Osijek, 2023.

ELPROS
REGIONALNI CENTAR KOMPETENTNOSTI
ELEKTROTEHNIČKA I PROMETNA ŠKOLA OSIJEK

Silvio Papić

PROJEKTIRANJE, IZRADA I ODRŽAVANJE RAČUNALNIH MREŽA

Priručnik za polaznike

ESF UČINKOVITI LJUDSKI POTENCIJALI
EUROPSKI STRUKTURNI I INVESTICIJSKI FONDOVI

Sadržaj publikacije isključiva je odgovornost Elektrotehničke i prometne škole Osijek.

Projekt je sufinancirala Europska unija iz Europskog socijalnog fonda.

Za više informacija o EU fondovima molimo pogledajte web-stranicu Ministarstva regionalnoga razvoja i fondova Europske unije.

www.strukturnifondovi.hr

IMPRESUM

Autor: **Silvio Papić, dipl. ing. aeronaut.** (ALGEBRA d.o.o.)

Urednik i stručnjak za metodologiju izrade nastavnih materijala: **Karlo Josić, mag. oec.** (ALGEBRA d.o.o.)

Recenzent: **Damir Regvart, mag. ing. el.** (ALGEBRA d.o.o.)

Grafičko oblikovanje: **BARREK d.o.o.** (ALGEBRA d.o.o.)

Lektorica: **Ivana Previšić, mag. philol. croat. et hist.** (ALGEBRA d.o.o.)

Naslov: **Stručnjak za projektiranje, izradu i održavanje računalnih mreža** (priručnik za polaznike u programu obrazovanja „Stručnjak za računalne mreže“ za stjecanje djelomične kvalifikacije razine 5 HKO)

Mjesto i godina izdanja: **Osijek, 2023.**

Nakladnik i nositelj prava: **Elektrotehnička i prometna škola Osijek**

Odgovorna osoba: **ravnatelj Antun Kovačić, dipl. ing. el.**

Za više informacija:
Elektrotehnička i prometna škola Osijek
Istarska 3
31000 Osijek
E-mail: ured@ss-elektrotehnicka-prometna-os.skole.hr

Regionalni centar kompetentnosti ELPROS

Osijek, 2023.

STRUČNJAK ZA PROJEKTIRANJE, IZRADU I ODRŽAVANJE RAČUNALNIH MREŽA

Priručnik za polaznike

Algebra d.o.o.

Osijek, 2023.

Projekt je sufinancirala Europska unija iz Europskog socijalnog fonda.

Sadržaj:

1. Temeljne tehnologije računalnih mreža

U OVOJ CJELINI NAUČIT ĆETE:

- što je tehnologija
- što je IKT sustav
- što je računalna mreža i koje su karakteristike računalne mreže
- koji su slojevi OSI modela i koja je njihova uloga u komunikaciji
- koristiti naredbe u naredbenom retku računala
- izraditi adresnu shemu i sažimati podmreže.

Za početak važno je razumjeti što je tehnologija i kako se ona koristi u svakodnevnom čovjekovu životu i radu. U ovoj cjelini upoznaju se osnovni pojmovi i načini sagledavanja tehnologija koje se koriste u modernim informacijsko-komunikacijskim sustavima (IKT). Postoje mnoge tehnologije koje čine sustav IKT-a: od fizičke opreme, kabela i uređaja, preko operativnih sustava, mrežnih uređaja, vatrozida, pa do raznih aplikacija. Priručnik u središte stavlja mrežne tehnologije, koje čine dio infrastrukture u pozadini, a omogućuju korištenje različitih aplikacija i usluga, od kojih je možda najpoznatija − internet.

1.1. Osnovni pojmovi

1.1.1. Što je tehnologija

U kontekstu ovog priručnika pojmove tehnologija i protokol možemo koristiti kao istoznačnice, iako to nisu, međutim njihova razlika bitno ne utječe na razumijevanje što je to računalna mreža i kako ona funkcionira. To se može objasniti sveobuhvatnošću značenja pojma tehnologije. Tehnologija obuhvaća i uređaje, npr. vatrozid, usmjernik, preklopnik, operacijski sustav, virtualizacijsku platformu; bilo koju uslugu u okruženju računarstva u oblaku; ili neki protokol usmjeravanja. Dok s druge strane, protokoli su konkretna specifična primjena određenog koncepta na lokalnoj razini unutar sustava IKT-a. Tako se, na primjer, može govoriti o tehnologijama usmjeravanja u računalnim mrežama i navesti protokole, kao što su EIGRP, RIP ili OSPF. Može se reći da je tehnologija sve što čovjek izmisli kako bi podredio stvarnost sebi, dakle bilo kakva pomoć koju je osmislio u svrhu lakšeg rješavanja određenoga problema određene razine složenosti u određenom kontekstu na određeni način. Tehnologija je, dakle, i način obrade metala, i obrade obradivih površina, i izrada alata, i osmišljavanje procedura, informacijskih sustava, metoda i metodologije rada; to su i razni protokoli u medicini ili u znanstvenim istraživanjima; i obrada podataka, i procesi odlučivanja i upravljanja; i razni sustavi koji čine infrastrukturu, i još puno toga.

Primjeri raznih tehnologija su:

Čekić, vreteno, tkalački stan, tiskarski stroj, peć na električnu energiju, raketni motor, hidraulika, hidraulična presa, kombajn, traktor, plug za oranje, lemilica, indukcijska peć, dalekozor, teleskop, mehanički sat, atomski sat, procesor, memorija računala, transformator, raketni motor, dizelski motor, električni motor, proizvodnja kulena i pršuta, itd.

Sve navedene tehnologije nastaju na temelju našeg razumijevanja stvarnosti, od razine njezina promatranja, preko teorija kojima opisujemo tu stvarnost, mnoštva eksperimenata u svrhu potvrde naših teorija pa do prihvaćenih znanstvenih spoznaja.

Primjer kako od teorije nastaju tehnologije naše je razumijevanje prirode svjetlosti, optike, poluvodičkih i drugih materijala, tranzistora, lasera i proizvodnih procesa, što je

omogućilo osmišljavanje i proizvodnju optičkih kabela, koji se danas koriste u računalnim mrežama.

1.1.2. Što je sustav IKT-a

Na početku je iznimno važno shvatiti i zapamtiti da je jedina svrha sustava IKT-a u cjelini i njezinih pojedinih tehnologija podrška ljudskim aktivnostima. Ta podrška može biti vidljiva u omogućavanju i olakšavanju naših privatnih aktivnosti, poput pristupa internetu, njegova pretraživanja, gledanja televizije, igranja videoigara, razmjene poruka na društvenim medijima, rada od kuće, učenja, itd. Također i u podršci poslovanju malih i velikih organizacija, poput škola, bolnica, banaka, burzi, proizvodnih pogona i slično. Dakle, sustavi IKT-a nisu svrha sami sebi, već podržavaju ljudske aktivnosti. Da bi mogli ostvariti svrhu za koju su stvoreni, sustavi IKT-a moraju imati određene karakteristike koje nisu direktno vidljive u pojedinim njegovim komponentama, ali se pojavljuju kada se na smislen način te komponente povežu, npr. mrežna oprema, mrežne tehnologije, mehanizmi i protokoli, baze podataka, poslužitelji s različitim operacijskim sustavima (npr. *Microsoft*, *Linux*), različite vrste medija za prijenos podataka, tehnologije za virtualizaciju, sigurnosna pohrana podataka i slično. Među njima razlikujemo karakteristike „nižeg reda“, kao što su sigurnost, skalabilnost, visoka dostupnost, zalihost, otpornost sustava na otkaze i slično.

Sustavi IKT-a mogu biti mali i jednostavni, pa ih čak i ne zovemo tako, već samo „mreža“. Primjer takvih malih sustava IKT-a bili bi oni koje svi imaju u svojim domovima a sastoje se od jednog univerzalnog uređaja (bežični usmjernik, koji je ujedno i preklopnik, i vatrozid) koji daje telekomunikacijski operator na raspolaganje. Međutim, ti sustavi mogu biti vrlo složeni, poput podatkovnih centara velikih tvrtki (npr. *Amazon*, *Google*, *Microsoft*), čije sustave IKT-a koristi veliki broj drugih organizacija i pojedinaca. Oni omogućuju kompleksna okruženja, poput okruženja računarstva u oblaku (engl. *Cloud Computing*) koja u sebi sadrže sve potrebne elemente da jedna tvrtka uz vrlo malo fizičke opreme uspostavi vlastiti sustav IKT-a koji podržava njezino poslovanje. Od vremena kada je svaka tvrtka kupovala svoje poslužitelje, preklopnike (engl. *Switch*), usmjernike (eng. *Router*), vatrozide (engl. *Firewall*), dogovarala licence za operacijske sustave i razne aplikacije, do danas je to u velikoj mjeri automatizirano i ne traži se od korisnika veliko tehničko znanje ni velika početna investicija (engl. *Capital expenditures* – *CapEx*), već je korisnik usredotočen na svoje poslovanje, a

sustav IKT-a mu je dio operativnog troška (engl. *Operating expense – OpEx*). Ovisno o prirodi posla neke tvrtke, odnosno djelatnosti (engl. *Core Business*), veličina i razina kompleksnosti sustava IKT-a bit će različita, kao i potreba za stručnjacima za izgradnju i održavanje tog sustava. Tako nisu iste potrebe za tehnologijama ni stručnjacima IKT-a npr. u banci ili školi koja obrazuje buduće sistemske inženjere, ili, pak, tvrtki koja dizajnira cjelovita rješenja IKT-a. Bez obzira na složenost i razlike u veličini i funkcionalnostima koje obavljaju, svi sustavi IKT-a počivaju na istim principima, koji će dijelom biti obrađeni u ovom priručniku.

1.1.3. Što je računalna mreža

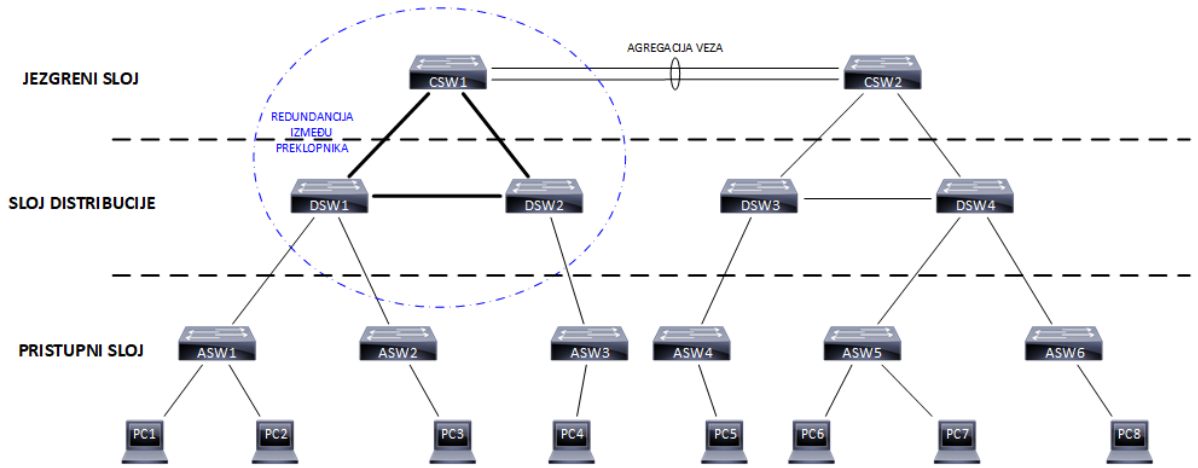
Svrha je mrežnih tehnologija funkcionalna računalna mreža koja omogućava brzu, efikasnu, pouzdanu i sigurnu komunikaciju krajnjih uređaja koji se spajaju na računalnu mrežu (npr. poslužitelj, računala, baze podataka, IP kamere, IP telefoni itd.). Svaka mrežna tehnologija odgovara na određene potrebe specifične računalne mreže, i ne koriste se sve tehnologije u svim situacijama. Hoće li se neka tehnologija koristiti, ovisi o mnogo čimbenika, kao što su njezina složenost, zahtjevnost za implementaciju i održavanje, njezina cijena, kompatibilnost s već postojećim mrežnim tehnologijama, životnom vijeku, mogućnosti podrške u slučaju problema u radu neke tehnologije, skalabilnosti, razini sigurnosti i slično. Da bi se zaista odabrala adekvatna tehnologija, treba biti dobro upoznat s poslovanjem organizacije i njezinim zahtjevima s obzirom na sustav IKT-a, pa tako i računalnu mrežu. Također treba uzeti u obzir njezina ograničenja i mogućnosti u pogledu tehnologije i upravljanja njome u životnom ciklusu. Stoga je ključno baviti se dizajnom i ekonomijom kako bi i računalna mreža, i sustav IKT-a bili optimalni u odnosu na poslovne zahtjeve organizacije.

Promatrajući sustav IKT-a kao kompleksnu cjelinu koju čine razni međusobno povezani podsustavi, računalna mreža se može smatrati podsustavom IKT-a čija je svrha povezivanje i međusobna komunikacija svih ostalih elemenata sustava IKT-a (npr. klijentsko računalo i poslužitelj na kojem se nalazi aplikacija). Računalna mreža tako dijeli sve karakteristike sustava IKT-a , kao što su sigurnost, skalabilnost, visoka dostupnost, zalihost i slično, samo se one ostvaruju na nižoj razini, korištenjem tehnologija specifičnih za mrežni podsustav, poput usmjernika, preklopnika, bežičnih pristupnih točaka, vatrozida, protokola usmjeravanja, protokola za agregaciju veza, logičku segmentaciju računalne mreže i slično.

Računalnu mrežu čine razni posrednički uređaji (engl. *Intermediary devices*) koji se nalaze unutar mreže i povezuju krajnje uređaje (engl. *End devices*) koji se nalaze na rubu mreže. Posrednički uređaji, koje u svakodnevnom radu zovemo mrežni uređaji, uglavnom su usmjernici i preklopnici, ali mogu biti i bežične pristupne točke, „*proxy* poslužitelj“, VPN koncentratori, vatrozidi, raspoređivači opterećenja (engl. *Load Balancers*) i razni drugi uređaji koji imaju specijalne namjene, dok su krajnji uređaji svi oni koji koriste računalnu mrežu, poput računala, poslužitelja, printera, IP telefona, IP kamera, CNC strojeva koji se povezuju na računalnu mrežu. Osim krajnjih i posredničkih uređaja, neizostavni su elementi računalnih mreža i mrežni ormari u kojima se nalazi razni mrežni uređaji, prespojni paneli, pretvornici signala (engl. *Media Converters*), ponavljači signala (engl. *Repeaters*), bakreni i optički kabeli i slično.

Posrednički uređaji, dakle, čine infrastrukturu računalne mreže koja služi krajnjim uređajima kako bi oni komunicirali posredstvom te računalne mreže. Te krajnje uređaje ljudi koriste za međusobnu komunikaciju i pristup raznim uslugama, kao što su društveni mediji, videoigre, novinski portali, sustavi nadzora proizvodnje, upravljanje udaljenim računalima itd. Stabilan rad i dostupnost svih tih usluga i aplikacija omogućava sustav IKT-a kao cjelina sa svim svojim temeljnim karakteristikama (npr. skalabilnost, visoka dostupnost, zalihost odnosno redundancija, sigurnost itd.).

Posrednički uređaji koriste razne protokole koji služe za razmjenu informacija o krajnjim mrežama korisnika i kako bi znali kojim putem i na koji način trebaju slati promet računalnom mrežom. Postoje razne mrežne topologije u koje možemo povezati mrežne uređaje, ovisno o veličini i kompleksnosti računalne mreže. Danas je najkorištenija topologija tzv. kampus mreža, koja je povezana hijerarhijski i sastoji se od tri sloja.



Slika 1. Ilustracija hijerarhijske računalne kampus mreže

Tri su sloja u hijerarhiji: pristupni sloj, sloj distribucije (ili sloj agregacije) i sloj jezgre. Svaki od njih ima različite karakteristike i funkcionalnosti, a samim time i uređaji koji se ugrađuju u pojedini sloj, bit će drugačiji. Na pristupnom sloju povezuju se krajnji uređaji, pa je nužno da posrednički (mrežni) uređaji imaju veliku gustoću sučelja kako bi povezali što veći broj uređaja na računalnu mrežu. Neki od njih trebaju još i napajanje električnom energijom (kao što su IP telefoni, IP kamere, bežične pristupne točke i slično) pa uređaji na pristupnom sloju trebaju napajanje *Ethernetom* (*PoE-Power over Ethernet*). Za razliku od pristupnog sloja, uređaji sloja jezgre ne trebaju tu funkcionalnost, ali svakako trebaju veliku brzinu sučelja i što manje kašnjenje (engl. *Latency*) u obradi mrežnog prometa. Uređaji distribucijskog sloja također imaju svoje specifičnosti, pa je tako njihov zadatak podržavati razne protokole za agregaciju, redundanciju i visoku dostupnost, ali i razne protokole za usmjeravanje prometa i implementaciju osnovnih sigurnosnih protokola i logičku segmentaciju. Taj je hijerarhijski dizajn modularan i skalabilan, a može se primijeniti na više razina. Tako, npr., mogu biti tri sloja hijerarhije unutar jedne bolnice, dok u mrežnoj infrastrukturi bolničkog sustava na državnoj razini, može biti hijerarhijska organizacija između pojedinih regija, pa su tako neke bolnice niže u hijerarhiji, dok su neke više. Iako je ovdje navedena primarno organizacijska hijerarhija bolničkog sustava, postoje neki elementi mrežne infrastrukture koji reflektiraju tu organizacijsku strukturu. Tako, na primjer, glavni izlaz na internet za sve bolnice može biti na primarnoj lokaciji u glavnom gradu države, gdje je i podatkovni centar s virtualnom infrastrukturom povezanom 100 Gbps višestrukim redundantnim optičkim vezama. Također mogu postojati i vatrozidi koji rade u načinu (modu) visoke dostupnosti (engl. *High Availability*), na toj se lokaciji

koristi BGP (engl. *Border Gateway Protocol*), protokol za povezivanje s telekomunikacijskim operatorima i slično.

Najjednostavnija je moguća mreža ona koju čine dva povezana računala, međutim taj opis ovdje ne odgovara jer pod pojmom mreža podrazumijevamo povezanost više korisnika u neku smislenu i funkcionalnu cjelinu, poput, recimo, učionice u školi s 20-ak računala povezanih mrežnim kabelima na preklopnik. Uz to je za njihovu međusobnu komunikaciju potrebna i određena konfiguracija (upute) kako se ta računala trebaju „predstaviti“ jedno drugome. Te postavke zovemo IP postavke, i svaki uređaj koji se povezuje na računalnu mrežu mora imati nekoliko ključnih elemenata. Za početak je dovoljno znati samo to da svako računalo koje se povezuje na računalnu mrežu nekim preklopnikom mora imati jedinstvenu IP adresu u logičkoj mreži (engl. *Subnet*) kojoj će pripadati, a sva računala u istoj logičkoj mreži moraju imati istu mrežnu masku (engl. *Subnet Mask*), trebaju koristiti isti uređaj za izlaz iz mreže (engl. *Default Gateway*) te isti imenski poslužitelj (engl. *Domain Name System — DNS*). Tehnički gledano, svako računalo moglo bi koristiti poseban uređaj za izlaz iz mreže i poseban DNS, ali to je besmisleno i praktički teško izvedivo pa se ne radi.

1.2. Karakteristike računalne mreže

1.2.1. Vrste komunikacije u računalnoj mreži

U računalnim mrežama postoji više načina na koji uređaji mogu komunicirati, ovisno o tome što žele postići. Dakle, konceptualna je razlika u komunikaciji uređaja s poznatim uređajem i kada pokušavaju otkriti uređaje u mreži. U prvom slučaju nema potrebe da pakete mrežnog prometa šalju svima koji bi mogli biti spojeni na mrežu, već samo onom uređaju s kojim su uspostavili vezu. S druge strane, kada neki uređaj nikada nije komunicirao na mreži i nema IP postavke (IP adresa, mrežna maska, izlaz iz mreže i DNS), tada ima smisla svima slati upite u nadi da u mreži postoji neki uređaj koji će mu dati IP postavke (DHCP poslužitelj).

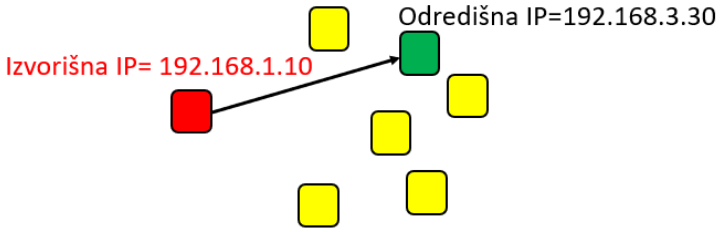
Postoje tri vrste komunikacije između samih uređaja i jedna koja se oslanja na usmjeravanje prometa u računalnoj mreži:

1. Jednoodredišna komunikacija (engl. *Unicast*)
2. Višeodredišna komunikacija (engl. *Multicast*)

3. Sveodredišna komunikacija (engl. *Broadcast*)
4. Komunikacija prema najbližem odredištu (engl. *Anycast*).

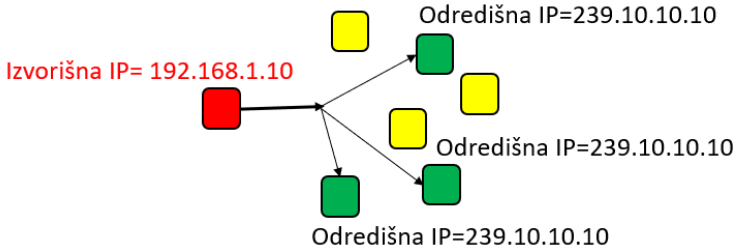
U nastavku su pobliže opisane.

- **Jednoodredišna komunikacija** – vrsta komunikacije između bilo koja dva uređaja posredstvom bilo kojeg protokola. Bitno je da oba uređaja znaju s kim komuniciraju i sav promet šalju direktno jedan drugome bez posrednika u komunikaciji. Primjer te komunikacije bio bi *ping* prema odredištu, poruka ARP odgovora, ili uspostava TCP veze između dva uređaja.



Slika 2. Jednoodredišna komunikacija

- **Višeodredišna komunikacija** – podatkovni tok (engl. *Stream*) od jednog uređaja (poslužitelja) s više drugih uređaja (primatelja prijenosa) koji imaju mogućnost „slušati“ na određenoj IP adresi. Primjer takve komunikacije bio bi IP TV videoprijenos, a u kontekstu mrežnih tehnologija, te adrese koriste protokoli usmjeravanja i razni drugi „infrastrukturni“ protokoli, npr. RIP, EIGRP, OSPF, HSRP itd.

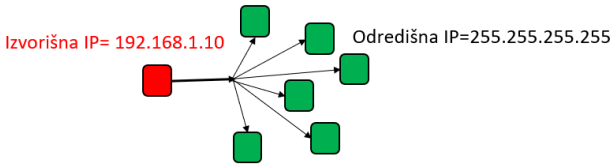


Slika 3. Višeodredišna komunikacija

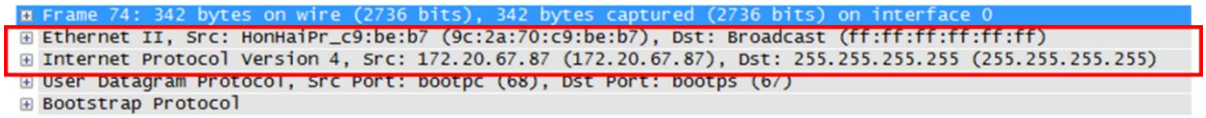
Početna IP adresa	Krajnja IP adresa	Opis
224.0.0.0	224.0.0.255	Rezervirano za često korištene multicast namjene
224.0.1.0	238.255.255.255	Rezervirane globalno dostupne multicast IP adrese (u teoriji)
239.0.0.0	239.255.255.255	Rezervirano za lokalnu primjenu unutar organizacija

Slika 4. Rasponi adresa i namjena u višeodredišnoj komunikaciji

- **Sveodredišna komunikacija** – vrsta komunikacije u kojoj jedan uređaj šalje određenu vrstu mrežnog prometa na sve uređaje u nekoj podmreži (engl. *Subnet*). Tu vrstu komunikacije uređaji koriste kada otkrivaju mrežu, npr. DHCP poruke otkrivanja (engl. *DHCP Discover*), poruke ARP zahtjeva i slično, dakle kada uređaj traži nešto, ali ne zna što traži. Odredišna IP adresa u ovom je slučaju 255.255.255.255 i taj će promet primiti svi uređaji u računalnoj mreži. Preklopnici ga prosljeđuju na sva aktivna sučelja, osim na ono odakle su primili takav promet.



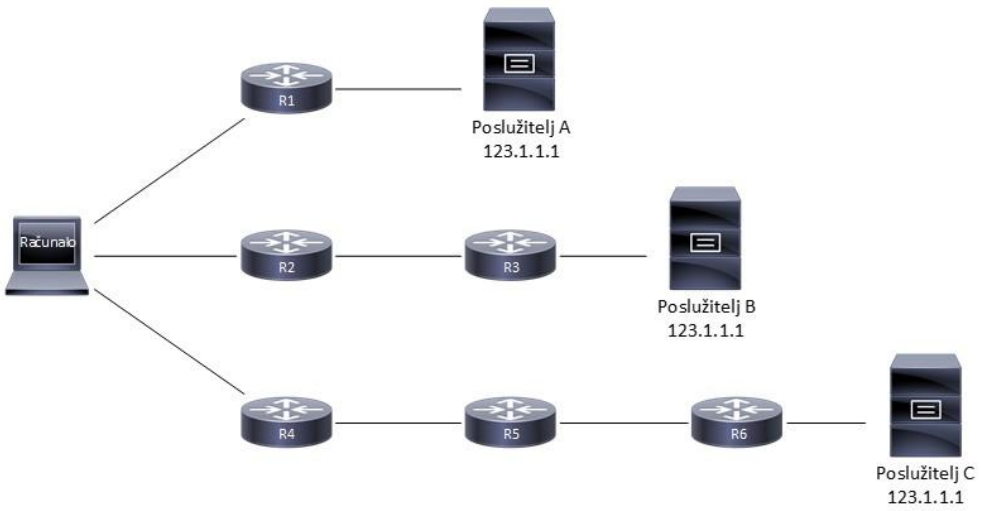
Slika 5. Sveodredišna komunikacija



Slika 6. Sveodredišna komunikacija u Wireshark alatu

- **Komunikacija prema najbližem odredištu** – vrsta komunikacije koja traži najbliže odredište i za to se oslanja na protokole usmjeravanja koji postoje u računalnoj mreži. Primjer je kada su u mreži tri DNS poslužitelja koji imaju istu IP adresu i stoga što će određenim računalima u određenom dijelu mreže biti bliži poslužitelj A nego poslužitelj B, to znači da će komunikacija uvijek biti najkraćim putem, a ako otkaže poslužitelj A jednostavno će poslužitelj B preuzeti upite od svih računala koja su prije koristila poslužitelj A. Na slici 7. prikazan je princip ove komunikacije, gdje je IP adresa DNS poslužitelja 123.10.10.10, a u stvarnosti postoje tri poslužitelja s tom IP adresom. U tom slučaju računalo će kontaktirati onaj poslužitelj do kojega vodi „najkraći put“ kroz mrežu – na slici to bi bio poslužitelj A, jer je do njega samo jedan usmjernik (R1). Računalu nije važno koji će mu poslužitelj odgovoriti na upite, dok god ima odgovora. Slično kao što ljudima

nije bitno u kojoj će trgovini kupiti mlijeko, ima li ga u svim trgovinama, stoga idu u najbližu.

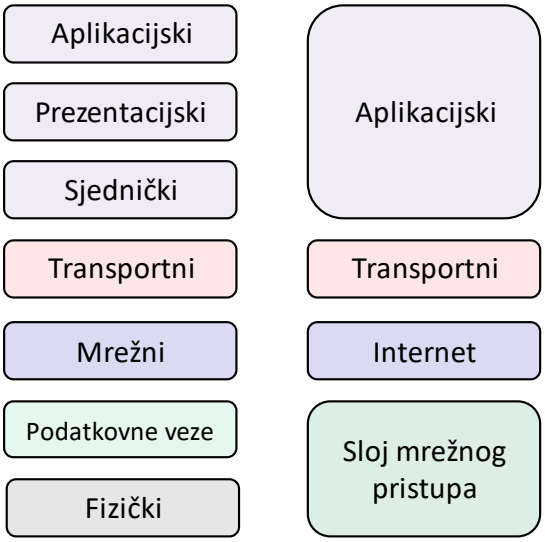


Slika 7. Komunikacija prema najbližem odredištu

1.2.2. Model OSI i TCP/IP

Kada je internet nastajao, bilo je vrlo malo korisnika te nove tehnologije i samim time nije bilo potrebe, niti je bilo opravdano razvijati nekakav kompleksan standard jer nije bilo ni jasno u kojem smjeru će se on razvijati. Tek kada su prednosti komunikacije na daljinu posredstvom TCP/IP protokola osjetili mnogi, i kada su se počele povezivati različite organizacije, postalo je nužno razviti standard koji će omogućiti jasno razumijevanje funkcioniranja interneta, pisanja koda, kako treba izgledati mrežni kabel, koje frekvencije i razine napona koristiti za odašiljanje signala i slično. Kao odgovor na ove realne potrebe standardizacije razvijen je model OSI (engl. *Open Systems Interconnection*). To je referentni arhitekturni model od sedam slojeva korišten u računalnim mrežama za razumijevanje mrežne komunikacije i interoperabilnosti različitih mrežnih komponenti. OSI model je definirala organizacija ISO (engl. *International Organization for Standardization*) 1977. godine. Sama organizacija ISO stvorena je kao odgovor na potrebu standardizacije tehnologija, a u današnjem svijetu gdje se tehnologije ubrzano razvijaju, potreba za standardizacijom koja će omogućiti interoperabilnost, efikasnost, isplativost i jednostavnost, neophodna je. Jedna je od glavnih prednosti standardizacije svakako kompatibilnost hardvera i softvera različitih proizvođača odnosno da je krajnjim korisnicima svejedno koju će kombinaciju opreme koristiti. Tako će danas mrežna kartica proizvođača *Helvet Packard* uspješno raditi ako

se spoji na preklopnik tvrtke *Cisco* ili tvrtke *Arista*. Za razliku od OSI modela, koji je vrlo detaljan i postoje jasne granice i jasne uloge svakog od sedam slojeva njemu, TCP/IP model više je usmjeren prema praktičnoj upotrebi i koristi se za generalizaciju onoga što se događa u stvarnosti kada uređaji u računalnoj mreži komuniciraju. Tako da su peti, šesti i sedmi sloj OSI modela u TCP/IP modelu jednostavno spojeni u jedan – aplikacijski sloj. U praktičnoj primjeni i samoj mrežnoj komunikaciji ti slojevi zaista i jesu dio nekog softvera koji radi sve ono što je opisano u petom, šestom i sedmom sloju OSI modela, tj. od svega što neka aplikacija radi, stvaraju se podatci za slanje u binarnom obliku i proslijede transportnom sloju na segmentiranje i označavanje.



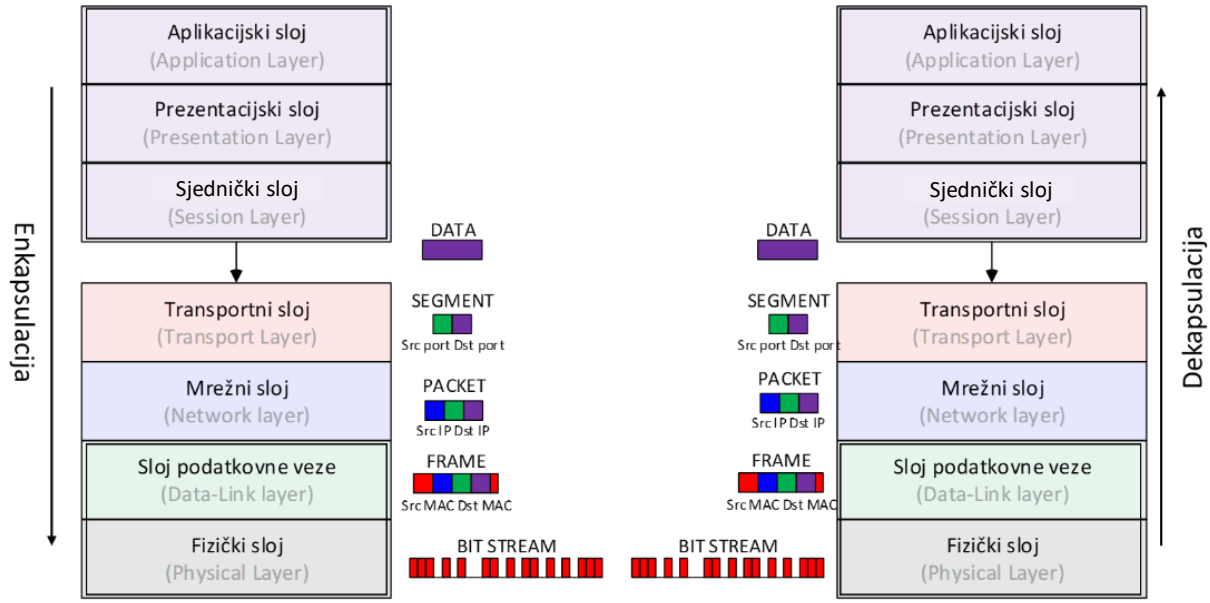
Slika 8. Usporedba modela OSI i TCP/IP

Transportni sloj ima istu jedinstvenu ulogu u oba modela, a to je segmentiranje podataka koji dolaze od aplikacije i označavanje koja aplikacija šalje određene podatke koristeći TCP/UDP ulaze. Isti slučaj je i s trećim slojem OSI modela, koji segmente objedinjuje u pakete i označava odredište koje treba primiti te segmente (podatke), samo se u slučaju TCP/IP modela taj sloj naziva internetski sloj. Gledajući što se zaista događa u stvarnosti, TCP/IP model objedinjuje prvi i drugi sloj OSI modela, čija je glavna uloga da svu apstraktnu logiku koja se događa nad podatcima u računalu nekako pretvori u stvarne električne ili svjetlosne impulse i odašilje ih na neki medij za prijenos podataka. Može se, stoga, reći da je OSI model vrlo detaljan referentni model koji opisuje što se „u teoriji“ događa za vrijeme komunikacije u računalnoj mreži, dok TCP/IP model objašnjava što se događa „u praksi“, bez ulaska u detalje. U suštini oba

modela objašnjavaju isto, stoga je važnije razumjeti kako se komunikacija između uređaja odvija koristeći OSI model, pa će TCP/IP samo trebati prihvatiti.

1.2.3. Uloga pojedinih slojeva OSI modela − enkapsulacija i dekapsulacija

Slojevi OSI modela i njihova uloga u komunikaciji mogu se predstaviti od bilo kojeg sloja, a primarno 1. ili 7. Početak je stvaranja podataka koji će u konačnici biti poslani u stvarni svijet u 7. sloju, pa je najbolje krenuti od njega, od početka komunikacije. Taj se proces opisivanja onoga što se događa s podatcima aplikacijskog sloja prema fizičkom sloju, i koje su karakteristike, protokoli i adrese (tamo gdje ih ima) na svakom sloju, naziva enkapsulacija, dok se proces na odredišnom računalu koje prima te podatke na fizičkom sloju i prosljeđuje ih do aplikacijskog sloja naziva dekapsulacija. Prikazan je na slici 9.



