# Protokoli za nadzor i konfiguraciju mreže računala

**Stojanović, Marko**

**SVEUČILIŠTE U ZAGREBU**

**PRIRODOSLOVNO–MATEMATIČKI FAKULTET**

**MATEMATIČKI ODSJEK**

Marko Stojanović

# PROTOKOLI ZA NADZOR I KONFIGURACIJU MREŽE RAČUNALA

Diplomski rad

Voditelj rada:
prof. dr. sc. Robert Manger

Zagreb, rujan, 2015.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

*Zahvaljujem se svom mentoru prof. dr. sc. Robertu Mangeru i Krunoslavu Komugoviću, sistem administratoru pri PMF - MO, za pomoć i podršku prilikom izrade ovog rada.*

# Sadržaj

# Uvod

Računala su svakim danom sve veći dio našeg života, što privatno što poslovno. Svakim danom sve više ovisimo o njima te je potreba za njihovim ispravnim radom, međusobnom komunikacijom i razmjenom podataka sve važnija. Postavljanje računala u mrežu omogućuje njihovu međusobnu komunikaciju, ali i stvara potrebu za održavanjem mreže u funkciji. To je posao sistem administratora.

Administracija mreže zahtjeva njenu konfiguraciju te nadzor njenog rada i sprečavanje ili popravak kvarova u njenom radu. Konfiguracija mreže računala podrazumijeva njeno definiranje i postavljanje mrežnih parametara, logičko oblikovanje mreže (topologija mreže, broj i vrste mreža) i izvedba svega navedenog.

Nadzor računalne mreže je pojam koji opisuje nadgledanje parametara rada mreže; da li prijenos podataka unutar jedne ili između više mreža teče bez poteškoća i da li su svi potrebni servisi za rad mreže u funkciji.

Posao administratora mreže otežava činjenica da postoji više od jedne vrste mreža. Svaka vrsta mreže ima svoj način komunikacije i razmjene podataka, a podaci moraju često putovati između dvije ili više različitih mreža. To putovanje podataka između različitih mreža nam omogućuju protokoli koji definiraju oblik u kojem podaci moraju biti bez obzira na mrežu.

U prvom poglavlju ovog rada dan je pregled osnovnih pojmova i tehnologija vezanih za mreže računala. Tu su definirane i opisane vrste mreža, TCP/IP stog protokola, paketi koji putuju mrežom te vrste topologija mreže.

Drugo poglavlje posvećeno je nekim osnovnim protokolima koji omogućavaju rad jedne ili više mreža zajedno. Protokoli ARP, IP, TCP, i UDP su temeljni protokoli TCP/IP stoga i oni su zaslužni za prijenos podataka mrežom (bolje rečeno cijelim Internetom).

Treće poglavlje opisuje rad i važnost nekih protokola za nadzor i konfiguraciju mreže

računala. Protokoli DNS, SNMP, HTTP i DHCP pomažu sistem administratoru da uspješno obavlja svoj posao.

Četvrto poglavlje opisuje izradu i rad projektnog zadatka "PMF - Matematički odsjek Nadzor računalne mreže" koji je vezan uz ovaj rad. Predstavljene su tehnologije korištene u izradi projektnog zadatka te funkcionalnost izrađene aplikacije.

# Poglavlje 1

# Mreže računala

U ovom poglavlju uvedeni su i opisani osnovni pojmovi koji se javljaju u mrežama računala i bitni su za razumijevanje daljnih dijelova ovog rada.

## 1.1 Pojam mreže i protokola

Mreža računala ili podatkovna mreža je telekomunikacijska mreža koja omogućava razmjenu podataka među računalima. U mreži računala, mrežni uređaji prenose podatke jedan drugom putem mrežne poveznice (žičani ili bežični prijenos). Podaci se prenose u obliku paketa. Mrežni uređaji koji započinju prijenos, prosljeđuju ili zaprimaju podatake nazivaju se mrežni čvorovi. Mrežni čvorvi uključuju uređaje poput osobnih računala, pametnih telefona, serverskih računala ali i mrežne opreme poput sklopki (*eng.* switch), usmjernika (*eng.* router) i koncentratora (*eng.* hub).

**Definicija 1.1.1.** Mreža računala *je skup samostalnih računala koja mogu međusobno komunicirati tako da razmjenjuju poruke preko nekog medija za prijenos podataka.*

Za dva uređja kažemo da su povezana ako mogu razmjenjivati podatke, direktno ili preko trećeg uređaja, tj. skupa uređaja. Računalne mreže razlikujemo po vrsti medija koji prenosi signal, komunikacijskim protokolima, veličini i topologiji. U večini slučajeva koriste se protokoli koji su slojeviti, tj. u svom radu koriste protokole koji su standard u telekomunikacijskim protokolima. Najpoznatija i najveća mreža je *Internet*. [2], [7]

Komunikacijski *protokoli* su skup pravila kojima se razmjenjuju podaci u računalnalnom sustavu (računalnoj mreži) sa svojstvom da čvorovi ne moraju biti u dirktnoj interakciji sa hardverom.

**Definicija 1.1.2.** Protokol *je skup pravila koja definiraju format i značenje poruka putem kojih se odvija komunikacija dva računala ili dva programa. Ista riječ "protokol" može označavati i softver kojim se realizira određeni skup pravila za komunikaciju.*

Komunikacijski problemi se organiziraju kao stog slojeva (*eng.* layering model). Jedan od najpoznatijih stogova protokola je 5-slojni TCP/IP stog protokola koji se danas koristi u praksi (postoji i 7-slojni OSI model stoga protokola ali on nikad u praksi nije implementiran, koristi se za neka teorijska razmatranja o protokolima ([8])). Slojevi TCP/IP stoga protokola su (odozdo prema gore): fizički, mrežni ili vezni, internet, transportni te aplikacijski sloj. TCP/IP stog protokola i njegovi slojevi će biti detaljnije opisani u kasnijem dijelu ovog rada.

Komunikacija u mrežama računala se, uz protokole, odvija pomoću *ulaza*.

**Definicija 1.1.3.** Ulaz *(eng. port) je logički konstrukt koji definira servis ili proces. Ulaz se označava sa 16-bitnim brojem kojeg nazivamo* broj ulaza *(eng. port number).*

Ulazi omogućuju protokolima da razlikuju procese koji rade na istom računalu. Drugim riječima, Ulazi omogućuju slanje poruke od jednog procesa do drugog procesa, umjesto od jednog računala do drugog računala. Protokoli koji primarno koriste ulaze za komunikaciju su TCP i UDP protokoli, iz transportnog sloja TCP/IP stoga protokola, koji brojeve ulaza pošiljatelja i primatalja poruke imaju zapisane u svom zaglavlju. Ulaz je uvijek pridružen IP adresi nekog čvora i nekom tipu protokola za komunikaciju. Specifični brojevi ulaza su po konvenciji rezervirani za određene servise i protokole. Kažemo da neki servis ili protokol *slušaju* na nekom ulazu kad postoji rezerviran ulaz na kojem servis ili protokol očekuju zahtjev. Na primjer, na nekom serveru koji ima dodjeljenu IP adresu, HTTP protokol sluša na ulazu 80 i koristi TCP protokol za prijenos podataka, SNMP na ulazima 161 i 162 i koristi UDP protokol[1]. Treba razlikovat takve logičke ulaze od fizičkih ulaza na mrežnim uređajima.

## 1.2 Vrste mreža

Broj računala koja su danas spojena na mrežu broji se u stotinama miliona. Uz stalni porast memorijskih i procesorskih kapaciteta računala, i njihovu brojnost, pruža se mogućnost da se ti resursi upotrijebe na nove načine. Takav trend nužno dovodi do zahtjeva za novim načinom povezivanja, što sa softwarske strane (protokoli) što samog umrežavanja. [8]

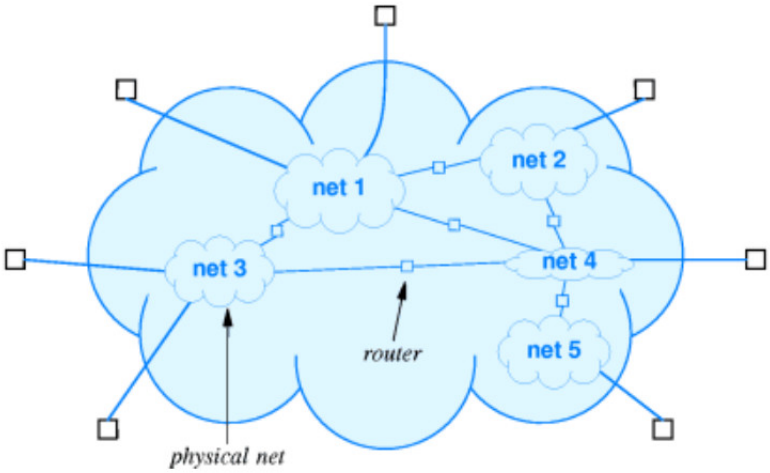
Tri su osnovne vrste mreža na koje danas nailazimo:

[1] TCP, UDP, SNPM i HTTP su protokoli TCP/IP stoga i detaljnije su objašnjeni u poglavljima 2 i 3.

- *LAN - Lokalna mreža (eng. Local Area Network)*: manja mreža koja se stastoji od umreženih računala u, na primjer, jednoj zgradi. Za povezivanje se koristi jedna određena tehnologija (npr. Ethernet) s jednim određenim protokolom nižeg sloja.
- *WAN - Rasporstranjena (globalna) mreža (eng. Wide Area Network)*: mreža koja je veća od LAN mreže i povezuje računala koja su udaljenija, na primjer u različitim gradovima. U WAN mreži se osim samih računala nalaze i posebni telekomunikacijski uređaji *sklopke* (*eng.* switch) koji omogućuju povezivanje udaljenih dijelova mreže i prijenos podataka.
- *internet* - skup raznorodnih mreža (LAN i WAN) koje su međusobno povezane tako da izvana djeluju kao jedna mreža. Za povezivanje raznorodnih mreža koristi se uređaj *usmjernik* (*eng.* router). Taj uređaj je istovremeno čvor u obje mreže koje povezuje i njegova uloga je da, osim što je fizička veza između dvije mreže, omogući konzistentnost protoka podatak između dvije mreže tako što konvertira podatke i usmjerava ih prema njihovom odredištu. Da bi komunikacija računala u takvoj mreži bila moguća važno je da sva računala i usmjernici u mreži koriste iste protokole za usmjeravanje i transport podataka. Najpoznatija mreža te vrste danas je *Internet* (s velikim "I") koji koristi protokole TCP/IP stoga (vidi sliku 1.1). Važno je napomenuti da je Internet virtualna mreža koja je dobivena hardversko-softverskim povezivanjem raznorodnih LAN-ova i WAN-ova. Iluziju jedne virtualne mreže daje IP protokol (više na stranici 15) koji svim čvorovima (bez obzira u kojoj fizičkoj mreži se oni nalazili) pridružuje jedinstvenu IP adresu, te omogućuje razmjenu poruka referenciranjem na IP adrese umjesno na fizičke adrese.

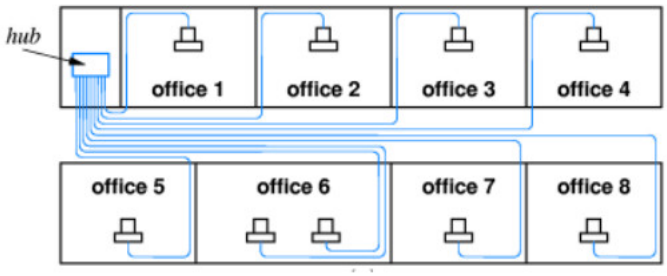
Razlike između LAN i WAN mreža se mogu generalizirati na sljedeći način:

1. Kao što je već rečeno, LAN mreža je manja mreža, unutar jedne zgrade ili nekoliko bliskih zgrada. Računala su međusobno povezana *koncentratorima* (*eng.* hub). Na slici 1.2 je prikazana jedna LAN mreža.
2. U većini slučajeva, vlasnik mrežne opreme je isti kao i vlasnik računala koja su povezana u LAN mrežu. Kod WAN mreže je bitna razlika ta što je dio mrežne opreme u vlasništvu neke druge, vanjske, kompanije (uglavnom su to telekomunikacijske kompanije koje pružaju Internet usluge). To ima dvije vrlo bitne posljedice. Prvo, treba uzeti u obzir cijene zakupa za mrežnu opremu ili cijene prijenosa podataka koju određuje vanjska kompanija koja je vlasnik opreme. Drugo, održavanje LAN mreže koja je spojena na WAN mrežu je obveza i odgovornost vlasnika LAN mreže, tj. korisnik neke LAN mreže nemože utjecati na kvarove mrežne opreme u valsništvu vanjske kompanije. Na slici 1.3 je prikazana jedna WAN mreža gdje su veze između sklopki dio mrežne opreme koje bi u stvarnom svijetu bile iznajmljene od vanjske kompanije.




Slika 1.1: Prikaz Interneta.

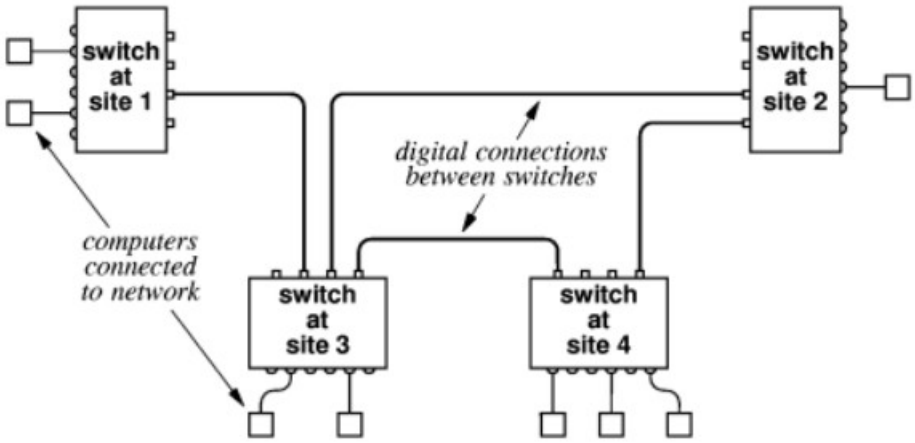
3. Brzine unutar LAN mreže su veće nego u WAN mreži.




Slika 1.2: Prikaz mreže računala koja je ostvarena pomoću koncentratora - LAN.

## 1.3 Slojevi protokola

Komunikacija između raznorodnih mreža zahtjeva da su protokoli koji obavljaju preusmjeravanje podataka jednaki. Prvi model za složene mrežne protokole bio je 7-slojni *OSI* model, ali on nikad nije implementiran do kroja. 1982. godine predložen je 5-slojni *TCP/IP* ili *Internet protocol suite* stog protokola koji je i danas u upotrebi te omogućuje funkcional-




Slika 1.3: Prikaz mreže računala koja je ostvarena pomoću sklopke - WAN.

nost Inetrneta kakvog danas poznajemo.

