UPOTREBA SIMULACIONIH SISTEMA U NASTAVI IZ RAČUNARSKIH MREŽA - II provera

By Zoran Jovanović

WORD COUNT 23104 TIME SUBMITTED 07-JUL-2015 03:41PM

PAPER ID 18699342

66

UNIVERZITET SINGIDUNUM

BEOGRAD

DEPARTMAN ZA POSLEDIPLOMSKE STUDIJE I MEĐUNARODNU SARADNJU

DOKTORSKA DISERTACIJA

UPOTREBA SIMULACIONIH SISTEMA U NASTAVI IZ RAČUNARSKIH MREŽA

MENTOR

Prof.dr Aleksandar Jevremović

KANDIDAT

Zoran Jovanović

BEOGRAD, 2015. godina

Sadržaj

Sadržaj 1
Spisak slika 4
Spisak tabela 7
Spisak programskog koda 8
Apstrakt 9
Abstract 10
1. Uvod 11
1.1. Predmet i cilj istraživanja 12
1.2. Polazne hipoteze 14
1.3. Metode istraživanja 15
1.4. Doprinos 16
1.5. Struktura rada 17
2. Osnovi računarskih mreža 18
2.1. LAN mreže 19
2.1.1. Mreže istog prioriteta 20
2.1.2. Serverske mreže 21
2.2. Mrežni protokoli 21
2.3. Slojeviti referentni modeli 23
2.3.1. OSI model 24
2.3.2. TCP/IP referentni model 29
2.3.3. Enkapsulacija 32
2.4. Mrežne topologije 34
2.4.1. Topologija zvezde 35
2.4.2. Topologija prstena 35
2.4.3. Topologija magistrale 36
2.4.4. Topologija rešetke 36

2.5. Ethernet 37
2.6. Mrežni sloj i IP protokol 40
2.6.1. IP protokol 40
2.6.2. Internet adresa 43
2.6.3. ARP (Adrress Resolution Protocol) 46
2.6.4. Ruter i rutiranje 47
3. Model mrežnog simulatora 55
3.1. Mrežni elementi 56
3.2. Korisnički interfejs 58
3.3. Kontrolni modul 59
4. Mrežni simulatori u nastavi iz računarskih mreža 60
4.1. Razlozi za izradu i upotrebu mrežnih simulatora 60
4.2. Pregled mrežnih simulatora 62
4.2.1.OPNET IT Guru Academic Edition 62
4.2.2. SSF mrežni simulator 64
4.2.3. NS2 simulator 64
4.2.4. NS3 simulator 66
4.2.5. GNS3 simulator 67
4.2.6. GTNetS simulator 69
4.2.7. GloMoSim simulator 69
4.2.8. KivaNS 72
4.2.9. MIMIC 73
4.2.10. CNET 75
4.2.11. NetSIM 76
4.2.12. Nessi 77
4.2.13. WnetSim 77
5. Uporedna analiza simulatora računarskih mreža 80
5.1.Kriterijumi za analizu simulatora računarskih mreža 80
5.2. Rezultati analize 83
6. Interaktivni sistem za učenje osnova IP umrežavanja 89
6.1. Modelovanje IP adrese 90
6.2. Rad sa sistemom 94
6.2.1. Kalkulator IP adresa 95

6.3. Laboratorijske vežbe 98
6.3.1. Vežba 1 – Rad sa brojnim sistemima 99
6.3.2. Vežba 2 - Klase IP adresa 101
6.3.3. Vežba 3 - Podela IP mreže na podmreže 103
6.3.4. Vežba 4 - Određivanje podmrežne adrese 105
6.3.5. Vežba 5 - CIDR 107
6.3.6. Vežba 6 - Subneting - topologija 1 108
6.3.7. Vežba 7 - Subnetig - topologija 2 109
6.3. Procena upotrebljivosti sistema 111
7. Zaključak 114
Literatura 116
Dodatak A. Izvorni kod aplikacije za učenje osnova IP umrežavanja 125
Dodatak B. Pregled objavljenih radova 132
Biografija kandidata 133

Spisak slika

Slika 2.1. Mreža istog prioriteta 20
Slika 2.2. Serverska mreža 21
Slika 2.3. OSI model 24
Slika 2.4. Podela sloja veze 26
Slika 2.5. Struktura okvira 27
Slika 2.6. Slojevi TCP/IP referentnog modela 31
Slika 2.7. Veza između slojeva OSI modela i TCP/IP 32
Slika 2.8. Protokli TCP/IP po slojevima 32
Slika 2.9. Enkapsulacija podataka 33
Slika 2.10. Primer enkapsulacije podataka 34
Slika 2.11. Topologija zvezde 35
Slika 2.12. Topologija prstena 36
Slika 2.13. Topologija magistrale 36
Slika 2.14. Topologija rešetke 37
Slika 2.15. Funkcionisanje Etherneta 38
Slika 2.16. Povezivanje mreža 41
Slika 2.17. Načini komunikacije u mrežama 42
Slika 2.18. IP paket 42
Slika 2.19. Klase IP adresa 44
Slika 2.20. Kreiranje podmreže 45
Slika 2.21. Maska podmreže 46
Slika 2.22. Arp zahtev 46
Slika 2.23. ARP odgovor 47
Slika 2.24. Povezivanje mreža ruterima 48
Slika 2.25. Usmeravanje paketa 49
Slika 2.26. Algoritam za usmeravanje 50

Slika 2.27. Algoritam za usmeravanje za podmrežno adresiranje 51
Slika 2.28. Primer rutiranja 51
Slika 2.29. Radni modovi rutera 53
Slika 2.30. Konfigurisanje rutera 54
Slika 3.1. Blokovska struktura modela mrežnog simulatora 56
Slika 3.2. Osnovna klasa koja modeluje statičke osobine mrežnih elemenata 57
Slika 3.3. UML model klase Attribute 57
Slika 3.4: Asinhrona komunikacija između mrežnih elemenata 58
Slika 4.1. OPNET IT Guru 63
Slika 4.2. NS-2 Script Generator 65
Slika 4.3. NS-3 NetAnim 67
Slika 4.4. GNS3 mrežni simulator 68
Slika 4.5. KivaNS simulator 73
Slika 4.6. MIMIC Simulation Studio 74
Slika 4.7. CNET mrežni simulator 75
Slika 4.8. NetSim mrežni simulator 76
Slika 4.9. NESSI mrežni simulator 77
Slika 4.10. WnetSim mrežni simulator 78
Slika 4.11. Konfigurisanje rutera u WnetSim mrežnom simulatoru 79
Slika 6.1. Web laboratorija za informatiku i računarstvo 90
Slika 6.2. Java klasa IPadresa 91
Slika 6.3. Web aplikacija za učenje osnova IP umrežavanja 94
Slika 6.4. IP Kalkulator 96
Slika 6.5. Primer IP mrežne adrese 97
Slika 6.6. Podmrežne adrese 97
Slika 6.7. Brojni sistemi 99
Slika 6.8. Rad sa brojnim sistemima 100
Slika 6.9. Klase IP adresa 101
Slika 6.10. Primer upotrebe aplikacije 102
Slika 6.11. Podela IP mreže na podmreže 103
Slika 6.12. Primer popunjenog obrasca 104
Slika 6.13. Određivanje podmrežne adrese 105
Slika 6.14. Primer generisanja IP adresa 106

Slika 6.15. Primer CIDR 107
Slika 6.16. Odgovori za dati CDIR blok 107
Slika 6.17. Mrežna topologija 108
Slika 6.18. Mrežna topologija sa slučajno generisanom IP adresom 108
Slika 6.19. Unos IP adresa 109
Slika 6.20. Mrežna topologija sa unetim IP adresama 109
Slika 6.21. Statistika postignutih rezultat 110

Spisak tabela

Tabela 2.1. Značenje polja okvira 27
Tabela 2.2. Osobine Etherneta 37
Tabela 2.3. Verzije Etherneta 39
Tabela 2.4. Opis polja IP paketa 43
Tabela 2.5. Privatne adrese 45
Tabela 2.6. Tabela rutiranja 51
Tabela 4.1. Modeli i protokoli 71
Tabela 4.2. Osnovne karakteristike predstavljenih simulatora 79
Tabela 5.1. Osnovne teme koje se koriste kao kriterijumi pokrivenosti 82
Tabela 5.2. Vrednovanje simulatora prema usvojenim kriterijumima 85
Tabela 5.3. Udeo pojedinih oblasti u ukupnom gradivu 86
Tabela 5.4. Rezultati upoređivanja simulatora prema pokrivenosti oblasti 86

Spisak programskog koda

Listing 3.1: Aktivni objekti se realizuju kao Java niti 58
Listing 6.1. Konstruktor klase IPadresa 91
Listing 6.2. Metoda podmreznaAdresa 92
Listing 6.3. Metoda opsegPodmreznihAdresa 92
Listing 6.4. Metoda broadcast 93
Listing 6.5. Metoda klasa 93
Listing 6.6. Metoda prvaAdresa 93
Listing 6.7. Metoda zadnjaAdresa 94
Listing A.1. klasa App 125
Listing A.2. klasa BinarnaAdresa 125
Listing A.3. klasa BinarnaMaska 126
Listing A.4. klasa Broadcast 126
Listing A.5. klasa PodrazumevanaMaska 127
Listing A.6. klasa NovaIPadresa 128
Listing A.7. klasa BrojniSistemi 128

Apstrakt

Tradicionalni obrazovni sistem, koji se uglavnom zasniva na teorijskom učenju, ne daje dobre rezultate kada je u pitanju obrazovanje studenata radi rešavanja praktičnih problema. Simulacije i simulacioni sistemi predstavljaju važnu podršku za praktične aspekte nastave i pružaju dobar način za povezivanje i razumevanje apstraktnih koncepata koji su obrađeni na predavanju. U ovoj disertaciji dat je pregled simulatora koji se mogu koristiti u nastavi iz predmeta Računarske mreže; definisani su i kriterijumi koji se mogu koristiti u procesu evaluacije navedenih sumulatora i izvršena je njihova evaluacija.

Informacione tehnologije pružaju mogućnost povećanja efikasnosti obrazovnog procesa. Internet servisi su postali sastavni deo života mnogih ljudi koji Internet primenjuju u svim oblastima pa i u obrazovnom procesu. U ovoj disertaciji je prikazano modelovanje računarske mreže i razvijen je web-zasnovan sistem ua učenje osnovnih koncepata IP adresiranja. Sistem je interaktivan i omogućava, simulacijom i vizuelizacijom, jasnije razumevanje komplikovanih procesa u računarskoj mreži.

Sistem je u upotrebi nekoliko poslednjih godina i na osnovu te upotrebe izvedeni su kvalitativni i kvantitativni zaključci. Kvalitativni rezultati uključuju broj studenata koji je učestvovao u procesu evaluacije i razmena iskustava sa nastavnicima koji su primenjivali sistem u nastavnom procesu. Kvantitativna analiza se zasniva na statističkoj analizi uspeha studenata koji su primenjivali sistem.

Abstract

Traditional educational system, which is mostly based on theoretical learning, does not give good results when it comes to the qualification of students for solving practical problems. Simulators have become very useful support for practical aspects of teaching methods. This paper describes various simulators of computer networks which are mainly used for educational purposes. This paper attempts to give a survey of simulators suitable for teaching courses in computer network, to establish the evaluation criteria and to evaluate selected simulators according to these criteria.

Current technologies give us the ability to enhance and replace developmental classes with computer-based resources, often called web labs. E-learning tools became a very useful support for practical aspects of teaching methods. This paper presents web-based e-learning tools which have possibility of use as components for distant learning in the process of engineer education for informatics. The system is interactive and covers system simulation and process visualization, and has possibility of explaining complicated work of computer networks.

The system is realized in Java. Primary requests for the development environment have been the supporting of facilities of the computing and informatics as well as to minimize the cost. A zero cost environment based on Java has been used. Major characteristics of Java such as hardware independence, internal security, and network based environment, compels the use of Java environment as quite promising.

We have used for the last five year our system and we have conducted a qualitative and quantitative evaluation of using our e-learning tools in the classroom. The qualitative evaluation included a number of student surveys and discussions with fellow instructors who teach courses that directly or indirectly have the course using our web lab as a prerequisite. The quantitative evaluation substantiated this subjective perception: the percentage of students passing the exam increased.

1. Uvod

Ubrzani razvoj tehnoloških dostignuća i mogućnosti koje pruža informaciono doba nameće nova pravila ponašanja u obrazovanju. Obrazovni sistem mora da se prilagodi i da omogući nove obrazovne forme, koje bi pružile mogućnost polaznicima da bez dodatnih materijalnih troškova, koristeći tehnološka dostignuća, učestvuju u nastavnom procesu, a da nemaju poteškoća zbog toga što nisu fizički prisutni u istom.

Računarske mreže predstavljaju veoma važnu oblast u računarstvu i informatici, pa kursevi u ovoj oblasti moraju da ostvare više ciljeva. Primarni cilj je da moraju da pruže pregled osnovnih koncepata na kojima se zasnivaju računarske mreže i da daju uvid u rad tipične računarske mreže. Osim toga, oni bi trebalo da naglase sva važna pitanja u računarskim mrežama sa kojima se inženjeri suočavaju u praksi.

Nastava iz oblasti računarskih mreža se nalazi u preporukama vodećih međunarodnih organizacija koje se bave obrazovanjem i navodi se kao bitan deo celokupnog obrazovanja stručnjaka informatike i računarstva. Studenti se sa konceptima računarskih mreža susreću na predavanjima i vežbama i od njih se traži da razumeju apstraktne pojmove o radu ovih sistema. Tradicionalni obrazovni sistem, koji se uglavnom zasniva na teorijskom učenju, ne daje dobre rezultate kada je u pitanju osposobljavanje studenata za rešavanje praktičnih problema. Upotreba stvarne opreme, u praktičnoj nastavi, je ograničena cenom te opreme i skalabilnošću. Simulacije i simulacioni sistemi predstavljaju važnu podršku praktičnim aspektima nastave i pružaju dobar način za povezivanje i razumevanje apstraknih koncepata koji su obrađeni na predavanju.

Simulatori realizovani za upotrebu u edukacione svrhe omogućavaju poboljšanje procesa učenja i proširivanje znanja studenata putem simulacije realnih situacija na lako razumljiv način. Sa druge strane, razvoj novih tehnologija zahteva neprestano prilagođavanje i unapređivanje mogućnosti koje pružaju sistemi za simulaciju. Simulatori treba da omoguće verno praćenje ponašanja simuliranih procesa u računarskim mrežama, sa dovoljnim nivoom detalja, tako da rezultati odgovaraju stvarnim sistemima.

Simulatori razvijeni u akademske svrhe imaju raspoloživ izvorni kod tako da studenti, pored opisa sistema koji simuliraju, mogu da prate i implementaciju pojedinih delova simulatora, ali oni nisu prilagođeni za razvoj i simulaciju kompleksnih sistema računarskih mreža. Nasuprot simulatorima koji su razvijeni za akademske svrhe, postoje i komercijalni sistemi koji su suviše komplikovani za korišćenje i ne nude odgovarajuće načine za prikaz rezultata rada simulacije studentima koji uče apstraktne koncepte. Pored toga izvorni kod ovih simulatora nije raspoloživ studentima kako bi pratili detalje vezane za implementcaiju.

Detaljnom analizom raspoloživih softverskih okruženja uočeno je da postojeći sistemi ne pokrivaju u potpunosti sve aspekte u nastavu iz računarskih mreža, pa se u tom smislu otvara prostor za realizaciju novog sistema uz pomoć kojeg bi se ovaj problem rešio.

Imajući ovo u vidu, razvijen je novi interaktivni na web-u zasnovan sistem koji, kao deo web laboratorije, omogućava izvođenje laboratorijskih vežbi vezanih za osnove IP umrežavanja. Osnovne funkcionalnosti ovog sistema, kao i skup laboratorijskih vežbi koje su implementirane, detaljno su opisani u nastavku.

1.1. Predmet i cilj istraživanja

Predmet ovog istraživanja jeste nastava na visokoškolskim institucijama iz oblasti Računarskih mreža. Načini na koje studenti stiču znanja iz oblasti Računarskih mreža su raznovrsni. Jedan od osnovnih načina, pored teorijske nastave, jeste upotreba stvarne opreme. Stvarna oprema čini laboratoriju za Računarske mreže koja omogućava praktičan rad u realnom okruženju. Međutim realne laboratorije su pored cene ograničene i limitiranim pristupom u pogledu pristupa određenog broja studenata. Da bi se ovi problemi prevazišli koriste se i drugi pristupi, a jedan od njih jeste upotreba simulatora i simulacionih sistema u nastavnom procesu.

Razvoj mrežnih sistema, posebno u oblastima novih tehnika rutiranja, uštede energije i sve većih razmera ovih mreža, zahteva neprestano razvijanje novih simulacionih sistema, koji su sa druge strane sve zahtevniji i traže sve više računarskih resursa pa se zato projektovanje ovako složenih sistema javlja kao jedan od problema koji treba rešiti.

Mnoge obrazovne ustanove na ovaj izazov odgovaraju razvojem programa on-line učenja. Kada se pruži mogućnost da se nastavni sadržaji plasiraju putem Interneta, neophodno

je i da se laboratorijske vežbe izvode putem Interneta, odnosno neophodno je omogućiti studentu on-line interakciju sa stvarnom ili virtuelnom opremom kao što su računari, ruteri, habovi svičevi ...

Kada se omogući kontrola opreme, stvarne ili virtuelne na on-line način, sa udaljenog računara dobija se on-line laboratorija. Virtuelne laboratorije se koriste kako u oblasti tehničko-tehnoloških nauka [1][2] i prirodno-matematičkih nauka [3] tako i u polju društveno-humanističkih nauka.

Da bi virtuelne laboratorije bile adekvatan alat u edukacionom procesu one moraju da realizuju tri teorijska principa:

- Autentičnost: Omogućiti laku deljivost istraživačkih alata
- Kompleksnost: Mogućnost vizualizacije kompleksnih procesa
- Kolaboracija: Mogućnost komunikacije

Na Internetu su na raspolaganju različiti tipovi on-line laboratorija. Neke od njih nude on-line interfejs za pristup stvarnim fizičkim uređajima, da bi se sa udaljenog mesta vršilo podešavanje, kontrola ili očitavanje parametara tih uređaja [1]. Ovi sistemi omogućavaju kreiranje i on-line publikaciju laboratorijskih vežbi koje se zasnivaju na tim uređajima.

Drugi tip on-line laboratorija radi sa simulacijama stvarne opreme sa ciljem prikazaivanja funkcionalnosti tih uređaja [2]. Ova okruženja dozvoljavaju studentu da konfiguriše uređaje na isti način kao što se radi sa realnom opremom, ali u kontrolisanim uslovima.

Neke virtuelne laboratorije su realizovane u formi Java apleta i omogućavaju jednostavan interaktivan rad i vizuelizaciju procesa u posmatranom sistemu.

Predmet istraživanja u ovom radu je komparativna analiza postojećih simulacionih sistema za učenje Računarskih mreža i razvoj novog interaktivnog web-zasnovanog sistema za primenu u nastavi iz Računarskih mreža. Komparativna analiza performansi simulatora računarskih mreža nije, u velikoj meri, zastupljena u stručnoj literaturi, a raspoloživi radovi uglavnom uključuju ograničen skup eksperimenata sa jednostavnim simulacionim scenarijima. U ovom radu se simulatori proučavaju sa aspekta primene u nastavi, pa je i predmet ovog rada definisanje i odgovarajućih kriterijuma za upoređivanje karakteristika simulatora.

Opšti cilj istraživanja ovog rada sastoji se u tome da se podaci o ponašanju nekog sistema dobiju, ne proučavanjem samog sistema, nego se ponašanje sistema proučava tako da se napravi (kreira) model, pa se, eksperimentisanjem na takvom modelu, dolazi do saznanja o stvarnom sistemu.

Naučni cilj istraživanja se može opisati kao potreba za definisanjem kriterijuma za ocenjivanje i evaluaciju selektovanih simulatora i da se na osnovu tih kriterijuma izvrši njihovo upoređivanje.

Praktični cilj ovog istraživanja je da da pregled mrežnih simulatora pogodnih za nastavne predmete u oblasti računarskih mreža, da se utvrede prednosti i mane postojećih sistema i da se razvije novi simulacioni web-zasnovani sistem za upotrebu u nastavi Računarskih mreža.

1.2. Polazne hipoteze

Osnovna hipoteza u ovoj disertaciji je: potrebno je i moguće je realizovati efikasnu virtuelnu laboratoriju, zasnovanu na simulacijama i tehnikama vizuelizacije procesa koji se odigravaju u računarskim mrežama, a koji bi na jasan način približili studentima kompleksne i apstraktne procese na kojima se zasniva rad jedne računarske mreže.

U izradi disertacije polazi se od sledećih hipoteza:

- Procesi koji se odigravaju u računarskim mrežama su veoma kompleksni i apstraktni, pa se svaki predavač iz ove oblasti susreće sa problemom kako studentima približiti te apstraktne pojmove i koncepte.
- Simulacioni sistemi omogućavaju vizuelizaciju procesa i povezivanje apstraktnih koncepata što zahteva razvoj i korišćenje softverskih sistema za simulaciju rada realnih sistema razvijenih posebno za tu namenu.
- Nastava iz računarskih mreža na visokoškolskim institucijama može se organizovati u okruženju koje se zasniva na mrežnom simulatoru.
- Postojeći simulacioni sistemi koji simuliraju računarske mreže nisu u potpunosti pogodni za edukaciju, pa je potrebno definisati jasne kriterijume za procenu upotrebljivosti pojedinih simulatora u nastavnom procesu.

- Softverski sistemi se mogu koristiti prilikom transfera znanja iz oblasti računarskih mreža u oblast projektovanja simulatora za računarske mreže.

1.3. Metode istraživanja

U ovom radu primenjena je kombinovana teorijsko-eksperimentalna metoda istraživanja.

Teorijska metoda rada biće zasnovana na objektno orijentisanim tehnologijama i podrazumeva razvoj modela pomoću koga će biti moguće modelovanje elemenata računarske mreže. Tipični objekti su IP adresa, MaskaMreže, BroadcastAdresa ...

Eksperimentalna metoda rada podrazumeva primenu razvijenog modela na složene simulacije konkretnih procesa u računarskoj mreži.

Za modelovanje sistema, prilagođavanje sistema okruženju za učenje na daljinu i za kreiranje mrežnih scenarija koristi se deskriptivna metoda.

Izrada disertacije obuhvata sledeće faze procesa istraživanja:

- Sistemsko proučavanje domaće i strane literature iz oblasti računarskih mreža.
- Definisanje ključnih pojmova iz oblasti računarskih mreža.
- Identifikovanje i određivanje problema nastave iz računarskih mreža i kritička analiza istih.
- Evaluacija softverskih sistema za simulaciju rada računarskih mreža, sa aspekta primene u procesu obrazovanja.
- Izvođenje logičkih posledica iz hipoteza i izbor istraživačke strategije i plana istraživanja.
- Obrađivanje i analiziranje prednosti i mana analiziranih simulacionih sistema i rezultata evaluacije.
- Definisanje analitičkog modela interaktivnog web zasnovanog sistema za učenje računarskih mreža.

- Verifikacija polaznih hipoteza rezultata dobijenih na osnovu realizacije i implementacije odgovarajućeg sistema u relavantnim oblastima iz računarskih mreža.

1.4. Doprinos

Naučni doprinosi ovog rada je u oblasti analize i sinteze web zasnovanog softverskog sistema koji treba da omogući kreiranje i upotrebu labaratorijskih vežbi u nastavi iz računarskih mreža i oni mogu da se raščlane na sledeće doprinose:

- Pregled i analiza dosadašnjih istraživanja iz oblasti primene simulatora u edukaciji.
- Analiza nedostataka postojećih modela i sistema.
- Definisanje kriterijuma za uporednu analizu simulatora koji mogu da se koriste za učenje Računarskih mreža.
- Uporedna analiza simulatora koji mogu da se koriste u nastavi RM, na osnovu definisanih kriterijuma.
- Predlog i implementacija novog sistema.
- Predlog daljih pravaca rada na osnovu prethodnih istraživanja.

Stručni doprinosi ove disertacije su:

- Pregled iskustava upotrebe simulatora računarskih mreža u nastavi.
- Pregled simulatora koji se mogu koristiti u nastavi Računarskih mreža
- Definisanje modela edukacionog okruženja za računarske mreže
- Razvoj novog interaktivnog simulacionog sistema za učenje i testirane znanja iz Računarskih mreža.

Rezultati rada će biti vrednovani na osnovu analize rada sistema procene na osnovu koje se koriguje upotreba i povećava efikasnost nastavnog procesa. Rezultati rada na ovoj doktorskoj disertaciji biće objavljeni u više radova u časopisima međunarodnog značaja i

saopšteni na više naučnih skupova u zemlji i inostranstvu. Pojedine ideje su već izložene u poznatim međunarodnim stručnim časopisima.

1.5. Struktura rada

Ova disertacija se sastoji od osam poglavlja.

Nakon uvodnih razmatranja, u poglavlju pod naslovom: "Osnovi računarskih mreža", opisani su osnovni pojmovi iz računarskih mreža. Posebno je dat akcenat na opisu prenosa podataka u mreži. Sam proces prenosa je razdeljen na nekoliko sistematičnih koraka i radnji koje se odvijaju na računaru koji šalje podatke i na računaru koji prima podatke, kao i dva referentna modela OSI i TCP/IP model. Treće poglavlje "Model mrežnog simulatora", definiše model simulatora koji može da se koristu nastavnom procesu iz računarskih mreža. U četvrtom poglavlju prikazan je opis najpoznatijih simulatora. Opisane su njihove najznačajnije karakteristike i data je njihova analiza sa stanovišta moguće upotrebe u nastavnom procesu. U petom poglavlju su definisani kriterijumi za evaluaciju predloženih simulatora i urađena je detaljna analiza i kvantitativno upoređivanje simulatora. Šesto poglavlje: "Interaktivni sistem za učenje osnova IP umrežavanja" opisuje aplikaciju koja se koristi za izvođenje laboratorijskih vežbi koje su vezane za osnove IP umrežavanja, kao i faze iz kojih se sastoji svaka laboratorijska vežba.

Posle završnog sedmog poglavlja, koje predstavlja zaključak svega opisanog, navode se literaturni izvori koji su korišćeni u toku izrade rada. Izvorni kod aplikacije i pregled objavljenih radova su dati kao posebni dodaci. Na kraju je data biografija kandidata.

2. Osnovi računarskih mreža

Računarsku mrežu[4] čine dva ili više računara koji su međusobno povezani telekomunikacionim linijama. Svrha ovog povezivanja jeste da se omogući prenos podataka kao i upotreba zajedničkih, deljenih resursa. U resurse spadaju baze podataka, aplikacije i periferni uređaji. Periferni uređaji su štampači, diskovi, modemi i dr.

Drugi razlog umrežavanja jeste omogućavanje komunikacija umreženih računara. Slanje podataka je složen postupak koji se može raščlaniti na sledeće poslove:

- prepoznavanje podataka,
- podela podataka na pakete,
- dodavanje informacija u svaki paket, da bi se ustanovio položaj podataka i da bi se identifikovao primalac,
- dodavanje podataka za proveru greške i
- slanje podataka kroz mrežu.

Postoje dve osnovne grupe računarskih mreža i to:

- lokalne računarske mreže LAN (Local Area Network)-mreže i
- globalne računarske mreže WAN (Wide Area Network)-mreže.

Prve su lokalizovane na manjem prostranstvu, kao što je kancelarija, sprat ili čitava zgrada. Druge su rasprostranjene na širim geografskim područijima, kao što je oblast nekog grada, države, kontinenta i dr.

Računarska mreža se sastoji od računara, druge opreme koja omogućava prenos podataka i od prenosnih puteva kroz koje se odvija protok podataka.

Svaka računarska mreža ima neke svoje karakteristike kao što su:

- topologija,
- arhitektura,
- prenosni kablovi i
- kontrola pristupu medijima.

Topologija definiše prostorni raspored računara u mreži. Arhitektura propisuje način funkcionisanja uređaja, način povezivanja uređaja i protokole koji omogućavaju komunikaciju.

Prenosni medijumi predstavljaju različite materijale i vrste kablova koji se koriste u mreži za fizičko povezivanje računara.

Kako računarsku mrežu čini više računara koji mogu da pokušavaju da u isto vreme pošalju podatke kroz isti prenosni medijum, moraju postojati različite metode koje treba da kontrolišu pristup tom medijumu.

Performanse računarske mreže su:

- propusna moć,
- raspoloživost,
- sigurnost i
- cena koštanja.

Propusna moć mreže se definiše kao broj bita koji se prenose u jedinici vremena. Izražava se jedinicom broj bita u sekundi (b/s). Veće jedinice su kilo-bita u sekundi (Kb/s), mega-bita u sekundi (Mb/s), giga-bita u sekundi (Gb/s).

Raspoloživost zavisi od srednjeg vremena između dva kvara u mreži i srednjeg vremena otklanjanja kvara u mreži. Što je vreme između kvarova veće, a vreme otklanjanja kvarova manje, raspoloživost mreže je veća.

Sigurnost je definisana merama koje su preduzete da se obezbedi sigurna isporuka podataka i da se otkloni mogućnost neovlašćenog korišćenja podataka.

2.1. LAN mreže

Manje računarske mreže koje se prostiru na manjim područjima (obično se nalaze u jednoj zgradi) nazivaju se lokalnim računarskim mrežama [5]. Sve lokalne mreže imaju neke zajedničke komponente, funkcije i osobine:

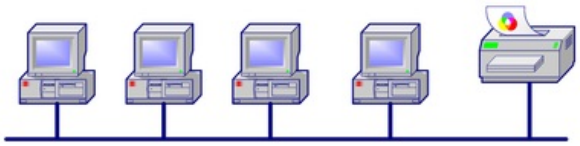
- serveri - računari koji obezbeđuju resurse kojima pristupaju umreženi korisnici,
- klijenti - računari koji pristupaju zajedničkim deljenim resursima koje obezbeđuje server,
- medijumi - prenosni put kojim se prenose podaci i
- resursi - datoteke, štampači i drugi elementi kojima pristupaju umreženi računari.

Postoji više kriterijuma na osnovu kojih se prave klasifikacije LAN mreža. U te kriterijume spadaju: topologija mreže, prenosni put koji povezuje računare u mreži, metoda kontrole pristupa medijumu. A postoji i opšta klasifikacija po kojoj LAN-mreže možemo podeliti na:

- mreže istog prioriteta i
- serverske mreže.

2.1.1. Mreže istog prioriteta

Kod mreža istog prioriteta (Slika 2.1.) ne postoji namenski server, već računari su ravnopravni i imaju jednak prioritet prilikom pristupa deljenim resursima[6]. Svaki računar se ponaša i kao server i kao klijent, svaki korisnik administrira svoj računar i brine se o bezbednosti tako da ne postoji centralizovano administriranje mreže. Ove mreže su pogodne ako se u mreži nalazi manje od 10 računara i ako pitanje bezbednosti nije primarno.

