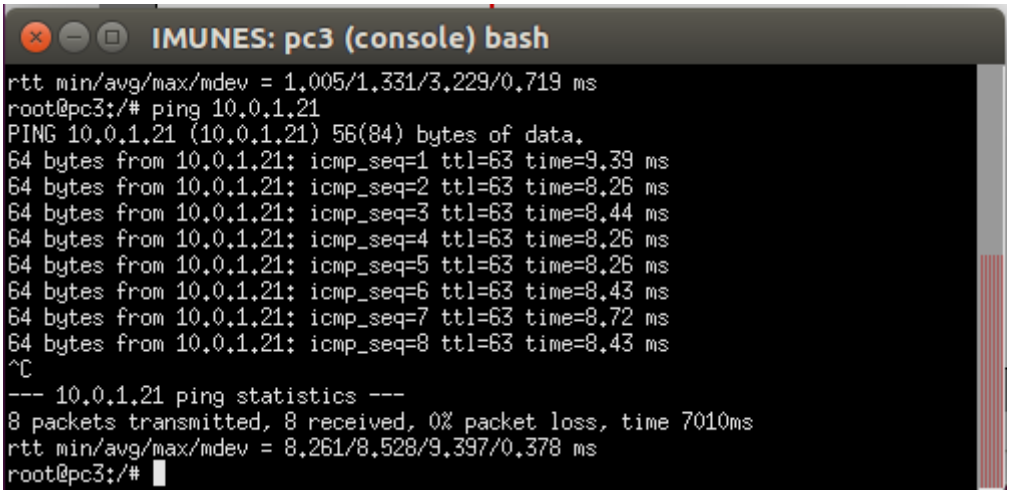
Kako bi se provjerila uspješnost spajanja i konfiguracije elemenata mreže, proizvoljno se odabire jedan od mrežnih uređaja te se automatski pokreće aplikacija *console.* Unutar te aplikacije će se izvršiti slanje kontrolnih paketa što će omogućiti uvid u uspješnost eksperimenta. Sa računala *pc2*, naredbom *ping*[23], poslano je 8 *ICMP*[24] paketa veličine 64 bajta na računalo *pc5* čija je IPv4 adresa: 10.0.1.21 što je prikazano na slici 21.

[23] **Ping** je administrativni alat koji služi za provjeru dostupnosti klijenata u računalnim mrežama koje se temelje na IP protokolu.
[24] **ICMP (engl. *Internet Control Message Protocol*)** predstavlja kontrolni protokol koji se koristi za otkrivanje pogrešaka u računalnim mrežama na način da šalju tzv. ICMP paketi.



Slika 21. Slanje *ICMP* paketa sa računala *pc2* na računalo *pc5*

Može se primijetiti da niti jedan paket nije izgubljen, te kako su kašnjenja relativno mala za što su zaslužni širina prijenosnog pojasa te kašnjenje koji su definirani u konfiguraciji prijenosnog kanala.



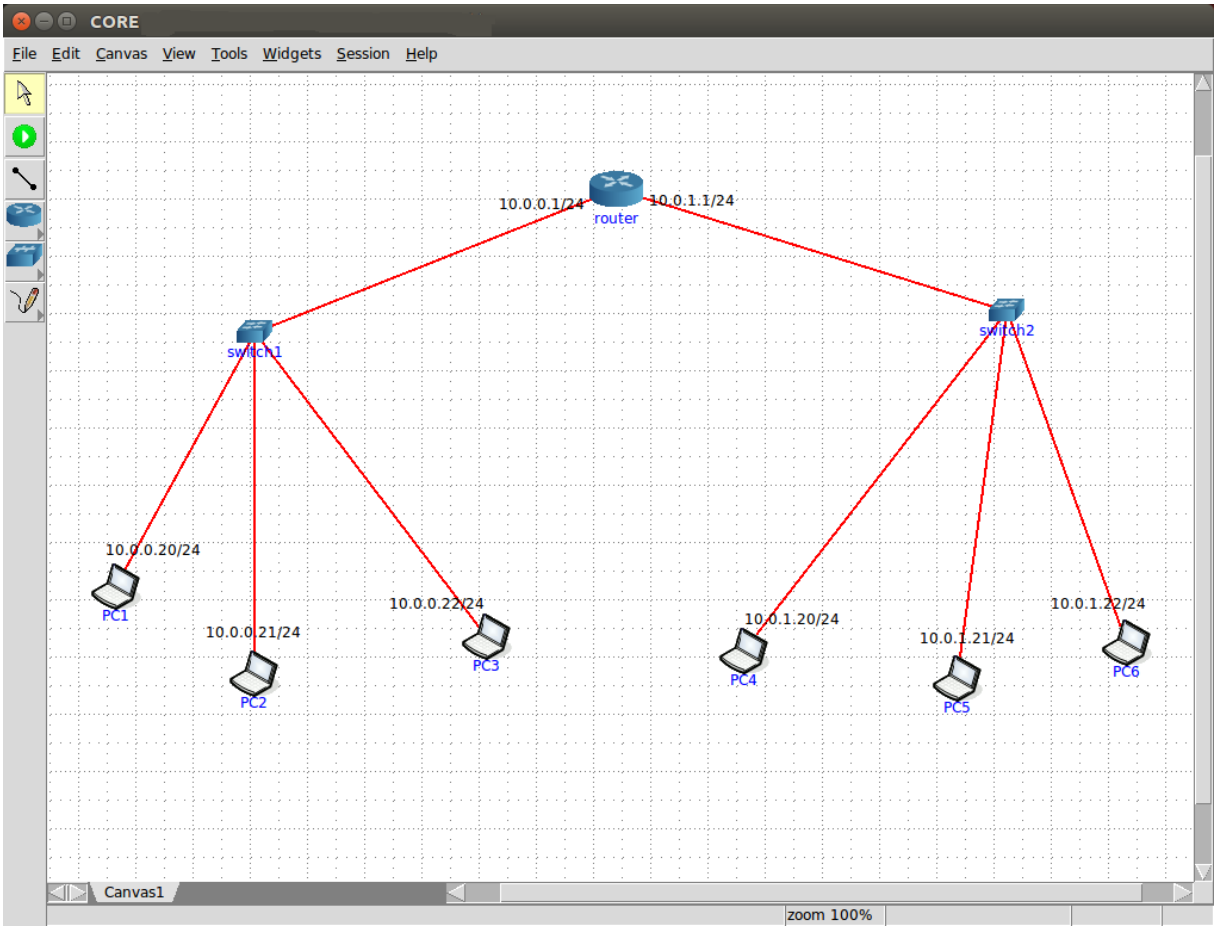
Slika 22. Ponovljeni postupak slanja *ICMP* paketa s drugačijom konfiguracijom prijenosnog kanala

Za bolji uvid u „stvarnost“ rezultata, unutar konfiguracije prijenosnog kanala povećano je kašnjenje sa 100μs na 1ms te je na slici 22 prikazan utjecaj istog na kašnjenje paketa te je vidljivo kako se kašnjenje povećalo.

6.2. Eksperiment emulacije unutar CORE emulatora

S obzirom na to da je CORE emulator temeljen na otvorenom kodu IMUNES emulatora, neke postavke i konfiguracije mrežnih elemenata su poprilično slične. Ipak postoje razlike pa će se postupak konfiguracije i emulacije prikazati u nastavku. Topologija koja će se projektirati je ista kao i u IMUNES emulatoru, a isto tako će biti i u GNS3 emulatoru. U principu, način postavljanja elemenata je isti, no za razliku od IMUNES emulatora, CORE emulator nema mogućnost prikaza postavljenih mrežnih elemenata na radnu površinu što je vidljivo na slici 23. No zato u alatnoj traci, pod

izbornikom *tools*, može se odabrati željena mrežna topologija te će je program, nakon ručno postavljenih elemenata na radnu površinu, posložiti kako je korisnik zadao.

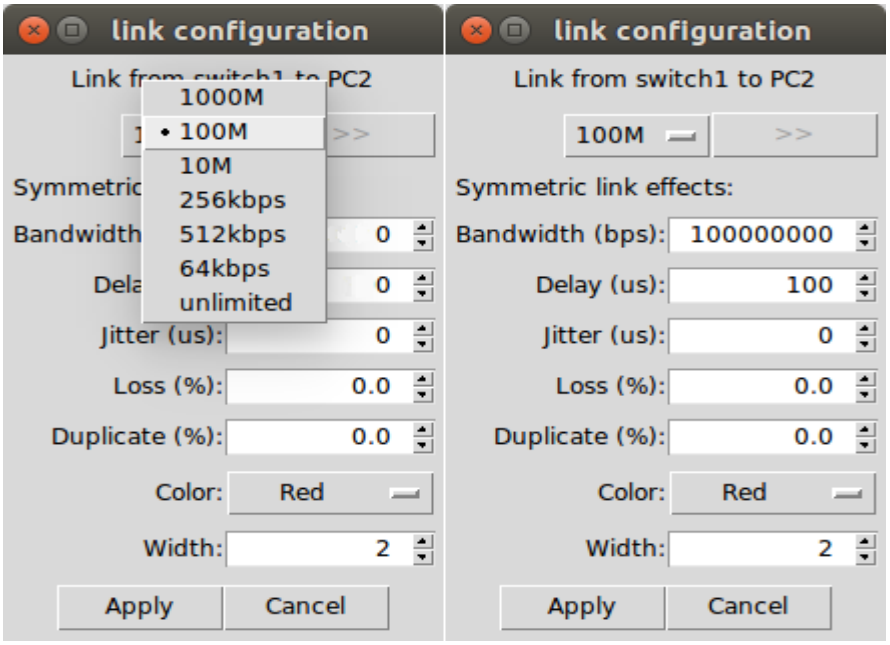


Slika 23. Prikaz složenih i spojenih mrežnih elemenata u CORE emulatoru

Kao što je već spomenuto, mrežni elementi su predkonfigurirani, no u nastavku će se prikazati postupci konfiguracije istih.