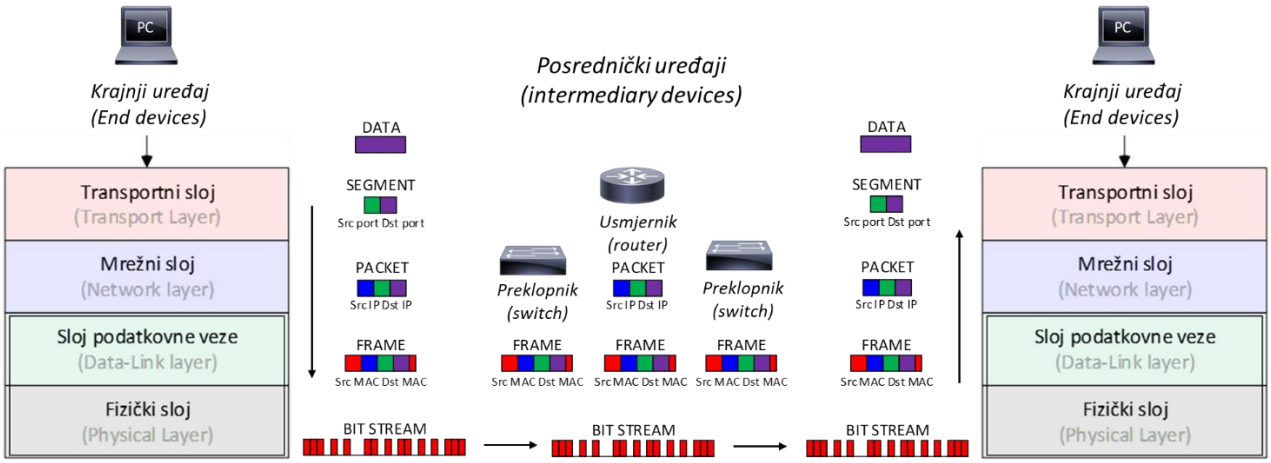
Slika 9. Prikaz procesa enkapsulacije i dekapsulacije podataka u slojevima OSI modela

Razumijevanje procesa enkapsulacije ključno je za kasnije razumijevanje ukupne komunikacije u mreži i kako uređaji na temelju podataka u različitim poljima na različitim slojevima OSI modela donose odluke što činiti s pojedinim segmentom (naziv za PDU na transportnom sloju), paketom (naziv za PDU na mrežnom sloju) ili okvirom (naziv za PDU na sloju podatkovne veze). Podatci koje generira aplikacija prosljeđuju se transportnom sloju koji ih opisuje svim potrebnim podatcima da bi ih odredišno računalo moglo adekvatno obraditi. Ovdje je važno zapamtiti da se, između ostaloga,

koriste brojevi izvorišnoga i odredišnog ulaza za identifikaciju aplikacije koja šalje podatke i aplikacije koja prima podatke. Kada transportni sloj obavi svoj dio posla, prosljeđuje segment mrežnom sloju. Mrežni sloj sada taj segment opisuje podatcima koji su potrebni da dođe do odredišta i tu se, između ostalih polja, koriste izvorišna i odredišna IP adresa uređaja. Kada mrežni sloj obavi svoj dio posla, prosljeđuje paket prema sloju podatkovne veze, koji tom paketu dodaje svoje opisne podatke potrebne kako bi se paket mogao prosljeđivati od uređaja do uređaja na putu do odredišne mreže. Treba spomenuti da se, između ostalih podataka, koriste MAC adrese, zovemo ih još i fizičke adrese. Svaki uređaj koji ima mrežnu karticu, ima i tvornički dodijeljenu MAC adresu, koja se koristi samo za prosljeđivanje prometa unutar lokalne računalne mreže između direktno spojenih uređaja i uređaja koji su povezani posredstvom preklopnika L2 vezom (kasnije se objašnjava što je L2 veza).

Za razliku od svih ostalih slojeva, sloj podatkovne veze, osim zaglavlja, dodaje i završetak (engl. *Trailer*) u kojem se nalaze podatci potrebni za kontrolu grešaka u prijenosu i jasno se označava kraj okvira. Sloj podatkovne veze još ima ključnu zadaću upravljati fizičkim slojem kako bi podatke odaslao na mrežu u obliku električne energije, svjetlosnih signala ili elektromagnetskih valova.

Kada ti podatci stignu do odredišta, proces se događa obrnutim redoslijedom. Signal koji putuje medijima za prijenos podataka (npr. bakreni ili optički kabeli) posrednički uređaji, kao što su usmjernici ili preklopnici, više puta obrade, i svaki od tih uređaja dekapsulira podatke do sloja na kojem se nalaze podatci na temelju kojih konkretan uređaj donosi odluku. Primjer je usmjernik koji mora pogledati odredišnu IP adresu i dekapsulira sve podatke do mrežnog sloja, a zatim ih opet sve enkapsulira i odašilje na medij za prijenos podataka.



Slika 10. Prikaz enkapsulacije i dekapsulacije podataka na putu kroz mrežu na prva četiri sloja OSI modela

1.2.4. Aplikacijski sloj (sedmi)

Aplikacijski sloj, iako je sedmi po udaljenosti od stvarnog svijeta, u stvari je prvi sloj s kojim je većina korisnika u interakciji. Aplikacijski sloj ono je što korisnik vidi kada koristi računalo. Ključne mogućnosti koje pruža odnose se na korisničko sučelje i sve procedure i oblike kojima se korisnici služe za interakciju s nekom aplikacijom na računalu. Taj je sloj povezan s utvrđivanjem raspoloživih resursa za aplikaciju, kao što su brzina promjene slike (engl. *Frame Rate*) videoigara, količina raspoložive grafičke memorije i slično. Dijeljenjem tih resursa između aplikacija omogućava se autentikacija i autorizacija korisnika za pristup resursima sustava IKT-a, pruža se podrška za prijenos podataka (sučelje, procedure, programski kod itd.) i mnogi drugi važni elementi za uspješnu komunikaciju i korištenje sustava IKT-a. Aplikacijski sloj može se vidjeti svaki puta kada se koriste klijenti e-pošte (npr. *Microsoft Outlook*, *Gmail*), internetski pretraživači (npr. *Google Chrome*, *Mozilla Firefox*), uspostavlja VPN veza ili koriste alati za videokonferencije (npr. *Skype*, *Zoom*, *Google Meet*)

1.2.5. Prezentacijski sloj (šesti)

Uloga prezentacijskog sloja priprema je, oblikovanje i konverzija podataka kako bi ih aplikacijski sloj mogao koristiti (dalje: aplikacije). Ključne su funkcionalnosti prezentacijskog sloja kompresija i dekompresija podataka, enkripcija i dekripcija podataka, pretvaranje podataka iz jednog formata u drugi kako bi bili čitljivi aplikacijama različitih proizvođača. Primjeri primjene prezentacijskog sloja vidljivi su prilikom korištenja alatima poput *WinZip* ili *7-Zip* za kompresiju ili alatima *BitLocker*, *TrueCrypt* za kriptiranje podataka, prilikom promjene formata podataka na primjeru

video i audio kodeka H.264 u MP3 i slično. Aplikacije koriste prezentacijski sloj kako bi ispravno oblikovale podatke prije nego ih se pošalje preko mreže, tako da ih primatelj može koristiti. Zbog konzistentnosti podataka prezentacijski sloj utječe na to da komunikacija između različitih sustava (npr. *Windows* i *Linux*) bude uredna i bez grešaka ili barem dovoljno dobra iz perspektive korisnika (kao u primjeru čitanja *MS Word* dokumenta u *Libre Office* alatu).

1.2.6. Sjednički sloj (peti)

Uloga sjedničkog sloja uspostava je, održavanje i uredno zatvaranje veze između aplikacija na dva ili više računala. To ne treba miješati s uspostavom veze na transportnom sloju jer nije isto. Sjednički sloj uspostavlja vezu na razini aplikacije, dok transportni sloj uspostavlja vezu na razini računala kao uređaja i glavna uloga mu je da dolazni promet ispravno usmjeri prema odgovarajućoj aplikaciji. Ta su dva sloja neovisna u svojim zadaćama. Sjednički sloj odgovoran je za uspostavu veze, razmjenu podataka, „koordinaciju“ protoka podataka i sinkronizaciju aplikacija koje komuniciraju. Osim uspostave, održavanja i raskidanja veze između aplikacija, sjednički sloj odgovoran je i za detekciju i ispravljanje grešaka u komunikaciji između aplikacija (niži slojevi toga nisu „svjesni“). Primjer primjene sjedničkog sloja vidljiv je prilikom korištenja raznih aplikacija poput RDP, *Telnet*, FTP, VPN i slično. Sjednički slojkoristi se i za uspostavu, održavanje, raskidanje i upravljanje vezom između različitih aplikacija na različitim operacijskim sustavima (npr. kada se uspostavlja VPN veza između uređaja *Cisco* i *Fortinet*)

Iako su slojevi OSI modela od petoga do sedmog opisani zasebno, u praksi ih se rijetko tako promatra. Obično je to jedan sloj, baš kako je prikazano u TCP/IP modelu, pa je potrebno samo tako prihvatiti. To su dijelovi aplikacije koja ima sve potrebno za upotrebu (npr. grafičko sučelje) te standardno formatiranje i kriptiranje podataka kako bi bili čitljivi i drugim sličnim aplikacijama (npr. uspostava IPSec VPN S2S tunela između vatrozida *Cisco* i *Fortinet*), a također za uspješno i ispravnim redoslijedom slaganje koraka da bi se veza uspostavila (npr. uspostava IPSec VPN tunela gdje je nužno u nekom trenutku razmijeniti ključeve ili u nekom drugom trenutku „pregovarati“ o razini enkripcije ili periodički slati poruke kako bi tunel bio aktivan i slično).

1.2.7. Transportni sloj (četvrti)

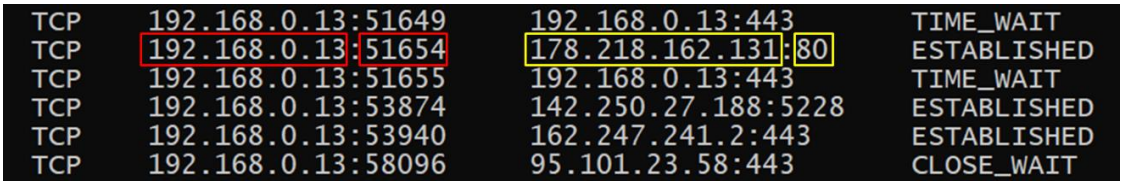
Dok se prethodna tri sloja mogu promatrati kao dio aplikacije i direktno vidljivo korisniku, slojevi od četvrtog do prvog (transportni, mrežni, sloj podatkovne veze i fizički sloj) infrastrukturni su slojevi koji podatke aplikacije moraju opisati na određen način kako bi uređaji koji čine mrežnu infrastrukturu (npr. preklopnici, usmjernici i vatrozidi) znali što s tim podatcima učiniti. Kada to odrede, trebaju te podatke pretvoriti u signal koji odašilju u fizički svijet pomoću elektromagnetskih valova (*WiFi*), električne energije (UTP kabeli) ili svjetlosnih impulsa (optički kabeli).

Ključne su uloge transportnog sloja: segmentiranje aplikacijskih podataka, multipleksiranje i detekcija grešaka.

1. **Segmentiranje podataka** – podatke koje aplikacija „spušta“ prema mreži važno je segmentirati (usitniti) do veličine koja omogućava da se ti podatci uspješno pošalju računalnom mrežom. Postoje razna ograničenja veličine segmenata koje mreža podržava, ovisno o tipu veze i operacijskom sustavu, je li u pitanju TCP ulaz ili UDP ulaz itd. Tako usitnjene podatke u obliku TCP segmenata ili UDP datagrama transportni sloj na odredišnom računalu prihvaća i prosljeđuje odredišnoj aplikaciji.
2. **Multipleksiranje/demultipleksiranje** – budući da na računalu postoji mnogo različitih aplikacija koje istovremeno komuniciraju, zadaća je transportnog sloja nedvosmisleno označavanje i razlikovanje tih aplikacija. Svaki TCP segment odnosno UDP datagram nastao usitnjavanjem podataka koji dolaze od aplikacije, dobije svoj „broj ulaza“ koji jasno identificira koja aplikacija šalje i koja aplikacija na odredišnom računalu treba primiti podatke. U kombinaciji s IP adresom izvorišta i odredišta računalo formira „logičku utičnicu“ (engl. *Socket*) koja jedinstveno identificira komunikacije svih aplikacija na računalu.

   Zahvaljujući brojevima TCP/UDP ulaza u kombinaciji s IP adresama izvora i odredišta komunikacije računala, među stotinama pokrenutih aplikacija, mogu se precizno dostaviti podatci pravoj aplikaciji. Ta je funkcionalnost nužna kako bi sve aplikacije mogle pristupiti mediju za prijenos podataka. Bez multipleksiranja ne bismo mogli istovremeno koristiti više aplikacija kao što je danas uobičajeno, npr. istovremeno igrati videoigru i slušati glazbu, kopirati

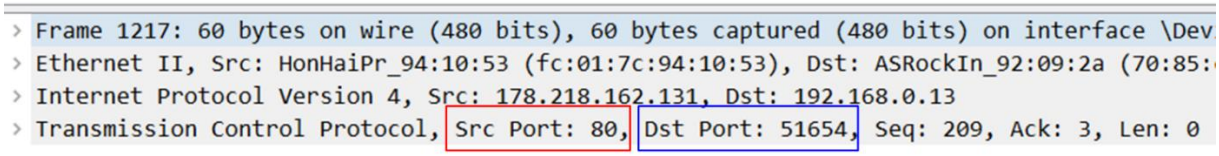
datoteke na mrežni disk, imati otvoreno dvadesetak mrežnih stranica u internetskom pregledniku, a operativni sustav u pozadini preuzimati najnovija ažuriranja.



Slika 11. Prikaz jedne „logičke utičnice“ na računalu – kombinacija src IP + src port i dst IP + dst port

Postoje tri grupe ulaza:

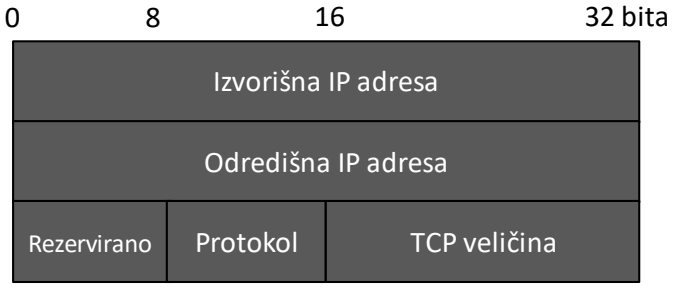
- Dobro poznati tzv. Sistemski TCP/UDP ulazi (0 – 1023) – registrirani su za dobro poznate sveprisutne aplikacije i može ih koristiti samo sustav, poput *Telneta* (23), SSH (22), FTP (20 i 21), DNS (53), HTTP (80), HTTPS (443), SNMP (161), NTP (123).
- Registrirani TCP/UDP ulazi (1024 – 49151) – dodjeljuje ih organizacija IANA i može ih se koristiti za razne korisničke aplikacije i servise koji trebaju isti ulaz za sve korisnike. Primjer je primjene ovakvih ulaza videoigra *Call of Duty*, koja koristi TCP i UDP ulaz 3074, baza podataka SQL, koja koristi TCP ulaze 1433 i 1434, razni korisnički alati kao *Adobe Media,* koji koristi ulaz 1935, infrastrukturni mrežni protokoli poput HSRP-a za IPv4, koji koristi UDP ulaz 1985.
- Privatni ili dinamički TCP/UDP ulazi – ne mogu se registrirati, i bilo koja aplikacija ih može koristiti kada se uspostavlja veza prema nekom poslužitelju. Računalo za uspostavu veze koristi izvorišni broj TCP/UDP ulaza viši od 49151.



Slika 12. Prikaz izvorišnog Dinamičkog TCP/UDP ulaza i odredišnog tzv. Sistemskog TCP/UDP ulaza

Detekcija grešaka – u zaglavlju TCP segmenta, ali i UDP datagrama postoji polje naziva kontrolni izračun (eng *Checksum*) koje se koristi za osnovno otkrivanje grešaka. Taj kontrolni izračun koji se radi nad TCP/UDP zaglavljem, podatcima u TCP segmentu / UDP datagramu i pseudo IP zaglavlju (koje se sastoji od 12 bajta podataka prikazanih na slici 13.) izračunava pošiljatelj koristeći određeni algoritam. Izračun se šalje u zaglavlju TCP segmenta. Kada taj segment/datagram stigne na odredišno računalo,

ponovno se radi kontrolni izračun koristeći isti algoritam i uspoređuje se vrijednost koja je bila u polju kontrolnoga izračuna i dobivena vrijednost na odredišnom računalu. Ako vrijednosti nisu iste, taj se segment/datagram odbacuje. Iako na putu između izvora i odredišta može postojati mehanizam kao što je NAT, gdje se mogu mijenjati IP adrese ili ulazi, uređaji će na putu (npr. usmjernici) napraviti ponovni kontrolni izračun tako da komunikacija bude moguća i da se promet ne odbacuje. IP pseudo zaglavlje koristi se samo za kontrolni izračun i nije dio podataka koji se šalju. Elementi IP pseudo zaglavlja izvorišna su i odredišna IPv4 adresa (kopirana iz normalnog IP zaglavlja), polje protokola (TCP ili UDP kopirano iz normalnog IP zaglavlja), veličina TCP segmenta/datagrama (kopirano iz normalnog IP zaglavlja) i 8 rezerviranih bitova koji su nula.



Slika 13. Pseudo zaglavlje

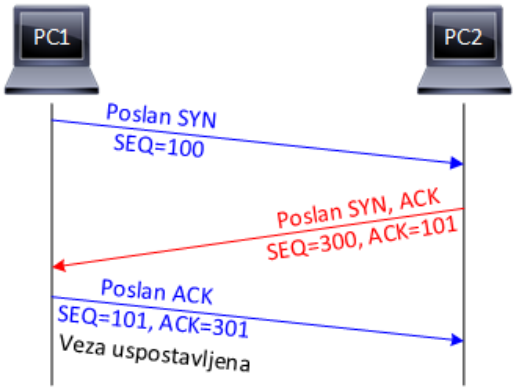
TCP I UDP

Glavni protokoli transportnog sloja su TCP i UDP. Razlika između ovih protokola u njihovoj je namjeni. TCP se koristi kada svi podatci koji se šalju moraju doći na odredište i točno određenim redoslijedom biti predstavljeni aplikaciji kako bi ih ona mogla obraditi i prikazati korisniku. Primjer aplikacije koja treba TCP slanje poruka je e-pošte s fotografijom u prilogu. U tom slučaju ukoliko nešto od podataka ne dođe na odredište, utoliko vjerojatno neće biti moguće otvoriti e-poruku, niti pogledati fotografiju u prilogu. Primjeri su aplikacija koje koriste UDP sve koje rade u realnom vremenu, poput prijenosa govora posredstvom IP protokola (IP telefonija), videoprijenosa ili aplikacija koje otkrivaju mrežu, npr. DHCP ili DNS upiti, gdje nije potrebno svrsishodno ostvarivati vezu.

TCP

Glavne su karakteristike TCP protokola:

Uredno uspostavljanje i raskidanje dvosmjerne komunikacijske veze: TCP koristi mehanizam trostrukog rukovanja (engl. *Three-way Handshake*) mehanizam za pouzdanu uspostavu dvosmjerne komunikacije između računala na transportnom sloju. Prije nego što aplikacija na izvorišnom i aplikacija na odredišnom računalu počnu komunicirati, TCP uspostavlja vezu na transportnom sloju koristeći mehanizam trostrukog rukovanja, vidljiv na slici 14.



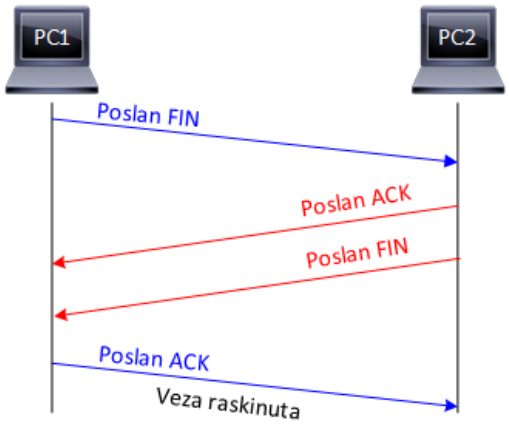
Slika 14. Ilustracija uspostave TCP veze posredstvom mehanizma trostrukog rukovanja

Na slici 15. prikazana je uspostava TCP veze u *Wireshark* alatu.

Source	Destination	Protocol	Length	Info
192.168.0.13	178.218.162.131	TCP	66	51539 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
178.218.162.131	192.168.0.13	TCP	66	80 → 51539 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=1024
192.168.0.13	178.218.162.131	TCP	54	51539 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0

Slika 15. Prikaz mehanizma trostrukog rukovanja u Wireshark alatu

TCP također upravlja tom vezom dok je uspostavljena koristeći druge mehanizme i, nakon što su aplikacije međusobno razmijenile sve podatke, TCP na uredan način zatvara vezu, kako je prikazano na slici 16.



Slika 16. Ilustracija raskida TCP veze

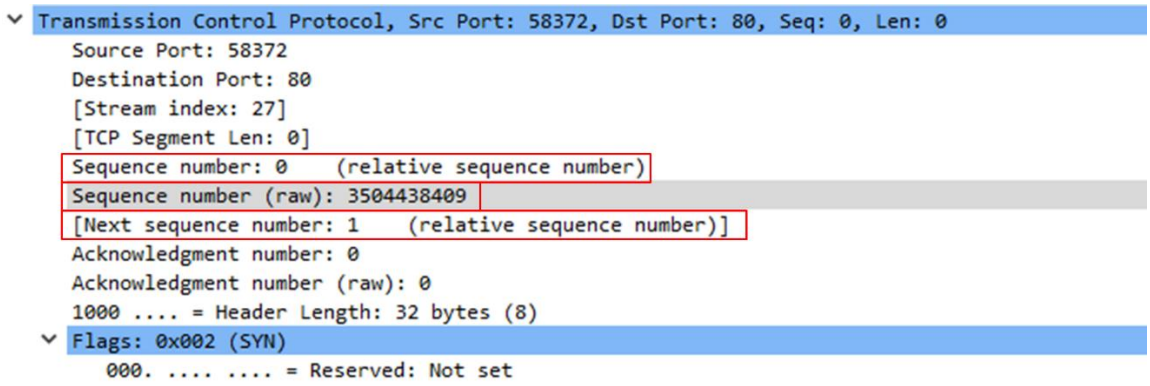
Zatvaranje TCP veze na uredan način odvija se u četiri koraka. U prvom koraku jedna strana postavlja FIN zastavicu kako bi signalizirala da traži raskid TCP veze. Drugi je korak da poslužitelj taj zahtjev za prekidom potvrđuje. Treći je korak da poslužitelj, nakon što je poslao sve preostale segmente klijentu, također traži raskid veze s klijentom slanjem FIN zastavice. Četvrti je korak da klijent potvrdi raskid veze s poslužiteljem slanjem ACK zastavice. Nakon tih četiriju koraka TCP veza je prekinuta. Dakle obje strane u komunikaciji moraju i zatražiti prekid TCP veze i dobiti potvrdu da je veza raskinuta, slično kao i među ljudima, samo što u slučaju računala svaki korak mora biti izveden svaki puta na isti način (protokolarno) jer računala nemaju svijest o tome što se događa, niti imaju vremensku vezu prošlosti, sadašnjosti i budućnosti kao ljudi, ona prolaze unaprijed jasno definiranim koracima raskida veze. Ako računalo slučajno ne bi dobilo zastavicu FIN od poslužitelja, ono bi moralo koristiti neki drugi mehanizam da tu vezu u nekom trenutku zatvori i oslobodi resurse, a to je obično neki predefinirani vremenski period (engl. *Timer*).

Source	Destination	Protocol	Length	Info
178.218.162.131	192.168.0.13	TCP	60	80 → 51539 [FIN, ACK] Seq=1 Ack=1 Win=29696 Len=0
192.168.0.13	178.218.162.131	TCP	54	51539 → 80 [ACK] Seq=1 Ack=2 Win=262656 Len=0
192.168.0.13	178.218.162.131	TCP	54	51539 → 80 [FIN, ACK] Seq=1 Ack=2 Win=262656 Len=0
178.218.162.131	192.168.0.13	TCP	60	80 → 51539 [ACK] Seq=2 Ack=2 Win=29696 Len=0

Slika 17. Prikaz raskida TCP veze u Wireshark alatu

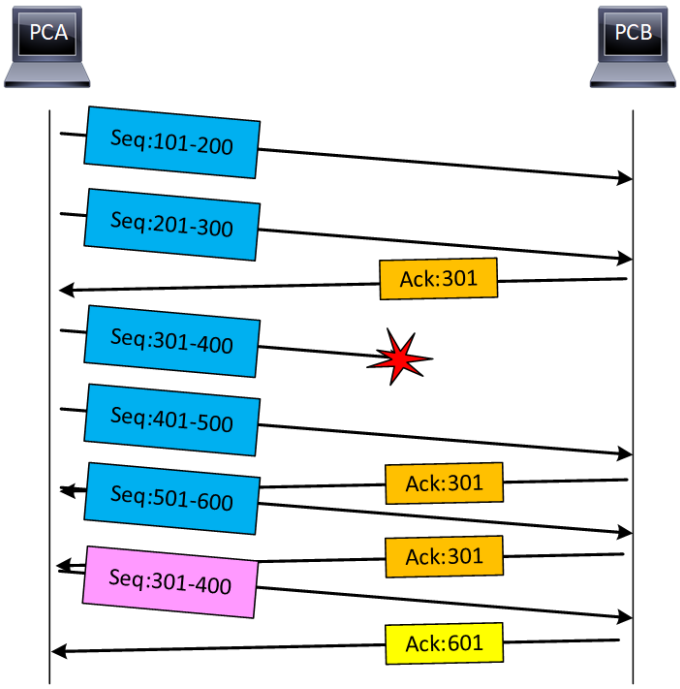
Treba imati na umu da je početni broj sekvence TCP-a koji računalo generira nasumičan i to je stvarni broj (engl. *Raw number*) koji je dodijeljen poruci TCP protokola (TCP segmentu). U alatu *Wireshark* taj je broj prikazan kao relativni broj u odnosu na

prvu poruku TCP protokola koju je primio, jer je tako lakše analizirati promet. Na slici 18. vidljivi su i relativni i stvarni slijedni brojevi.



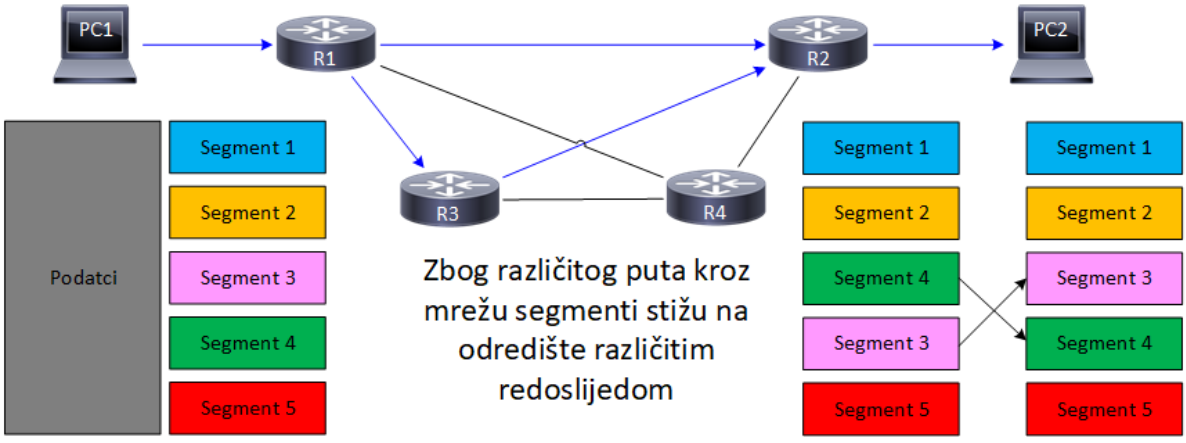
Slika 18. Prikaz slijednih brojeva u TCP segmentu u Wireshark alatu

Pouzdana dostava i preslagivanje segmenata (engl. *Data Reassembly*): TCP koristi slijedne brojeve (engl. *Sequence numbers*) kako bi označio svaki segment koji šalje prema odredištu. Kada segmenti dođu na odredišno računalo, ono će potvrditi da je primilo određene segmente tako da u odgovoru postavi naredni slijedni broj segmenta koji očekuje (engl. *Expectational acknowledgement*). Odredišno računalo može poslati izvorišnom računalu naredni slijedni broj jedino ako je primilo prethodno poslane segmente. Zbog bilo kojeg razloga može se dogoditi da neki segmenti ne dođu na odredište, i u ovom slučaju TCP mehanizam ima rješenje: ponovno slanje segmenata koji nisu stigli na odredište. Računalo koje nije dobilo određene segmente zapravo šalje slijedni broj onog segmenta koji nije stigao na odredište, dok god taj segment ne bude ponovno poslan. Nakon što je izgubljeni segment poslan i primljen, komunikacija se nastavlja kao i prije.



Slika 19. Ilustracija potvrde izgubljenog TCP segmenta do ponovnog slanja

Kada odredišno računalo primi određeni niz segmenata svaki sa svojim slijednim brojem, iskoristit će te slijedne brojeve kako bi ispravnim redom proslijedilo primljene podatke aplikaciji.



Slika 20. Prikaz preslagivanja segmenata na odredišnom računalu prije prosljeđivanja aplikaciji

Veličina prozora (engl. *Windows Size*) – to se polje iz perspektive pošiljatelja naziva prozor slanja (engl. *Send Window*), a iz perspektive primatelja prozor primanja (engl. *Receive Window*). Svako računalo kada koristi ovo polje, upisuje informaciju o tome koliko podataka (u bajtovima) može primiti i držati u privremenoj memoriji tijekom komunikacije (engl. *Buffer*). Računalo koje šalje podatke, nakon što pošalje

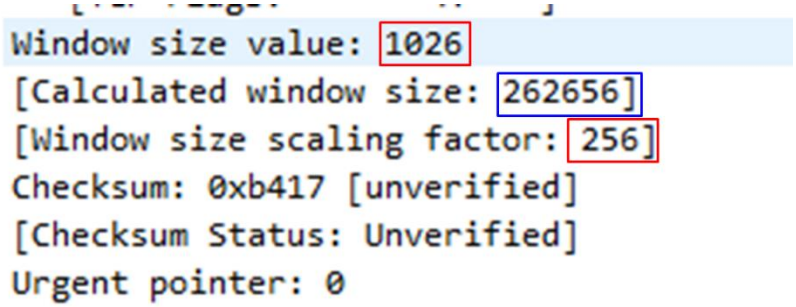
dogovorenu količinu podataka (engl. *Window size*), morat će pričekati potvrdu primitka poslanih podataka prije nego što nastavi slati nove. O veličini prozora pregovara se prilikom uspostave veze, i to ovisi o implementaciji TCP-a i vrsti veze (npr. za *Ethernet* veze obično početna iznosi 16384 bajta, odnosno dvanaest segmenata veličine 1460 bajta). Maksimalna je vrijednost 16-bitne veličine prozora 65536 bajta (64KB), ali postoji opcija skaliranja prozora (engl. *Window scaling*), povećanja količine podataka koji se mogu poslati prije slanja potvrde o primitku do čak 1GB, i tako iskoristiti kapacitete modernih sustava IKT-a. Ponekad vrijednost veličine prozora može biti nula (0), npr. u slučaju kada računalo komunicira s printerom ili nekim drugim uređajem koji ima fizička ograničenja iskorištavanja podataka koje prima (printer ima malu memoriju i nužnost pomicanja glave za printanje u stvarnom svijetu, što je daleko sporije od slanja i primanja podataka), pa podatci mogu brže pristizati nego što ih printer može koristiti.

Upravljanje tokom podataka (engl. *Flow Control*) – svrha je tog mehanizma da računalo koje šalje podatke (segmente), ne pošalje više podataka nego što odredišno računalo može primiti, dakle da se ne preoptereti primatelj, ali isto tako da računalo koje šalje podatke poveća količinu podataka koje šalje kako bi se efikasno raspolagalo resursima. Za tu namjenu koristi se veličina prozora (engl. *Windows Size*) od 16 bita. Transportni sloj primljene podatke privremeno sprema u dedicirani memorijski prostor iz kojega onda aplikacija te podatke čita. Ne iskoristi li zbog bilo kojeg razloga aplikacija sve podatke iz memorije, možda je memorija puna i nije više moguće primati nove podatke, što će uzrokovati odbacivanje i greške u prijenosu. Kako bi se to izbjeglo, TCP koristi mehanizam kontrole toka podataka. S obzirom na to da je komunikacija dvosmjerna, oba će računala jedno drugome komunicirati koliko podataka mogu primiti, a da ne potroše svu memoriju – to se zove prozor primanja (engl. *Receive Window*). Na temelju prozora primanja odredišnog računala, računalo koje šalje podatke prilagodit će svoj prozor slanja (engl. *Send Window*). Taj je mehanizam aktivan tijekom cijele komunikacije i prilagođava se po potrebi (engl. *Sliding Window*). Ako je računalo koje šalje podatke poslalo količinu podataka koja odgovara prozoru primanja odredišnog računala, tada prestane slati podatke i čeka potvrdu odredišnog računala da ih je sve primilo.

Faktor skaliranja prozora (engl. *Window Scaling factor*) – to se polje koristi i o njemu pregovara prilikom uspostave veze trostrukim rukovanjem (engl. *Three-way*

Handshake) i služi da bi komunikacija u modernim mrežama, koje imaju znatno veće kapacitete i puno su pouzdanije nego u početcima korištenja TCP-a, bila učinkovitija. Faktor skaliranja (engl. *Window Size Scaling Factor*) koristi se kako bi njegovim množenjem s početnom veličinom prozora (engl. *Windows Size Value*) dobili stvarnu veličinu prozora (engl. *Calculated Window Size*) koja će se koristiti u komunikaciji te ona može biti čak 1 GB.

Primjer: *Window size scaling factor* × *Windows size value* = 262656 bajta.



Slika 21. Umnožak (stvarna veličina prozora) faktora skaliranja i početne veličine prozora

Osim tih mehanizama, TCP koristi i složenije mehanizme za upravljanje zagušenjima, ali oni nisu dio ovog priručnika. Svi oni rade u pozadini i nepromjenjiva su činjenica vezana za TCP protokol.

OBJAŠNJENJE POLJA U TCP ZAGLAVLJU

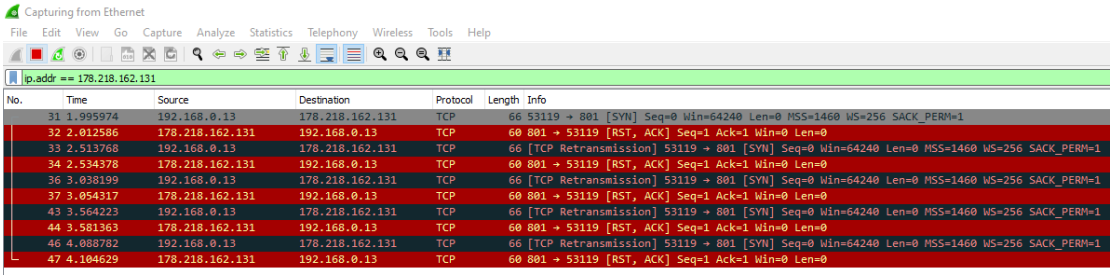
Na slici 22. prikazan je izgled TCP zaglavlja sa svim poljima koja se koriste.