**Definicija 1.3.1.** Protokol *je apstrakcija koja definira skup pravila po kojoj čvorovi u mreži mogu izmjenjivati poruke bez da direktno stupaju u interakciju s hardverom.*

Apstraktni koncept stoga protokola je jednostavan. Pošiljatelj prolazi po slojevima stoga odozdo prema gore dok primatelj poruku obrađuje u suprotnom smijeru.
TCP/IP stog protokola se sastoji od sljedećih slojeva (odozdo prema gore):

- *fizički sloj* - sloj koji služi za obradu informacija na hardvreskoj razini (npr. mrežna kartica pretvara bitove u impulse koji putuju dalje mrežom do odredišta)
- *mrežni* ili *vezni sloj* - na ovom sloju su informacije za prosljeđivanje podataka između internet slojeva dvije različite mreže i služi kao veza između fizičkog i internet sloja
- *internet sloj* - ovaj sloj je zadužen samo za usmjeravanje paketa između dvije potencijalno različite mreže. Na ovom sloju se odvijaju dvije osnovne stvari; adresiranje i identifikacija pošiljatelja pomoću IP adresa i slanje paketa do sljedećeg čvora koji je najbliži primatelju. Na ovom sloju su zapisani i podaci o protokolima koji se koriste na višim slojevima te su oni numerirani prirodnim brojevima (na primjer u transportnom sloju se koriste protokoli Internet Control Message Protocol (ICMP, dijagnostičke informacije) sa oznakom 1 i Internet Group Management Protocol (IGMP, upravlja IP podacima) sa oznakom 2).

- *transportni sloj* - prenosi podatke o protokolima koji određuju vrstu komunikacije između dva uređaja na mreži, na primjer ovdje se nalaze protokoli za detekciju grešaka, segmentaciju paketa i aplikacijskog adresiranja (broj ulaza na usmejrniku). Dvije su osnovne kategorije za vrstu komunikacije između dva uređaja: konekcijski orijentirano (TCP protokol) ili bezspojno orijenitrano (UDP protokol).

- *aplikacijski sloj* - sadrži protokole koje aplikacije koriste kako bi korisniku dostavile neki servis ili kako bi dvije aplikacije razmijenile podatke koji su obrađeni na prethodnim slojevima. Protokoli aplikacijskog sloja tretiraju protokole nižih slojeva kao "crnu kutiju" koja omugućava vezu između dva mrežna čvora. Podaci aplikacijskog sloja enkapsulirani su u transportnom sloju (putem TCP ili UDP protokola). Neki od poznatijih protokola ovog sloja su HTTP, FTP, SNMP...

TCP/IP stog protokola nema nikakvih zahtjeva na specifične hardverske ili softverske komponente nekog čvora u mreži. Jedino što je potrebno je da hardver i softver znaju slati i primati poruke prema pravilima stoga. Razvoj mrežnih aplikacija uglavnom zahtjeva poznavanje protokola u transportnom i aplikacijskom sloju, dok se za ostale slojeve brine operativni sustav. U tablici 1.1 je dan prikaz gornja četri sloja TCP/IP stoga i nekih protokola koji se tamo nalaze (fizički sloj nije naveden jer je to čisto hardverska komponenta i nije zanimljiva za naša razmatranja). U nekim slojevima treba obratit pozornost na postojanje podslojeva. Neki protokoli se odvijaju između dva sloja koji su gore opisani, ali postoji određena hijerarhija među njima.

| | |
|---|---|
| Aplikacijski sloj | DNS, TLS/SSL, FTP, HTTP, IMAP, POP3, SMTP, SNMP, SSH, TELNET... |
| | Neki protokoli za usmjeravanje mogu biti dio aplikacijskog ali i Internet sloja (BGP - border gateway protokol). |
| Transportni sloj | TCP, UDP, DCCP, SCTP, IL, RUDP |
| Internet | Protokoli za usmjeravanje koji se odvijaju u IP protokolu se smatraju dijelom Internet sloja (OSPF). |
| | IP (IPv4, IPv6) |
| | ARP, InARP |
| Mrežni sloj | Ethernet, Wi-Fi, ATM |

Tablica 1.1: Prikaz TCP/IP stoga s protokolima.

## 1.4 Mrežni uređaji

Mrežni uređaji se dijele na dvije osnovne kategorije: *veze* i *čvorove*.

Veze su prijenosni medij kojim podaci putuju od čvora do čvora i pripadaju fizičkom sloju u korištenju protokola. Glavna podjela veza je na žičanu i bežičnu vezu. Neki od standardnih medija u žičanim vezama su:

- *koaksijalni kabel* - medij koji se koristi za prijenos podataka u televizijskim sustavima i poslovnim objektima. Sastoji se od bakrene ili aluminijske jezgre obložene izolacijskim materijalom koji je umotan u vodič koji je prekriven još jednim slojem izolacije.
- *parica* - oklopljena (UTP) ili neoklopljena (UTP i koaksijalni kabel). UTP kabel sastoji se od dvije isprepletene bakrene žice. Oklopljena parica je današnji standard.
- *optička vlakna* - staklena vlakna koja dopuštaju prijenos signala u obliku svjetlosnih impulsa. *LED* diode ili laserska svjetlost različitih frekvencija putuju medijem i njihov puls je signal u poruci. Prednosti optičkih medija su mali gubitak podataka, otpornost na električne smetnje i paralelno slanje više poruka (svaka poruka koristi svjetlost druge frekvencije). Mana im je krhkost staklenih vlakana prilikom savijanja. Koriste se za veze kod kojih je potrebna velika propusnost podataka i za veze na velike udaljenosti (npr. međukontintalne veze položene na dnu mora).

Mediji bežičnog prijenosa su:

- zemljani *mikrovalovi* - komunikacija se vrši u nižem spektru gigahertz valova. Komunikacija je limitirana na to da se uređaji moraju nalaziti u *vidnom polju* jedan od drugoga (standarna udaljenost je oko 42 km).
- *komunikacijski sateliti* - mikrovalni radio valovi se šalju između satlelita stacioniranih u orbiti iznad ekvatora. Koriste se za prijenos mnogih vrsta signala.
- *radio i širokospektralna tehnologija* - bežične lokalne mreže koriste visokofrekventne i niskofrekventne radio valove, tj. radio valove širokog spektra. Osnovni standard za bežičnu komunikaciju je definiran standardom IEEE 802.11 i popularno je poznat kao *Wi-Fi*.

Gore navedeni primjeri veza povezuju mrežne čvorove. *Mrežni čvorovi* su svi uređaji koji odašilju, prosljeđuju i zaprimaju podatke koji se prenose vezama.

*Mrežno sučelje* (*eng.* Network Interface Controller, NIC) ili mrežna kartica je hardverski dio mrežnog čvora koji omogućuje čvoru da pristupi vezi te obrađuje informacije koje

se nalaze u nižim slojevima stoga protokola. U *Ethernet* vrsti mreža svako mrežno sučelje ima svoju jedinstvenu *MAC* (*eng.* Media Access Control) adresu koja je pohranjena u trajnu memoriju samog sučelja.

*Ponavljač* (*eng.* repeater) je elektronički uređaj koji prima signal koji putuje mrežom, očisti ga od šumova te ga ponovo pošalje prema odredištu. Njegova uloga je da omogući prenošenje poruke na velike udaljenosti sa velikom točnosti. U pravilu se kod mreža koje kao vezu koriste oklopljenu paricu ponavljač postavlja na svakih 100 metara duljine kabla, dok se kod optičkih kablova te udaljenosti između ponavljača povečavaju i na desetke kilometara. Oni djeluju na fizičkom sloju stoga protokola i potrbno im je neko kraće vrijeme kako bi propagirali dobiveni signal, pa time mogu dovesti do latencije u slanju i primanju poruka na krajnjem čvoru.

*Koncentrator* (*eng.* hub) je ponavljač koji ima više priključaka (ulaza). U moderno doba polako izlazi iz upotrebe i zamjenjuje se sklopkama, dok ponavljači i dalje ostaju u upotrebi za dugačke veze (položene na dnu oceana).

*Sklopka* (*eng.* switch) je uređaj koji filtrira i prosljeđuje datagrame drugog sloja TCP/IP stoga protokola koristeći pri tome MAC adrese koje su tamo zapisane. Razlika između sklopke i koncentratora je u tome što sklopka prosljeđuje podatke na fizički ulaz gdje podaci trebaju ići za razliku od koncentratora koji podatke šalje svima. Ako sklopka ne zna na koji ulaz treba poslati podatke onda pošalje svima. Sklopka "uči" povezivati MAC adrese i na kojem su one ulazu. Sklopke uglavnom imaju više ulaza, te stoga imaju zvjezdastu topologiju i mogu biti međusobno povezane preko nekog ulaza.

*Vatrozid* (*eng.* firewall) je mrežni uređaj koji upravlja mrežnom sigurnosti i pravilima pristupa. Uglavnom su podešeni tako da prihvaćaju zahtjeve poznatih izvora (na primjer IP adrese oblika 192.168.xxx.yyy) dok se ostale odbacuju. Vatrozidi imaju sve veću važnost u mrežnoj sigurnosti s povećanjem *cyber* napada.

## 1.5 Paketi

Podaci koji putuju mrežom između dva mrežna čvora su kontinuirani niz bitova. Takvim načinom prijenosa može doći do gubitka podataka u prijenosu. Stoga se svi podaci dijele na manje nizove i stvaljaju u *pakete* koji se šalju zasebno. Pošiljatelj dijeli poruku u pakete koji tada nezavisno putuju mrežom, a primatelj skuplja te iste pakete te ih spaja u prvotnu poruku.

Prijenos podataka putem paketa ima višestruke prednosti:

- *Efikasnije i pravednije korištenje zajedničkih resursa.* Kada bi se kroz zajednički resurs slale kontinuirane poruke, tada bi jedan par računala mogao zauzeti resurs, a drugi bi morali dugo čekati da dođu na red. Razbijanjem poruka u pakete postiže se vremensko dijeljenje zajedničkog resursa.
- *Mogućnost da paketi paralelno putuju različitim putovima kroz mrežu.* Time se ubrzava prijenos podataka.
- *Lakše ispravljanje grešaka u prijenosu podataka.* Ako dođe do greške u prijenosu tada je potrebno ponovno prenjeti samo jedan manji paket.

Korištenje paketa ima i svoje mane. Određeni slojevi protokola moraju se baviti dijeljenjem poruka u pakete, te kasnijim sortiranjem i ponovnim sastavljanjem paketa u poruke. Nije moguće garantirati propusnost veze između dva računala. Budući da veza nije ekskluzivno rezervirana za jednu poruku, prijenos podataka može se usporiti zbog dijeljenja resursa s drugim porukama.

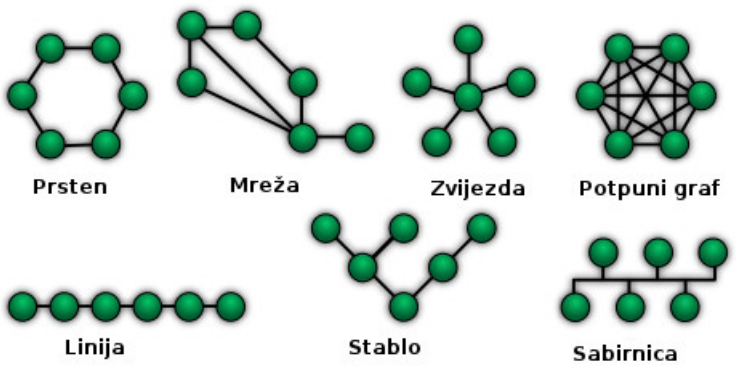
Svaka mrežna tehnologija definira u detalje kako izgledaju paketi koji se mogu prenositi kroz tu vrstu mreže. Općenitu ideju paketnog prijenosa nije moguće ralizirati na svim slojevima TCP/IP stoga, stoga svaki sloj uvodi drugi pojam za vrstu paketa koja se u njemu koristi. Mrežni sloj koristi pojam paketa ili okvira, internet sloj i transportni sloj koriste pojmove datagrama i segmenta. Ovi nazivi se koriste kako bi se naglasilo kojoj strukturi paketa i sloju stoga protokola neki paket pripada. [2], [3]

## 1.6 Topologija i geografija mreže

Fizički razmještaj mreže računala je manje važan nego topolgija same mreže, tj. bitnije nam je kako su čvorovi povezani nego gdje se oni fizički nalaze. Dijagrami mreža računala stoga prikazuju topologiju mreže a ne geografiju, dok su elementi dijagrama čvorovi i veze u mreži.

Slika 1.4 prikazuje jedan od općenitih načina prikaza mrežne topologije, drugi, detaljniji, način prikaza topologije je kada u čvorove i veze unesemo stvarne elemnte koji vrše tu funkciju (koncentratori, sklopke, vrste kablova koji ih povezuju itd.).

Geografija mreže računala prikazuje fizički položaj mrežne opreme, i naglasak je na njihovoj fizičkoj lokaciji a ne na odnosu same opreme unutar mreže.




Slika 1.4: Prikaz različitih oblika mrežnih topologija, kružiči predstavljaju čvorove a linije veze.

# Poglavlje 2

# Temeljni internetski protokoli

U ovom poglavlju navedeni su neki temeljni internetski protokoli koji su važni za iduće poglavlje. Svi protokoli su u TCP/IP stogu protokola i oni omogućavaju postojanje internet mreža.

## 2.1 ARP - Address Resolution Protocol

*Address Resolution Protocol* (ARP) se nalazi na trećem sloju TCP/IP stoga. Njegova zadaća je da pruži mapiranje između fizičke (MAC) i mrežne (npr. IP) adrese. U tablici 1.1 (stranica 8) vidimo da je ARP protokol naveden ispod IP protokola, koji se nalazi u trećem sloju, jer je on veza između protokola na nižem sloju (npr. Ethernet) i IP protokola. Ovaj protokol je vrste *zatraži-i-odgovori* te djeluje samo unutar jedne mreže (npr. LAN) i nikada se ne usmjerava na vanjske mreže (preko sklopki). Mapiranje (ARP tablica) IP-MAC se učitava sa sklopki prilikom pokretanja operativnog sustava.

Primjer paketa za ARP protokol dan je u tablici 2.1. Dana tablica ilustrira korištenje IPv4 i Ethernet mrežu. Polja THA i SHA su 48-bitna, SPA i TPA 32-bitna iz čega slijedi da je veličina cijelog paketa 24 bajta. EtherType za ARP je 0x0806 (taj podatak se javlja i u zaglavlju Ethernet protokola u mrežnom sloju TCP/IP stoga).

**Primjer 2.1.1.** *Neka se dva računala A i B nalaze u uredu i spojena su na LAN mrežu ethernet kablom koji je spojen na sklopku. Računalo A želi poslati paket računalu B. Pomoću DNS-a računalo A pronalazi da je IP adresa računala B 192.168.0.55. Kako bi poruka bila poslana potrebno je znati i MAC adresu od B. Prvo A pregledava unaprijed učitanu ARP tablicu i traži ako postoji MAC adresa (00:eb:24:b2:05:ac) pridružena IP adresi od B. Ako je postoji MAC adresa pridružena IP adresi tada se paket može poslati na tu MAC adresu. Ako nije bilo pridružene MAC adrese IP adresi tada A šalje ARP zahtjev*

| | |
|---|---|
| Hardware type (HTYPE) | Polje koje označava mrežni protokol.<br>Ethernet ima oznaku 1. |
| Protocol type (PTYPE) | Polje za intertworking protokol koji ARP zatražuje.<br>IPv4 ima oznaku 0x0800. |
| Hardware length (HLEN) | Duljina (u bajtovima) tražene hardveverske adrese.<br>Ethernet ima duljinu 6. |
| Protocol length (PLEN) | Duljina (ubajtovima) adrese koja se koristi u<br>protokolima gornjih slojeva.<br>IPv4 ima duljinu 4. |
| Operacija | Označava operaciju koju pošiljatelj zahtjeva.<br>1 za zahtjev, 2 za odgovor. |
| Sender hardware address (SHA) | U ovom polju je zapisana adresa pošiljatelja<br>ako je operacija zahtjev.<br>Ako je operacija odgovor ovdje je zapisana adresa<br>čvora kome je zahtjev poslan.<br>Sklopke ne pregledavaju ovo polje kada<br>rade mapiranje IP-MAC. |
| Sender protocol address (SPA) | Internetworking adresa pošiljatelja. |
| Target hardware address (THA) | U ARP zahtjevu ovo je polje prazno,<br>u odgovoru piše adresa onoga koji je zahtjev poslao. |
| Target protocol address (TPA) | Internetworking adresa primatelja. |

Tablica 2.1: Prikaz polja jednog ARP paketa na Ethernet mreži sa IPv4 adresiranjem. Lijevo se nalaze imena polja a desno opis njihovog sadržaja.