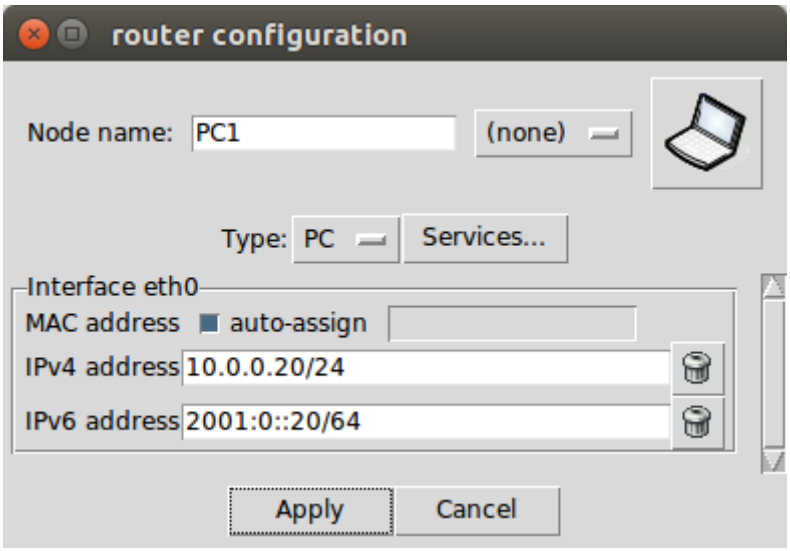
Postupak konfiguracije će se započeti od prijenosnog kanala što je vidljivo na slici 24. Primjećuje se kako postoje razlike u konfiguraciji prijenosnog kanala između IMUNES i CORE emulatora. U CORE konfiguraciji prijenosnog kanala moguć je odabir standarda prijenosnog kanala, nakon kojeg aplikacija sama definira parametre prijenosa. U ovom slučaju odabran je standard *100M* koji predstavlja *fastEthernet* standard. Također se može primijetiti kako uz kašnjenje i širinu prijenosnog pojasa te gubitke paketa, koji su izraženi u postocima, konfiguracijski prozor sadrži i varijacije kašnjenja[25] (engl. *jitter*).

[25] **Varijacije kašnjenja** nastaju zbog varijacija u međudolaznim vremenima paketa iste sesije. [22]



Slika 24. Konfiguracija prijenosnog kanala

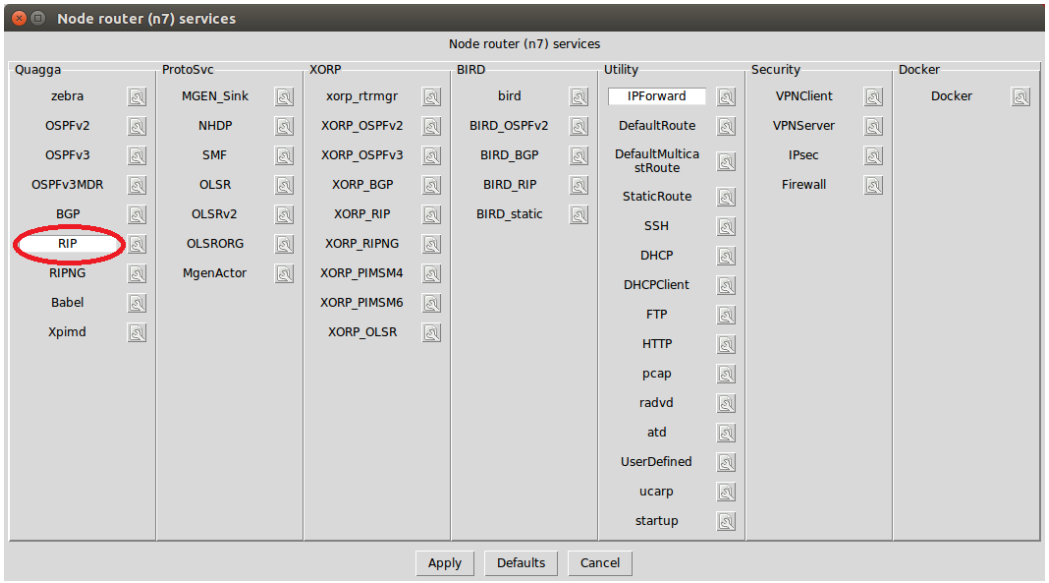
Nakon prikaza konfiguracije prijenosnog kanala slijedi prikaz konfiguracije računala. Na slici 25 može se primijetiti kako naziv konfiguracijskog prozora nije *PC configuration* već *router configuration* što znači da emulator sve elemente gleda kao isti mrežni element, ali korisnik može proizvoljno mijenjati oblik simbola kao i vrstu mrežnog elementa što u IMUNES emulatoru nije moguće. No princip mijenjanja i dodjele adresa računala ostaje isti pa se može reći kako su po tom pitanju, konfiguracije emulatora CORE i IMUNES, međusobno ekvivalentni.



Slika 25. Konfiguracija računala

S obzirom na to da su prozori konfiguracije isti za sve elemente, za usmjernik će se prikazati samo odabir protokola usmjeravanja koji će biti isti kao i u CORE emulatoru.

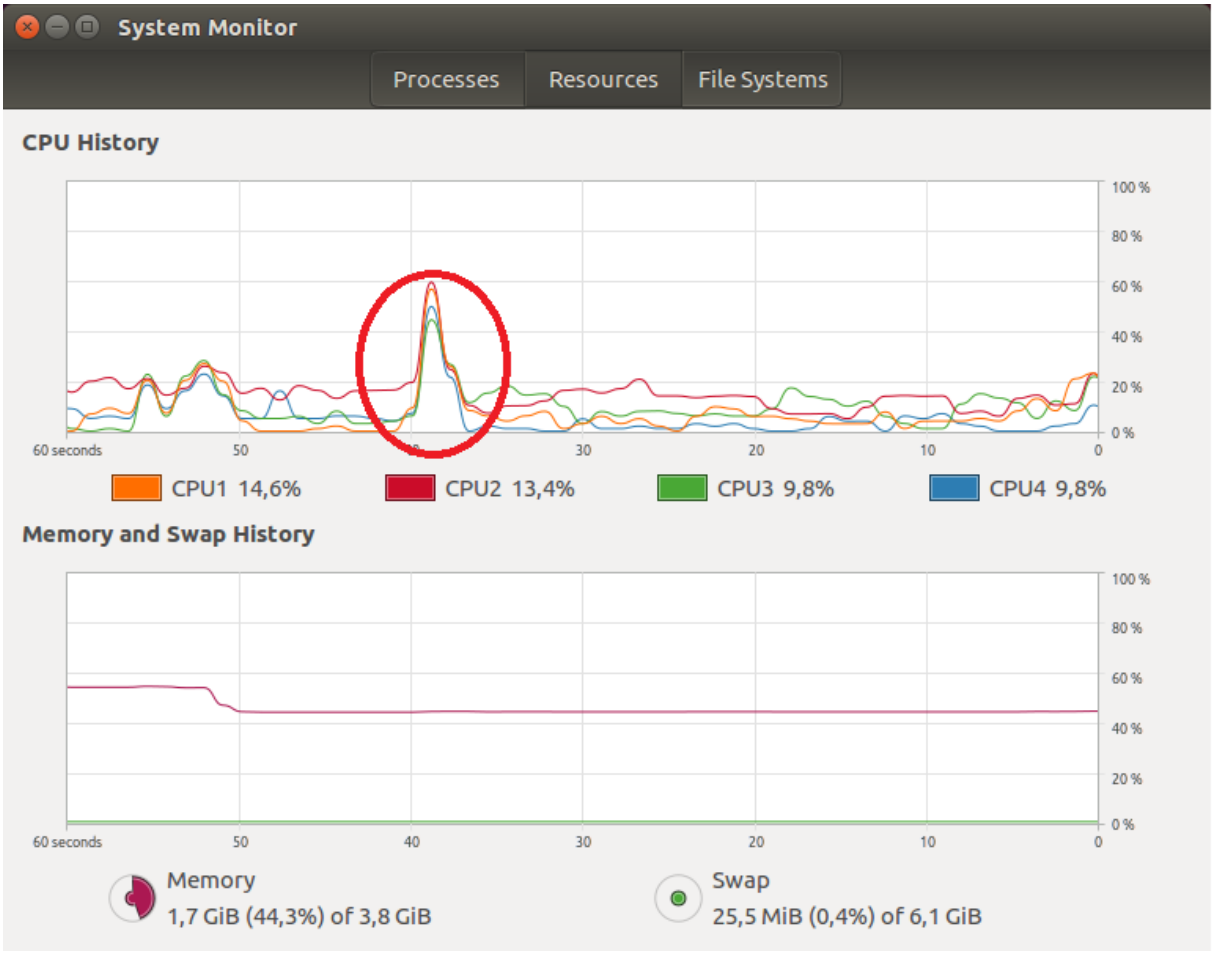
Na slici 26 može se primijetiti poveći broj mogućnosti odabira protokola, modela te ostalih funkcionalnosti koje konfiguracijski prozor nudi. Kao što je spomenuto odabran je *RIP* protokol.