Slika 22. Zaglavlje TCP segmenta

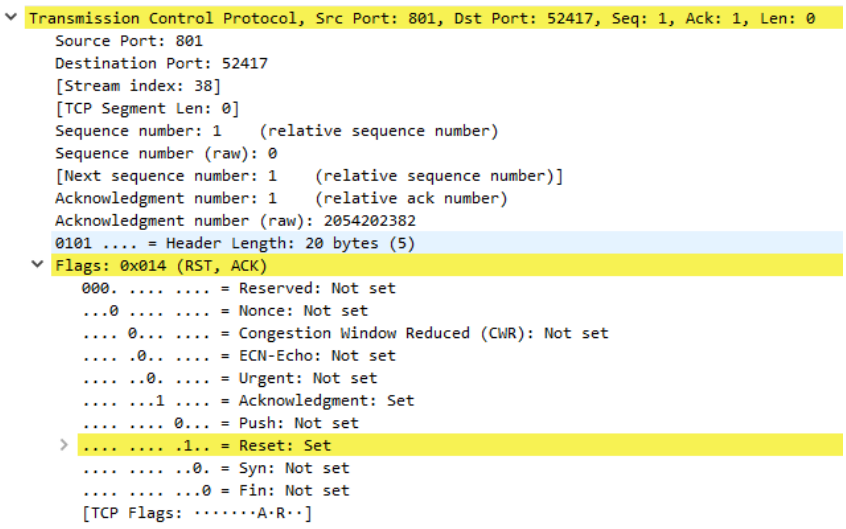
- **Izvorišni broj priključka** (engl. *Source Port Number*) – veličina 16 bita. Identificira aplikaciju koja šalje podatke s izvorišnog računala;
- **Odredišni broj priključka** (engl. *Destination Port Number*) – veličina 16 bita. Identificira aplikaciju koja prima podatke na odredišnom računalu;
- **Slijedni broj** (engl. *Sequence Number*) – veličina 32 bita. U početku komunikacije odabran nasumično i označava prvi oktet (bajt) u podatkovnom dijelu TCP segmenta. To je ključno za sinkronizaciju i upravljanje pouzdanim prijenosom podataka;
- **Potvrdni broj** (engl. *Acknowledgement Number*) – veličina 32 bita. Koristi se kao potvrda primitka nekog segmenta na način da ovaj broj odgovara narednom slijednom broju bajta koji se očekuje (engl. *Expectational acknowledgement*) u nekom drugom segmentu. Ne potvrđuje se svaki bajt. Na primjer, kada odredišno računalo primi TCP segment s bajtovima 0, 1 ,2 i 3, šalje potvrdu 4, a to je oznaka prvog bajta u segmentu koji se očekuje;
- **Polje 'veličina zaglavlja'** (engl. *Header Length – HLEN*) – veličina 4 bita. U tom polju piše koliko 32-bitnih riječi (engl. *Word*) čini TCP zaglavlje. Pomaže da primatelj zna gdje završava zaglavlje, a gdje počinje. Veličina TCP zaglavlja može biti od 20 bajtova (5 32-bitnih riječi) do 60 bajtova (15 32-bitnih riječi);

- **Bitovi kontrolnih zastavica** (engl. *Control Flag Bits*) – postoji 6 bitova koji se koriste za kontrolu komunikacije:
  - **Zastavica hitnosti: URG** (engl. *Urgent pointer*) – taj bit označava da se podatci trebaju dostaviti što je prije moguće, bez čekanja u redu. Također, odredišno računalo provjerava to polje kako bi pročitalo pomak (engl. *Off-set*) u trenutnom toku podataka. Zastavica hitnosti ne sadrži slijedni broj, već pokazuje gdje hitni podatci završavaju. Kada računalo primi segment s postavljenim URG bitom, treba dodati zadanu vrijednost pomaka koja piše u zastavici hitnosti na vrijednost slijednog broja trenutnog segmenta kako bi znalo koji je krajnji slijedni broj do kojega su podatci hitni, a to može biti neki bajt u nekom drugom segmentu;
  - **Zastavica potvrde: ACK** (engl. *Acknowledgement pointer*)– koristi li se, znači da je poslani segment s pripadajućim potvrdnim brojem potvrda prethodno primljenoga segmenta, npr. prilikom uspostave veze TCP-a. Praktički samo prvi segment nema ovu zastavicu jer se veza tek uspostavlja, dok svi ostali segmenti u komunikaciji imaju potvrdu u sebi za neki primljeni segment;
  - **Zastavica guranja: PUSH** – koristi li se ova zastavica, to znači da se podatci trebaju proslijediti aplikaciji što je prije moguće, bez zadržavanja u privremenoj memoriji (engl. *Buffer*). To se obično koristi kada se šalju male količine podataka koje imaju značenje u stvarnom vremenu, nekakva interakcija poput *Telnet* konekcije prema udaljenom poslužitelju;
  - **Zastavica resetiranja konekcije: RST** (engl. *Reset*) – ta zastavica signalizira prekid uspostave TCP veze. Uzroci mogu biti razni, a jedan od uobičajenih je pokušaj uspostave veze na broj TCP/UDP ulaza koji se ne koristi. Npr. na slikama 23., 24. i 25. broj TCP/UDP ulaza je 801, a on se ne koristi na odredištu. Tu činjenicu koriste i napadači na sustav kako bi saznali koji se TCP/UDP ulazi koriste, a koji ne;

Slika 23. Prikaz pokušaja uspostave veze na TCP/UDP ulaz koji se ne koristi na odredišnom računalu



Slika 24. Prikaz resetiranja veze u Wiresharku kada na odredištu nije otvoren određeni TCP/UDP ulaz, npr. 801



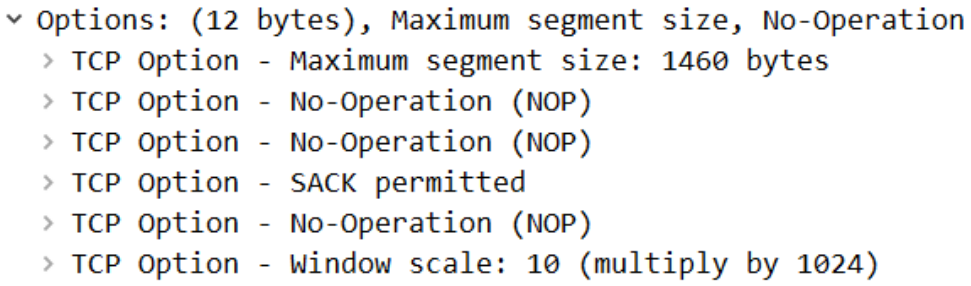
Slika 25. Detalji RST zastavice u *Wireshark* alatu

- **Zastavica sinkronizacije: SYN** (engl. *Synchronize*) – koristi se prilikom uspostave TCP veze kao dio mehanizma trostrukoga rukovanja. Ta zastavica signalizira uređajima koji uspostavljaju vezu da se u segmentima nalaze početni slijedni brojevi koje treba koristiti;
- **Zastavica završetka: FIN** (engl. *Finish*) – postavlja se kad uređaj želi prekinuti komunikaciju. To je znak da i druga strana također treba prekinuti komunikaciju slanjem zastavice potvrde: ACK da je primila zastavicu završetka: FIN, i nakon toga računalo koje je prvo poslalo FIN zastavicu, šalje završnu zastavicu potvrde: ACK, i komunikacija je prekinuta.

Kontrolni izračun (engl. *TCP Checksum*) koristi se za osnovno otkrivanje grešaka. Taj kontrolni izračun koji se radi nad TCP zaglavljem, podatcima u TCP segmentu i pseudo IP zaglavlju (koje se sastoji od 12 bajta podataka prikazanih na slici 26.) radi pošiljatelj koristeći određeni algoritam. Izračun se šalje u zaglavlju TCP segmenta. Primatelj zatim ponovno radi kontrolni izračun koristeći isti algoritam i uspoređuje te

dvije dobivene vrijednosti. Ako one nisu iste, taj segment se odbacuje. Iako na putu između izvora i odredišta može postojati mehanizam kao što je NAT, gdje se mogu mijenjati IP adrese ili TCP/UDP ulazi, uređaji na putu napravit će ponovni kontrolni izračun tako da komunikacija bude moguća. Pseudo IP zaglavlje koristi se samo za kontrolni izračun i nije dio podataka koji se šalju.

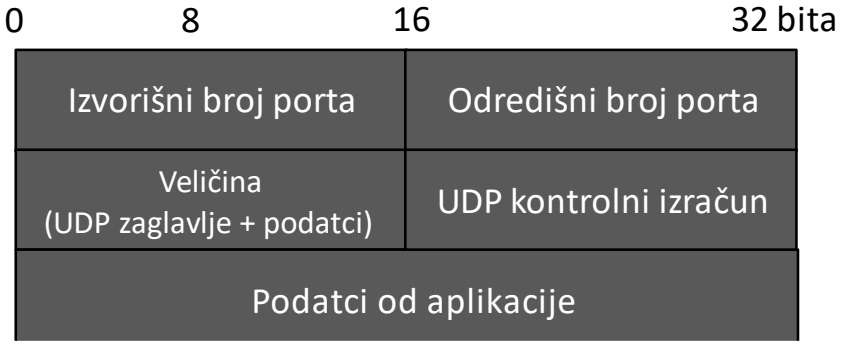
Polje mogućnosti (engl. *Options*) koristi se prilikom uspostavljanja TCP veze i sadrži do 40 bajtova podataka. Neke od opcija su: maksimalna veličina segmenta (MSS), veličina prozora, faktor skaliranja, selektivno potvrđivanje (engl. *Selective acknowledgement* – SACK), vremenska oznaka itd.



Slika 26. Izgled polja Opcija u Wireshark alatu

UDP

S druge strane, UDP se koristi kada je priroda aplikacije takva da nije nužno da svi podatci dođu na odredište i da budu predstavljeni aplikaciji točno određenim redom. UDP ne uspostavlja vezu, pa samim time nema ni mehanizama za njezinu uspostavu, održavanje ili raskidanje, kao ni sva polja koja za tu namjenu koristi TCP. UDP zaglavlje čini izvorišni i odredišni broj ulaza, polje s veličinom zaglavlja i kontrolni izračun za detekciju grešaka. Primjer aplikacije koja koristi UDP jest videoprijenos ili telefonski poziv, gdje nema smisla čekati da svi podatci dođu na odredište, već je ključno da oni dođu brzo, gotovo u stvarnom vremenu.



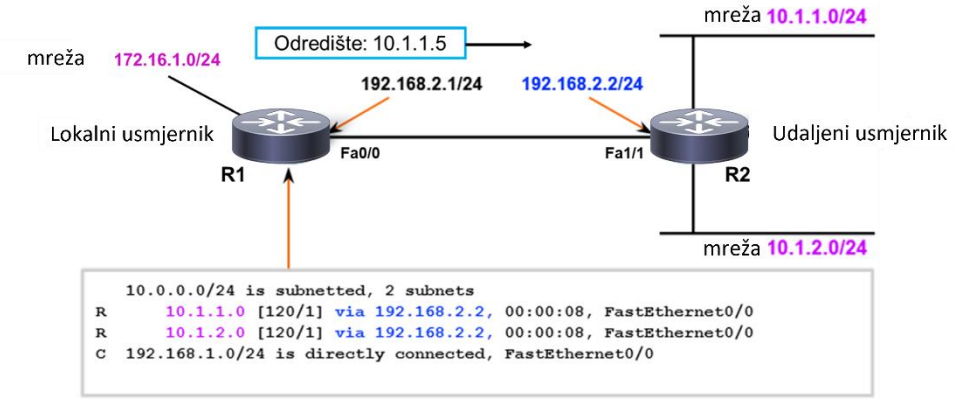
Slika 27. Zaglavlje UDP protokola

Aplikacije koje koriste UDP rade s pretpostavkom da je mrežna infrastruktura vrlo učinkovita i pouzdana, ali i da postoje mehanizmi upravljanja greškama u samoj aplikaciji (npr. *Buffering* za videoprijenos). Koristi se, nadalje, u aplikacijama koje periodički šalju određeni promet, i kada zbog prirode komunikacije, nije potrebna uspostava veze kao što je slučaj s DHCP i DNS servisima, gdje računala periodički šalju upite. U tim slučajevima ako računalo ne dobije odgovor, ono će jednostavno ponoviti upit, i tako dok ne dobije odgovor. Odabir se UDP protokola za neku aplikaciju temelji i na činjenici da je moguće, unatoč tome što neki podatci nisu došli na odredište, ipak razumjeti što netko govori u telefonsku slušalicu ili što je prikazano na televiziji, dok bi naknadno ubacivanje podataka koji nisu došli kad su trebali samo narušilo korisničko iskustvo i svrhu te komunikacije. Ne može se odrediti koji je od protokola bolji, TCP ili UDP, jer je svaki osmišljen s određenom namjenom, i tako ga treba sagledati.

1.2.8. Mrežni sloj (treći)

Glavna je uloga mrežnog sloja usmjeravanje prometa između uređaja i između podmreža na temelju logičkih adresa uređaja (IPv4 ili IPv6 adrese). To čine primarno usmjernici koristeći tablicu usmjeravanja. Nakon što je transportni sloj segmentirao podatke i upisao određeni broj TCP/UDP ulaza u za to predviđena polja u zaglavlju segmenta, sada ti segmenti trebaju biti opisani tako da svi uređaji u mreži koji prosljeđuju te podatke prema odredištu znaju kako to učiniti. Rezultat obrade segmenata na mrežnom sloju, gdje se dodaju razni opisni podatci specifični za mrežni sloj, poput izvorišne i odredišne IP adrese, zove se paket. IP adrese ključni su elementi u komunikaciji između uređaja u mreži. Na temelju IP adrese odredišta mrežni uređaji odlučuju kojim putem poslati paket mrežom. Taj se proces naziva usmjeravanje

paketa, i za tu namjenu uređaji koji se zovu usmjernici koriste protokole i tablice usmjeravanja kako bi pronašli najkraći put do odredišta. Još jedna ključna funkcionalnost mrežnog sloja prekidanje je beskonačnih petlji na temelju vrijednosti TTL polja. Kada paketima koji „zapnu“ u petlji između dva ili više uređaja i istekne im TTL vrijednost (spusti se na vrijednost nula), ti paketi bivaju uništeni i na taj način se mreža rasterećuje. TTL vrijednost smanjuje se svaki puta kada neki uređaj u mreži očita vrijednost TTL polja. Kada ne bi postojalo TTL polja, paketi do čijeg odredišta mrežni uređaji ne mogu doći, lako bi mogli vječno putovati mrežom i vremenom se nagomilati do te razine da bi mreža prestala raditi. Mogući je razlog zašto paketi ne stižu na odredište, što odredišta jednostavno nema ili trenutno odredišna mreža nije aktivna.



Slika 28. Ilustracija usmjernika i tablica usmjeravanja

Protokol IPv4

Na mrežnom sloju danas postoje dva protokola, IPv4 (*Internet Protocol version 4*) i IPv6 (*Internet Protocol version 6*). Za početak, usmjerenje će biti na IPv4, dok će IPv6 biti obrađen u posebnom poglavlju. Principi protokola IPv4 i IPv6 vrlo su slični, pa razumijevanje kako se promet prosljeđuje u IPv4 mrežama, olakšava usvajanje IPv6 protokola.

Osnovne karakteristike protokola IPv4 su:

- **Ne uspostavlja vezu** (engl. *Connectionless*) – to je zadaća transportnog sloja i ne bi bilo korisno da više slojeva brine o uspostavljanju veze ili upravljanju komunikacijom jer bi, između ostalog, zahtijevalo neku vrstu sinkronizacije i koordinacije između slojeva, a upravo je cilj OSI modela da svi slojevi budu neovisni. Uspostava veze zadaća je TCP protokola.

- **Princip „*Best Effort*“, tj. najbolja moguća usluga** – to znači da protokol radi kako radi i nema nikakvih dodatnih kontrolnih paketa ili polja koja bi jamčila isporuku paketa ili poseban tretman paketa u mreži u odnosu na druge pakete. Taj princip omogućava da IP protokol puno ne kasni u obradi podataka prilikom slanja ili primanja. Nauštrb pouzdanosti i drugih kompliciranih mehanizama koji postoje na transportnom sloju TCP protokola, u IP protokolu dobivena je brzina. Pouzdanost i upravljanje komunikacijom zadaća je TCP-a.
- **Neovisnost o mediju za prijenos podataka** – ta je karakteristika direktna posljedica ideje OSI modela, a to je međusobna neovisnost slojeva, pa tako IP adrese koristimo na računalima koja se povezuju žičnom mrežom, isto kao i na onima koja se povezuju optičkom ili bežično.

IP protokol, kao i svi drugi protokoli, ima svoju strukturu koja je predstavljena IP zaglavljem prikazanim na slici 29.

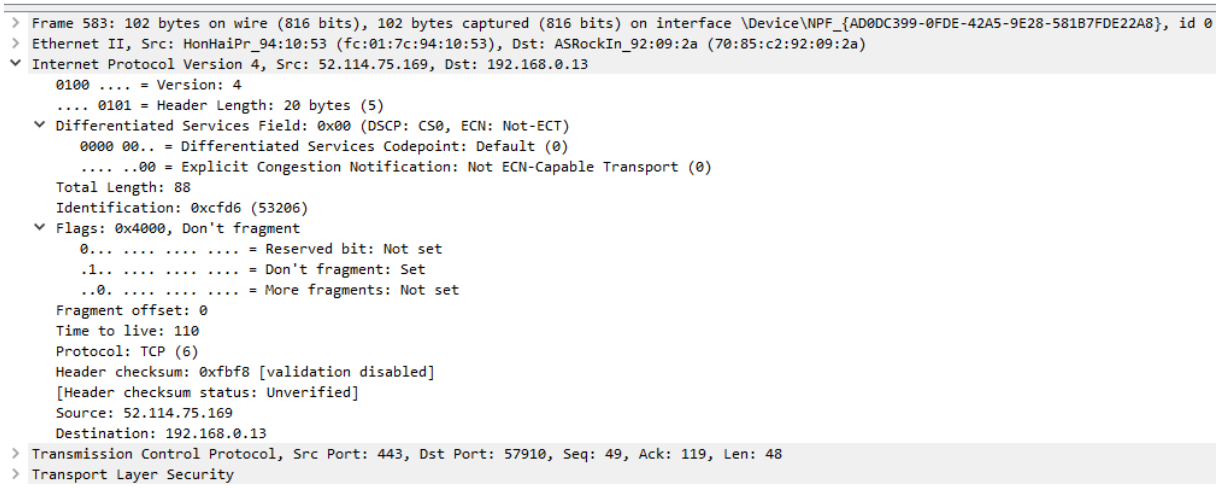
Slika 29. Prikaz IPv4 zaglavlja sa svim poljima

- **Verzija** (engl. *Version*) – to polje pokazuje o kojoj se verziji internetskog protokola radi, mogu biti IPv4 (oznaka 4) ili IPv6 (oznaka 6), samo kod IPv6 cijelo zaglavlje izgleda drugačije.
- **Veličina zaglavlja** (engl. *Internet Header Length)* – to polje pokazuje veličinu IPv4 zaglavlja kao broj 32-bitnih riječi. Minimalno je 5, što znači da je minimalna veličina IPv4 zaglavlja 20 bajtova, odnosno 160 bitova (5 × 32 × 4 = 160 bitova).

- **Vrsta usluge** (engl. *Type of Service*) – koristi se postoje li u mreži implementirani mehanizmi za upravljanje kvalitetom usluge. Tada se u ta polja upisuju određene vrijednosti na temelju kojih mehanizmi za kvalitetu usluge u mreži drugačije prosljeđuju IP pakete.
- **Ukupna duljina** (engl. *Total Length*) – odnosi se na IP paket, a uključuje veličinu polja podatci, veličinu zaglavlja transportnog sloja (TCP/UDP) i veličinu zaglavlja mrežnog sloja (IP zaglavlje).
- **Identifikacija** (engl. *Identification*) – to polje sadrži jedinstvenu oznaku za svaki paket. Koristi se ako se IP paketi fragmetiraju kada se šalju između uređaja u mreži te da bi se mogli svi fragmenti određenog IP paketa posložiti ispravnim redoslijedom u originalni paket kakav je bio prije fragmentiranja. Svi fragmenti jednog paketa imaju istu identifikaciju.
- **Zastavice** (engl. *Flags*) − koriste se tri zastavice:
    - Rezervirana (engl. *Reserved*) – jedini neiskorišteni bit u IP zaglavlju,
    - DF (engl. *Don't Fragment*) – upravlja fragmentacijom i jasno označava pakete koji se ne smiju fragmentirati. Preferirano je da se paketi smiju fragmentirati, i tada je zastavica 0, ako se ne smiju fragmentirati, tada je zastavica 1. Ako se paket ne smije fragmentirati, a trebao bi, taj se paket odbacuje.
    - MF (engl. *More Fragments*) – označava ima li još fragmenata originalnog paketa koji trebaju pristići. Ako nema, onda je zastavica 0, ako ima, onda je zastavica 1.
- **Oznaka fragmenta (engl. *Fragment Offset*)** – to se polje koristi prilikom sastavljanja fragmentiranih paketa. Označava početnu poziciju podataka u fragmentiranom paketu u odnosu na početak podataka u originalnom paketu.
- **Vrijeme života (engl. *Time to Live – TTL*)** – vrijednost je od 0 do 255. To polje određuje koliko daleko u mrežu neki paket može otići prije nego mu istekne životni vijek i bude odbačen. To je polje ključno u prekidanju usmjerničkih petlji. Svaki put kada neki L3 uređaj (obično usmjernik) obradi paket, vrijednost toga polja smanjuje se za 1.

- **Protokol (engl. *Protocol*)** – to polje označava koji je protokol enkapsuliran u IP paket (npr. 6 označava da paket nosi TCP segment, 17 označava da je u paketu UDP protokol ili 1 ako je ICMP).
- **Kontrolni izračun (engl. *Header Checksum*)** – u tom se polju nalazi kontrolni izračun koji se koristi kako bi se utvrdio integritet IP paketa. To znači da izvor i odredište nekog paketa rade kontrolni izračun koristeći isti algoritam i podudaraju li se vrijednosti, tada je paket ispravan. Ako su kontrolni izračuni različiti, taj se paket odbacuje. Kontrolni izračun se radi samo nad IP zaglavljem, zato što podatci u TCP segmentu ili ICMP paketu imaju svoju kontrolu.

Kada odredišna mreža nije dostupna iz bilo kojeg razloga, promet će se prosljeđivati na temelju predefinirane putanje, koju uglavnom svi usmjernici u mreži imaju za slučaj da dođe neki promet koji ne znaju kamo proslijediti. Činjenica da većina usmjernika umjesto da zna gdje se nalazi svaka moguća mreža, imaju predefiniranu putanju, nužna je želimo li imati optimalnu računalnu mrežu. U suprotnom, svaki usmjernik trebao bi imati stotine tisuća putanja u svojoj tablici usmjeravanja, što nikako nije izvedivo u praksi. Tema usmjeravanja više se odnosi na dizajn mreže, specifično dizajn usmjeravanja u mreži, pa će o tome biti nešto kasnije više riječi.



Slika 30. Izgled polja IPv4 zaglavlja u Wireshark alatu

Na slici 30. prikazan je IP paket u *Wireshark* alatu te se može vidjeti izgled svih polja u stvarnosti.

Internet Control Message Protocol – ICMP

Ranije je spomenuto kako IP (eng. *Internet Protocol*) nema nikakve mehanizme pouzdanosti komunikacije, već je njegov zadatak u što kraćem vremenu proslijediti pakete do odredišta (engl. *Best Effort*). U svrhu provjere jesu li podatci na odredište stigli, koristi se ICMP protokol, kao pomoćni protokol za IP. Kako bi računala koja komuniciraju dobila povratnu informaciju o tome jesu li paketi stigli na odredište, je li odredišna podmreža uopće dostupna, znaju li posrednički uređaji u mreži kako doći do odredišne podmreže, je li istekao životni vijek paketu i slično, koristi se ICMP. Važnost je ICMP-a velika jer ima zadatak slanja kontrolnih poruka i poruka o greškama i zbog toga se često koristi u dijagnostičke svrhe. Ako računalo A pošalje paket računalu B, i taj paket zbog neispravnog sučelja na usmjerniku C ne može biti proslijeđen računalu B, usmjernik C će koristiti ICMP da bi poslao poruku računalu A da paket nije stigao na odredište. Treba naglasiti da ICMP ne ispravlja pogreške, već samo šalje poruke o njima.

Budući da se zna samo adresa ishodišnog uređaja, a ne i put kojim je paket prošao do cilja, ICMP poruka se šalje samo ishodišnom uređaju.

Zbog toga što su ICMP poruke enkapsulirane u IP zaglavlje, može doći do pogreške u prijenosu paketa s ICMP porukom, kao i svakog drugog paketa. Kako ne bi ICMP poruke javljale pogreške o slanju ICMP poruka, i da ne bi došlo do eventualnog zagušenja porukama, pogreške pri prijenosu ICMP poruka ne javljaju se novim ICMP porukama. To otvara mogućnost da neke pogreške pri prijenosu paketa nikada ne budu dojavljene. Na slici 31. prikazano je zaglavlje ICMP poruke.

Slika 31. Zaglavlje ICMP poruke

- **Tip (engl.*Type*)** – tip ICMP poruke
- **Kod (engl. *Code*)** – specifični podtip ICMP poruke

- **Kontrolni izračun (engl. *Checksum*)** – polje za provjeru pogrešaka
- **ID** – sadrži ID vrijednost, koristi se radi lakšeg mapiranja upita i odgovora
- **Slijedna vrijednost (engl. *Sequence*)** – sadrži slijednu vrijednost ICMP paketa (da bi se mogao mapirati upit i odgovor).

TIPOVI ICMP PORUKA

Postoji mnogo različitih ICMP poruka, a neke od uobičajenih prikazane su na slici 32.

Tipovi ICMP poruka	
0	Echo Reply - odgovor na ping
3	Destination Unreachable - kada usmjernik ne zna doći do odredišne podmreže
5	Redirect/Change Request - obavijest o boljoj putanji kroz mrežu
8	Echo Request - ping
11	Time Exceeded - kada istekne TTL vrijednost nekom paketu - tracert naredba

Slika 32. Tipovi ICMP poruka

Svaki tip poruke dijeli se na podtipove u polju koda (engl. *Code*) koji daju više detalja prilikom analize mrežnog prometa.

Sve te poruke računalo koristi kako bi zaključilo što se događa s IP paketima u mreži. Tako na primjer, kada računalo dobije ICMP odgovor u kojem je poruka tip 3 i kod 4, to je znak da bi se paketi na mrežnom sloju trebali fragmentirati želimo li da prođu mrežom. Primjer takve situacije je kada mrežom preklopnika želimo propustiti IP videoprijenos s velikim paketima, a mrežni preklopnici nisu za to konfigurirani. Dok mrežne preklopnike ne konfiguriramo za prijenos velikih paketa, svi će paketi biti odbačeni, a računalo koje je izvor IP videoprijenosa, dobivat će takve odgovore.

Za razliku od poruka pogrešaka, kontrolne poruke nisu posljedica pogrešaka pri prijenosu paketa, već se koriste da bi se uređaji obavijestili o određenim događanjima u mreži, kao što su zagušenja u mreži ili postojanje boljeg izlaza iz mreže.

NAREDBA PING

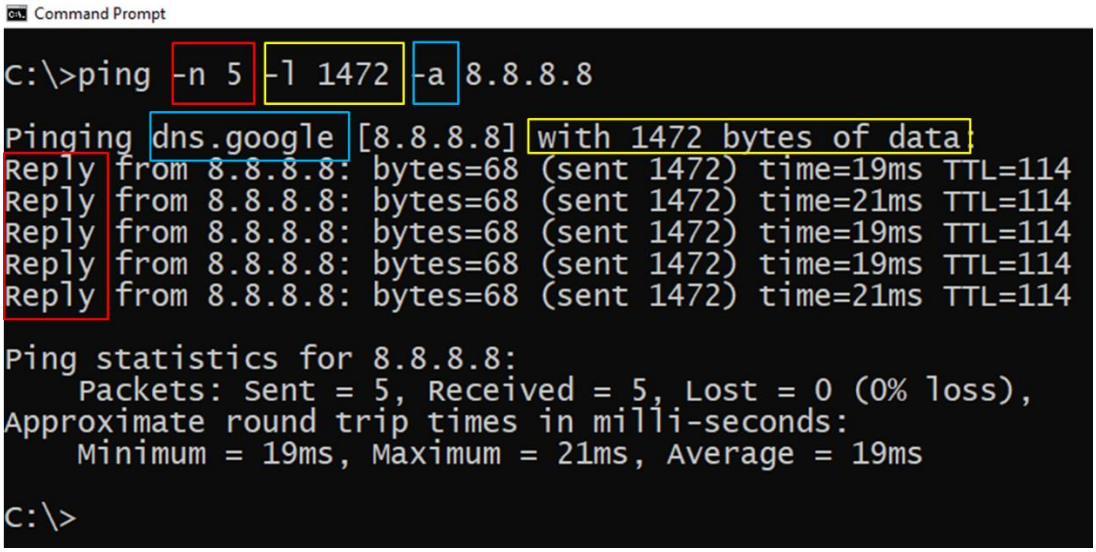
Naredba *ping* (engl. *Packet Internet Grouper*) osnovni je alat za provjeru postojanja veze između dva uređaja u mreži i rješavanje problema ako s vezom nešto nije u redu. Ona šalje ICMP poruke – *echo request* (tip 8) do odredišnog uređaja. Ako je veza ispravna, odredišni uređaj će primiti ICMP poruke i na njih odgovoriti porukom – *echo reply* (tip 0). *Ping* paket obično sadrži 32 bajta podataka, ali to se može mijenjati. Ako

uređaj koji šalje zahtjev primi odgovor u predefiniranom vremenu (4 sekunde na *Windows* računalima), veza između dva uređaja je ispravna. To znači da su svi mrežni uređaji između ishodišnoga i odredišnog uređaja ispravni i ispravno konfigurirani. U suprotnom na računalu će se ispisati poruka „*Request timed out*“. Prilikom korištenja naredbe *ping* svakom se ICMP paketu dodjeljuje broj te se tako može doznati koji se paket vratio ili je izgubljen. Također se svakom ICMP paketu dodaje vremenska oznaka (engl. *Timestamp*) pomoću koje se može izračunati koliko je vremena trebalo paketu da se vrati (engl. *Round Trip Time – RTT*).

Osnovna je sintaksa te naredbe *ping* IP adresa, npr. Ping 192.168.0.254. Za promjenu osnovne sintakse naredbe *ping* koriste se dodatne opcije koje variraju ovisno o operacijskom sustavu.

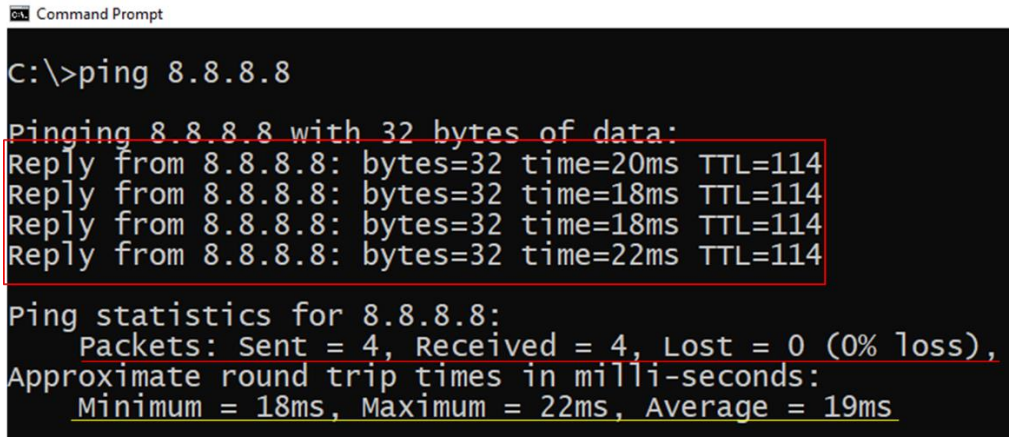
Ping -n 7 178.79.149.215 – šaljemo 7 ICMP request poruka (predefinirano je 4)
Ping -a 178.79.149.215 – šaljemo ICMP request poruku uz obrnuti DNS upit (eng. Reverse DNS Lookup)
Ping -t 178.79.149.215 – šaljemo ICMP request poruke neprestano (CTRL+C prekida slanje poruka)
Ping -l 1500 178.79.149.21 – šaljemo ICMP request poruke veličine 1500 bajta
Ping -w 10 178.79.149.215 – šaljemo ICMP request poruku s vremenom čekanja na odgovor od 10 sekundi

Slika 33. Varijacije naredbe ping



Slika 34. Prikaz složenijeg ICMP upita

Ako je veza između dva uređaja u mreži ispravna i odredišni uređaj je aktivan, spojen na računalnu mrežu, ispravno konfiguriran i nema zapreka u odgovaranju na upite (npr. vatrozid), dobije se poruka na slici 35.



Slika 35. Echo response poruka

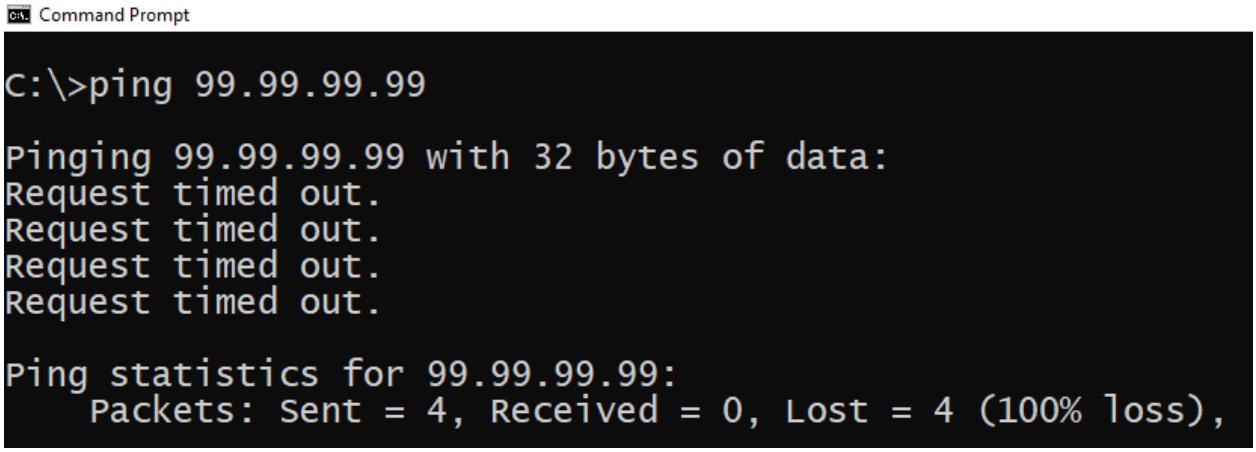
Na prethodnoj slici u odgovoru su sljedeći parametri:

- **bytes = 32** – poslana su i vratila su se četiri paketa veličine 32 bajta
- **time** – pokazuje vrijeme potrebno da ICMP paketi dođu do odredišta i da se vrate. To je zapravo vrijeme odziva koje se naziva RTT (engl. *Round Trip Time*).
- **TTL** (engl. *Time to Live*) – parametar koji određuje životni vijek paketa, odnosno određuje kroz koliko podmreža (usmjernika) može paket proći prije nego što će biti uništen. TTL je mehanizam „čišćenja“ računalne mreže od paketa koji ne mogu naći odredište i počinju beskonačno kružiti mrežom. Da takvi paketi ne bi vječno kružili mrežom i gomilali se, svaki paket ima životni vijek u kojemu mora stići do odredišta.

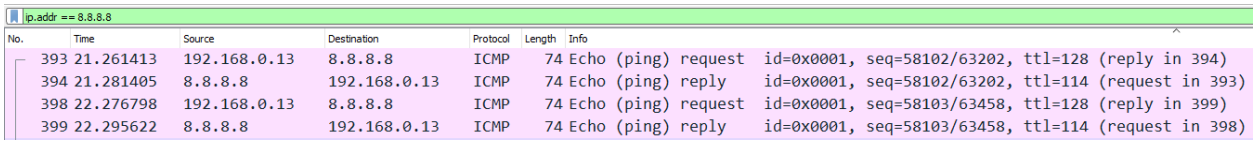
Na kraju je izvještaja statistika koja govori:

- koliko je paketa poslano i koliko ih se vratilo,
- najkraće i najduže vrijeme puta paketa do odredišta i nazad,
- ukupno srednje vrijeme.

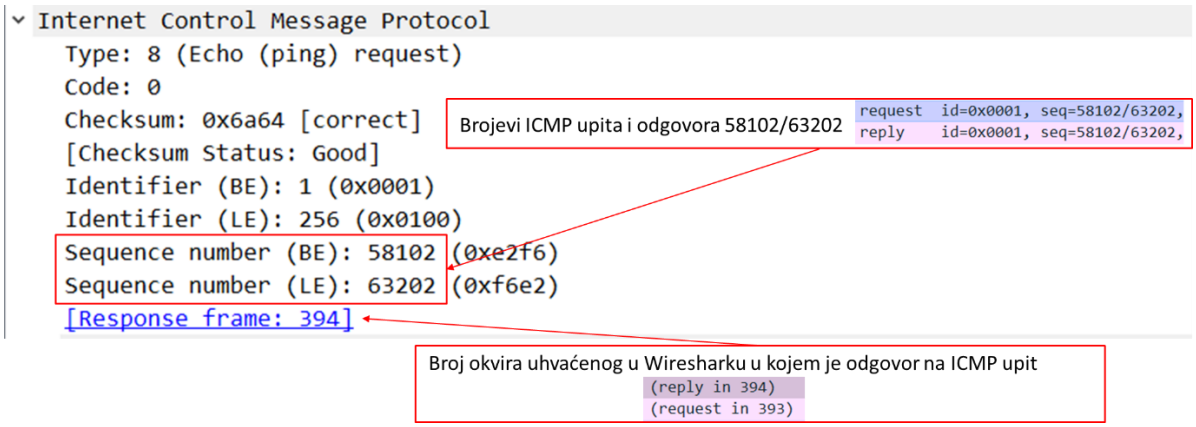
Ako veza između dva uređaja u mreži nije ispravna, dobije se poruka na slici 36.



Slika 36. Poruka koja se dobije ako se u predviđenom vremenu poslani paketi ne vrate



Slika 37. ICMP poruke uhvaćene u Wireshark alatu



Slika 38. Detalji ICMP poruke u Wireshark alatu

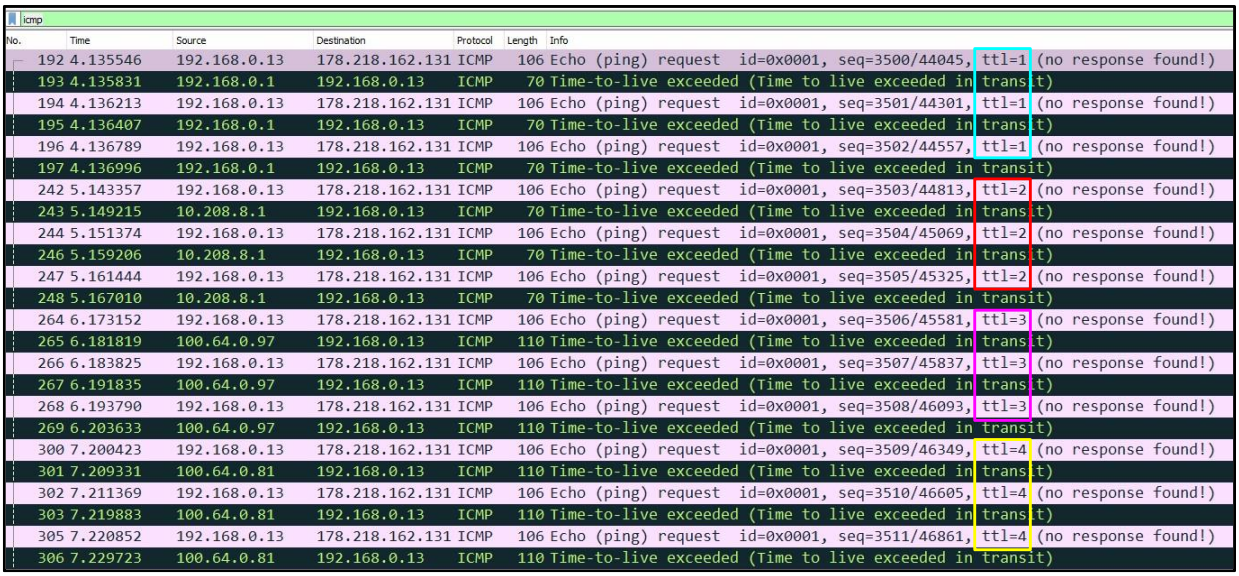
NAREDBA *TRACEROUTE*

Naredbom „traceroute“ ili njezinom varijacijom, ovisno o operacijskom sustavu (u *Windowsu* „tracert“), dobije se informacija o uređajima kojima paket prolazi od ishodišta do odredišta. Koristi iste tipove ICMP paketa kao i *ping*, samo na nešto drugačiji način.