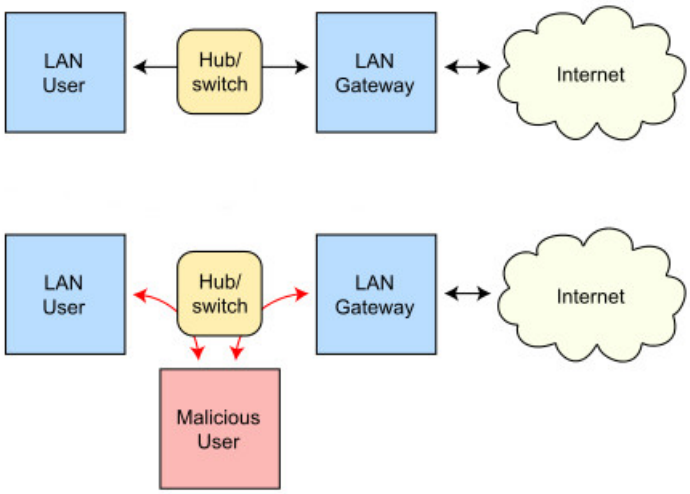
*svim računalima na mreži (MAC adresa je postavljena na FF:FF:FF:FF:FF:FF) i traži odgovor od IP adrese 192.168.0.55. B tada šalje odgovor sa svojom MAC (i IP) adresom. B i A unose nove podatke u svoje ARP tablice. Poruka sada može biti poslana.*

ARP protokol može se koristiti i za osvježavanje ARP tablica svih računala na mreži. Takvo korištenje ARP protokola naziva se još i *ARP upoznavanje* i vrši se tako što se SPA upiše u TPA i THA se postavi na 0 te se pošalje ARP zahtjev. Ovakav zahtjev ne traži odgovor, već kad ga ostala računala zaprime ona osvježe svoje ARP tablice. Alternativno može se poslati i ARP odgovor takav da TPA=SPA i THA=SHA. Oba slučaja, zahtjev i odgovor, su moguća jer se ARP operacija izvršava nakon što se provjeri i osvježi ARP tablica. Mnogi operativni sustavi šalju ARP upoznavanje prilikom pokretanja kako bi sva računala u mreži imala osvježene ARP tablice. [7]

Postoji i Inverse ARP protokol (InARP), tj. ARP protokol koji radi u suprotnom smijeru, iz MAC adrese pronalazi IP adresu. InARP koristi isti oblik paketa kao ARP ali druge oznake za operacije.
Postoji još jedan oblik ARP protokola, Reverse ARP (RARP) koji je mapirao adrese trećeg sloja dva različita mrežna čvora. Ovaj protokol više nije u uptrebi.

ARP protokol nema sistem autentifikacije te ga to čini pogodanim za prikupljanje podataka u svrhu cyber napada. Slika 2.1 prikazuje napad koji se naziva *ARP spoofing* gdje se neki neovlašteni korisnik spojio na mrežu te sluša promet ARP paketa. Skuplja podatke o ARP tablicama te presjeca put između paketa koji putuje između dva računala (A i B), predstavljajući se kao računalo primatelj. Takvi napadi se nazivaju *man-in-the-middle*.

Funkcionalnost ARP protokola za IPv6 adrese pruža Neighbor Discovery Protocol (NDP).

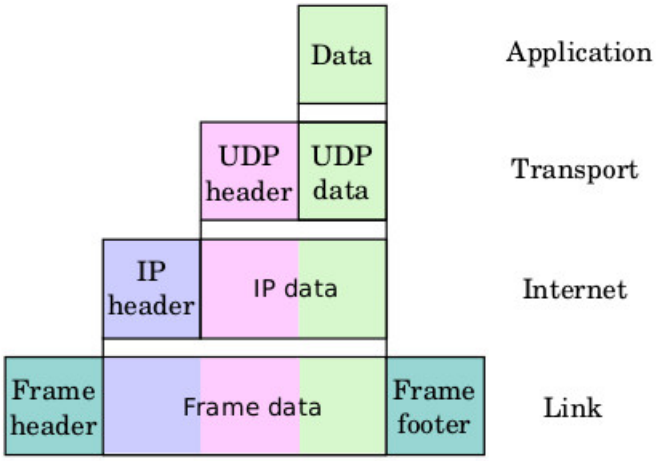



Slika 2.1: ARP-spoofing: na gornjem dijelu je prikazan normalan tok podataka u mreži dok je dolje prikazan man-in-the-middle cyber napad.

## 2.2 IP - Internet Protocol

IP protokol ima funkcionalnost usmjeravanja datagrama kroz mrežu i time se ostvaruje interworking što je temelj Interneta. Zadatak IP je da dostavlja datagrame, od pošiljatelja do primatelja, samo na osnovi IP adrese koja je zapisana u zaglavlju datagrama. U tu svrhu

IP definira enkapsuliranu strukturu datagrama i pruža sistem adresiranja koji ima dvije funkcije: identifikaciju mrežnih čvorova i pružanje servisa za usmjeravanje datagrama. Svaki datagram ima dvije komponente: *zaglavlje* i *teret*. U zaglavlju su zapisane IP adresa pošiljatelja, IP adresa primatelja i drugi podaci potrebni za dostavljanje datagrama. Teret je podatak koji se prenosi. Takav način umetanja tereta u datagram sa zaglavljem naziva se *enkapsulacija* (slika 2.2).




Slika 2.2: Primjer enkapsulacije u datagram po slojevima TCP/IP stoga.

*IP adresiranje* podrazumijeva dodjeljivanje IP adrese i ostalih pridruženih parametara mrežnim sučeljima čvorova. IP adresa se satoji od dva dijela: *prefiksa* i *sufiksa*. Prefiks identificira adresu mreže dok sufiks identificira čvor na mreži. Nikoje dvije fizičke mreže ne mogu imati istu mrežnu adresu i nikoja dva čvora u fizičkoj mreži ne mogu imati isti sufiks. Svaki čvor ima jedinstvenu IP adresu koja je uređeni par (prefiks, sufiks).

IP adrese su 32 bitni brojevi, ali se najčešće pišu u dekadskom sustavu kako bi bili lakše čitljivi. Notacija IP adresa je takva da se po 8 bitova zapiše u decimalnom broju i takve skupine su odjeljene točkom, na primjer adresu 11000000 00000101 00110000 00000011 zapisujemo kao 192.5.48.3. Dijeljenje na sufiks i prefiks određuje *adresna maska* (često se koristi samo pojam maska [8]). Maska je 32 bitni broj koji počinje nizom uzastopnih jedinica i završava nizom uzastopnih nula. Jedinice označavaju duljinu prefiksa a nule duljinu sufiksa IP adrese kojoj su pridružene. Jednostavniji zapis pisanja para adresa-maska se vrši pomoću *CIDR notacije* (Classless Inter Domain Routing) tako što se desno od IP adrese kosom crtom (zankom '/') zapiše dekdskim brojem duljina prefiksa. Na primjer 192.5.48.3/16 označava masku koja se sastoji od 16 jedinica i 16 nula (u dekadskom obliku

255.255.0.0) i za dani IP daje prefiks 192.5.0.0.

Usmjeravanje pomoću IP adresa vrši se od strane svih čvorova, ali najvažniju ulogu u tome imaju usmjernici koji prenose pakete preko različitih mreža. Usmjernik je čvor u dvije mreže, a protokoli za usmjeravanje ga tretiraju kao i bilo koji drugi čvor u mreži. Zbog toga svaki usmjernik po TCP/IP protokolu ima svoju IP adresu. Štoviše, svaki usmjernik ima barem dvije pridružene IP adrese, budući da je usmjernik čvor u više fizičkih mreža i svaka IP adresa ima prefiks koji označava fizičku mrežu. [2], [7]

## 2.3 TCP - Transmission Control Protocol

*Transmission Control Protocol* (TCP) je složeniji od dva komunikacijska prtokola. Nalazi se u četvrtom, transportnom, sloju TCP/IP stoga, i uz IP je jedan od centralnih protokola tog stoga. TCP je pouzdan servis koji kontrolira da paketi dolaze u redosljedu u kojem su poslani te da će primljeni podaci biti identični onima koji su poslani.
Osnovne zančajke TCP protokola su:

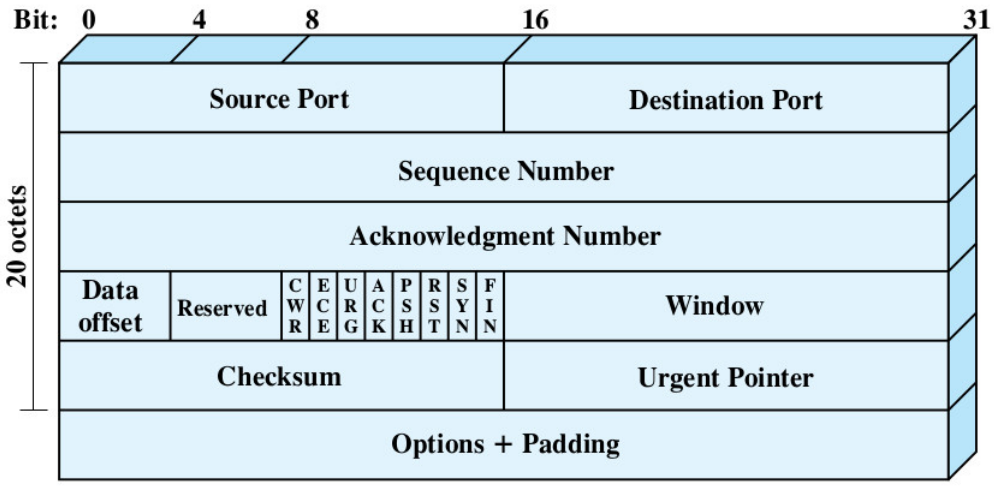
- *Spojna usluga*: aplikacijski program mora prvo zatražiti vezu, a tek onda slijedi prijenos podataka.
- *Point-To-Point* (*host-to-host*, *čvor-čvor*): svaka TCP veza ima točno dva kraja.
- *Puni dupleks*: oba aplikacijska programa mogu slati podatke u svakom trenutku. Omogućuje i pretpostavlja optimizaciju korištenjem komunikacije u oba smjera.
- *Pouzdanost*: protokol osigurava da će podaci doći u redoslijedu u kojem su poslani i da neće biti gubitka ili duplikacije podataka.
- *Pouzdano otvaranje veze*: pri stvaranju čvor-čvor veze, oba čvora moraju pristati na komunikaciju. Paketi koji kasne iz prethodnih veza među tim čvorovima neće interferirati s novom vezom.
- *Pouzdano zatvaranje veze*: TCP osigurava da će svi poslani podaci biti isporučeni prije nego li se veza raskine.

TCP datagram se sastoji od zaglavlja i dijela sa podacima. Prilikom slanja, poruka se dijeli na *komade* koji se smještaju u dio s podacima i njemu se pridružuje zaglavlje. Takav paket (još se naziva i segment prema [8]) se tada enkapsulira u IP (slično kao na slici 2.2 na strani 16). Zaglavlje TCP segmenta sastoji se od sljedećih osnovnih dijelova:

- Polja SOURCE PORT i DESTINATION PORT označavaju ulaze preko kojih će se odvijati komunikacija. DESTINATION PORT je broj ulaza na kojem primatelj sluša

dolazne zahtjeve, a SOURCE PORT je broj ulaza na kojem pošiljatelj očekuje odgovor. Ova polja moraju biti popunjena kako bi komunikacija bila moguća.

- Polja ACKNOWLEDGMENT NUMBER i WINDOW se odnose na dolazeći stream. ACKNOWLEDGMENT NUMBER sadrži SEQUENCE NUMBER sljedećeg paketa, a WINDOW daje informaciju o slobodnom dijelu međuspremnika za podatke koji polaze iz čvora kojem se šalje potvrda.
- Polje SEQUENCE NUMBER se uvijek odnosi na izlazeći stream i pokazuje na prvi oktet koji se nalazi u segmentu.
- CHECKSUM sadrži kontrolnu sumu za TCP zaglavlje i podatke.




Slika 2.3: Zaglavlje TCP segmenta.

Djelovanje TCP protokola može se podijeliti u 3 faze: otvaranje veze, prijenos podataka i zatvaranje veze. Za otvaranje veze koristi se algoritam *trostrukog rukovanja*. Otvaranje veze započinje time što pošiljatelj primatelju pošalje *SYN segment* na koji primatelj odgovara sa *SYN-ACK* i na kraju pošiljatelj šalje *ACK* čime je veza uspostavljena i otvorena. Koraci 1 i 2 uspostavljaju konekcijske parametre u jednom smjeru dok koraci 2 i 3 uspostavljaju te parametre u drugom smjeru. Time je uspostavljen puni dupleks.
Zatvaranje veze se koristi algoritmom *četvorostrukog rukovanja*. Kada neki čvor A želi prekinuti vezu šalje segment *FIN* na koji čvor B odgovara sa ACK i šalje svoj FIN segment. Nakon toga A odgovori sa ACK na FIN od B i čeka neko vrijeme (time-out) te se veza zatvara. Ovakav način zatvaranja veze omogućava da samo jedna strana prekine vezu

(veza je *polu-otvorena*). Čvor koji je zatvorio vezu više ne može slati pakete ali ih može primati sve dok drugi čvor ne prekine vezu. [8]
Zatvaranje veze moguće je izvesti i pomoću trostrukog rukovanja tako što drugi čvor odgovara sa jednim *FIN-ACK* segmentom umjesto sa dva odvojena. [2]
Podaci koji se šalju TCP-om su okteti ali ih promatramo kao jedan kontinuirani niz. Svaki od okteta je numeriran nekim brojem modulo $2^{32}$ koji je zapisan u SEQUENCE NUMBER svakog poslanog segmenta. TCP sam odlučuje kako će segmentirati podatke koje šalje. Na primateljovoj strani podaci se spremaju u memoriju a TCP sam odlučuje kada će ih predati korisniku. Zastavica PUSH u zaglavlju označava da se paket objavi odmah po primitku. Ako primatelj primi segment koji nema ispravan SEQUENCE NUMBER javlja prošiljatelju da opet pošalje taj segment.