Slika 26. Odabir protokola za usmjernik

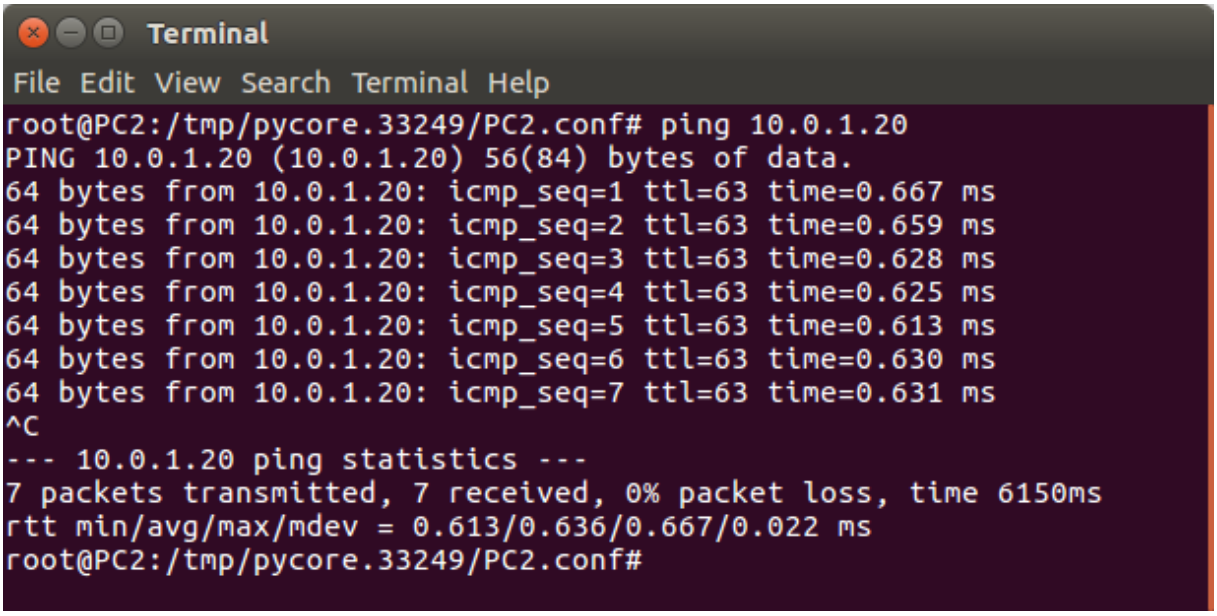
Klikom na gumb *Start* započinje se eksperiment emulacije mrežnih elemenata. U slučaju bilo kakve greške prilikom emuliranja elemenata ili prethodne konfiguracije prikazat će se simbol naziva *CEL (*engl. *Check Emulation Light*) koji predstavlja dijaloški okvir koji sadrži popis grešaka primljenih od CORE emulatora. Greške koje se nalaze na popisu sadrže vrijeme, razinu težine, broj čvora i izvor. Treperenje CEL simbola ukazuje na jednu ili više kobnih grešaka što znači da nije moguće emulirati jedan ili više mrežnih elemenata stoga je potrebno provesti ponovni postupak konfiguracije istih.

Za razliku od IMUNES emulatora, prilikom eksperimenta emulacije u CORE emulatoru, intenzitet rada procesora je malo drugačiji. Primjetno je da se puno više zauzimaju resursi te da je „skok“ intenziteta rada procesora viši i iznosi oko 60 posto. No isto tako, period trajanja „skoka“ je dosta kraći i iznosi svega par sekundi što je vidljivo na slici 27.



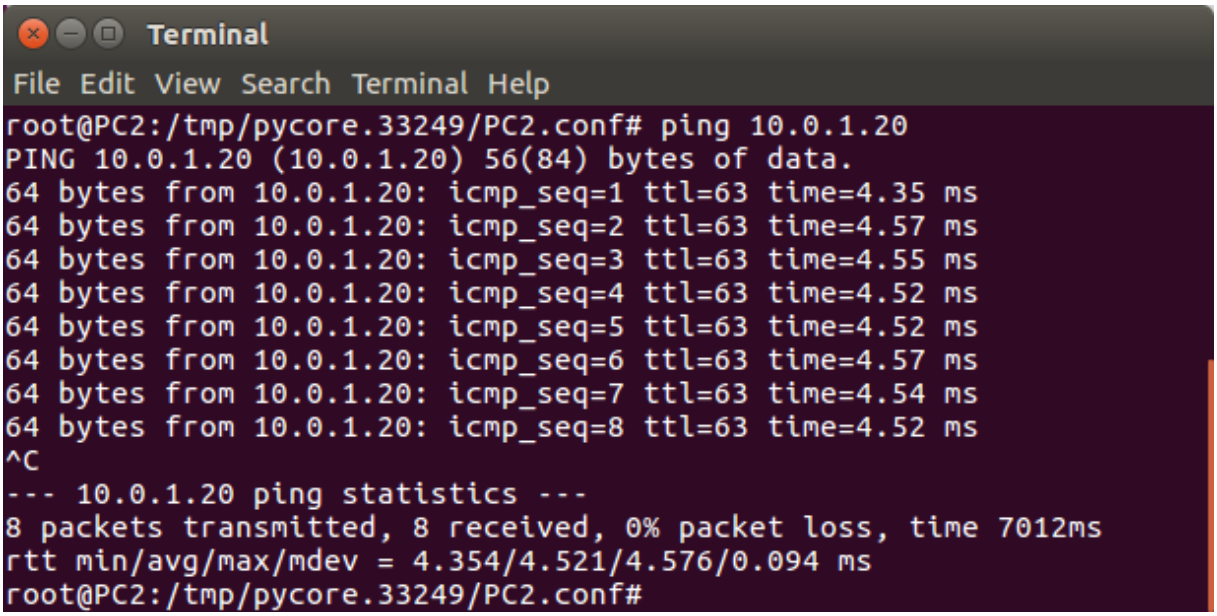
Slika 27. „Skok“ intenziteta rada procesora računala prilikom pokretanja eksperimenta emulacije

Kao i u IMUNES emulatoru, provjeriti će se uspješnost povezanosti mrežnih uređaja. Na slici 28 je vidljivo kako CORE emulator nema vlastitu *Console* aplikaciju već koristi *Terminal* aplikaciju *Linux* operativnog sustava. Obavlja se slanje 7 kontrolnih paketa veličine 64 bajta naredbom *ping* sa računala *PC2* prema računalu *PC4 čija je IPv4 adresa: 10.0.1.21.* Primjećuje se da niti jedan paket nije izgubljen te kako su kašnjenja manja nego kašnjenja u IMUNES emulatoru.



Slika 28. Slanje *ICMP* paketa sa računala *PC2* na računalo *PC4*

Nakon ponovljene konfiguracije prijenosnog kanala povećano je kašnjenje sa 100μs na 1ms te je na slici 29 prikazan ponovljeni postupak slanja kontrolnih paketa u svrhu boljeg uvida u rezultate emulacije.



Slika 29. Ponovljeni postupak slanja *ICMP* paketa s drugačijom konfiguracijom prijenosnog kanala

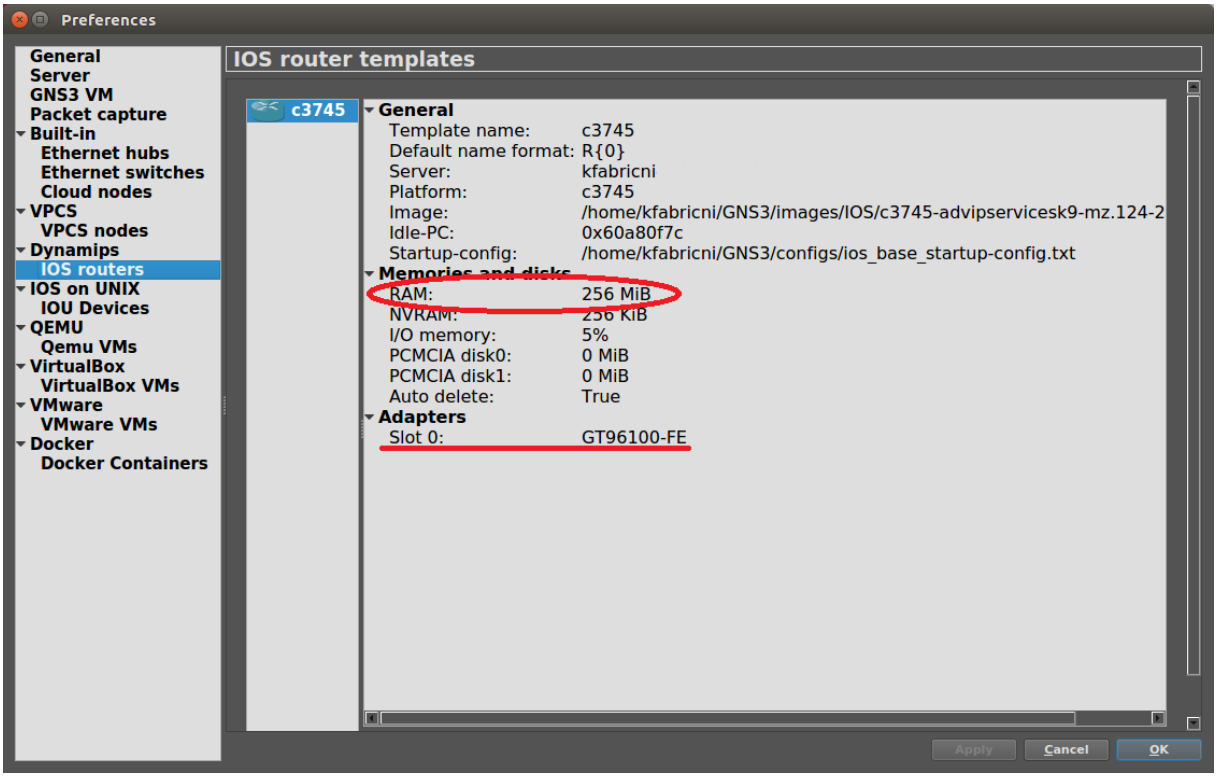
Može se primijetiti kako je kašnjenje poraslo, ali ne tolikim intenzitetom koliko u IMUNES emulatoru.