Naredbom „traceroute“ šalje se niz od tri ICMP paketa koji imaju IP adresu odredišnog uređaja. Tri se paketa šalju kako bi se mogla izračunati prosječna RTT vrijednost. Najprije se šalje niz ICMP paketa s vrijednošću TTL parametra 1. Prvi usmjernik na

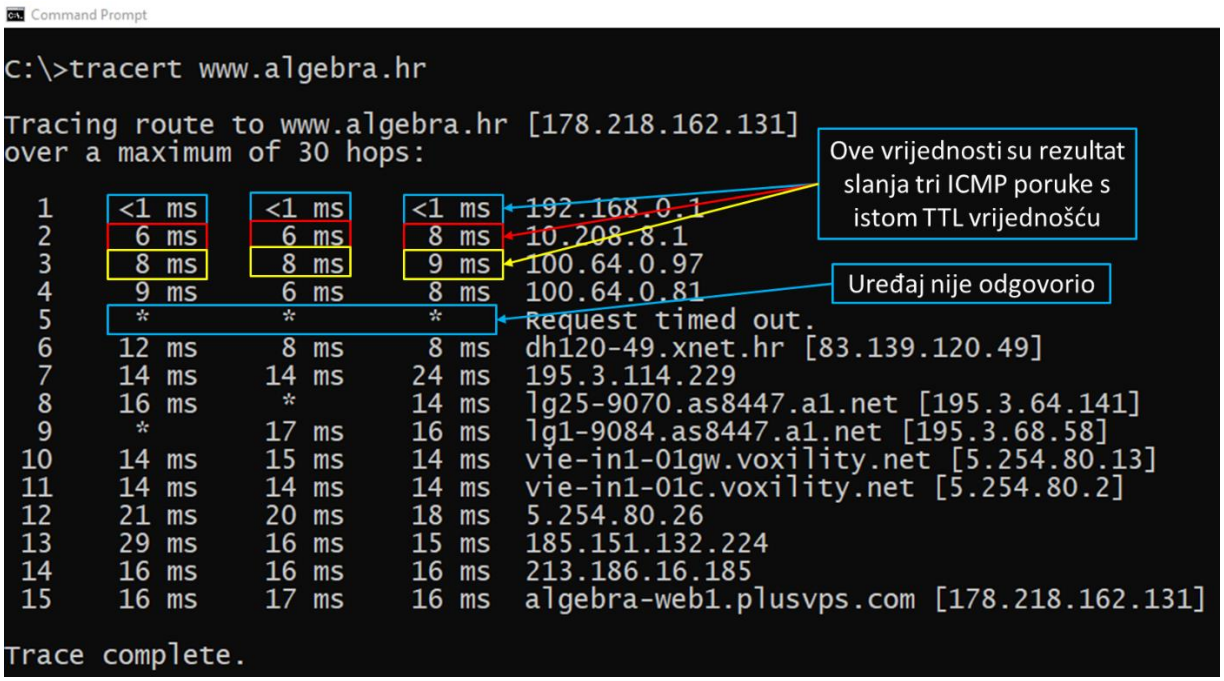
koji ICMP paket dođe, smanjit će TTL vrijednost za 1. Budući da će TTL vrijednost postati 0, paket će biti uništen i bit će vraćena ICMP poruka pogreške: „Time to live exceeded in transit“. Sljedeći će niz paketa imati postavljenu TTL vrijednost u 2. U tom će slučaju TTL vrijednost doći na 0 na drugom usmjerniku. Drugi usmjernik će također uništiti paket i vratiti poruku pogreške: „Time to live exceeded in transit“. Taj se proces nastavlja (ICMP paketi se šalju s povećavanjem TTL parametra) sve dok paket ne stigne na odredište i dok odredište ne pošalje ICMP poruku.

Na taj se način dobiju, jedna po jedna, poruke od svih usmjernika kojima paket prolazi do odredišta, a na računalu se vidi popis IP adresa svih uređaja koji su u nekom trenutku odbacili ICMP poruke i poslali odgovor: „Time to live exceeded in transit“. Tako se može utvrditi kojim putem promet između izvorišnoga i odredišnog uređaja ide mrežom.



Slika 39. Prikaz povećavanja TTL vrijednosti svaka tri paketa

Dakle, „traceroute“ koristi ICMP poruke za stvaranje popisa uređaja kroz koje paketi prolaze do odredišta. „Traceroute“ šalje tri puta uzastopno ICMP pakete do svakog usmjernika na putu do odredišnog uređaja i daje informaciju o RTT-u (engl. *Round Trip Time*), odnosno o vremenu odziva za svakoga od njih. Osnovna sintaksa naredbe „tracert“: ***tracert IP adresa***, npr. ***tracert* 178.218.162.131**



Slika 40 Prikaz ispisa tracert naredbe na Windows računalu

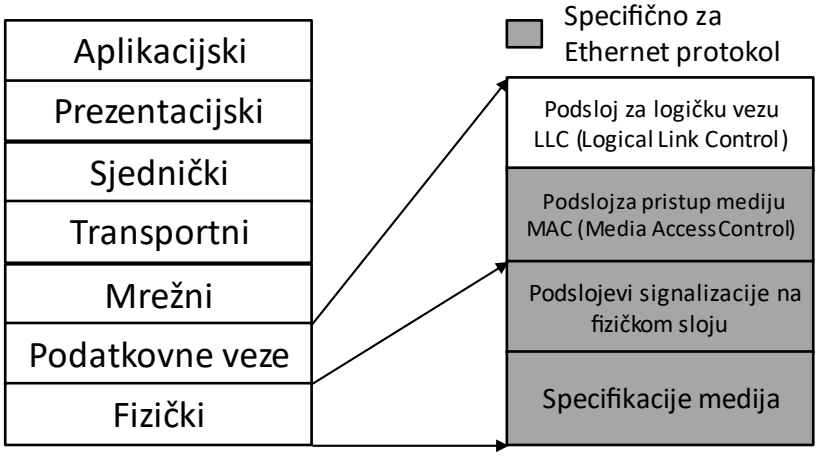
Naredba „tracert“ korisna je prilikom rješavanja problema u mreži jer se pomoću nje može utvrditi ide li promet optimalnim putem kroz mrežu, postoje li usmjerničke petlje ili na kojem je dijelu mreže (na kojem usmjerniku) prekinuta veza i gdje treba tražiti rješenje problema.

1.2.9. Sloj podatkovne veze (drugi)

Uloga je sloja podatkovne veze u komunikaciji priprema podataka za slanje na fizički medij za pouzdan prijenos podataka vezom (engl. *Link*) između dva uređaja u mreži. Sloj podatkovne veze zadužen je za pristup mediju za prijenos podataka i upravlja komunikacijom na dijeljenim medijima kojima pristupa više uređaja istovremeno (danas je to najočitije u bežičnim mrežama). Također bitna funkcionalnost sloja podatkovne veze provjera je događaju li se greške u prijenosu podatka kako bi odbacio sve okvire koji su zbog bilo kojeg razloga promijenjeni u slanju. Za tu namjenu koristi polje FCS (engl. *Frame Check Sequence*) u završetku okvira u kojem se nalazi oznaka (engl. *Hash*) izračuna okvira prilikom slanja koji se uspoređuje s ponovnim izračunom oznake (*hash*) vrijednosti okvira na odredištu. Ako izračunata vrijednost nije ista, taj se okvir odbacuje. Primjeri primjene toga sloja u stvarnom svijetu su *Ethernet*, *Wi-Fi*, *Bluetooth* i slične tehnologije, koje još zovemo i L2 (*Layer 2*) tehnologije.

ETHERNET

Ethernet je jedna od najstarijih i najkorištenijih tehnologija u računalnim mrežama i prihvaćena je kao opći standard lokalnih računalnih mreža. Poznat je još i kao IEEE (engl. *Institute of Electrical and Electronics Engineers*) 802.3 standard. Najviše se koristi u LAN mrežama koje su sveprisutne i gdje god postoje krajnji uređaji koji se povezuju na računalnu mrežu, gotovo uvijek se koristi *Ethernet* protokol. On radi na prva dva sloja OSI modela (Fizički sloj podatkovne veze).



Slika 41. Ethernet protokol u kontekstu OSI modela

Sve što je označeno sivo na slici 41. specifično je za *Ethernet*, dok je LLC podsloj isti za sve LAN tehnologije, iako je danas *Ethernet* prevladavajuća tehnologija LAN mreža. Ispod MAC podsloja *Ethernet* protokol bit će drugačiji ovisno o vrsti i tipu medija. *Ethernet* protokol je doživio mnoga unapređenja kako bi različiti proizvođači opreme bili kompatibilni i kako bi postigli veće brzine i efikasniju komunikaciju fizičkim medijima koji su se također vremenom razvijali.

Danas postoje različite implementacije *Ethernet* protokola u primjeni, a neke su od njih:

- *Ethernet* (IEEE 802.3): originalni standard razvijen 1970-ih, definira brzine do 10 Mbps.
- *Fast Ethernet* (IEEE 802.3u): standard predstavljen 1990-ih, definira brzine do 100 Mbps.
- *Gigabit Ethernet* (IEEE 802.3ab): standard predstavljen kasnih 1990-ih, definira brzine do 1000 Mbps, odnosno 1 Gbps. Danas najčešće korišten u LAN mrežama za povezivanje krajnjih uređaja.

- 10 *Gigabit Ethernet* (IEEE 802.3ae): definira brzine do 10 Gbps. Obično se koristi u okruženjima podatkovnih centara ili velikim mrežama za povezivanje mrežnih uređaja.
- 40 *and* 100 *Gigabit Ethernet* (IEEE 802.3ba): najnovije verzije *Etherneta*, omogućavaju brzine prijenosa podataka do 40 Gbps ili 100 Gbps. Primarno se koriste u okruženjima podatkovnih centara i kao dio infrastrukture računarstva u oblaku.

Ethernet protokol koristi razne mehanizme prilikom slanja i primanja podataka na medij za prijenos podataka. Jedan od najpoznatijih je tzv. CSMA/CD (engl. *Carrier Sense Multiple Access / Collision Detection*), nužan u mrežama koje su imale dvosmjernu, ali neistovremenu komunikaciju, tj. radile u *Half-duplex* režimu (slično kao što radi *Walkie Talkie* gdje samo jedna strana može slati podatke, a druga mora čekati svoj red za slanje), poput 10BASE5 ili 10BASE2. Danas takvih okruženja gotovo nema, već je svugdje dvosmjerna istovremena komunikacija, tj. *Full-duplex* komunikacija, što znači da sva računala mogu istovremeno i slati i primati podatke bez čekanja i bez primjene CSMA/CD mehanizma. Ono što danas postoji su bežične LAN mreže u kojima se komunicira dvosmjerno neistovremeno, ali u takvim okruženjima se koristi drugačiji mehanizam naziva CSMA/CA (engl. *Carrier Sense Multiple Access / Collision Avoidance*), a o tome više u poglavlju o bežičnim LAN mrežama (eng. *Wireless LAN – WLAN*). U početcima razvoja mreža bitan su koncept bile kolizijske domene, upravo zato što su sva računala zaista dijelila isti fizički medij (isti kabel i iste žice za odašiljanje i primanje signala na fizičkom mediju), a jedini su mrežni uređaji bili HUB uređaji koji su samo povezivali više računala na istu žicu i tada su se kolizije signala zaista događale. Danas je to prošlost pa pojam dijeljeni medij više nema isto značenje kao prije. Tema kolizijskih domena nije aktualna, niti se spominje u ovom priručniku.

Ethernet okvir (engl. *Ethernet Frame*)

7 bajta	1	6	6	2	46 do 1500	4
Preambula	SDF	Odredišna MAC adresa	Izvorišna MAC adresa	Veličina /Tip	Zaglavlje i podatci	FCS

Slika 42. Ethernet okvir

- Preambula i Početak okvira (engl. *Start of Frame Delimiter – SDF*) – koriste se za sinkronizaciju komunikacije kako bi se pripremio odredišni uređaj koji prima podatke
- Odredišna i izvorišna MAC adresa (engl. *Destination and Source Address*) – ključna su polja u upravljanju komunikacijom u računalnoj mreži, postavljanjem sigurnosnih mehanizama i slično. U tim se poljima nalaze MAC adrese uređaja koji komuniciraju neposredno u LAN mreži
- Veličina/Tip (engl. *Length/Type*) definira veličinu podatkovnog polja koje se prenosi u okviru. Koristi se zajedno s FCS poljem za provjeru ispravnosti primljenog okvira ili definiranje L3 protokola koji se nosi enkapsuliran u okvir. Koristi se ili jedno ili drugo (ovisno o standardu)
- Podatci (engl. *Data*) i 802.2 zaglavlje (ako se koristi) – svi okviri moraju biti minimalne veličine 64 bajta, u suprotnom se dodaju nule u polje podataka (eng. *Padding*)
- FCS polje za provjeru ispravnosti primljenog okvira – u tom se polju nalazi kontrolni izračun tzv. *Cyclic Redundancy Check – CRC*.

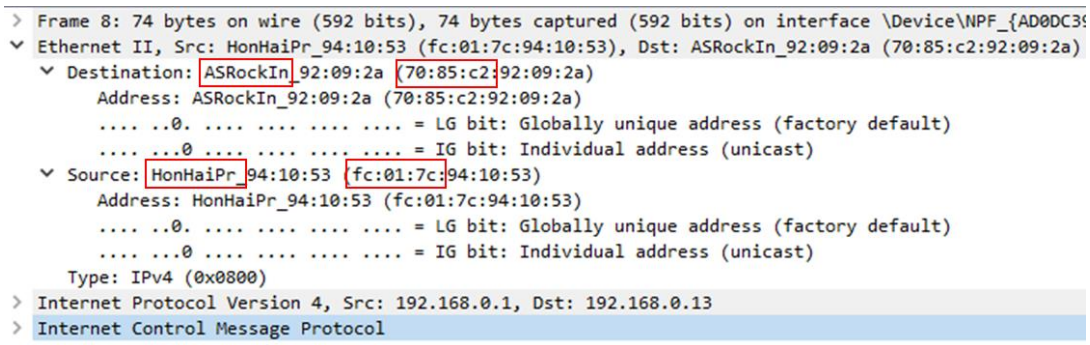
MAC adrese su mrežnim karticama dodijeljene prilikom proizvodnje. Iako je u svijetu računala sve binarno, puno puta se ti nizovi prikazuju u nekom drugom zapisu prilagođenom ljudima, pa tako su MAC adrese, koje su zapravo binarni zapis od 48 bitova, predstavljene kao heksadecimalni zapis od 12 znamenki (od 0 do 9 i od A do F). Na slici 43. primjer je MAC adrese stolnog računala.

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Realtek PCIe GbE Family Controller
   Physical Address. . . . . . . . . : 70-85-C2-92-09-2A
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 192.168.0.13(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 11. veljače 2023. 13:44:06
   Lease Expires . . . . . . . . . . : 11. veljače 2023. 21:44:19
   Default Gateway . . . . . . . . . : 192.168.0.1
   DHCP Server . . . . . . . . . . . : 192.168.0.1
   DNS Servers . . . . . . . . . . . : 83.139.103.3
                                       83.139.121.8
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Slika 43. Ispis postavki mrežne kartice računala

MAC adresa sastoji se od dva dijela. Prvih šest znamenki predstavlja proizvođača mrežne kartice (eng. *Organizatonal Unique Identifier – OUI*), a drugih šest znamenki

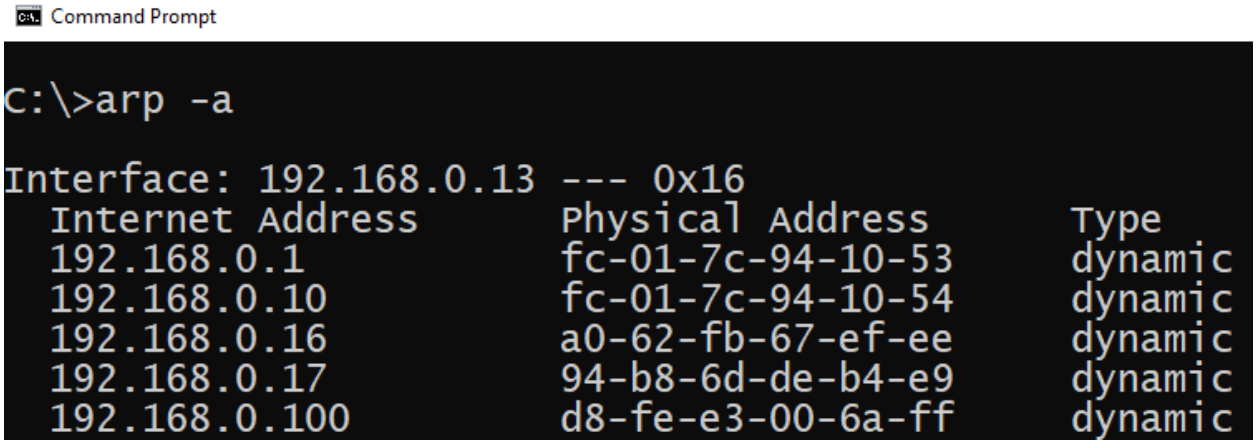
predstavlja konkretnu mrežnu karticu. Kao što se može vidjeti na slici 44., *Wireshark* već ima u svojoj bazi popis proizvođača mrežnih kartica.



Slika 44. Prikaz Ethernet okvira u Wireshark alatu — MAC adrese

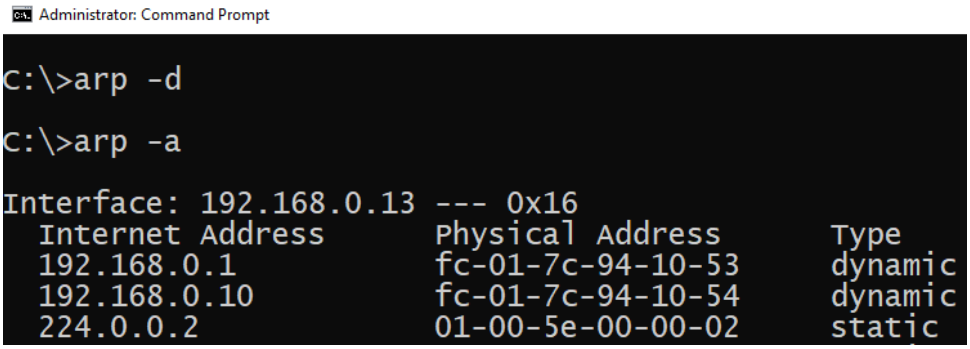
ARP

Da bi uređaji u lokalnoj računalnoj mreži mogli međusobno komunicirati, nužno je da nekako saznaju MAC adrese. U tu se svrhu koristi ARP (eng. *Address Resolution Protocol*). ARP je mehanizam koji računalu omogućava na temelju poznate L3 adrese (IPv4 adresa) otkriti nepoznatu L2 adresu (MAC adresa). Taj proces otkrivanja MAC adrese uređaja u LAN mreži počinje tako da npr. želimo poslati *ping* na IP adresu 192.168.0.100. U tom trenutku računalo će pregledati svoju ARP tablicu. Postoji li u ARP tablici zapis koji povezuje IP i MAC adresu, računalo zna kome poslati *ping*. Međutim, ako tog zapisa nema, onda računalo mora poslati ARP upit kako bi saznalo koju MAC adresu ima računalo s poznatom IP adresom na koju se *ping* šalje. Naredbom ARP-a dobiva se ispis svih MAC adresa koje je lokalno računalo naučilo u ARP protokolu.



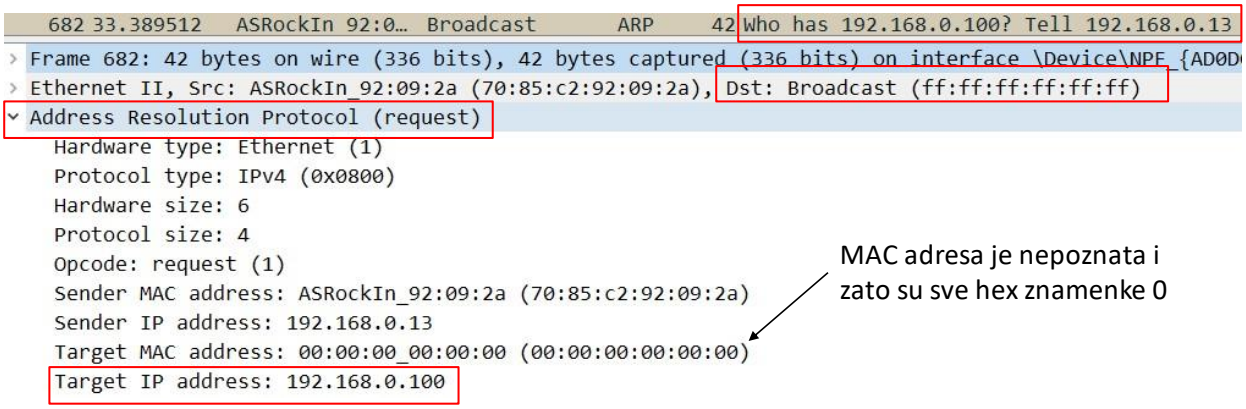
Slika 45. ARP tablica u kojoj postoji zapis za IP adresu 192.168.0.100

Nakon brisanja ARP tablice računala naredbom: „arp-d“ u CMD prozoru pokrenutom administratorski, računalo više nema taj zapis i morat će poslati ARP poruku kako bi ponovno stvorilo potreban zapis u ARP tablici.



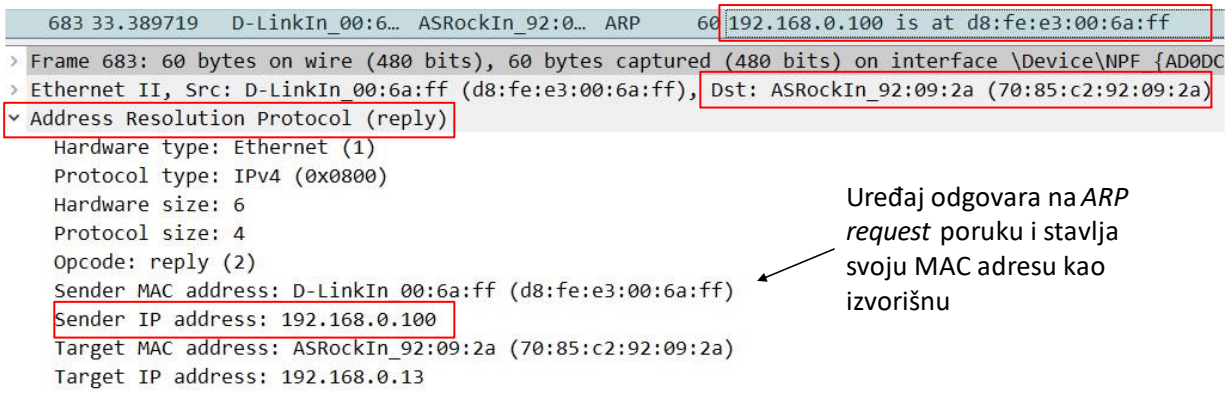
Slika 46. Nakon brisanja ARP tablice računalo više nema potreban zapis

Računalo prije bilo kakve komunikacije prema IP adresi 192.168.0.100 mora saznati MAC adresu odredišnog uređaja i zato će poslati ARP upit svim uređajima u LAN mreži, za što koristi sveodredišnu (engl. *Broadcast*) MAC adresu (ff:ff:ff:ff:ff:ff).



Slika 47. ARP zahtjev s računala 192.168.0.13 koje traži MAC adresu za poznatu IP adresu 192.168.0.100

Tek ako postoji uređaj s IP adresom 192.168.0.100, taj uređaj će odgovoriti porukom ARP odgovora u kojoj će poslati svoju MAC adresu pošiljatelju. U ovom slučaju računalo s IP adresom 192.168.0.13 poslalo je ARP zahtjev.



Slika 48. Odgovor na ARP zahtjev

Nakon što je uređaj 192.168.0.13 dobio odgovor na ARP zahtjev, može dodati novi zapis u svoju ARP tablicu. Bitno je znati da će neki uređaj ARP upit slati samo ako zaključi kako je odredište u njegovoj lokalnoj računalnoj mreži. Zaključi li da je odredište izvan lokalne računalne mreže, tada promet šalje posredstvom *Gateway* uređaja (obično usmjernik). Ako za nj nema zapis u svojoj ARP tablici, mora poslati ARP upit, kao što je prethodno opisano, i naučiti MAC adresu *Gateway* uređaja kako bi mogao njemu poslati promet koji ide izvan lokalne računalne mreže.

1.2.10. Fizički sloj (prvi)

Fizički sloj jednim je dijelom ono što možemo opipati rukama i izmjeriti mjernim instrumentima. Uloga je fizičkog sloja primarno povezivanje uređaja i odašiljanje podataka u binarnom obliku kodiranom u signal za slanje na medij za prijenos podataka (npr. električni napon u bakrenim kabelima ili svjetlosni impulsi u optičkim nitima). Fizički sloj podrazumijeva konektore, kabele, optičke niti, prigušnice svjetlosnog signala, frekvencije, valne duljine, razine napona itd. Može se govoriti o standardima i karakteristikama mrežnih uređaja poput mehaničkih svojstava, karakteristika medija za prijenos podataka (dimenzije, materijali, izvedba), funkcionalnim specifikacijama (razine napona, signalizacija, modulacija, kodiranje signala) i slično. Uz fizički sloj vežu se i standardi prilikom implementacije određenih medija za prijenos podataka, što podrazumijeva razna mjerenja, načine implementacije pojedinog medija (npr. optičkih niti), izradu prespojnih panela, izradu konektora (npr. raspored boje u UTP kabelu) itd. Stručnjake za mreže jedino zanimaju, prilikom implementacije, održavanja ili otkrivanja uzroka problema u radu, vrste kabela i jesu li oni ispravno korišteni, ne i karakteristike samog signala na žici ili optičkoj niti

jer je to previše kompleksno, a gotovo uvijek radi kako je zamišljeno. Prilikom sumnje u problem na fizičkom sloju, npr. neispravan kabel ili neispravno zavarene odnosno spojene optičke niti (engl. *Spliced*), neće to sami pokušati popraviti.

Trošak koji bi nastao pokušajem popravljanja, npr. mrežne kartice ako bi se nakon njezina testiranja u proizvodnji dogodilo da neispravno radi modulaciju signala, daleko premašuje razinu opravdanosti. Tako da se u velikom broju slučajeva jednostavno zamijeni kabel, uređaj ili komad opreme. Osim toga, tu je i pitanje jamstva i podrške za različite komponente koje se u slučaju pokušaja neovlaštenog popravljanja, gubi. Naravno, treba donijeti ispravnu odluku u kojem trenutku zamijeniti kabel, i koji dio kabela, tako da prilikom uspostave nove optičke mrežne infrastrukture, i nakon što je optički kabel neispravno povezan na prespojnom panelu, neće se mijenjati cijeli kabel, već treba ponoviti postupak povezivanja s više pažnje u prikladnijim uvjetima (temperatura, vlažnost, prašina itd.) i ponovno mjeriti gubitke.

1.2.11. Vrste medija za prijenos podataka

Danas su u Hrvatskoj osnovni mediji za prijenos podataka žičani bakreni (npr. UTP) i optički kabeli. Koristi se u manjoj mjeri i koaksijalne kabele. UTP kabeli (eng. *Unshielded Twisted Pair*) imaju četiri para žica koje su međusobno usukane kako bi se smanjile elektromagnetske smetnje. UTP kabel u osnovnoj izvedbi najčešće se koristi u lokalnoj računalnoj mreži za povezivanje krajnjih uređaja na mrežu (npr. mreža učionice). S druge strane, optički kabeli koriste se za povezivanje mrežnih uređaja u zgradi (npr. veza između dva mrežna ormara na različitim katovima zgrade) ili između zgrada u gradskim četvrtima, za povezivanje zgrade s telekomunikacijskom mrežom operatora i slično. Međutim, najveći značaj optičkih kabela je u povezivanju uređaja na velike udaljenosti, npr. između gradova, država, pa čak i kontinenata.

Žičani bakreni kabeli kategorizirani su prema izvedbi zaštite usukanih parova žica unutar kabela, te se razlikuju:

- U/UTP kabel (engl. *Unshielded with Unshielded Twisted Pair*) – najčešće korišten u mrežama; sastoji se od četiri para međusobno usukanih bakrenih žica koje su izolirane, ali nisu zaštićene od elektromagnetskih smetnji.

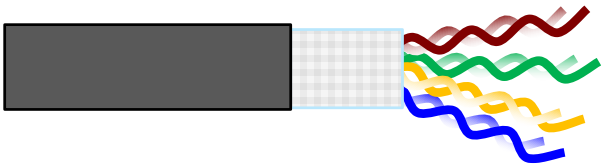
Slika 49. U/UTP kabel (eng. Unshielded with Unshielded Twisted Pair)

- F/UTP kabel (engl. *Foiled with Unshielded Twisted Pair*) – to je UTP kabel s metalnom folijom oko svih žica zajedno u kabelu i žicom za uzemljenje, koja kada se pravilno poveže, znatno smanjuje posljedice elektromagnetskih smetnji okoline, ali nema zaštitu pojedinačnih parova žica.

Te se verzije dalje mogu razlikovati ovisno o zaštiti koja je implementirana na razini para žica, na razini svih žica i na razini samog kabela.

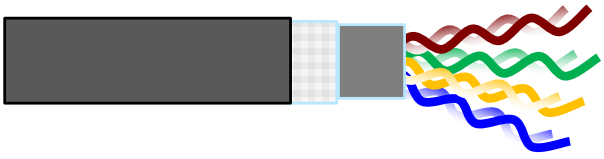
Slika 50. F/UTP kabel (Eng. Foiled with Unshielded Twisted Pair)

- S/UTP kabel (engl. *Shielded with Unshielded Twisted Pairs*) – to je kabel koji ima ožičenje oko svih žica, dok su parovi žica nezaštićeni. Pogodan je za veće brzine i veće udaljenosti nego UTP kabel, a ima i veću mehaničku čvrstoću.



Slika 51. S/UTP kabel (eng. Shielded with Unshielded Twisted Pairs)

- SF/UTP kabel (engl. *Shielded and Foiled with Unshielded Twisted Pairs*) – to je S/FTP kabel koji dodatno ima i metalnu foliju oko svih žica zajedno, kao što to ima U/FTP.



Slika 52. SF/UTP kabel (eng. Shielded and Foiled with Unshielded Twisted Pairs)

- U/FTP kabel (engl. *Foiled with Unshielded Twisted Pair*) – to je kabel koji nema metalnu foliju oko svih parova žica zajedno, ali ima metalnu foliju oko svakog pojedinačnog para žica, što smanjuje smetnje unutar kabela između žica (engl. *Crosstalk*).

Slika 53. U/FTP kabel (eng. Shielded and Foiled with Unshielded Twisted Pairs)

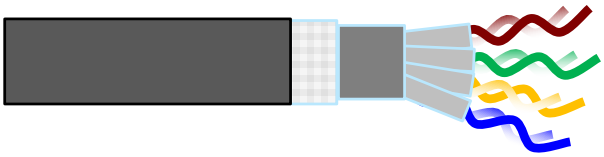
- F/FTP kabel (engl. *Foiled with Foiled Twisted Pair*) – isto kao i U/FTP, ali ima metalnu foliju oko svih žica zajedno i oko svakog para žica u kabelu pojedinačno. Pruža veću zaštitu od smetnji unutar kabela i od vanjskih elektromagnetskih smetnji.

Slika 54. F/FTP kabel (eng. Shielded and Foiled with Unshielded Twisted Pairs)

- S/FTP kabel (engl. *Shielded with Foiled Twisted Pair*) – u tom kabelu svaki je par žica omotan metalnom folijom, a onda su sve žice zajedno zaštićene fleksibilnim, ali snažnim metalnim ožičenjem, što daje bolja mehanička svojstva, ali i bolju zaštitu od elektromagnetskih smetnji.

Slika 55. S/FTP kabel (eng. Shielded and Foiled with Unshielded Twisted Pairs)

- SF/FTP kabel (engl. *Shielded and Foiled with Foiled Twisted Pair*) – takav kabel pruža najveću zaštitu od elektromagnetskih smetnji iz okoline i unutar kabela između parova žica. Svaki par žica ima svoju metalnu foliju, sve žice zajedno imaju metalnu foliju i, dodatno, sve žice zajedno imaju fleksibilno metalno ožičenje.

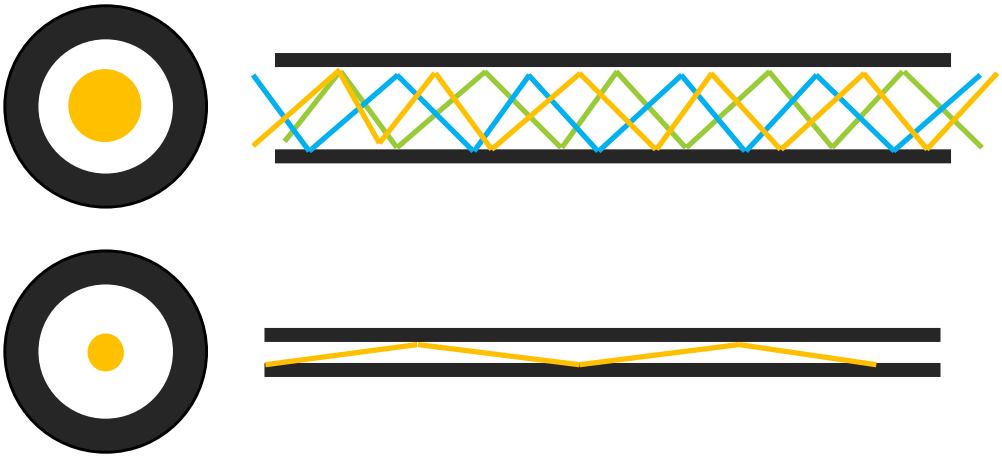


Slika 56. SF/FTP kabel (eng. Shielded and Foiled with Unshielded Twisted Pairs)

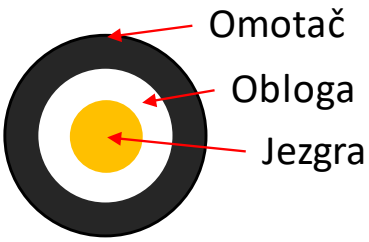
Žičani bakreni UTP kabeli još se dijele i prema frekvencijama, odnosno brzinama prijenosa podataka koje podržavaju. Postoje i druge kategorije poput Cat5 ili Cat3, ali one nisu važne jer nisu aktualne. Minimalna kategorija danas bi bila Cat5e.

Optički kabeli također imaju svoju kategorizaciju i različite izvedbe s obzirom na namjenu, tako mogu biti podvodni, podzemni i nadzemni, ali također mogu biti i u izvedbi prespojnih kabela koji su vrlo fleksibilni i koriste se unutar zaštićenih prostora poput podatkovnih centara za povezivanje uređaja koji imaju optička sučelja. Optički kabeli se sastoje od staklenih ili plastičnih niti kojih može biti i više desetaka unutar jednog kabela. Signali koji se prenose optičkim nitima svjetlosni su signali i oni odgovaraju bitovima koji predstavljaju podatke koji se šalju i primaju između uređaja u mreži. Karakteristike optičkih kabela su neosjetljivost na elektromagnetske smetnje, puno veći kapacitet prijenosa podataka u odnosu na bakrene kabele, puno su otporniji na prisluškivanje, što pridonosi sigurnosti sustava IKT-a u cjelini, iskoristivi su na puno većim udaljenostima od bakrenih kabela (i više stotina kilometara, dok se bakreni koriste do 100 metara), također vrlo su otporni na mehanička oštećenja i utjecaje okoline. Dvije glavne kategorije optičkih kabela su jednomodno optičko vlakno (engl. *Single-mode Fiber* – SMF) i višemodno optičko vlakno (engl. *Multimode Fiber* – MMF). Obje vrste imaju svjetleće diode kao izvor svjetlosti, ali u jednomodnim optičkim nitima izvor svjetlosti puno je koherentniji i snažniji (laser). Na slici 57. vidljiva je propagacija svjetlosnog signala kroz višemodno vlakno koje ima veći promjer jezgre (50 mikrometara; u starijim verzijama 62,5 mikrometara), a kako propagacija svjetlosnog

signala izgleda kroz jednomodno vlakno koje ima manji promjer jezgre (9 mikrometara).