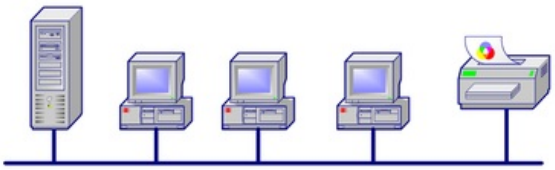


Slika 2.1. Mreža istog prioriteta

Kao operativni sistem za mrežu istog prioriteta može se koristiti Windows for Workgroups, Windows NT Workstation, Windows 95 (Windows 98), Windows 2000, Windows XP, Windows 7, Windowsw 8 itd.

2.1.2. Serverske mreže

Serverske mreže (Slika 2.2.) sadrže namenske servere koji su optimizovani da opsluže zahteve mrežnih klijenata i da pruže visok stepen bezbednosti datoteka i kataloga[7].



Slika 2.2. Serverska mreža

Namenski serveri mogu biti:

- server za datoteke i štampanje,
- server za aplikacije,
- server za elektronsku poštu i
- komunikacijski server.

Ove mreže mogu da opsluže na hiljade umreženih klijenata. Administriranje i upravljanje mrežom je centralizovano.

Često moderne mreže u poslovnim sredinama predstavljaju kombinaciju najboljih osobina mreža istog prioriteta i serverskih mreža[59].

2.2. *Mrežni protokoli*

Svi skupovi pravila za umrežavanje nazivaju se mrežnim protokolima[8]. Koncept mrežnog protokola zasniva se na pretpostavci da dva računara moraju da se zajednički pridržavaju jednog protokola (skupa pravila) kako bi mogli da komuniciraju.

Ceo proces prenosa podataka u mreži može da se podeli na nekoliko odvojenih, sistematičnih koraka. Na svakom koraku obavlja se određena radnja:

Na računaru koji šalje podatke protokol:

- rastavlja podatke na pakete,
- dodaje podatke o adresi u paket i
- priprema podatke za prenos.

Na računaru koji prima podatke protokoli rade iste radnje ali obrnutim redosledom:

- uzima podatke sa kabla,
- dovodi podatak u računar,
- oslobađa pakete svih podataka o prenosu koji je dodao računar pošiljalac,
- sklapa pakete u celinu i
- predaje tako sređene podatke aplikaciji.

Stek protokola je niz više međusobno zavisnih protokola. Svaki nivo traži drugačiji protokol za rukovanje određenom funkcijom.

Protokoli i mrežne kartice mogu da se mešaju i da se uparuju po potrebi. Na primer dva različita protokola mogu da se vežu za jednu mrežnu karticu. Od redosleda vezivanja zavisi kojim će redosledom operativni sistem pokretati te protokole[11].

Protokoli koji podržavaju komunikaciju LAN-ova preko više putanja nazivaju se usmerivi protokoli.

Neki od najpoznatijih mrežnih protokola su:

- DLC (Data Link Control)
- IPX/SPX (Internetwork Packet Exchange/Sequenced Packet Exchange)
- NetBIOS/NetBEUI (Network Basic Input/Output System/NetBIOS Extended User Interface)
- TCP/IP (Transmission Control Protocol/ Internet Protocol)

2.3. Slojeviti referentni modeli

Postoje brojna pitanja oko kojih računari, koji komuniciraju, moraju da se slože:

- Koji naponski nivoi se koriste za predstavljanje 0, a koji za predstavljenje 1
- Kako primalac zna koji je poslednji bit u poruci
- Kako može da se detektuje greška u prenosu
- Kako se identifikuju pošiljalac i primalac
- Kako se predstavljaju podaci različitog tipa i dr.

Pošiljalac i primalac moraju da se slože o problemima koji mogu da se klasifikuju u slojeve, počev od najnižeg sloja koji određuje detalje prenosa bitova, pa do najvišeg sloja koji određuje način predstavljanja podataka. Da bi se opisale i standardizovale sve mrežne funkcije, koje omogućavaju da računari međusobno komuniciraju, koriste se referentni modeli koji te funkcije grupišu u slojeve[9].

Slojevi služe za apstrakciju različitih nivoa u organizaciji računarskih mreža. Kreiraju se tako da svaki sloj ima dobro definisanu funkciju u okviru koje se upotrebljavaju međunarodno standarizovani protokoli[60].

Prednosti slojevitih modela su:

- Oni standardizuju mrežne komponente, tako da omogućavaju proizvodnju i podrški različitih firmi.
- Oni razbijaju mrežnu komunikaciju na manje, jednostavnije delove, tako da je učenje i razumevanje jednostavnije.
- Dozvoljavaju komonikikaciju različitih softverskih i hardverskih komponenti.
- Promene na jednom sloju ne utiču na rad ostalih slojeva.

2.3.1. OSI model

CCITT (Consultative Committee for International Telephony and Telegraphy i ISO (International Organization for Standardization) predložili su 1978. godine skup specifikacija koje opisuju arhitekturu računarskih mreža[10]. ISO je 1984. godine objavio reviziju ovog modela i nazvao ga OSI (Open System Interconnection) referentni model, koji je prihvaćen kao standard za računarske mreže. Ovaj model ima slojevitu strukturu. Slojevi OSI modela su:

- Fizički sloj
- Sloj veze podataka
- Mrežni sloj
- Transportni sloj
- Sloj sesije
- Sloj prezentacije
- Sloj aplikacije

Na svakom OSI nivou definisane su mrežne funkcije, a funkcije svakog nivoa komuniciraju sa funkcijama nivoa koji se nalazi neposredno iznad ili neposredno ispod njega. Svrha svakog nivoa je da pruža usluge sledećem višem nivou i da taj nivo štiti od detalja implementacije tih usluga. Nivoi su tako raspoređeni da se svaki ponaša kao da komunicira sa odgovarajućim nivoom na drugom računaru (slika 2.3.).



Slika 2.3. OSI model

U stvarnosti svaki nivo komunicira samo sa nivoom koji je ispod ili iznad njega.

Najniži sloj OSI modela bavi se protokolima vezanim za fizički deo računarske mreže koji omogućavaju komunikaciju računara. Fizički sloj odgovoran je za prenos bitova kroz fizički medijum. On omogućava da signali, električni, optički ili radio signali, budu razmenjeni između mašina koje komuniciraju. Fizički sloj definiše električne, mehaničke i proceduralne karakteristike koje su neophodne da bi se uspostavila, održavala i prekinula fizička veza između računara[61].

Fizički sloj uključuje razmatranje:

- Signala
- Mrežnih kablova
- Repetitora
- Hub-ova
- Mrežnih topologija

Namena hub-a je da regeneriše i prosleđuje signal u mreži. Hub se koristi da bi se u mreži kreiralo centralno komunikaciono čvorište i da bi se povećala pouzdanost u mreži. Pouzdanost se povećava zbog mogućnosti da svaki kabl u mreži otkaže i da to bude bez uticaja na rad čitave mreže.

Hub radi na fizičkom nivou OSI modela jer on jednostavno regeneriše signal i šalje ga u mrežu preko svojih portova.

Računari u mreži komuniciraju razmenom okvira. Okvir sadrži MAC adrese odredišta i izvora i podatke (enkapsulirane podatke viših OSI slojeva). Okviri se pretvaraju u niz bitova koji se pomoću mrežnog kabla prenose hub-u. Hub prihvata, preko jednog svog mrežnog interfejsa, binarne podatke i prosleđuje ih preko ostalih interfejsa do svih računara koji su povezani na hub (osim do računara koji je izvor podataka).

Osnovne funkcije sloja veze podataka su:

- Komunikacija sa višim slojevima OSI modela pomoću LLC (Logical Link Control)
- Jednoznačna identifikacija mrežnih uređaja pomoću fizičkih adresa (MAC adresa)
- Grupisanje podataka u okvire (frame)

- Kontrola pristupa medijumu (Media Access Control - MAC)

Dva najniža OSI sloja definišu kako dva računara mogu da koriste mrežu a da se međusobno ne ometaju.

Da bi stvorio specifikaciju koja dominantno definiše LAN okruženje IEEE je za model projekta 802 posebno razmatrao ta dva nivoa i odlučio je da treba detaljnije razraditi nivo veze i to deleći ga na dva podnivoa (slika 2.4.):

- Podnivo za kontrolu logičkih veza LLC (Logical Link Control)
- Podnivo za kontrolu pristupa medijumu MAC (Media Access Control)

Podnivo za kontrolu logičkih veza upravlja komunikacijom veza sa podacima i definiše upotrebu veza.

Kontrola pristupa medijumu predstavlja podnivo sloja veze podataka koji se odnosi na protokole koji definišu koji računar u mreži, u okruženju deljenog medijuma, može da šalje podatke.



Slika 2.4. Podela sloja veze

Okvir se sastoji od sledećih polja:

- Start
- Adresa
- dužina/Tip
- Podaci
- FCS
- Stop

START	ADRESA	TIP	PODACI	FCS	STOP

Slika 2.5. Struktura okvira

Značenje pojedinih polja je dato u Tabeli 2.1.

START	Predstavlja početnu sekvencu bitova koji čine početak okvira
ADRESA	Svaki okvir ima informacije o računaru koji šalje okvir (izvorna MAC adresa) i informacije o računaru koji prima okvir (odredišna MAC adresa)
DUŽINA/TIP	Neki okviri sadrže polje DUŽINA (LENGTH) koje predstavlje ukupnu dužinu okvira u bajtima, a neki olkviri sadrže polje TIP (TYPE) u kome se specificira protokol trećeg sloja koji je zahtevao slanje
PODACI	Ovo polje sadrži podatke koje korisnik želi da pošalje i neke kontrolne podatke koji služe da bi se omogućila pravilna isporuka paketa višim slojevima OSI modela
FCS	Frame Check Sequence služi za proveru ispravnosti prenetih podataka. Ovo polje sadrži broj koji se izračunava na bazi sadržaja okvira i koji se prenosi do odredišta gde se ovaj broj ponovo izračunava i upoređuje sa prenetim FCS brojem. Ako se oni ne slažu došlo je do greške prilikom prenosa podataka.
STOP	Sekvenca bajtova koja predstavlja kraj okvira

Tabela 2.1. Značenje polja okvira

Mrežni sloj je odgovoran za kretanje podataka između različitih mreža[62]. Logičko adresiranje na ovom sloju omogućava određivanje odredišta podataka koji se prenose kroz mrežu. Logička adresa identifikuje i mrežu i računar u mreži, za razliku od MAC adrese koja identifikuje samo računar.

Za povezivanje različitih mreža na trećem sloju OSI modela koristi se ruter. Ruter pravi logičke odluke, na bazi logičkih adresa, i vrši usmeravanje paketa kroz različite mreže birajući najbolju putanju.

Protokoli mrežnog nivoa OSI modela su[63]:

- IP (Internet Protocol)

- ICMP (Internet Control Message Protocol)
- ARP protokol (Address Resolution Protocol)
- RARP (Reverse Adrress Resolution Protocol)

Transportni sloj omogućava unapređenje servisa mrežnog sloja. Ovaj sloj omogućava da se uspostavi pouzdana isporuka podataka i očuva njihov integritet sa kraja na kraj veze. Da bi se osigurao pouzdan prenos podataka, transportni sloj se gradi na mehanizmima kontrole grešaka koje obezbeđuju niži slojevi. Ovaj sloj predstavlja poslednju mogućnost da se greške otklone. On se obično implementira na samom čvoru i može da poveća propusnu moć mreže tako što kreira više paralelnih kanala za jednu vezu. Primeri transportnih protokola su SPX (Sequenced Exchange Protocol) i TCP (Transmission Control Protocol) protokol.

Sloj sesije omogućava da dve aplikacije sa različitih računara uspostave, koriste i završe vezu koja se zove sesija. U ovom procesu, korišćenje transportnog sloja omogućava poboljšanje servisa tokom sesije. Primer sesije je korisnik koji je prijavljen na računar preko mreže ili sesija otvorena da bi se izvršio prenos datoteke. Ako, na primer, treba da izvršite jedno-časovni prenos datoteka, a veza puca na približno 30 minuta, može se dogoditi da se ceo transfer nikada ne obavi. Ali, ako se ceo transfer tretira kao jedna aktivnost, sa kontrolnim punktovima (checkpoints), koji su ubačeni u protok podataka, može se izbeći ovaj problem. Ako se desi prekid sloj sesije, može da se sinhronizuje sa prethodnim kontrolnim punktom. Primer protokola za sloj sesije je NetBIOS.

Sloj prezentacije određuje format podataka koji se razmenjuju. On upravlja načinom na koji se prikazuju podaci. Postoji više načina prezentacije podataka. Ako dve strane koje komuniciraju koriste različite reprezentacije podataka, one ne mogu da se razumeju. Uloga sloja prezentacije je da predstavi podatke zajedničkom sintaksom i semantikom. Na računaru koji šalje podatke, taj nivo prevodi podatke iz formata koji je poslao aplikacijski nivo, u prelazni, lako razumljivi format. Na računaru koji prima podatke, ovaj nivo prevodi prelazni format u format koji može da razume aplikacijski nivo računara.

Sloj aplikacije služi kao prozor za pristup aplikacije mrežnim uslugama. Ovaj sloj sadrži protokole i funkcije koje su neophodne korisničkim aplikacijama da bi izvršile određene zadatke. To su, na primer, protokoli koji obezbeđuju servise za rad sa udaljenim datotekama, za prenos datoteka, za pristup udaljenim bazama podataka...

Neki od aplikacijskih protokola su:WWW (World Wide Web), FTP (File Transport Protocol), SMTP (Simple Mail Transfer Protocol), Telnet[64].

2.3.2. TCP/IP referentni model

Kada je 1957. godine lansiran prvi veštački satelit Sputnjak, američka vlada je formirala agenciju za napredne istraživačke projekte ARPA (Advanced Research Projects Agency), da bi omogućila Americi vodeću ulogu u naučnom razvoju u svetu. Glavni cilj bio je da se razvije decentralizovani sistem komunikacije računara različitih proizvođača, kako bi se omogućila kontrola nad vojnim resursima i nakon eventualnog nuklearnog udara[11].

Internet tehnologija je nastala kao rezultat istraživanja koje je obavila Agencija za napredne istraživačke projekte ARPA. Ove projekte je podržala Nacionalna fondacija za nauku NSF (National Science Foundation), Ministarstvo za energetiku DOE (Department Of Energy), Ministarstvo odbrane DOD (Department Of Defense) i Nacionalna uprava za aeronautiku i svemir NASA (National Aeronautics and Space Administration).

Američka vlada je 1969. godine započela niz strateških projekata iz oblasti komunikacija, a sa ciljem povezivanja računara koji su se nalazili u raznim vojnim laboratorijama, vladinim biroima i univerzitetskim laboratorijama u kojima su se razvijali projekti od interesa za armiju. Tako je stvorena mreža koja je nosila naziv ARPANET. Tokom sedamdesetih ARPANET mreža je stalno rasla da bi je 1975. godine u potpunosti preuzelo ministarstvo odbrane, pretvorivši je u današnju DDN (Defence Data Network) mrežu.

Najznačajniji rezultat razvoja ARPANET-a jeste TCP/IP (Transmission Control Protocol /Internet Protocol) protokol koji je definisan 1983. godine i koji predstavlja način za razmenu podataka između različitih računara i mreža.

U isto vreme pojavljuju se i neke druge mreže koje su najpre povezivale računare američkih univerziteta, a kasnije i računare u Evropi i drugim krajevima sveta.

Povezivanje računara je bilo interesantno i za komercijalne organizacije, pa se javljaju brojne mreže koje povezuju računare različitih organizacija. Zatim se javljaju pokušaji povezivanja tih mreža prvo na nacionalnom principu, a potom i na globalnoj osnovi. Kao

rezultat tih napora nastao je INTERNET koji se još naziva i mrežom svih mreža. Ovakav način nastanka Interneta uslovilo je i to da Internet nema vlasnika.

Tehničke smernice razvoja TCP/IP protokola određuje grupa koja se zove Odbor za arhitekturu Interneta IAB (Internet Architecture Board), koji je osnovan 1983. godine. Ovaj odbor određuje koji protokoli su neophodan deo TCP/IP paketa i definiše standarde.

1992. godine formirana je grupa pod nazivom Internet zajednica ISOC (Internet Sociaty), čija je uloga bila da podstiče povezivanje ljudi i korišćenje Interneta širom sveta.

Dokumentacija o Internetu, specifikacije i standardi za TCP/IP su dostupni besplatno u vidi tehničkih izveštaja pod nazivom Zahtevi za komentarima - RFC (Requests for Coments).

Da bi se osigurao jedinstveni mrežni deo u Internet adresama, pitanje adrese se rešava na centralizovan način tako da svaki računar u mreži ima jedinstvenu adresu. Prvobitno se ovim pitanjima bavila Uprava za dodelu Internet adresa - IANA (Internet Assigned Number Authority). Krajem 1998. godine formirana je nova organizacija za dodelu Internet adresa koja se zove Internet korporacija za dodelu imena i brojeva ICANN (Internet Corporation For Assigned Names and Numbers).

Internetu se može pristupiti posredstvom Posrednika za Internet servise - ISP (Internet Service Provider), firme koja nudi pristup Internetu posredstvom svoje infrastrukture.

Provajder treba da obezbedi korisničko ime, šifru za pristup i IP adresu za računar ili klasu IP adresa za mrežu.

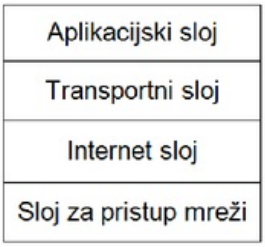
Troškovi pristupa Internetu se svode na troškove zakupa vremena provedenog na Internetu koje plaćamo provajderu i na troškove zauzetosti telefonske linije, obzirom da se Internetu obično pristupa preko telefonske linije.

Pošto se razvoj TCP/IP protokola vezuje za američko ministarstvo odbrane (Department of Defense) ovaj model koji opisuje slojevitost TCP/IP protokola naziva se i DoD model.

DoD model ima četiri sloja (Slika 2.6.):

- sloj za pristup mreži,
- internet sloj (međumrežni sloj)
- transportni sloj

- aplikacijski sloj



Slika 2.6. Slojevi TCP/IP referentnog modela

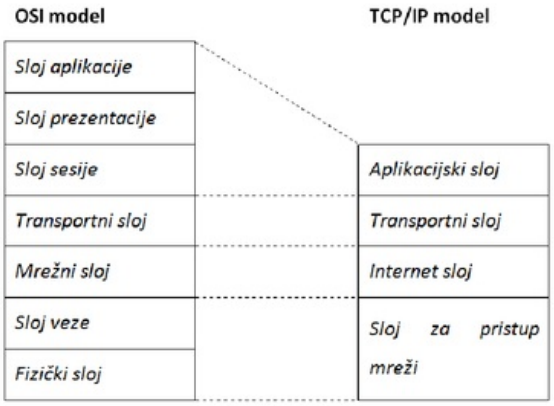
Sloj za pristup mreži ili sloj mrežnog interfejsa obezbeđuje komponente koje fizički povezuju mrežu i on je odovoran za slanje TCP/IP paketa na fizički prenosni medijum i za prihvatanje paketa sa mreže. To su kablovi, mrežne kartice, protokoli veze, standardi za kontrolu pristupa medijumu (CSMA/CD, Token Ring) i dr. TCP/IP funkcioniše nezavisno od načina pristupa mreži, od formata i nosioca podataka, tako da se TCP/IP protokol može koristiti efikasno i kod povezivanja računara u lokalnu mrežu (Ethernet, TokenRing...) i za regionalne WAN mreže (X.25)

Internet sloj (međumrežni sloj) obezbeđuje logičke adrese za fizički mrežni interfejs. Ovaj sloj čine protokoli: IP (Internet Protocol), ARP (Adress Resolution Protocol), RARP (Reverse Adrress Resolution Protocol) i ICMP (Internet Controm Message Protocol) i odgovoran je za adresiranje i usmeravanje paketa.

Transportni sloj implementira vezu između dva hosta u mreži. Ponekad se zove i Host to Host sloj. Protokoli koji pripadaju ovom sloju su: TCP (Transmission Control Protocol) i UDP (User Datagram Protocol).

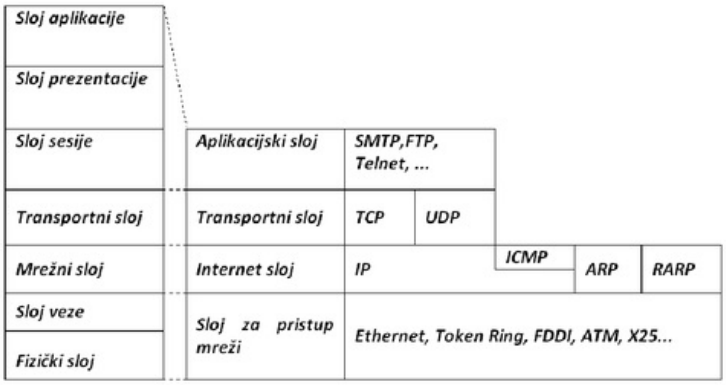
Aplikacijski sloj obezbeđuje aplikacije koje koriste protokole Host to Host sloja. Protokoli ovog sloja su FTP (File Transfer Protocol), TELNET, SMTP (Simple Mail Transfer Protocol), HTTP (Hyper Text Transfer Protocol), SNMP (Simple Network Management Protocol), DNS (Domain Name System), RIP (Routing Information Protocol) [65].

Veza između slojeva OSI modela i TCP/IP modela prikazana je na slici 2.7.



Slika 2.7. Veza između slojeva OSI modela i TCP/IP

Detaljan slojevit model TCP/IP protokola prikazan je na slici 2.8.



Slika 2.8. Protokli TCP/IP po slojevima

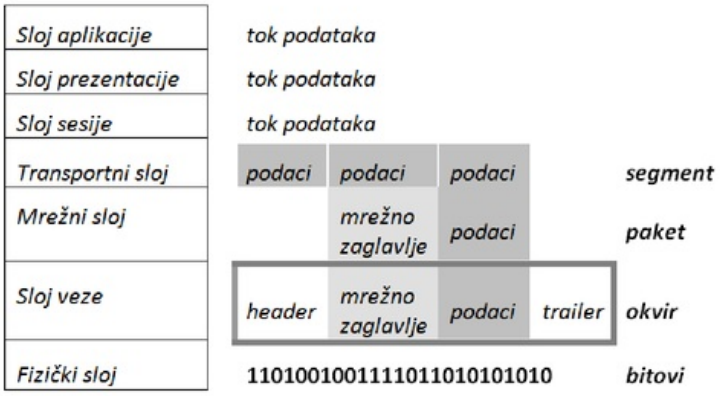
2.3.3. Enkapsulacija

Kada dva računara razmenjuju informacije podaci moraju da se pakuju pomoću procesa koji se naziva enkapsulacija. Enkapsulacijom se podaci omotavaju sa neophodnim informacijama o mrežnim protokolima koji omogućavaju komunikaciju[66].

Podaci se formiraju na izvoru i šalju odredištu. Informacije koje se šalju kroz mrežu predstavljaju se kao podaci ili paketi podataka. Da bi računari uspešno komunicirali u mreži

svaki sloj na izvoru podataka mora da komunicira sa odgovarajućim slojem na oderedištu podataka. Tokom ovog procesa protokoli slojeva razmenjuju informacije koje se zovu PDU (Protocol Data Units) [12]. Svaki nivo obezbeđuje određene funkcije višim nivoima. Da bi to uspešno radio niži nivo koristi enkapsulaciju kako bi stavio PDU višeg nivoa u svoje polje podataka i dodao header i trailer, koji su mu neophodni za obavljanje svoje funkcije.

Slojevi OSI modela i odgovarajući PDU-ovi prikazani su na slici 2.9.



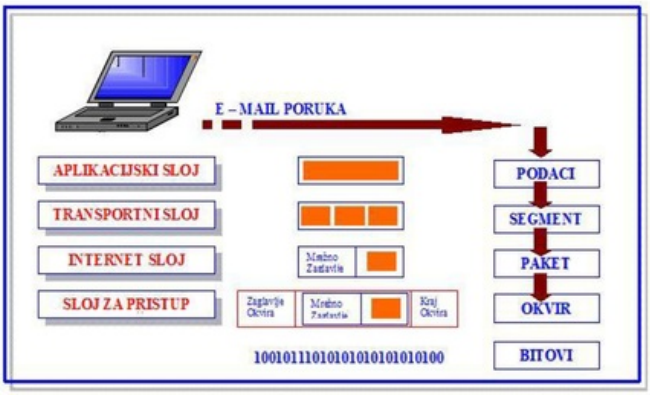
Slika 2.9. Enkapsulacija podataka

Kako se vrši enkapsulacija možemo da vidimo kroz praktičan primer kada korisnik šalje e-mail (slika 2.10.).

Kada korisnik pošalje e-mail prvo se alfanumerički znaci konvertuju u odgovarajuće podatke koji mogu da se prenesu kroz mrežu.

Pomoću segmenta ostvaruje se komunikacija sa kraja na kraj.

Zatim se segmenti smeštaju u paket ili datagram koji sadrži odgovarajuće logičke adrese. Paketi se smeštaju u okvire koji sadrže fizičke adrese mrežnih uređaja, kojima treba isporučiti paket. Okviri se konvertuju u niz bitova (0 ili 1) koji se transportuju kroz mrežu.



Slika 2.10. Primer enkapsulacije podataka

2.4. *Mrežne topologije*

Topologiju računarske mreže čini geometrijska raspoređenost čvorova mreže. Mogućnosti mreže zavise od njene topologije[13]. Od izabrane topologije zavisi:

- vrsta potrebne opreme za mrežu,
- mogućnosti te opreme,
- razvoj mreže i
- način upravljanja mrežom.

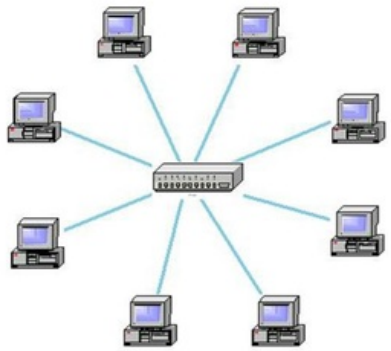
Od izabrane topologije zavisi koji se kablovi koriste u mreži, kao i na koji način se ti kablovi postavljaju. Od topologije zavisi kako računari međusobno komuniciraju, a metod komunikacije ima veliki uticaj na mrežu[67].

Osnovne topologije računarskih mreža su:

- zvezda,
- prsten,
- topologija magistrale (stabla) i
- rešetka.

2.4.1. Topologija zvezde

Kod topologije zvezde komuniciranje između bilo koja dva čvora mreže mora proći kroz centralni uređaj koji se zove hab (hub) (slika 2.11.). Signal se prenosi od računara koji ga šalje, kroz hab, do svih računara u mreži. Hab mora da bude pouzdan i treba da obezbedi izolaciju signala između portova[68].

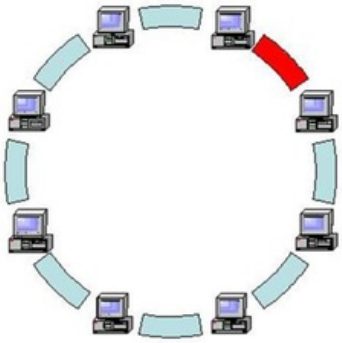


Slika 2.11. Topologija zvezde

Topologija zvezde omogućava centralizovane resurse i upravljanje. Pošto je svaki računar povezan sa centralnim mestom ta topologija zahteva dosta kablova kod velikih mreža. Ukoliko hab otkaže, otkazaće čitava mreža. Ako otkaže jedan računar, onda samo on neće moći da prima i šalje podatke.

2.4.2. Topologija prstena

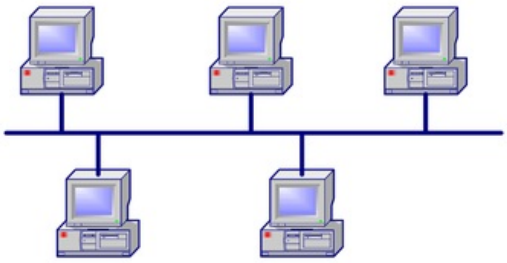
Topologija prstena se sastoji od kabla u obliku petlje sa stanicama priključenim na nju. Signal putuje po petlji u jednom smeru i prolazi kroz svaki računar. Svaki računar prima signal, pojačava ga i šalje ga sledećoj stanici. Ako otkaže jedan računar, cela mreža postaje neupotrebljiva (slika 2.12.) [69].



Slika 2.12. Topologija prstena

2.4.3. Topologija magistrale

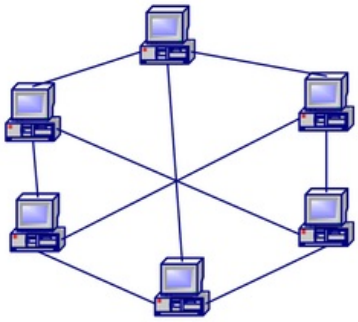
Topologija magistrale se sastoji od linearnog kabla na koji su stanice priključene (slika 2.13.) [70].



Slika 2.13. Topologija magistrale

2.4.4. Topologija rešetke

Topologija rešetke se sastoji od većeg broja računara koji su među sobom povezani u obliku rešetke (slika 2.14.) [71].



Slika 2.14. Topologija rešetke

2.5. Ethernet

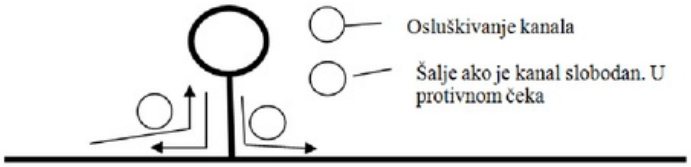
Ethrnet mrežni standard razvile su kompanije Xerox, DEC i Intel 1981. godine. On se zasniva na mehanizmu detekcije nosioca signala[14]. Ove mreže su premošćivale razdaljinu do jednog kilometra i povezivale do 100 radnih stanica. Nakon standardizacije Ethernet postaje poznat kao standard IEEE 802.3, mada se Ethernet donekle razlikuje od standarda IEEE 802.3.

Pregled osobina Ethernet-a prikazan je u sledećoj tabeli.

Klasična topologija	Linearna magistrala
Druga topologija	Kombinacija zvezde i magistrale
Vrsta arhitekture	Osnovni opseg
Metod pristupa	CSMA/CD
Specifikacija	*IEEE 802.3*
Brzina prenosa	10 Mb/s ili 100 Mb/s
Vrsta kablova	Koaksijalni, UTP

Tabela 2.2. Osobine Etherneta

Funkcionisanje Ethernet-a se sastoji u sledećem: pre nego što radna stanica pošalje podatke ona osluškuje aktivnosti na prenosnom kanalu (slika 2.15.).



Slika 2.15. Funkcionisanje Etherneta

Ako stanica otkrije zauzetost kanala ona odlaže slanje podataka, a ako je kanal slobodan stanica šalje podatke. Kako više stanica koriste mehanizam detekcije nosioca signala metod se zove CSMA/CD (Carrier Sense with Multiple Access / Collision Detect), odnosno, višestruki pristup sa detekcijom nosioca signala i detekcijom kolizije. Detekciju signala obavljaju elektronski sklopovi u radnoj stanici (mrežna kartica) [72].