6.3. Eksperiment emulacije unutar GN3 emulatora

Grafički dizajn GNS3 emulatora se na prvu čini dosta različit od prethodna dva emulatora, no daljnjim razmatranjem se primjećuje kako i nije baš tako. No bitne razlike

se vide u korištenju i konfiguraciji mrežnih elemenata, te ostalim mogućnostima samog emulatora.

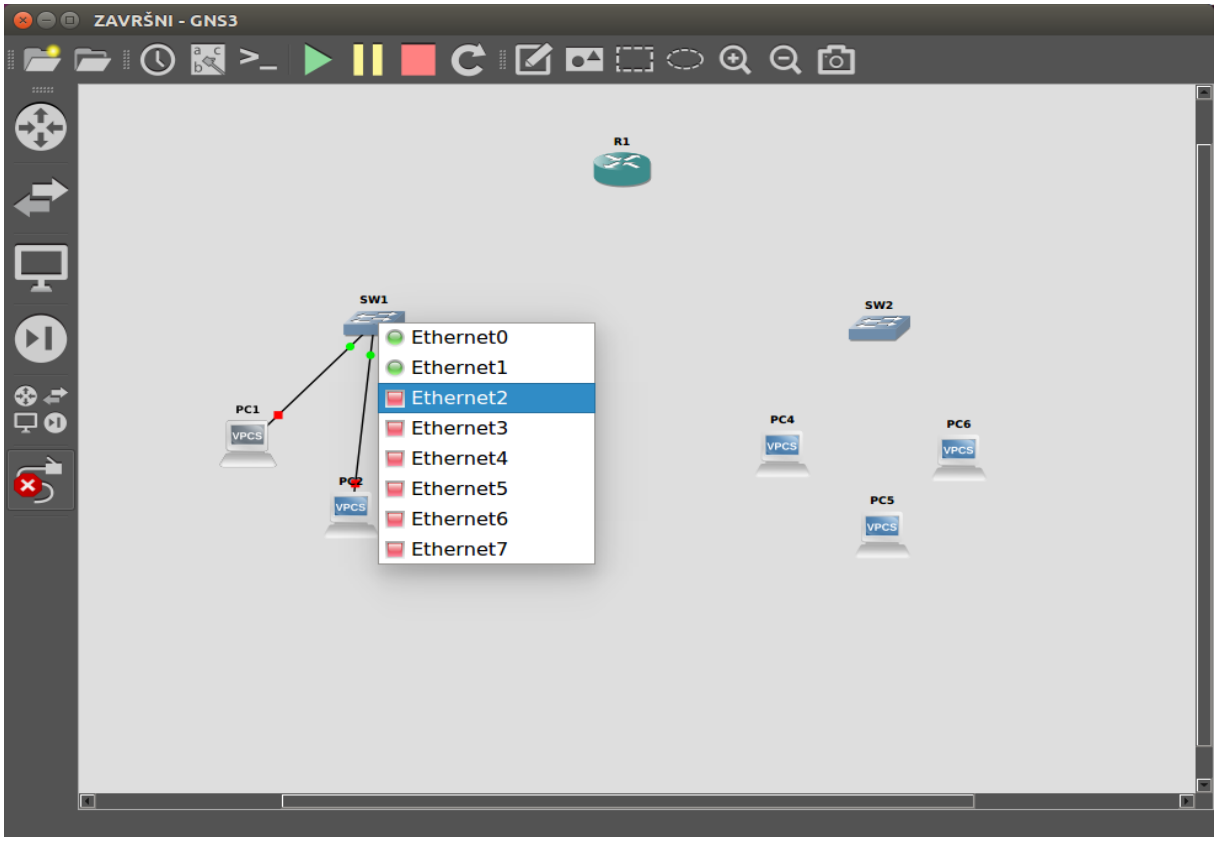
U ovom slučaju se neće prvo započeti s izradom mrežne topologije jer je, za projektiranje iste, potrebno prvo instalirati usmjernik. S obzirom na to da uz GNS3 emulator nisu integrirani usmjernici, potrebno je iste preuzeti s Interneta. Nakon preuzimanja u izborniku *Edit* odabire se opcija *Preferences* zatim se otvara prozor koji je prikazan na slici 30 te je potrebno kliknuti na opciju *IOS Routers*. Odabire se opcija *New* te nakon uspješne instalacije usmjernik je spreman za konfiguraciju i daljnji rad.



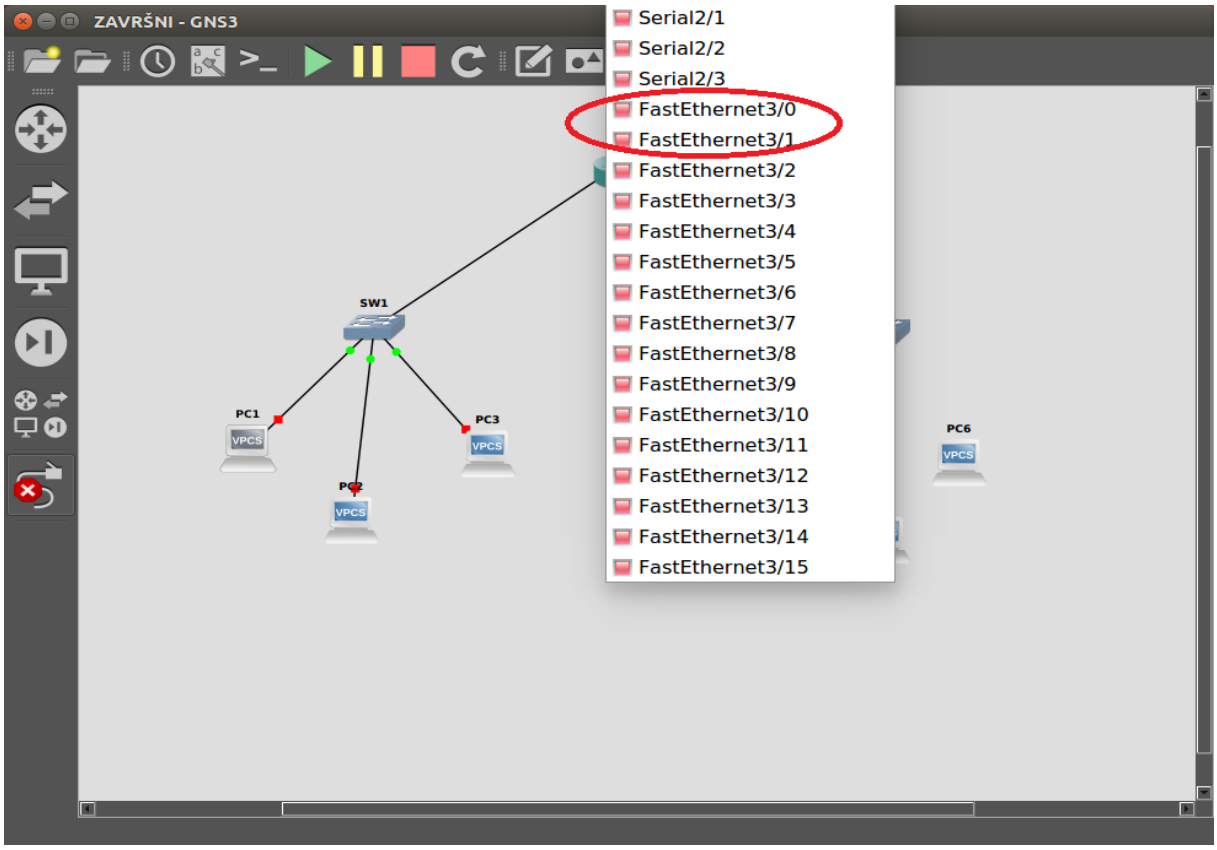
Slika 30. Prikaz uspješno instaliranog usmjernika

Može se primijetiti kako, za razliku od prethodnih emulatora, GNS3 nudi i opis usmjernika kao i mogućnost odabira određenih fizičkih sučelja (pod nazivom *adapters*). Osim toga, za vrijeme instalacije usmjernika je bilo potrebno dodijeliti određeni dio radne memorije, što znači da će se za vrijeme emulacije koristiti resursi iste.