Slika 57. Prikaz presjeka optičke niti − vidljiva je jezgra i obloga



Slika 58. Presjek jedne optičke niti s objašnjenjem

Slike 57. i 58. prikazuju optičku nit u presjeku i usporedbu dimenzija jezgre s većim indeksom loma svjetlosti i obloge s nižim indeksom loma svjetlosti, zbog čega „reflektira“ svjetlost nazad prema jezgri. Također, optičke niti imaju zaštitni omotač od plastike u svrhu mehaničke zaštite.

- **SMF** optički kabeli imaju jezgru malog promjera i služe za prijenos vrlo preciznih valnih duljina svjetlosti za prijenos podataka. To omogućava prijenos podataka većim brzinama i na veće udaljenosti. Obično se koristi za povezivanje gradskih četvrti, gradova i u podatkovnim centrima za velike brzine prijenosa podataka. Također, gubitci u SMF kabelima znatno su manji od gubitaka u MMF kabelima.
- **MMF** optički kabeli imaju jezgru većeg promjera i služe za prijenos svjetlosti širega spektra valnih duljina. Prijenos podataka je moguć velikim brzinama, ali na kraće udaljenosti, obično unutar jedne zgrade ili kampus mreže, ali isto tako u podatkovnim centrima. Cijena tih kabela i opreme koja ih može koristiti nešto je niža u odnosu na SMF kabele i opremu.

1.3. Korištenje osnovnih naredbi

Iako će se u priručniku puno puta koristiti razne naredbe na *Windows* računalu za provjeru IP postavki ili spajanje na udaljeni usmjernik ili preklopnik, dobro je često korištene naredbe imati na jednom mjestu.

Sve naredbe na računalu upisivat će se u naredbeni redak do kojeg se dolazi pritiskom na tipku START te se upisuje CMD ili se kombinacijom tipki *Windows*+r otvori prozor za naredbe u koji će se upisati CMD, nakon čega se pritisne tipka ENTER.

Naredbe su napisane po važnosti, odnosno učestalosti korištenja.

- *cls* → briše sadržaj ekrana;
- *ipconfig* → tom naredbom mogu se saznati IP postavke na mrežnim karticama računala (IP adresu računala, mrežnu maska, IP adresu mrežnog pristupnika i IP adresu DNS-a);
- *ipconfig/all* → tom naredbom dobiju se sve informacije koje daje naredba *ipconfig*, uz još neke dodatne, kao što je informacija o MAC adresi mrežne kartice, DNS poslužitelju, DHCP poslužitelju i slično;
- *ping* → tom naredbom računalo, koristeći ICMP protokol, generira i šalje mali testni paket od 32 bita prema zadanoj IP adresi. Cilj je saznati je li neki uređaj dostupan u računalnoj mreži. Ta naredba ima razne varijacije, a neke su od njih:

ping-n 7 178.218.162.131 → šalje 7 ICMP paketa prema 178.218.162.131

ping-a 178.218.162.131 → uz *ping*, šalje i DNS upit za IP adresu 178.218.162.131

ping-t 178.218.162.131 → šalje ICMP pakete dok ga se ne zaustavi (CTRL+C)

ping-l 1500 178.218.162.131 → šalje ICMP paket veličine 1500 bita

ping-w 10 178.218.162.131 → šalje ICMP paket i čeka 10 sekundi na odgovor

ping-a-n 3-l 1500 178.218.162.131 → kombinacija više prethodnih naredbi

```
C:\>ping -a -n 3 -l 1500 178.218.162.131

Pinging algebra-web1.plusvps.com [178.218.162.131] with 1500 bytes of data:
Reply from 178.218.162.131: bytes=1500 time=16ms TTL=54
Reply from 178.218.162.131: bytes=1500 time=18ms TTL=54
Reply from 178.218.162.131: bytes=1500 time=17ms TTL=54

Ping statistics for 178.218.162.131:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 16ms, Maximum = 18ms, Average = 17ms

C:\>
```

Slika 59. Primjer kombinacije parametara za ping naredbu

- *tracert* → tom naredbom računalo, koristeći ICMP protokol, saznaje kojim putem promet između računala i odredišta ide računalnom mrežom. Ta naredba ima razne varijacije, a neke su od njih:

  tracert 178.218.162.131 → dobije se putanja uređaja na putu, uz DNS imena, ako ih ima

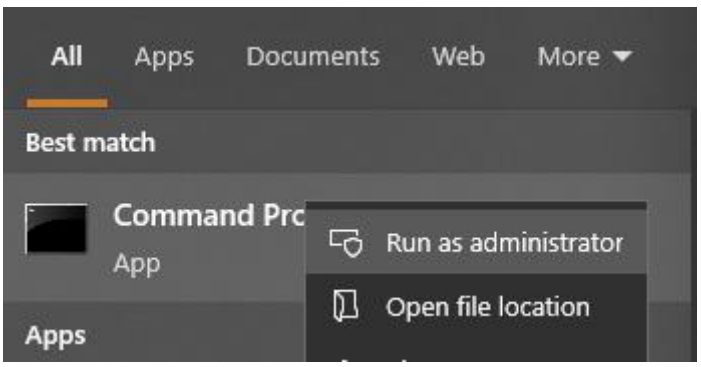
  tracert-d 178.218.162.131 → brže se dobije ispis putanje, ali bez DNS imena

  tracert-h 50 162.252.205.157 → računalo će saznati najviše do 50 koraka na putu do odredišta (predefinirano je 30)

  tracert -h 50-d 162.252.205.157 → brža verzija prethodne naredbe jer računalo ne radi DNS upite za svaki korak

- *arp-a* → dobije se ispis tablice u kojoj se nalaze IP adrese i MAC adrese svih uređaja s kojima je naše računalo komuniciralo (na bilo koji način);

- *arp-d* → briše se cijela ARP tablica (traži administratorske ovlasti u pokretanju CMD alata, a prilikom pokretanja koristi se desni klik miša);



Slika 60. Pokretanje naredbenog retka na Windows računalu

- *telnet* → tom naredbom može se uspostaviti veza s udaljenim uređajem u računalnoj mreži koji podržava *Telnet* protokol. Također može se provjeriti je li neki TCP ulaz otvoren na odredišnom uređaju (npr. port 80 = HTTP ili 443 = HTTPS);

  telnet 178.79.149.215 80

  telnet 178.79.149.215 443

- *nslookup* → ta naredba otvara dijalog s DNS poslužiteljem koji je konfiguriran na računalu;

- *ipconfig/displaydns* → ta naredba ispisuje sve DNS zapise koje računalo ima u svojoj memoriji. Zapisi nastaju uglavnom iz odgovora koje računalo dobije od svog DNS poslužitelja tijekom normalnog rada (otvaranje mrežnih stranica i sl.). Može ih biti jako puno, pa ih nekad treba brisati prilikom testiranja;

- *ipconfig/flushdns* → ta naredba briše sve DNS zapise u memoriji računala (gotovo sve što je prethodna naredba pokazala);

- *ipconfig/release* → ako je IP postavke računalo dobilo od DHCP poslužitelja, tom ih naredbom otpušta pa želimo li da računalo opet komunicira s drugim uređajima, postavke ponovno treba zatražiti;

- *ipconfig/renew* → ta naredba obnavlja IP postavke na računalu (računalo opet traži IP postavke od DHCP poslužitelja koji se obično nalazi negdje u lokalnoj računalnoj mreži);

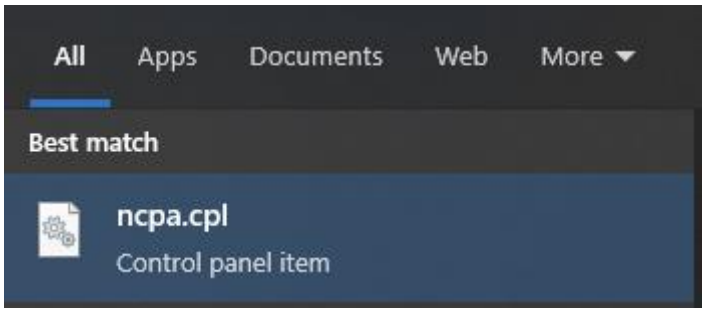
- *netstat-r* → ta naredba ispisuje tablicu usmjeravanja na računalu. Iako računalo nije usmjernik, ipak ima tablicu usmjeravanja u kojoj je najvažnija putanja ona koja kaže računalu da sav promet šalje na svoj *gateway*, tzv. predefinirana putanja (engl. *Default route*). Varijacija te naredbe je i *route print*;

- *netstat-a* → ta je naredba ponekad korisna jer ispisuje stanje konekcija na računalu s vidljivim IP adresama i TCP/UDP ulazima izvora i odredišta komunikacije (engl. *Socket Pair*).

  Važno je prilikom upravljanja računalom i njegovim korištenjem u konfiguraciji mrežnih uređaja znati kako promijeniti postavke mrežne kartice računala. Neka će računala, ovisno o kontekstu u kojem ih koristimo, imati više mrežnih kartica, što

stvarnih fizičkih uređaja, što virtualnih mrežnih kartica, ali način je konfiguriranja postavki uglavnom isti.

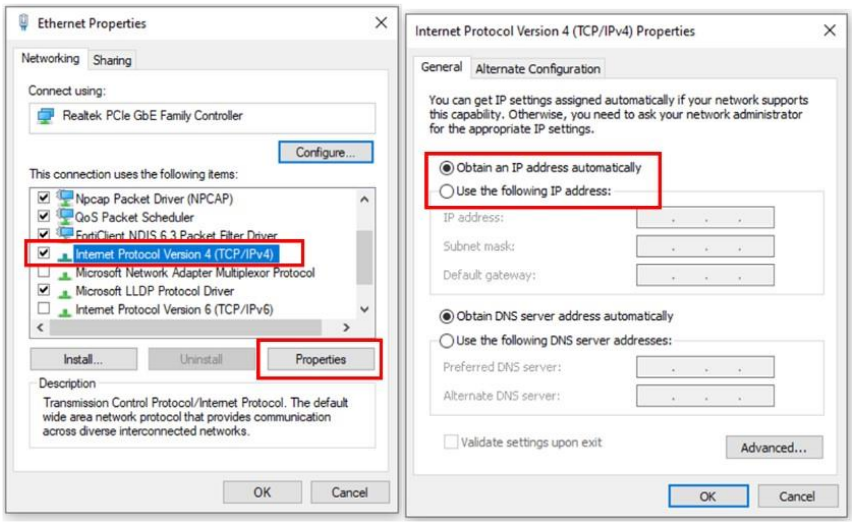
Za otvaranje postavki mrežne kartice, potrebno je pronaći konfiguracijski okvir mrežne kartice. To se može na nekoliko načina, a najlakše je u tražilicu (engl. *Windows Search*) upisati: „ncpa.cpl“ (engl. *Network Connections Properties Control Panel File*) i pritisnuti tipku ENTER.



Slika 61. Otvaranje popisa mrežnih kartica na Windows računalu

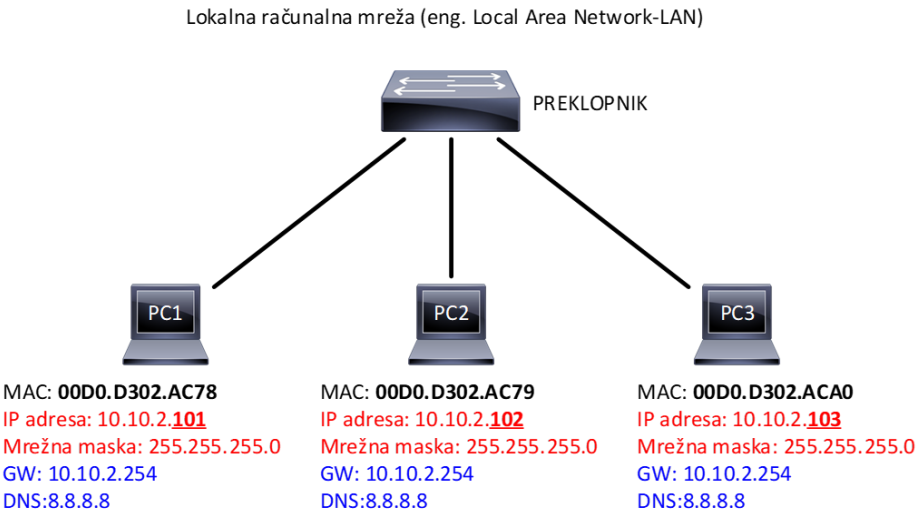
Slika 62. Prikaz mrežnih kartica na računalu

Nakon toga može se desnim klikom na određenu mrežnu karticu promijeniti postavke (engl. *Properties*) te mrežne kartice. Ono što se prvo mijenja način je na koji se kartica uopće konfigurira, je li to ručno upisivanje IP postavki ili će računalo IP postavke dobiti od DHCP poslužitelja u mreži.



Slika 63. Postavke jedne mrežne kartice na Windows računalu

U toj je fazi bitno znati da za većinu lokalnih mreža, pogotovo malih kućnih mreža u kojima nema mnogo uređaja, želimo li da računala mogu međusobno komunicirati, sve IP postavke na računalima moraju biti iste, osim IP adrese koja mora biti jedinstvena u odabranoj podmreži (eng. *Subnet*), kako je prikazano na slici 64.



Slika 64. Prikaz male lokalne računalne mreže

Crveni je dio konfiguracije obavezan da bi računala mogla međusobno komunicirati unutar lokalne mreže posredstvom preklopnika (engl. *Switch*), dok je plavi dio obavezan želimo li da računalo može izaći izvan lokalne mreže ili želimo li koristiti internet (npr. https://www.algebra.hr/). Svako računalo također ima i jedinstvenu MAC adresu, dodijeljenu mrežnoj kartici prilikom proizvodnje, te je ne konfiguriramo, niti mijenjamo. Ona se može mijenjati, ali to prosječan korisnik računala nikad ne radi.

1.4. IPv4 adresiranje

Danas je najkorišteniji internetski protokol verzije 4 (IPv4). Svaki uređaj koji treba komunicirati u računalnoj mreži mora imati logičku IPv4 adresu. IPv4 logičke adrese su 32-bitni zapisi koje računala koriste za komunikaciju unutar lokalne računalne mreže i između različitih podmreža u sustavu IKT-a. IPv4 adrese su hijerarhijski organizirane kako bi omogućile mapiranje organizacijske strukture na računalnu mrežu. S obzirom da su IPv4 adrese 32-bitni zapisi, ukupni broj IPv4 adresa iznosi 2^{32} (što iznosi 4294967296 IPv4 adresa). Iako taj broj izgleda velik, većina toga adresnog prostora neiskoristiva je iz raznih razloga. Međutim, danas adresni prostor IPv4 jednostavno nije dovoljan za sve potrebe modernih sustava IKT-a i javnoga interneta. Zato i postoji IPv6 protokol s mnogo više IP adresa, a kako vrijeme bude odmicalo, tako će IPv6 biti sve više u primjeni.

1.4.1. Binarni brojevni sustav

Računala i ostali uređaji u sustavu IKT-a danas koriste binarni brojevni sustav. U binarnom brojevnom sustavu postoje samo dvije znamenke, dva stanja, a to su 0 i 1.
Iako računala 32-bitne IPv4 adrese koriste kao neprekinuti niz od 32 bita, za potrebe onih koji konfiguriraju računala i ostale uređaje u sustavu IKT-a taj 32-bitni zapis predstavlja se kao decimalni zapis tako što su 32 bita podijeljena u četiri bajta, svaki po 8 bitova, odvojena točkama.
Primjer je IPv4 adresa u decimalnom obliku: 192.168.0.13 – kada bi se ta decimalna IPv4 adresa, koja je ljudima jasna, prevela u binarni zapis, to bi izgledalo ovako: 11000000.10101000.00000000.00001101
Prilikom pretvorbe iz binarnog u decimalni sustav, i obratno, svaki bajt se promatra i prevodi zasebno.

Postoje razni načini na koje se može naučiti kako se pretvaraju binarni u decimalne brojeve, i obratno. Važno je znati da svaka pozicija (P0 do P7) unutar jednog bajta ima svoju vrijednost i to se ponavlja u sva četiri bajta na isti način. Ako svaki bajt čitamo s desne na lijevu stranu, te vrijednosti gledamo kao potencije broja 2, počevši na P0 s 2^0 i tako sve do 2^7.

Slika 65. Vrijednost pozicija bitova unutar jednog bajta

Ovisno o tome koja je binarna kombinacija u pitanju, u svakom bajtu zbrojit će se vrijednosti samo onih pozicija čiji su bitovi 1.

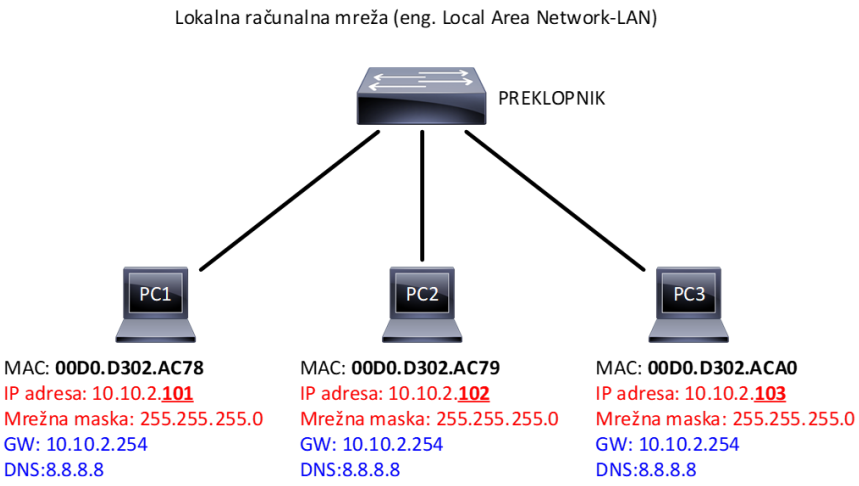
Tako, npr., za binarni niz 11000000.10101000.00000000.00001101 pretvaranje u decimalni izgleda ovako:

→128+64+0+0+0+0+0+0.128+0+32+0+16+0+0+0.0+0+0+0+0+0+0+0.0+0+0+0+1+1+0+1

→192.168.0.13

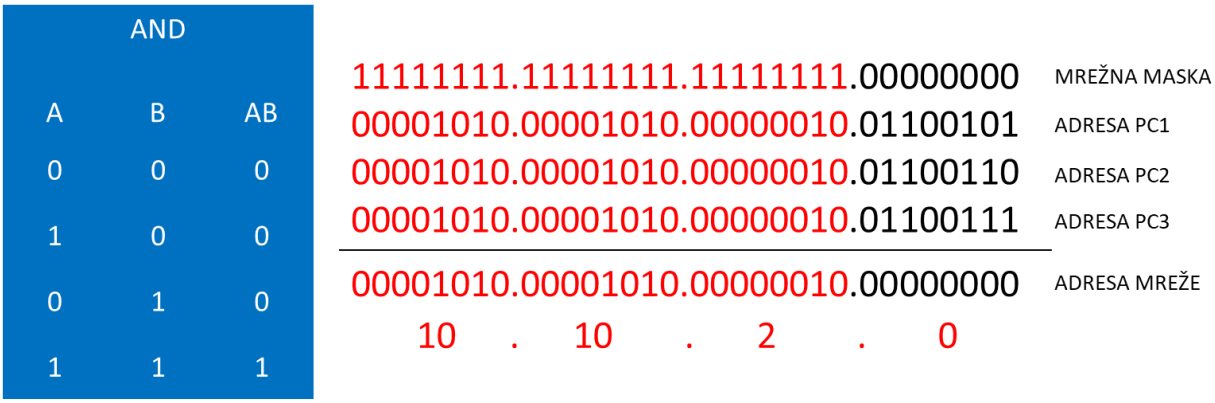
Zbog vrijednosti koje se koriste, najveći je broj koji se može dobiti u bilo kojem bajtu, kada radimo s IP adresama, 255. Najveći broj nije 256 jer je i binarna kombinacija 0000000 također valjana, pa se brojiti ne kreće od 1, nego od 0, i dolazi se do 255.

Želi li se konfigurirati uređaje koji trebaju međusobno komunicirati unutar lokalne računalne mreže, svaki uređaj mora imati jedinstvenu IP adresu. Sve ostale postavke im trebaju biti iste.



Slika 66. Prikaz male lokalne računalne mreže

Da bi uređaji znali u kojoj mreži se nalaze, i trebaju li slati promet direktno uređajima unutar lokalne mreže ili izvan posredstvom mrežnog pristupnika, svaka IP adresa ima dva dijela. Prvi dio IP adrese zajednički je svim uređajima koji su u istoj lokalnoj računalnoj mreži, i taj dio zove se adresa podmreže, kao što je vidljivo na slici 66. – svim je računalima zajednički dio 10.10.2. Međutim, računalo ne može, kao što to može čovjek, „pogledati“ i nešto zaključiti, ono mora obaviti unaprijed definirane logičke operacije i izračunati što napraviti s prometom koji treba poslati prema nekoj odredišnoj IP adresi. Da bi računalo koje šalje nekakve podatke prema odredišnom računalu moglo izračunati ima li odredišno računalo isti ili različit mrežni dio adrese, mora se koristiti mrežna maska. Upravo mrežna maska određuje dio IP adrese koji je adresa mreže, a koji dio je adresa računala u nekoj podmreži. Naravno, to se u računalu događa na binarnoj razini pomoću logičkih operacija. Broj bitova 1 u mrežnoj maski određuje broj bitova s lijeva na desno koji su adresa mreže u nekoj IP adresi. Na primjeru sa slike 66. mrežna maska je 255.255.255.0, što bi binarno bilo 11111111.11111111.11111111.0000000. Još je jedan uobičajen način pisanja mrežne maske u obliku prefiksa /n, tako može umjesto 255.255.255.0 biti pisano /24, što znači da su prva 24 bita u IP adresi mrežni dio IP adrese.



Slika 67. Logička operacija AND i određivanje adrese mreže

Koristeći logičku operaciju AND računalo između svoje mrežne maske (u ovom slučaju je ista svim računalima) i 32-bitne IP adrese zaključuje da su mrežni dio njegove IP adrese prva 24 bita, odnosno 00001010.00001010.00000010.0000000. Kada računalo želi komunicirati s nekim drugim računalom, uspoređuje bitove koji su mrežni dio njegove IP adrese i isti broj bitova odredišnog računala kako bi zaključio jesu li u

istoj mreži. Ovisno o tome jesu li u istoj mreži ili nisu, komunikacija će se drugačije odvijati.

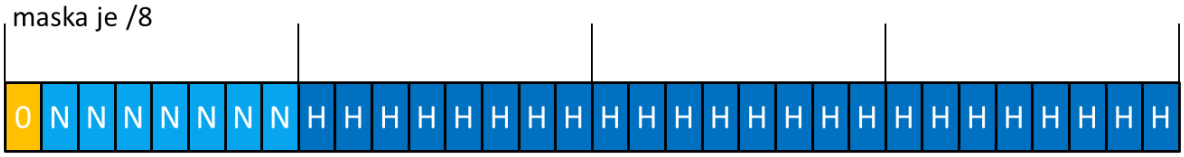
Sad kada je „tehnički dio“ objašnjen, pitanje je kako se uopće određuje koja mreža će se koristiti i koliko velika ta mreža treba biti.

U praksi izrada adresne sheme počinje identificiranjem potreba korisnika, odnosno koliko mu podmreža treba i koliko će računala biti u svakoj podmreži. Broj podmreža ovisit će o organizacijskoj strukturi tvrtke. Na primjer, ako se radi o bolnici s 10 različitih odjela, poput kirurgije, pedijatrije, hitnoga prijema, uprave itd. trebat će izraditi adresnu shemu za tih 10 odjela, i svaka podmreža treba biti dovoljno velika da svaki uređaj u određenom odjelu može dobiti IP adresu. Također, pitanje je koje se IP adrese smiju koristiti, a to ovisi o njihovoj namjeni.

Podjela IPv4 adresa

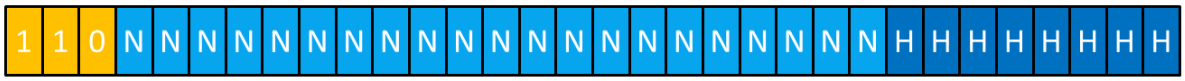
IPv4 adrese dijele se u pet velikih klasa: A, B, C, D i E. Ta je podjela jedan od prvih pokušaja određivanja mrežnog dijela IP adrese koji se danas tako ne koristi, iako podjela ima važnost i danas. Tih pet klasa određeno je kombinacijom bitova u prvom bajtu. Klasa A, B i C su one IP adrese koje se koriste u svakodnevnoj konfiguraciji mrežnih uređaja. Klasa D je raspon za višeodredišni promet, dok je klasa E rezervirana za eksperimentalnu upotrebu. U priručniku će usmjerenje biti na klasama A, B i C.

Klasa A – prvi bit je 0, a mrežni dio IP adrese ima 7 bitova na raspolaganju tako da je raspon od **0 do 127** u prvom oktetu. Za računala (hostove) na raspolaganju je 24 bita. Zadana mrežna maska je /8



Klasa B – prva dva bita su 1 i 0, a mrežni dio IP adrese ima 14 bitova na raspolaganju tako da je raspon od **128 do 191** u prvom oktetu. Za računala (hostove) na raspolaganju je 16 bitova. Zadana mrežna maska je /16

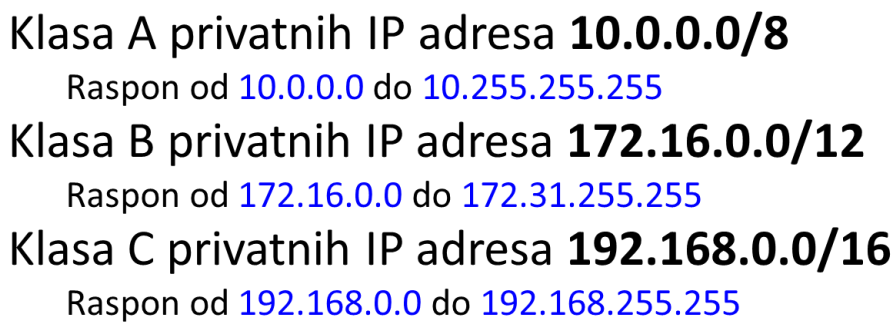
Klasa C – prva tri bita su 1, 1 i 0, a mrežni dio IP adrese ima 21 bit na raspolaganju tako da je raspon od **192 do 223** u prvom oktetu. Za računala (hostove) na raspolaganju je 8 bitova. Zadana mrežna maska je /24



Slika 68. Prikaz privatnih raspona IPv4 adresa iz klase A,B i C

Na temelju slike 68. možemo zaključiti koliko je moguće izraditi mreža iz svake klase i koliko računala može biti u svakoj mreži. Tako iz klase A možemo izraditi 2^7 mreža i svaka od tih mreža može imati 2^{24} računala. Od klase B možemo izraditi 2^{14} mreža i svaka od tih mreža može imati 2^{16} računala. I, konačno, od klase C možemo izraditi 2^{21} mreža i svaka od tih mreža može imati 2^8 računala. Takav način podjele IPv4 adresnog prostora na mrežni dio koji, npr. za B klasu uvijek mora biti /16 odnosno u takvim mrežama uvijek mora biti 2^{16} IP adresa, koristio se u ranim danima interneta i primjene IPv4 protokola kada se pogrešno mislilo da postoji i više nego dovoljno IPv4 adresa za primjenu u praksi. Vrlo brzo je postalo jasno da to nije tako, i danas se takav način ne koristi. Danas se koristi izrada podmreža koristeći varijabilnu mrežnu masku (engl. *Variable Length Subnet Mask*) pa se može izraditi adresna shema u skladu s potrebama organizacije i biti /16, ali i /22, /25 ili /30. Iako se u praksi gotovo nikada ne koriste mrežne maske klasa, neki uređaji prilikom konfiguracije i danas nude predefiniranu mrežnu masku na temelju klase IP adresa upisanu u mrežne postavke. Upiše li se tako u postavke mrežne kartice na *Windows 10* operacijskom sustavu IP adresa 192.168.10.10, on će automatski ponuditi mrežnu masku /24 (255.255.255.0); upiše li se 172.16.10.10, on će automatski ponuditi mrežnu masku /16. To će se, naravno, promijeniti da odgovara stvarnom stanju, jer klase nisu više u praktičnoj primjeni podmrežavanja.

Još jedna podjela predstavlja temelj za sve što se praktično primjenjuje u adresiranju: na privatne i javne adrese. Tako su za upotrebu u privatnim mrežama organizacija ostavljena tri raspona, po jedan iz klase A, B i C. Te se adrese smiju koristiti samo unutar privatnih organizacija i neće se vidjeti na javnom internetu. Sve ostale IP adrese iz klase A, B i C uglavnom su ostavljene za upotrebu na javnom internetu, uz neke iznimke, poput 127.0.0.0/8 raspona koji je rezerviran za *Loopback* mreže i dijagnostiku.



Slika 69. Raspon IP adresa za privatnu upotrebu?

1.4.2. Stvaranje podmreža (engl. *subnetting*) − izrada adresne sheme (podmrežavanje)

Nije moguće dovoljno naglasiti kako je izrada adresne sheme ključan korak u izgradnji efikasne računalne mreže. Ako je adresna shema pravilno definirana, tada će sve što kasnije treba implementirati u računalnu mrežu, poput usmjeravanja prometa, tuneliranja, migracija, sigurnosne pohrane, sigurnosnih politika i pravila na vatrozidima, biti jednostavno i smisleno. Međutim, nije li adresna shema izrađena pravilno, takva računalna mreža, a samim time i sustav IKT-a u cjelini, bit će ograničen u smislu skalabilnosti, sigurnosti, pouzdanosti, visoke dostupnosti i raznih drugih funkcionalnosti.

Za izradu adresne sheme u nekoj organizaciju koriste se tri privatna raspona sa slike 69. Najveći se dio podmrežavanja radi upravo koristeći privatne IP adrese. Javni adresni prostor se nešto drugačije koristi. Budući da IPv4 adresa nema mnogo na raspolaganju, vrlo rijetka je situacija podmrežavanja javnoga adresnog prostora. Uglavnom će javne IP adrese biti dane na korištenje kao već subnetirane male mreže ili, uobičajenije, kao samo jedna javna IP adresa, što je slučaj u mnogo malih i srednjih poslovnih organizacija.

U ovom priručniku neće puno prostora zauzeti dokazivanje besmisla *classful* pristupa u izradi adresne sheme. Iako su se u prošlosti podmreže „izrađivale" koristeći mrežne maske klase /8, /16 i /24, bez obzira na stvarne potrebe, danas treba prihvatiti da za mrežu u kojoj će biti samo 2 uređaja, kao što je to veza između dva usmjernika, koristiti mrežnu masku /24, a pogotovo /16 ili /8 nema nikakvog smisla iz aspekta učinkovitosti korištenja IPv4 adresnog prostora.

Kako se danas radi, može se vidjeti na primjeru izrade adresne sheme za malu tvrtku s četiri odjela. Kreće se od zahtjeva korisnika, odnosno informacije koliko će uređaja (računala, printera, IP kamera, IP telefona i slično) biti u svakom odjelu, a potom treba odrediti koliko će bitova u IP adresama tih računala biti isto kako bi svi ti uređaji bili dio iste podmreže. Na slici 70. prikazani su odjeli s brojem potrebnih IP adresa. U ovom primjeru neće se računati adrese za veze između mrežnih uređaja u samoj mreži jer bi za to bila potrebna shema mreže, ali svakako i taj bi dio trebalo uračunati kao 2 uređaja na svakoj vezi. Također, bitno je znati da se u svakoj podmreži dvije IP adrese ne koriste, a to je prva i zadnja binarna kombinacija. Prva binarna kombinacija u podmreži ima sve *host* bitove 0 i smatra se kao adresa same mreže, dok zadnja binarna kombinacija u podmreži ima sve *host* bitove 1 i to je sveodredišna (engl. *Broadcast*) adresa koja se može koristiti samo kao odredišna IP adresa. Budući da uređaji i šalju i primaju podatke, to znači da sveodredišnu IP adresu ne može koristiti ni jedan uređaj. Na temelju toga računa se koliko bitova treba u *host* dijelu IP adrese kako bi svi uređaji bili dio iste podmreže. Postavlja se pitanje:

> ***Na koju potenciju (n) mora biti 2 da bi se, oduzevši dvije adrese (network i broadcast), dobilo binarnih kombinacija barem koliko je uređaja u podmreži?***

Ovdje se jednostavno uvrsti *n* koji se smatra dovoljnim, i kada se izradi nekoliko adresnih shema, proces se automatizira i pojednostavnjuje.

Također treba biti oprezan s utvrđivanjem koliko uređaja zaista ima u nekoj podmreži. Često se zaboravlja da i mrežni pristupnik treba IP adresu ili preklopnik, ako se koristi u mreži, također treba IP adresu i slično. Važno je točno utvrditi sve uređaje koji će se koristiti u mreži, a trebaju IP adresu.

Odjel	Broj uređaja koji trebaju IP adresu
Uprava	15
Marketing	40
Prodaja	100
Proizvodnja	200

Slika 70. Zahtjevi korisnika za IP adresama

U navedenom primjeru postavlja se pitanje vezano za upravu: Da bi 2^n - 2 bilo ≥ 15, koliko je n?

Odrediti n = 1 jasno je da nije dovoljno, jer 2^1 -2 = 0, a treba ≥15. Za 15 korisnih IP adresa treba 5 bitova, dakle n = 5. lako se čini da je 4 dovoljno, uvijek treba oduzeti prvu (*network*) i zadnju (*broadcast*) kombinaciju, koje se ne koriste za računala. Stoga je 2^5 -2 = 30 dovoljno za podmrežu *Uprava*. Kada se utvrdi da za računala u podmreži treba 5 bitova, lako se izračuna koliko bitova je mrežni dio IP adrese svakog računala u toj podmreži, oduzimanjem od ukupnog broja bitova u IPv4 adresi onaj dio bitova koji je potreban za računala (*hostove*).

Taj je broj zapravo mrežna maska te je za *Upravu* to /27 ili 255.255.255.224, odnosno binarno 11111111.11111111.11111111.11100000.

Sada se pojavljuje pitanje je li ova mreža prevelika? Odgovor je nevažan jer nema izbora. Računalo radi na binarnoj razini i traži se minimalan broj bitova koji će zadovoljiti potrebe, a gotovo uvijek je više kombinacija nego što treba za računala. Želimo li podmrežu za 15 računala, najmanja moguća je upravo /27. To je najmanja veličina podmreže potrebna kada se oduzme adresa mreže (engl. N*etwork*) i sveeodredišna (engl. *Broadcast*) adresa, koje se ne mogu koristiti.

Da bi se završio izračun za navedenu malu mrežu, treba izračunati:

- Marketing: Da bi 2^n - 2 bilo ≥ 40, koliko je n?
- Prodaja: Da bi 2^n - 2 bilo ≥ 100, koliko je n?
- Proizvodnja: Da bi 2^n - 2 bilo ≥ 200, koliko je n?

Odgovori:

- Marketing n = 6; mrežna maska je /26, odnosno 255.255.255.192
- Prodaja n = 7; mrežna maska je /25, odnosno 255.255.255.128

- Proizvodnja n = 8; mrežna maska je /24, odnosno 255.255.255.0

Nakon izračuna koje mrežne maske će koristiti svaka podmreža, bilo bi dobro znati koja će to mreža biti, i hoće li to biti privatni ili javni adresni prostor. Kako je ranije napisano, postoje tri raspona namijenjena upotrebi u privatnim organizacijama, i to su 10.0.0.0/8, 172.16.0.0/12 i 192.168.0.0/16. Odabire se raspon ovisno o vlastitim potrebama ili sva tri raspona, što je slučaj u složenijim mrežama, s obzirom na više mogućnosti hijerarhijskog dizajna mreže. Dakle, bez obzira koji raspon se koristi, izračun i dalje vrijedi. Ako se koristi raspon iz klase C, što bi bilo 192.168.0.0/16, važno je prepoznati da je mrežna maska za raspon C /16, da se zna koliko je velik taj raspon i je li odgovarajuć. Može biti i drugačije, ali treba odlučiti i usredotočiti se na izradu adresne sheme za odjele u organizaciji. U tom procesu bitno je znati nekoliko stvari:

- uvijek se kreće od najvećeg zahtjeva prema najmanjem kako bi mreža bila skalabilna i kako bi usmjeravanje prometa bilo učinkovito (o tome više kasnije);
- ono što je zadano kao adresni prostor za podmrežavanje (u ovom slučaju C klasa privatnih IP adresa), koristi se kao adresa prve mreže;
- podmreže se ne smiju preklapati.