Besmislena transmisija naziva se kolizija. Ako dve stanice šalju istovremeno, a nema aktivnosti na kanalu, dešava se kolizija. Ethernet stanice smanjuju neželjen efekat kolizije otkrivanjem kolizije čim se desi. Otuda i oznaka CSMA/CD. Kada se desi kolizija, stanica koja to prva otkrije šalje svim stanicama signal upozorenja. Tada sve stanice prekidaju slanje za određeni interval.

Postoje dve vrste Ethernet-a, standardni i tanki Ethernet.

IEEE verzija standardnog Ethernet-a se zove 10BASE5. Broj 10 označava rad na 10Mb/s, BASE označava rad u osnovnom opsegu (baseband), a 5 označava 500 metara po segmentu. Najveći broj stanica po segmentu može biti 100, a maksimalni broj stanica ne može preći 1024.

IEEE verzija tankog Ethernet-a se zove 10BASE2. Broj 10 označava rad na 10Mb/s, BASE označava rad u osnovnom opsegu (baseband), a 2 označava približno 200 metara po segmentu (tačnije 185 metara). Koaksijalni kabl koji se koristi za tanki Ethernet je tanji pa otuda i ime. Broj stanica po segmentu je 30, a maksimalan broj stanica može biti 1024. Segmenti se mogu spajati uređajem koji se naziva ripiter. Na ove mreže obično se primenjuje topologija magistrale.

10BASET je novija verzija Ethernet-a koja koristi upredene parice za povezivanje. Broj 10 označava rad na 10Mb/s, BASE označava rad u osnovnom opsegu, a T označava povezivanje upredenim paricama (UTP). Većina mreža ovog tipa ima fizičku topologiju

zvezde. Hab za 10BASET mreže obično služi kao repetitor sa više priključaka. Svaki računar je povezan kablom sa HUB-om. Maksimalna dužina kabla je 100 metara. Minimalna dužina kabla je 2.5 metra. 100BASEX je oznaka za takozvani "Brzi Ethernet" (Fast Ethernet) koji je definisan standardom IEEE 802.3u. Fast Ethernet radi na brzini od 100 Mb/s, koristi fizičku topologiju zvezde, a kao prenosni medijum upotrebljavaju se optička vlakna i upredene parice[73].

Sve verzije Etherneta, njihovi nazivi, brzina rada i tip prenosnog medijuma su prikazani u tabeli 2.3.

Tip etherneta	Brzina	Tip kabla
Desetomegabitni ethernet		
10Base5	10 Mb/s	Koaksijalni debeli
10Base2	10 Mb/s	Koaksijalni tanki
10Base-T	10 Mb/s	UTP kategorije 3/5
10Base-F	10 Mb/s	Optičko vlakno
10Base-FL	10 Mb/s	Optičko vlakno
Stomegabitni ethernet		
100Base-T	100 Mb/s	UTP kategorije 5
100Base-T4	100 Mb/s	UTP kategorije 5
100Base-T2	100 Mb/s	UTP kategorije 5
100Base-TX	100 Mb/s	UTP kategorije 5
100Base-TX	200 Mb/s	UTP kategorije 5
100Base-FX	100 Mb/s	Višerežimsko vlakno
100Base-FX	200 Mb/s	Višerežimsko vlakno
Gigabitni ethernet		
1000Base-T	1 Gb/s	UTP kategorije 5e
1000Base-TX	1 Gb/s	UTP kategorije 6
1000Base-CX	1 Gb/s	STP tvinaksijalni
1000Base-SX	1 Gb/s	Višerežimsko vlakno
1000Base-LX	1 Gb/s	Jednorežimsko vlakno
1000Base-ZX	1 Gb/s	Jednorežimsko vlakno
Desetogigabitni ehternet		
10GBase-CX4	10 Gb/s	STP tvinaksijalni
10GBase-T	10 Gb/s	UTP kategorije 6a/7
10GBase-LX4	10 Gb/s	Višerežimsko vlakno
10GBase-LX4	10 Gb/s	Jednorežimsko vlakno
10GBASE-SR/W	10 Gb/s	Višerežimsko vlakno
10GBASE-LR/W	10 Gb/s	Jednorežimsko vlakno
10GBASE-ER/W	10 Gb/s	Jednorežimsko vlakno

Tabela 2.3. Verzije Etherneta

2.6. Mrežni sloj i IP protokol

Mrežni sloj je odgovoran za kretanje podataka između različitih mreža. Logičko adresiranje na ovom sloju omogućava određivanje odredišta podataka koji se prenose kroz mrežu. Logička adresa identifikuje i mrežu i računar u mreži, za razliku od MAC adrese koja identifikuje samo računar[15].

Za povezivanje različitih mreža na trećem sloju OSI modela koristi se ruter. Ruter pravi logičke odluke, na bazi logičkih adresa, i vrši usmeravanje paketa kroz različite mreže, birajući najbolju putanju.

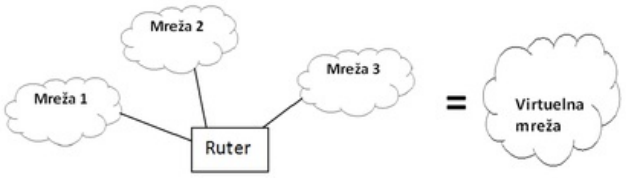
Protokoli mrežnog nivoa OSI modela su:

- IP (Internet Protocol)
- ICMP (Internet Control Message Protocol)
- ARP protokol (Address Resolution Protocol)
- RARP (Reverse Adrress Resolution Protocol)

2.6.1. IP protokol

Internet protokol (IP) je jedan od dva glavna protokola (drugi je TCP) koja se koriste za povezivanje računarskih mreža, pri čemu je osnovni cilj da skup operacija koje se upotrebljavaju za komunikaciju i prenos podataka ostane nezavisan od mrežnih tehnologija i od računarske opreme koje čine te mreže[16].

IP protokol je *nepouzdan protokol bez uspostavljenja veze* (connectionless) i on je odgovoran za adresiranje i usmeravanje paketa između različitih mreža. IP je nepouzdan zato što ne garantuje siguran prenos podataka. Bez uspostavljanja veze znači da se poruke usmeravaju od izvora do odredišta na osnovu adresnih informacija koje se nalaze u samoj poruci. Svaka poruka se usmerava posebno pa se ne može garantovati siguran prenos.



Slika 2.16. Povezivanje mreža

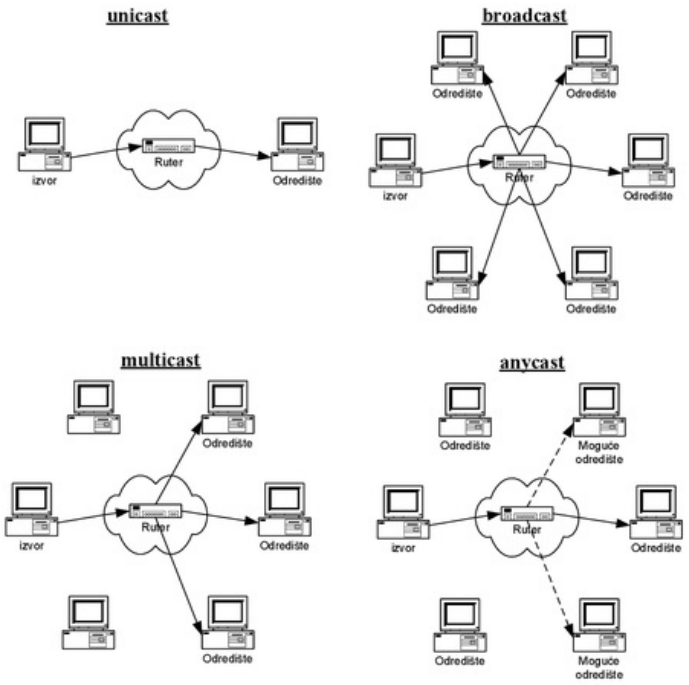
Više mreža mogu biti fizički međusobno povezane samo preko zajedničkog uređaja koji je povezan sa svim mrežama.Uređaj koji međusobno povezuje različite mreže naziva se ruter (router). Na slici 2.16. je prikazan ruter koji povezuje 3 mreže. On mora da bude u stanju da prihvati pakete iz jedne mreže i da ih prosledi mreži kojoj su paketi namenjeni, pri čemu nije bitan hardver povezanih mreža. Ruter se povezuje na mrežu pomoću interfejsa.

Kada jedan mrežni uređaj treba da komunicira sa drugim mrežnim uređajem, koji se nalazi u drugoj mreži, on u tu svrhu mora da koristi default gateway. Default gateway predstavlja logičku adresu interfejsa rutera i povezuje datu mrežu sa mrežom kojoj je namenjen paket[74].

Postoji četiri načina komuniciranja koja su podržana IP protokolom:

- unicast,
- broadcast,
- multicast i
- anycast.

Ovi načini komuniciranja prikazani su na slici 2.17.



Slika 2.17. Načini komunikacije u mrežama

U fizičkoj mreži osnovna jedinica prenosa naziva se okvir (frame). Osnovna jedinica prenosa na Internetu naziva se internet datagram (IP datagram) ili paket. U opštem obliku IP paket sadrži polje zaglavlja i polje podataka. Zaglavlje sadrži izvornu i odredišnu adresu i polje tipa koje određuje sadržaj paketa.

Detaljan sadržaj paketa prikazan je na slici 2.18.

Ver	IHL	Tip servisa	Ukupna dužina	
Identifikacija			Indikator	Fragment offset
Životni vek		Protokol	Kontrolna suma zaglavlja	
Izvorna IP adresa				
Odredišna IP adresa				
IP opcije				Popuna
Podaci				

Slika 2.18. IP paket

Opis pojedinih polja od kojih se paket sastoji dat je u tabeli 2.4.

Ver	Ovo polje sadrži verziju IP protokola
IHL	Predstavlja ukupnu dužinu zaglavlja u 32-bitnim rečima
Tip servisa	Određuje način postupanja sa datagramima
Ukupna dužina	Daje dužinu IP datagrama merenu u bajtovima
Identifikacija	Sadrži ceo broj koji identifikuje datagram
Indikatori	Ovo polje kontroliše fragmentaciju
Fragment offset	Određuje relativnu poziciju podataka u fragmentu
Zivotni vek	Određuje koliko sekundi datagram može da ostane u sistemu
Protokol	Kazuje koji protokol visokog nivoa je upotrebljen za izradu poruke koja se prenosi u polju DATA
Kontrolni zbir zaglavlja	Obezbeđuje integritet zaglavlja
Izvorna IP adresa	Adresa pošiljaoca
Odredišna IP adresa	Adresa primaoca
IP opcije	Koriste se za testiranje mreže
Popuna	Popunjavaju datagram do umnoška 32-bitnih reči
Podaci	

Tabela 2.4. Opis polja IP paketa

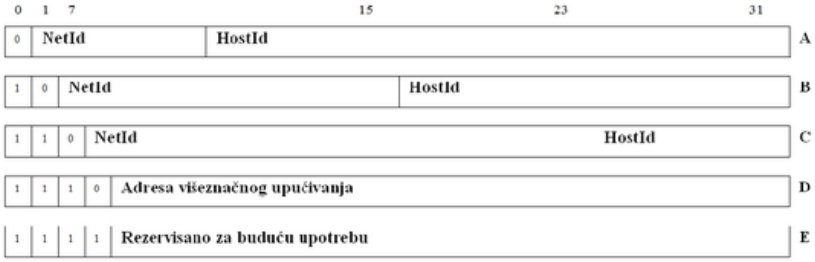
2.6.2. Internet adresa

Za jednu računarsku mrežu kažemo da pruža univerzalni komunikacioni servis ako omogućava svakom računaru u mreži da komunicira sa bilo kojim drugim računarom u mreži. Da bi se to omogućilo neophodno je da postoji sistem identifikacije računara koji bi bio opšteprihvaćen. Svaki računar u mreži može biti identifikovan pomoću *imena*, *adrese* i *puteva*. Ime identifikuje šta neki objekat predstavlja, adresa identifikuje gde se dati objekat nalazi, a putevi pokazuju kako se može stići do datog objekta. Za čoveka je prirodnije da se računari identifikuju imenom, dok je za funkcionisanje jedne virtuelne strukture koja je potpuno implementirana softverski, kakav je Internet , efikasnija reprezentacija identifikatora u vidu binarnih adresa, koje omogućavaju i efikasan način za izbor najbolje komunikacione putanje[17].

Internet adresa, ili IP adresa, je 32-bitni broj koji se piše u obliku četiri osmobitmna broja razdvojena tačkom pri čemu se osmobitni brojevi predstavljaju dekadno (npr. 192.101.121.6).

Adrese se klasifikuju na adrese A, B, C, D i E klase.

IP adresa se sastoji od dva dela: mrežni deo (netID) i hostID. NetID deo IP adrese jednoznačno identifikuje mrežu a hostID određuje adresu čvora u mreži[75].



Slika 2.19. Klase IP adresa

Razni tipovi klasa IP adresa su definisani da bi se izašlo u susret potrebama mreža različitih veličina (slika 2.19.).

IP adrese klase A imaju 7 bitova rezervisanih za NetId i 24 bita za HostId. Namenjene su za veoma velike mreže i mogu da identifikuju po 16777214 (2^{24}-2) računara u 126 (2^{7}-2) mreža. Prvi broj u adresama klase A može biti broj od 1 do 126 .

Adrese klase B su srednje veličine i one su pogodne za srednje i velike organizacije. Za identifikaciju mreže koristi se 14 bitova a za identifikaciju čvora 16 bitova. Ove adrese mogu da identifikuju po 65534 (2^{16}-2) računara u mreži. Prvi broj u IP adresi treba da bude u opsegu od 128 do 191.

Kod adrese klase C za NetId se koristi 21 bit, a za HostId 8 bitova. Ove adrese su namenjene za male mreže, i one mogu da identifikuju 254 (28-2) računara u mreži. Prvi broj u IP adresi klase C treba da bude u opsegu od 192 do 223.

Adrese klase D služe za multicast i ne koriste se za adresiranje pojedinačnih računara. Prvi broj u adresi mora biti u opsegu od 224 do 239.

Adrese iz klase E se koriste za eksperimentalne potrebe. Prvi broj u adresi mora biti iz opsega od 240 do 255.

Nekoliko adresa su rezervisane za neke specijalne namene.

HostId nikada ne može da bude 0. Ako su svi bitovi u HostId-u jednaki nuli onda se data IP adresa koristi za ukazivanje na mrežu.

HostId nikada ne može da bude 1. Ako su svi bitovi HostId-a jednaki jedinici onda se paket isporučuje difuzno svim računarima u mreži i ta adresa se naziva broadcast address.

Prva adresa u svakoj mreži (klasi) predstavlja mrežnu adresu, a poslednja adresa je rezervisana za broadcast adresu.

U svakoj klasi postoji određen broj adresa koje se ne dodeljuju i koje se nazivaju privatnim adresama.

Privatne adrese su prikazane u tabeli 2.5.

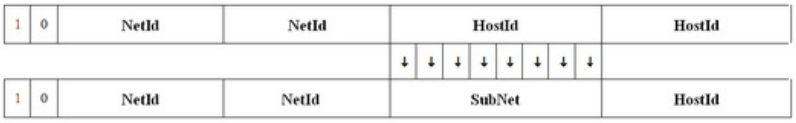
10.0.0.0	-	10.255.255.255
172.16.0.0	-	172.31.255.255
192.168.0.0	-	192.168.255.255

Tabela 2.5. Privatne adrese

IP mreža može biti podeljena na manje mreže koje se zovu podmreže. Deljenje mreže na podmreže pruža veću fleksibilnost i efikasnije korišćenje IP adresa. Na primer, mreža sa mrežnom adresom 172.16.0.0 može biti podeljena na podmreže sa adresama: 172.16.1.0, 172.16.2.0, 172.16.3.0, 172.16.4.0...

Adresa podmreže se kreira tako što se bitovi iz polja HostId koriste kao polja podmreže. Broj pozajmljenih bitova je promenljiv i on se određuje pomoću maske podmreže.

Način na koji se bitovi iz HostId polja koriste za kreiranje podmrežnog adresnog polja prikazan je na slici 2.20.



Slika 2.20. Kreiranje podmreže

Maska podmreže koristi isti format i način reprezentacije kao i IP adresa. Maska podmreže sadrži 1 na pozicijama svih bitova koji pripadaju NetId i SubNet poljima a 0 na pozicijama svih bitova HostId polja (slika 2.21.).

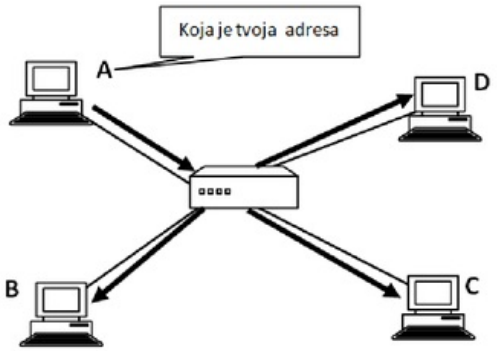
1	0	NetId	NetId	SubNet	HostId
11111111			11111111	11111111	00000000
255.			255.	255.	0

Slika 2.21. Maska podmreže

2.6.3. ARP (Adrress Resolution Protocol)

TCP/IP šema adresiranja obezbeđuje 32-bitnu IP adresu za svaki računar u mreži, koja se sa korisnikove tačke gledišta posmatra kao virtuelna tvorevina, pri čemu protokoli na visokom nivou koriste jedino te adrese[18].

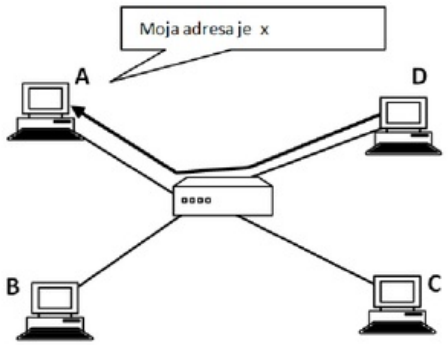
Međutim, da bi dva računara u datoj fizičkoj mreži komunicirala, svaki od njih mora da poznaje stvarnu fizičku adresu drugog računara, tako da svaki računar u mreži ima stvarnu fizičku adresu i dodeljenu IP adresu. Cilj je da se primenom IP adresiranja sakriju hardverski detalji i da se omogući da korisnik radi samo sa IP adresama. Ali, kako komunikacija mora da se odvija kroz stvarne fizičke kanale, uz primenu stvarnih fizičkih adresa, potrebno je obezbediti mehanizam koji će omogućiti mapiranje IP adrese u stvarnu fizičku adresu. U tu svrhu se koristi protokol za razrešavanje adresa (ARP Adress Resolution Protocol).



Slika 2.22. Arp zahtev

Da bi smo objasnili kako se vrši razrešavanje adrese, posmatrajmo jednu fizičku Ethernet mrežu koja se sastoji od četiri računara (A, B, C, D) (slika 2.22.). Svaki Ethernet

mrežni intefejs sadrži 48-bitnu fizičku adresu koja identifikuje svaki računar u mreži. Kako je IP adresa 32-bitna to nije moguće izvršiti direktno kodiranje fizičke adrese i IP adresu. Ideja na kojoj se zasniva ARP protokol je sledeća: računar A treba da komunicira sa računarom D pomoću stvarne fizičke adrese računara D, koju računar A ne zna. Računar A poznaje samo IP adresu računara D.



Slika 2.23. ARP odgovor

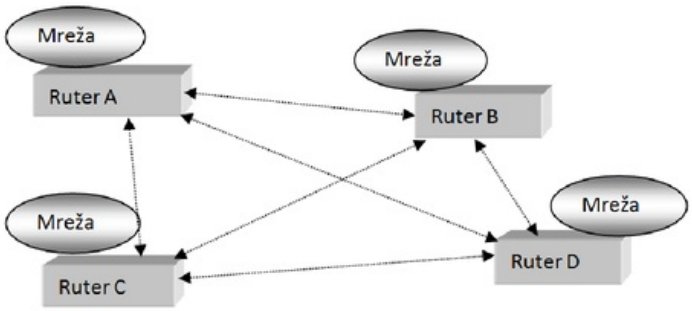
Da bi saznao potrebnu fizičku adresu računar A difuzno šalje zahtev koji od računara D traži da mu na osnovu njegove IP adrese pošalje fizičku adresu. Zahtev primaju svi računari u mreži, ali samo računar D prepoznaje svoju IP adresu i odgovara na zahtev tako što šalje svoju fizičku adresu računaru A (slika 2.23.).

2.6.4. Ruter i rutiranje

U okruženju koje se sastoji od više mrežnih segmenata, sa različitim protokolima i arhitekturama, most ne može da obezbedi brzu komunikaciju. Tako kompleksna mreža zahteva uređaj koji ne samo da zna adrese svih segmenata nego može da omogući najbolju putanju za slanje podataka od izvora do odredišta. Takvi uređaji zovu se ruteri[19]. Usmeravanje (engl. routing) ili rutiranje predstavlja proces biranja putanje preko koje se šalje IP datagram. Ruter ili usmeravač predstavlja računar koji vrši izbor putanje.

Ruter radi u trećem OSI sloju. Ako mreže rade sa istim protokolima, ruter može da odredi optimalnu moguću putanju paketa do odredišta (slika 2.24.). Ako, na primer, ruter A treba da pošalje podatke ruteru D, on može da pošalje podatke ruterima B i C, a podaci će biti

isporučeni ruteru D. Ruteri imaju sposobnost da procene obe putanje i odrede koja putanja je najbolja za usmerenje saobraćaja.



Slika 2.24. Povezivanje mreža ruterima

Ruteri mogu da čitaju kompleksne mrežne adrese iz paketa, tako da mogu da komutiraju i usmeravaju pakete preko više mreža. Hardver rutera može da bude mrežni server, poseban računar ili specijalan uređaj[76].

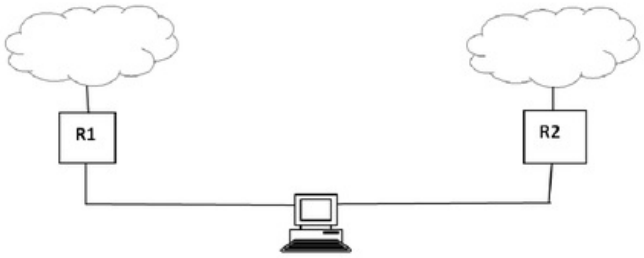
Kada ruter primi paket za udaljenu mrežu, on ga šalje ruteru koji upravlja odredišnom mrežom. Ruter ne komunicira sa udaljenim računarom, već samo usmerava mrežne pakete. Oni ne pregledavaju adrese odredišnih čvorova već pregledavaju samo mrežne adrese. Tabela usmeravanja kod rutera sadrži mrežne adrese.

Ruteri mogu biti statički i dinamički.

Kod statičkih rutera administrator mora ručno da podešava tabelu usmeravanja, dok dinamički ruteri sami automatski otkrivaju putanje i zahtevaju najmanje podešavanja i konfigurisanja.

Osnovni cilj IP protokola je da obezbedi virtuelnu mrežu koja obuhvata više vizičkih mreža koje su međusobno povezane usmerivačima.

U procesu odlučivanja o usmeravanju IP datagrama učestvuju i računar koji šalje podatke i ruter koji povezuje mrežu u kojoj se nalazi dati računar, kao i ostale mreže (Slika 2.25.).



Slika 2.25. Usmeravanje paketa

Računar koji šalje podatak mora da donese početnu odluku da li će poslati datagram preko rutera R1 ili R2. Osim ove početne odluke, o putanji prenosa IP datagrama između različitih mreža odlučuju ruteri.

Usmeravanje može biti direktno i indirektno.

Direktan prenos predstavlja prenos IP paketa sa jednog računara, preko jedne fizičke mreže, na drugi računar.

Indirektni prenos se odnosi na slučaj kada se odredište ne nalazi u istoj fizičkoj mreži kao i pošiljalac, već pošiljalac mora da preda paket ruteru da bi ga ovaj dalje preneo.

Kod direktnog prenosa pošiljalac enkapsulira paket u fizički okvir, mapira IP adresu odredišta u fizičku adresu i prenosi paket pomoću mrežnog hardvera, pri čemu ne koristi usluge rutera.

Indirektan prenos je teži od direktnog jer pošiljalac mora da identifikuje ruter kome može da pošalje paket, a ruter treba dalje da prosledi paket preko određene putanje do odredišta.

Kada se mrežna adresa izvora i odredišta razlikuje, tada pošiljalac enkapsulira paket u okvir i šalje okvir na adresu default gateway-a. Default gateway je adresa interfejsa rutera koji datu mrežu povezuje sa drugim mrežama.

Ruteri u virtuelnoj mreži formiraju kooperativnu strukturu. Paket prolazi od jednog do drugog rutera sve dok ne stigne do rutera koji može direktno da prosledi paket do odredišta.

Osnovno pitanje koje se javlja jeste: kako ruter zna kojom putanjom treba da prosledi datagram do odredišta?

Uobičajeni algoritam za rutiranje koristi *tabelu usmeravanja* koja čuva informacije o svim odredištima i mogućim putanjama do njih.

U tabeli usmeravanja mogu se naći tri vrste putanja do odredišta:

- direktna putanja,
- indirektna putanja i
- podrazumevana putanja, koja se koristi za prosleđivanje datagrama kada se odredišna IP adresa ne nalazi u tabeli usmeravanja.

Kada IP protokol treba da prenese IP paket on se obraća tabeli usmeravanja. Na osnovu informacija u tabeli usmeravanja odlučuje se kojom putanjom treba poslati paket.

Tabela usmeravanja treba da sadrži samo mrežne adrese što čini usmeravanje efikasnim, a tabele usmeravanja malim.

Tabela usmeravanja treba da sadrži parove (IP adresa odredišne mreže, IP adresa interfejsa rutera duž putanje ka odredišnoj mreži).

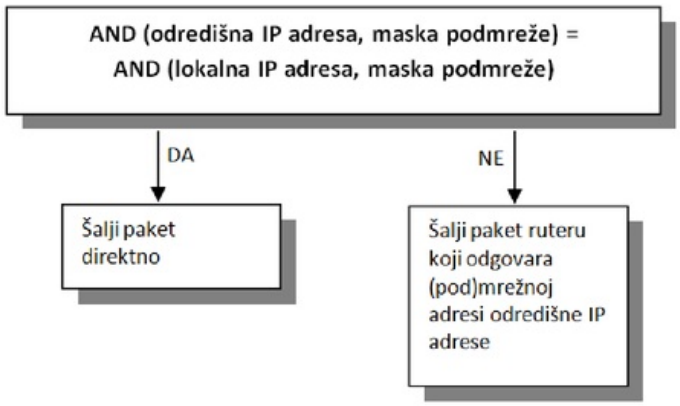
Dakle, tabela usmeravanja, na ruteru R određuje samo jedan korak duž putanje do odredišne mreže. Ruter ne poznaje celu putanju do odredišta.

Opšti oblik algoritma za usmeravanje možemo prikazati na sledeći način (Slika 2.26.):



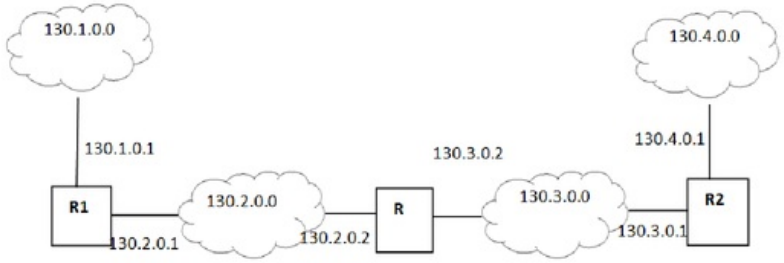
Slika 2.26. Algoritam za usmeravanje

U slučaju podmrežnog adresiranja algoritam usmeravanja možemo predstaviti na sledeći način (Slika 2.27.):



Slika 2.27. Algoritam za usmeravanje za podmrežno adresiranje

Kada ruter dobije paket on pokušava da pronađe u tabeli usmeravanja kojom putanjom treba da prosledi datagram. Ako se u tabeli usmeravanja ne pojavi ni jedan put, paket se šalje na podrazumevanu adresu.



Slika 2.28. Primer rutiranja

Na primeru sa slike (slika 2.28.) imamo četiri mreže koje su povezane sa tri rutera. Ruter R je povezan na dve mreže sa IP adresama 130.2.0.0 i 130.3.0.0 tako da ima dve IP adrese 130.2.0.2 i 130.3.0.2. Ovim mrežama ruter R može direktno da prosledi podatak.

Odr. adresa	Skok
130.2.0.0	direktno
130.3.0.0	direktno
130.1.0.0	130.2.0.2
130.4.0.0	130.3.0.2

Tabela 2.6. Tabela rutiranja

Ako je paket namenjen računaru u mreži 130.1.0.0 ruter R ga usmerava na adresu interfejsa 130.2.0.2 prema ruteru R1 koji zatim direktno prosleđuje paket.

Tabela usmeravanja na ruteru R može da izgeleda kao u tabeli 2.6.

Protokoli koji obezbeđuju podršku za mrežni nivo OSI modela nazivaju se rutabilni (routed), odnosno, usmerivi protokoli. Primeri takvih protokola su IP, IPX/SPX i Apple Talk. Ne-usmerivi protokoli ne obezbeđuju podršku za mrežni nivo. Primer ne-usmerivog protokola jeste NetBEUI.

Osnovni uslov da bi protokol bio usmeriv jeste da ima mogućnost da adresira mrežu (pomoću mrežne adrese) kao i svaki uređaj u mreži.

Ruting protokoli (protokoli za rutiranje) određuju putanje kojim usmerivi protokoli dolaze do svog odredišta[77]. Osnovni zadatak ruting protokola jeste razmena informacija koje su neophodne za pravilno rutiranje. Ruting protokoli popunjavaju tabelu usmeravanja.

Primeri ruting protokola su:

- Routing Information Protocol (RIP)
- Interior Gateway Routing Protocol (IGRP)
- Enhanced Interior Gateway Routing Protocol (EIGRP)
- Open Shortest Path First (OSPF)

Da bi se izvršilo konfigurisanje rutera korisnik mora da pristupi korisničkom interfejsu rutera. Ruter može da se konfiguriše sa različitih lokacija:

- pomoću konzolnog terminala (računar povezan na ruter pomoću konzolnog porta)
- pomoću modema (auxiliary port)
- pomoću Virtualnih Terminala
- pomoću TFTP servera

Ruter ima dva nivoa pristupa komandama:

1. korisnički mod: dozvoljava proveru statusa rutera

2. privilegovani mod: omogućava promenu konfiguracionih parametara

Kada se korisnik prvi put uloguje na ruter on će se nalaziti u korisničkom modu. Da bi korisnik imao pristup punom skupu instrukcija neophodno je da uđe u privilegovan mod. To se postiže naredbom enable kojom se prompt korisničkog moda > menja u prompt privilegovanog moda #.

Iz privilegovanog moda korisnik može da pristupi sledećim modovima:

- globalni konfiguracioni
- interface
- router...