Odabir elemenata, koji će se koristiti u projektu, moguć je na lijevoj strani grafičkog sučelja. Proizvoljnim odabirom postavljaju se elementi na radnu površinu te se pomoću opcije *link* međusobno povezuju. Nakon aktivacije opcije *linka* odabiremo svaki uređaj zasebno te klikom na isti prikazuju se ulazi istog što je vidljivo na slici 31. Kako je već spomenuto GNS3 nema mogućnost konfiguracije linka, no parametre koje smo postavljali u prethodnim emulatorima, GNS3 generira odabirom određenog fizičkog sučelja na usmjerniku stoga je odabrano *FastEthernet* sučelje, što je vidljivo na slici 32 kako bi se, usporedbe radi, imalo iste uvjete rada za vrijeme eksperimenta emulacije.

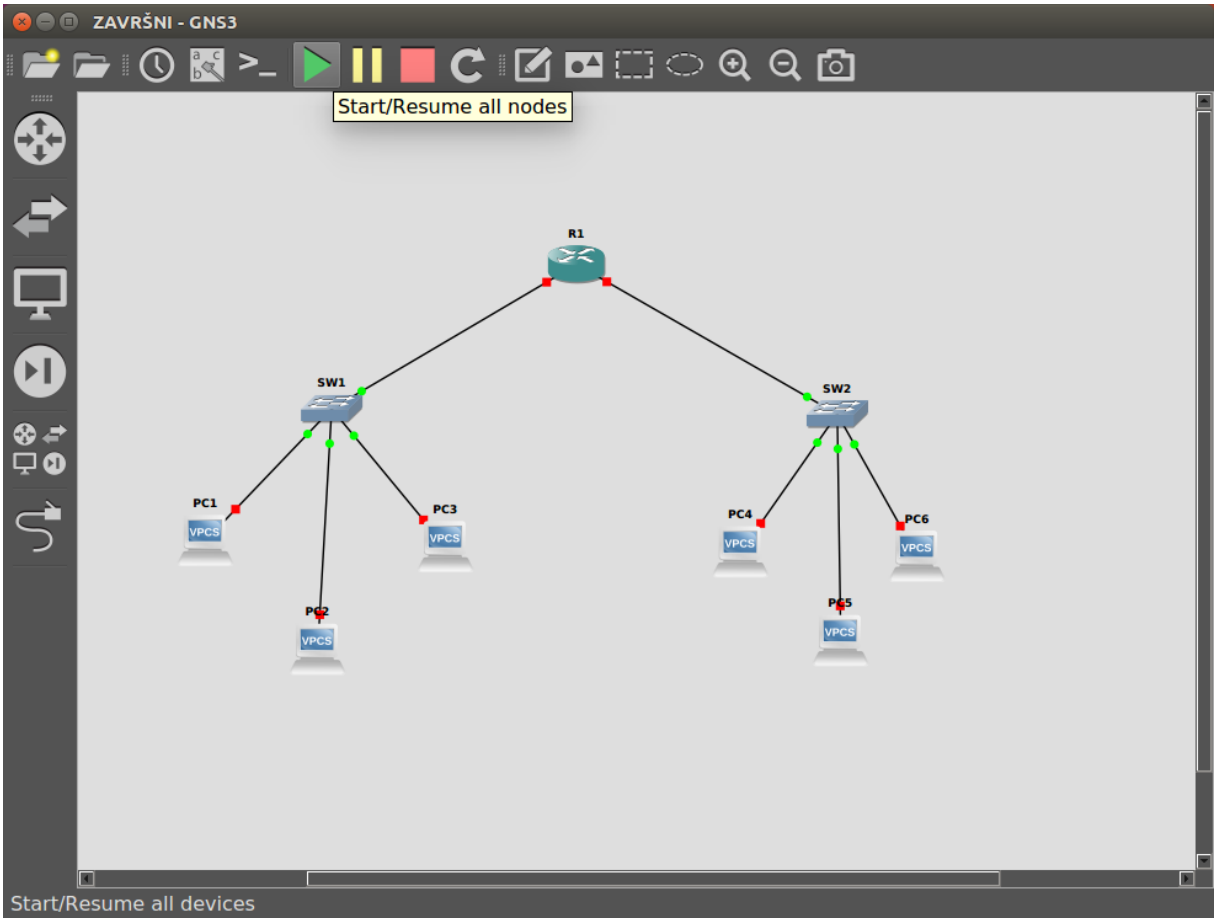


Slika 31. Odabir slobodnih ulaza na mrežnim uređajima



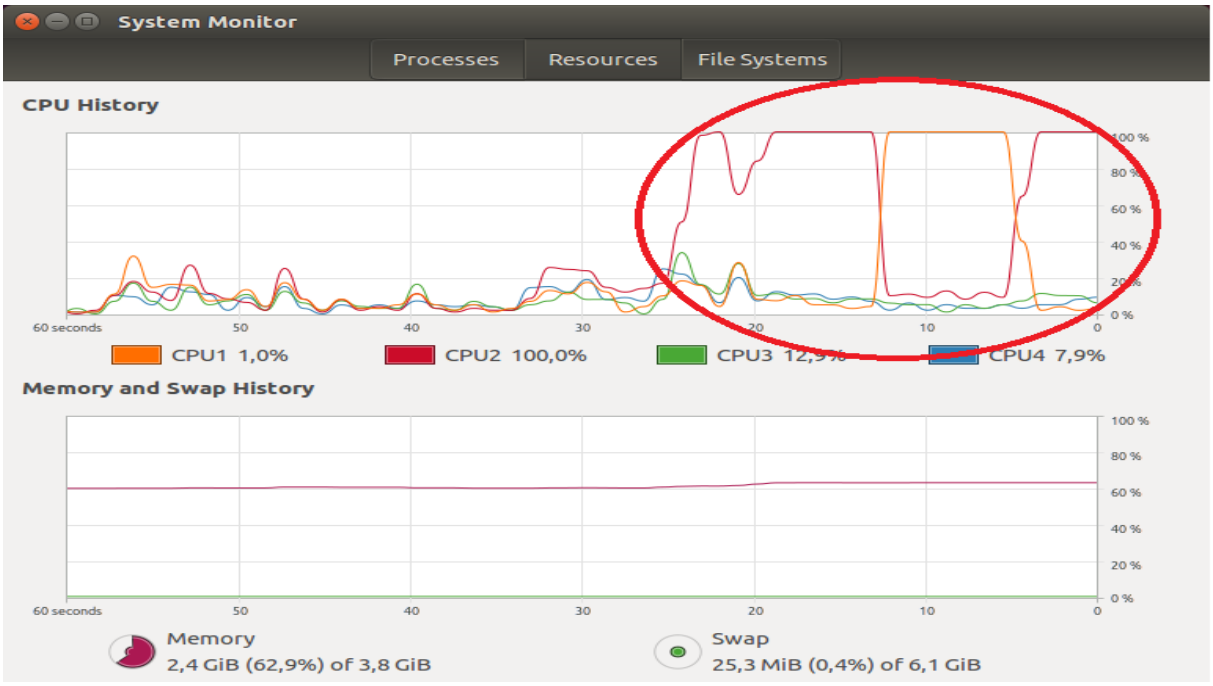
Slika 32. Odabir *FastEthernet* sučelja na usmjerniku

Nakon što su svi elementi povezani, dobiva se mrežna topologija kao i u prethodnim emulatorima te se odabirom opcije *start* započinje eksperiment emulacije što je prikazano na slici 33.



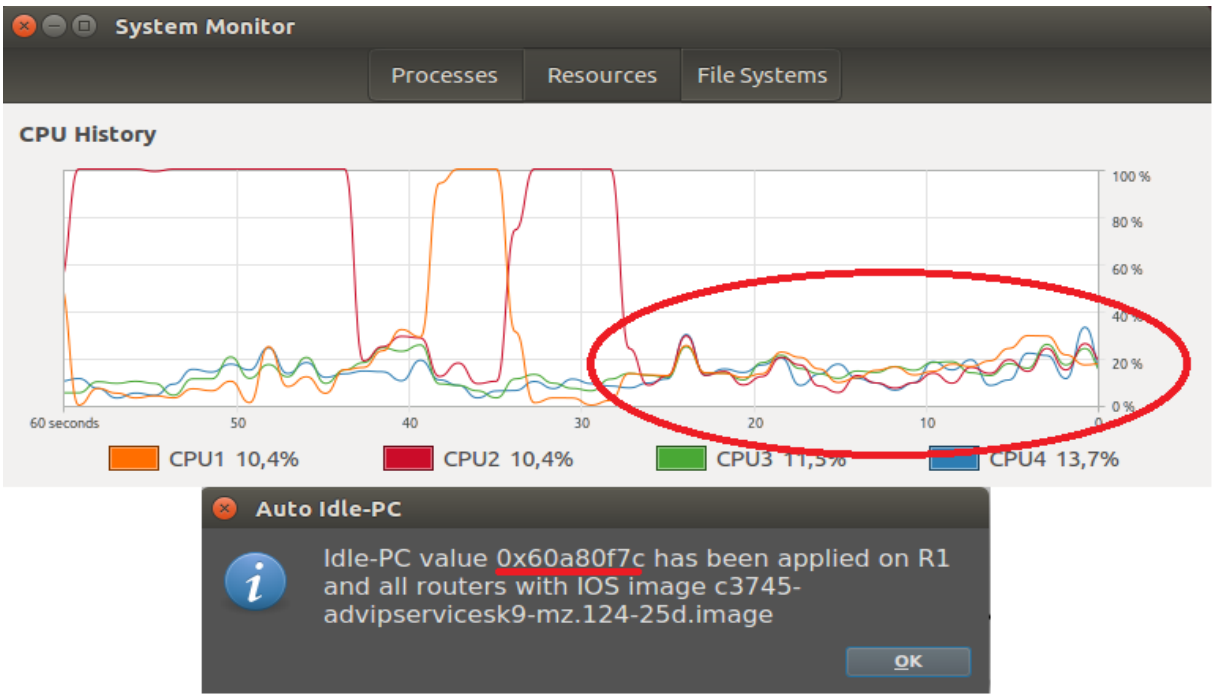
Slika 33. Prikaz složenih i spojenih mrežnih elemenata u GNS3 emulatoru