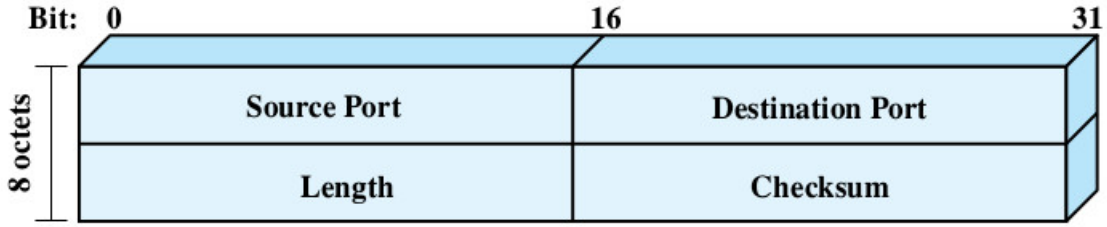
TCP ima sigurnosnih slabosti. Napad uskračivanjem usluge moguće je izvesti tako što se *IP spoofingom* mogu slati SYN segmenti zatim ACK segmenti u velikom broju što opterećuje primatelja tih segmenata. Ovaj tip napada je poznat pod imenom *SYN poplava*.
Zlonamjerni korisnik može "prisluškivati" vezu druga dva korisnika (korisnik A je pošiljatelj a korisnik B primatelj). Prisluškivanjem TCP protokola može saznati SEQUENCE NUMBER segmenta koji se trenutno razmjenjuje. Zlonamjerni korisnik tada napravi segment koji označi sa SEQUENCE NUMBER segmenta koji sljedeći treba biti prihvaćen od korisnika B. Korisnik B tada prihvati taj segment jer je SEQUENCE NUMBER u redu. Kada korisnik B javi korisniku A da je primio segment s danim SEQUENCE NUMBER dolazi do pucanja sinkronizacije prijenosa jer A još nije poslao segment s tim SEQUENCE NUMBER. Ova vrsta napada naziva se *otimanje veze*.

## 2.4 UDP - User Data Protocol

*User Data Protocol* (UDP) je uz TCP važan protokol za prosljeđivanje poruka. Nalazi se u transportnom sloju TCP/IP stoga kao i TCP. Ovaj protokol se zasniva na jednostavnoj bezspojnoj tehnologiji koja ne zahtjeva da se uspostovi komunikacijski kanal između dva čvora. Koristi se kada provjera i ispravljanje grešaka mogu biti obavljeni od strane aplikacije koja poruku prime. Time se oslobađaju resursi na mrežnom sučelju i poruke se mogu brže razmjenjivati. Najčešća upotreba UDP protokola je u aplikacijama koje ovise o realnom vremenu (na primjer Internet telefonija, prijenos videa uživio i sl.) i u kojima je čekanje da neki paket pristigne nedopustivo.

UDP datagram se satoji od zaglavlja i tereta (prostora za podatke). Zaglavlje je jednostavno (slika 2.4) i sastoji se od samo nekoliko elemenata. Razlog tome je što UDP koristi IP protokol za svu komunikaciju (slika 2.2 na stranici 16) i aplikacijskim programima pruža istu (naslijeđenu) semantiku best-effort komunikacije koju ima i IP protokol. U zaglavlju

se nalaze i polja za ulaze na kojima će se izvršavati komunikacija. UDP koristi cijeli raspoloživi raspon broja ulaza (0 - 65535), i polje broja ulaza pošiljatelja ne treba nužno biti popunjeno.




Slika 2.4: Zaglavlje UDP datagram.

UDP ne dijeli poruke u pakete i ne sastavlja poruke po primitku. Svaka poruka koju neki program pošalje direktno se prenosi kao zasebni IP datagram kroz Internet do krajnjeg odredišta. Rezultat korištenja opisanog sučelja je da UDP poruke mogu uzrokovati neefikasno korištenje fizičke mreže. Ukoliko aplikacijski program šalje male UDP poruke omjer korisnog tereta i zaglavlja će biti loš. Protokol takvo korištenje mreže ne ograničava. Ukoliko program šalje jako velike poruke, UDP protokol ne osigurava da datagrami koji se šalju neće biti veći od MTU-a[1] mreže. Nedostatak UDP protokola je što slanje (pre)velikih poruka usporava komunikaciju. Ponekad UDP šalje poruke koje IP protokol mora fragmentirati već na polaznom računalu. [3], [8]

## 2.5 Razlike između TCP i UDP protokola

TCP je konekcijski orijentiran protokol što znači da zahtjeva *rukovanje* kako bi se ostvarila end-to-end veza. Jednom kada je veza uspostavljena podaci se mogu slati u oba smjera između čvorova. Svojstva TCP protokola su:

- Pouzdanost - TCP zna upravljati potvrdama, ponovnim slanjem i vremenom čekanja na poruke. Poruku je moguće poslati više puta ako je potrebno, ako se poruka "zagubila" u prijenosu primatelj će zatražiti ponovno slanje poruke koja mu nedostaje. U TCP-u ne može doći do gubljenja podataka. U slučaju da se na poruku treba čekati više puta, veza se prekida.
- Poruke stižu redosljedom kojim su poslane. Ako segmenti pristignu drukčijim redosljedom nego su poslane, TCP sprema u memoriju sve segmente koje prima dok

[1] Najveća cjelina za prijenos (Maximum Transmission Unit - MTU) je veličina podataka u baytovima koju neki protokol može prenjeti u jednom slanju paketa.

podaci ne budu presloženi u ispravan poredak kako bi bili prosljeđeni aplikaciji koja prima podatke.

- Težak je s gledišta resursa. TCP zahtjeva 3 poruke samo za uspostavljanje konekcije, prije nego što razmjena podataka započne. Postoji kontrola zagušenja mreže. Ako dolazi do kašnjenja u prijenosu segmenata, TCP smanjuje svoj prostor za podatke eksponencijalno do polovice normalnog prostora. Kada se zagušenje mreže smanji prostor za podatke se linearno proširuje a vrijeme između dva slanja segmenta se povećava.

- Stream interface - sučelje koje TCP pruža aplikacijskim programima omogućuje slanje kontinuiranih nizova okteta kroz čvor-čvor vezu. TCP ne definira pojam zapisa koji ima fiksnu veličinu. Primatelj ne mora čitati pristigle podatke u komadima kako ih je pošiljatelj poslao, već ih opet čita kao kontinuirani niz okteta.

UDP je jednostavniji protokol za bezspojno prosljeđivanje poruka. Komunikacija se odvija jednosmjerno bez predhodne potrebe za provjerom da li je primatelj spreman primiti poruku. Svojstva UDP protokola su:

- Nepouzdanost - kada se poruka pošalje putem UDP protokola ne možemo biti sigurni da će ona i stići na svoje odredište. UDP nema osobine potvrđivanja primljene poruke, ponovnog slanja izgubljenih poruka ili isteka vremena čekanja na poruku. Jednom kada se poruka izgubi ne može se opet poslati.

- Poruke poslane UDP protokolom nemaju redni broj te ne moraju nužno stizati u redosljedu u kojem su i poslane. Zbog toga ne troši puno resursa.

- Nema nikakvu kontrolu zagušenja mreže. Ako dođe do zagušenja paketi mogu kasniti ili se izgubiti.

# Poglavlje 3

# Protokoli za nadzor i konfiguraciju mreže računala

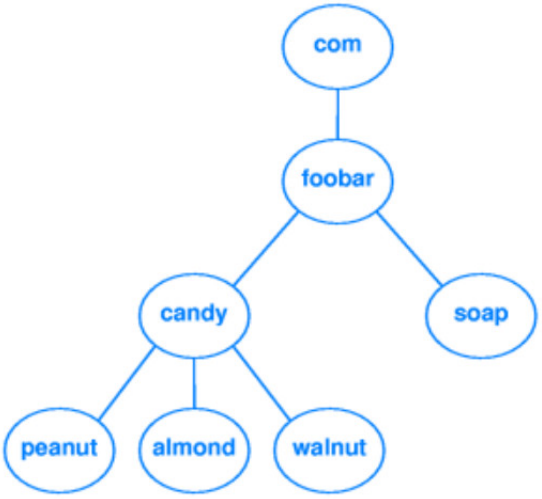
U ovom poglavlju je dan pregled nekih protokola koji se koriste za nadzor i konfiguraciju mreže. Opisani protokoli nisu svi u TCP/IP stogu ali navedeni su ovdje zbog njihove važne uloge u konfiguraciji mreže računala.

## 3.1 DNS - Domain Name System

*Domain Name System* (DNS) je poslužitelj koji prevodi simbolička imena računala u IP adrese. Nalazi se u petom, aplikacijskom, sloju TCP/IP stoga protokola. Ovaj prtokol je nužan za funkcionalnost većine servisa koji koriste Internet. Baza podataka koja sadrži veze IP adresa i simboličkih imena se ne nalazi na jednom računalu već je distribuirana između mnogo DNS poslužitelja. Struktura simboličkih imena računala je strogo hijerarhijska, s najvažnijim dijelom imena na krajnjem desnom kraju. Na primjer adresa `wallnut.candy.foobar.com` pripada hijerarhiji koja je prikazana na slici 3.1. Ta hijerarhija mogla bi odgovarati nekoj korporaciji, a njeni dijelovi podružnicama. Čvorovi u hijerarhiji na najnižoj razini obično odgovaraju konkretnim računalima unutar podružnica ili odjela. Svi DNS poslužitelji znaju kako se povezati s vršnim poslužiteljem (root serverom) i kako se povezati s poslužiteljima koji su odgovorni za pod-domene koje su niže u hijerarhiji. [3]

Dijelovi hijerarhije zovu se domene. Domene na vrhu hijerarhije zovu se vršne ili top-level domene (na primjer com, org, edu i sl.), i one su pod kontrolom ustanove koja se zove Internet Corporation for Assigned Names (ICANN). Domene odmah ispod vršne kontrolira ustanova koju je ovlastio ICANN.

Četri su osnovna dijela od kojih se DNS sastoji:




Slika 3.1: Hijerarhija simboličkih imena.

- *Prostor domenskih imena*: DNS koristi strukturu označenog stabla kako bi pronalazio domene.
- *DNS baza podataka*: konceptualno, svaki čvor i list u DNS stablu sadrži informacije o tom čovoru ili listu, između ostalog i IP adresu. Te informacije su pohranjene u tablici resursa (*eng.* resource record (RR)). RR je pohranjen u distribuiranoj bazi podataka.
- *Name serveri*: serverski servisi koji sadrže informacije o stablu domenskih imena i pripadajući RR.
- *Resolver*: programi koji uzimaju informacije od name servera na klijentov zahtjev. Tipičan zahtjev je IP adresa čvora za odgovarajuću domenu.

DNS ima više vrsta tablica resursa (RR) koji su u upotrebi. Svaki zapis u tablici ima svoj tip (ime i broj), vrijeme koliko je valjan, klasu i specifične podatke za taj zapis. Svi upiti na DNS putem IP protokola odgovaraju sa istim podacima. IME je puno domensko ime čvora (lista) u stablu. TIP ukazuje na format podataka s kojim server radi i daje naznaku svoje svrhe. Na primjer, A zapis se koristi za prevođenje domenskog imena u IPv4 adresu, MX zapis određuje koji se e-mail server koristi za domenu neke e-mail adrese. RDATA daje opis specifičan za server poput prioriteta i domenskog imena e-mail servera.

Poruke za DNS prenose se uglavnom UDP protokolom. Komunikacija se sastoji od jedne poruke zahtjeva i jednog odgovora od strane servera. TCP se koristi kada su podaci

za DNS veći od 512 bytova. [8]

DNS ima sigurnostnih propusta. Funkcija DNS-a je prevođenje domenskog imena u IP adresu. Lažni DNS (poznati i kao "rogue" DNS) server se može konfigurirati tako da preuzima upite za neke poznate domene (na primjer banke ili neke velike kompanije) i vrati IP adresu neke druge lokacije. DNS je podložan i promjenama zbog sigurnosnih propusta na drugim mjestima. DNS pravila se mogu promjeniti zbog loše sigurnosti sučelja za upravljanje usmjernika. "Zombi" računala nekada zlonamjernim softverom znaju promjeniti postavke na kućnim usmjernicima na neki rouge DNS.

## 3.2 SNMP - Simple Network Management Protocol

*Simple Network Management Protocol* (SNMP) je standarni protokol za nadzor i konfiguraciju računalnih mreža. Nalazi se u petom, aplikacijskom, sloju TCP/IP stoga. SNMP definira način kako manager komunicira s agentom[1]. Dakle, SNMP definira format i značenje managerovih zahtjeva odnosno agentovih odgovora ([3]). Uređaji koji koriste SNMP su usmjernici, sklopke, pisači, radne stanice i mnogi drugi. Trenutna verzija je SNMPv3. Tri osnovne komponente od kojih se sastoji mreža koja koristi SNMP su: uređaji kojima se upravlja, agent i softver kojim manager upravlja (Network management station - NMS).

SNMP koristi UDP protokol za prijenos podatak između managera i agenta. UDP se koristi umjesto TCP zbog svojeg svojstva bezspojnosti. Zbog toga što UDP ne nudi nikakav sistem za potvrdu primitka poslane poruke, SNMP ima opciju time-outa koja određuje vrijeme čekanja na odgovor pa se ponovno šalje paket ako je potrebno. Ta mana UDP-a ne ometa SNMP protokol u izvršavanju njegove zadaće. Naprotiv, to je poželjno. UDP ima manje pakete koji ne zagušuju jako mrežu, a SNMP će administrator koristiti tek kada dođe do nekog kvara na mreži (na primjer do zagušenja). SNMP koristi ulaz 161 za UDP komunikaciju (zamka ili trap se šalje preko ulaza 162, više na strani 25).[5]

Skup svih objekata unutar uređaja kojima SNMP može pristupiti zove se *Baza upravljačkih informacija* (Management Information Base - MIB). SNMP zapravo ne definira MIB. Umjesto toga, SNMP standard samo definira format poruke i način kako se poruke kodiraju. Za imena objekata u MIB koristi se općenita hijerarhijska shema ASN.1 s dugačkim prefiksima. Osigurano je da će imena biti jedinstvena. Na primjer, brojač IP datagrama koje je uređaj primio zove se

[1]Da bi se naglasila razlika između aplikacija za “obične” korisnike i onih za mrežne administratore, kod sustava za upravljanje mrežama izbjegavaju se termini "klijent" i "poslužitelj". Aplikacijski program na administratorovom računalu naziva se manager, a aplikacijski program na mrežnom računalu zove se agent.

`iso.org.dod.internet.mgmt.mib.ip.ipInReceives`.

Kad se ime objekta prikaže unutar SNMP poruke, svaki dio imena pretvara se u određeni cijeli broj. Spomenuto ime brojača IP datagrama unutar SNMP poruke izgleda:

`1.3.6.1.2.1.4.3`.

SMP koristi 7 *protokolnih podatkovnih jedinica* (protocol data unit - PDU):

- GetRequest: manager-to-agent zahtjev za dohvaćanje neke varijable.
- SetRequest: manager-to-agent zahtjev za promjenu neke varijable. Varijabla i njena nova vrijednost nalaze se u tijelu zahtjeva. Agent vraća odgovor sa novom vrijednosti promjenjenih varijabli.
- GetNextRequest: manager-to-agent zahtjev za popis svih dostupnih varijabli nekog uređaja. Odgovor se sastoji od varijable, koja se leksikografski u MIB-u nalaze iza tražene varijable, i njene vrijednosti. Popis svih varijabli dohvaća se tako što se GetNextRequest ponavlja a prvi zahtjev ima identifikacijski broj objekta (object identifier - OID) postavljen na 0.
- GetBulkRequest: optimizirana verzija GetNextRequest za dohvaćanje više OID-a odjednom
- Response: odgovor agenta na zahtjeve GetRequest, SetRequest, GetNextRequest, GetBulkRequest i InformRequest.
- Trap: poruka koju agent šalje manageru (NMS-u) o promjenama satusa bez da manager traži zahtjev za time. Ove poruke također koriste UDP ali NMS "sluša" na ulazu 162 za ovu vrstu poruke. Poruka se satoji od sistemskog vremena klijenta, OID-a koji označava zamku i varijabli sa njihovim novim vrijednostima.
- InformRequest: managerov odgovor na agentovu Trap poruku da je poruka zaprimljena.