Da bi se izvršilo konfigurisanje parametara rutera potrebno je koristiti komandu configure terminal u privilegovanom modu. Nakon upotrebe ove komamande ruter ulazi u globalni konfiguracioni mod, a prompt se menja u (config)#.

Da bi se korisnik vratio na prethodni mod rada treba da koristi komandu exit.

Korisnički mod ----------	`Router>`
Privilegovan mod ----------	`Router#`
Globalni konfiguracioni mod ----------	`Router(config)#`

Slika 2.29. Radni modovi rutera

Svaki ruter ima ime koje je moguće promeniti u globalnom konfiguracionom modu pomoću komande:

```
Router(config)#hostname novo_ime
```

Nakon ove komande prompt menja oblik u:

```
novo_ime(config)#
```

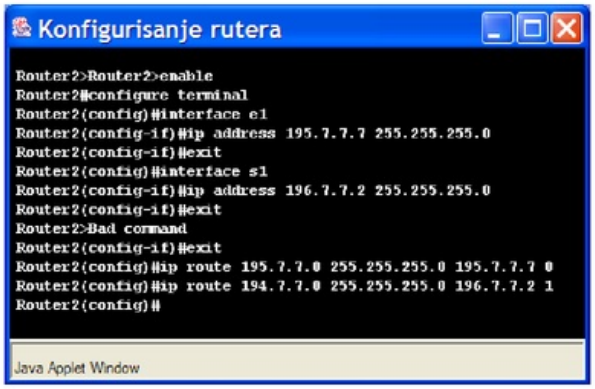
Da bi ruter rutirao pakete i prosleđivao ih preko odgovarajućih interfejsa do željene mreže on mora da ima podatke o putanjama u svojoj ruting tabeli. Ruting tabela može da se popunjava dinamički ili statički.

Nova ruta u ruting tabeli se dodaje komandom:

```
ip route mreža maska_podmreže adresa_interfejsa distanca
```

Administrativna distanca se izražava brojnom vrednošću od 0 do 255. Ako je ruter direktno povezan sa traženom mrežom, distanca je 0. Što je mreža udaljenija od rutera (zavisi od metrike), to je distanca veća.

Primer konfigurisanja statičkih ruta u simulatoru WnetSim prikazan je na slici 2.30.



Slika 2.30. Konfigurisanje rutera

3. Model mrežnog simulatora

Model svakog simulacijskog okruženja treba da što bliže odgovara stvarnom sistemu koji se simulira. U ovom poglavlju detaljnije se razmatra model mrežnog simulatora, kao i tipovi simulacijskih objekata i stanja potrebnih za modelovanje procesa u računarskoj mreži[20].

Modelovanje simulatora pomoću objekata stanja podrazumeva modelovanje svakog elementa stvarnog sistema: računar, hab, ruter, mrežna kartica, linkovi, protokoli, paketi itd.

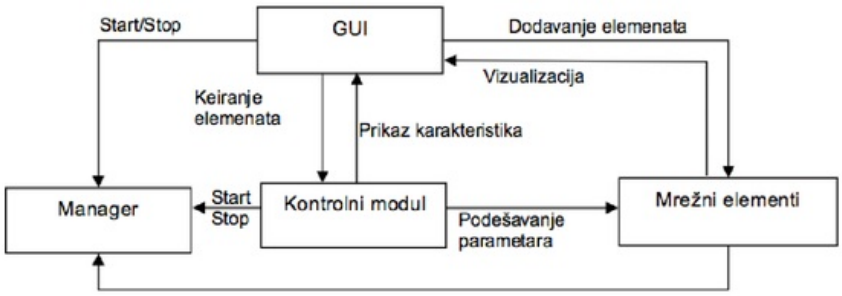
Primarna motivacija modelovanja mrežnog simulatora jeste u stvaranju okruženja koje može da ponudi izvođenje laboratorijskih vežbi za veliki broj studenata i da obezbedi vizuelne povratne informacije o posmatranom sistemu. Osnovni ciljevi modelovanja su:

- Projektovanje i implementacija simulatora koji može da se koristi u nastavi iz računarskih mreža[78].
- Kreiranje softverske aplikacije koja može da pruži brze povratne informacije o posmatranom sistemu i da da podršku pri donošenju odluka.
- Kreiranje grafičkog korisničkog interfejsa koji bi omogućio kreiranje proizvoljne mrežne topologije jednostavnom manipulacijom miša.

Model se zasniva na paketnom nivou koji omogućava praćenje individualnih paketa kroz mrežu. Na ovom principu se zasnivaju procesi u uobičajenim simulatorima, kao što je na primer ns-2. Model simulatora se zasniva na modularnim gradivnim blokovima kao što je prikazano na slici.

Osnovni modularni gradivni blokovi su (Slika 3.1.):

- Grafički korisnički interfejs
- Menadžer
- Mrežni elementi
- Kontrolna jedinica



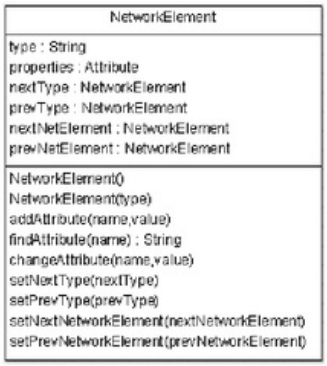
Slika 3.1. Blokovska struktura modela mrežnog simulatora

Objekat klase GUI pruža mogućnost editovanja željene računarske mreže. Objekat klase Controls omogućava kreiranje objekata klasa PC, Hub, Switch, UTP... i prikazivanje istih, kao i procesa koji se u njima odvijaju. Objekat klase Monitor omogućava prikaz trenutnih dešavanja u komponentama mreže. Objekat klase Manager pruža mogućnost upravljanja radom simulatora.

3.1. Mrežni elementi

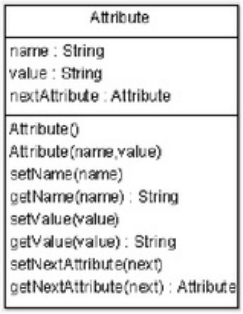
Mrežni element predstavlja bilo koji entitet koji šalje ili predaje pakete. To može biti krajnji sistem (korisnički računar) ili npr. web server, *router* u unutrašnjosti mreže, optički *switch* ili Ethernet *hub*. Svi ovi uređaji kreiraju i šalju ili primaju, odnosno prosleđuju pakete[79].

Simulaciono mrežno okruženje je kolekcija entiteta, a svaki mrežni element je generalizacija entiteta. Mrežni elementi, koji su komponente računarske mreže, su definisani osobinama i operacijama koje izvršavaju. Oni se modeluju objektima koji su instance klase NetworkElement (slika 3.2).



Slika 3.2. Osnovna klasa koja modeluje statičke osobine mrežnih elemenata

Statičke osobine objekata klase NetworkElement su definisane ulančanom listom čiju su elementi instance klase Attribute (Slika. 3.3.). Poseban tip mrežnih elemenata jesu portovi koji predstavljaju logičke konstrukcije i oni definišu pristupne tačke između funkcionalnih objekata i prenosnih veza[20].



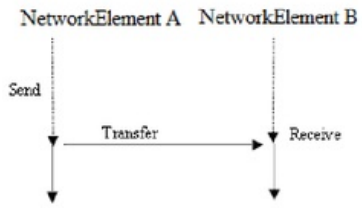
Slika 3.3. UML model klase Attribute

Funkcionalnost mrežnih elemenata se modeluje objektima koji se izvršavaju u formi nezavisnih niti. Ti objekti su aktivni mrežni elementi koji se modeluju kao Java niti (Listing 3.1.).

Listing 3.1: Aktivni objekti se realizuju kao Java niti

```
class ActiveNetworkElemen xtends
          NetworkElement implements Runnable {
      // ...
      Thread thread;
      public ActiveEntity() {
            // ...
            thread = new Thread(this);
            thread.start();
       }
       public void run(){
             // Definise se ponasanje aktivnog entiteta
             // ...
       }
}
```

Aktivni mrežni elementi komuniciraju asinhrono tako što jedan elemenat nezavisno poziva neku operaciju u drugom elementu i nastavlja svoju aktivnost. Drugi elemenat, kad bude spreman, prima poruku i takođe nastavlja svoju aktivnost (slika 3.4.).



Slika 3.4: Asinhrona komunikacija između mrežnih elemenata

3.2. Korisnički interfejs

Implementacioni detalji o procesu slanja paketa od jednog do drugog mrežnog elementa su transparentni. Aktivnosti kao što je kontrola pristupa prenosnom mediju, kreiranje signala za koji se šalju kroz prenosne medije, određivanje strukture paketa ili otkrivanje greške u prenosu na mrežnom interfejsu su aktivnosti za koje je odgovoran objekat korisničkog interfejsa. Sistem treba jasno da razdvoji akcije obrade paketa i proces predaje i prijema paketa. Zbog toga objekat koji predstavlja korisnički interfejs treba da ima nekoliko promenljivih stanja[81].

U svakom trenutku, izlazni interfejs je ili u procesu slanja paketa ili je slobodan i čeka paket za slanje. Ako je interfejs u procesu slanja, onda je prenosni medij zauzet i ne može da započne drugo slanje. Ovo stanje se modeluje jednom promenljivom tipa boolean.

Procesi obrade paketa se ne bave upravljanjem komunikacijskim kanalima, pa nisu ni svesni da li se paketi mogu poslati u nekom trenutku. Zbog toga interfejs mora biti u stanju da primi zahtev za slanje od mrežnog elementa i u slučaju kada je prenosni medij nedostupan. Ovo se ostvaruje implementacijim reda za čekanje u svakom interfejsu.

3.3. Kontrolni modul

Krajnji korisnik i generator podataka u mreži jeste aplikacija, pa tip aplikacija u simulacijskom okruženju ima značajan efekat na ukupne performanse mreže.

Postoji nekoliko načina da se kontrolišu realni aplikativni modeli u simulacijskom okruženju. Prvi način je da se definiše ponašanje aplikacije u analitičkom smislu. U tom slučaju bi se podaci generisali slučajnom poretku, a model se jednostavno zasniva na slučajnim promenljivama za **„uključeno“** i **„isključeno“** i brzinu prenosa podataka tokom perioda „uključeno“.

Drugi način jeste posmatranje mreže i kreiranje empirijske raspodele na osnovu posmatranja ponašanja stvarnih aplikacija. Ovakav pristup vodi realnom modelu performansi aplikacije, koji je ograničen na performanse aplikacije u mreži koja se koristi za nadzor.

Treći pristup podrazumeva stvarne aplikacije u mrežni simulator. Ovaj način je pogodan kada aplikacija ne odgovara na korisničke zahteve. Primer ovog pristupa jesu protokoli rutiranja koji se zasnivaju na stanju veze[82].

4. Mrežni simulatori u nastavi iz računarskih mreža

4.1. Razlozi za izradu i upotrebu mrežnih simulatora

Analiza performansi i evaluacije kod umrežavanja računara igra važnu ulogu i predstavlja jedan od kompleksnijih aspekata u ovoj oblasti. Udžbenici iz ove oblasti ukazuju na trend ka većoj integraciji u analizi performansi kurseva iz ovog polja.

Istraživanja u oblasti mreža sa tehnologijom Internet protokola (IP) zasnivaju se na analitičkim metodama, simulaciji, eksperimentima i merenjima. Dok merenja i eksperimenti obezbeđuju sredstva za ispitivanje realnih mreža, simulacija i analiza su ograničene na ispitivanje konstruisanih, apstraktnih modela.

Ograničenje metoda merenja i eksperimentalnih metoda je u tome što se oni mogu primenjivati samo na postojeći sistem ili delimično nova okruženja. Sa druge strane, iako su analitički metodi ključni za suštinsko razumevanje ponašanja mreže, postoji rizik primene toliko pojednostavljenih modela da se gube bitne karakteristike ponašanja mreže. Simulacija je komplementarna sa analizom zbog toga što omogućava verifikaciju ispravnosti analize i ispitivanje kompleksnih modela koje bi bilo teško ili nemoguće rešavati analitički.

Usled heterogenosti i brzih promena IP tehnologije, ne postoji jedinstven i ograničen skup scenarija simulacije dovoljan da pokaže da će se predloženi protokol ili projektovani sistem dobro ponašati u uslovima stalnog razvoja mreže. Umesto toga, simulacije imaju bitnu ulogu u ispitivanju različitih aspekata predloženog rešenja i razumevanju dinamičkog ponašanja mreže. Pored problema da se definiše relevantan model, mogu se javiti teškoće u verifikaciji da simulator tačno implementira željeni model.

Zbog toga je preporučljivo koristiti poznate i proverene alate, a jedan od najboljih načina rešavanja problema verifikacije rezultata simulacije je da simulatori i pridruženi skriptovi budu slobodno dostupni, tako da drugi istraživači mogu sami lako da provere efekte promene polaznih pretpostavki o mrežnom scenariju.

Korišćenje mrežnih simulatora ima pozitivan efekat i naglašava analizu performansi u osnovnim kursevima računarskog umrežavanja. Upotreba mrežnih simulatora ima nekoliko prednosti. Prvo, dobro konstruisan simulator izbegava ograničavajuće pretpostavke zahtevane mnogim analitičkim modelima koji ne zahtevaju skupi hardver. Takođe, oni omogućavaju studentima iskustva sa širim spektrom mrežnih sistema kroz simulaciju, pre nego što bi to bilo moguće sa stvarnim hardverom [21].

Vežbe, koncipirane za rad sa simulatorima, mogu da se razvrstaju u nekoliko kategorija. Uvodne vežbe su donekle usmerene na jačanje koncepata predstavljenih na časovima predavanja, pa su aktivnosti na vežbama uglavnom usmerene ka ilustraciji konkretnih pojava ili ponašanja, kao i pružanju iskustva sa merenjem i proučavanjem željenih karakteristika sistema. Međutim, srednje teške i napredne vežbe obuhvataju kompleksnije tematske probleme.

Kao rezultat, studenti su dužni da dizajniraju i implementiraju rešenja, da izvode eksperimente, pribavljanja podataka potrebnih za završetak projektovanja ili za merenje različitih karakteristika. Sve ove aktivnosti se najefikasnije vrše korišćenjem odgovarajućeg simulatora računarske mreže [22].

Simulatori su posebno važni za profesionalce koji teže da dobiju i nadograde svoj profesionalni sertifikat.

Istraživanja u oblasti računarskih mreža sa IP (Internet Protokol) tehnologijom se bazira na analitičkim metodama, simulacijama, eksperimentima i merenjima. Dok merenja i eksperimentisanje obezbeđuju sredstva za testiranje realnih mreža, simulacija i analiza su ograničeni na testiranje dizajniranih, apstraktnih modela. Ograničavajuće metode merenja i eksperimentalnih metoda su te što se one mogu primeniti samo na postojeći sistem ili u delimično novim sredinama. Sa druge strane, iako su analitičke metode od suštinskog značaja za razumevanje mrežnog ponašanja, postoji rizik od primene pojednostavljenih modela koji zanemaruju esencijalne karakteristike mreže.

Simulacija je komplementarna sa analizom jer omogućava verifikaciju ispravnosti analize i ispitivanja složenih modela koje bi bilo teško ili nemoguće analitički rešiti. Stoga je preporučljivo koristiti poznate i proverene alate, a jedan od najboljih načina za rešavanje problema verifikacije rezultata simulacije je da simulatori i povezane skripte budu besplatno dostupni, tako da drugi istraživači mogu lako testirati efekte promena polaznih pretpostavki o mogućim scenarijima u mreži.

Ključno svojstvo koje modeliranje i simulaciju IP mreža uglavnom čini teškom je heterogenost [23], koja se ogleda u nekoliko aspekata:

- topologija mreže,
- izbor simuliranih aplikacija,
- generisnje saobraćaja,
- dinamičko rutiranje,
- razlike u verzijama protokola itd.

To znači da ne postoji prethodno definisan skup scenarija simulacije koji je dovoljan za proveru predloženih rešenja. Problem je delimično rešen sa identifikacijom invarijantnih karakteristika IP saobraćaja, procesom donošenja poziva, trajanjem sesija, kao i redovnom aproksimacijom modela i ispitivanjem ponašanja mreže u uslovima promene postavljenih parametara simulacija u širokom opsegu. Primarna motivacija u upotrebi i dizajnu mreže simulatora je bila da ponudi manualno lako kontrolisane vežbe umrežavanja koje bi pružale vizuelno povratne informacije. Osnovni ciljevi dizajna umrežavanja se ogledaju u obezbeđivanju okruženja za projektovanje i rad praktičnih vežbi za veliki broj studenata [24].

Rezultat ovog rada bi trebalo da bude i pomoć budućim korisnicima simulatora u odabiru simulatora, u cilju postizanja boljih znanja u ovoj naučnoj disciplini.

U nastavku rada dat je pregled sledećih mrežnih simulatora: OPNET, SSFNet, ns-2, ns-3, GNS3, GTNetS, GloMoSim, KivaNS, MIMIC, CNET, NetSim, Nessi i WnetSim.

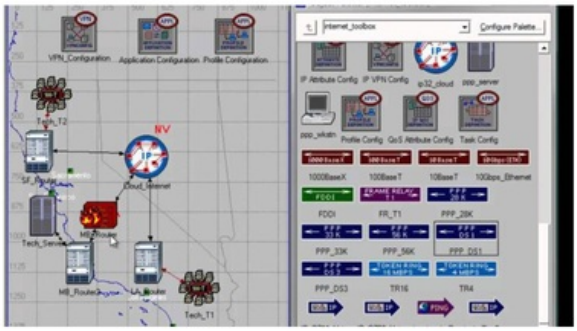
4.2. Pregled mrežnih simulatora

4.2.1.OPNET IT Guru Academic Edition

OPNET IT [25] Guru Academic Edition je softver za upravljanje mrežama i aplikacijama koji omogućava:

- ispitivanje već postojeće mreže da bi se pronašle i otklonile greške u njoj,
- proveravanje mrežne konfiguracije pre njene izgradnje,

- planiranje izdržljivosti i kapaciteta mreže,
- planiranje razvoja aplikacija i svega ostalog što je vezano za mrežnu tehnologiju.



Slika 4.1. OPNET IT Guru

OPNET omogućava bolje razumevanje suštine pojmova umrežavanja i pomaže da se efikasnije rešavaju problemi i upravljanje realnom mrežnom infrastrukturom[83]. On omogućava prikupljanje informacija, koje su dobijene analiziranjem podataka dobijenih na osnovu iskustva i poznavanja mrežnih uređaja (koji su isprogramirani), protokola, aplikacija i servisne opreme (Slika 4.1).

Ovo omogućava korisnicima:

- otkrivanje postojećih problema;
- otkrivanje problema koji nisu poznati;
- predviđanje problema u buducnosti i njihovo izbegavanje;
- smanjenje operativnih troškova i troškova infrastrukture.

OPNET IT Guru pruža virtuelno mrežno okruženje (VNE – *Virtual Network Environment*) tako što oponaša način rada celokupne mreže uključujuci rutere, svičeve, protokole, servere i individualne aplikacije[84].

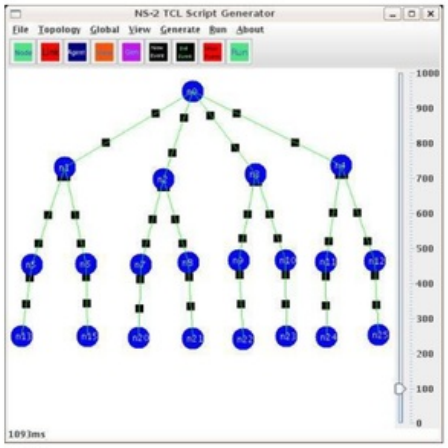
4.2.2. SSF mrežni simulator

SSF (Scalable Simulation Framework) simulator je otvoreni alat za simuliranje kompleksnih sistema [26]. Razvijen je zajednički od strane Rutgers University i Dartmouth University, napisan u Java jeziku, a takođe postoji i C++ varijanta. U okviru ovog alata razvijen je SSFNet, koji predstavlja skup Open-source Java modela za različite protokole, mrežne elemente i omogućava objektno-orijentisanu podršku za simulaciju multiprotokolnih i multidomenskih okruženja. Razvoj simulatora je prilagođen radu na multiprocesorskim sistemima, a moguće je i paralelno izvršavanje na različitim operativnim sistemima. Postavljanje i upravljanje simulacijom vrši se korištenjem DML (Domain Modeling Language) jezika, odnosno jezika zasnovanog na XML-u (eXtensible Markup Language).

Distribucija ovog simulatora sadrži i izvorni kod sa dve verzije: SSF.OS - koji je specijalizovan za modeliranje komponenti operativnog sistema host-a (posebno protokola) i SSF.Net - za modeliranje topologije mreže, kreiranje čvorova i konfiguraciju linkova. Ova distribucija takođe sadrži i kod za upravljanje saobraćajem, gotove modele protokola i host-ova sa različitim generisanim saobraćajem, kao i modul za prikupljanje i obradu rezultata simulacije (CERN Colt). Zbog svoje mogućnosti obrade velike količine podataka, ovaj simulator je pogodan pre svega za simulaciju velikih i kompleksnih sistema (kao što je Internet), te za praćenje rada u mreži u dužim vremenskim intervalima.

4.2.3. NS2 simulator

Simulator NS2 (*Network Simulator ver. 2*) pripada klasi simulatora vođenih događajima u diskretnom vremenu. Razvijen je polovinom devedesetih godina u okviru DARPA VINT (*Virtual InterNetwork Testbed*) projekta, realizovanog u saradnji nekoliko vodećih američkih univerziteta. Razvijen je metodama objektno-orijentisanog programiranja, na programskom jeziku C++, sa korisničkim interpreterom OTcl (*Object-oriented Tools Command Language*) skriptova [27] (Slika 4.2.).



Slika 4.2. NS-2 Script Generator

OTcl je programski jezik sa jednostavnom sintaksom, koji se koristi za razvoj samostalnih aplikacija ili je ugrađen u druge aplikacione programe. NS2 je od početka koncipiran kao simulator sa slobodno dostupnim izvornim programima i slobodnom distribucijom, što je omogućilo brzu ekspanziju, efikasan razvoj i kontinuiranu nadgradnju.

NS2 se primenjuje u većini američkih i zapadno-evropskih naučno-istraživačkih institucija za potrebe studijskih istraživanja u oblasti LAN i IP WAN mreža, mobilnih i ad hoc IP mreža i satelitskih komunikacija. Raspoloživ je za različite verzije operativnih sistema Unix (Sun OS, free BSD, Linux) i Windows (95/98/2000/NT/XP). NS2 je multi-protokolski simulator koji implementira različite aplikacione protokole, transportne protokole, protokole rutiranja, protokole pristupa medijumu (MAC) u lokalnim računarskim mrežama, protokole za mobilne IP mreže i protokole za satelitsku komunikaciju.

U simulatoru su implementirani mehanizmi kvaliteta servisa, kao što su algoritmi za opsluživanje paketa i upravljanje redovima (baferima), diferencirani servisi, multiprotokolska komutacija labela (MPLS) i dr. NS2 takođe obuhvata biblioteke za generisanje izvora saobraćaja i topologije mreže, animator mreže, kao i interfejs za emulaciju koji omogućava interakcije simulatora sa saobraćajem u realnoj mreži[85].

Osnovne mrežne komponente simulatora su čvor, link i paket. Funkcije čvora su obrada paketa na osnovu odgovarajućih zaglavlja (adresa odredišta, adresa izvora, tip protokola i dr.) i prosleđivanje paketa sledećem čvoru, na osnovu tabele rutiranja i zadatog

protokola rutiranja. U zavisnosti od načina implementacije kvaliteta servisa, u NS2 čvoru mogu da budu implementirani različiti mehanizmi za upravljanje redovima (baferima).

Čvorovi mreže su međusobno povezani jednosmernim ili dvosmernim linkovima, za koje se definiše kašnjenje usled propagacije i prenosa paketa po linku. Linkovima se dodeljuju cene – u opštem slučaju po jedna cena za svaki smer komunikacije. Takođe je moguće simulirati ispad linkova u zadatim intervalima simulacionog vremena. NS2 paket je struktura podataka, koja se sastoji od skupa zaglavlja različitih protokola i opcionog polja sa korisničkim podacima.

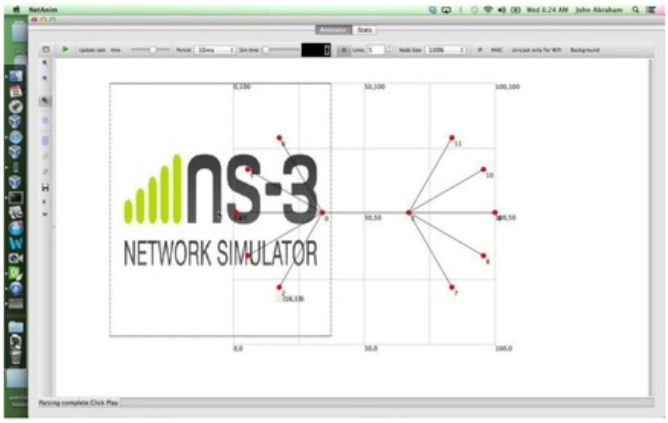
Kompleksnost IP arhitektura, topologija i saobraćaja čine simulaciju izuzetno perspektivnim alatom za istraživanje realnih problema vezanih za dinamiku saobraćaja. Multiprotokolski simulator NS2 sa pridruženim alatima pruža veliki skup mogućnosti za definisanje različitih scenarija simulacije IP mreža. Jedna od njegovih velikih prednosti je u tome što je slobodno dostupan. Za istraživače je atraktivan i zbog mogućnosti dodavanja novih karakteristika (predlog novih algoritama, protokola i dr.).

4.2.4. NS3 simulator

NS-3 je mrežni simulator razvijen 2006. godine kako bi zamenio NS-2 simulator (slika 4.3.). Cilj NS- 3 projekta je stvoriti alat koji će nakon prestanka početnog finansiranja nastaviti razvijati akademska zajednica i zainteresovane kompanije. Mrežni simulator NS-3 zasniva se na diskretnim događajima[86]

.

NS-3 je klasična C++ biblioteka koja se može prevesti pomoću GCC i Cygwin prevodioca (*compiler)*. Simulacijski programi su izvršne C++ datoteke. Pri pisanju simulacija moguće je pridružiti Python programski kod što omogućava direktno kombinovanje sa istraživačkim softverom napisanim u Python. Razlikuje se od NS-2 po novom softverskom jezgru koji poboljšava sklabilnost, modularnost, stil kodiranja i dokumentaciju po Internetskim čvorovima koji bolje reprezentuju stvarne računare, po mogućnosti softverske integracije s ostalim softverima i po podršci za virtualizaciju [28].

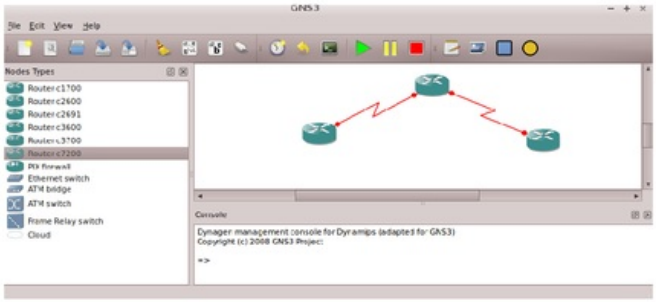


Slika 4.3. NS-3 NetAnim

Proces od modeliranja do simulacije sastoji se od implementacije protokolnog modela, postavljanja scenarija simulacije, pokretanja simulacije i konačno analize rezultata. Tipično se koristi Linux operativni sistem, ali može se koristiti i Windows pomoću Cygwin okoline. Osnovni pojmovi od kojih se kreiraju modeli su čvor (Node), aplikacija *(Application)*, kanal *(Channel)* i mrežni uređaj (*NetDevice*).

4.2.5. GNS3 simulator

Graphical Network Simulator (GNS3) jeste višeplatformski softver za simulaciju Ciscovih mrežnih uređaja (Slika 4.4.). Podržava operativne sisteme Linux, Mac OS X i Windows. *GNS3* koristi prave slike operativnog sistema *Cisco IOS* za simulaciju uređaja, zbog čega prvo moraju da se obezbede odgovarajuće slike *(images)* diskova da bi *GNS3* uopšte mogao da se koristi[29].



Slika 4.4. GNS3 mrežni simulator

GNS3 se nalazi u riznicama svih većih distribucija. Ukoliko se kojim slučajem tamo ne nalazi, sa sajta projekta moguće je preuzeti i instalirati izvorni kôd. Korisnici Arch Linuxa mogu da nađu *GNS3* u riznici AUR i instaliraju ga komandom *yaourt-S gns3*, a menadžer paketa će se postarati o svim međuzavisnostima. *GNS3* se oslanja na dva bitna paketa: *Dynamips,* koji je Cisco IOS emulator, i *Dynagen,* tekstualni front-end za *Dynamips. GNS3* nudi mogućnost simuliranja Cisco rutera (c1700, c2600, c2691, c3600, c3700, c7200), Cisco PIX firewalla, Cisco ASA firewalla, Juniper rutera, ATM Bridge/Switcha i Frame Relay Switcha. Iako su tu uvršćeni i Ciscovi switchevi, simulacija ovih uređaja trenutno nije moguća[30].

Što se konfiguracije tiče, po završenoj instalaciji i pokretanju programa neophodno je podesiti određene parametre. Pošto se na uređaje povezuje *telnet* konekcijom, potrebno je izabrati željeni emulator terminala koji će se koristiti za tu operaciju. U opciji *Dynamips* potrebno je staviti putanju do izvršnog fajla *dynamips* (/usr/bin/dynamips), zatim, definisati radni direktorijum i osnovne portove (podrazumevano port je 7200). Opcija *Capute* omogućava podešavanje alata *Wireshark.* Pošto se konfiguriše *GNS3,* neophodno je definisati slike IOS-a koje će koristiti ruteri koji se simuliraju.

GNS3 interfejs podeljen je na nekoliko sekcija. Leva strana programa sadrži uređaje koje je moguće dodati, dok je centralni deo najveći i tu se dodaju uređaji se simuliraju. Dodavanje uređaja vrši se prevlačenjem željenog uređaja s leve strane na centralnu površinu. Desna strana sadrži informacije o statusu dodatih uređaja (hostname je, u zavisnosti od stanja, ili crveno ili zeleno). Gornja traka sadrži neke korisne opcije kao što su dugme za dodavanje

veze (Ethernet, WAN), uvoz/izvoz konfiguracionih fajlova, povezivanje sa uređajima preko emulatora terminala, pokretanje i zaustavljanje uređaja. Deo tih komandi moguće je dobiti i desnim klikom miša na neki od uređaja. Postoji mogućnost dodavanja teksta i različitih geometrijskih oblika.

Što se hadverske zahtevnosti tiče, *GNS3* poprilično zahteva procesorsku snagu, dok je memorija zauzeta u granicama normale.

GNS3 je odličan simulator mrežnih uređaja. S obzirom na to da i *Cisco Packet Tracer* služi za simuliranje *(Boson NetSim* nema Linux klijent), *GNS3* je odlično rešenje za simulaciju Cisco mrežnih uređaja[87].