Nakon što je pokrenut eksperiment emulacije na slici 34 se može primijetiti jedna od karakteristika usmjernika GNS3 emulatora, a to je veliki intenzitet korištenja procesora koji, u određenim trenutcima, iznosi 100%. Također je vidljiv i blagi porast u zauzimanju radne memorije no to je bilo očekivano s obzirom na prethodnu konfiguraciju prilikom instalacije usmjernika.



Slika 34. Prikaz visokog intenziteta korištenja procesora

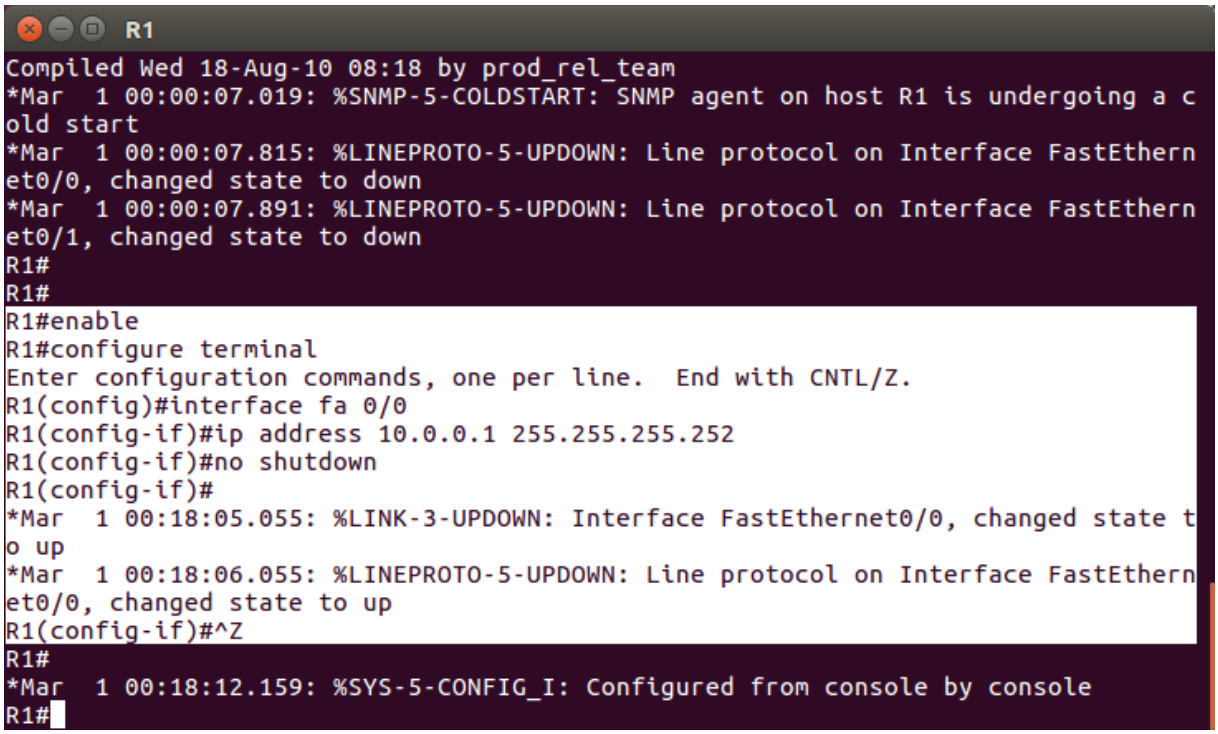
Za razliku od prethodnih emulatora, ovakvo intenzivno korištenje traje sve dok se usmjernik ne postavi u stanje *Idle-PC.* Odabirom ove opcije, postavljaju se vrijednosti određenih parametara koje kontroliraju stopu uzoraka emuliranu od strane GNS3 programa. Nakon odabira *Auto Idle-PC* na slici 35 se vidi vrijednost parametra koji je primijenjen na usmjerniku te se vidi i pad intenziteta korištenja procesora.



Slika 35. Prikaz *Idle-PC* parametra te prikaz smanjenja intenziteta korištenja procesora

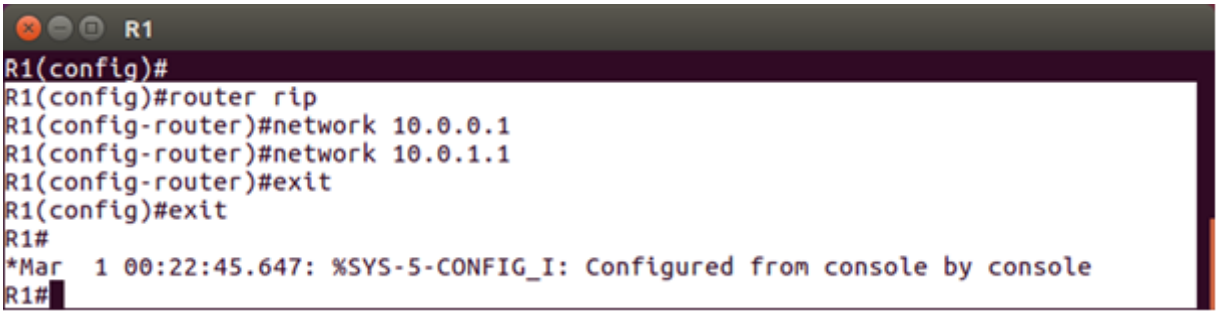
Iako je eksperiment emulacije uspješan i svaki mrežni element radi, to nužno ne znači da je moguća komunikacija između istih. Za razliku od IMUNES i CORE emulatora, GNS3 nema predkonfigurirane mrežne uređaje pa je iste potrebno konfigurirati. Bitno je napomenuti kako postupak konfiguracije odabranog mrežnog uređaja nije moguć sve dok se isti ne emulira.

Pomoću aplikacije *Console* konfigurirat će se usmjernik što će biti prikazano na slici 36. Kod prvog korištenja usmjernika potrebno je upisati naredbu *enable* kako bi se omogućila konfiguracija usmjernika. Slijedi naredba *interface* uz koju se upisuje fizičko sučelje na koje su mrežni uređaji spojeni. U ovom slučaju to su *FastEthernet FA* 0/0 i *FA* 0/1. Nadalje se upisuje proizvoljna *IPv4* adresa fizičkog sučelja, te unosom naredbe *no shutdown* usmjernik se postavlja u aktivno stanje. Kombinacijom tipki *CTRL+Z* konfiguracija se integrira u usmjernik.



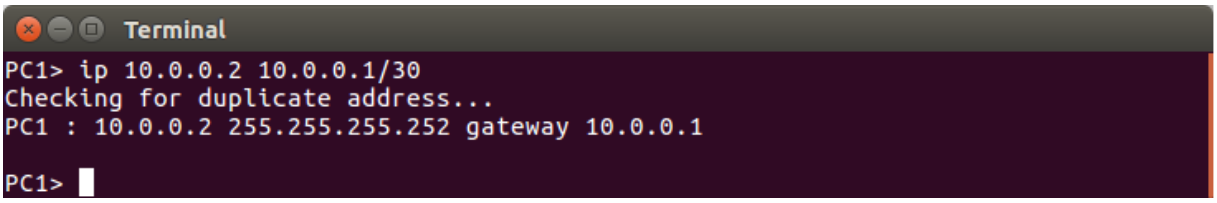
Slika 36. Prikaz uspješne konfiguracije usmjernika u aplikaciji *Console*

Također je potrebno postaviti *RIP* protokol radi usmjeravanja paketa unutar mreže. Kako bi se *RIP* protokol integrirao u usmjernik, u aplikaciju *Console* se upisuje naredba *configure terminal* koja omogućuje konfiguraciju usmjernika. Nadalje, naredbom *router rip* se ulazi u konfiguracijski način rada *RIP* protokola te upisom *network* naredbi navode se adrese svih fizičkih sučelja koje pripadaju usmjerniku. Time je završena konfiguracija RIP protokola na usmjerniku što je vidljivo na slici 37 te je usmjernik sada spreman za komunikaciju u mreži.