Imajući to na umu, adresna shema će izgledati kao na slici 71.

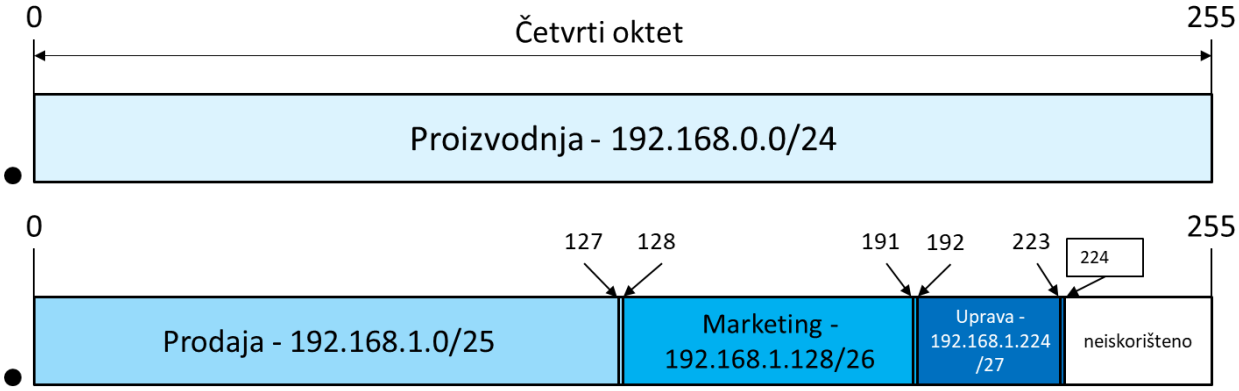
Odjel	Broj uređaja koji trebaju IP adresu	Adresa mreže	Mrežna maska
Proizvodnja	200	192.168.0.0	255.255.255.0
Prodaja	100	192.168.1.0	255.255.255.128
Marketing	40	192.168.1.128	255.255.255.192
Uprava	15	192.168.1.192	255.255.255.224

Slika 71. Rješenje podmrežavanja za primjer male organizacije

Kako to da je podmreža *Marketinga* 192.168.1.128, a *Uprave* 192.168.1.192? Promotri li se na binarnoj razini, može se vidjeti da je za podmrežu *Proizvodnja* mrežna maska /24, što znači da je za računala određeno 8 bitova, odnosno cijeli je četvrti bajt bio samo za tu podmrežu. To automatski znači da sljedeća mreža ne može imati prva tri bajta ista (jer se podmreže ne smiju preklapati, ako želimo funkcionalan sustav IKT-a) i zato treba uzeti 192.168.1.0. Ta četiri bajta mogu se zamisliti kao brojači na nekakvom brojilu (npr. mjerenje potrošnje plina), gdje svaki puni krug četvrtog bajta učini inkrementalni pomak od 1/255 na trećem bajtu. Cijeli je krug napravljen u četvrtom bajtu (od 0 do 255), i sad je povratak na nulu, a u tom procesu povlači se sljedeći broj

u trećem bajtu, a to je 1. Pitanje je, nadalje, kako je *Marketing* na 192.168.1.128. Jasno je da u trećem bajtu mora biti promjena 0 u 1 jer je podmreža 192.168.0.0/24 iskorištena za odjel *Proizvodnja* pa podmreža odjela *Prodaja* mora početi na 192.168.1.0. Ono što je nepoznato, gdje završava podmreža *Prodaja*. Promatra li se binarno, vidljivo je da je mrežna maska za odjel *Prodaja* /25, a to znači da taj 25. bit pripada mrežnom dijelu IP adrese za sva računala u podmreži odjela *Prodaja* i da je u toj podmreži 7 bitova ostalo za računala. S tih 7 bitova može se dobiti 2^7 kombinacija. Kada svi bitovi u dijelu IP adrese za računala budu 1, a to je slučaj za zadnju moguću kombinaciju u nekoj podmreži – sveodredišnu (engl. *Broadcast*) IP adresu, onda je jasno gdje završava podmreža odjela *Prodaja*, a to je na 192.168.1.127. To znači da sljedeća podmreža, u navedenom slučaju za odjel *Marketing,* može početi tek na 192.168.1.128. Kada je to poznato, kao i koja je određena mrežna maska za podmrežu odjela *Marketing*, tada se može istom logikom odrediti koja je sveodredišna IP adresa za podmrežu *Marketing*, i gdje počinje podmreža *Uprava*. Ako podmreža odjela *Marketing* počinje na 192.168.1.128 i ima 6 bitova na raspolaganju za računala, to znači da podmreža odjela *Uprava* može početi tek na $128+2^6$ (gleda se samo četvrti bajt), a to je 192, dakle na 192.168.1.192.

Može se postaviti pitanje kako se odmah došlo do mrežne adrese za odjel *Uprava* i preskočilo sveodredišnu (engl. *Broadcast*) adresu za podmrežu odjela *Marketing*. Da bi se odredilo sveodredišnu adresu za podmrežu odjela *Marketing*, treba samo sve bitove u četvrtom bajtu pretvoriti u decimalni broj, a to znači da binarni niz 10111111 postaje decimalni broj 191. Crveno su označeni samo bitovi koji su određeni za računala, a to je poznato zato što je mrežna maska za odjel *Marketing* /26, dakle zadnjih je 6 bitova za računala. Dakle, sveodredišna IP adresa je za odjel *Marketing* 192.168.1.191, što znači da je adresa mreže za odjel *Uprava* 192.168.1.192.



Slika 72. Rezultat podmrežavanja za organizaciju s četiri odjela

Zaključak je da mrežna maska određuje veličinu neke podmreže, a ovisno o tome koliko podmreža ima, i koje su njihove veličine, te podmreže će imati svoje početke i završetke na različitim vrijednostima u različitim bajtovima.

Naravno, u praksi je nešto drugačije, pa da bi bilo lakše i da bi bilo dovoljno IP adresa u nekoj podmreži, čak i ako se za pola godine pridruži još 10 ili 20 uređaja, obično se koristi /24 mrežna maska. Međutim, nije to primjenjivo u svim situacijama iako u prosječnim odjelima tvrtki obično nema više od 254 uređaja koja trebaju IP adresu. Važno je točno i precizno podmrežavati kako bi se moglo upravljati usmjeravanjem u mreži, kreirati pravila za vatrozide, konfigurirati tunele, NAT mehanizam i puno drugih bitnih stvari u mreži. U edukaciji o mrežama podmrežavanje ima prioritet, upravo stoga što implementacija svih naprednih tehnologija podrazumijeva brzo, točno i precizno podmrežavanje.

Također je važno znati gdje podmreže određene veličine (mrežna maska definira veličinu podmreže) uopće mogu imati svoj početak, i svoj kraj. Bez obzira o kojem bajtu se radi, vrijedi isti princip, samo što će podmreže biti različitih veličina. Tako podmreže koje imaju mrežnu masku /26 mogu početi samo u četvrtom bajtu jer je bit 26. u četvrtom bajtu. Po istom principu, podmreže koje imaju mrežnu masku /18 mogu početi samo u trećem bajtu jer je bit 18. u trećem bajtu. Podmreže koje imaju mrežnu masku /10 mogu početi samo u drugom bajtu, a podmreže s mrežnom maskom /2 mogu početi samo u prvom bajtu.

Da bi se znalo na kojim vrijednostima u pojedinom bajtu neka podmreža može imati svoj početak, treba na temelju mrežne maske pogledati samo bitove koji čine

nepotpuni bajt. Na primjer, za mrežnu masku /26 nisu važni bajtovi 1., 2. i 3., jer su ti bajtovi potpuni, već samo četvrti bajt jer je on nepotpun. Za mrežnu masku /26 u prva se tri bajta nalaze 24 bita, a u četvrtom bajtu, koji je nepotpun, nalaze se još 2 bita da bi bila mrežna maska /26. Vrijednosti koje se mogu dobiti kombiniranjem ta dva bita jedine su moguće vrijednosti na kojima podmreže s mrežnom maskom /26 mogu imati svoj početak, bez obzira o kojoj klasi IP adresa se radi. Pozicije prvog i drugog bita s lijeva na desno u svakom bajtu vrijede 128 i 64. Stoga, ovisno o kombinaciji ta dva bita (00=0, 01=64, 10=128, 11=192), podmreže **/26** mogu početi na N.N.N.**0**/**26**, N.N.N.**64**/**26**, N.N.N.**128**/**26**, N.N.N.**192**/**26**.

Za podmreže koje imaju mrežnu masku **/18** rješenje je vrlo slično, samo što je u trećem bajtu. Tako podmreže /18 mogu početi na N.N.**0**.**0**/**18**, N.N.**64**.**0**/**18**, N.N.**128**.**0**/**18**, N.N.**192**.**0**/**18**.

Za podmreže koje imaju mrežnu masku **/10** promjena je u drugom bajtu. Tako podmreže /10 mogu početi na N.**0**.**0**.**0**/**10**, N.**64**.**0**.**0**/**10** 6, N.**128**.**0**.**0**/**10**, N.**192**.**0**.**0**/**10**.

Podmreže koje imaju mrežnu masku **/2** imaju promjenu u prvom bajtu. Tako podmreže /2 mogu početi na **0**.**0**.**0**.**0**/**2**, **64**.**0**.**0**.**0**/**2**, **128**.**0**.**0**.**0**/**2**, **192**.**0**.**0**.**0**/**2**.

Drugi je način saznanja gdje mogu početi podmreže, ovisno o mrežnoj maski koja se koristi, da se vrijednost zadnjeg bita koji je obuhvaćen mrežnom maskom (npr. 26. bit za mrežnu masku /26) koristi kao korak počevši od nule u bajtu u kojem je taj zadnji bit obuhvaćen mrežnom maskom. Tako se za mrežnu masku /26, uzima da 26. bit vrijedi 64, i to je korak. Krene li se od nule u četvrtom bajtu, onda se može zaključiti da podmreže s mrežnom maskom mogu početi jedino na 0, 0 + 64 = 64, 64 + 64 = 128 i na 128 + 64 = 192. Koristeći isti princip, podmreže koje imaju mrežnu masku /27 (vrijednost je 27. bita 32, i nalazi se u četvrtom bajtu) mogu početi samo na 0, 32, 64, 96, 128, 160, 192 i 224. Navedeno vrijedi bez obzira o kojoj adresi mreže se radi.

Slika 73. Prikaz podmrežavanja s mrežnom maskom /26

1.4.3. Sažimanje IP adresa (sumarizacija)

Sažimanje putanja ključno je za učinkovito usmjeravanje u računalnoj mreži. Preduvjet je za to adekvatno definirana adresna shema, izrađena od najvećeg zahtjeva prema najmanjem, bez preklapanja i uvažavajući geografsku razdiobu podmreža. Geografska razdioba potrebna je zbog čvorišta (usmjernici) u mreži na kojima je moguće predstaviti više podmreža koje se koriste u nekoj geografskoj regiji kao jednu zajedničku podmrežu (sažimanje adresa). Treba li sažeti podmreže koje nisu blizu jedna drugoj, rezultat će biti puno veća sažeta podmreža, a to može uzrokovati probleme prilikom usmjeravanja prometa u mreži. Zato je ključno ispravno definirati adresnu shemu (podmrežavati).

Da bi saželi određene podmreže, najlakše je koristiti binarni pristup. U tom postupku nije bitna mrežna maska pojedinačnih podmreža ili IP adresa. To znači da se sve podmreže koje treba sažeti u jednu zajedničku podmrežu zapišu u binarnom obliku. Zatim se s lijeva na desno traži prvi bit koji se u tim mrežama razlikuje. Ispred tog bita „povlači se crta“ (nije nužno, ali je korisno), prebroje svi zajednički bitovi i njihov broj zapiše kao mrežna maska, u decimalnom obliku ili u obliku prefiksa. Svi se bitovi desno od te crte pretvaraju u nule, a zatim cijeli binarni niz pretvara nazad u decimalni oblik. Tako se dobiva zajednička sažeta podmreža koja predstavlja sve pojedinačne podmreže.

Primjer:

Potrebno je pronaći sažetu podmrežu za dvije podmreže.

- 10.11.12.0/26
- 10.11.12.96/27

1. Korak – zapisati obje podmreže u binarnom obliku

10 . 11 . 12 . 0

00001010.00001011.00001100.00000000

10 . 11 . 12 . 96

00001010.00001011.00001100.01100000

2. Korak – tražiti s lijeva na desno sve iste bitove i prije prvog različitog bita „povući crtu“

00001010.00001011.00001100.0|0000000

00001010.00001011.00001100.0|1100000

3. Korak – svi isti bitovi prije crte čine mrežni dio nove sažete podmreže; na temelju toga jasno je koja je mrežna maska. U ovom slučaju, to je /25, odnosno 255.255.255.128

4. Korak – svi se bitovi desno od crte pretvore u nule, a zatim cijeli binarni zapis prevede u decimalni kako bi se dobila čitljiva sažeta IP adresa podmreže koja obuhvaća sve pojedinačne podmreže. U ovom slučaju rezultat je sažimanja dvije zadane podmreže 192.168.0.0/25.

2. Preklapanje i usmjeravanje u računalnoj mreži

U OVOJ CJELINI NAUČIT ĆETE:

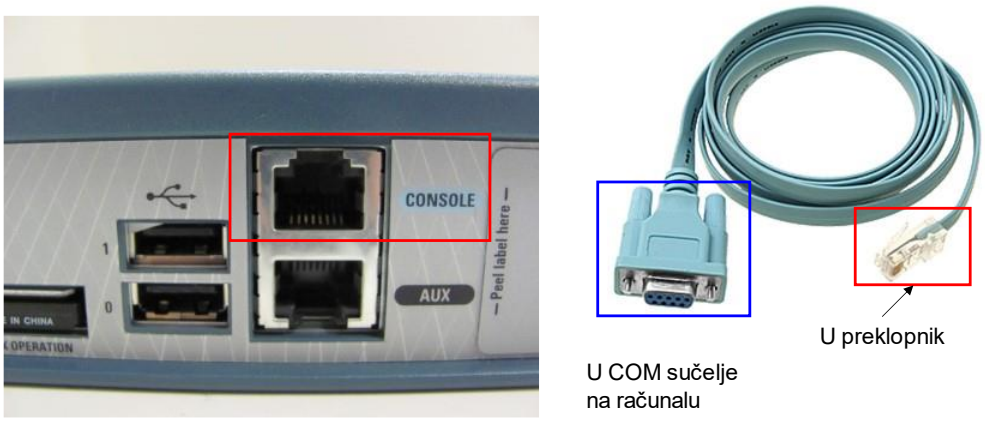
- konfigurirati preklopnike i usmjernike za udaljeni pristup
- logički segmentirati računalnu mrežu (VLAN)
- implementirati DHCP
- implementirati STP
- implementirati HSRP i VRRP
- implementirati statičko usmjeravanje
- implementirati protokole usmjeravanja RIP, OSPF i EIGRP
- implementirati redistribuciju putanja između protokola usmjeravanja.

Dvije su osnovne funkcionalnosti u računalnoj mreži preklapanje i usmjeravanje prometa. Te zadaće obavljaju i dva najzastupljenija uređaja u računalnim mrežama, preklopnici i usmjernici. Da bi računalna mreža bila efikasna i omogućavala uspješnu komunikaciju između krajnjih uređaja, nužno je posvetiti vrijeme osmišljavanju adresne sheme koja je preduvjet za pravilno usmjeravanje i preklapanje prometa. Potom treba adekvatno osmisliti osnovno usmjeravanje u računalnoj mreži, osobito usmjeravanje prometa u slučaju otkaza primarnih veza kako bi se komunikacija nesmetano nastavila bilo da se radi o otkazu veza između pojedinih ključnih uređaja, bilo uređaja u cjelini.

2.1. Osnovna konfiguracija preklopnika

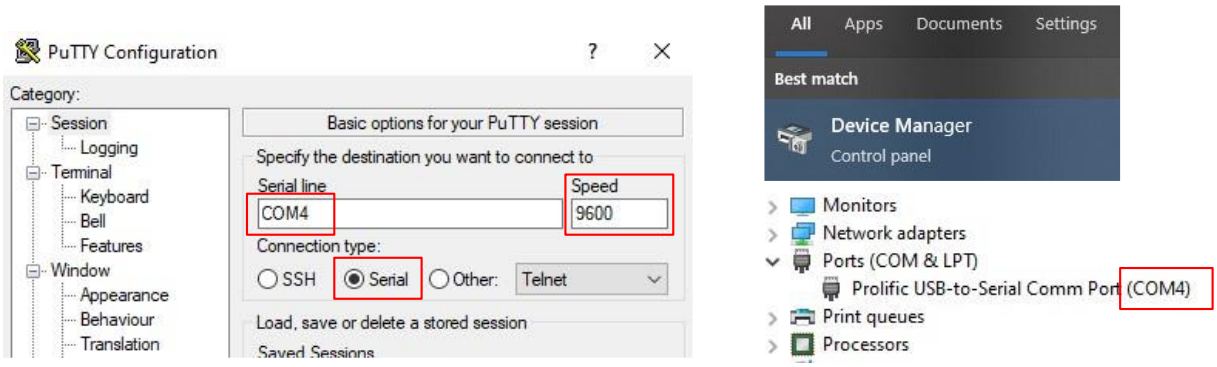
Moguće je da se preklopnik koristi, a da korisnik to i ne zna. Ima li pristup internetu u svom domu, vrlo vjerojatno je od svog pružatelja internetskih usluga dobio na raspolaganje uređaj kojim se njegovo računalo povezuje na internet. Taj univerzalni uređaj je preklopnik, usmjernik, vatrozid i bežična pristupna točka u jednom. Na njega se može povezati i konfigurirati ga koristeći bilo koji internetski preglednik i predefinirane postavke koje taj uređaj koristi (IP adresa, korisničko ime i lozinka). Isto vrijedi i za naprednije uređaje koji su namijenjeni isključivo jednoj zadaći unutar sustava IKT-a, poput preklopnika. Iako će preklopnik funkcionirati bez ikakve konfiguracije omogućujući komunikaciju svim uređajima koji su na nj povezani, to obično nije cilj. Kada već podržava napredne tehnologije poput agregacije veza i slično, onda ga ima smisla konfigurirati da koristi te napredne tehnologije. Da bi se preklopniku moglo pristupiti i konfigurirati ga na siguran način, posebno s udaljene lokacije, potrebno je konfigurirati neke osnovne postavke. Za to je svakako potrebno koristiti prateću dokumentaciju proizvođača. U toj su dokumentaciji opisani postupci, koraci rukovanja uređajem, instaliranje i konfiguriranje uređaja. U nastavku će biti opisano kako se pristupa i radi osnovna konfiguracija mrežnih uređaja tvrtke *Cisco* koji se koriste u mrežnom simulatoru *PacketTracer*.

Preklopniku se može pristupiti u internetskom pregledniku, konzolnom sučelju ili udaljenom vezom koristeći *Telnet* protokol ili vrlo siguran SSH protokol. Odabir je subjektivan i ovisi o potrebama. Ponekad je *Telnet* dovoljan, npr. za potrebe vježbi na nastavi, dok je drugdje SSH protokol nužan, npr. u stvarnom okruženju poslovne organizacije kada nije siguran *Telnet* (primjer je ispis u *Wiresharku*, gdje je bilo lako pročitati lozinke). Osnovno je povezivanje uglavnom konzolnim sučeljem koje koristi serijski RS232 komunikacijski protokol. Ta se veza može koristiti kada sve drugo otkaže, npr. u situaciji da nestane sva konfiguracija s uređaja ili se nešto dogodi s prostorom za pohranu na mrežnom uređaju (NVRAM ili FLASH memorija).



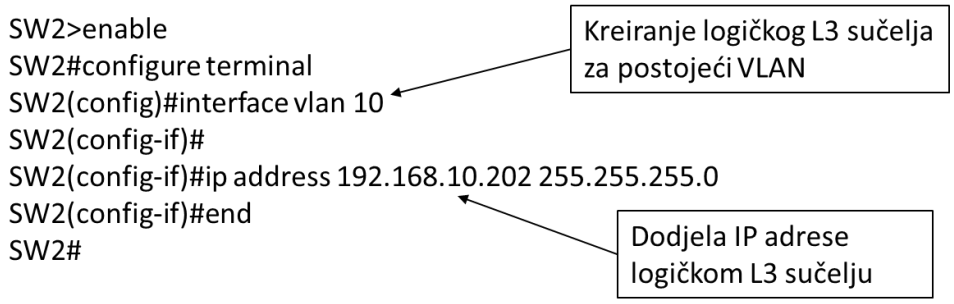
Slika 74. Prikaz konzolnog sučelja na mrežnom uređaju i konzolnog kabela za povezivanje računala i mrežnog uređaja-slika s interneta davno preuzeta ne znam izvor

Nakon što se poveže računalo s mrežnim uređajem pomoću konzolnog sučelja, može se koristiti aplikaciju koja će omogućiti konfiguracijski prozor u koji će se unositi potrebne naredbe. Za tu namjenu koristit će se alat *Putty*, ali može i bilo koji sličan. Bitno je ispravno odabrati COM sučelje u alatu koje zaista postoji na osobnom računalu. Neka računala imaju dedicirano COM sučelje, ali u prijenosnom računalu to obično nije slučaj, pa se koriste tzv. *USB-to-Serial* konverteri/adapteri. Na slici 75. prikaz je kako bi to moglo izgledati na računalu. Kada je odabrano prikladno COM sučelje, može se otvoriti konekcija pritiskom na gumb *Open* u alatu *Putty*.



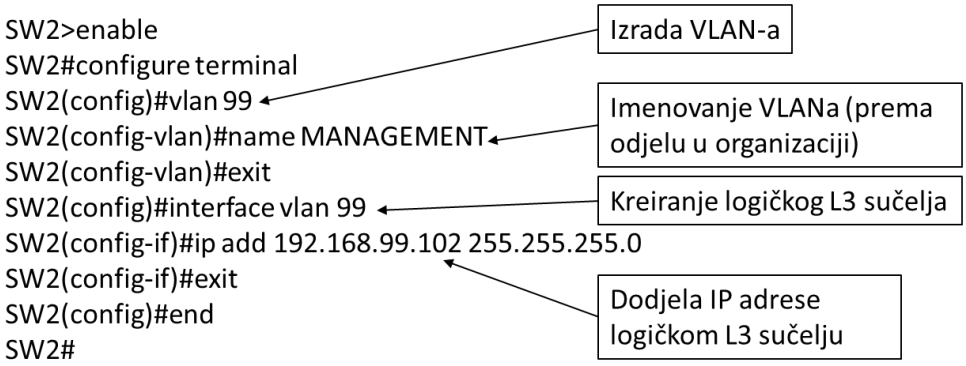
Slika 75. Povezivanje na preklopnik s računala posredstvom konzolnog sučelja (uz USB-to-Serial konverter)

Želi li se preklopniku moći pristupiti koristeći IP adresu umjesto konzolnog pristupa, što se obično koristi za upravljanje mrežnim uređajima s udaljenih lokacija, treba mrežni uređaj konfigurirati s IP adresom.



Slika 76. Priprema preklopnika za upravljanje postojećim podatkovnim VLAN-om u kojem su računala

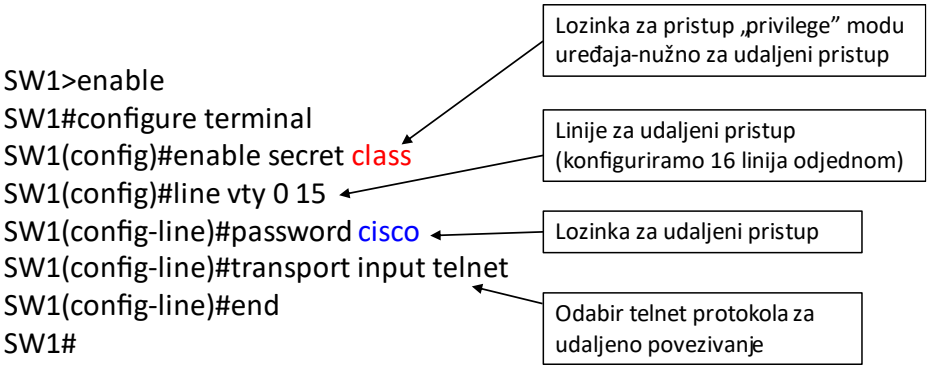
Ta IP adresa može biti konfigurirana na L3 sučelju preklopnika koje je u istom VLAN-u kao i računalo, ali može biti i posebno L3 sučelje u tzv. *Manegement* VLAN-u koji se posebno štiti od pristupa neovlaštenih podmreža i osoba. Potonje je češće u implementaciji upravljanja mrežnim uređajima. Međutim, da bi se računalo moglo povezati na preklopnik tim posebnim upravljačkim VLAN-om, ono mora imati konfiguriranu IP adresu mrežnog pristupnika uređaja u postavkama svoje mrežne kartice. Ako u mreži ne postoji usmjernik, mrežni pristupnik za računalo mora biti logičko L3 sučelje na preklopniku za VLAN u kojem je računalo; u ovom slučaju to bi bilo sučelje *interface VLAN 10.* To nije i dovoljan preduvjet. Naime, preklopnici se predefinirano ponašaju kao L2 uređaji, što znači da promet prosljeđuju samo na temelju MAC adresa unutar istog VLAN-a i ne usmjeravaju promet između VLAN-ova. Da bi preklopnik dobio mogućnost usmjeravanja prometa između različitih VLAN-ova, a samim time između različitih podmreža (VLAN 10 = 192.168.10.0/24, VLAN 99 = 192.168.99.0/24), treba aktivirati tu funkcionalnost naredbom *SW2(config)#ip routing*, i tek nakon toga preklopnik će omogućiti računalu povezivanje na nj koristeći IP adresu koja je u drugom VLAN-u, a osim toga preklopnik sada djeluje i kao usmjernik za promet između VLAN-ova. Iako nije najbolja praksa u mrežama, jedna je to od mogućnosti, ne i nužnost. Zadatak je mrežnih stručnjaka naći optimalno rješenje koje će adekvatno podržati poslovanje uvažavajući sve potrebe i ograničenja sustava IKT-a, i same poslovne organizacije.



Slika 77. Priprema preklopnika za upravljanje posebnim upravljačkim VLAN-om (engl. Management VLAN)

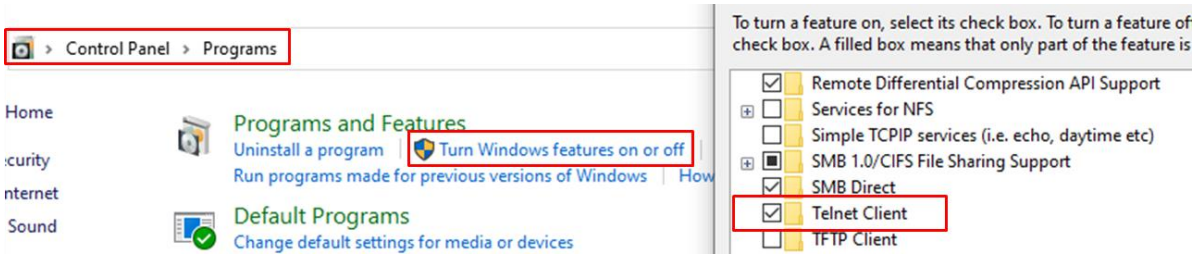
Noviji i napredniji preklopnici imaju posebno sučelje koje se može koristiti isključivo za upravljanje uređajem (engl. *Management*) i ni za što drugo. To je najbolja praksa jer je mreža u kojoj je to sučelje potpuno odvojena od ostatka podatkovne mreže. Na taj je način pristup preklopniku zajamčen unatoč otkazu komunikacije u podatkovnoj mreži te administratori mreže mogu nesmetano ukloniti uzroke otkaza u mreži. Koji god bio slučaj, bitno je da IP adresa na mrežnom uređaju bude dostupna računalu s kojeg pristupamo mrežnom uređaju, i da mrežni uređaj „zna“ vratiti promet prema tom računalu.

Nakon što postoji IP adresa na preklopniku, on se mora konfigurirati za prihvaćanje udaljene veze. I u ovom slučaju koristit će se primjer *Telnet* protokola.



Slika 78. Konfiguracija preklopnika za udaljeni pristup pomoću telnet protokola

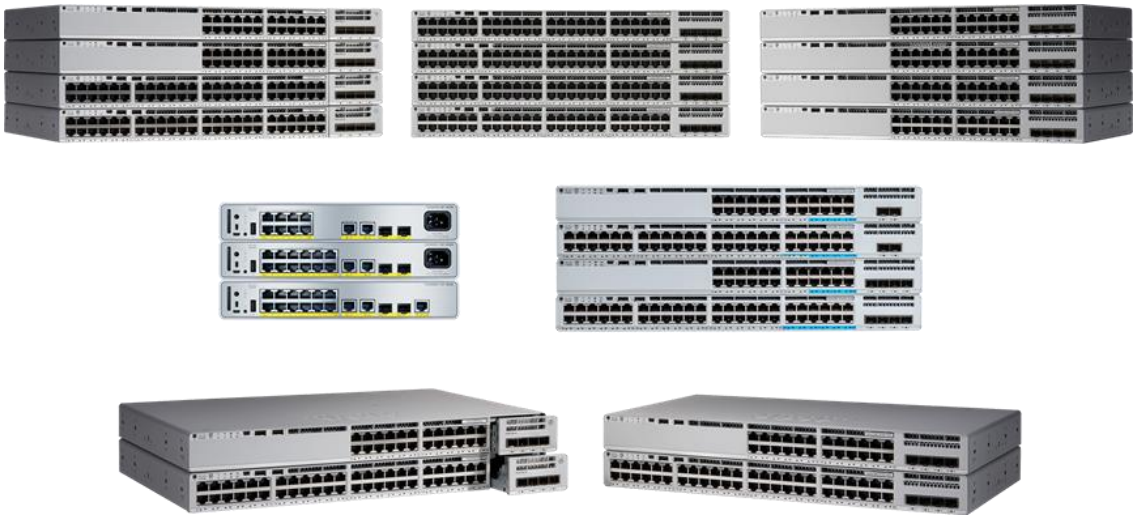
Testiranje se izvodi s računala iz naredbenog retka (CMD.exe) naredbom *telnet [IP adresa preklopnika].* Početne postavke *Windows* operativnog sustava ne omogućuju *Telnet* pa to treba promijeniti u kontrolnoj ploči u: „Turn Windows features on or off“, kako je prikazano na slici 79.



Slika 79. Omogućavanje Telnet protokola na Windows računalu za povezivanje na stvarne mrežne uređaje

2.2. Definiranje L2 računalne mreže — preklopnik

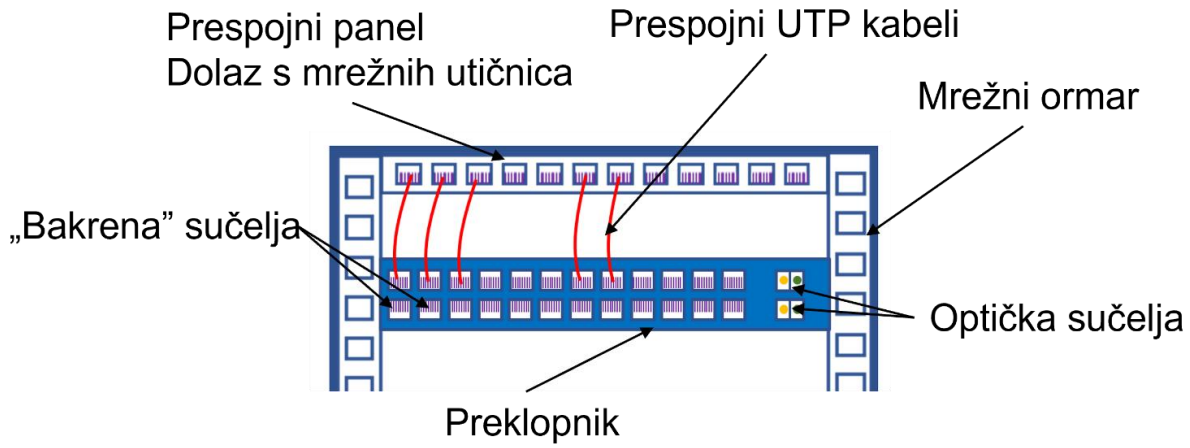
L2 mreža (*Layer* 2) prvenstveno podrazumijeva mrežno okruženje u kojem se komunikacija odvija na temelju MAC adresa. Takve mreže zovu se i lokalne računalne mreže (LAN). Temelj za njihovu izradu mrežni su uređaji koje zovemo preklopnici (engl. *Switch*). Preklopnik se može prepoznati po velikom broju sučelja na koja se povezuju krajnji uređaji, kao što su računala, IP telefoni, IP kamere, pametni televizori, poslužitelji, drugi mrežni uređaji (usmjernici, vatrozidi, uređaji za raspored opterećenja itd.). Velike mreže poput mreža banaka, tvornica, a posebno bolnica, sveučilišnih kampusa ili podatkovnih centara sastoje se od velikog broja preklopnika, ponekad i njih više tisuća međusobno povezanih.



Slika 80. Primjer preklopnika tvrtke Cisco serija 9200

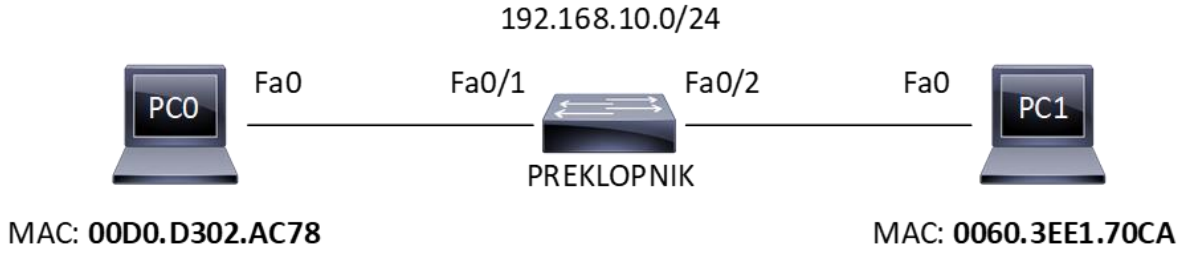
Postoje različite vrste preklopnika, različitih karakteristika, funkcionalnosti i tehnologija koje podržavaju, a to sve ovisi o njihovoj namjeni i mjestu unutar sustava IKT-a. Preklopnik koji se koristi na pristupnom sloju računalne mreže neće imati iste karakteristike kao onaj koji se koristi u sloju jezgre mrežne infrastrukture. Ovisno o

karakteristikama preklopnika, cijene će također varirati od svega 10-ak eura za 8-portni metalni preklopnik za kućnu upotrebu brzine 1Gbps do ultra brzih preklopnika koji se koriste za povezivanje podatkovnih centara brzinama više Tbps i koštaju stotine tisuća eura. U računalnim mrežama obično se ne vide mrežni uređaji jer se nalaze u kontroliranim mikroklimatskim uvjetima zaštićeni od potencijalnih napadača. Ono što se vidi, mrežne su utičnice u zidovima u koje kabelima spajamo krajnje uređaje na računalnu mrežu, i upravo su one kroz zidove, pod ili strop povezane na prespojne panele u mrežnim ormarima, a onda su prespojnim kabelima povezane svaka u jedno sučelje na preklopniku (prikazano na slici 81.).



Slika 81. Povezivanje prespojnog panela i preklopnika u mrežnom ormaru

Preklopnik održava tablicu MAC adresa kako bi prosljeđivao okvire između uređaja u lokalnoj računalnoj mreži. U tablici MAC adresa bitni elementi su MAC adresa i sučelje na koje je povezan uređaj koji koristi tu MAC adresu. Na slici 82. dva su računala povezana na preklopnik i svako ima svoju MAC adresu.