4.2.6. GTNetS simulator

GTNetS *(Georgia Tech Network Simulator)* je razvijen na *Georgia Tech – Department of Electrical and Computer Engineering*, kao objektno-orijentisani alat napisan u potpunosti u C++ programskom jeziku [31]. Dizajn ovog simulatora veoma blisko odgovara dizajnu stvarnog skupa mrežnih protokola i mrežnih elemenata. Alat je dizajniran tako da radi u distribuiranom okruženju, što dovodi do veće skalabilnosti.

Da bi kreirao simulaciju korišćenjem GTNetS, korisnik treba da kreira C++ glavni program, koji kreira potrebne objekte za opis i izvršenje simulacije. Program se kompajlira standardnim C++ kompajlerom i linkuje sa bibliotekama koje sadrže simulacijske modele. Da bi se pokrenula simulacija izvršava se rezultirajuća binarna datoteka[88].

4.2.7. GloMoSim simulator

GloMoSim (Global Mobile System Simulator) je skalabilno simulaciono okruženje za mobilne i bežične mreže, razvijen je u Parallel Computing Laboratory na UCLA univerzitetu u Los Anđelesu, Kalifornija, Sjedinjene Američke države. On je simulator diskretnih događaja razvijen korišćenjem PARSEC (for PARallel Simulation Environment for Complex systems) programskog jezika [32].

Kao i većina mrežnih sistema i GloMoSim je dizajniran da koristi pristup zasnovan na slojevima sličnim OSI mrežnoj arhitekturi sa standardnim API (Acces Point Interface) između njih. Jednostavan čvor se definiše između različitih simulacionih slojeva. Dizajn sa slojevima pristupačan je jer stvara modularno okruženje. Pod tim se podrazumeva da se protokoli i modeli na različitim slojevima tretiraju nezavisno i mogu se modelovati ili zameniti bez uticaja na protokole i modele na drugim slojevima. Modulacioni dizajn takođe omogućava korisnicima da razvijaju i implementiraju nove protokole na različitim slojevima tako da dizajn tih protokola odgovara standardnim API između slojeva. GloMoSim je skalabilni simulator pa ima mogućnost simulacija sa hiljadama čvorova.

Kada se instalira simulator dobiju se u folderu `/glomosim` sledeći podfolderi:

`/application` - u kom se nalaze kodovi za aplikacioni sloj

`/bin` - u kom se nalaze egzikucioni i ulazno/izlazni fajlovi

`/doc` - u kom se nalazi dokumentacija

`/include` - u kom se nalaze zajednički include dokumenti

`/java gui` - u kom se nalazi tool za vizuelizaciju

`/main` - sadrži osnovne *framework* dizajn programe

`/network` - u kom se nalaze kodovi za mrežni sloj

`/radio` - u kom se nalaze kodovi za radio sloj

`/scenarios` - u kom se nalaze primeri nekih scenarija

`/tcplib` - u kom se nalaze biblioteke za TSR

`/transport` - u kom se nalaze kodovi za transportni sloj

Kao što je već rečeno GloMoSim je dizajniran po principu slojeva sa standardnim API između dva različita sloja. Da bi se jednostavno implementirao novi protokol, svaki sloj je predstavljen kao poseban entitet.

Na taj način uključeni su modeli za kanal, radio, MAS, mrežni, transportni i više slojeve.

Modeli i protokoli koji su podržani na različitim slojevima prikazani su u Tabeli 4.1.:

Sloj	Protokol
Mobilnost	Random Wayponit, Trace Based
Radio propagacija	Two ray, Free space

Radio model	Noise Accumulating
Model primanja paketa	SNR, BER sa BPSK/QPSK modulacijom
MAC	CSMA, IEEE 802.11, MACA
Rutiranje (mrežni sloj)	IP: AODV, Bellman-Ford, DSR, Fisheye, LAR, ODMRP, WRP
Transportni sloj	TCP, UDP
Aplikativni sloj	CBR, FTP, HTTP, Telnet

Tabela 4.1. Modeli i protokoli

GloMoSim podržava mobilnost čvorova. Čvorovi se mogu kretati po modelu Random Waypoint Mobility Model (RWPM).

GloMoSim ima nekoliko ugrađenih tajmera na različitim slojevima. Uopšteno pomoću tajmera se ostvaruju događaji koji se dešavaju za određeni čvor. Fajl `/glomosim/include/structmsg.h` definiše događaje u GloMoSim, a korisnik može da definiše svoje događaje i tajmere.

Paketi i poruke koji prolaze kroz slojeve moraju u sebi imati informacije o tome sa kog sloja su poslati i kom sloju se šalju. Slanje paketa i poruka ostvaruje se zakazivanjem događaja (Scheduling Events) i svakom događaju mora biti dodeljen njegov jedinstveni "Event type ID".

U simulatoru postoje dve vrste poruka:

- Paketi (Packets/ Cell Messages): paketi koji se razmenjuju između čvorova (Inter-node packets) ili paketi koji se razmenjuju između slojeva ((Inter-layerpackets).
- Poruke koje ne sadrže mrežne pakete (Non-Packet Messages): poruke koje opisuju događaje između slojeva (Inter-layer event messages) i događaji koji predstavljaju samo postavljene tajmere (Self scheduled (timer) events).

Glavna prednost svih slobodno dostupnih simulatora jeste mogućnost dodavanja novih ili izmena postojećih protokola i algoritama za potrebe. Osnovni nedostatak GloMoSim simulatora, u odnosu na komercijalne simulatore, jeste nedostatak literature za korišćenje simualtora, loš grafički interfejs i nemogućnost kvalitetne vizuelizacije simulacije[89].

Prednost je to što je simulator pisan u Parsec simulacionom jeziku, koji je jedan od osnovnih programskih jezika, i široko je rasprostranjen. Međutim, samo kompajliranje i

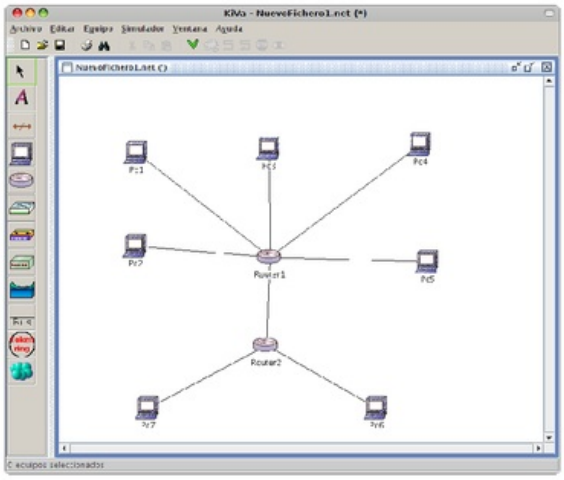
povezivanje novih ili izmene postojećih protokola zahtevaju odlično poznavanje ne samo materije i protokola već i programiranja u simulacionim paketima. Još jedan od nedostataka je to što je simulator razvijen 1999. godine nakon čega ga je preuzela firma ScalabileNetworks i razvijala komercijalnu verziju QualNet, pa sam GloMoSim simulator nema razvijene najnovije standarde i protokole.

Razvijeni simulator je otvorenog tipa i pogodan je za korišćenje i primenu u naučno-istraživačke i edukativne svrhe. Implementirani protokol rutiranja i metrike zasnovane su na implementiranju posebno pisanog novog programa i njegovom povezivanju sa simulatorom.

Na taj način efikasno je kombinovan postojeći open source simulator sa razvojem sopstvenih programa za simulaciju. Izborom odgovarajućih parametera simulacije moguće je izvršiti različite analize mreža sa različitim brojem čvorova i različitim saobraćajnim optrerećenjima. Postojeći simulator, i protokoli koji su u njemu implementirani, mogu se nadograditi saglasno sa definisanim temama i ostvarenim rezultatima istraživanja

4.2.8. KivaNS

KivaNS je besplatna aplikacija otvorenog koda programirana u Java programskom jeziku, koji je fokusiran na simulaciju računarskih mreža povezanih sa TCP - IP arhitekturom. Ova aplikacija je razvijena kao virtuelna laboratorija koja omogućava studentima obavljanje eksperimenata iz IP rutiranja, bez korišćenja složene i skupe opreme. KivaNS je ocenjena od strane studenata Univerziteta u Alikanteu uz pomoć praktičnih slučajeva za rešavanje problema sa ARP, IP, ICMP protokolima [33].



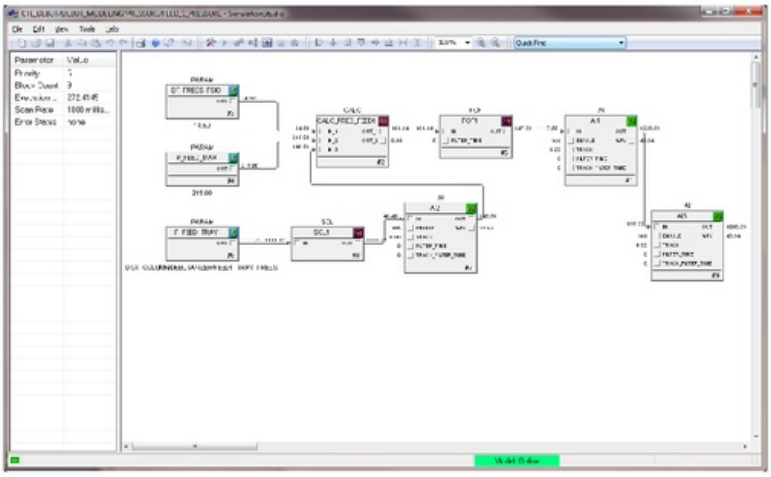
Slika 4.5. KivaNS simulator

Glavni ciljevi KivaNS simulatora je da simulira kako radi IP, i posebno, da proučava različite tehnike za rutiranje paketa podataka kroz različite mreže. KivaNS takođe uključuje simulaciju pomoćnih protokola, kao što su ARP, ICMP pored IP, i oponaša funkcionisanje osnovne link tehnologije kao što su PPP , Ethernet , ili uključen Ethernet (Slika 4.5.).

Arhitektura KivaNS je dizajnirana da ima simulaciju i usluge korisničkog interfejsa dobro diferencirane. KivaNS se sastoji od dva glavna bloka, oba implementirana u Javi . Prvi blok je API koji nudi simulacione mašine za mreže za prenos podataka. Drugi blok je kompletan grafički korisnički interfejs, koji omogućava korisniku da dizajnira mrežama podataka šeme, kao i da ih simulira i analizira pomoću API bloka na transparentan način[34].

4.2.9. MIMIC

MIMIC Simulator kreira realno globalno laboratorijsko okruženje, sa preko 50.000 uređaja, uz minimalne troškova fizičke opreme. Pruža interaktivnu laboratoriju za osiguranje kvaliteta, razvoj, prodajne prezentacije, evaluaciju, raspoređivanje i osposobljavanje aplikacija za upravljanjem preduzećem[35].



Slika 4.6. MIMIC Simulation Studio

Korisnici kreiraju prilagodljivo virtuelno okruženje sa simuliranim ruterima, habovima, sviječevima, ViFi/ViMAKS/LTE uređajima, kablovskih modema, servera i radnih stanica.

MIMIC Simulator paket obuhvata:

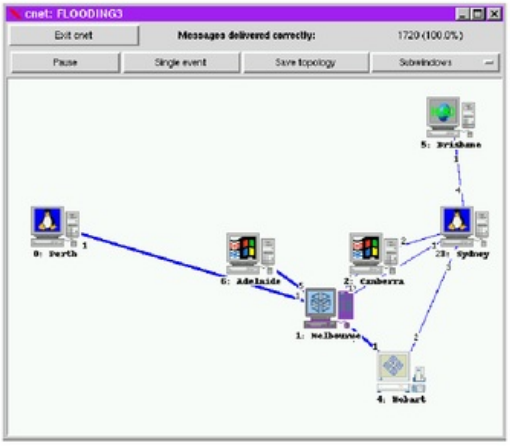
- MIMIC SNMP Simulator - SNMPv1, SNMPv2c, SNMPv3 and SYSLOG Simulation
- MIMIC NetFlow Simulator - NetFlow, IPFIX, sFlow, SYSLOG along with SNMP
- MIMIC sFlow Simulator - sFlow, NetFlow, IPFIX, SYSLOG along with SNMP
- MIMIC IOS/JUNOS/Telnet Simulator - Cisco IOS, Juniper JUNOS, Telnet, SSH, SYSLOG
- MIMIC Server Simulator - SMTP, DNS, FTP i SNMP
- MIMIC IPMI Simulator - IPMI and SNMP
- MIMIC Cable Modem Simulator - SNMP, DHCP, TFTP, ToD, MGCP

Ovaj svestrani paket softverskih alata se obično koristi za:

- Razvoj i testiranje;
- Evaluacija;
- Prodaja;
- Obuka okruženja.

4.2.10. CNET

CNET je open-source mrežni simulator razvijen na Univerzitetu Zapadne Australije i dizajniran je za učenje (Slika 4.7.). To je mrežni simulator koji omogućava eksperimentisanje sa različitim slojevima podataka, mrežnim slojevima, rutiranjem i transportnim slojevima mrežnih protokola u mrežama koje sadrže bilo koje kombinacije linkova širokopojasnih mreža (WAN), lokalnih mreža (LAN), ili bežičnih mreža (WIFI). Povezan sa OSI/ISO mrežnim referentnim modelom, CNET pruža Aplikativne i Fizičke slojeve[36].

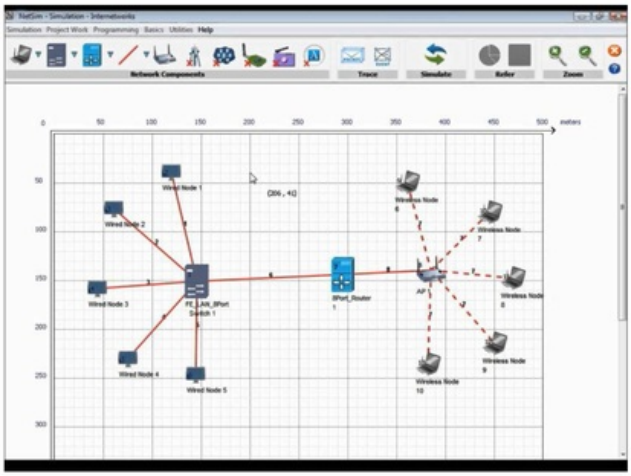


Slika 4.7. CNET mrežni simulator

CNET obezbeđuje pregršt osnovnih osobina za osnovno umrežavanje, kao što je prenos i prijem, uvod u verovatnoću grešaka, tačno vreme i statističke funkcije, mrežne kartice različitih tipova i proizvođača, i mogućnost programiranja mobilnih čvorišta, ali ne sadrži sve karakteristike u osnovnoj verziji. Moguće je koristiti CNET za programiranje i razvoj protokola koji su sastavni deo TCP/IP paketa, ali CNET ne obezbeđuje sve vrste protokola u okviru svog korisničkog interfejsa.

4.2.11. NetSIM

NetSIM je mrežni simulator koji se koristi za labaratorijsko eksperimentisanje i istraživanje (Slika 4.8.). Boson NetSIM Network Simulator je aplikacija koja simulira Cisco Systems mrežne uređaje i softver i dizajniran je da pomogne korisniku savladavanje Cisco IOS komandne strukture. NetSIM je proizvod namewen Windows platformi koji simulira širok spektar Cisco rutera. NetSIM podržava više protokola rutiranja, uključujući Rip, IGRP, EIGRP, BGP i OSPF. Podržava različite WAN/LAN protokole, uključujući PPP/CHAP, ISDN i Frame Relay[37].

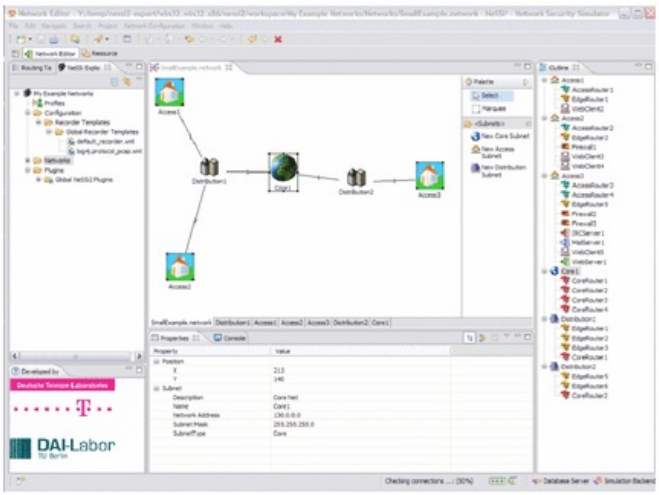


Slika 4.8. NetSim mrežni simulator

NetSIM obezbeđuje jednostavan metod za uključivanje interaktivne, diskretno promenljivih promena simulacije WWW sajtova. Koristi grafički interfejs za kreiranje modela, promenu i korišćenje, dozvoljava korisničku kontrolu izvršavanja modela i obezbeđuje animirani i brojevni izlaz. Simulator koristi Java-u za obezbeđivanje: objektno orijentisane fleksibilnosti, nezavisnosti platforme kroz WWW i kompatibilnost sa drugim Java baziranim grafičkim i analitičkim alatima[90].

4.2.12. Nessi

Nessi (Network Security Simulator) je mrežni simulacioni alat koji sadrži različite osobine vezane za mrežnu sigurnost i bitno se razlikujuje od ostalih opštih mrežnih simulatora (Slika 4.9.). Mogućnosti kao što su profilno bazirano generisanje automatskih napada, analiza saobraćaja i podrška za prepoznavanje algoritmnih plugin-ova omogućava mu da bude korišćen za istraživanje sigurnosti i za procese ocenjivanja[38].



Slika 4.9. NESSI mrežni simulator

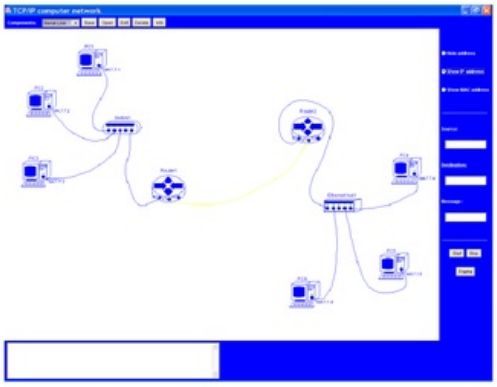
Nessi je uglavnom orijentisan za korišćenje u učenju, gde omogućava studentima da primene ili promene simulacione modele protokola sa minimalnim opštim znanjima. Druga primena Nessi je verifikacija i procena performansi novih protokola, gde se dozvoljava razvojnom licu da lako istražuje različite opcije.

4.2.13. WnetSim

WnetSim je simulator TCP/IP računarske mreže, koji je razvijen na Visokoj poslovnoj školi u Blacu (Slika 4.10.). Simulator je u upotrebi kao nastavno sredstvo za predmet Računarske mreže na odseku Računarstvo i informatika. Sistem je u potpunosti

razvijen u Java programskom jeziku. Izvršavanje se može izvesti u okviru bilo kog web browsera koji podržava Javu. Na računarima gde se simulator izvršava, neophodna je prethodna instalacija Java programskog paketa [39][45][47].

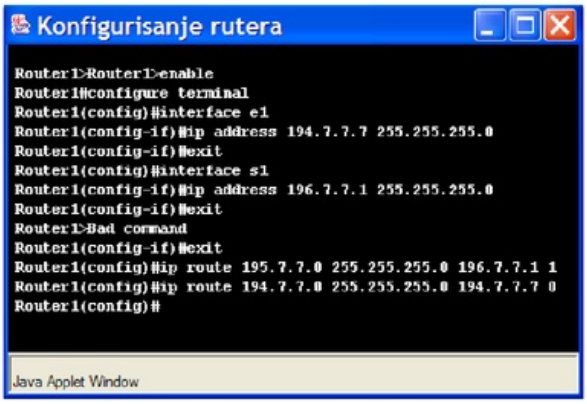
Tokom rada sa simulatorom računarskih mreža pojavljuju su standardni Windows prozori sa svojim podrazumevanim izgledom i funkcijama koje su uobičajene kod njih, kao što su smanjenje, povećavanje ili menjanje dimenzija i spuštanje prozora na statusnu liniju kao i zatvaranje prozora. Ovim operacijama korisnik može ekrane prilagoditi svojim potrebama, tako da od korisnika zavisi koliko prozora će u kom trenutku biti prikazano, ili koji prozor će biti aktivan, a koji će se nalaziti na statusnoj liniji. Sistem omogućava editovanje računarske mreže proizvoljne topologije i praćenje procesa u mreži u "realnom vremenu".



Slika 4.10. WnetSim mrežni simulator

Računarsku mrežu proizvoljne topologije moguće je editovati pomoću palete komponenti. Komponente, sa kojima je moguće raditi u sistemu su: PC, hub, switch, ruter, UTP kabl i serijska veza.

Primer konfigurisanja rutera u WnetSim mrežnom simulatoru je prikazan na slici 4.11.



Slika 4.11. Konfigurisanje rutera u WnetSim mrežnom simulatoru

Osnovne karakteristike predstavljenih simulatora date su u tabeli 4.2 i uključuju autora, operativne sisteme koje podržavaju, programske jezike koji se koriste za njihov razvoj i dostupnost.

Sistem	Autor	OS	Jezik	Pristup
OPNET Modeler	OPNET Technologies suite	Windows, Unix	C/C++	not
SSFNet	SSF Research Network	Windows, Unix	Java	free
NS-2	Berkeley University	Linux, BSD, Solaris, Mac OS, Windows	C++	free
NS-3	Berkeley University	Linux, BSD, Solaris, Mac OS, Windows	C++, Python	free
GNS3	EPITECH Innovative Project	Windows, Linux, Mac OS	Python	free
GTNets	Georgia Institute of Technology	Linux, OSX, Solaris, Windows	C++	free
GloMoSim	UCLA University	Windows, Linux, FreeBSD, Solaris...	C, Parsec	free
KivaNS	University of Alicante	Windows, Linux	Java	free
MIMIC	Gambit Communications	Windows, Solaris, Linux, Amazon Cloud	Tcl, C++, Java, Perl	not
CNET	University of Western Australia	Linux, Mac OS	C	free
NetSim	Tetcos	Windows	C/C++	not
NESSI	University of Applied Sciences of Western Switzerland	Windows, MAC OS X, Linux, Unix, BSD	Python	free
WNetSim	Business school of Professional Study Blace – Serbia	Windows, Linux, Mac OS	Java	free

Tabela 4.2. Osnovne karakteristike predstavljenih simulatora

5. Uporedna analiza simulatora računarskih mreža

Predmet istraživanja u ovom poglavlju je komparativna analiza postojećih simulacionih sistema za učenje Računarskih mreža i razvoj novog interaktivnog web-zasnovanog sistema za primenu u nastavi iz Računarskih mreža. Komparativna analiza performansi simulatora računarskih mreža nije u velikoj meri zastupljena u stručnoj literaturi, a raspoloživi radovi uglavnom uključuju ograničen skup eksperimenata sa jednostavnim simulacionim scenarijima. U ovom radu se simulatori proučavaju sa aspekta primene u nastavi, pa je i predmet ovog rada definisanje i odgovarajućih kriterijuma za upoređivanje karakteristika simulatora.

5.1.Kriterijumi za analizu simulatora računarskih mreža

Za procenu sumulatora uspostavljen je skup relavantinih kriterijuma koji mogu da se svrstaju u grupu koja je nazvana kriterijumi pokrivenosti (coverage criteria). Kriterijumi pokrivenosti procenjuju koje teme kurseva računarskih mreža podržavaju pojedini simulatori.

Razmatrani su i drugi kriterijumi kao što su: lakoća upotrebe, uticaj na učinak učenja, uticaj na razumevanje pređenog gradiva, transparentnost distrbuiranih simulatora, prikrivanje informacija. Ali, uključivanje ovih kriterijuma zahteva stvarnu upotrebu simulatora, što je nepraktično, pa su kriterijumi ograničeni samo na one koji se mogu analizirati na osnovu dostupnih informacija iz literature.

Takođe, autor se zahvaljuje na pomoći, savetima i informacijama koje je dobio od kolega: Chris McDonald za cnet, Pankaj Shah za MIMIC, Francisco Andrés Candelas Herías za KivaNS, Juergen Ehrensberger za Nessi i Brian Scheibe za NetSim.

Kriterijumi pokrivenosti su ustanovljeni korišćenjem IEEE Computer Engineering (2004) smernicama nastavnog plana i programa za računarske nauke [40] [22]. Ovaj dokument predstavlja finalni izveštaj Zajedničke radne grupe o nastavnom planu i programu računarstva - poduhvat SIGITE (Special Interest Group on Information Technology Education) u sklopu ACM (Association for Computing Machinery), ACM i IEEE Kompjuterskog udruženja - za programe u računarskom inženjeringu i definisanja široke

oblasti znanja koje se mogu primeniti na sve studijske programe iz oblasti računarske nauke. Računarske mreže su jedna od ovih oblasti znanja koje podrazumevaju sledeće nastavne teme:

- Pregled
- Komunikacija i mrežne arhitekture
- Komunikacije i mrežni protokoli
- LAN i WAN mreže
- Klijent-server
- Bezbednost i integritet podataka
- Bežično i mobilno računarstvo

Ove nastavne jedinice su izabrane kao osnovni kriterijumi i dati su sa detaljnim listama svojih tema u tabeli 5.1. Prva nastavna jedinica se ne uzima u obzir.

Nastavne jedinice	Teme
Komunikacija i mrežne arhitekture	Konfigurisanje mreže (point-to-point, multipoint)
	Mrežni uređaji: ripiteri, mostovi, svičevi, ruteri
	Mrežne topologije (zvezda, prsten, magistrala, rešetka)
	Servis sa i bez uspostavljanja veza
Komunikacije i mrežni protokoli	Mrežni protokoli
	Stek protokola (TCP/IP)
	Koncepti fizičkog sloja; koncepti sloja veze; rutiranje
	Mrežni standardi
LAN i WAN mreže	LAN topologije
	LAN tehnologije (Ethernet, token Ring, Gigabit Ethernet)
	Detekcija i korekcija grešaka
	Carrier sense multiple access (CSMA)
	Velike mreže
	Mreže sa komutacijom kola i komutacijom paketa
	Protokoli (adresiranje, kontrola zagušenja, virtualne veze, quality of service)
Klijent-server	Web tehnologije
	Karakteristike web servera
	Podrška za kreiranje web sajtova
Bezbednost i integritet podataka	Osnovi bezbednosti u mreži; kriptografija
	Enkripcija i privatnost
	Protokoli autentifikacije
	Filtriranje paketa
	Firewalls
	Virtuelne privatne mreže
	Bezbednost na transportnom sloju
Bežično i mobilno računarstvo	Pregled, evaluacija, kompatabilnost [48] nih standarda
	Specijalni problemi bečičnih mreća i mobilnog računarstva
	Bežične lokalne mreže i mreže zasnovane na satelitima
	Mobilni Internet protokol
	Mobilne adaptacije [40]
	Prošireni klijent-server model za prilagođavanje mobilnosti
	Mobilni pristup podacima
	Softverski paketi za podršku bežičnom i mobilnom računarstvu
	Uloga srednjeg sloja i alati za podršku
	Problemi sa performansama
	Nove tehnologije

Tabela 5.1. Osnovne teme koje se koriste kao kriterijumi pokrivenosti

5.2. Rezultati analize

Ovo poglavlje procenjuje mrežne simulatore u odnosu na utvrđene kriterijume pokrivenosti.

Komparativna analiza performansi ovih simulatora nije objavljena u velikom broju radova, a već postojeće publikacije obično uključuju ograničen skup eksperimenata sa jednostavnim scenarijom simulacije[48].

Na primer, skalabilnost ns-2 i SSFNet simulatora, u smislu brzine performanse potrebne za simulacije i memorijskog resursa računara se upoređuje i rezultati analize su pokazali da je ns-2 najbrži, ali i najzahtevniji u smislu memorijskog resursa.

Velika prednost slobodno dostupnih simulatora za istraživanja u oblasti IP mreža je ta što se mogu dodati nove ili modifikovati već postojeće karakteristike (preporučivanje novih algoritama, modifikacija protokola, itd).

Na primer, pri upotrebi CNET simulatora u nastavnim jedinicama moguće je da se CNET koristi za programiranje i razvoj protokola koji su deo TCP/IP paketa, ali CNET, kao deo API-ja ih prosto ne može sve obezbediti. Studenti, na primer, kao zadatak mogu dobiti da ih kodiraju, ili da ih priključuju (od, recimo, BSD ili Linux-a) u CNET, i zatim da mere njihovu efikasnost.

Osnovna karakteristika KivaNS je da su protokoli programirani u biblioteci nezavisni od grafičkog korisničkog interfejsa, prateći ekvivalent objektno orijentisane strukture prema slojevima i PDU mrežne arhitekture. Ova biblioteka može biti proširena novim protokolima i karakteristikama.

SSFNet je alat mrežne simulacije koji koristi otvoren kod softvera sa različitim aplikacijama mrežne simulacije. On je dizajniran u cilju proširenja mreže, uključujući topologiju, protokole, saobraćaj, i sl, pa je u stanju da podrži simulaciju za velike mreže poput Interneta. Međutim, opštim korisnicima nije lako da obavljaju mrežnu simulaciju koristeći SSFNet jer SSFNet ne obezbeđuje bilo koje druge dodatne suplementarne alate za projektovanje mrežnih elemenata i topologije, kao i analizu rezultata simulacije. Mrežno modeliranje i analiza procesa mora biti ručno obavljena od strane samih korisnika.

GloMoSim je dizajniran tako da može biti proširiv i lako kontrolisan. Paket komunikacionog protokola za bežične mreže je podeljena na skup slojeva, svaki sa sopstvenim API-jem. Model protokola u jednom sloju intereaguje sa onima na nižim (ili višim) slojem samo preko ovih API-ja [42].

U ns-2 simulatoru, mrežni modeli su programirani u obliku front-end jezika a koji je API za skript oTcl jezika. Programeri simulatora izričito navode kojim C++ metodama ili atributima se može pristupiti iz oTCL skripti. Ovo omogućava programerima da korisnicima ponude pojednostavljen API za ns-2, smanjujući poteškoće u učenju koje su povezane sa razvojem simulacije. Nedostatak korišćenja punopravnog jezika za skriptovanje je što takav pristup povećava režijske troškove što, opet, smanjuje performanse simulatora.

NS-3 nudi dva odvojena API-ja: "front-end" i API " backend" API. "Front-end" API, sadrži pomoćne klase koje olakšavaju inicijalizaciju i omogućava korišćenje klasa koje predstavljaju mrežne uređaje. "Backend" API uključuje sve detalje simulatora. U ns-3, pomoćne klase se mogu aktivirati samo iz korisničkih programa; to jest, njihovo korišćenje iz pozadine nije dozvoljeno, i doprinosi kompilaciji grešaka. Međutim, korisnici nisu ograničeni na korišćenje pomoćnih klasa; oni mogu da koriste pozadinski API za inicijalizaciju, što može dovesti do dugih kodova simulacionog modela. Benefit od pružanja dva API-ja je taj da se modeli mreža mogu implementirati na jeziku na kom je simulator i programiran . Front- end API ns-3 simulatora ima podršku za Python programski jezik. Međutim, cilj ove podrške nije u sakrivanju detalja implementacije, već u obezbeđivanju platformi za korisnike koji nisu upoznati sa C++ programskim jezikom [41].