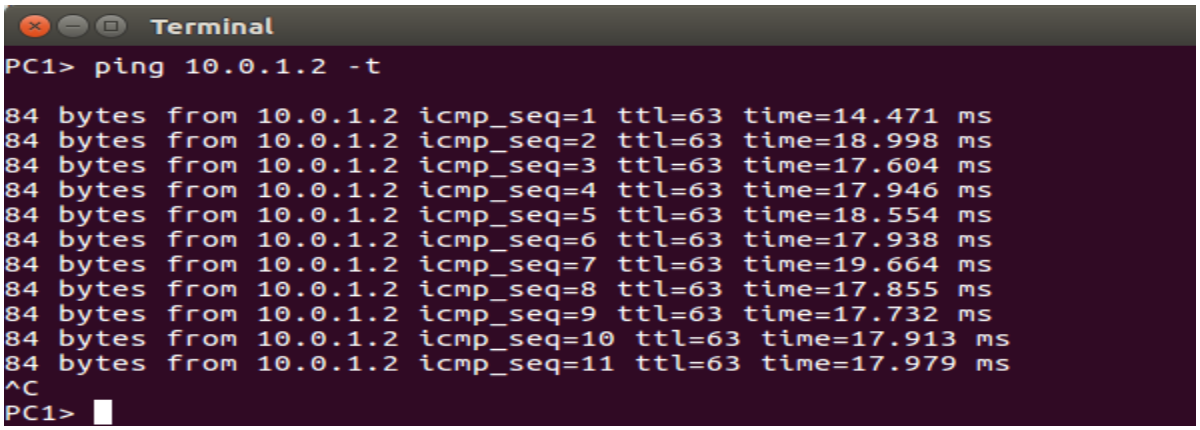
Slika 37. Prikaz uspješno integriranog *RIP* protokola

Uz usmjernik, potrebna je i konfiguracija računala koja se emuliraju. Postupak konfiguracije se također vrši unutar aplikacije *Console*, no za razliku od usmjernika, proces je poprilično jednostavan što je vidljivo na slici 38. Upisuje se naredba *ip* te proizvoljna *IPv4* adresa. Dodatna adresa koja se upisuje je adresa *Default Gateway-a* ili adresa fizičkog sučelja usmjernika na kojeg je računalo spojeno.



Slika 38. Prikaz uspješno konfiguriranog računala u aplikaciji *Console*

S obzirom da nije moguće konfigurirati prijenosne kanale, biti će proveden samo jedan test provjere uspješnosti povezanosti mrežnih uređaja. Test se također provodi u aplikaciji *Console* te se obavlja slanje 11 kontrolnih paketa veličine 84 bajta (što je za 20 bajta veće nego u IMUNES I CORE emulatorima) naredbom *ping* sa računala *PC1* prema računalu *PC6 čija je IPv4 adresa: 10.0.1.2.*



Slika 39. Slanje *ICMP* paketa sa računala *PC1* na računalo *PC6*

Za razliku od IMUNES te CORE emulatora, GNS3 ne dostavlja statistiku poslanim kontrolnim paketima pa nije moguć uvid u broj mogućih izgubljenih paketa. Na slici 39 je primjetno kako su kašnjenja puno veća nego u prethodna dva emulatora.

Za kraj ovog poglavlja, u tablici 3, prikazane su i uspoređene mogućnosti, jednostavnost korištenja te ostali kriteriji emulatora koji su se analizirali kroz poglavlje.

Tablica 3. Usporedba mrežnih emulatora prema kriterijima

	IMUNES	CORE	GNS3
Jednostavnost korištenja	**Visoka**	**Visoka**	**Srednja**
Integrirani mrežni uređaji	**Da**	**Da**	**Uglavnom da**
Zahtjevnost konfiguracije mrežnih elemenata	**Niska**	**Niska**	**Srednja**
Mogućnost konfiguracije prijenosnih kanala	**Da**	**Da**	**Ne**
Intenziteti zauzimanja resursa računala na kojem se izvršava eksperiment emulacije	**Srednji**	**Srednji**	**Visoki**
Gubici paketa za definiranu konfiguraciju mrežnih elemenata	**Nema**	**Nema**	**Nije poznato**
Kašnjenja paketa za definiranu konfiguraciju mrežnih elemenata	**Jako niska**	**Niska**	**Srednja**
Integrirana aplikacija za korištenje i konfiguraciju mrežnih elemenata	**Da**	**Ne**	**Da**

Prilagođeno prema izvoru [23]

U tablici 3 može se primijetiti poprilična sličnost IMUNES i CORE emulatora jer su razlike vidljive samo u vremenima kašnjenja paketa te postojanosti aplikacija za konfiguraciju mrežnih elemenata. Također, vidljivo je kako se GNS3 emulator razlikuje skoro u svim kriterijima od prethodna dva emulatora.

7. Zaključak

U početnom dijelu rada detaljno opisanom klasifikacijom računalnih mreža pojašnjene su trenutne razlike između istih. Također, daljnjim razmatranjem arhitekture računalne mreže i komponenata koje mreža koristi za povezivanje, dolazi se do zaključka kako je kompleksnost mreže vrlo visoka. No zahvaljujući razvitku mrežnih topologija te mrežnih modela, koji definiraju način prijenosa u mreži, moguće je lakše razumijevanje kompleksnosti računalne mreže.

Nadalje, u radu su objašnjene tehnike i komponente pomoću kojih je moguć „privid“ stvarnosti u smislu projektiranja računalnih mreža te ispitivanja rada komponenata i elemenata istih. Mrežni emulatori su te komponente koje omogućuju eksperiment emulacije mrežnih elemenata te omogućuju projektiranje računalnih mreža. Također provedena je klasifikacija istih te se može zaključiti kako postoji širok spektar mrežnih emulatora.

U ovom radu odabrani su istraživački emulatori mrežnih topologija *Integrated Multiprotocol Network Emulator/Simulator, Common Open Research Emulator* te *Graphical Network Simulator 3*, čije se performanse i rad temelje na softverskom dijelu otvorenog koda. Kako bi se dobio bolji uvid u odabrane relevantne mrežne emulatore, provedena je analiza funkcionalnosti i radnog okruženja istih te se može zaključiti kako su mrežni emulatori IMUNES i CORE praktički isti, no to je i očekivano s obzirom na to da se CORE temelji na otvorenom kodu IMUNES emulatora. GNS3 ima puno veći spektar mogućnosti od druga dva emulatora, ali je samim time i kompleksniji za korištenje.

U praktičnom dijelu rada proveden je eksperiment emulacije lokalne mrežne topologije unutar odabranih mrežnih emulatora radi detaljnijeg prikaza i usporedbe rada te mogućnosti istih. Na temelju tablice (tablica 3), koja je izrađena nakon provedenih emulacija, može se zaključiti kako se GNS3 emulator ističe intenzitetom zauzimanja resursa računala na kojem se izvršava eksperiment emulacije te vremenom kašnjenja kontrolnih paketa, čiji je uzrok intenzivan rad procesora računala, dok se IMUNES i CORE emulator ističu svojom jednostavnošću konfiguracije mrežnih elemenata, ali i realnim vremenima kašnjenja kontrolnih paketa, s obzirom na to da se radi o lokalnoj mreži, što je zapravo i cilj eksperimenata emulacije.

Razlog zbog kojeg se koristi eksperiment emulacije, osim troškova izvođenja te pouzdanosti rezultata je mogućnost izvođenja u stvarnom vremenu te mogućnost povezivanja emuliranih elemenata s vanjskim komponentama računalnih mreža. Stoga se može zaključiti kako su rezultati eksperimenta emulacije bliski stvarnosti čime je omogućena bolja procjena mrežne opreme, protokola i aplikacija prije implementacije istih u računalnu mrežu.

Literatura

[1] Ćukušić, M., Jadrić, M.: *IT-sigurnost*, Sveučilišni računski centar, Zagreb, 2015.

[2] Kavran, Z., Grgurević I.: *Autorizirana predavanja iz kolegija "Računalne mreže",* Sveučilište u Zagrebu, Fakultet prometnih znanosti, Zagreb, 2017.

[3] URL: https://sysportal.carnet.hr/node/342. (pristupljeno: srpanj 2017.)

[4] Radovan, M.: *Računalne mreže*, Digital point tiskara, Rijeka, 2010.

[5] Korać, M., Car, D.: *Uvod u računalne mreže*, Algebra d.o.o., Zagreb, 2014.

[6] URL:http://searchtelecom.techtarget.com/definition/fiber-to-the-x-FTTx (pristupljeno: srpanj 2017.)

[7] URL: http://searchnetworking.techtarget.com/definition/Ethernet (pristupljeno: srpanj 2017.)

[8] Tennenhouse, L., D., Wetherall, J., D.: *Towards an active network architecture*, ACM SIGCOMM Computer Communication Review, New York, p. 81-94, 2007.

[9] Bush, S., F.: *Toward in vivo nanoscale communication networks: utilizing an active network architecture*, Frontiers of Computer Science in China, New York, p. 316-326, 2011.

[10] URL: http://mreze.layer-x.com/s010402-0.html. (pristupljeno: srpanj 2017.)

[11] URL: http://www.phy.pmf.unizg.hr/~dandroic/nastava/rm/hub_vs_switch.pdf . (pristupljeno: srpanj 2017.)

[12] URL: http://pc-market.i-mall.hr/images/artikli/5/59089/velika.jpg. (pristupljeno: srpanj 2017.)