Slika 82. Shema male LAN mreže s dva računala i preklopnikom

Prije nego računala komuniciraju na mreži, preklopnik ima tablicu MAC adresa bez ikakvih zapisa. MAC tablicu popunjava na temelju izvorišne MAC adrese, što znači da

preklopnik kada primi prvi okvir od bilo kojeg računala, bez obzira koje je odredište, unosi to u tablicu MAC adresa, i taj će zapis kasnije koristiti za povratni promet.

```
Switch#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

Switch#
```

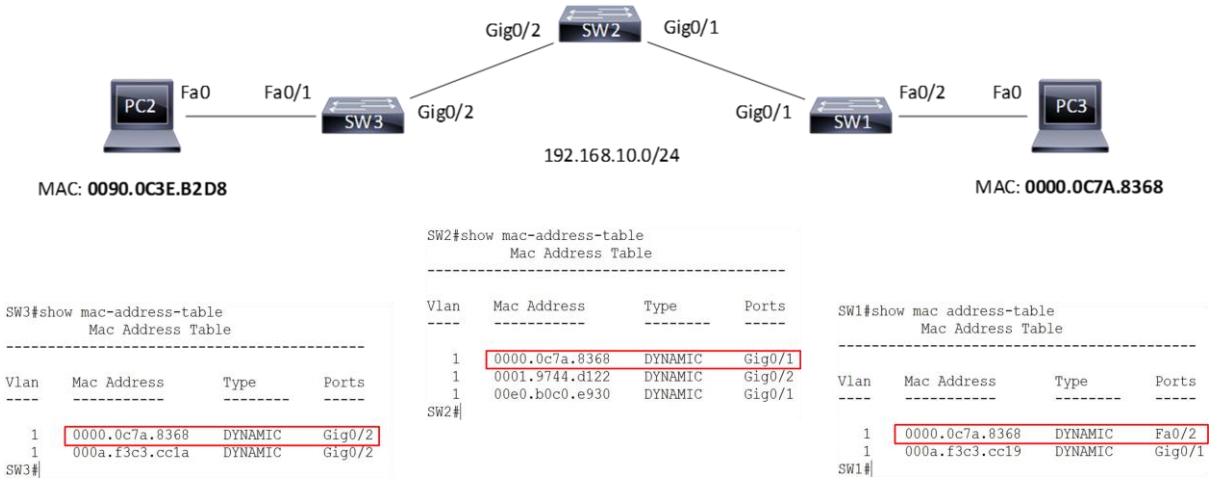
Slika 83. Početno stanje tablice MAC adresa na preklopniku

Nakon što računalo pošalje bilo kakav promet na mrežu, npr. *ping* prema nekoj IP adresi (koja ne mora ni postojati, bitno je da je računalo konfigurirano IP postavkama), preklopnik će zapisati MAC adresu na pripadajuće sučelje. U stvarnom svijetu računalo neprestano pokušava komunicirati, koristi različite aplikacije i protokole poput DHCP, ARP; različite aplikacije pokušavaju kontaktirati poslužitelje kako bi se ažurirale i slično, tako da je vrlo teško na preklopniku vidjeti tablicu MAC adresa bez ikakvih zapisa. U mrežnom simulatoru je to puno lakše postići.

```
Switch#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0060.3ee1.70ca    DYNAMIC     Fa0/2
Switch#
```

Slika 84. MAC tablica na preklopniku nakon što je računalo poslalo okvir na mrežu

Ako se mreža sastoji od više povezanih preklopnika, svi preklopnici kojima prolazi okvir koji je poslalo neko računalo, napravit će zapis u svojoj tablici MAC adresa. Ta karakteristika L2 mreže (mreža koju čine preklopnici) osnova je za razne napade na L2 mrežu, poput poplave MAC adresa (engl. *MAC Address Flooding*). Rezultat komunikacije računala u tablici MAC adresa svih preklopnika u mreži vidljiv je na slici 85.



Slika 85. Izgled tablica MAC adresa na preklopnicima u mreži s više preklopnika

2.3. Definiranje L2 računalne mreže — VLAN

Preklopnici su, dakle, uređaji koji se koriste da bi se krajnji uređaji u nekoj organizaciji povezali na računalnu mrežu. Kada se povežu, svi ti uređaji mogu međusobno komunicirati koristeći MAC adrese, a to znači da se svi uređaji međusobno „vide“. U kontekstu sigurnosti to znači da se svi uređaji međusobno mogu i „napasti“, što, naravno, nije poželjno. No, prije govora o sigurnosnim aspektima L2 mreže, treba se podsjetiti svrhe sustava IKT-a, a tako i računalne mreže. To je podrška poslovanju organizacije pa sustav IKT-a i računalna mreža, u određenoj mjeri, reflektiraju organizacijsku strukturu tvrtke, što je najbolje vidljivo upravo u L2 mreži koju čine preklopnici.

Za mapiranje organizacijske strukture tvrtke na računalnu mrežu koristi se koncept VLAN-ova (engl. *Virtual Local Area Networks*). Važan je to koncept i treba posvetiti adekvatnu pažnju i vrijeme njegovu razumijevanju.

Preklopnik u početnom stanju na sebi ima jedan ili više predefiniranih VLAN-ova. Na prethodnim slikama prvi element na ispisu upravo je oznaka VLAN-a, i to predefinirani VLAN 1. U početku sva sučelja na preklopniku su dio VLAN-a 1 i zato sva računala koja se povežu na preklopnik mogu međusobno komunicirati bez zapreka. Ali, kao što je već rečeno, u poslovnim organizacijama ne želi se da sva računala, a to znači svi korisnici tih računala budu u istom VLAN-u i da mogu međusobno direktno komunicirati koristeći MAC adrese. Razlozi leže u lakšem upravljanju mrežom, održavanju i uklanjanju uzroka problema (TSHOOT), ograničavanju sveodredišnih (engl.

Broadcast) domena, do povećane stabilnosti, skalabilnosti i sigurnosti računalne mreže. Sigurno nije poželjno da članovi uprave bolnice budu u istoj L2 mreži kao i njezini posjetitelji, i da se na taj način ugrozi ukupno poslovanje bolnice. To se još više odnosi na organizacije poput banaka, policije, vojske, podatkovnih centara, telekomunikacije i slično. Također, ne želi se da sveodredišni promet u mreži dođe do 20 000 računala ako se može ograničiti na 200 računala (npr. poruke ARP zahtjeva).

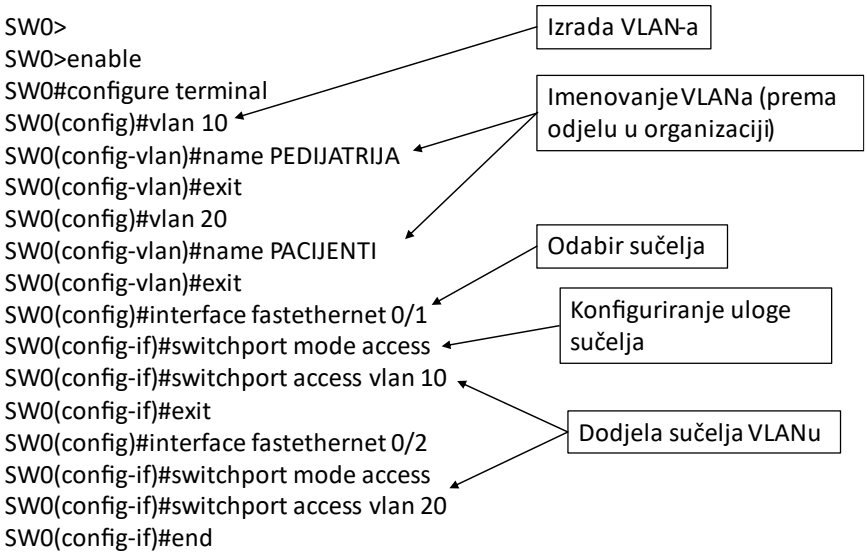
Sva računala koja su dio iste L2 domene, a to bi upravo bio neki VLAN, mogu međusobno direktno komunicirati, dok računala koja su u različitim L2 domenama moraju za to koristiti L3 uređaj, poput usmjernika.

Računala unutar istog VLAN-a moraju biti dio iste podmreže, što znači da svaki VLAN mora imati određenu posebnu podmrežu kako bi računalna mreža kao cjelina mogla ispravno funkcionirati.

Čak i kada računala koja su u istom VLAN-u nisu dio iste podmreže, ta će računala i dalje vidjeti promet drugih računala (npr. poruke ARP zahtjeva, DHCP poruke otkrivanja i drugi sveodredišni promet), a onda samo treba promijeniti IP adresu, i mogu ostvariti punu komunikaciju.

Preklopnik kao ograničen resurs može se zamisliti kao nogometni teren. Za jednu nogometnu utakmicu u nekom trenutku jedan je veliki teren dovoljan, međutim, želi li se istovremeno igrati dvije nogometne utakmice, a ne može se samo tako izgraditi novi nogometni teren, onda se postojeći, koji je jedna cjelina, može podijeliti na dva dijela i istovremeno igrati dvije utakmice ili raditi bilo što na drugoj polovici igrališta. Dakle, postojeći ograničeni resurs logički se dijeli na dva dijela istih karakteristika i funkcionalnosti.

U slučaju preklopnika logičkom podjelom nastaju logičke cjeline koje nazivamo virtualnim lokalnim računalnim mrežama (engl. *Virtual Local Area Network — VLAN*) i tim logičkim cjelinama dodjeljuju se određena sučelja prema potrebi pojedinog odjela unutar organizacije. Na slici 86. prikazane su naredbe koje se koriste na preklopniku kako bi se kreirala dva VLAN-a i dodijelilo po jedno sučelje u svaki VLAN.



Slika 86. Primjer konfiguriranja VLAN-a

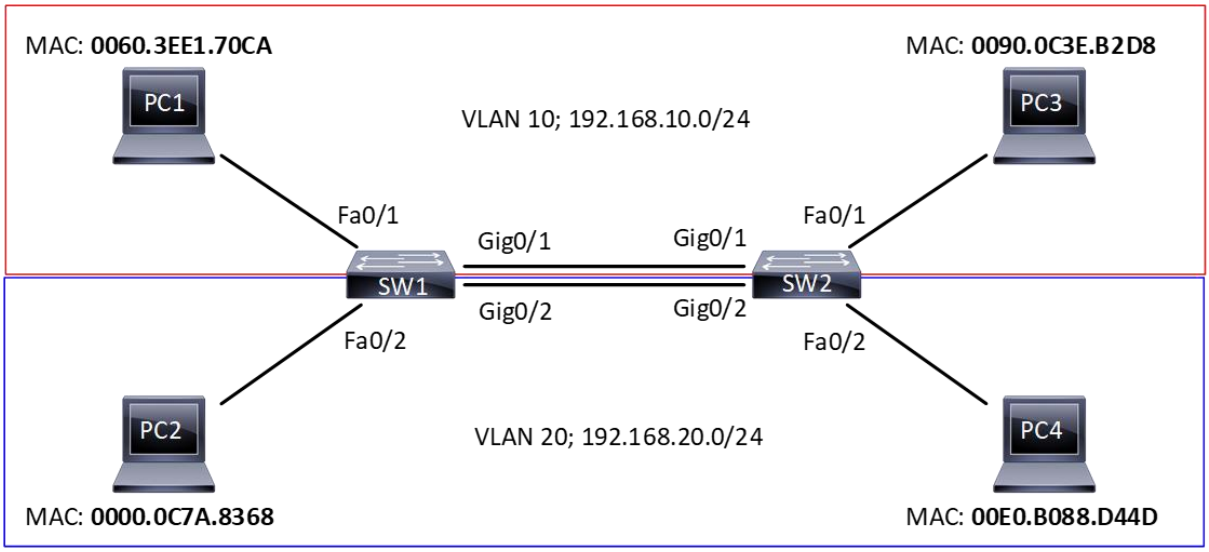
Odvija li se to zaista prema očekivanjima, može se testirati na sljedeći način:

1. Konfigurirati računala s IP adresama u istoj podmreži
2. *Pingati* s jednog računala drugo i uvjeriti se da računala mogu međusobno komunicirati
3. Izraditi VLAN (npr. VLAN 10)
4. Jedno računalo prebaciti u novi VLAN (npr. VLAN 20)
5. Ponovno *pingati* s jednog računala drugo i potvrditi da komunikacija više nije moguća.

Iako su oba računala i dalje u istoj podmreži, recimo 192.168.10.0/24, komunikacija nije moguća, jer su na preklopniku dio različitih logičkih cjelina koje zovemo VLAN-ovi. Ispravno bi bilo ili da oba računala postanu dio istog VLAN-a (npr. VLAN 10), ili da se jedno računalo prebaci u novi VLAN (npr. VLAN 20), ali u tom slučaju nužno je da to računalo bude dio iste podmreže, npr. 192.168.20.0/24.

Konfiguracija prikazana na slici 86. i funkcioniranje te male mreže odvija se kad su oba računala spojena na isti preklopnik. U slučaju da ih ima više, npr. dva, na oba preklopnika treba kreirati iste VLAN-ove i povezati ta dva preklopnika s dva kabela koristeći po dva sučelja na svakom preklopniku, tako da su dva međusobno povezana sučelja na svakom preklopniku dio istog VLAN-a – na taj se način proširuje doseg jedne logičke L2 domene (npr. VLAN 10 i VLAN 20).

To je vidljivo na slici 87. gdje se povezuju po dva računala unutar istog VLAN-a, jedno na svakom preklopniku. Neučinkovit je pristup kada za svaki VLAN trebamo po jedno sučelje za vezu između preklopnika jer u slučaju da ima desetak VLAN-ova, što nije neuobičajeno u praksi, broj potrebnih sučelja bi vrlo brzo prešao broj dostupnih na preklopniku. Kako bi se adresirao taj problem, osmišljen je mehanizam za označavanje okvira koji se prosljeđuju između preklopnika s oznakom VLAN-a kojem ti okviri pripadaju (gdje im je izvor). Protokol koji se danas najčešće koristi jest 802.1q. Cilj je označavanja okvira oznakom VLAN-a smanjiti količinu potrebnih veza između preklopnika na samo jednu, a sav promet koji ide posredstvom te veze, sa sobom nosi oznaku VLAN-a kojem pripada. Tako odredišni preklopnik zna dostaviti svaki okvir na pravo odredište.



Slika 87. Povezivanje računala unutar istog VLAN-a pomoću više preklopnika

Da bi računala PC1 i PC3 na slici 87. mogla komunicirati, potrebno je da na preklopniku SW1 sučelja Fa0/1 i Gig0/1 budu dio VLAN-a 10, a istovremeno je potrebno da sučelja Gig0/1 i Fa0/1 na preklopniku SW2 također budu dio VLAN-a 10. Isto vrijedi i za komunikaciju između PC2 i PC4, gdje sučelja FA0/2 i Gig0/2 na preklopniku SW1 i sučelja Gig0/2 i Fa0/2 na preklopniku SW2 moraju biti dio VLAN-a 20. Tehnički gledano podmreža bi mogla biti ista za oba VLAN-a, ali u praksi to se nikada ne radi ako se želi funkcionalna mreža. Kada bi podmreža bila ista, čak i uz dodan usmjernik na tu topologiju, računala u oba VLAN-a imala bi velikih problema s komunikacijom izvan svog VLAN-a.

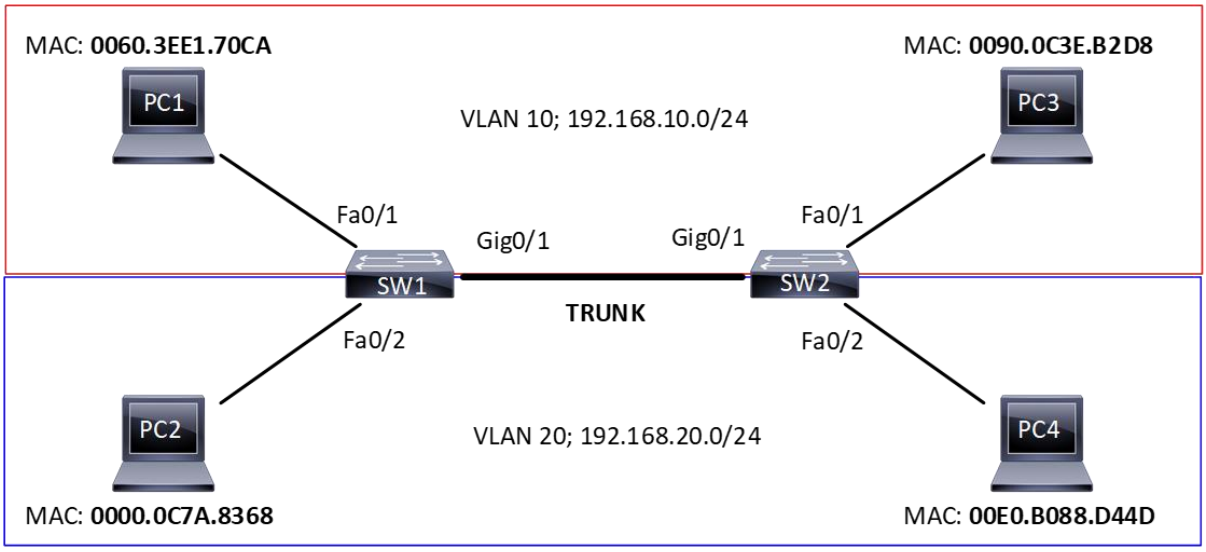
Kada je mreža povezana kao što je prikazano na slici 87., i kada računala unutar istog VLAN-a komuniciraju, tablica MAC adresa na preklopnicima jasno pokazuje kojem VLAN-u pripadaju pojedina sučelja na koja su povezana računala i drugi uređaji kao što su preklopnici, usmjernici i slično.

```
SW1#sh mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

  10    000a.f3c3.cc19    DYNAMIC     Gig0/1
  10    0060.3ee1.70ca    DYNAMIC     Fa0/1
  10    0090.0c3e.b2d8    DYNAMIC     Gig0/1
  20    0000.0c7a.8368    DYNAMIC     Fa0/2
  20    000a.f3c3.cc1a    DYNAMIC     Gig0/2
  20    00e0.b088.d44d    DYNAMIC     Gig0/2
SW1#
```

Slika 88. Tablica MAC adresa na preklopniku SW1

Narančasto su označene MAC adrese koje je preklopnik na slici 37. naučio na temelju okvira koje su poslala računala koja nisu direktno povezana na preklopnik SW1, ali to je očekivano, jer preklopnici svoju tablicu MAC adresa popunjavaju na temelju izvorišne MAC adrese, tako da čim prvi okvir dođe do preklopnika, odmah se zapisuje u tablicu MAC adresa kao novi unos.

U praksi se gotovo nikad neće preklopnici povezivati vezama koje su svaka dio jednog VLAN-a, nego će se konfigurirati sučelja između preklopnika tako da označavaju okvire oznakama pripadajućeg VLAN-a. Takva se sučelja još zovu i TRUNK sučelja ili „tagirana“ sučelja. Shema povezivanja uređaja će u tom slučaju izgledati kako je prikazano na slici 89.



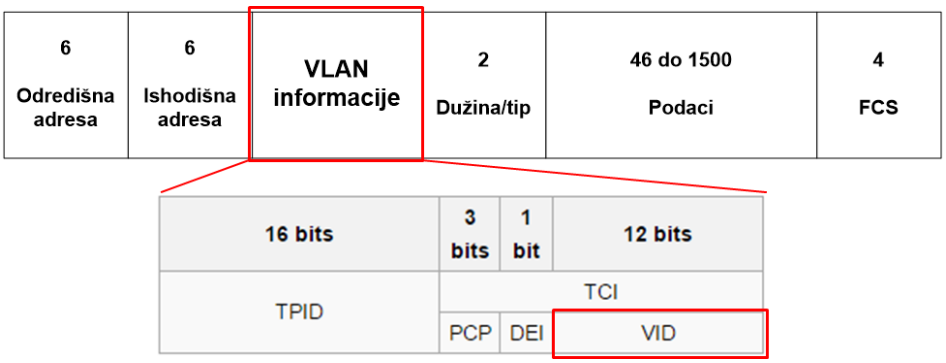
Slika 89. Povezivanje dva preklopnika TRUNK vezom

Kada su preklopnici tako povezani, svaki okvir koji se šalje između preklopnika pomoću TRUNK linka, dobiva oznaku VLAN-a kojem pripada. Okviri se jedino označavaju oznakama VLAN-a na TRUNK sučeljima, a nikako na sučeljima prema računalima koja su u pristupnom modu (engl. *Access Mode*). Kada bi se označavali oznakom VLAN-a na sučeljima prema računalima, takva komunikacija ne bi bila uspješna jer računala ne koriste 802.1q protokol i ne mogu „čitati“ tako dodane oznake okvirima.

```
SW1(config-if)#switchport mode trunk
Command rejected: An interface whose trunk encapsulation
is "Auto" can not be configured to "trunk" mode.
SW1(config-if)#switchport trunk encapsulation dot1q
SW1(config-if)#switchport mode trunk
```

Slika 90. Prikaz konfiguracije TRUNK sučelja na preklopniku

Na slici 90. vidi se da preklopnik nije prihvatio ulogu TRUNK sučelja, jer preklopnik ulogu sučelja pregovara s drugim preklopnikom s kojim je povezan. Ovdje se govori o pregovaranju protokola za označavanje okvira oznakom VLAN-a, a neki preklopnici, osim 802.1q protokola, podržavaju i druge, pa je najprije potrebno ručno definirati da će preklopnik koristiti 802.1q protokol, a tek onda sučelje može biti konfigurirano kao TRUNK.

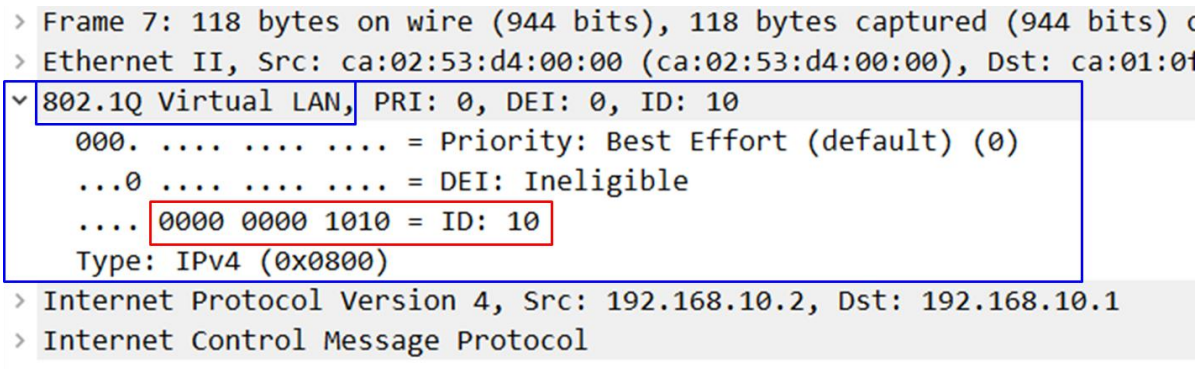


Slika 91. Izgled 802.1Q zaglavlja koje se dodaje okviru na TRUNK vezama

Elementi 802.1q zaglavlja su:

- *Tag Protocol Identifier* (TPID) – 16-bitna vrijednost postavljena na 0 x 8100 koja upućuje na IEEE 802.1Q okvir;
- *Tag Control Information* (TCI)
  - ✓ *Priority Code Point* (PCP) – potreban kada se koristi kvaliteta usluge u mreži (engl. *Quality of Service* – QoS)
  - ✓ *Drop Eligible Indicator* (DEI) – potreban kada se koristi kvaliteta usluge u mreži
  - ✓ *VLAN Identifier* (VID) – 0 - 4095, 0 x 000 (VLAN 0) i 0 x FFF (VLAN 4095) rezervirane su vrijednosti.

U ovom je trenutku jedino važno polje 802.1q zaglavlja VLAN ID, koji preklopnici koriste kako bi znali kamo proslijediti pojedini okvir s obzirom na svoju tablicu MAC adresa.



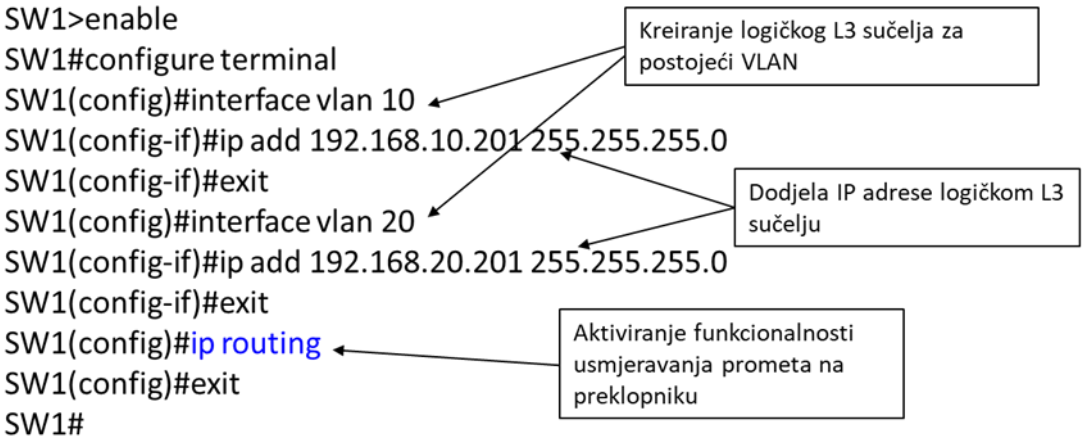
Slika 92. Prikaz 802.1q zaglavlja u Wireshark alatu — „uhvaćeno“ na TRUNK vezi

2.4. Usmjeravanje između VLAN-ova — IVR (engl. Inter VLAN Routing)

Kada je L2 mreža logički segmentirana koristeći VLAN-ove te odgovara organizacijskoj strukturi tvrtke tako da svaki odjel bude odvojen unutar svog VLAN-a i koristi jedinstvenu podmrežu, sada tim podmrežama treba omogućiti međusobnu komunikaciju. To se može napraviti na dva načina: jedan je usmjeravanje prometa između VLAN-ova na preklopniku koji ima te mogućnosti, a drugi način je da se u mrežu poveže usmjernik koji će osim povezivanja VLAN-ova međusobno, omogućiti komunikaciju računalima iz svih VLAN-ova prema ostatku sustava IKT-a i prema internetu.

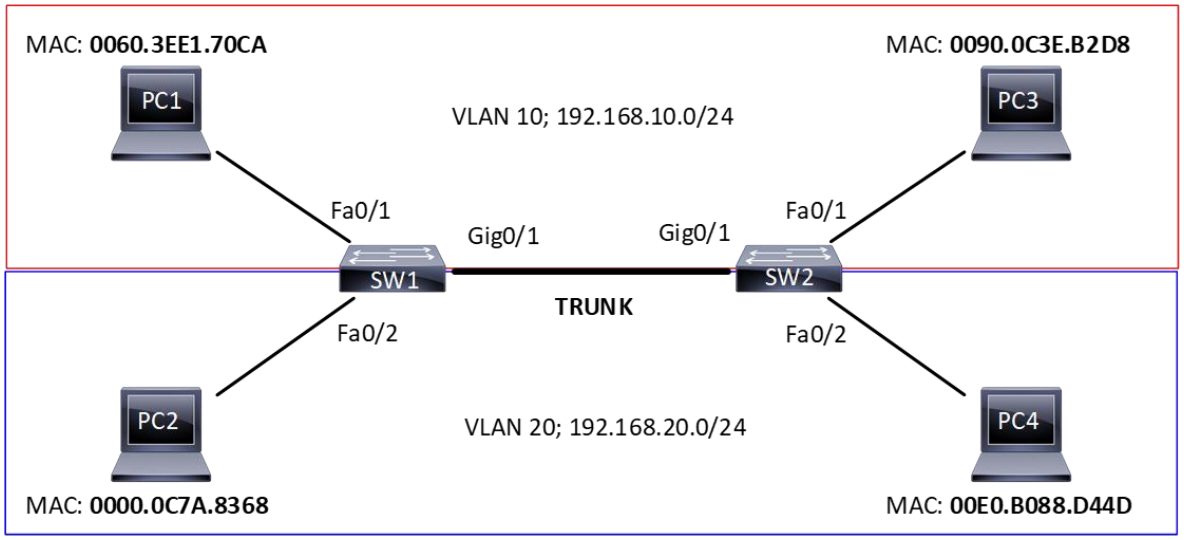
U prvom je slučaju nekoliko stvari potrebno istaknuti:

1. Preklopnik mora podržavati usmjeravanje prometa, tzv. višeslojni (engl. *Multilayer*) preklopnici
2. Ta mogućnost mora biti aktivirana – naredba: “ip routing”
3. Preklopnik mora imati konfigurirane VLAN-ove za koje će usmjeravati promet
4. Preklopnik mora imati L3 (*Layer* 3) sučelje za svaki VLAN s IP adresom
5. Ako preklopnik usmjerava promet za cijelu mrežu, između svih preklopnika moraju biti konfigurirane TRUNK veze
6. U mreži s više preklopnika treba odabrati jedan koji će usmjeravati za sve VLAN-ove (obično u sloju jezgre mreže).



Slika 93. Primjer konfiguracije usmjeravanja između VLAN-ova na preklopniku

Posljednja je točka (6.) vrlo diskutabilna. Iako će takva mreža raditi i ima dosta implementacija u praksi, bolje bi bilo ugraditi usmjernik ili vatrozid u mrežu na sloju jezgre i izmjestiti zadaću usmjeravanja prometa s preklopnika na usmjernik.



Slika 94. Prikaz male logički segmentirane računalne mreže koju čine samo preklopnici

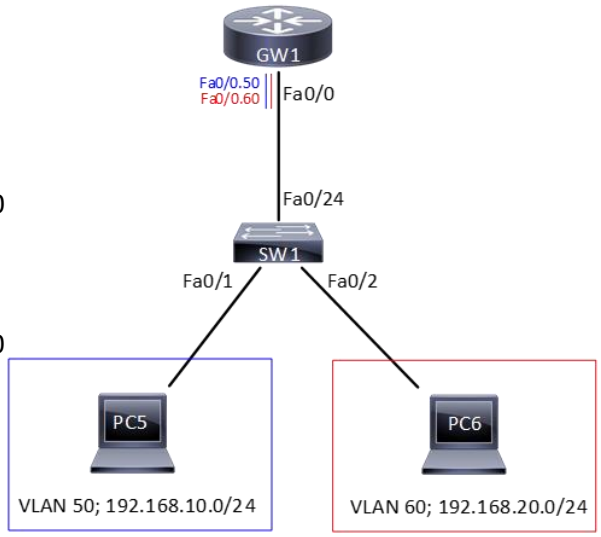
Gledajući sliku 94. postavlja se pitanje koji bi od dva preklopnika trebao usmjeravati promet između VLAN-a 10 i VLAN-a 20. Svejedno je, ali treba imati na umu da mreža kakva je prikazana na slici, nije uobičajena u praksi. Ona bi bila mreža malog ureda te preklopnici koji bi se koristili, ne bi imali ništa od naprednih funkcionalnosti, već bi samo služili za povezivanje uređaja u toj mreži na univerzalni uređaj koji je ustupio telekomunikacijski operator za pristup internetu.

S druge strane, mreža prikazana na slici 95. zahtijeva promišljanje o tome kako riješiti logičku segmentaciju i usmjeravanje prometa između VLAN-ova i slično. Takve su mreže tipične za okruženja kao što su bolnice, škole, fakulteti, ali i razne druge poslovne organizacije, poput tvornica, banaka itd.

Drugi slučaj usmjeravanja prometa između VLAN-ova podrazumijeva korištenje usmjernika, i to najčešće implementiran kao „usmjernik na štapu“ (engl. *Router on a Stick*). To podrazumijeva da je usmjernik povezan obično s preklopnikom u sloju jezgre mreže jednim sučeljem, a to sučelje je onda logički segmentirano na više podsučelja, koji su dio jednog VLAN-a. IP adresa koja je konfigurirana na svakom podsučelju izlaz je iz VLAN-a za sva računala unutar njega. U toj implementaciji bitno je na svakom podsučelju unutar jednog fizičkog sučelja na usmjerniku uključiti 802.1q protokol te