Zbog mogućnosti SNMP protokola da mijenja vrijednost varijabli kod agenta (SetRequest) i korištenje UDP protokola za komuikaciju, cyber napadi su jako opasni. SNMP verzije 1 i 2 nisu koristili nikakvu enkripciju pa je postojala velika opasnost da zlonamjerni korisnik prisluškuje i mijenja sadržaj paketa koji putuju mrežom. SNMP verzije 3 ima enkripciju pa je takav scenarij teže ostvariv. Postoje načini da se SNMP implementira da koristi TCP prtokol, ali to se rijetko čini jer takva komunikacija zagušuje mrežu.

## 3.3 HTTP - Hypertext Transfer Protocol

*Hypertext Transfer Protocol* (HTTP) je najvažniji protokol za ostvarivanje World Wide Web-a (WWW). Nalazi se u petom, aplikacijskom, sloju TCP/IP stoga. *Hypertext* (hipertekst) je strukturirani tekst koji koristi logičke veze (hyperlinks) između čvorova koji sadrže tekst. HTTP ima puno veće mogućnosti nego što mu samo ime govori. Ovaj protokol može prenositi i druge vrste podataka osim hiperteksta, poput teksta, videa i slike. HTTP djeluje na principu zahtjev-odgovor u klijent-server paradigmi. HTTP prezentira podatke neke web stranice korisniku preko web preglednika. Tada je web prekglednik klijent a web server (npr. Apache) je server za našu paradigmu.

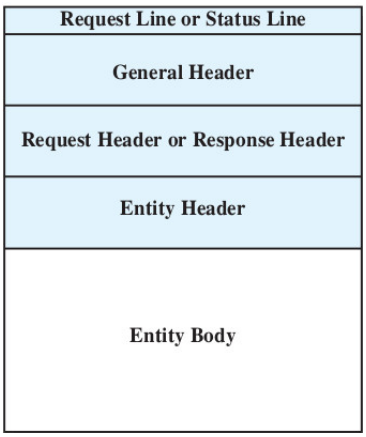
HTTP sesija je niz zahtjeva i odgovora koji putuju mrežom. Klijent započinje komunikaciju uspostavom TCP konekcije za određeni ulaz na kojem server sluša (standarni ulazi za HTTP su 80 i 8080) te onda šalje HTTP zahtjev. Zahtjev se sastoji od metode (GET , PUT, DELETE, POST, OPTIONS, HEAD, TRACE, CONNECT, DEBUG...), adrese (Uniform Resource Locator (URL)) i poruke koja sadrži parametre zahtjeva i informacije o klijentu. Kada server primi zahtjev pokuša izvršiti klijentov zahtjev te nakon toga šalje odgovor. Odgovor se sastoji od koda za uspijeh ili neuspjeh, informacije o odgovoru i tražene podatke (na primjer HTML dokumentom). TCP konekcija se tada zatvara.

HTTP poruke se dijele na zahtjeve i odgovore stoga nema jedinstvene strukture poruke. Generalizirana struktura poruke HTTP protokola (slika 3.2) dana je sa:

- *Request-Line*: identificira tip poruke kao zahtjev.
- *Status-Line*: pruža informacije o statusu odgovora.
- *General-Header*: polje koje je identično za zaglavlje zahtjeva i odgovora.
- *Request-Header*: sadrži informacije o zahtjevu i klijentu.
- *Response-Header*: sadrži informacije o odgovoru.
- *Entity-Header*: informacije o zahtjevu i Entity-Body.
- *Entity-Body*: traženi podaci. [8]

Kao što je već rečeno, HTTP ima velik broj metoda koje su uključene u zaglavlje ovog protokola. HTTP protokol zahvaljujući nekim od tih metoda može biti korišten kao medij da se izvrše određene promjene na serveru koji upravlja mrežom računala, poput:

- *POST*: klijent od servera zahtjeva da se poruka koja je poslana spremi na server (unos u bazu podataka, pohranjivanje neke datoteke).




Slika 3.2: Generalizirana struktura HTTP poruke.

- *PUT*: klijent zahtjeva od servera da poslanu poruku pohrani pod određenim URI-jem (Uniform Resource Identifier). Ako neki resurs sa danim URI-jem već postoji, onda dolazi do izmjene postojećeg, a ako ne postoji onda se stvara novi.
- *DELETE*: zahtjev za brisanje određenog resursa sa servera.

Pošto HTTP koristi TCP konekciju, poruke poslane tim protokolom su jednako "ranjive" na cyber napade kao i sam TCP. HTTP protokol koristimo svakodnevno dok koristimo web preglednike za pregledavanje Interneta. To uključuje pregledavanje e-maila ili korištenje internet bankarstva. Koristeći servise na Internetu koji zahtjevaju neke privatne podatke od korisnika (na primjer lozinku), povjerljivi podaci putuju mrežom sa HTTP protokolom. Ranjivost TCP protokola na "otimanje veze" znači da se privatni i tajni podaci o korisniku mogu pročitati. Za spriječavanje takvih napada koristi se HTTPS protokol koji podatke prije slanja enkriptira, pa zlonamjerni korisnik koji "prisluškuje" vezu ne može pročitai podatke koje je prikupio.

## 3.4 DHCP - Dynamic Host Configuration Protocol

*Dynamic Host Configuration Protocol* (DHCP) nije protokol koji se nalazi u TCP/IP stogu. Uloga DHCP-a je da dinamički dodjeljuje mrežne parametre (poput IP adrese) mrežnim čvorovima i servisima. Pmoću DHCP-a računala samo šalju zahtjev DHCP-serveru za dodjelu IP adrese i mrženih parametara i time se administratora ili korisnika oslobađa da te paramtere ne upisuju ručno. Moderne mreže (od malih kućnih do velikih mreža na

kampusima i lokalnih internet poslužitelja) koriste DHCP od 2011. kao standarni protokol. [7]

Kada se neko računalo spoji na mrežu, DHCP klijent šalje zahtjev za potrebnim mrežnim parametrima. DHCP server ima određeni raspon adresa koje može dodjeliti te šalje dodatne mrežne parametre (maska, name server, time server i sl.). Na velikim mrežama jedan DHCP server može posluživati više podmreža uz pomoć DHCP agenata koji se nalaze na usmjernicima te komuniciraju sa serverom i klijentima. Klijent može zatražiti od servera da mu uvijek dodjeljuje iste paramtre ali ne znači da će ih uvijek i dobiti nazad.

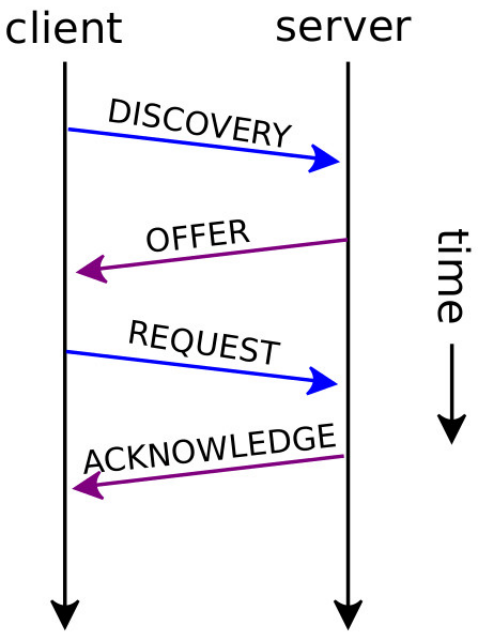
S obzirom na konfiguraciju, DHCP server može na 3 načina dodjeljivati IP adrese:

- *Dinamička alokacija*: prilikom pokretanja mrežnog servisa, računalo klijent je podešeno da zatraži podatke od DHCP servera. DHCP server odgovara sa nekom slobodnom adresom iz zadanog raspona.
- *Automatska alokacija*: slično kao dinamička alokacija, ali DHCP server čuva tablicu IP adresa kojom je prije posluživao klijente. Pomoću te tablice klijentu opet dodjeljuje istu adresu kao i ranije.
- *Statička alokacija*: dodjeljuje IP adresu na osnovu predkonfiguriranog mapiranja IP-MAC adresa.

DHCP protokol se satoji od 4 osnovnih i 2 opcionalne poruke koje klijent i server izmjenjuju: otkriće (*eng.* discovery), ponuda (*eng.* offer), zahtjev (*eng.* request), potvrda (*eng.* acknowledgement), informacije (*eng.* information) i oslobađanje (*eng.* releasing). 4 osnovne poruke još su poznate i kao DORA (discovery, offer, request i acknowledgement). Dio tijeka razmjene je prikazan na slici 3.3. Poruke se izmjenju pomoću UDP protokola na ulazu 67 za server i 68 za klijenta.

Klijent šalje poruku (DHCPDISCOVER) cijeloj mreži na adresu 255.255.255.255 ili na unaprijed konfiguriranu adresu. Klijent u ovom koraku može odmah zahtjevati neku prije dodjeljnu IP adresu. Odbijanje ili prihvaćanje takvog zahtjeva ovisi o postavkama servera. Autoritativni server takav zahtjev prihvaća ako klijent nije promjenio mrežu na kojoj se nalazi, u protivnom ju odbija. Neautoritativni server takav zahtjev ignirira i dolazi do zastarjevanja zahtjeva pa je klijent primoran poslati novi zahtjev. Kada DHCP server zaprimi DHCPDISCOVER poruku, zahtjev za IP adresom, on rezervira jednu IP adresu za klijenta i šalje poruku DHCPOFFER koja sadrži: klijentovu MAC adresu, IP adresu koju server nudi, oznaku maske, vrijeme valjanosti IP adrese i IP adresu DHCP servera koji je dao ponudu.
Nakon primitka povratne poruke klijent šalje novu poruku DHCPREQUEST kojom zatražuje da mu se dodjeli ponuđena adresa. Klijent može primiti DHCPOFFER poruke od više

DHCP servera odjenom ali adresu smije zatražiti samo od jednog servera. Ako server ne dobije DHCPREQUEST poruku on ponuđenu adresu vraća u skup raspoloživih adresa. Nakon primanja DHCPREQUEST poruke, server šalje DHCPPACK poruku koja sadrži vrijeme valjanosti IP adrese i druge konfiguracijske parametre. Time je faza potvrde završila. Kada klijent primi poruku sa konfiguracijskim paketom te izvrši konfiguraciju mrežnih postavki, on šalje ARP poruku cijeloj mreži kako bi svi mogli osvježiti svoje ARP tablice.




Slika 3.3: Tok razmjene osnovnih poruka između klijenta i DHCP servera.

Klijent može zatražiti više informacija nego što ih dobije putem DHCPOFFER i DHCPPACK poruka. Na primjer mogu se zatražiti postavke za različite aplikacije i servise (web preglednici koriste DHCPINFORM kako bi saznali postavke za proxy server). Kada je klijent gotov sa korištenjem IP adrese koju je zaprimio od DHCP servera, on šalje DHCPRELEASE poruku i time vraća IP adresu u skup slobodnih IP adresa.

DHCP ne pruža nikakvu vrstu autentifikacije prilikom komunikacije i to ga čini podložnim za napade kao što su *DHCP-spoofing* i man-in-the-middle. Takvi napadi spadaju u tri najčešće kategorije:

- neautorizirani DHCP server koji pruža informacije klijentima,

- neautorizirani klijent ima pristup podacima na mreži,
- iscrpljivanje resursa DHCP servera od strane zlonamjernih klijenata.

Neatorizirani DHCP server (*odmetnuti, lažni ili engleski "rouge" DHCP*) može se postaviti u mrežu i slati netočne informacije o konfiguraciji mreže. Takvi serveri mogu prouzročiti man-in-the-middle napade ili mogu klijentima uskratiti pristup mreži (*eng.* denial-of-service attack). S druge strane, pošto DHCP server ne može autentificirati klijenta, neki klijent može zatražiti podatke od servera i time steći pristup informacijama na mreži. Zbog istog razloga, zlonamjerni klijent može DHCP serveru konstantno slati nove zahtjeve s tim da svaki put promjeni podatke koji ga predstavljaju i time može iscrpiti sve IP adrese koje server ima mogućnost dodijeliti.
Gore opisane situacije se izbjegavaju korištenjem DHCP agenata koji tada vrše kontrolu zahtjeva prema DHCP serveru.

# Poglavlje 4

# Projektni zadatak: PMF - Matematički odsjek Nadzor računalne mreže

Projektni zadatak uz ovaj rad je implementirati sustav za nadzor mreže na Matematičkom odsjeku Prirodoslovno-Matematičkog fakulteta.
Sustav se satoji od skripti koji prikupljaju podatke sa sklopki - MAC adrese računala koja su spojena na mrežu, IP adresu pridružuje pronađenim MAC adresama, te ulaz na sklopki na koju su računala spojena. Ti podaci se spremaju u bazu podataka u kojoj se nalaze i podaci o sobama (uredi, kabineti i predavaone), osoblju koji se nalazi u tim sobama, oznakama i vezama utičnica na koje su računala u sobama spojena. Podaci iz baze se prikazuju u izrađenom web sučelju koje je intuitivno za korištenje.

Skripte, baza i web sučelje se nalaze na virtualnom Ubuntu 14.04 serveru sa Apache 2 servisom i MySQL bazom podataka.