Rezultati upoređivanja simulatora prema pokrivenosti tema su predstavljeni u tabelama 5.2 i 5.3.

Nastavne jedinice	Teme	OPNET	SSFNet	NS-2	NS-3	GNS3	GTNets	GloMoSi	KivaNS	MIMIC	CNET	NetSim	NESSI	WNetSim
Komunikacija i mrežne arhitekture	Konfigurisanje mreže (point-to-point, multipoint)	+	+	+	+	+	+	+	+	-	+	+	-	+
	Mrežni uređaji: ripiteri, mostovi, svičevi, ruteri	+	+	+	+	+	+	+	+	-	+	+	-	+
	Mrežne topologije (zvezda, prsten, magistrala, rešetka)	+	+	+	+	+	+	-	+	-	+	+	+	+
	Servis sa i bez uspostavljanja veza	+	+	+	+	+	+	-	-	-	+	+	-	+
Komunikacije i mrežni protokoli	Mrežni protokoli	+	+	+	+	+	+	+	+	+	+	+	+	+
	Stek protokola (TCP/IP)	+	+	+	+	+	+	+	+	+	-	+	+	+
	Koncepti fizičkog sloja; koncepti sloja veze; rutiranje	+	+	+	+	-	+	-	+	+	+	+	+	+
	Mrežni standardi	-	-	-	-	+	-	-	+	+	-	+	-	+
LAN i WAN mreže	LAN topologije	+	+	+	+	+	+	+	+	-	+	+	+	+
	LAN tehnologije (Ethernet, token Ring, Gigabit Ethernet)	+	+	+	+	+	+	+	+	-	+	+	+	+
	Detekcija i korekcija grešaka	-	+	-	-	-	+	-	+	-	+	+	+	+
	Carrier sense multiple access (CSMA)	+	+	+	+	+	+	+	-	-	+	+	+	-
	Velike mreže	+	+	+	+	+	+	-	-	-	-	+	-	+
	Mreže sa komutacijom kola i komutacijom paketa	+	+	+	+	-	+	-	-	-	-	+	-	+
	Protokoli (adresiranje, kontrola zagušenja, virtualne veze, quality of service)	+	+	+	+	+	+	+	+	-	+	+	+	+
Klijent-server	Web tehnologije	-	-	-	-	-	-	-	-	-	-	-	-	-
	Karakteristike web servera	+	+	+	+	-	+	-	-	-	-	-	-	-
	Podrška za kreiranje web sajtova	-	-	-	-	-	-	-	-	-	-	-	-	-
Bezbednost i integritet podataka	Osnovi bezbednosti u mreži; kriptografija	-	-	-	-	-	-	-	-	-	+	+	-	-
	Enkripcija i privatnost	-	-	-	-	-	-	-	-	-	+	+	-	-
	Protokoli autentifikacije	+	+	-	+	+	-		-	-	+	+	-	-
	Filtriranje paketa	+	-	-	-	+	-	--	-	-	+	+	-	-
	Firewalls	-	-	-	-	-	-		-	-	+	-	-	-
	Virtuelne privatne mreže	+	-	+	+	+	+	-	-	-	+	+	-	-
	Bezbednost na transportnom sloju	+	+	+	+	+	-	+	-	-	+	+	-	-
Bežično i mobilno računarstvo	Pregled, evaluacija, kompatabilnost b[48]ih standarda	-	-	-	-	-	-	-	-	-	-	-	-	-
	Specijalni problemi bečičnih mreća i mobilnog računarstva	-	-	-	-	-	-	+	-	-	+	-	-	-
	Bežične lokalne mreže i mreže zasnovane na satelitima	+	+	+	+	+	-	+	-	-	+	-	-	-
	Mobilni Internet protokol	+	+	+	+	-	-	+	-	-		-	-	-
	Mobilne adaptacije	-	-	-	-	-	-	-	-	-		-	-	-
	Prošireni klijent-server model za prilagođavanje mobilnosti	-	-	-	-	-	-	+	-	-	+	-	-	-
	Mobilni pristup podacima	-	-	-	-	-	-	+	-	-		-	-	-
	Softverski paketi za podršku bežičnom i mobilnom računarstvu	-	-	-	-	-	-	+	-	-	-	-	-	-
	Uloga srednjeg sloja i alati za podršku	-	-	-	-	-	-	-	-	-	-	-	-	-
	Problemi sa performansama	-	-	-	-	-	-	-	-	-	-	-	-	-
	Nove tehnologije	-	-	-	-	-	-	-	-	-	-	-	-	-

Tabela 5.2. Vrednovanje simulatora prema usvojenim kriterijumima

Procena simulatora prema kriterijumima koji su zasnovani prema smernicama Computer Engineering (2004) nastavnog plana i programa za studente računarskih tehnika je predstavljena u tabeli 5.2. Oznaka "+" znači da simulator ima određenu funkcionalnost neophodnu za pokrivanje nastavnog materijala u određenoj oblasti računarske mreže, dok oznaka "-" znači da simulator nema određenu funkcionalnost neophodnu za pokrivanje nastavnog materijala u određenoj oblasti.

Na osnovu rezultata vrednovanja mogućnosti upotrebe simulatora za pokrivanje navedenih tema, u tabeli 5.3. je dat udeo pojedinih oblasti procentualno u odnosu na ukupan broj časova.

Nastavne jedinice	**h**	**%**
Komunikacija i mrežne arhitekture	3	15
Komunikacije i mrežni protokoli	4	20
LAN i WAN mreže	4	20
Klijent-server	3	15
Bezbednost i integritet podataka	4	20
Bežično i mobilno računarstvo	2	10

Tabela 5.3. Udeo pojedinih oblasti u ukupnom gradivu

U tabeli 5.4. sumirani su rezultati upoređivanja simulatora prema pokrivenosti oblasti, u odnosu na udeo pojedinih oblasti u ukupnom broju časova.

Knowledge units	OPNET	SSFNet	NS-2	NS-3	GNS3	GTNets	GloMoSim	KivaNS	MIMIC	CNET	NetSim	NESSI	WNetSim
Komunikacija i mrežne arhitekture	15.00	15.00	15.00	15.00	15.00	15.00	7.50	11.25	0.00	15.00	15.00	3.75	15.00
Komunikacije i mrežni protokoli	15.00	15.00	15.00	15.00	15.00	15.00	10.00	20.00	20.00	10.00	20.00	15.00	20.00
LAN i WAN mreže	17.14	20.00	17.14	17.14	14.29	20.00	11.43	11.43	0.00	14.29	20.00	14.29	17.14
Klijent-server	5.00	5.00	5.00	5.00	0.00	5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Bezbednost i integritet podataka	11.43	5.71	5.71	8.57	11.43	2.86	2.86	0.00	0.00	20.00	17.14	0.00	0.00
Bežično i mobilno računarstvo	1.82	1.82	1.82	1.82	0.91	0.00	5.45	0.00	0.00	2.73	0.00	0.00	0.00
Ukupna pokrivenost	**65.39**	**62.53**	**59.68**	**62.53**	**56.62**	**57.86**	**37.24**	**42.68**	**20.00**	**62.01**	**72.14**	**33.04**	**52.14**

Tabela 5.4. Rezultati upoređivanja simulatora prema pokrivenosti oblasti

Teme Komunikacija i mrežne arhitekture su pokrivene u velikoj meri u gotovo svim simulatorima. Izuzetak su simulatori kao što su MIMIC i Nessi koje su razvijeni za specijalizovane potrebe.

Ukoliko planiramo da predajemo o mrežnom menadžmentu i o upotrebi aplikacija mrežnog menadžmenta u cilju prikazivanja navedenih tema, onda MIMIC može biti koristan za većinu njih. Ako je SNMP protokol uključen, onda MIMIC simulator može biti veoma koristan.

Nessi je alat mrežne simulacije koji uključuje različite karakteristike koje su relevantne za mrežnu bezbednost i koje ga izdvajaju od mrežnih simulatora opšte namene. Dok se drugi simulatori fokusiraju na minimiziranje vremena simulacije, nessi pokušava da minimizuje vreme razvoja i teškoća u implementaciji novih simulacionih modela. Prednost Nessi-ja je ta što se simulacioni modeli mogu razviti za mnogo kraće vreme nego što je to potrebno sa drugim simulatorima. Nessi stoga omogućava studentima da kreiraju ili modifikuju modele mrežnih protokola sa minimalnim opterećenjem i na taj način je savršeno pogodan za povezivanje mrežnih vežbi ili semestralnih projekata.

Teme iz oblasti LAN i WAN mreža su pokrivene kod gotovo svih simulatora. Izuzetak je MIMIC koji nije razvijen da se bavi temama u toj oblasti.

Teme iz oblasti Komunikacije i mrežni protokoli su najbolje pokrivene simulatorima KivaNS, MIMIC, NetSim i WNetSim.

Uglavnom, KivaNS sa dosta detalja realizuje sledeće protokole: Ethernet za IP, PPP, IP i ARP. Takođe, sprovodi IP rutiranje i fragmentaciju. IP usmeravanje se može konfigurisati na sličan način kao u realnom OS. Osim toga, on takođe podrazumeva premošćavanje i priključivanje za Ethernet frejmove koji se tiču topologije, i Spanning Tree algoritma (pojednostavljeni 802.1d). Fizički mediji mogu simulirati greške u frejmovima, dok link i mrežni protokoli rade konsenkventno.

Teme iz oblast bezbednosti i integriteta podataka imaju veoma dobru pokrivenost sa cnet i NetSim simulatorima.

Cnet pruža mnoge od osnovnih karakteristika za niži nivo umrežavanja, kao što su prenos podataka i prijem, verovatnoća uvođenja grešaka, precizan tajming i statističke funkcije, raznovrsnost NIC tipova, i programirajuću mobilnost čvora, ali ne omogućava sve karakteristike u svojoj standardnoj distribuciji.

Teme u oblasti bežičnog i mobilnog računarstva i klijent-server računarstva su slabo podržane od strane gotovo svih simulatora. Izuzetak je GloMoSim simulator koji je razvijen za specijalizovane potrebe. Razlog za to je što se većina analiziranih simulatora uglavnom koncentriše na teme vezane za osnovne komponente računarskih mreža.

Postoje simulatori sa generalno nižom pokrivenosti tema, ali sa visokim nivoom pokrivenosti koje se tiču određenih, konkretnih jedinica. MIMIC ima visoku pokrivenost za oblast Komunikacije i mrežni protokoli, a cnet i NetSim za jedinice bezbednosti integriteta podataka i. Ova vrsta informacija je mnogo važnija od ukupne pokrivenosti ograničenih na određene nastavne jedinice.

Rezultat procene pokazuje da ne postoji jedinstven simulator koji pokriva sve teme. Rezultat je posledica činjenice da je oblast računarskih mreža raznovrsna i da uključuje veliki broj tema, pa bi razvoj simulatora koji pokriva sve teme kao rezultat dao sistem koji bi bio veoma glomazan i nepraktičan za korišćenje.

Najbolji ukupna pokrivenost tema se postiže sa simulatorima NetSim (72,14%). Većina simulatora postiže više od 50,00% pokrivenosti. Izuzetak je samo MIMIC simulator kod koga je procenat pokrivenosti 20% i koji je specijalizovan za rad sa SNMP protokolom. Ostali simulatori postižu oko 30.00% pokrivenosti.

Na kraju vredi napomenuti da procenat pokrivenosti ne daje informacije o prednosti korišćenja pojedinih simulatora, već samo o broju obuhvaćenih tema, odnosno o temama iz oblasti računarskih mreža koje se lako mogu pokriti pomoću datog simulatora.

Zaključak je da je korišćenje simulatora veoma korisno u primeni za učenje računarskih mreža.

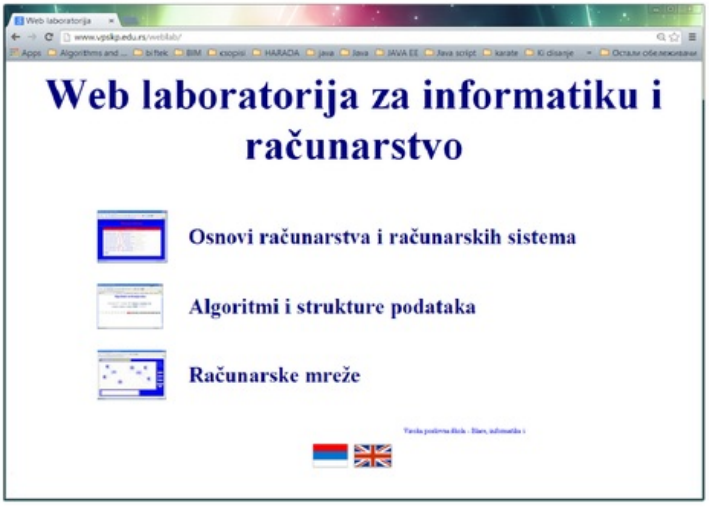
6. Interaktivni sistem za učenje osnova IP umrežavanja

U ovom poglavlju opisana je web aplikacija, zasnovana na Java programskom jeziku, koja može da se koristi kao komponenta u procesu daljinskog obrazovanja za izvođenje laboratorijskih vežbi u oblasti računarskih mreže. Ova aplikacija je deo web laboratorije za računarstvo i informatiku (weblab.vpskp.edu.rs) [45]. Sastavni delovi web laboratorije[49] odseka za računarstvo i informatiku Visoke poslovne škole u Blacu su (slika 6.1.) [47]:

- Osnovi računarstva i računarskih sistema
    - Brojni sistemi; Sistem služi za rad sa brojnim sistemima. Omogućava konvertovanje vrednosti u binarnom, oktalnom, dekadnom i heksadekadnom brojnom sistemu
    - Digitalna logička kola[46]; sitem omogućava projektovanje i simulaciju procesa u digitalnim ligičkim kolima. Podržan je kompozitni model, što znači da je moguće razviti nezavisnu kompoziciju, sa proizvoljnim brojem ulaza i izlaza, koja se sastoji od elementarnih digitalnih logičkih elemenata i koristiti je kao jednu celinu[50].
    - Simulator računarskog sistema; Softverski sistem simulira proces unošenja asemblerskog programa, kompajliranja asemblerskog programa, učitavanja mašinskog programa u memoriju i izvršavanja instrukcija mašinskog programa. Instrukcije mogu da se izvršavaju jedna po jedna ili kontinualno.
- Algoritmi i strukture podataka[52]
    - Stek; Rad sa linearnom strukturom Stek. Sistem omogućava kreiranje, punjenje, pražnjenje steka i rad sa bezadresnim aritmetičkim operacijama,
    - Stablo; Editor za rad sa nelinearnom strukturom Stablo. Sistem omogućava dodavanje novih čvorova, obilazak stabla, traženje putanje između dva proizvoljna čvora...,

- Algoritmi sortiranja; Sistem omogućava predstavljanje nekih od najpopularnijih algoritama sortiranja kao što su: Bubble sort, Selection sort, Insertion sort, Quicksort and Binary Sort.

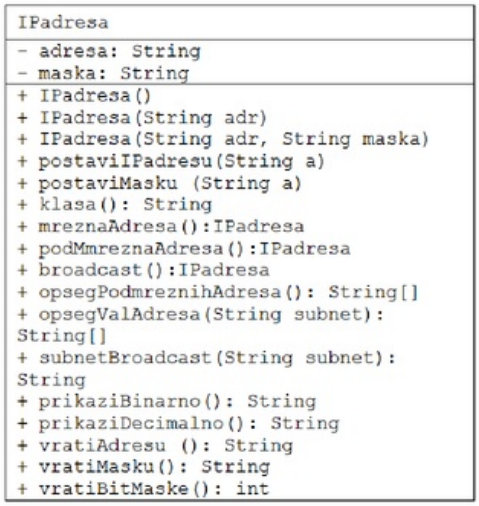
- Računarske mreže
  - Osnove IP umrežavanja; Sistem za rad sa IP adresama, omogućava podelu IP mreže na podmreže, određivanje maske podmreže...
  - Simulator računarske mreže[51]; Okruženje omogućava vizuelizaciju i simulaciju procesa u računarskoj mreži proizvoljne topologije. Osnovne komponente sistema su računari, hub-ovi, switch-evi, ruteri, UTP kablovi i serijske veze. Podržano je fizičko i logičko IP adresiranje računara i interfejsa rutera. Ruteri omogućavaju statičko rutiranje unošenjem statičkih ruta u konfuguracionoj proceduri.



Slika 6.1. Web laboratorija za informatiku i računarstvo

6.1. Modelovanje IP adrese

IP adresa se modeluje pomoću Java klase IPadresa[51]. UML dijagram klase je prikazan na slici 6.2. Klasa sadrži privatna polja adresa i maska i javne metode koje omogućavaju rad sa IP adresama, mrežnim i podmrežnim adresama.



Slika 6.2. Java klasa IPadresa

U nastavku će biti opisane neke od najvažnijih metoda klase IPadresa.

Listing 6.1. prikazuje konstruktor klase IPadresa. Konstruktor ima dva argumenta tipa String i on kreira objekat klase IPadresa i uspostavlja početno stanje objekta definišući vrednost IP adrese i maske podmreže.

Listing 6.1. Konstruktor klase IPadresa

```
 // Konstruktor
IPadresa(String adresa, String maska){
      this.adresa = adresa;
      this.maska = maska;
}
```

Metoda podmreznaAdresa je prikazana u listingu 6.2. Ova metoda, na osnovu IP adrese i maske podmreže, izračunava podmrežnu adresu i vraća rezultat u formi String podatka.

U metodi se izdvajaju okteti adrese i maske, pretvaraju se u binarne vrednosti i na tim vrednostima se primenjuje logička operacija „&“. Rezultat se ponovo konvertuje u

decimalnu vrednost i rezultantni okteti se ponovo spajaju u rezultujuću adresu, koja se vraća kao rezultat.

Listing 6.2. Metoda podmreznaAdresa

```
// Izracunavanje Podmrezne adrese
public String podMreznaAdresa( ) {
      int Iok = vratiOktet(adresa, 1) & vratiOktet(maska, 1);
      int IIok = vratiOktet(adresa, 2) & vratiOktet(maska, 2);
      int IIIok = vratiOktet(adresa, 3) & vratiOktet(maska, 3);
      int IVok = vratiOktet(adresa, 4) & vratiOktet(maska, 4);
      return (Iok +"."+ IIok +"."+ IIIok +"."+IVok);
}
```

U listingu 6.3 je prikazan deo metode opsegPodmreznihAdresa. U metodi se pravi ulančana lista, proračunavaju se sve podmrežne adrese, date mrežne adrese, i onda se smeštaju u listu. Rezultat se vraća u obliku niza stringova.

Listing 6.3. Metoda opsegPodmreznihAdresa

```
public String[] opsegPodmreznihAdresa(){
   ArrayList svePodmrezneAdrese = new ArrayList();
   if (klasa().equals("C"){
       String triOkteta = vratiOktet(adresa, 1) +
       "."+vratiOktet(adresa, 2)+"."+vratiOktet(adresa, 3);
       int inkrement = dvaNa(32-n);
       int cetvrtiOktet = 0;
       do{
            svePodmrezneAdrese.add(triOkteta+"."+cetvrtiOktet);
           cetvrtiOktet = cetvrtiOktet + inkrement;
      }while (cetvrtiOktet <= 255);
   }
   ...
```

U metodi broadcast(), koja je prikazana u listingu 6.4, prvo se adresa pretvara u binarnu formu, zatim se izračunava adresa usmerenog razglašavanja (broadcast) za tu podmrežu i na kraju se dobijena dresa pretvara u decimalan oblik i vraća kao podatak tipa String.

Listing 6.4. Metoda broadcast

```
public String broadcast(){
      ...
     String binarno = binarnaAdresa();
     String mrezniDeo = binarno.substring(0,brBita);
     String hostDeo = new String();
     for (int i = brBita; i< 32; i++ ){
         hostDeo = hostDeo + "1";
     }
     broadcast = mrezniDeo+hostDeo;
     //// Pretvoriti u decimalno
     return prikaziDecimalno(broadcast);
}
```

Metoda klasa, koja je prikazana u listingu 6.5, proračunava kojoj klasi pripada data adresa i vraća rezultat u vidu stringa.

Listing 6.5. Metoda klasa

```
public String klasa(){
       String klasa = "";
       if (vratiOktet(adresa,1) >= 1 &
            vratiOktet(adresa,1) <= 126)
              klasa = "A";
       else if (vratiOktet(adresa,1) >= 128 &
             vratiOktet(adresa,1) <= 191)
              klasa = "B";
       else if (vratiOktet(adresa,1) >= 192 &
            vratiOktet(adresa,1) <= 223)
              klasa = "C";
       return klasa;
}
```

U listingu 6.6 prikazana je metoda prvaAdresa(). Ova metoda prvo određuje kojoj podmreži pripada data IP adresa, a zatim proračunava koja je prva validna IP adresa u datom segmentu.

Listing 6.6. Metoda prvaAdresa

```
public String prvaAdresa() {
     String mrez = podMreznaAdresa();
    int cetvrtiOktet = vratiOktet(mrez,4) +1;
     return vratiOktet(mrez,1)+"."+vratiOktet(mrez,2)+"."+
            vratiOktet(mrez,3)+"."+cetvrtiOktet;
}
```

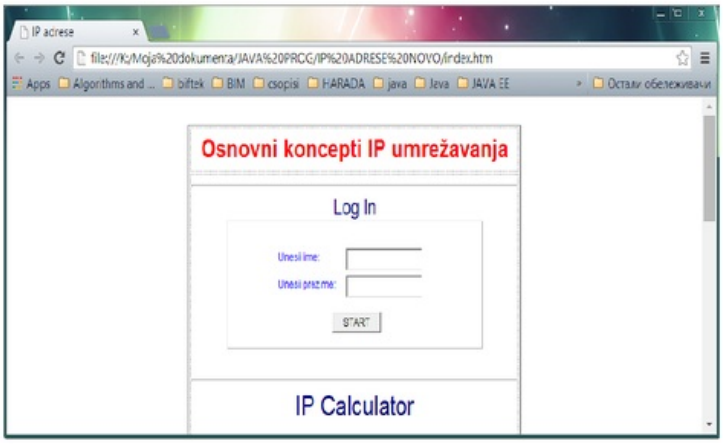
U listingu 6.7 prikazana je metoda zadnjaAdresa(). Ova metoda prvo određuje broadcast adresu za datu podmrežu, a zatim proračunava koja je zadnja validna IP adresa u datom segmentu.

Listing 6.7. Metoda zadnjaAdresa

```
public String zadnjaAdresa() {
      String broad = broadcast();
      int cetvrtiOktet = vratiOktet(broad,4) -1;
     return vratiOktet(broad,1)+"."+vratiOktet(broad,2)+"."+
            vratiOktet(broad,3)+"."+cetvrtiOktet;
 }
```

6.2. Rad sa sistemom

Uvodni obrazac aplikacije za učenje osnova IP umrežavanja prikazan je na slici 6.3.



Slika 6.3. Web aplikacija za učenje osnova IP umrežavanja

On se satoji od forme, koja se koristi za unos osnovnih podataka o studentu i dugmeta koje služi za pokretanje aplikacije. Na raspolaganju je i link **„IP Calculator“** koji startuje aplikaciju kalkulatora IP adresa[54][55][56].

Ova aplikacija se koristi za izvođenje laboratorijskih vežbi vezanih za osnove IP umrežavanja. Svaka laboratorijska vežba se sastoji od četri faze:

- pripremna faza
- faza procene potrebnog predznanja
- faza izvođenja vežbe
- faza pisanja izveštaja

Da bi se pripremio za izvođenje date laboratorijske vežbe, student mora da pregleda odgovarajući materijal sa teorijskih predavanja i udžbenika, kao i da pročita odgovarajuće uputstvo za izvođenje vežbe. Svakoj laboratorijskoj vežbi prethodi odgovarajući kratak test, koji treba da potvrdi da student razume sadržaj vežbe.

Nakon uspešno urađenog testa student pristupa izvođenju prethodno zadate vežbe.

Na osnovu zapažanja dobijena tokom vežbe, student pravi pisani izveštaj i ubacuje u njega odgovore na relavantna pitanja.

6.2.1. Kalkulator IP adresa

IP kalkulator omogućava unos proizvoljne IP adrese u polju **IP address** i proračunavanje svih potrebnih podataka vezanih za datu adresu: klasa, mrežna adresa, maska, subnet, broadcast adresa... Podaci se prikazuju i u decimalnoj i u binarnoj formi[56].

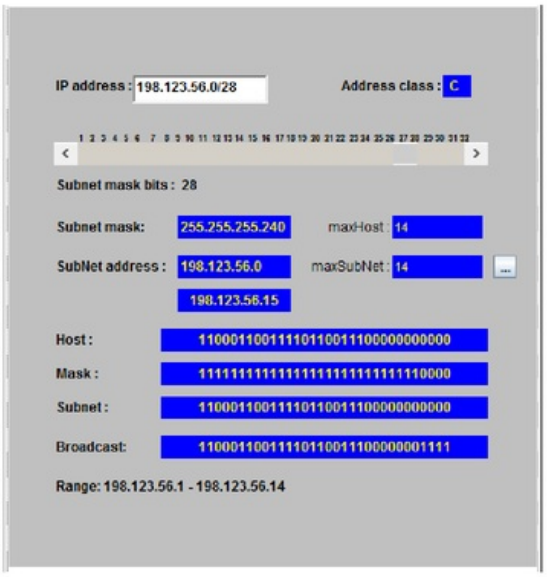
Kalkulator IP adresa može poslužiti za proračun svih prethodno nabrojanih elemenata IP adrese. Osnovni ekran IP kalkulatora je prikazan na slici 6.4.



Slika 6.4. IP Kalkulator

U slučaju da želimo da podelimo datu mrežu na određen broj podmreža, potrebno je da definišemo koliko bitova je potrebno pozajmiti iz **HosID** dela IP adrese. To se može uraditi na dva načina: da pomoću klizača definišemo broj bitova koji određuju mrežu i podmrežu, ili da IP adresu unesemo u obliku CIDR notacije (npr. 198.123.56.23/28).

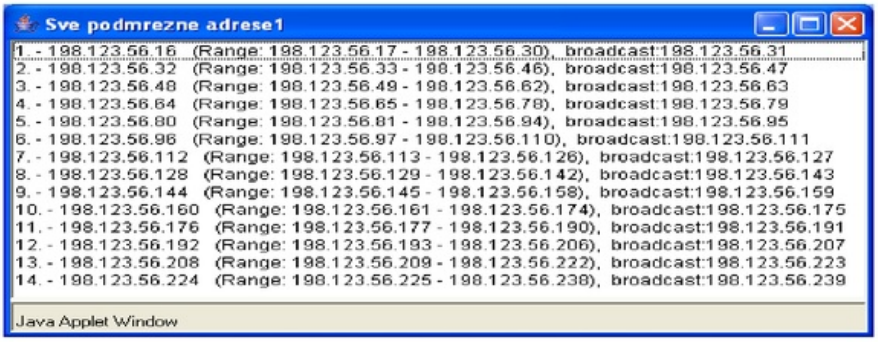
Neka je, na primer, data IP mrežna adresa 198.123.56.0 koja pripada klasi C. Ako hoćemo da datu IP adresu podelimo na najmanje 10 podmreža, onda je potrebno da iz HosID dela IP adrese izdvojimo 4 bita, pošto je $2^4 > 10$. U tom slučaju bit podmrežne adrese je 28 (slika 6.5.).



Slika 6.5. Primer IP mrežne adrese

Vidimo da je maska podmreže 255.255.255.240, da na ovaj način možemo da formiramo maksimalno 14 podmreža, sa po maksimalno 14 hostova u svakoj podmreži.

Klikom na dugme, pored polja **maxSubNet**, možemo dobiti spisak svih raspoloživih podmrežnih adresa, sa opsegom validnih IP adresa hostova, kao i brodcast adresu za svaku podmrežu (slika 6.6.).



Slika 6.6. Podmrežne adrese

6.3. Laboratorijske vežbe

U nastavku disertacije biće predstavljene laboratorijske vežbe koje su zasnovane na aplikaciji za učenje osnova IP umrežavanja. Sistem je tako koncipiran da može da se koristi za učenje na daljinu, kao i za praktičan rad u laboratoriji. Laboratorijski rad se sastoji od vežbi, koje su koncipirane tako da pokrivaju značajan deo tema iz predmeta Računarske mreže koji je deo nastavnog plana i programa na Visokoj poslovnoj školi strukovnih studija u Blacu.

Svrha i cilj svake vežbe predstavljena je studentima kroz uvodna predavanja.

Sam rad sa **IP** umrežavanjem je podeljen na sedam laboratorijskih vežbi:

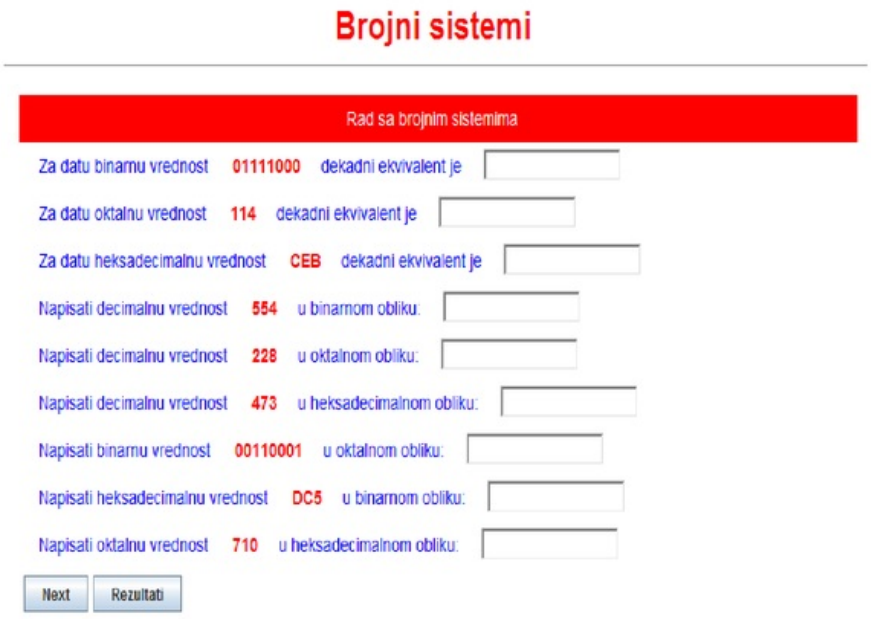
- Rad sa brojnim sistemima
- Klase IP adresa
- Podela IP mreže na podmreže
- Određivanje podmrežne adrese
- CIDR
- Topologija 1
- Topologija 2

U nastavku sledi opis laboratorijskih vežbi.

6.3.1. Vežba 1 – Rad sa brojnim sistemima

Upotreba binarnog oktalnog i heksadecimalnog brojnog sistema je od velikog značaja za računarstvo informatiku, a razumevanje IP adresiranja nije moguže bez poznavanje ovih brojnih sistema pa je, na početku, posebna pažnja poklonjena ovim brojnim sistemima[91].