[13] URL: http://mreze.layer-x.com/s010403-0.html. (pristupljeno: srpanj 2017.)

[14] URL: http://www.cisco.com/c/en/us/products/routers/index/_jcr_content/Grid/category_atl/layout-category-atl/blade_4/bladeContents/thirds_0/Th-Third-2/tile_639b/ image.img.png/1498087172822.png (pristupljeno: srpanj 2017.)

[15] Havaš, L., Keček, D., Knez, K.: *Usporedba i primjena "Distance Vector" i "Link State" mrežnih protokola*, Tehnički glasnik, vol. 7, no. 2, p. 108-115, Varaždin, 2013.

[16] URL: http://www.phy.pmf.unizg.hr/~dandroic/nastava/diplome/drad_bruno_plancic.pdf. (pristupljeno: srpanj 2017.)

[17] Goleniewski, L.,: *Telecommunications Essentials*, Addison-Wesley Professional, Boston, 2002.

[18] URL: http://mreze.layer-x.com/s020100-0.html. (pristupljeno: kolovoz 2017.)

[19] URL: https://sysportal.carnet.hr/node/379. (pristupljeno: kolovoz 2017.)

[20] URL: https://www.tutorialspoint.com/data_communication_computer_network/computer_network_topologies.htm. (pristupljeno: kolovoz 2017.)

[21] URL: http://free-zg.t-com.hr/Zlatko_Jurelinac/lekcija5.pdf. (pristupljeno: kolovoz 2017.)

[22] Mrvelj, Š.: *Autorizirana predavanja iz kolegija "Tehnologija telekomunikacijskog prometa 1",* Sveučilište u Zagrebu, Fakultet prometnih znanosti, Zagreb, 2017.

[23] Beuran, R.: *Introduction to Network Emulation*, Pan Stanford Publishing, Singapur, 2013.

[24] Zec, M.: *Simulacija i emulacija računalnih mreža i raspodijeljenih sustava primjenom virtualizacijskih mehanizama,* Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, Zagreb, 2012. URL: https://www.fer.unizg.hr/download/repository/Zec,_KDI.pdf (pristupljeno: kolovoz 2017.)

[25] Guruprasad, S., Ricci, R., Lepreau, J.: *Integrated network experimentation using simulation and emulation,* Testbeds and Research Infrastructures for the Development of Networks and Communities, Trento, 2005.

[26] URL: http://imunes.net/ (pristupljeno: kolovoz 2017.)

[27] Puljiz, P., Mikuc, M.: *IMUNES Based Distributed Network Emulator*, Proceedings of International Conference on Software in Telecommunications and Computer Networks, Split, Croatia, p. 198-203, 2006.

[28] U.S. Naval Research Laboratory: *CORE Documentation,* Washington, 2015.

[29] URL: http://docs.gns3.com (pristupljeno: kolovoz 2017.)

[30] URL: https://sysportal.carnet.hr/node/352 (pristupljeno: kolovoz 2017.)

[31] URL: http://mreze.layer-x.com/s020302-0.html (pristupljeno: kolovoz 2017.)

[32] URL: http://tvolaric.com/preuzimanja/protokoli.pdf (pristupljeno: kolovoz 2017.)

[33] Bažant, A.: *Osnove prijenosa podataka,* Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, Zagreb, 2012.

[34] CERT: *Virtualizacija računala,* CARNet, Zagreb, 2009.

[35] URL: https://www.ibm.com/support/knowledgecenter/hr/ssw_i5_54/rzaja/rzajaipsec.htm. (pristupljeno: kolovoz 2017.)

[36] Seder, L., Ilić. Ž., Kos, M.: *Sigurno usmjeravanje u ad hoc mrežama,* Automatika, p. 269-278, Zagreb, 2011.

[37] Interoperability Laboratory: *Fast Ethernet Consortium,* University of New Hampshire, New Hampshire, 2006.

[38] URL: http://www.cis.hr/www.edicija/LinkedDocuments/CCERT-PUBDOC-2007-02-183.pdf. (pristupljeno: kolovoz 2017.)

Popis kratica i akronima

BER	(Bit Error Rate) broj pogrešno prenesenih bita u odnosu na ukupan broj bita
CEL	(Check Emulation Light) simbol greške u CORE emulatoru
CORE	(Common Open Research Emulator) istraživački mrežni emulator otvorenog koda
FTTB	(Fiber to the Building)
FTTC	(Fiber to the Curb)
FTTH	(Fiber to the Home)
FTTN	(Fiber to the Node)
GNS3	(Graphical Network Simulator 3) grafički mrežni simulator
ICMP	(Internet Control Message Protocol) kontrolni protokol .
IEEE	(Institute of Electrical and Electronics Engineers) institut za inženjere elektrotehnike i elektronike
IMUNES	(Integrated Multiprotocol Network Emulator/Simulator) integrirani multirpotokolarni mrežni emulator/simulator
IP	(Internet Protocol) Internet protokol
IPSec	(IP Security) sigurnosni protokol
ISO	(International Organization for Standardization) Međunarodna organizacija za standardizaciju
LAN	(Local Area Network) lokalna računalna mreža
LSA	(Link State Advertisement) paket stanja veze
MAC	(Media Access Control) kontrola pristupa mediju
MAN	(Metropolitan Area Network) gradska računalna mreža
MANET	(Mobile Ad Hoc Network) mobilna ad hoc mreža
MDR	(OSPF MANET Designated Routers) usmjernik za bežične mreže

OSI	(Open System Interconnection Model) slojeviti teoretski model mreže
P2P	(Peer to Peer) prijenos „od točke do točke“
PAN	(Personal Area Network) osobna računalna mreža
ROM	(Read Only Memory) memorija samo za očitavanje
STP	(Shielded Twisted Pair) zaštićena upletena bakrena parica
TCP/IP	(Transmission Control Protocol / Internet Protocol) protokol kontrole prijenosa / Internet protokol
UTP	(Unshielded Twisted Pair) nezaštićena upletena bakrena parica
WAN	(Wide Area Network) mreže širokog područja

Popis slika

Slika 1. Preklopnik 8
Slika 2. Usmjernik 8
Slika 3. *Point-to-point* mrežna topologija 10
Slika 4. Sabirnička mrežna topologija 11
Slika 5. Hijerarhijska mrežna topologija 11
Slika 6. Zvjezdasta mrežna topologija 12
Slika 7. Prstenasta mrežna topologija 12
Slika 8. Isprepletena mrežna topologija 13
Slika 9. Prikaz korisničkog grafičkog sučelja IMUNES emulatora 25
Slika 10. Prikaz alatne trake s mrežnim elementima u IMUNES emulatoru 25
Slika 11. Usporedba grafičkih sučelja CORE i IMUNES emulatora 27
Slika 12. Prikaz alatne trake s mrežnim elementima u CORE emulatoru 27
Slika 13. Prikaz korisničkog grafičkog sučelja GNS3 emulatora 29
Slika 14. Prikaz složenih i spojenih mrežnih elemenata u IMUNES emulatoru 30
Slika 15. Popis mrežnih uređaja te prijenosnih kanala 31
Slika 16. Konfiguracija prijenosnog kanala 32
Slika 17. Konfiguracija računala 32
Slika 18. Konfiguracija usmjernika 33
Slika 19. Pokretanje eksperimenta emulacije 33
Slika 20. „Skok“ intenziteta rada procesora računala prilikom pokretanja eksperimenta emulacije 34
Slika 21. Slanje *ICMP* paketa sa računala *pc2* na računalo *pc5* 35
Slika 22. Ponovljeni postupak slanja *ICMP* paketa s drugačijom konfiguracijom prijenosnog kanala 35
Slika 23. Prikaz složenih i spojenih mrežnih elemenata u CORE emulatoru 36
Slika 24. Konfiguracija prijenosnog kanala 37
Slika 25. Konfiguracija računala 37
Slika 26. Odabir protokola za usmjernik 38
Slika 27. „Skok“ intenziteta rada procesora računala prilikom pokretanja eksperimenta emulacije 39

Slika 28. Slanje *ICMP* paketa sa računala *PC2* na računalo *PC4*..... 40

Slika 29. Ponovljeni postupak slanja *ICMP* paketa s drugačijom konfiguracijom prijenosnog kanala 40

Slika 30. Prikaz uspješno instaliranog usmjernika 41

Slika 31. Odabir slobodnih ulaza na mrežnim uređajima 42

Slika 32. Odabir *FastEthernet* sučelja na usmjerniku..... 42

Slika 33. Prikaz složenih i spojenih mrežnih elemenata u GNS3 emulatoru 43

Slika 34. Prikaz visokog intenziteta korištenja procesora..... 44

Slika 35. Prikaz *Idle-PC* parametra te prikaz smanjenja intenziteta korištenja procesora..... 44

Slika 36. Prikaz uspješne konfiguracije usmjernika u aplikaciji *Console* 45

Slika 37. Prikaz uspješno integriranog *RIP* protokola 46

Slika 38. Prikaz uspješno konfiguriranog računala u aplikaciji *Console* 46

Slika 39. Slanje *ICMP* paketa sa računala *PC1* na računalo *PC6*..... 46

Popis tablica

Tablica 1. Usporedba eksperimentalnih tehnika prema navedenim kriterijima 18
Tablica 2. Usporedba mrežnih emulatora 23
Tablica 3. Usporedba mrežnih emulatora prema kriterijima..... 47