## 4.1 O mreži na PMF - MO

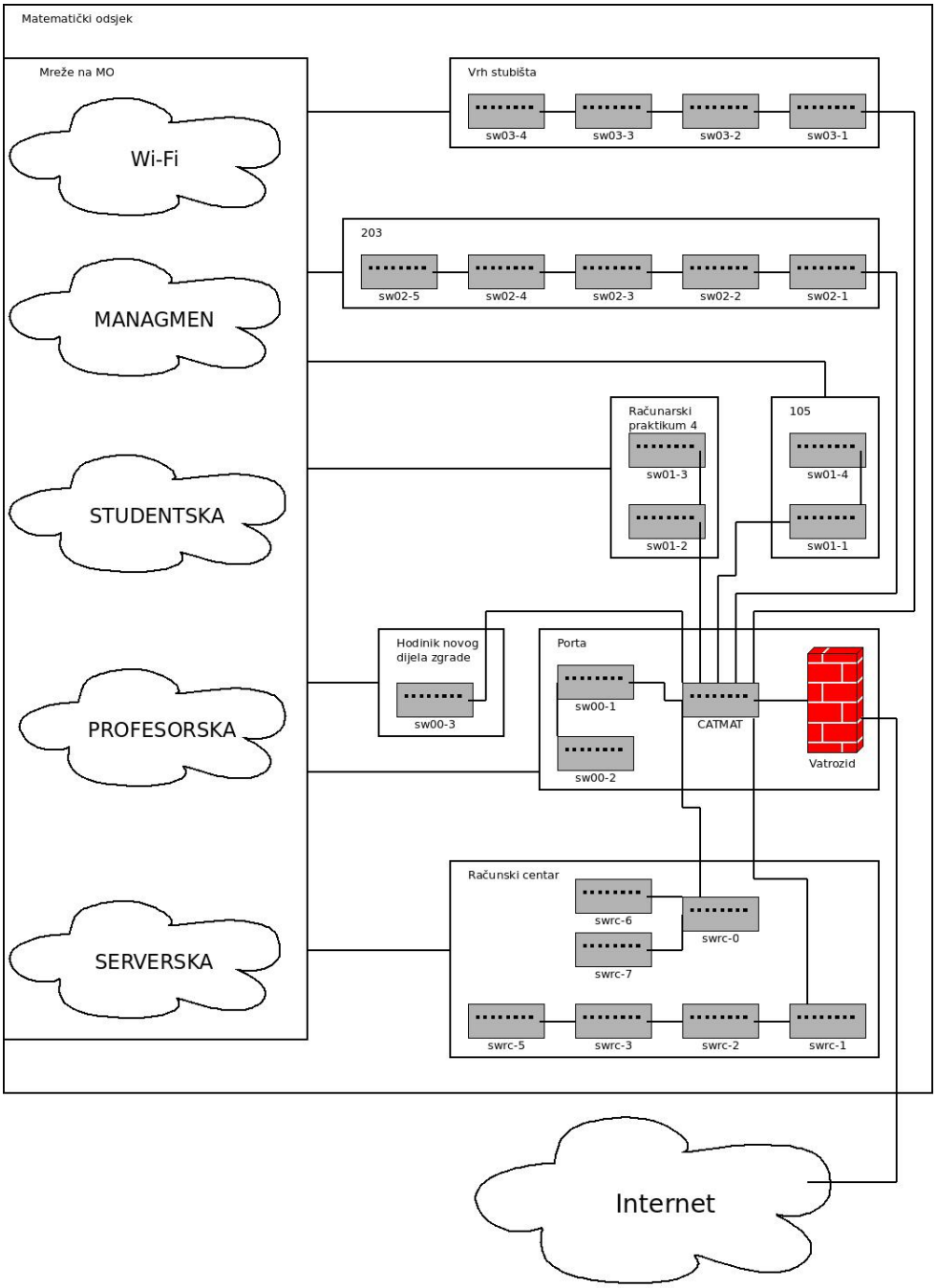
Mreža na PMF - MO sastoji se od više različitih mreža povezanih sklopkama. Mreže na PMF - MO su:

- *SERVERSKA mreža* sa IP adresama 161.53.8.0/24. Ona usmjerava podatke prema vatrozidu i Internetu. Sve mreže se usmjeravaju na ovu mrežu.
- *PROFESORSKA mreža* sa IP adresama 192.168.104.0/23. Računala u kabinetima nastavnog osoblja i predavaonama su spojena na tu mrežu.
- *STUDENTSKA mreža* sa IP adresama 192.168.88.0/23. Na ovu mrežu su spojena računala u praktikumima.

- *MANAGMENT mreža* sa IP adresama 192.168.168.0/24. Mreža se koristi za potrebe kadrovske službe i nenastavnog osoblja.
- *Wi-Fi mreža* sa IP adresama 192.168.68.0/23. Mreža za bežični pristup na PMF-MO.

IP adrese mrežnim uređajima dodjeljuje DHCP server, koji je pridružen nekoj od gore navedenih mreža, prilikom povezivanja uređaja na mrežu (na primjer paljenje računala ili povezivanje prijenosnog računala na bežičnu mrežu).

Topologija mreže je zvjezdasta. Svaka gore navedena mreža je zvjezdaste topologije i dostupne su sa svih sklopki u mreži. Sklopke se nalaze na svakom katu (podrum, prizemlje i 3 kata) odakle se granaju prema sobama, a spojene su na centralnu sklopku CATMAT (slika 4.1). Zbog fizičkog razmještaja sklopki, svaki ulaz sklopke je spojen na utičnicu u podu neke sobe. To utječe na geografiju mreže i čini ju jako kompliciranom. Mreža ima ukupno 23 sklopke dodjeljene na svoje podmreže od čega 3 imaju 48 ulaza a ostale po 24.




Slika 4.1: Prikaz mreže na PMF - MO.

## 4.2 Aplikacija za nadzor i prikupljanje podataka

Nadzor velike mreže poput mreže koja je implementirana na PMF - MO je težak zadatak te se mogu javiti razni kvarovi na mreži. Najčešči kvarovi su zagušenje neke od mreža ili prestanak rada nekog dijela mrežne opreme (na primjer neke sklopke). Dajmo dva primjera nekih standardnih kvarova na mreži postavljenoj na PMF - MO.

**Primjer 4.2.1.** *U svim praktikuma se istodobno piše kolokvij iz kolegija Matematički softver koji je u obliku "open book", što znači da se studenti mogu koristiti svim raspoloživim materijalima uključujući i Internetom. To je 68 korisnika računala u praktikumima koji istodobno koriste razne resurse sa Interneta. Taj broj računala nebi smio zagušiti STUDENTSKU mrežu ali ipak dođe do zagušenja, što ometa tijek ispita. Dežurni nastavnik prijavi sistem administratoru da je došlo do zagušenja mreže. Sistem administrator tada preko aplikacije može vidjeti promet po sklopkama, i njenim fizičkim ulazima, na grafovima koje aplikacija prikazuje. Kada uoči da određena sklopka na određenom ulazu ima povećani promet može pomoću jednostavne navigacije odrediti koje računalo stvara povećani promet.*

Aplikacija se može koristiti svakodnevno da se pregledaju grafovi kako bi se uočile anomalije na grafovima koje mogu ukazivati na neki kvar mreže. Na primjer virus na nekom računalu stvara povećani promet ili je graf konstantan sa vrijednosti 0 što znači da je nastao kvar na sklopki. Pravovremeno uočavanje anomalija u radu mreže, od strane sistem administratora, osigurava krajnjem korisniku kvalitetnu i sigurnu uslugu.

**Primjer 4.2.2.** *Sistem administratori na PMF - MO koriste više sustava za nadzor. Neki od tih sustava imaju mogućnost da sistem administratora obavijeste o netipičnom ponašanju nekog servisa na korisničkim računalima. Primjerice, sistem administrator dobije e-mail obavijest da korisnik sa danom IP adresom ima puno istodobnih HTTP upita (prkeo nekoliko tisuća zahtjeva u vrlo kratkom vremenskom roku). Pretragom izrađene aplikacije po IP adresama sistem administrator može vidjeti kojoj osobi (to jest kojem računalu i tko je korisnik računala) je ta IP adresa dodijeljena. Tada sistem administrator može obavijestiti korisnika da postoji velika mogućnost da mu računalo ne radi ispravno te da je nužno izvršiti pregled. Takva vrsta problema se nebi vidjela na grafovima jer HTTP upiti nebi zagušili mrežu*[1].

Aplikacija na intuitivan način vodi korisnika do krajnjeg rezultata. Web sučelje je jednostavno za korištenje i svi podaci su lako razumljivi. Baza podataka pohranjuje sve potrebne podatke za uspješan rad aplikacije. Aplikacija nudi dva načina za dolazak do

[1] Mreža na PMF - MO ima veliku propustnost, tj. veliki broj paketa može istovremeno putovati mrežom. Manje, primjerice kućne mreže, imaju puno manju propustnost pa bi tolika količina HTTP upita zagušila mrežu.

krajnjeg rezultata, pretraživanje po sklopkama i ulazima ili po IP adresama.

Pretraživanje po sklopkama i ulazima služi da se može provjeriti cijelo putovanje paketa, od sklopke do korisnika. Potreba za takvom pretragom nastaje uglavnom kada dode do zagušenja mreže. Aplikacija implementira grafove o zagušenju sklopki i ulaza na sklopkama koji su preuzeti s druge aplikacije za nadzor mreže. Uvođenjem podataka (grafova) s drugih aplikacija za nadzor u ovu, daje bolji pregled mreže, pruža uvid u kojem smjeru krenuti za pronalazak problem i smanjuje vrijeme da se problem detektira u što ranijoj fazi.

Pretraga preko IP adresa koristi se kada korisnik ili servis prijave neki kvar koji u svom opisu sadrži IP adresu. Tada se preko IP adrese mogu saznati ime sklopke i ulaz na koji su korisnik ili servis spojeni pa se može suziti broj mogućih problema na mreži.

Baza podataka pohranjuje podatke o osobama, sobama, sklopkama, ulazima na sklopkama i mapiranju IP i MAC adresa. Podaci o osobama i sobama su preuzeti sa službenih Web stranica fakulteta. Podaci o odnosu ulaza na sklopkama, oznakama utičnica i njihovim lokacijama su preuzeti iz dokumenata koji su u posjedu Računskog centra. Ostali podaci se prikupljaju više puta dnevno pomoću skripti koje se automatski izvršavaju nekoliko puta dnevno.

## 4.3 Baza podataka za pohranu prikupljenih podataka

Baza podataka je napravljena pomoću MySQL poslužitelja baze podataka (Data Base Management System - DBMS). Za komunikaciju s korisnikom MySQL koristi svoju inačicu jezika SQL (koju bi mogli nazvati MySQL-ov SQL).

Baza podataka je skup međusobno povezanih podataka. Podaci su istovremeno dostupni raznim korisnicima i aplikacijskim programima. Ubacivanje, promjena, brisanje i čitanje podataka obavlja se posredstvom zajedničkog softvera. Korisnici i aplikacije pritom ne moraju poznavati detalje fizičkog prikaza podataka, već se referenciraju na logičku strukturu baze.

MySQL spada u grupu jezika koji koriste relacijski model baza podataka. Relacijski model baze podataka zasnovan je na matematičkom pojmu relacije. Podaci i veze među njima su prikazane tablicama u ovom modelu i nazivamo ih relacija. Entitet je opisan atributima, to jest svojim tabličnim elementima. Ime entiteta, zajedno sa pripadnim atributima određuje tip entiteta. Može postojati mnogo primjeraka entiteta zadanog tipa (na primjer STUDENT je tip čiji primjerci su Petrović Petar, Marković Marko,...). Kandidat za ključ je

atribut, ili skup atributa, čije vrijednosti jednoznačno određuju primjerak entiteta zadanog tipa. Između dva, ili više, tipa entiteta uspostavlja se veza.

Svaka relacija ima svoje ime po kojem je razlikujemo od ostalih u istoj bazi. Jedan stupac relacije obično sadrži vrijednost jednog atributa (za entitet ili vezu), zato stupac poistovjećujemo s atributom i obratno. Atribut ima svoje ime po kojem ga razlikujemo od ostalih u istoj relaciji. Vrijednosti jednog atributa su podaci istog tipa.

Izrada baze podataka započinje modeliranjem entiteta i veza (Entity-Relationship Modelling), to jest izradom *ER-sheme* baze podataka. Riječ je o oblikovanju jedne manje precizne, konceptualne sheme, koja predstavlja apstrakciju realnog svijeta. Sljedeći korak je ER-shemu pretvoriti u *relacijska shema* (dakle u tablice). Relacijska shema može sadržavati nedorečenosti koje treba otkloniti prije implementacije. Proces daljnjeg dotjerivanja sheme zove se normalizacija. Teorija normalizacije zasnovana je na pojmu normalnih formi (NF) kojih ima 6: 1NF, 2NF, 3NF, Boyce-Codd-ova normalna forma (BCNF), 4NF i 5NF. Relacijska shema koja se dobije normalizacijom se implementira kao baza podataka.

MySQL, kako i svi jezici iz klase SQL (Structured Query Language), nudi mogućnosti za unos, izmjenu i brisanje podataka iz tablica. [4]

Radi lakašeg snalaženja u nastavku rada, baza podataka koja je izrađena za potrebe aplikacije zvat će se APLIKACIJA. Taj naziv nije isti kao i implementirane baze koja se na serveru naziva *snmp*.

APLIKACIJA je baza u kojoj su pohranjeni podaci o osobama (ime, prezime, titula, zvanje), sobama (broj i tip sobe), sklopkama (ime sklopke, broj ulaza na sklopci,...), MAC i IP adresama (njihove vrijednosti i vrijeme unošenja u bazu), ulazima na sklopke (na kojoj sklopci se nalazi ulaz, oznaka ulaza, s kojom je utičnicom u podu povezana) i utičnicama koje se nalaze u zgradi fakulteta. Podaci o IP i MAC adresama koje su spojene na mrežu osvježavaju se više puta dnevno (unos novih podataka i brisanje starih podatka). Podaci se osvježavaju pomoću bash skripti. Pristup bazi se ostvaruje preko web sučelja.
Baza podataka APLIKACIJA ima 7 tablica i zadovoljava 3NF. 4NF bi bila zadovoljena kada bi entiteti PORT i UTIČNICA bili jedan entit te bi taba bila zadovoljena i BCNF nad tim tablicama. To je namjerno učinjeno kako bi se u obzir uzeo ljudski faktor te olakšalo pronalaženje grešaka koje on uzrokuje. Podaci koji su sadržani u tablici UTIČNICA su fiksni te ih se ne može fizički promijeniti, dok podaci tablice PORT ovise o tome gdje je utičnica neke oznake spojena na sklopku.

## Popis entitea i veza u bazi podataka APLIKACIJA

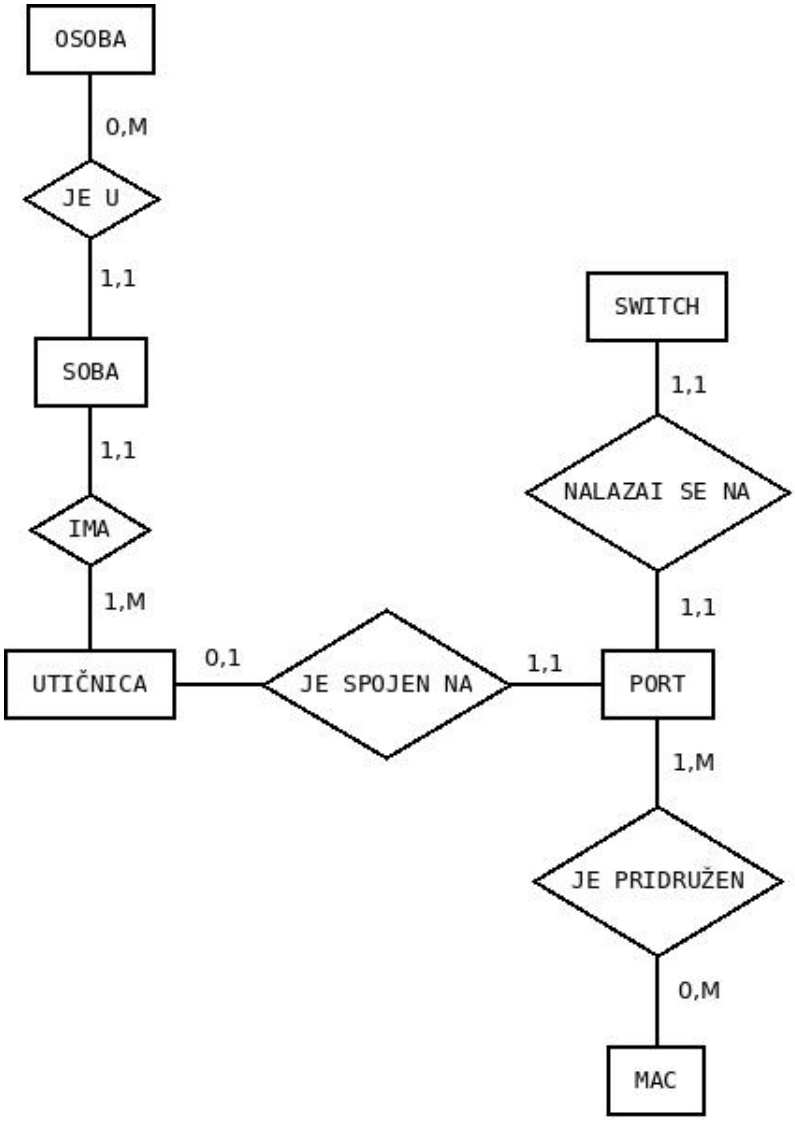
Slijedi popis entiteta s pripadnim atributima za bazu koja je korištena u aplikaciji.