Prvi deo je web aplikacija za rad sa brojnim sistemima koju čini interaktivan sistem u kome se generišu pitanja na koje student daje odgovore (slika 6.7.). Pitanja se generišu na slučajan način, a student upisuje svoje odgovore u odgovarajuća tekstualna polja.



Slika 6.7. Brojni sistemi

Vežba se sastoji od devet pitanja. U svakom pitanju se, na slučajan način, generiše određena vrednost u nekom brojnom sistemu. Pitanja se odnose na konverziju slučajno generisanih vrednosti iz jednog u drugi brojni sistem. Zastupljeni su brojni sistemi sa onovom dva, osam, deset i šesnaest. Nakon datih odgovora student može da vidi statistiku tačnih/netačnih odgovora.

Primer upotrebe aplikacije je prikazan na slici 6.8.

Student odmah nakon unosa odgovora u odgovarajuće polje ima povratnu informaciju o tačnosti unešenog odgovora. Tačni odgovori su markirani plavom bojom, dok netačni odgovor je obojen crnom bojom.



Slika 6.8. Rad sa brojnim sistemima

6.3.2. Vežba 2 - Klase IP adresa

Druga laboratorijska vežba se bavi klasama IP adresa[92].

Prilikom startovanja aplikacije generiše se slučajna **IP** adresa, koja se nalazi u gornjem delu web browser-a (slika 6.9.).

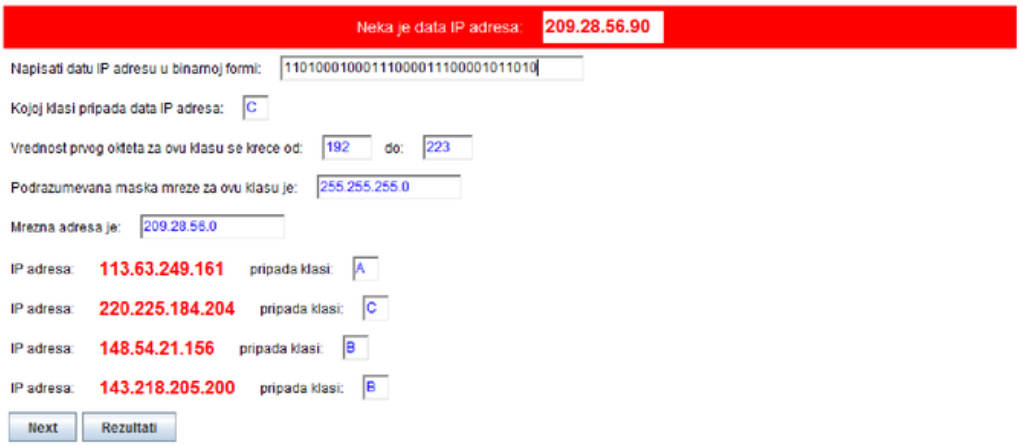


Slika 6.9. Klase IP adresa

U ovom delu potrebno je dati odgovore na pitanja koja se odnose na datu **IP** adresu, koja se generiše na slučajan način. Potrebno je napisati binarni oblik adrese, odrediti klasu adrese, opseg prvog okteta date klase **IP** adrese, podrazumevanu masku podmreže i mrežnu adresu.

U nastavku je generisano četiri IP adrese, na slučajan način, a studen bi trebalo da da odgovor kojim klasama pripadaju generisane adrese.

Primer upotrebe aplikacije je prikazan na slici 6.10.



Slika 6.10. Primer upotrebe aplikacije

6.3.3. Vežba 3 - Podela IP mreže na podmreže

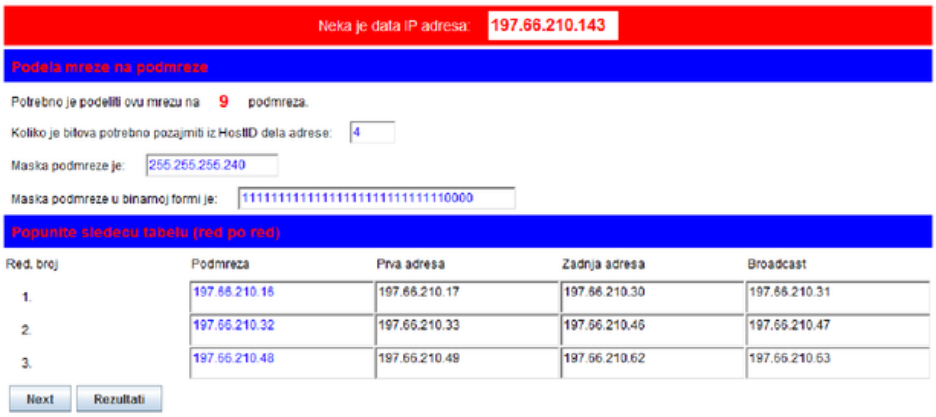
Treća laboratorijska vežba služi za proveru stečenog znanja u vezi sa podelom IP mreže na podmreže (slika 6.11.).



Slika 6.11. Podela IP mreže na podmreže

Sada je potrebno odrediti koliko bitova je potrebno pozajmiti da bi se data mrežna adresa (za IP adresu koja je generisana slučajno) podelila na zadati broj podmreža. Zatim je potrebno odrediti masku podmreže i prikazati masku podmreže u binarnoj formi. Nakon toga potrebno je popuniti tabelu u koju je potrebno uneti nekoliko važečih podmrežnih adresa, odrediti opseg korisnih adresa za svaku podmrežu i broadcast adresu.

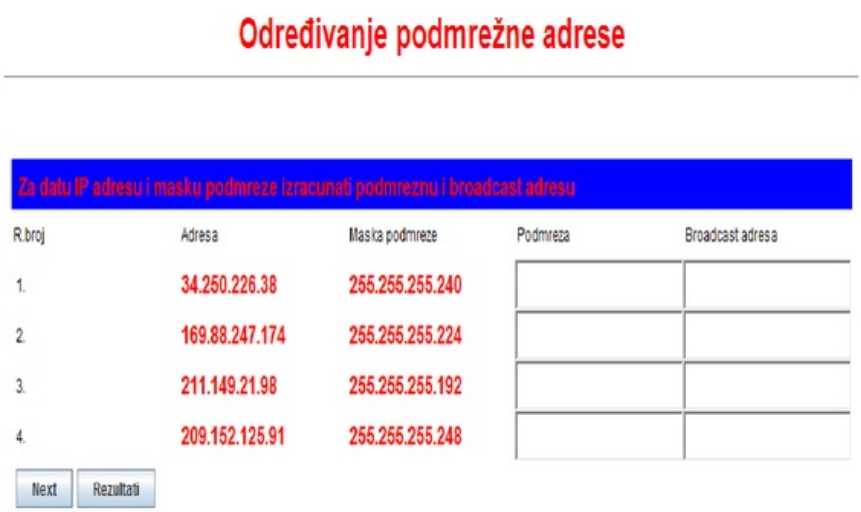
Primer popunjenog obrasca je prikazan na slici 6.12.



Slika 6.12. Primer popunjenog obrasca

6.3.4. Vežba 4 - Određivanje podmrežne adrese

U četvrtoj laboratorijskoj vežbi data je tabela koju treba dopuniti (slika 6.13.).



Slika 6.13. Određivanje podmrežne adrese

U svakom redu tabele slučajno se generišu IP adresa i maska podmreže, a korisnik treba da proračuna podmrežnu adresu i broadcast adresu i da ih unese u odgovarajuća polje.

U konkretnom primeru prikazanom na slici 6.14. prva generisana IP adresa je 38.88.26.112 i maska podmreže 255.240.0.0. Kojoj podmreži ova adresa pripada?

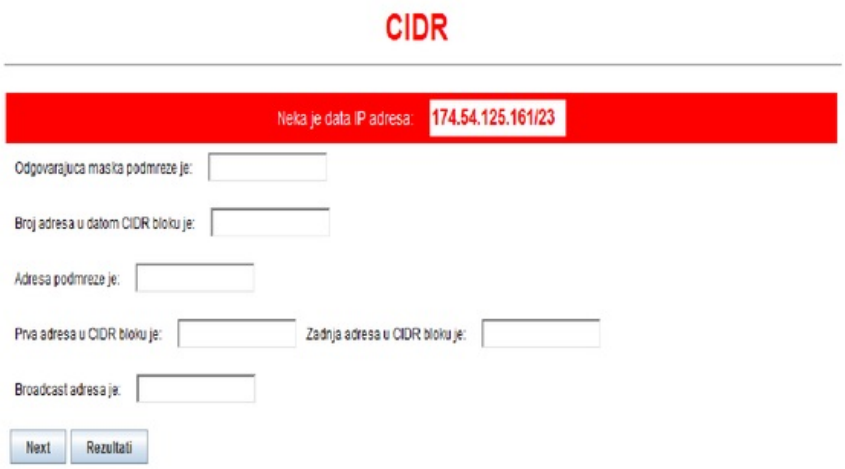
Na osnovu maske podmreže zaključujemo sledeće: postoji 14 podmreža, a inkrement za svaku je 16. Tako je adresa prve podmreže 38.0.0.0, druge 38.16.0.0, treće 38.32.0.0, itd. a adresa poslednje podmreže biće 38.240.0.0 Dakle, možemo zaključiti da naša, slučajno generisana IP adresa, pripada podmreži 38.80.0.0. Ova vrednost upisuje se u predviđeno polje na slici 6.13. Broadcast adresa je 38.95.255.255.



Slika 6.14. Primer generisanja IP adresa

6.3.5. Vežba 5 - CIDR

Peta laboratorijska vežba se bavi CDIR (Classless Inter-Domain Routing) notacijom za predstavljanje IP adresa (slika 6.15.).



Slika 6.15. Primer CIDR

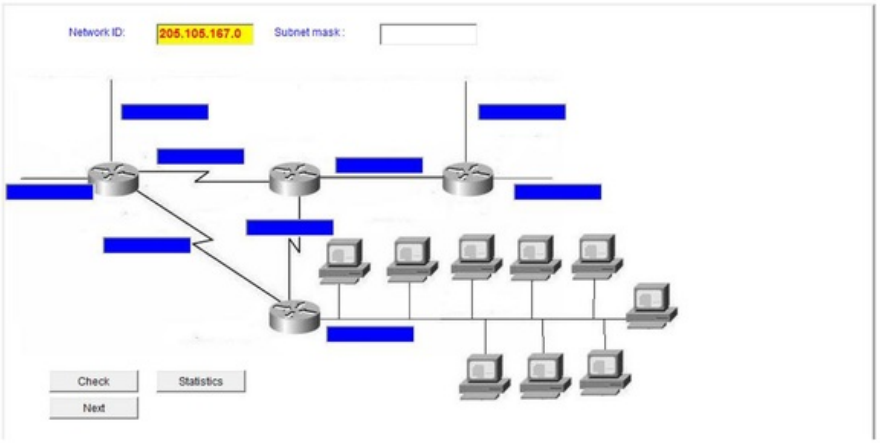
U ovoj vežbi se generiše, na slučajan način jedan CDIR blok, a zatim je potrebno u odgovarajuća polja upisati odgovore na pitanja vezana za dati CDIR blok (slika 6.16.).



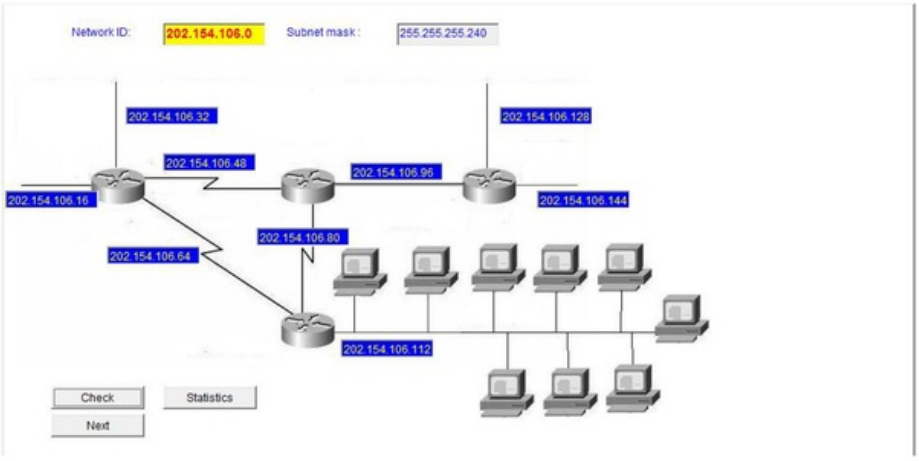
Slika 6.16. Odgovori za dati CDIR blok

6.3.6. Vežba 6 - Subneting - topologija 1

U šestom delu (slika 6.17.) data je mrežna topologija za koju student treba da odredi odgovarajuću masku podmreže i da izvrši adresiranje svakog segmenta mreže sa validnim IP podmrežnim adresama.



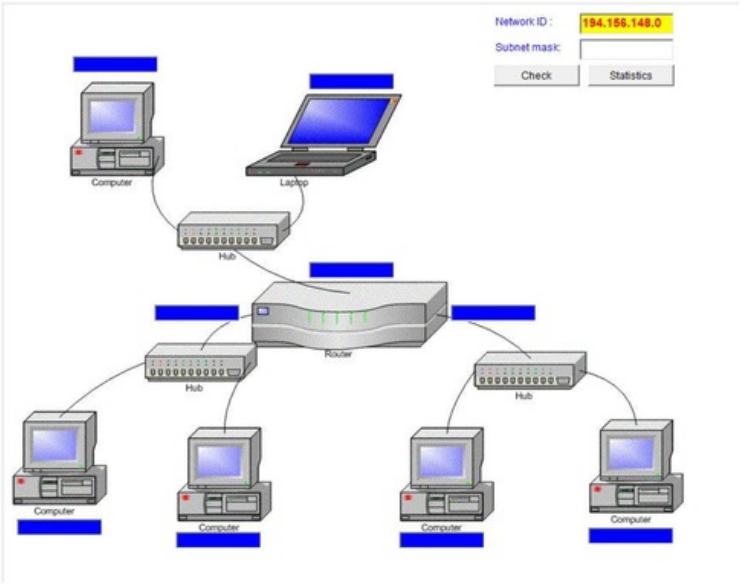
Slika 6.17. Mrežna topologija



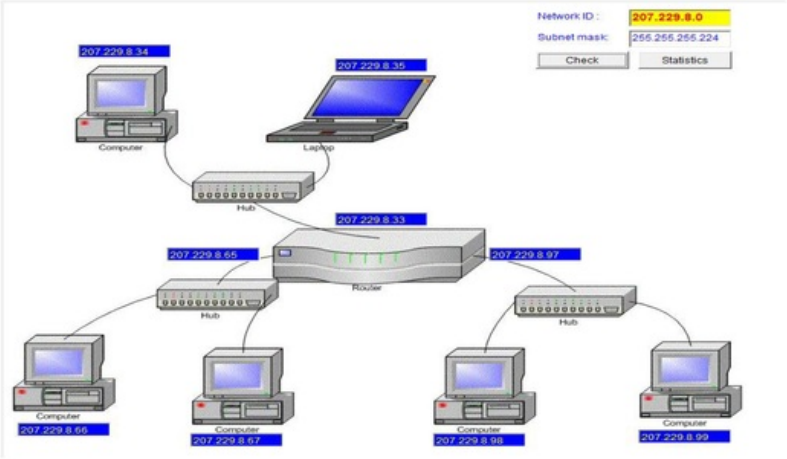
Slika 6.18. Mrežna topologija sa slučajno generisanom IP adresom

6.3.7. Vežba 7 - Subnetig - topologija 2

U sedmom delu (slika 6.19.) data je mrežna topologija i slučajno generisana mrežna IP adresa. Student treba da odredi odgovarajuću masku podmreže i da u odgovarajuća tekst polja unese IP adrese za svaki interfejs svakog mrežnog uređaja.



Slika 6.19. Unos IP adresa



Slika 6.20. Mrežna topologija sa unetim IP adresama

Na kraju, ili nakon svake celine, klikom na dugme Rezultati, korisnik može da ima uvid u postignute rezultate (slika 6.20.).



Slika 6.21. Statistika postignutih rezultat

6.3. Procena upotrebljivosti sistema

Ocenjivanje znanja najčešće se vrši na tradicionalan način, usmenim ispitivanjem ili putem testova na papiru. Međutim, danas je razvijen veliki broj testova za učenje i ocenjivanje koji se obavljaju uz pomoć računara. Predloženi sistem generiše računarske testove koji predstavljaju izuzetno efikasan način provere znanja. Smanjuje se vreme provere znanja kao i vreme izdavanja rezultata. Praktično, u momentu kada ispitanik završi test sistem generiše izveštaj (ocena ili procentualna prolaznost, u nekim sistemima i preporuke za učenje onih oblasti iz kojih su pitanja loše urađena i sl.)[47]

Stilovi učenja predstavljaju različite pristupe ili načine učenja. Svaki student pri usvajanju znanja prednost daje informacijama koje dobija preko određenog čulnog modaliteta, i tako, koristeći te informacije, najefikasnije uči. Osnovnu tipologiju stilova učenja prema tom modalitetu čine [57]: vizuelni, auditivni i taktilni/kinestetički stil učenja.

Vizuelni stil učenja je dominantan za one koji najlakše usvajaju neko gradivo kada su informacije prezentovane vizuelno u obliku teksta (grafičko-vizuelni) ili slika. Uglavnom preferiraju samostalno učenje. Oni koji najlakše uče slušajući predavanja, diskusije, razmenu ideja, koriste auditivni stil učenja. Zbog toga je za ovaj stil učenja karakteristično dobro snalaženje u radu u grupi ili paru. Oni koji tokom procesa učenja hvataju beleške, crtaju slike i dijagrame kako bi lakše zapamtili informacije imaju izražen taktilni/kinestetički stil učenja. Oni najbolje uče kroz pokret, igru, glumu ili konkretnu radnju, aktivno istražujući fizički svet oko sebe. Uzimajući u obzir ove stilove učenja napravljena je paralela između nekih poznatih sistema učenja i testiranja. Izabrani kriterijum predstavlja način izražavanja znanja.

Zavisno od stila učenja pojedinog studenta, razlikovće se i njegove mogućnosti izražavanja stečenog znanja. Pojedini studenti se najbolje izražavaju usmeno kroz diskusiju sa ispitivačem, drugi više vole da svoje sistematizovano znanje pokažu pismenim putem, a neki su najuspešniji kada praktično rešavaju zadati problem, kada su u mogućnosti da ispolje svoju kreativnost, intuitivnost, čak i mogućnost inovacije rešenja. Analizirajući neke poznate sisteme, sa aspekta načina na koji studenti mogu da izraze svoje znanje, utvrđen je kriterijum upoređivanja prema načinu izražavanja znanja, koji se bazira na mogućnostima izražavanja znanja kojima sistemi raspolažu. Ovaj kriterijum obuhvata tri dominantna tipa:

verbalno izražavanje (usmeno ili pismeno), vizuelno izražavanje (slikom ili grafikom ili multimedijom) i praktično-kinestetičko izražavanje (individualno ili grupno).

Na osnovu prethodne analize zaključuje se da posmatrani sistemi za testiranje ne prave razlike između individualnih karakteristika pojedinih studenata i da su, kako sa aspekta načina ispitivanja tako i sa aspekta načina izražavanja, svi studenti tretirani na isti način, tj. svi rešavaju iste testove.

Polazeći od činjenice da u mnogim sferama razvoja čovečanstva jača proces individualizacije, predlažemo model koji se bazira na evaluaciji znanja putem adaptivnih testova, pri čemu se ono neće procenjivati prema jednakim i stalnim merilima za sve njegove učesnike, već će se bazirati na njihovim kompetencijama. Ovim testiranjem je potrebno utvrditi afinitete studenta, odnosno način testiranja koji mu najviše odgovara.

Shodno prethodnom, testirali smo jednu grupu studenata (njih 25) i došli smo do sledećih rezultata.

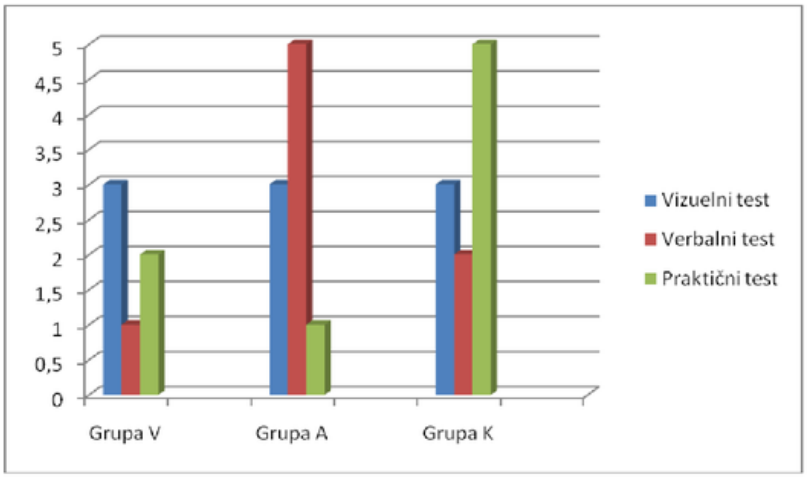
U pomenutom sistemu najpre je obavljeno preliminarno testiranje (statičko testiranje) koje je sprovodeno da bi se ustanovio stil učenja studenata i način izražavanja znanja.

Rezultati testa bili su sledeći: 24% studenata (njih 6) preferira učenje na osnovu viđenja - vizualisti, njih 36% (9 studenata) preferira učenje na osnovu slušanja - auditorni i 40% studenata (njih 10) preferira učenje na osnovu praktične aktivnosti – kinestetički.

Na osnovu ove podele studenti su rangirani u tri grupe (V – vizualisti, A – auditorni, K – praktičari).

Ocenjivanje (proveravanje) znanja je proces skupljanja i kombinovanja informacija iz pojedinih zadataka (na primer testova znanja ili praktičnog rada da bi se ocenio pojedinac ili da bi se uporedio njegov napredak (rezultat) sa određenim kriterijumom [58].

Obavljene su tri vrste testiranja: verbalno pismeno (klasičan test tekstualnog tipa), vizuelno (test čija su pitanja data kroz prikaz slika i grafikona) i praktični test. Interesantno je primetiti da su se rezultati studenata na testovima uglavnom poklapali sa njihovim afinitetima izraženim kroz preliminarno testiranje. Sledeći grafikon pokazuje prethodnu tvrdnju:



Slika 6.22. Grafički prikaz testiranja

Ovakav način provere znanja isključuje sve nedostatke nastavnika kao ocenjivača (isti kriterijum procenjivanja za sve studente). Metrijske karakteristike ovakvih testova su: pouzdanost, valjanost, objektivnost i diskriminativnost. Osim toga ovakve testove karakteriše jednako raspoloživo vreme za rad i isti režim rada - ekonomičnost u ispitivanju, tačno određeni opseg znanja potreban za određenu ocenu, podjednaka i ravnomerna zastupljenost svih delova gradiva, dejstvo faktora sreće i slučajnosti svedeno na minimum.

7. Zaključak

Nastava iz oblasti računarskih mreža se nalazi u preporukama vodećih međunarodnih organizacija kao bitan deo kurseva iz računarskih nauka. Studenti se sa konceptima računarskih mreža susreću na predavanjima i vežbama, a od njih se traži da razumeju apstraktne pojmove o radu ovih sistema. Kao jedan od uobičajenih načina za povezivanje apstraknih koncepata koriste se sistemi za simulaciju i vizualizaciju procesa u računarskoj mreži.

Predmet istraživanja u ovoj disertaciji je komparativna analiza postojećih simulacionih sistema za učenje računarskih mreža i razvoj novog interaktivnog web-zasnovanog sistema za primenu u nastavi iz računarskih mreža.

Komparativna analiza performansi simulatora računarskih mreža nije u velikoj meri zastupljena u stručnoj literaturi, a raspoloživi radovi uglavnom uključuju ograničen skup eksperimenata sa jednostavnim simulacionim scenarijima. U ovom radu se simulatori proučavaju sa aspekta primene u nastavi, pa su u skladu sa tim definisani i odgovarajući kriterijumi za upoređivanje karakteristika simulatora. Za procenu sumulatora uspostavljen je skup relavantinih kriterijuma koji se mogu razvrstati u grupu koja je nazvana kriterijumi pokrivenosti (coverage criteria). Kriterijumi pokrivenosti procenjuju koje teme kurseva računarskih mreža podržavaju pojedini simulatori.

Na osnovu uspostavljenih kriterijuma izvršena je procene selektovanih simulatora i njihova kvantitativna analiza u pogledu upotrebljivosti svakog simulatora u procesu učenja računarskih mreža.

U nastavku teze formirana je metodologija modelovanja računarske mreže i opisan je opšti model računarske mreže koji se zasniva na blokovskoj strukturi. Primarna motivacija modelovanja mrežnog simulatora jeste stvaranje okruženja koje može da ponudi izvođenje laboratorijskih vežbi za veliki broj studenata i obezbeđivanje vizuelne povratne informacije o posmatranom sistemu.

Na osnovu izložene metodologije modelovanja izvršena je implementacija web zasnovane aplikacije za učenje osnova IP umrežavanja. U disertaciji je opisan sistem i

definisane su laboratorijske vežbe. Opisane laboratorijske vežbe osmišljene su tako da pokriju sve glavne teme osnova IP umrežavanja.

Na kraju je izvršena procena upotrebe predloženoog sistema. Obavljene su tri vrste testiranja: verbalno pismeno, vizuelno i praktično testiranje. Sveukupno, zainteresovanost studenata za teme pokrivene laboratorijskim vežbama su veće nego pre upotrebe sistema. Komentari studenata su brojni i naglasak je na tome da ovakav pristup može da pruži odgovore na brojna pitanja vezana za praktičan aspekat učenja računarskih mreža.

Predloženi model moguće je primeniti za simulaciju i vizualizaciju procesa u realnom vremenu u svim sistemima čije su komponente prostorno raspodeljene i koje komuniciraju razmenom poruka.

Dalja istraživanja mogu da se kreću u sledećim pravcima:

- Mogućnost modularizacije sistema uz primenu kompozitnih i komponentnih entiteta
- Modelovanje složenih mrežnih sistema čije komponente komuniciraju slanjem poruka asinhrono isinhrono
- Upotreba popularnih pristupa, kao što je, na primer, spektralna analiza grafova, u daljem modelovanju složenih procesa u računarskoj mreži, a naročito u procesu statičkog i dinamičkog rutiranja, klasterizacije itd.

Literatura

[1] M.Alfano, B.Lenzitti, R.Versace: *On-Lab: A web environment for On-line Labs development*, pp IV.3-1- IV.3-6,CompSysTech 2005,

[2] I.Stiubiener, W.V.Ruggiero, R.M.Silveira, I.Korolkovas, s.Skopp, C.Meiler: *NETLAB: A Framework for Remote Network Experiences*, IEEE Frontiers in Education Conference, 2006.

[3] P. J. Moriarty, B. L. Gallagher, C. J. Mellor,R. R. Baines, "*Graphical computing in the undergraduate laboratory: Teaching and interfacing with LabVIEW*", American Journal of Physics, 2003, Volume 71, Issue 10, pp. 1062-1074

[4] M.Veinović, A.Jevremović, Uvod u računarske mreže, Beograd, 2007.

[5] N.Jovanović, Računarske mreže – edukaciono okruženje zasnovano na web-u, Blace 2003.

[6] End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs Hardcover – November 19, 2004, by Tim Szigeti (Author), Christina Hattingh (Author)

[7] MCSA Windows Server 2012 R2 Complete Study Guide: Exams 70-410, 70-411, 70-412, and 70-417 Paperback – January 27, 2015, by William Panek (Author)

[8] Packet Guide to Core Network Protocols Paperback – June 15, 2011, by Bruce Hartpence (Author)

[9] Architecture of Network Systems (The Morgan Kaufmann Series in Computer Architecture and Design) [Kindle Edition], Dimitrios Serpanos (Author), Tilman Wolf (Author)

[10] The OSI Model: simply explained Paperback – February 1, 2013, by Ronald Schlager (Author)

[11] The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference Hardcover – October 1, 2005, by Charles M. Kozierok (Author)

[12] Encapsulation Technologies for Electronic Applications (Materials and Processes for Electronic Applications) Hardcover – June 25, 2009, by Haleh Ardebili (Author), Michael Pecht (Author)

[13] Network Topology 250 Success Secrets: 250 Most Asked Questions On Network Topology - What You Need To Know Paperback – November 10, 2014, by Bonnie Cooke (Author)

[14] Ethernet: The Definitive Guide Paperback – April 3, 2014, by Charles E. Spurgeon (Author), Joann Zimmerman (Author)

[15] TCP/IP Illustrated, Volume 1: The Protocols (2nd Edition) (Addison-Wesley Professional Computing Series) Hardcover – November 25, 2011, by Kevin R. Fall (Author), W. Richard Stevens (Author)

[16] TCP/IP: Architecture, Protocols, and Implementation with IPv6 and IP Security (McGraw-Hill Computer Communications Series)Paperback – October 10, 1996, by Sidnie Feit (Author)

[17] The Personal Internet Address & Password Log Book Hardcover-spiral – July 4, 2010, by Peter Pauper Press (Author, Editor)

[18] The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference Hardcover – October 1, 2005, by Charles M. Kozierok (Author)

[19] Routers and Routing Basics CCNA 2 Labs and Study Guide (Cisco Networking Academy) Paperback – August 13, 2006, by Allan Johnson (Author)

[20] N. Jovanović, Z. Jovanović, O. Popović, I.Stanković, A.Zakić, Computer Network Simulation and Visualization Tool for Educational Purpose, Telsiks, Niš, Serbia, 2013

[21] Barnett III, B. Lewis. "Netsim: A network performance simulator." Proceedings of the ACM SIGCSE (1993).

[22] Nikolic, Bosko, et al. "A survey and evaluation of simulators suitable for teaching courses in computer architecture and organization." Education, IEEE Transactions on 52.4 (2009): 449-458.

[23] S. Floyd, V. Paxson, "Difficulties in Simulating the Internet", IEEE/ACM Trans. On Networking, vol. 9, no. 4, August 2001, pp. 392-403.

[24] McDonald, C. "Network simulation using user-level context switching." Proceedings of the Australian UNIX Users' Group Conference. Vol. 93. 1993.

[25] "Introduction to Using OPNET Modeler", OPNETWORK 2002, Simulation and Modeling, SYSC 4005/5001.

[26] Scalable Simulation Framework and SSFNet [Online]. Available: http://www.ssfnet.org

[27] Ciraci, Selim, and Bora Akyol. "An evaluation of the network simulators in large-scale distributed simulations." Proceedings of the first international workshop on High performance computing, networking and analytics for the power grid. ACM, 2011.

[28] Riley, George F., and Thomas R. Henderson. "The ns-3 network simulator." Modeling and Tools for Network Simulation. Springer Berlin Heidelberg, 2010. 15-34.

[29] GNS3, GNS3: graphical network simulator, 31 July 2009, http://www.gns3.net/

[30] PENG, Chun-yan, and Bing LIU. "Application of GNS3 at Computer Network Teaching." Theory Research 20 (2010): 136.