Tip entiteta OSOBA s atributima RB_O (redni broj unosa u bazu, jedinstven unos), IME, PREZIME, TITULA, ZVANJE, SOBA_ID. Opisuje osbu, tj. zaposlenika fakulteta.
Tip entiteta SOBA s atributima SOBA_ID (alfanumerička oznaka sobe, jedinstven unos), KAT, TIP.
Tip entiteta UTIČNICA s atributima OZNAKA_U (jedinstvena alfa-numerička oznaka koja je zapisana iznad svake utičnice u zgradi), SOBA_ID.
Tip entiteta PORT s atributima BROJ_PORT (oznaka ulaza na sklopki), SW_IME (jedinstveno ime za svaku sklopku), BRZINA, OZNAKA_U.
Tip entiteta SWITCH s atributima SW_IME (jedinstveno ime za svaku sklopku), LOKACIJA (označava kat gdje se sklopka nalazi), BROJ_PORTOVA (koliko ulaza sklopka ima), IP, URL1, URL2, URL3, URL4. URL adrese služe da se neke konfiguracije sklopke mogu obaviti preko HTTP protokola.
Tip entiteta MAC s atributima RB_MAC (jedinstvena identifikacijska oznaka unosa), MAC_ADR, IP, DATUM_UPISA.

Dalje slijedi popis veza.

JE_U, 1:M veza između entiteta OSOBA i SOBA gdje OSOBA ima obavezno članstvo.
IMA, 1:M veza između entiteta SOBA i UTIČNICA. SOBA ima obavezno članstvo.
JE_SPOJEN_NA, 1:1 veza između entiteta UTIČNICA i PORT.
NALAZI_SE_NA, 1:1 veza između entiteta PORT i SWITCH.
JE_PRIDRUŽEN, 1:M veza između entiteta PORT i MAC. MAC ima obavezno članstvo u PORT.
Zbog praktičnih razloga ova relacija je prilikom implementacije baze nazvana MAC2PORT.

**Reducirani Chenov dijagram baze podataka APLIKACIJA**



## Relacijska shema baze podataka APLIKACIJA

Slijedi relacijska shema baze podataka APLIKACIJA.

OSOBA(RB_O, IME, PREZIME, TITULA, ZVANJE, SOBA_ID)
SOBA(SOBA_ID, KAT, TIP)
UTICNICA(OZNAKA_U, SOBA_ID)
PORT(BROJ_PORT, SW_IME, BRZINA, OZNAKA_U)
SWITCH(SW_IME, LOKACIJA, BROJ_PORTOVA, IP, URL1, URL2, URL3, URL4)
MAC(RB_MAC, MAC_ADR, IP, DATUM_UPISA)
MAC2PORT(RB_MAC2PORT, RB_MAC, BROJ_PORT, SW_IME, DATUM_UPISA)

## 4.4 Skripte za prikupljanje podataka sa sklopki

Skripte za prikupljanje podataka sa sklopki su pisane u *Bash* okolini. Bash je Unix ljuska i komandni programski jezik, tj. moguće je napisati skup nardbi u datoteku koje se izvršavaju na korisnikov zahtjev. Takvu datoteku nazivamo *skripta* i njena ekstenzija je .sh. U najopećnitijem smislu ljuska je program u koji korisnici upisuju naredbe koje operativni sustav zna izvršiti. Najpoznatije ljuske su Bourne shell (sh) i C shell (csh) te njihovi nasljednici bash i tcsh. Mogućnost izvršavanja skripti, ljuskama daje mogućnost za upravljanje konfiguracijom sustava (najpoznatiji primjer su *log-in* skripte koje postavljaju radnu okolinu kada se korisnik spoji na sustav). [1]

Postoje 4 skripte od čega dvije prikupljaju podatke (slika 4.2), jedna priprema i unosi podatke u bazu te jedna pomoćna skripta koja pokreće prethodno spomenute skripte. Koristi se i jedana pomoćna tekstualna datoteka koja u sebi sadrži samo imena sklopki s kojih se podaci prikupljaju (u slučaju projektnog zadatka to su sve sklopke koje se nalaze unutar zgrade Matematičkog odsjeka).

Prvo se pokreće pomoćna skripta *ports.sh*. Ona kontrolira tok pokretanja ostalih skripti, dok je njezino pokretanje kotrolirano pomoću *cronjob* servisa (sustav za automatsko pokretanje u zadano vrijeme) na serveru i pokreće se svakih 6 sati (počevši od ponoći).

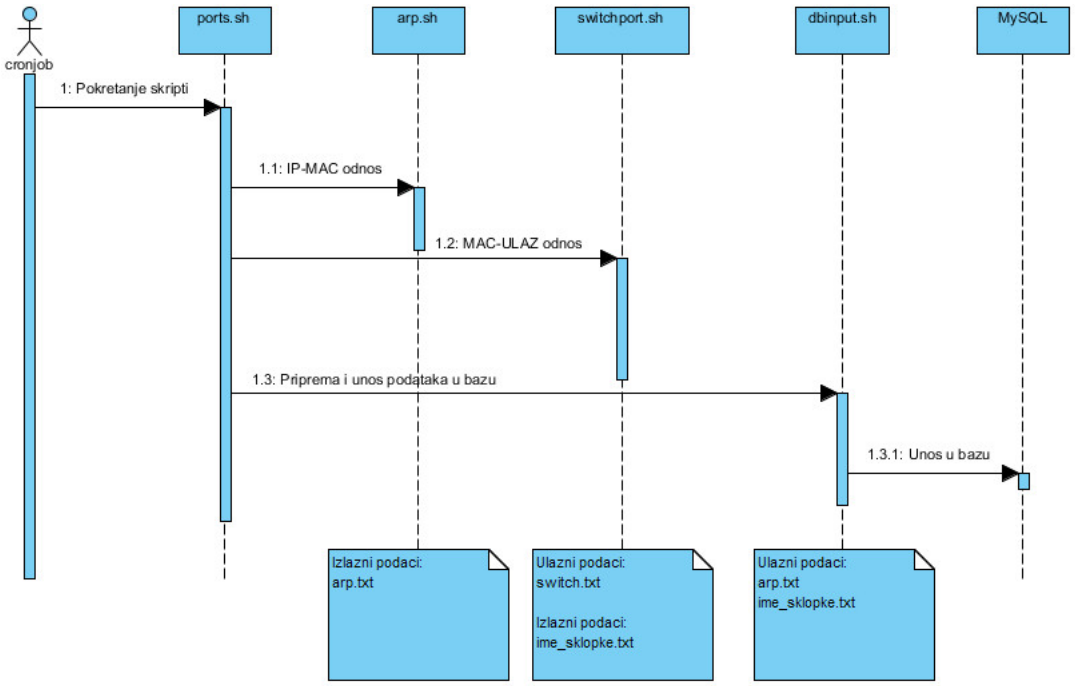
*arp.sh* skripta prikuplja podatke o pridruživanju IP adrese svakoj od MAC adresa koje su prisutne na mreži. Prikupljeni podaci se spremaju u pomoćnu datoteku *arp.txt*

*switchport.sh* prikuplja podatke o tome koja MAC adresa se nalazi na kojem ulazu za svaku od sklopki čije ime se nalazi u pomoćnoj datoteci *switches.txt* te se rezultat sprema u datoteke s nazivom *ime_sklopke.txt*.

*dbinput.sh* je posljednja skripta koja se pokreće iz *ports.sh*. Ona je zadužena da podatke, koje su predthodne dvije skripte prikupile, ubaci u bazu i ona briše sve podatke iz baze koji su stariji od 30 dana.

Skripte *arp.sh* i *switchport.sh* za prikupljanje podataka koriste nardbu *snmpwalk*. Ta nareba šalje zahtjev za svim SNMP varijablama mrežnih čvorova i njihove vrijednosti GetNextRequest metodom (strana 25) iz stabla varijabli. Zatim se pomoću MIB oznake pretražuju vrijednosti te se potrebne vrijednosti spremaju u za to namjenjene dokumente (arp.txt i ime_sklopke.txt).

Većina skripti koristi pomoćne datoteke kako bi se potrebni podaci mogli sortirati ili




Slika 4.2: Tok pokretanja skripti pomoću cronjob servisa.

privremeno pohraniti da se olakša izvršavanje naredbi koje se kasnije pozivaju. Te privremene datoteke se brišu na samom kraju izvršavanja skripte.

### ports.sh

```
1  #!/bin/bash
2
3  cd /home/mstojano/diplomski/scripts
4
5  cd Arp
6
7  ../arp.sh > arp.txt
8
9  for i in ‘cat ../switches.txt‘; do
10    echo $i;
11    ../switchport.sh $i > $i.txt;
12 done
13
14 cd ..
15 ./dbinput.sh
```

### arp.sh

```
1  #!/bin/bash
2
3  snmp_file=‘tempfile‘
4  ip_file=‘tempfile‘
5  mac_file=‘tempfile‘
6
7  snmpwalk -Os -c sismish -v 1 catmat mib-2.3.1.1.2 >
       $snmp_file
8
9  cat $snmp_file | grep -E "^mib-2.3.1.1.2" | cut -d : -f 2- |
       cut -d " " -f 2-7 | sed 's/ /:/g' > $mac_file
10
11 cat $snmp_file | grep -E "^mib-2.3.1.1.2" | cut -d . -f 8- |
       cut -d " " -f 1 > $ip_file
12
13 paste $ip_file $mac_file
14
15 rm $snmp_file $ip_file $mac_file
```

### switchport.sh

```
1  #!/bin/bash
2
3  snmp_file=‘tempfile‘
4  port_file1=‘tempfile‘
5  port_file2=‘tempfile‘
6  port_file3=‘tempfile‘
7  mac_file=‘tempfile‘
8
9  snmpwalk -Os -c sismish -v 1 $1 mib-2.17.4.3.1 > $snmp_file
10
11 cat $snmp_file | grep -E "^mib-2.17.4.3.1.1." | cut -d . -f
       7- | cut -d " " -f 1 > $port_file1
12 cat $snmp_file | grep -E "^mib-2.17.4.3.1.1." | cut -d : -f
       2 | cut -d " " -f 2-7 | sed 's/ /:/g' > $mac_file
13
14 cat $snmp_file | grep -E "^mib-2.17.4.3.1.2." | cut -d . -f
       7- | cut -d " " -f 1 > $port_file2
15 cat $snmp_file | grep -E "^mib-2.17.4.3.1.2." | cut -d : -f
       2 | cut -d " " -f 2 > $port_file3
16
17 paste $mac_file $port_file3 | sort -n -k 2
18
19 rm $snmp_file $mac_file $port_file1 $port_file2 $port_file3
```

### dbinput.sh

```
1  #!/bin/bash
2
3  #nisu navedeni username i password koji se koriste u skripti
4  DB_un=username
5  DB_pass=password
6
7  cd Arp
8  touch ../input.txt
9
10 sed -i '/^\t/d' *.txt
11
12 for f in * ; do
13   if [ "$f" != "arp.txt" ] ; then
14     fn=‘basename $f ".txt"‘
15     cat $f | while read MAC port ; do
16       if [ -z $port ]; then
17         echo "((SELECT MAX(RB_MAC) FROM MAC WHERE MAC.
              MAC_ADR='"$MAC"'), NULL, '"$fn"', NOW()),"  >> ../
              input.txt
18       else
19         if [ $port -lt 25 -a $fn != "sw00-2" -a $fn != "sw03
              -4" -a $fn != "swrc-0" -a $fn != "swrc-6" ] ;
              then
20           echo "((SELECT MAX(RB_MAC) FROM MAC WHERE MAC.
                MAC_ADR='"$MAC"'), $port, '"$fn"', NOW()),"  >>
                ../input.txt
21         elif [ $port -lt 49 -a \( $fn == "sw00-2" -o $fn ==
              "sw03-4" -o $fn == "swrc-0" -o $fn == "swrc-6" \)
               ] ; then
22           echo "((SELECT MAX(RB_MAC) FROM MAC WHERE MAC.
                MAC_ADR='"$MAC"'), $port, '"$fn"', NOW()),"  >>
                ../input.txt
23         else
24           echo $port > /dev/null
25         fi
26       fi
27     done
```

```
28    else
29      echo "INSERT INTO MAC (MAC_ADR, IP, DATUM_UPISA) VALUES"
            >> ../input.txt
30      cat $f | while read IP MAC ; do
31        echo "('"$MAC"', '"$IP"', NOW())," >> ../input.txt
32      done
33      sed -i '$s/,$/;/' ../input.txt
34      echo "INSERT INTO MAC2PORT (RB_MAC, BROJ_PORT, SW_IME,
          DATUM_UPISA) VALUES" >> ../input.txt
35    fi
36  done
37
38  sed -i '$s/,$/;/' ../input.txt
39
40  echo "DELETE FROM MAC WHERE DATUM_UPISA < DATE_SUB(NOW(),
      INTERVAL 30 DAY);" >> ../input.txt
41  echo "DELETE FROM MAC2PORT WHERE DATUM_UPISA < DATE_SUB(NOW
      (), INTERVAL 30 DAY);" >> ../input.txt
42
43  mysql snmp <../input.txt -u $DB_un -p$DB_pass
44
45  rm ../input.txt
```

## 4.5 Web sučelje

Web sučelje je izrađeno pomoću PHP tehnologije koja komunicira sa bazom podataka te prikazuje podatke. Ima i jednostavni CSS kod koji oblikuje prikazane podatke. PHP (Hypertext Preprocessor, u početku je PHP značilo Personal Home Page) je skriptni programski jezik koji se izvršava na serverskoj strani u HTTP komunikaciji, ali može se koristiti i kao samostalan jezik za izvršavanje zadataka. PHP se često koristi zajedno sa HTML (HyperText Markup Language) jezikom. PHP kod se izvršava pomoću PHP interpretatora koji je instaliran na serveru. Nakon što se PHP kod na serveru izvrši, klijentu se šalje odgovor, koji je rezultat PHP skripte, u obliku HTML stranice, slike ili neke druge vrste podataka. PHP ima mogućnost komunikacije (slanja upita i primanja odgovora) sa MySQL bazama podataka.

HTML je standarni jezik koji se koristi za izradu web stranica. Web preglednici znaju čitati HTML kod, te iz njega načiniti prikaz koji je krajnjem korisniku razumljiviji. HTML daje strukturu i oznake (tagove) za prezentaciju krajnjem korisniku što ga razlikuje od programskih jezika. HTML može u sebi sadržavati skripte (poput PHP skripti) čiji se rezultat šalje krajnjem korisniku kao dio HTML koda.