[31] Riley, George F. "The georgia tech network simulator." Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research. ACM, 2003.

[32] Bajaj, Lokesh, et al. "Glomosim: A scalable network simulation environment." UCLA Computer Science Department Technical Report 990027 (1999): 213.

[33] Candelas Herías, Francisco Andrés, and Pablo Gil Vázquez. "Practical experiments with KivaNS: A virtual laboratory for simulating IP routing in computer networks subjects." (2009).

[34] Gil Vázquez, Pablo, Francisco Andrés Candelas Herías, and Carlos Alberto Jara Bravo. "Computer networks e-learning based on interactive simulations and SCORM." (2011).

[35] MIMIC simulator, http://www.gambitcomm.com

[36] CNET simulator, http://www.csse.uwa.edu.au/cnet/introduction.html

[37] Veith, Tamie L., John E. Kobza, and C. Patrick Koelling. "Netsim: Java™-based simulation for the World Wide Web." Computers & operations research 26.6 (1999): 607-621.

[38] Vernez, Jérôme, Jürgen Ehrensberger, and Stephan Robert. "Nessi: a python network simulator for fast protocol development." Computer-Aided Modeling, Analysis and Design of Communication Links and Networks, 2006 11th International Workshop on. IEEE, 2006.

[39] Jovanovic, N., R. Popovic, and Z. Jovanovic. "WNetSim: a web-based computer network simulator." International Journal of Electrical Engineering Education 46.4 (2009): 383-396.

[40] V. Timčenko, S. Boštjančič Rakas, M. Stojanović, "Analiza karakteristika simulatora IP mreža NS2 u Linux i Windows okruženju", Zbornik radova TELFOR 2008 (CD),Beograd 2008.

[41] Ciraci, Selim, and Bora Akyol. "An evaluation of the network simulators in large-scale distributed simulations." Proceedings of the first international workshop on High performance computing, networking and analytics for the power grid. ACM, 2011.

[42] Computer Engineering 2004 Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering. Available: http://www.acm.org/education/curric_vols/

[43] Bajaj, Lokesh, et al. "Glomosim: A scalable network simulation environment." UCLA Computer Science Department Technical Report 990027 (1999): 213.

[44] Vernez, Jérôme, Jürgen Ehrensberger, and Stephan Robert. "Nessi: a python network simulator for fast protocol development." Computer-Aided Modeling, Analysis and Design of Communication Links and Networks, 2006 11th International Workshop on. IEEE, 2006.

[45] http://weblab.vpskp.edu.rs

[46] N. Jovanović, Z.Jovanović, A.Zakić, WEB simulator digitalnih logičkih kola, YU-INFO, CD, Kopaonik, 2007.

[47] N. Jovanović, R.Popović, S. Marković, Z. Jovanović, Web Laboratory For Computer Network, *Computer Applications in Engineering Education*, Vol 20, Issue 3, pp 493–502, 2012, ISSN: 1061-3773, http://onlinelibrary.wiley.com/doi/10.1002/cae.20417/abstract

[48] Z. Jovanović, O. Popović, N. Jovanović, COMPARATIVE EVALUATION OF COMPUTER NETWORK SIMULATORS, International Scientific Conference UNITECH, pp. III-339-III-342, Gabrovo, 2010.

[49] N. Jovanović, S. Marković, O. Popović, Z. Jovanović, WEB LABORATORY FOR COMPUTING AND INFORMATICS, International Scientific Conference UNITECH, pp. III-293-III-296, Gabrovo, 2008.

[50] N. Jovanović, S. Marković, Z. Jovanović, A. Zakic, R.Popovic, WEB Based Interactive Digital Logic Circuit Simulator, *JOURNAL of the Technical University of Gabrovo*, vol.37, 2009, pp.58-62 .

[51] Z.Jovanović, O.Popović, N. Jovanović, Pregled alata za simulaciju računarskih mreža u edukacionom okruženju, YU-INFO, CD, pp 688-690, Kopaonik, 2011.

[52] N. Jovanović, Z.Jovanović,O.Popović, Edukacioni sistem za vizualizaciju algoritama, RT 5.3. ETRAN, 2013.

[53] N. Jovanović, Z.Jovanović,O.Popović, Objektno orijentisan pristup modelovanja TCP/IP računarske mreže, RT 1.6. ETRAN, 2012.

[54] N. Jovanović, A.Zakić, Z.Jovanović, I. Stanković, Web Lab: Sistem za učenje osnovnih koncepata IPumrežavanja, INFOFEST, pp.154-158, Budva, 2007.

[55] N. Jovanović, A.Zakić, Z.Jovanović, I. Stanković, Web Lab: Sistem za učenje osnovnih koncepata IP umrežavanja, INFOFEST, pp.154-158, Budva, 2007.

[56] N. Jovanović, Z.Jovanović, A.Zakić, IPKalkulator: Web aplikacija za proračun IP adresa, RT6.2, ETRAN, 2007.

[57] G.Fleming "Learning Styles: Know and Use Your Personal Learning Style"

[58] http://www.businessballs.com/vaklearningstylestest.htm, pristupljeno 2014.

[59] Optimizing Network Performance with Content Switching: Server, Firewall and Cache Load Balancing Paperback – July 12, 2003, by Matthew Syme (Author), Philip Goldie (Author)

[60] Networking Self-Teaching Guide: OSI, TCP/IP, LANs, MANs, WANs, Implementation, Management, and Maintenance Paperback – May 4, 2009, by James Edwards (Author), Richard Bramante (Author)

[61] OSI Reference Model for Telecommunications Paperback – November 27, 200, by Debbra Wetteroth (Author)

[62] TCP/IP Network Administration (3rd Edition; O'Reilly Networking)Paperback – April 11, 2002, by Craig Hunt (Author)

[63] Network Routing: Algorithms, Protocols, and Architectures (The Morgan Kaufmann Series in Networking) Hardcover – April 12, 2007, by Deepankar Medhi (Author)

[64] Computer Networking: Internet Protocols in Action Paperback – January 3, 2005, by Jeanna Matthews (Author)

[65] TCP/IP Protocol Suite (Mcgraw-Hill Forouzan Networking) Hardcover Audiobook, March 25, 2009, by Behrouz A. Forouzan (Author)

[66] Encapsulation Technologies for Electronic Applications (Materials and Processes for Electronic Applications) Hardcover – June 25, 2009, by Haleh Ardebili (Author), Michael Pecht (Author)

[67] Network+ Guide to Networks (with Printed Access Card) Paperback – June 14, 2012, by Tamara Dean (Author)

[68] CompTIA Network+ All-In-One Exam Guide: Exam N10-005 Hardcover – January 9, 2012, by Mike Meyers (Author)

[69] Network Topology 250 Success Secrets - 250 Most Asked Questions On Network Topology - What You Need To Know [Kindle Edition], Bonnie Cooke (Author)

[70] Communication Networks Hardcover – July 16, 2003, by Alberto Leon-Garcia (Author), Indra Widjaja (Author)

[71] CompTIA Network+ Study Guide: Exam N10-006 (Comptia Network + Study Guide Authorized Courseware) Paperback – May 4, 2015, by Todd Lammle (Author)

[72] Networks and Services: Carrier Ethernet, PBT, MPLS-TP, and VPLSHardcover – October 16, 2012, by Mehmet Toy (Author)

[73] Ethernet Networking for the Small Office and Professional Home Office Paperback – June 18, 2007, by Jan L. Harrington (Author)

[74] TCP/IP: Architecture, Protocols, and Implementation with IPv6 and IP Security (McGraw-Hill Computer Communications Series)Paperback – October 10, 1996, by Sidnie Feit (Author)

[75] The Personal Internet Address & Password Log Book Hardcover-spiral – July 4, 2010, by Peter Pauper Press (Author, Editor)

[76] Cisco Routers for the Desperate: Router and Switch Management, the Easy Way [Kindle Edition], Michael W. Lucas (Author)

[77] Routing Protocols Companion Guide Hardcover – March 6, 2014, by Cisco Networking Academy (Author)

[78] CCNA Routing and Switching 200-120 Network Simulator – December 29, 2013, by Wendell Odom (Author), Sean Wilkins (Author)

[79] Elements of Network Protocol Design Hardcover – April 21, 1998, by Mohamed G. Gouda (Author)

[80] Networks: An Introduction Hardcover – May 20, 2010, by Mark Newman (Author)

[81] User Interface Design and Evaluation (Interactive Technologies)Paperback – April 5, 2005, by Debbie Stone (Author), Caroline Jarrett (Author), Mark Woodroffe (Author), Shailey Minocha (Author)

[82] Master IP routing Protocols OSPF EIGRP IS-IS BGP: By Redouane MEDDANE [Kindle Edition], MEDDANE Redouane (Author)

[83] The Practical OPNET User Guide for Computer Network SimulationHardcover – August 24, 2012, by Adarshpal S. Sethi (Author), Vasil Y. Hnatyshin (Author)

[84] Modeling and Simulation of computer Network Using OPNETPaperback – December 17, 2012, by Adil Adil Nazir Malik (Author)

[85] Introduction to Network Simulator NS2 Hardcover – December 2, 2011, by Teerawat Issariyakul (Author), Ekram Hossain (Author)

[86] An Introduction to Network Simulator 3 Hardcover – October 5, 2015, by Jack L. Burbank (Author)

[87] GNS3 Network Simulation Guide Paperback – October 25, 2013, by RedNectar Chris Welsh (Author)

[88] Modeling and Simulation of Computer Networks and Systems: Methodologies and Applications Paperback – May 6, 2015, by Mohammad S. Obaidat (Editor), Faouzi Zarai (Editor), Petros Nicopolitidis (Editor)

[89] Introduction to GloMoSim: Network Simulator Paperback – December 22, 2011, by Ayyaswamy Kathirvel (Author)

[90] "NETSIM" network simulator for the ultracomputer Paperback – April 27, 2015, by Marc Snir (Author)

[91] N. Jovanović, A.Zakić, Z.Jovanović, WEB laboratorija za računarstvo i informatiku, YU-INFO, CD, Kopaonik, 2008.

[92] N. Jovanović, A.Zakić, Z.Jovanović, O.Popović, Simulacija statičkog rutiranja IP paketa u WAN mreži, INFOFEST, pp.131-137, Budva, 2008.

[93] Zoran Jovanović, *Aleksandar Jevremović, Nenad Jovanović, Kriterijumi za evaluaciju računarskih mreža, SYNTHESIS - International Scientific Conference of IT and Business-Related Research, pp-116-118, DOI: 10.15308/Synthesis-2015.*

Dodatak A. Izvorni kod aplikacije za učenje osnova IP umrežavanja

Listing A.1. klasa App

```
class App {
     public static String imeStudenta;
     public static int deo = 0;
     public static String adresa;
     public static String podrazumevanaMaska;
     public static String maska;
     public static int brojPodmreza;
     public static int brojPozajmljenihBitova;

     public static String podmreza;
     public static String prvaAdresa;
     public static String zadnjaAdresa;
     public static String broadcast;
     public static String adresa1;
     public static String adresa2;
     public static String adresa3;
}
```

Listing A.2. klasa BinarnaAdresa

```
public class BinarnaAdresa extends TextField {
    String binarnaAdresa;
    public BinarnaAdresa() {
    super(35);
    setBackground(Color.white);
    this.addFocusListener (new java.awt.event.FocusAdapter() {
    public void focusLost (java.awt.event.FocusEvent evt) {
         mreza1Adresa1FocusLost(evt);
         IPadresa adr = new IPadresa(App.adresa);
         binarnaAdresa = adr.binarnaAdresa();
         if (!(getText()).equals("") && isEditable()== true){
             if (binarnaAdresa.equals(getText())){
                  if (App.deo == 1){
                      Statistika.deo1_brojTacnih++;
                      setForeground(Color.blue);
                  }
             }
             setEditable(false);
         }
     }
     });
}
```

Listing A.3. klasa BinarnaMaska

```
public class BinarnaMaska extends TextField {
    String binarnaAdresa;
    public BinarnaMaska() {
      super(35);
      setBackground(Color.white);
      this.addFocusListener (new java.awt.event.FocusAdapter() {
          public void focusLost (java.awt.event.FocusEvent evt) {
              mreza1Adresa1FocusLost(evt);
              IPadresa adr = new IPadresa(App.adresa, App.maska);
              binarnaAdresa = adr.binarnaMaskaPodmreze();
              if(!(getText()).equals("") && isEditable()== true){
                   if(binarnaAdresa.equals(getText())){
                        if(App.deo == 2){
                            Statistika.deo2_brojTacnih++;
                            setForeground(Color.blue);
                        }
                    }
                    setEditable(false);
              }
          }
      });
    }
}
```

Listing A.4. klasa Broadcast

```
public class Broadcast extends TextField{
    public Broadcast(final String ad, final String ma) {
         super(15);
         setBackground(Color.white);
         this.addFocusListener (new java.awt.event.FocusAdapter() {
         public void focusLost(java.awt.event.FocusEvent evt) {
             mreza1Adresa1FocusLost(evt);
             IPadresa adr = new IPadresa(ad,ma);
             String broadcast = adr.broadcastAdresa();
             if(!(getText()).equals("") && isEditable()== true){
                  if(getText().equals(broadcast)){
                      if(App.deo == 3){
                           Statistika.deo3_brojTacnih++;
                           setForeground(Color.blue);
                           App.broadcast = getText();
                      }
                  } else {
                      App.broadcast = "";
                  }
                  setEditable(false);
              }
         }
    });
}
```

Listing A.5. klasa PodrazumevanaMaska

```
public class PodrazumevanaMaska extends TextField {
    public PodrazumevanaMaska(final String ad) {
        super(15);
        setBackground(Color.white);
        this.addFocusListener (new java.awt.event.FocusAdapter() {
            public void focusLost (java.awt.event.FocusEvent evt) {
                rezaIAdresaIFocusLost(evt);
                IPadresa adresa = new IPadresa(ad);
                String klasa = adresa.vratiKlasu();
                String mmaska ="";
                if (klasa.equals("C")){
                    mmaska = "255.255.255.0";
                }
                if (klasa.equals("B")){
                    mmaska = "255.255.0.0";
                }
                if (klasa.equals("A")){
                    mmaska = "255.0.0.0";
                }
                App.podrazumevanaMaska = mmaska;
                if (!(getText()).equals("") && isEditable()== true){
                    if (mmaska.equals(getText())){
                        if (App.deo == 1){
                            Statistika.deo1_brojTacnih++;
                            setForeground(Color.blue);
                        }
                    }
                    setEditable(false);
                }
            }
        });
    }

/*
    public int dvaNa(int n){
        int rez=1;
        if (n == 0)
            return 1;
        else {
          for (int i = 1; i<=n; i++)
            rez = rez*2;
          return rez;
        }

    }
}
```

Listing A.6. klasa NovaIPadresa

```
public class NovaIPadresa extends Label {
    App aPP = new App();
    public NovaIPadresa() {
           super();
           setBackground(Color.red);
                  String adr="";
                  int br1 = (int)(Math.random()*223);
                         int br2=(int)(Math.random()*255);
                         int br3=(int)(Math.random()*255);
                         int br4=(int)(Math.random()*255);
                         adr = ""+br1+"."+br2+"."+br3+"."+br4;

                  aPP.adresa = adr;
                  this.setFont (new Font ("Dialog", Font.BOLD, 16));
                  this.setBackground (Color.white);
                  this.setForeground (Color.red);

                  this.setText(adr);
    }
}
```

Listing A.7. klasa BrojniSistemi

```
class BrojniSistemi{
      String binarnaVrednost;
      String dekadnaVrednost;
      String oktalnaVrednost;
      String heksadecimalnaVrednost;

      public BrojniSistemi(String vrednost, String sistem){
            if (sistem.equals("B")){
                  binarnaVrednost = vrednost;
                  bimarniUDecimalni();
            }
            if (sistem.equals("D")){
                  dekadnaVrednost = vrednost;
            }
            if (sistem.equals("O")){
                  oktalnaVrednost = vrednost;
            }
            if (sistem.equals("H")){
                  heksadecimalnaVrednost = vrednost;
            }
      }

      public v  bimarniUDecimalni(){
              int n = binarnaVrednost.length();
              int br = 0;
              for (int i = n-1; i >=0; i-- ){
                  br  =  br+  Integer.parseInt(binarnaVrednost.substring(n-
i-
                        1,n-i))*dvaNa(i);
               }
```

```
            dekadnaVrednost = br + "";
      }
      public v 37 oktalniUDecimalni(){
            int n = oktalnaVrednost.length();
            int br = 0;
            for (int i = n-1; i >=0; i-- ){
                br = br+ Integer.parseInt(oktalnaVrednost.substring(n-
i-
                    1,n-i))*osamNa(i);
            }
            dekadnaVrednost = br+ "";
      }
      public void heksadecimalanUDecimalni(){
            int n = heksadecimalnaVrednost.length();
            int br = 0;
            String cifra = "";
            73 vrednostCifre = 0;
            for (int i = n-1; i >=0; i-- ){
                cifra = heksadecimalnaVrednost.substring(n-i-1,n-i);
                if (cifra.equals("A")){
                            vrednostCifre = 10;
                      } else if (cifra.equals("B")){
                            vrednostCifre = 11;
                      } else if (cifra.equals("C")){
                            vrednostCifre = 12;
                      } else if (cifra.equals("D")){
                            vrednostCifre = 13;
                      } else if (cifra.equals("E")){
                            vrednostCifre = 14;
                      } else if (cifra.equals("F")){
                            vrednostCifre = 15;
                      } else {
                            vrednostCifre = Integer.parseInt(cifra);
                      }
                br = br+ vrednostCifre*sesnaestNa(i);
                }

            dekadnaVrednost = br+ "";
      }


   public void decimalanUBinaran(){
       String binarno = "";
       int rez = 0;
       int ostatak =0;
       int x = Integer.parseInt(dekadnaVrednost);
       for(;;) {
          rez = x/2;
          ostatak = x%2;
          binarno = ostatak+binarno;
          x=rez;
          if (rez == 0)
             break;

        }
        binarnaVrednost = binarno;
      }
```

```
public void decimalanUOktalan(){
  String oktalno = "";
  int rez = 0;
  int ostatak =0;
  int x = Integer.parseInt(dekadnaVrednost);
  for(;;) {
     rez = x/8;
     ostatak = x%8;
     oktalno = ostatak+oktalno;
     x=rez;
     if (rez == 0)
        break;

  }
  oktalnaVrednost = oktalno;
}

public void decimalanUHeksadecimalan(){
  String heksadecimalno = "";
  int rez = 0;
  int ostatak =0;
  int x = Integer.parseInt(dekadnaVrednost);
  String cifra = "";
  for(;;) {
     rez = x/16;
     ostatak = x%16;
     if (ostatak == 10)
        cifra = "A";
     else if (ostatak == 11)
        cifra = "B";
     else if (ostatak == 12)
        cifra = "C";
     else if (ostatak == 13)
        cifra = "D";
     else if (ostatak == 14)
        cifra = "E";
     else if (ostatak == 15)
        cifra = "F";
     else
        cifra = ""+ostatak;

     heksadecimalno = cifra+heksadecimalno;
     x=rez;
     if (rez == 0)
        break;

  }
  heksadecimalnaVrednost = heksadecimalno;
}
                    17
 public int dvaNa(int n){
         int rez=1;
         if (n == 0)
            return 1;
         else {
           for (int i = 1; i<=n; i++)
             rez = rez*2;
           return rez;
         }
```

```
    }

      public int osamNa(int n){
             int rez=1;
             if (n == 0)
                return 1;
             else{
               for (int i = 1; i<=n; i++)
                  rez = rez*8;
               return rez;
             }
    }

     public int sesnaestNa(int n){
             int rez=1;
             if (n == 0)
                return 1;
             else{
               for (int i = 1; i<=n; i++)
                  rez = rez*16;
               return rez;
             }
    }
}
```

Dodatak B. Pregled objavljenih radova

Kandidat je kao autor ili koautor objavio sledeće radove:

1. N. Jovanović, R.Popović, S. Marković, Z. Jovanović, Web Laboratory For Computer Network, Computer Applications in Engineering Education, Vol 20, Issue 3, pp 493–52, 2012, ISSN: 1061-3773 (rad sa SCI liste, kategorije M23), http://onlinelibrary.wiley.com/doi/10.1002/cae.20417/abstract
2. N. Jovanović, S. Marković, O.Popović, Z. Jovanović, Managing Network Elements In The Computer Netvork, International Journal of Computer and Electrical Engineering (IJCEE) Vol. 2, No. 2, April, p.p. 320-327, 2010. ISSN: 1793-8163, http://www.ijcee.org/papers/154.pdf
3. Jovanović Nenad, S. Markovićvić, Z. Jovanović, A. Zakic, R.Popovic, WEB Based Interactive Digital Logic Circuit Simulator, JOURNAL of the Technical University of Gabrovo, vol.37, 2009, pp.58-62 . ISSN 1310-6686
4. N. Jovanović, Z. Jovanović, O. Popović, I.Stanković, A.Zakić, Computer Network Simulation and Visualization Tool for Educational Purpose, Telsiks, Niš, Serbia, 2013
5. O. Popović, Z. Jovanović, N. Jovanović, R. Popović, A Comparison and Security Analysis of the Cloud Computing Software Platforms, Telsiks,Niš, Serbia, Vol.2, pp.632-634, 2011
6. N. Jovanović, Z.Jovanović,O.Popović, Edukacioni sistem za vizualizaciju algoritama, RT 5.3. ETRAN, 2013.
7. N. Jovanović, Z.Jovanović,O.Popović, Objektno orijentisan pristup modelovanja TCP/IP računarske mreže, RT 1.6. ETRAN, 2012.
8. Z. Jovanović, O. Popović, N. Jovanović, Comparative Evaluation of Computer Network Simulators, International Scientific Conference UNITECH, pp. III-339-III-342, Gabrovo, 2010.
9. O. Popović, N. Jovanović, Z. Jovanović, Implementation of the WEB Search Engine , International Scientific Conference UNITECH, pp., Gabrovo, 2009.

Biografija kandidata

Zoran M Jovanović je rođen 17.01.1977. u Prištini, republika Srbija. U Prištini je završio osnovnu školu „Miladin Popović“ a zatim i „Prvu prištinsku gimnaziju“ u Prištini, smer prirodno-matematički 1995. godine.

Diplomirao je 2005. godine na „Fakultetu za menadžment“ u Novom Sadu na smeru Menadžment u informatici.

Master studije je upisao školske 2007/2008 godine na univerzitetu Singidunum u Beogradu (smer Savremene informaciono komunikacione tehnologije). Master rad pod nazivom „Simulacioni modeli u nastavi Računarstva i informatike“ kod mentora prof. dr Ranka Popovića odbranio je 2009. godine i stekao akademski naziv diplomirani inženjer poslovne informatike - Master.

Školske 2010/2011 počinje da radi kao predavač na predmetima „Informatika“, „Aplikativni softver i Internet“, „Osnovi Internet tehnologija“ i „Web dizajn“ na Visokoj poslovnoj školi strukovnih studija Blace.

Od školske 2010/2011 bavi se izvođenjem nastave na Visokoj poslovnoj školi strukovnih studija u Blacu na sledećim predmetima:

- Informatika
- Aplikativni softver i Internet
- Osnovi Internet tehnologija
- Web dizajn

Poseduje instruktorski sertifikat za izvođenje ECDL kurseva.

UPOTREBA SIMULACIONIH SISTEMA U NASTAVI IZ RAČUNARSKIH MREŽA - II provera

ORIGINALITY REPORT

17%

SIMILARITY INDEX

PRIMARY SOURCES

1	ponude.biz Internet	1451 words — 6%
2	seminarski.in.rs Internet	281 words — 1%
3	Nenad Jovanovic. "Web laboratory for computer network", Computer Applications in Engineering Education, 2010 CrossCheck	184 words — 1%
4	www.mi.sanu.ac.rs Internet	170 words — 1%
5	www.singipedia.com Internet	170 words — 1%
6	www.amazon.com Internet	94 words — < 1%
7	2008.telfor.rs Internet	81 words — < 1%
8	www.vpskp.edu.rs Internet	65 words — < 1%
9	forums.sun.com Internet	64 words — < 1%
10	www.gambitcommunications.com Internet	63 words — < 1%

11	digituma.uma.pt Internet	60 words — < 1%
12	www.adambarker.org Internet	56 words — < 1%
13	www.its4dev.com Internet	52 words — < 1%
14	www.slideshare.net Internet	47 words — < 1%
15	msts-travnik.net Internet	47 words — < 1%
16	www.citeulike.org Internet	46 words — < 1%
17	k45kxc02.weebly.com Internet	43 words — < 1%
18	autoskola-l.com Internet	38 words — < 1%
19	www.ftn.kg.ac.rs Internet	37 words — < 1%
20	Doina Bein. "Efficient Broadcasting by Selective Forwarding", Studies in Computational Intelligence, 2009 CrossCheck	35 words — < 1%
21	www.amazon.ca Internet	32 words — < 1%
22	www.trendamericas.com Internet	32 words — < 1%
23	quantrimang.edu.vn Internet	32 words — < 1%

24	worldwidescience.org Internet	32 words — < 1%
25	0353-7129, 2004 Publications	31 words — < 1%
26	www.singipedia.singidunum.ac.rs Internet	31 words — < 1%
27	www.cs.columbia.edu Internet	30 words — < 1%
28	www.amazon.co.uk Internet	30 words — < 1%
29	leda.elfak.ni.ac.rs Internet	30 words — < 1%
30	www.caslab.queensu.ca Internet	28 words — < 1%
31	Tiernan, P.. "Enhancing the learning experience of undergraduate technology students with LabVIEW(TM) software", Computers & Education, 201012 CrossCheck	27 words — < 1%
32	www.a1outlet.com Internet	26 words — < 1%
33	Internet	24 words — < 1%
34	0b4af6cdc2f0c5998459-c0245c5c937c5dedcca3f1764ecc9b2f.r43.cf2.rackcdn.com Internet	24 words — < 1%
35	Suzana Marković. "Adaptive distance learning and testing system", Computer Applications in Engineering Education, 2010 CrossCheck	24 words — < 1%

36	www.cet.rs Internet	23 words — < 1%
37	turhancoban.com Internet	22 words — < 1%
38	www.actapress.com Internet	22 words — < 1%
39	www.preisonaut.de Internet	22 words — < 1%
40	www.fit.edu.rs Internet	22 words — < 1%
41	www.ece.uah.edu Internet	22 words — < 1%
42	www.1coolwebsite.co.uk Internet	21 words — < 1%
43	aln.coe.ttu.edu Internet	21 words — < 1%
44	Marković, Marko, Ivana Kostić Kovačević, Olivera Nikolić, and Boško Nikolić. "INSOS-educational system for teaching intelligent systems : INSOS", Computer Applications in Engineering Education, 2014. CrossCheck	20 words — < 1%
45	habble.fse.cs.ritsumei.ac.jp Internet	20 words — < 1%
46	docs.mipro-proceedings.com Internet	19 words — < 1%
47	tbook.com Internet	19 words — < 1%
48	svarog.metropolitan.edu.rs Internet	19 words — < 1%

49	crnarupa.singidunum.ac.rs Internet	19 words — < 1%
50	www.cet.co.yu Internet	19 words — < 1%
51	www.tbook.com Internet	18 words — < 1%
52	eprints.usq.edu.au Internet	18 words — < 1%
53	www.sim.uni-hannover.de Internet	18 words — < 1%
54	sharif.edu Internet	17 words — < 1%
55	undergraduate.csse.uwa.edu.au Internet	17 words — < 1%
56	lumbungbuku.com Internet	17 words — < 1%
57	dydaktyka.akademiasieci.pl Internet	17 words — < 1%
58	best-ebook.appspot.com Internet	16 words — < 1%
59	www.maturskiradovi.net Internet	15 words — < 1%
60	www.otpornik.com Internet	15 words — < 1%
61	Ding, . "Front Matter", Advances in Network Management, 2009. CrossCheck	15 words — < 1%
62	sunzi1.lib.hku.hk	

	Internet	14 words — < 1%
63	www.coder.tv Internet	14 words — < 1%
64	no-border-yosoku.net Internet	14 words — < 1%
65	solvethiserror.com Internet	13 words — < 1%
66	bs.scribd.com Internet	13 words — < 1%
67	www.mltlab.com Internet	12 words — < 1%
68	"Capella University Launches Three New Graduate Degree Programs in Public Safety and Information Tech", Business Wire, Feb 22 2007 Issue Publications	12 words — < 1%
69	Dupljanin, Djordjije, Momcilo Kujacic, and Dragana Sarac. "Managment of electronic services of the Serbian postal system by benchmarking index", Tehnika, 2014. CrossCheck	12 words — < 1%
70	www.amazon24.co.kr Internet	11 words — < 1%
71	www.crimsonbird.org Internet	11 words — < 1%
72	www.mid.gov.me Internet	11 words — < 1%
73	open-mind.ir Internet	10 words — < 1%

74	rua.ua.es Internet	10 words — < 1%
75	digit.inet.co.yu Internet	10 words — < 1%
76	forums.techarena.in Internet	10 words — < 1%
77	training.osisoft.com Internet	10 words — < 1%
78	singidunum.ac.rs Internet	9 words — < 1%
79	drupal.6krpm.jp Internet	9 words — < 1%
80	www.infofest.com Internet	9 words — < 1%
81	Pizzonia, Maurizio, and Massimo Rimondini. "Netkit: network emulation for education : NETKIT: NETWORK EMULATION FOR EDUCATION", Software Practice and Experience, 2014. CrossCheck	9 words — < 1%
82	www.mk.co.yu Internet	9 words — < 1%
83	mreze.layer-x.com Internet	9 words — < 1%
84	www.maturski.net Internet	9 words — < 1%
85	www.snap-scan.com Internet	8 words — < 1%

86	www.buchladen-mv.de Internet	8 words — < 1%
87	www.formatex.org Internet	8 words — < 1%
88	Dobrilovic, Dalibor, Vesna Jevtic, Zeljko Stojanov, and Borislav Odadzic. "The design guidelines for virtual network laboratories", 2012 IX International Symposium on Telecommunications (BIHTEL), 2012. CrossCheck	8 words — < 1%
89	Samcovic, Andrej a. "Application of multimedia technologies in distance learning", 2012 20th Telecommunications Forum (TELFOR), 2012. CrossCheck	8 words — < 1%
90	www.unizg.hr Internet	8 words — < 1%
91	maturskiradovi.net Internet	8 words — < 1%
92	svezaweb.dzaba.com Internet	8 words — < 1%
93	xldt.co.uk Internet	8 words — < 1%
94	apeironsrbija.edu.rs Internet	8 words — < 1%
95	"Revue de livres", Journal de Physique, 1987 CrossCheck	8 words — < 1%
96	itand.ru Internet	8 words — < 1%
97	www.linux.hr Internet	8 words — < 1%

98 Margetić, Nikola. "Mišljenja učitelja/nastavnika o pravilniku o načinima, postupcima i elementima vrednovanja učenika u osnovnoj i srednjoj školi", Napredak, 2013.
Publications

8 words — < 1%

EXCLUDE QUOTES	OFF	EXCLUDE MATCHES	OFF
EXCLUDE BIBLIOGRAPHY	OFF		