Web preglednici znaju interpretirati i CSS (Cascading Style Sheets) kod. CSS definira izgled i raspored teksta i ostalih elemenata HTML koda. [6]

Svaka stranica izrađena za projektni zadatak na vrhu ima jednostavnu navigaciju koja omogućuje kretanje kroz funkcionalnost koje pruža web sučelje (Naslovna, Switch, IP, PMF - MO).
Četri su web stranice izrađene:

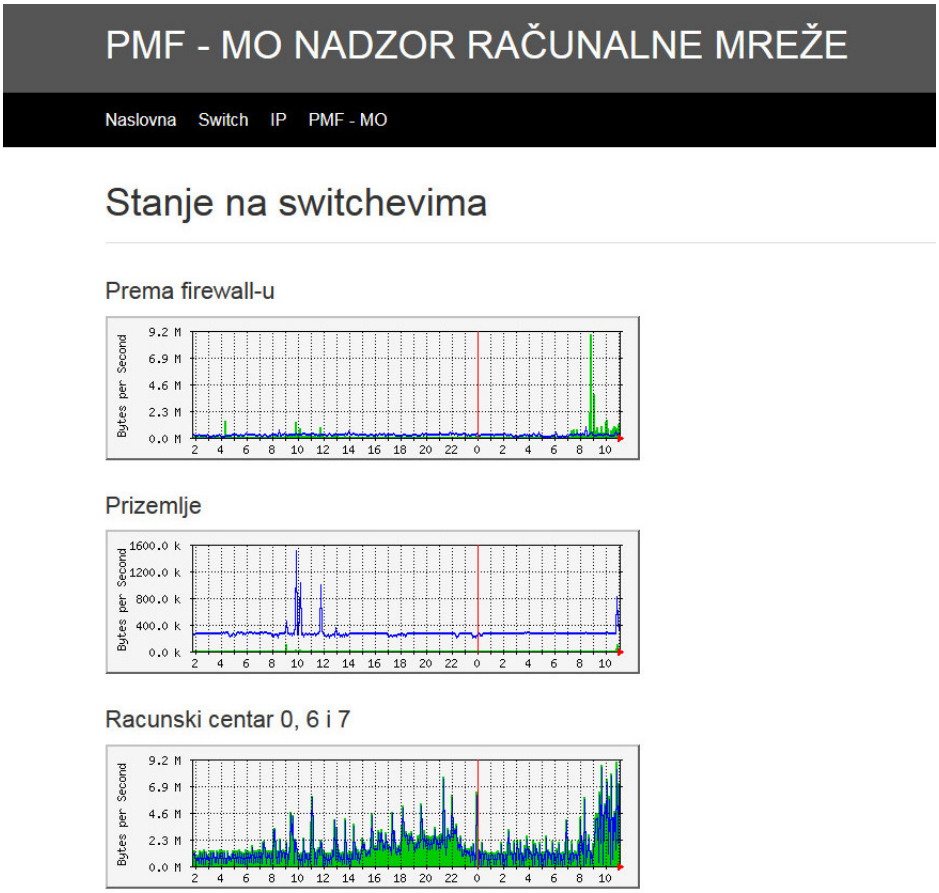
- *index.php* - naslova stranica web sučelja. Na njoj je kratki grafički pregled stanja na sklopkama. Grafikoni koji su tu prikazani su preuzeti sa drugog servera na kojem je pokrenuta druga aplikacija za nadzor stanja sklopki. Klik miša na grafikon neke sklopke vodi na web sučelje prije spomenute aplikacije za nadzor, gdje je prikazano više detalja o prometu na danoj sklopki. Slika 4.3.

- *switch.php* - dolaskom na ovu stranicu dinamički se ispisju sva imena sklopki koja se nalaze u bazi. Postavljanjem pokazivača miša preko imena otvara se izbornik u kojem su popisani svi ulazi koji se nalaze na sklopci (od 1 do 24 ili od 1 do 48). Klikom miša na neki broj ulaza na sklopki odlazimo na web stranicu *sve.php*. Slika 4.4.

- *ip.php* - prikazuje sve IP adrese koje su bile unešene u bazu podataka u zadnjih 24 sata. Pored svake adrese je zapisano na kojoj se podmreži ona nalazila (nastavnička,

studentska, Wi-Fi,...) te vrijeme kada je upisana u bazu. Klikom miša na neku IP adresu odlazimo na stranicu *sve.php*. Slika 4.5.

- *sve.php* - prikazuje podatke iz baze vezane za IP adresu ili za ulaz na nekoj sklopki. 4.6.




Slika 4.3: Naslovna stranica web sučelja, prikazuje grafove koji su preuzeti sa druge aplikacije za nadzor.

## Upiti na bazu podataka

Tri od četri izrađene web stranice imaju ostvarenu komunikaciju sa bazom podataka radi dohvaćanja podataka.

*switch.php* šalje bazi podataka dva upita. Prvi upit[2] je:

```
select SW_IME from SWITCH;
```

koji dohvaća imena svih sklopki čiji su podaci pohranjeni u bazi. Taj upit služi kako bi se mogla generirati lista imena sklopki (lijeva lista na slici 4.4) te kako bi se mogao izvršiti slijedeći upit po redu:

```
select BROJ_PORTOVA from SWITCH
where SW_IME="' . $row['SW_IME'] . '";
```

gdje varijabla $row['SW_IME'] iterativno prolazi po cijeloj tablici iz prvog upita. Ova naredba dohvaća broj ulaza na sklopki koji služi za generiranje popisa ulaza sklopke (desna lista na slici 4.4).

*ip.php* ima jedan upit prema bazi. Upit:

```
select IP, DATUM_UPISA from MAC
where date(DATUM_UPISA)=date(now()) order by IP;
```

dohvaća sve IP adrese koje su u bazu unešene na dan traženja upita i sortira ih uzlazno po vrijednosti IP adrese (leksikografski). Ova stranica sadrži i kratku PHP skriptu koja određuje kojoj mreži unutar PMF - MO neka IP adresa priprada. To je jednostavno parsiranje IP adrese kako bi se našla mreža. Slika 4.5.

Obje stranice sadže *linkove* (poveznice) koji upućuju na *sve.php*. Ti linkovi se generiraju dinamički uz pomoć gore navedenih upita. Oni se generiraju paralelno sa generiranjem HTML koda iz PHP skripti koje vrše upit. *sve.php* ima dva upita na bazu ovisno o tome s koje od predhodnih stranica smo došli. Pošto te stranice imaju različite podatke (IP adresa te ime sklopke i ulaz na sklopki), odluka koji od dva upita će se uputit bazi radi se simulacijom PHP GET metode (koja nije ista kao i HTTP GET metoda sa stranice 26). PHP GET metoda služi za prijenos podataka između HTML stranice i PHP skripte. PHP skripta provjerava da li je IP polje u GET metodi jednako nula, ako je (znači da smo došli preko *switch.php*) onda se izvršava upit:

```
select distinct MAC.IP, MAC.MAC_ADR, MAC2PORT.SW_IME,
MAC2PORT.BROJ_PORT, PORT.OZNAKA_U, UTICNICA.SOBA_ID,
OSOBA.PREZIME, OSOBA.IME
from MAC2PORT, MAC, PORT, UTICNICA, OSOBA
```

[2]Upiti su zapisani u SQL obliku, ali koriste PHP konkatenaciju stringova (navodnici i točka) i varijable ($ime_varijable) ako u upitu ima dinamički kreiranih elemnata.


Slika 4.4: Popis sklopki i njihovih ulaza, prikazani su ulazi sa prve sklopke na popisu (sw00-1).

```
where (MAC2PORT.RB_MAC=MAC.RB_MAC
and PORT.SW_IME="' . $_GET['SW_IME'] . '"
and PORT.BROJ_PORT=' . $_GET['BROJ_PORT'] . ')
and ((MAC2PORT.SW_IME=PORT.SW_IME)
and (MAC2PORT.BROJ_PORT=PORT.BROJ_PORT))
and (UTICNICA.OZNAKA_U=PORT.OZNAKA_U)
and (OSOBA.SOBA_ID=UTICNICA.SOBA_ID);.
```

Ako IP polje GET metode nije nula (došli smo preko *ip.php*) tada se izvršava upit:

```
select distinct MAC.IP, MAC.MAC_ADR, MAC2PORT.SW_IME,
MAC2PORT.BROJ_PORT, PORT.OZNAKA_U, UTICNICA.SOBA_ID,
OSOBA.PREZIME OSOBA.IME
from MAC2PORT, MAC, PORT, UTICNICA, OSOBA
where (MAC2PORT.RB_MAC=MAC.RB_MAC
and MAC.IP="' . $_GET['IP'] . '")
```


Slika 4.5: Popis IP adresa koje su bile aktivne na dan kada su skripte pokrenute.

```
and ((MAC2PORT.SW_IME=PORT.SW_IME)
and (MAC2PORT.BROJ_PORT=PORT.BROJ_PORT))
and (UTICNICA.OZNAKA_U=PORT.OZNAKA_U)
and (OSOBA.SOBA_ID=UTICNICA.SOBA_ID);.
```

Kao što se vidi iz upita *sve.php* daje istu tablicu kao krajnji rezultat bez obzira kako se do te stranice dođe. Slika 4.6 prikazuje podatke koji su dobiveni gore napisanim upitima. To su podaci o IP adresi, MAC adresi računala, imenu sklopke na kojoj je MAC adresa pronađena, broju ulaza na sklopki koji je povezan sa utičnicom, oznaci utičnice, sobe gdje se utičnica nalazi te osobama koje se nalaze u toj sobi.


Slika 4.6: Krajnji rezultat navigacije kroz web sučelje. Pristup preko sklopki ili IP adrese daje jednak rezultat. Do rezultata prikazanog na slici smo mogli doći tako što smo kliknuli na ime sklopke sw03-01 i njen ulaz broj 15 na web starnici *switch.php* ili klikom na IP adresu 192.168.104.107 na web stranici *ip.php*.

# Zaključak

U radu je dan pregled pojmova iz mreža računala, temeljnih protokola za ispravan rad velikih i raznorodnih mreža te nekih osnovnih protokola koji omogućuju nadzor i konfiguraciju mreže i mrežnih uređaja. Navedeni su način rada, prednosti i mane za svaki od tih protokola. Projektni zadatak izrađen u sklopu rada daje prikaz funkcionalnosti i korištenja nekih protokola koji su opisani.

U radu su opisani i sigurnosni nedostaci navedenih protokola te razni načini kako se rad i sigurnost mreže mogu kompromitirati. Daljnji razvoj protokola koji se koriste u mrežnoj komunikaciji usmjeren je na unapređivanje u izvršavanju njihovih zadaća, smanjenje utrošenih resursa te jačanje sigurnosnih mjera unutar samih protokola.

Aplikaciju koja je izrađena kao dio ovog rada koristit će u svom radu sistem administratori na PMF - MO.

# Bibliografija


[1] C. Albing, J.P. Vossen i C. Newham, *bash Cookbook*, O'Reilly Media, Inc., 2007.

[2] D.E. Comer, *Computer Networks and Internets with Internet Applications, Fifth Edition*, Pearson - Prentice Hall, 2009.

[3] L. Grubišić i R. Manger, *Mreže računala*, Interna skripta, PMF - Matematički odsjek, 2013.

[4] R. Manger, *Baze podataka*, Element, 2012.

[5] R.D. Mauro i K.J. Schmidt, *Essential SNMP*, O'Reilly Media, Inc., 2005.

[6] R. Nixon, *Learning PHP, MySQL, JavaScript, CSS and HTML5*, O'Reilly Media, Inc., 2014.

[7] L.L. Peterson i B.S. Davie, *Computer Networks: A Systems Approach, Fifth Edition*, Morgan Kaufmann - Elsevier, 2011.

[8] W. Stallings, *Data and computer communications, Eight edition*, Pearson Education, Inc., 2007.

# Sažetak

Nadzor računalnih mreža defnira se kao praćenje stanja mreže i spojenih uređaja te detekcija problema unutar mrežnog sustava. Konfguracija računalnih mreža podrazumijeva podešavanje parametara mrežnih uređaja i servisa u svrhu njihovog boljeg rada. U ovom radu dan je pregled protokola koji se koriste za nadzor i konfguraciju računalnih mreža (SNMP, DHCP, DNS, HTTP). Podaci prikupljeni putem protokola za nadzor pohranjuju se u bazu podataka. Za jednostavniji prikaz trenutnih i prošlih stanja mreže i interakciju s bazom podataka izrađeno je web sučelje. U radu su navedene i objašnjene sve tehnologije korištene za izradu baze podataka i rad s bazom, te tehnologije potrebne za izradu i ispravan rad web sučelja.

# Summary

Survelliance of a computer network is defined as network status supervision, status checking of all devices connected on the network and detection of all problems inside the network system. The configuration of a network system consists of settings paramters for hardware components and software services to optimize their performance. In this thesis an overview of protocols that are used for surveliance and configuration of a computer network (SNMP, DHCP, DNS, HTTP) are given. Data obtained using network surveliance protocols is stored in a database. For a simpler overiew of the network status in real and past times and for an easier database interaction a web interface is made. In this thesis all the techonlogies used for the data gathering, the making of the database and the web interface are explained.

# Životopis

Rođen 22. veljače 1987. godine u Kopru, Republika Slovenija. Osnovnu školu i gimnaziju (srednja škola "Vladimir Gortan") pohađao sam u Bujama, Istra. Tijekom osnovne škole pohađao sam učenička natjecanja iz matematike, fizike i kemije, dok sam u srednjoj školi odlazio na natjecanja iz matematike. 2005. godine, po završetku gimnazije, upisao sam prvu godinu studija na PMF - Matematički Odsjek u Zagrebu, a 2012. godine upisujem diplomski studij računarstva i matematike na PMF - MO.

Tijekom studija vršio sam sljedeće dužnosti: studentski predstavnik u vijeću Matematičkog odsjeka (2011.-2015.); studentski predstavnik u Vijeću voditelja studija Matematičkog odsjeka (2012.-2015.); demonstrator na kolegijima Računarski praktikum 1 i 2 (prof.) (2012.-2015.); demonstrator u računarskim praktikumima (2009.-2015.); demonstrator (audio-video tehničar) u multimedijskoj predavaoni (na postiplomskom studiju) (2012.-2015.); voditelj demonstratora u računarskim praktikumima (2013.-2015.); u Računskom centru PMF - MO zamjena djelatnika na bolovanju (svibanj - srpanj 2012.); suradnik (audio-vizualni tehničar) na međunarodnoj konfereciji o popularizaciji matematike *Diderot*, prosinac 2013. u Zagrebu, u suradnji s Sveučilištem u Exeteru i Sveučilištem u Berlinu; suradnik (audio-vizualni tehničar) na Kongresu nastavnika matematike održanom u Zagrebu 2014. godine.

Koautor sam stručnog članka iz matematike (M. Stojanović, D. Veljan, *Zgužvani papiri, čaša limunade, rukovanje, vjetar na Zemlji i Brouwerov teorem o fiksnoj točki*, Matematičko fizički list, godina LXIV, 1/253 (2013./14.), 36–40) i članka iz glazbe (M. Stojanović, M. Karaga, *Dig our rig*, Bass Player, Vol. 24. No.13., Holiday issue 2013.).

U slobodno vrijeme bavim se glazbom.
Oženjen.
Trenutno zaposlen u AVL-AST kao softverski inženjer za razvoj.