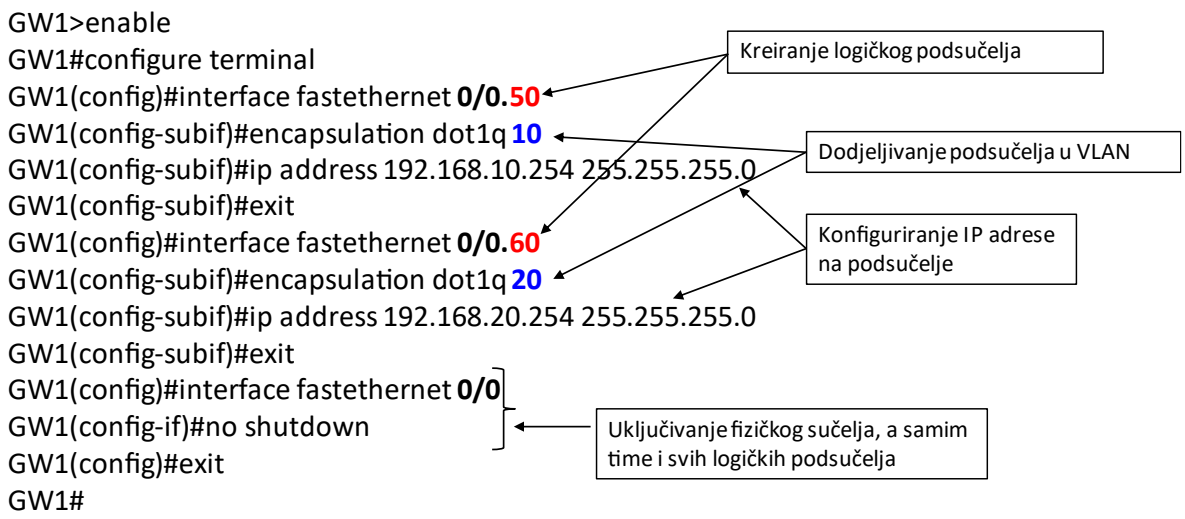
uključiti fizičko sučelje. Na preklopniku sučelje koje veže preklopnik s usmjernikom treba biti u TRUNK načinu.

```
Password:
SW1#configure terminal
SW1(config)#vlan 50
SW1(config-vlan)#name NEFROLOGIJA
SW1(config-vlan)#exit
SW1(config)#vlan 60
SW1(config-vlan)#name ONKOLOGIJA
SW1(config-vlan)#exit
SW1(config)#interface fastethernet 0/1
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 50
SW1(config-if)#exit
SW1(config)#interface fastethernet 0/2
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 60
SW1(config-if)#exit
SW1(config)#interface fastethernet 0/24
SW1(config-if)#switchport mode trunk
SW1(config-if)#exit
SW1(config)#exit
SW1#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```



Slika 95. Konfiguracija preklopnika kao „usmjernik na štapu“ za dva VLAN-a

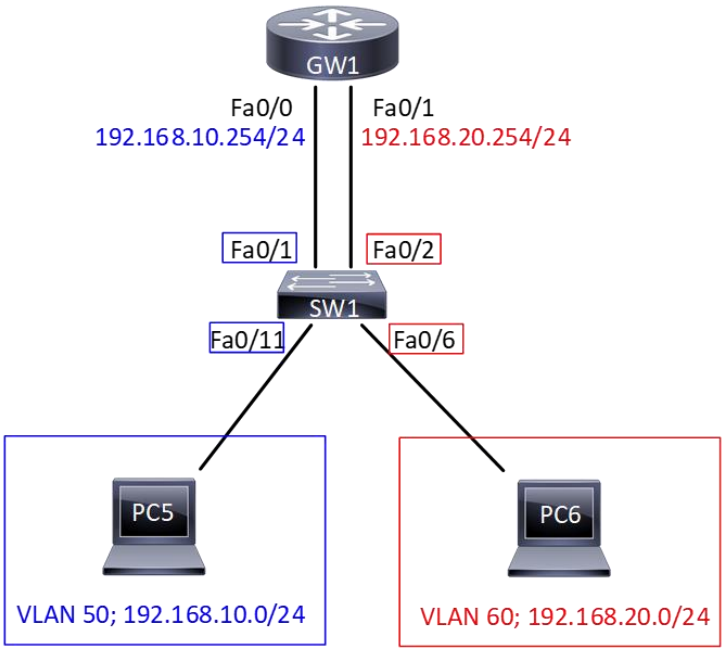
Kao što je vidljivo na slici 95., na preklopniku treba jasno definirati koja su pristupna sučelja, a koja su TRUNK sučelja. U mrežama su TRUNK sučelja gotovo uvijek između preklopnika i, često, prema usmjerniku. Kada je implementacija „usmjernik na štapu“, sučelje na preklopniku u koje se spaja usmjernik mora biti TRUNK kako bi se okviri koji se šalju prema usmjerniku mogli označavati oznakom VLAN-a.



Slika 96. Konfiguracija usmjernika kao „usmjernik na štapu“ za dva VLAN-a

Na slici 96. vidi se konfiguracija usmjernika za mrežni promet između dva ili više VLAN-ova. Sama oznaka podsučelja (npr. *fastethernet* 0/0.50) može biti bilo koji broj (.50, .60 i slično), ali broj koji upisujemo iza naredbe: „encapsulation dot1q“, mora odgovarati oznaci VLAN-a koji će koristiti neko sučelje za izlaz iz podmreže. Također obavezno treba uključiti fizičko sučelje jer bez toga neće raditi ni logička podsučelja.

Naravno, postoji i implementacija u kojoj se na usmjerniku koristi po jedno fizičko sučelje za svaki VLAN i gdje je usmjernik povezan s preklopnikom pristupnim sučeljem. U slučaju da se koriste dedicirana sučelja na usmjerniku, kao što je prikazano na slici 97., svako od tih fizičkih sučelja treba imati IP adresu koja je dio podmreže za pripadajući VLAN i treba uključiti fizičko sučelje na usmjerniku. Također, na strani preklopnika sučelje u koje je spojen usmjernik mora biti pristupno sučelje koje je dio VLAN-a za koji će usmjernik djelovati kao pristupnik za izlaz na internet (engl. *Gateway*). Iako je takav scenarij tehnički izvediv, rijetko se koristi jer nije skalabilno rješenje i nema opravdanja za takvo „slobodno“ korištenje ograničenog broja fizičkih sučelja na usmjerniku.



Slika 97. Implementacija kada usmjernik ima jedno dedicirano sučelje za svaki VLAN

2.5. VTP — VLAN Trunking Protocol

VTP (engl. *VLAN Trunking Protocol*) je protokol *Cisco*-a za dinamičku konfiguraciju VLAN-ova na preklopnicima. Omogućava konfiguraciju VLAN-ova na jednom preklopniku i prosljeđivanje te konfiguracije na ostale preklopnike na kojima je pokrenut VTP. VTP poruke prenose se isključivo TRUNK vezama.

VTP definira:

- VTP područje (engl. *VTP Domain*) unutar kojega se razmjenjuju informacije o VLAN-ovima;
- načine rada preklopnika, odnosno ulogu preklopnika u razmjeni informacija;
- tipove poruka koje se razmjenjuju.

Da bi dva preklopnika mogla razmjenjivati podatke o VLAN-ovima, trebaju biti u istoj domeni, odnosno u domeni istog imena. Preklopnici u različitim domenama ne razmjenjuju informacije o konfiguracijama VLAN-a.

Preklopnik na kojemu je pokrenut VTP može raditi u jednom od triju načina:

- VTP poslužitelj – samo se u poslužiteljskom načinu mogu dodavati, uređivati i brisati VLAN-ovi. Informacije o VLAN-ovima prosljeđuju se susjednim preklopnicima;

- VTP klijent – preklopnik u klijentskom načinu rada konfigurira se na temelju dobivenih podataka od VTP poslužitelja (ažurira svoju bazu VLAN-ova) i prosljeđuje podatke susjednim preklopnicima. Na preklopniku koji je u VTP klijentskom načinu rada, ne mogu se kreirati, uređivati ili brisati VLAN-ovi;
- transparentni VTP – ne sudjeluje aktivno u VTP procesu. Preklopnik u transparentnom načinu rada ne konfigurira se prema dobivenim podacima, već dobivenu konfiguraciju samo prosljeđuje ostalim preklopnicima. Na takvom je preklopniku moguće kreirati samo lokalne VLAN-ove koji neće biti distribuirani protokolom VTP.

Slika 98. Prikaz povezanih preklopnika s različitim VTP ulogama

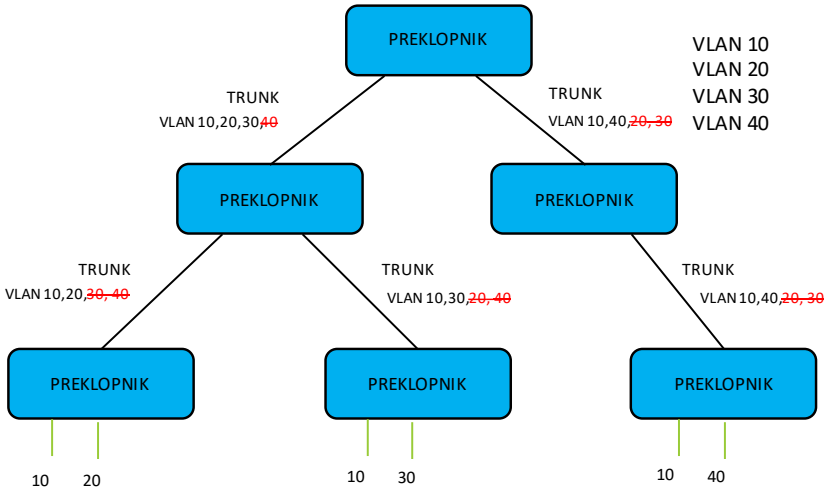
Preklopnik SW1 na slici 98. poslužitelj je, i na njemu će administrator konfigurirati VLAN-ove. Ta će konfiguracija VTP porukama biti proslijeđena svim preklopnicima u mreži. VTP klijenti će ažurirati svoju konfiguraciju VLAN-a i VTP poruke proslijediti susjednim preklopnicima. VTP transparentni preklopnik S4 dobit će VTP poruke, ali se neće konfigurirati; samo će susjedima proslijediti poruke.

Kako bi preklopnici znali je li konfiguracija koju su dobili od susjeda novija od one koju već imaju, VTP paketi imaju polje oznake verzije konfiguracije (engl. *Config Revision*). Svaki put kada dođe do promjene konfiguracije na preklopniku koji je VTP poslužitelj, broj u polju *Config Revision* poveća se za jedan. Kada preklopnik dobije poruku s VLAN konfiguracijom, uspoređuje broj konfiguracije koju ima s brojem konfiguracije koju je dobio. Ako je dobiveni broj konfiguracije veći, preklopnik ažurira svoju bazu podataka VLAN-a, a ako je manji, ignorira dobivenu informaciju.

VTP *PRUNNING*

Uključivanjem VTP orezivanja (engl. *Prunning*) dopuštaju se samo aktivni VLAN-ovi na TRUNK vezi. Okviri s VLAN-ovima koji ne postoje u određenoj grani, ne prosljeđuju se u tu granu. Preklopnici se međusobno obavještavaju koji su im VLAN-ovi aktivni (aktivni je VLAN onaj kojemu je pridruženo barem jedno sučelje). Tako preklopnici znaju koje okvire ne treba slati na TRUNK vezu. Na slici 99. prikazana je mreža s četiri

VLAN-a. VTP orezivanje omogućeno je i vidljivo da se okviri određenog VLAN-a ne prosljeđuju u grane gdje tog VLAN-a nema.



Slika 99. Princip rada VTP orezivanja

Za konfiguraciju VTP protokola u mreži, treba mrežu pripremiti tako da se veze između preklopnika konfiguriraju kao TRUNK veze. TRUNK je preduvjet za funkcioniranje VTP protokola i bez TRUNK veza VTP neće raditi očekivano. Također treba odlučiti koji će preklopnik u mreži biti poslužitelj (predefinirano je svaki preklopnik VTP poslužitelj), koji *Transparent*, a koji će preklopnici biti klijenti. Kada je preklopnik u klijentskom modu, više se na njemu ne može dodavati ni uklanjati VLAN-ove, niti mijenjati VTP verziju. Najprije ga treba vratiti u ulogu VTP poslužitelja da bi se mogle raditi izmjene. U mreži će većinom biti jedan ili dva VTP poslužitelja, i to će biti obično preklopnici sloja jezgre, dok će svi ostali biti klijenti, što je ponekad nekoliko desetaka, pa i stotina preklopnika. Potrebno je odabrati i verziju VTP protokola, ovisno o potrebama, ali svakako bi minimalna trebala biti verzija 2, ili u novijim mrežama VTP-a verzija 3, koja podržava više mogućnosti, kao što je enkripcija lozinki, veći broj VLAN-ova, privatni VLAN-ovi i slično.

Prilikom konfiguracije prvo što treba odlučiti naziv je domene. Predefinirano je taj naziv VTP domene *NULL*, odnosno nema naziva domene. Prvi put kada se promijeni naziv VTP domene, ako je konfigurirano TRUNK sučelje između preklopnika, svi će preklopnici u mreži prihvatiti taj naziv i postati dio nove domene. Potreban je oprez prilikom postavljanja naziva domene kako ne bi trebalo ručno mijenjati naziv na svakom preklopniku.

```
Switch#show vtp status
VTP Version capable             : 1 to 2
VTP version running             : 1
VTP Domain Name                 :
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 0060.5C7C.5400
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Local updater ID is 0.0.0.0 (no valid interface found)

Feature VLAN :
--------------
VTP Operating Mode                : Server
Maximum VLANs supported locally   : 255
Number of existing VLANs          : 5
Configuration Revision            : 0
MD5 digest                        : 0x7D 0x5A 0xA6 0x0E 0x9A 0x72 0xA0 0x3A
                                    0xF0 0x58 0x10 0x6C 0x9C 0x0F 0xA0 0xF7

Switch#
```

Slika 100. Početne postavke na preklopniku u simulatoru Cisco Packet Tracer — razlikuje se ovisno o modelu preklopnika

Predefinirano postoji na preklopnicima 5 VLAN-ova u mrežnom simulatoru *Packet Tracer*. Jedan je predefiniran i u njemu se nalaze sva sučelja na preklopniku, a nosi oznaku VLAN 1. Taj VLAN ne može se obrisati. Ostala su se četiri VLAN-a prije koristila, dok danas nemaju nikakvu praktičnu ulogu, tako da ih se neće vidjeti na novim preklopnicima u mrežama.

```
Switch#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gig0/2
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
Switch#
```

Slika 101. Ispis naredbe show VLAN brief na preklopniku

Topologija na kojoj će biti pojašnjeno kako VTP radi već je prikazana na slici 98. Dakle, SW1 je VTP poslužitelj i na njemu će se kreirati svi VLAN-ovi i njihovi nazivi. Preklopnici SW2 i SW4, kao klijenti, prihvatit će informacije o VLAN-ovima, dok će SW3 samo proslijediti te informacije, ali ih neće uzeti sebi, niti će ih koristiti. Stoga SW3 neće znati koji VLAN-ovi postoje u mreži, osim ako se ručno konfigurira.

```
SW1(config)#vtp mode server
Device mode already VTP SERVER.
---
SW2(config)#vtp mode client
Setting device to VTP CLIENT mode.
----
SW3(config)#vtp mode transparent
Setting device to VTP TRANSPARENT mode.
---
SW4(config)#vtp mode client
Setting device to VTP CLIENT mode.
```

Slika 102. Prikaz konfiguriranja uloge preklopnika u kontekstu VTP-a

Nakon što se na preklopniku, VTP poslužitelju, konfigurira naziv domene, svi ostali preklopnici će se automatski pridružiti toj domeni.

```
SW1(config)#vtp domain BOLNICA
Changing VTP domain name from NULL to BOLNICA
SW1(config)#
```

Slika 103. Konfiguracija naziva domene na VTP poslužitelju

Nakon što su svi preklopnici postali članovi nove VTP domene, mogu se konfigurirati različiti VLAN-ovi, i ta informacija bi se trebala propagirati cijelom mrežom. Svaki put kada se na VTP poslužitelju kreira novi VLAN, promijeni mu se naziv ili obriše polje verzije konfiguracije (engl. VLAN *Configuration Revision*), vrijednost će se povećati za jedan. Dakle, polje oznake verzije konfiguracije (engl. *Configuration Revision*) može biti 20 ako je jedan VLAN kreiran, zatim brisan, pa opet kreiran više puta, iako će na kraju biti samo jedan VLAN. Preklopnici informacije koje imaju najveću vrijednost broja oznake verzije konfiguracije, smatraju aktualnima i uvijek će ih koristiti kako bi promijenili svoju konfiguraciju. Zbog takvog načina funkcioniranja postoji opasnost da preklopnik koji ima veći broj oznake verzije konfiguracije, „pobriše“ sve VLAN-ove u mreži, a to uvijek rezultira nedostupnim sustavom IKT-a. Preklopnik VTP *Transparent*, čak i kada je dio VTP domene, a nema iste VLAN-ove kao i ostatak mreže, može uzrokovati probleme u komunikaciji jer da bi preklopnik mogao proslijediti okvire nekom

drugom preklopniku, mora imati konfiguriran isti VLAN kao i ostali preklopnici u mreži. Zato se obično preklopnici VTP *Transparent* nalaze na rubovima mreže kako ne bi „bili na putu“ prometu za VLAN-ove koje nemaju konfigurirane na sebi. Obično su preklopnici u DMZ zoni ili na „farmi“ mrežnog poslužitelja konfigurirani kao *Transparent*.

```
SW4>show vtp status
VTP Version capable             : 1 to 2
VTP version running             : 2
VTP Domain Name                 : BOLNICA
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 0000.0C22.2910
Configuration last modified by 0.0.0.0 at 3-1-93 00:05:52

Feature VLAN :
--------------
VTP Operating Mode              : Client
Maximum VLANs supported locally : 255
Number of existing VLANs        : 7
Configuration Revision          : 11
MD5 digest                      : 0x82 0x04 0xF1 0x0E 0xD0 0x94 0xB8 0xC3
                                  0xED 0xDB 0xA6 0xE9 0x5C 0x9F 0x1E 0xB6

SW4>
```

Slika 104. Primjer VTP stanja na preklopniku u klijentskom modu

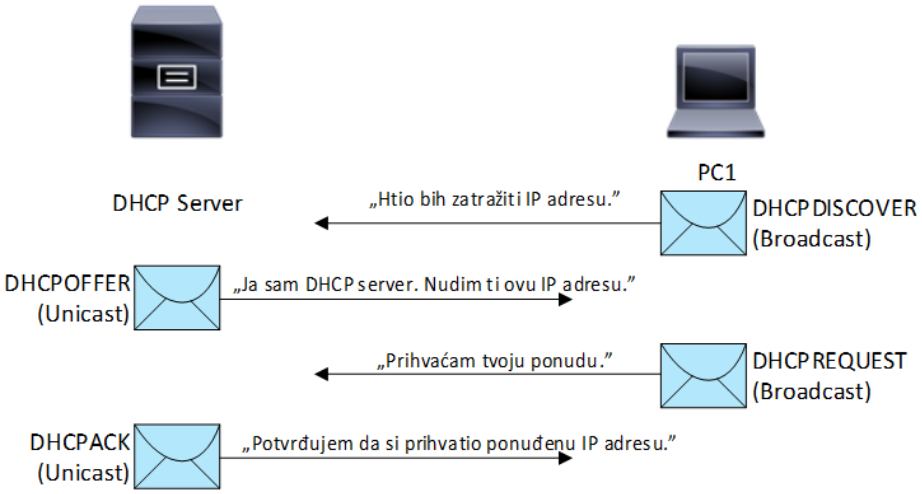
2.6. DHCP — Dynamic Host Configuration Protocol

Svaki uređaj koji se povezuje na računalnu mrežu treba imati jedinstvenu IP adresu kako bi mogao komunicirati s ostalim uređajima u mreži. IP adrese mogu se konfigurirati ručno na svakom računalu, a još se nazivaju statičke IP adrese. Problem prilikom ručne konfiguracije nastaje u mrežama u kojima postoji mnogo računala ili u situacijama kada ima mnogo računala koja nisu stalno spojena na istu mrežu, već se sele iz odjela u odjel, što je posebno izraženo u bežičnim mrežama. Ručni način konfiguracije zahtjeva puno administratorskog vremena i nije učinkovit za mreže s više od nekoliko računala. Naravno, postoje uređaji u mreži kojima se i dalje konfiguriraju statičke IP adrese, i to su uglavnom oni koji vrlo vjerojatno neće nikada mijenjati IP adresu, poput poslužitelja, printera, mrežnog pristupnika *(engl. Gateway)* i sličnih uređaja čije usluge koriste sva računala u mreži. Kako bi se olakšala konfiguracija IP adresa na računalima u mreži, koristi se DHCP (engl. *Dynamic Host Configuration Protocol*). DHCP servis može se konfigurirati na mrežnoj opremi ili na poslužitelju, ovisno o veličini i složenosti mreže. Osim IP adresa, DHCP-om može se računalima dinamički konfigurirati mrežnu (engl. *Subnet*) masku, mrežni pristupnik, DNS, ali i

druge elemente koji su potrebni računalima kako bi uspješno sudjelovala u mrežnoj komunikaciji, poput imena domene, TFTP poslužitelja i slično. Nužna je konfiguracija za većinu računalnih mreža IP adresa, mrežna maska, mrežni pristupnik i IP adresa DNS poslužitelja. U velikim mrežama koje su distribuirane na većem području poput studentskih kampusa, telekomunikacijskih operatora, banaka, bolnica i slično, DHCP poslužitelj se nalazi u zaštićenom dijelu sustava IKT-a, poput podatkovnog centra. DHCP je, zapravo, aplikacija ili usluga koja postoji u sustavu IKT-a, ali obično se kaže DHCP poslužitelj, jer se misli na uređaj na kojem je taj servis uključen i konfiguriran, pa tako i usmjernik ili preklopnik mogu djelovati kao poslužitelj (engl. *Server*). Taj odnos proizlazi iz komunikacije u sustavima IKT-a, a to je primarno arhitektura klijent—poslužitelj, gdje su neki uređaji poslužitelji za neke druge uređaje, bez obzira o kakvoj usluzi ili aplikaciji se radilo. Primjeri su takve komunikacije DHCP klijent koji traži IP postavke od DHCP poslužitelja, računalo koje od NTP poslužitelja traži točno vrijeme, računalo koje od mrežnog poslužitelja traži mrežnu stranicu ili računalo koje se povezuje na poslužitelj videoigara itd. Taj centralizirani DHCP poslužitelj zadužen je za konfiguraciju svih uređaja koji trebaju IP postavke. Većina mrežnih uređaja, poslužitelja i drugih uređaja koji čine infrastrukturu IKT-a, ne koriste DHCP poslužitelj za konfiguraciju IP adresa, već ih se konfigurira ručno, i to na specifičan način koji odgovara ulozi takvih uređaja u povezivanju sustava IKT-a. Za razliku od tih infrastrukturnih uređaja, krajnji uređaji poput računala, IP kamera, IP telefona i slično svoje će IP postavke dobiti od DHCP poslužitelja. Iako se DHCP poslužitelj obično nalazi u zaštićenom dijelu IKT sustava, ponekad se lokalni preklopnik ili lokalni usmjernik može konfigurirati kao DHCP poslužitelj, pa će to biti prikazano u nastavku.

2.6.1. Funkcioniranje DHCP-a

DHCP radi na principu klijent—poslužitelj. DHCP poslužitelj klijentu dodijeli IP adresu i ostale podatke na određeni rok, koji je predefiniran na poslužitelju, i može ga se mijenjati prema potrebi. Klijent mora periodički kontaktirati DHCP poslužitelj kako bi produžio najam IP adrese. Taj je mehanizam potreban kako se ne bi dogodilo da IP adrese dodijeljene računalima koja su isključena ili ih više uopće nema u mreži, ostanu blokirane i neiskoristive. Kada najam neke IP adrese istekne, DHCP poslužitelj ponovno koristi tu IP adresu za nova računala u mreži.

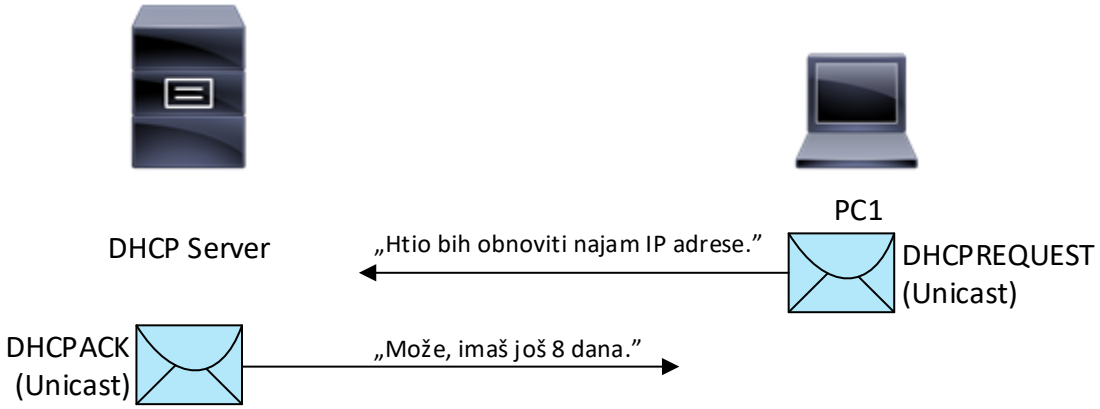


Slika 105. Princip rada DHCP protokola u LAN mreži

Funkcioniranje DHCP-a prikazano je na slici 105. Kada se klijentsko računalo uključi i podigne operativni sustav ili kada se želi spojiti na mrežu, započinje proces od četiri koraka kako bi dobio IP adresu u najam. Prva je poruka koju šalje: „DHCPDISCOVER" da bi pronašao DHCP poslužitelje na IPv4 mreži. S obzirom na to da klijentsko računalo, nakon što se uključilo, nema podatke o IP adresama, „DHCPDISCOVER" poruku šalje na L2 FFFF.FFFF.FFFF i L3 sveodredišnu (engl. *Broadcast*) adresu 255.255.255.255 kako bi pronašao DHCP poslužitelj. Kada DHCP poslužitelj primi *DHCPDISCOVER* poruku, rezervira jednu od dostupnih IPv4 adresa za najam računalu. Istovremeno poslužitelj stvara ARP zapis koji se sastoji od MAC adrese s koje je „DHCPDISCOVER" poruka došla i IP adrese koju će poslužitelj ponuditi računalu. Zatim DHCP poslužitelj šalje obvezujuću „DHCPOFFER" poruku prema računalu koje je tražilo IP adresu. „DHCPOFFER" poruka šalje se kao jednoodredišni promet (engl. *Unicast*) s izvorišnom MAC adresom poslužitelja i odredišnom MAC adresom računala. Kada računalo primi „DHCPOFFER" poruku, šalje „DHCPREQUEST" poslužitelju koji je ponudio IP adresu, kao obvezujući pristanak za korištenje ponuđene IP adrese i istovremeno kao implicitno odbijanje IP postavki svim ostalim DHCP poslužiteljima u L2 mreži koji bi mogli sudjelovati u DHCP procesu, zato se ta poruka šalje kao sveodredišni promet (engl. *Broadcast*). Nakon što poslužitelj primi „DHCPREQUEST" poruku, provjerava je li ponuđena IP adresa zaista slobodna koristeći ICMP *ping*. Ako je IP adresa slobodna, poslužitelj stvara novi ARP zapis za računalo koje je poslalo „DHCPREQUEST" poruku i šalje „DHCPACK" poruku računalu kako bi mu potvrdio da može koristiti ponuđenu IP adresu. Računalo nakon

što primi „DHCPACK“ poruku, također provjerava je li ponuđena IP adresa slobodna, koristeći ARP. Ako računalo ne dobije odgovor na ARP zahtjev za tu IP adresu, smatra je slobodnom i počinje je koristiti kao svoju.

U slučaju da je istekao najam i treba ga produžiti, računalo šalje „DHCPREQUEST“ poruku direktno poslužitelju koji mu je dao IP adresu na raspolaganje. Ako računalo ne dobije odgovor unutar predefiniranog vremena, šalje sveodredišni „DHCPREQUEST“ dok mu ne odgovori neki drugi DHCP poslužitelj. Poslužitelj će poslati „DHCPACK“ poruku kako bi računalu produžio najam. „DHCPREQUEST“ i „DHCPACK“ poruke gotovo su identične, jedina je razlika vrijednost u polju „type“.



Slika 106. Obnova IP postavki nakon predefiniranog vremena

Format DHCP poruke

Isti se format DHCP poruke koristi uvijek u komunikaciji pomoću DHCP protokola. Sve su poruke enkapsulirane UDP transportnim protokolom. DHCP poruke koje šalju računala za TCP/UDP ulaze koriste izvorišni UDP ulaz 68, a odredišni UDP ulaz 67. Poruke koje šalje poslužitelj, imaju izvorišni UDP ulaz 67, a odredišni UDP ulaz 68. Koriste li se liste za kontrolu pristupa u mreži zajedno s DHCP-om, treba paziti da se ne zabrani taj promet. Na slici 107. prikazan je format DHCP poruke. Objašnjenja polja su ispod slike.

8	16	24	32
OP Code (1)	Hardware Type (1)	Hardware addres Length (1)	Hops (1)
Transaction Identifier			
Seconds -2 bytes		Flags -2 bytes	
Client IP Address (CIADDR) -4 bytes			
Your IP Address (YIADDR) -4 bytes			
Server IP Address (SIADDR) -4 bytes			
Gateway IP Address (GIADDR) -4 bytes			
Client Hardware Address (CHADDR) -16 bytes			
Server Name (SNAME) -64 bytes			
Boot Filename -128 bytes			
DHCP Options -variable			

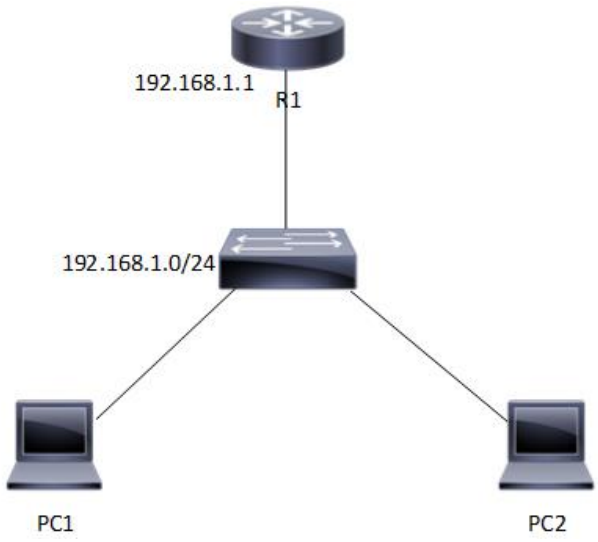
Slika 107. Polja u DHCP porukama

- ***Operation (OP) Code*:** određuje tip poruke (npr. 1 je zahtjev, a 2 je odgovor);
- ***Hardware Type***: pokazuje o kojem se tipu hardvera radi (npr. 1 je *Ethernet*, 15 je *Frame Relay* i slično. Isti se kodovi koriste i u ARP porukama);
- ***Hardware Address Length***: veličina adrese;
- ***Hops***: kontrola prosljeđivanja poruka. Klijent postavlja tu vrijednost na 0;
- ***Transaction Identifier***: koriste klijenti kako bi znali povezati odgovore poslužitelja sa svojim zahtjevima;
- ***Seconds***: vrijeme u sekundama koje je prošlo od kad je klijent počeo tražiti ili obnavljati IP adresu. Koriste poslužitelji kako bi zahtjeve složili po prioritetu;
- ***Flags* (zastavice):** koristi se samo jedan od 16 bitova, i to sveodredišna (engl. *Broadcast*) zastavica. Klijent tu zastavicu postavlja kada ne zna svoju IP adresu, i to je znak poslužitelju ili DHCP *relay* agentu da pošalju sveodredišni odgovor (engl. *Broadcast*);
- ***Client IP Address***: koristi je klijent samo za vrijeme obnove najma IP adrese kada ima valjanu IP adresu i ta je IP adresa ona koju je računalo koristilo do trenutka obnove njezina najma. Inače, to polje ima vrijednost 0;
- ***Your IP Address*:** koristi poslužitelj kada dodjeljuje IP adresu klijentu;
- ***Server IP Address***: koristi poslužitelj kada želi uputiti klijenta na onaj koji će koristiti u nastavku procesa. To može biti isti ili neki drugi poslužitelj. IP adresa poslužitelja koji šalje poruku nalazi se u opcionalnom polju „*Server Identifier*“;

- ***Gateway IP Address***: to se polje koristi kada DHCP poslužitelj i klijent nisu na istom L2 segmentu. U tom se slučaju koristi posrednički uređaj za prosljeđivanje DHCP upita (engl. *DHCP Relay Agent*);
- ***Client Hardware Address*****:** fizička adresa uređaja;
- ***Server Name***: koriste DHCP poslužitelji kada komuniciraju s klijentima. Može biti domensko ime poslužitelja;
- ***Boot Filename***: mogu koristiti klijenti ako im treba nekakva datoteka za podizanje sustava. Koristi se za IP telefone;
- ***DHCP Options***: to polje mogu koristiti i klijenti i poslužitelji za različite opcionalne parametre.

KONFIGURACIJA *CISCO* USMJERNIKA KAO DHCP POSLUŽITELJA

Usmjernik *Cisco* može se konfigurirati da radi kao DHCP poslužitelj, i to se često koristi u manjim mrežama. Primjer konfiguracije na slici je 108.



Slika 108. Jednostavna topologija za implementaciju DHCP servisa na usmjerniku

Prvi korak u konfiguraciji DHCP servisa na usmjerniku odnosi se na IP adrese koje se ne želi dijeliti klijentima. Takve su IP adrese obično one koje će se statički konfigurirati na uređaje u mreži koje će svi klijenti koristiti, poput printera, mrežnog pristupnika (engl. *Gateway*), DNS poslužitelja, FTP poslužitelja i slično. Zatim slijedi konfiguracija DHCP raspona (engl. *Pool*), odnosno raspon IP adresa koje će se DHCP-om dijeliti. DHCP raspon treba imenovati za lakše snalaženje u konfiguraciji, npr. „LAN-POOL-1“. Unutar njega minimalno što treba konfigurirati podmreža (engl. *Subnet*) je iz kojega će

IP adrese biti dijeljene klijentima, mrežni pristupnik i adresa DNS poslužitelja. Osim tih osnovnih elemenata može se konfigurirati i drugo, poput vremena najma, imena domene i slično.

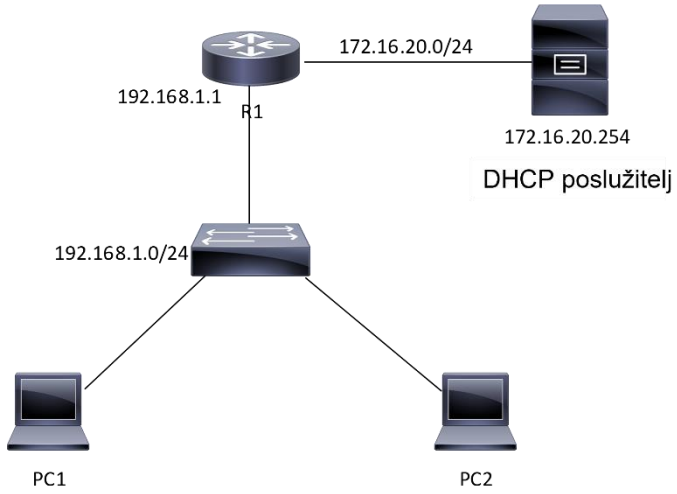
```
R1(config)#ip dhcp excluded-address 192.168.10.1 192.168.10.10
R1(config)#ip dhcp excluded-address 192.168.10.100 192.168.10.254
R1(config)#ip dhcp pool LAN-POOL-1
R1(dhcp-config)#network 192.168.10.0 255.255.255.0
R1(dhcp-config)#default-router 192.168.10.1
R1(dhcp-config)#dns-server 8.8.8.8
R1(dhcp-config)#option 150 ip 192.168.10.1
R1(dhcp-config)#end
```

Slika 109. Konfiguracijske naredbe za implementaciju DHCP servisa na usmjerniku

Iz naredbi na slici 109. vidljivo je da će taj usmjernik (R1) dijeliti IP adrese u mreži 192.168.10.0/24, ali neće dijeliti IP adrese od 192.168.10.1 do 192.168.10.10 i od 192.168.10.100 do 192.168.10.254 (*excluded-address*), što znači da će raspon iz kojega će usmjernik dijeliti IP adrese biti od 192.168.10.11 do 192.168.10.99. Također je vidljivo da će mrežni pristupnik za klijente biti IP adresa 192.168.10.1, a DNS će biti IP adresa 8.8.8.8. U konfiguraciji se još vidi da je konfigurirana opcija 150, i tu opciju koriste IP telefoni kako bi znali gdje se nalazi TFTP poslužitelj s kojeg će dobiti konfiguraciju za rad, a u ovom će to slučaju biti ista IP adresa kao i za mrežni pristupnik, što je zapravo usmjernik na kojem je izvršena konfiguracija DHCP-a.

Ako bi DHCP poslužitelj bio izvan L2 segmenta na kojem su klijenti, tada treba konfigurirati usmjernik između klijenata i DHCP poslužitelja tako da prosljeđuje DHCP zahtjeve prema pravom DHCP poslužitelju. Razlog tome je u činjenici što klijenti DHCP zahtjeve šalju sveodredišno, a već je rečeno da sveodredišni promet ne ide dalje od lokalnog L2 segmenta. Usmjernik u tom slučaju djeluje kao DHCP *relay*, i sve zahtjeve koje pošalju klijenti, prosljeđuje kao jednoodredišne prema konfiguriranom DHCP poslužitelju. Da bi to radilo, nužno je da DHCP poslužitelj i DHCP *relay* agent mogu doći jedan do drugoga, odnosno da DHCP poslužitelj zna kako doći do mreže za koju dijeli IP postavke. Za to je potrebno imati implementirane protokole usmjeravanja u računalnoj mreži. Na topologiji prikazanoj na slici 110. nije potreban nikakav protokol usmjeravanja, jer je DHCP poslužitelj spojen direktno u DHCP *relay* agenta koji je ujedno i mrežni pristupnik za DHCP poslužitelj. Međutim, kada bi između DHCP *relay*

agenta i DHCP poslužitelja bilo dva ili više usmjernika, tada bi trebalo na neki način uspostaviti komunikaciju između tih uređaja, bilo statičkim putanjama, bilo dinamičkim protokolima usmjeravanja.



Slika 110. Primjer konfiguracije usmjernika kao DHCP relay agenta

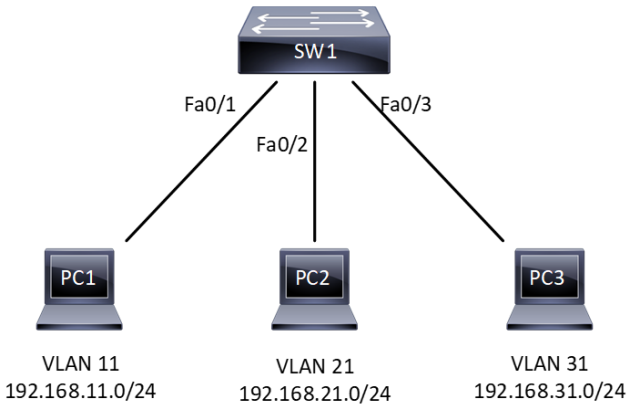
```
R1(config)#interface fa 0/0
R1(config-if)#ip add 192.168.1.1 255.255.255.0
R1(config-if)#ip helper-address 172.16.20.254
R1(config-if)#exit
```

Slika 111. Primjer konfiguracije usmjernika kao DHCP relay agenta

U prikazanom slučaju jedino što je potrebno konfigurirati jest „ip helper-address“, što je zapravo IP adresa pravog DHCP poslužitelja, i to na sučelje na usmjerniku na koje dolaze DHCP zahtjevi klijenata. U slučaju kad bi bilo više podsučelja za *InterVLAN* usmjeravanje, tada bi to trebalo napraviti na svakom od podsučelja za čiju se podmrežu želi koristiti DHCP. Ako je riječ o implementaciji „usmjernika na štapu“, gdje je jedan usmjernik mrežni pristupnik (engl. *Gateway*) za više VLAN-ova, konfiguracija DHCP-a se radi na potpuno isti način, ali za svaku mrežu podsučelja posebno. Ima li usmjernik četiri podsučelja, imat će i četiri DHCP raspona za konfiguraciju računala u svakom od VLAN-ova za koji usmjernik djeluje kao mrežni pristupnik (engl. *Gateway*). Ako usmjernik nije DHCP poslužitelj, već mu DHCP upite prosljeđuje, usmjernik će na svakom podsučelju imati naredbu „ip-helper“ s IP adresom uređaja – DHCP poslužitelja za pojedini VLAN. Obično je jedan DHCP poslužitelj za sve podmreže u sustavu.

KONFIGURACIJA PREKLOPNIKA *CISCO* KAO DHCP POSLUŽITELJA

Osim na usmjernicima, DHCP se može konfigurirati i na preklopnicima za svaki od VLAN-ova na sličan način. Da bi se to moglo, kao i slične naprednije funkcionalnosti, na preklopniku treba koristiti tzv. L3 preklopnik, tj. višeslojni preklopnik (engl. *Multilayer Switch*). Razlika između L2 i L3 preklopnika primarno je u funkcionalnostima koje podržavaju. Dok L2 preklopnik prosljeđuje promet samo na temelju MAC adresa i ne može usmjeravati promet na temelju IP adresa, L3 preklopnik uz to, podržava razne funkcionalnosti temeljene na IP adresama, poput usmjeravanja, DHCP-a, NAT-a i slično. Konfigurira li se DHCP usluga na preklopniku u mreži koja se sastoji od više podmreža, odnosno VLAN-ova, potrebno je izraditi DHCP konfiguraciju za svaku od njih, tako da preklopnik može adekvatno konfigurirati računala. Ako preklopnik ili usmjernik nema sučelje s IP adresom u rasponu u kojem bi DHCP servis trebao „dijeliti“ IP postavke, onda to neće moći ni napraviti.



Slika 112. Shema lokalne računalne mreže s podmrežama i VLAN-ovima

Na slici 112. vidljivo je kako je povezana mala lokalna računalna mreža za koju treba konfigurirati DHCP servis s postavkama za svaki VLAN. Iako su na slici prikazana samo tri računala, ista konfiguracija bila bi da je u svakom VLAN-u spojeno po 200 računala. Jasno je da u jedan preklopnik koji ima 24 sučelja nije moguće povezati tri puta po 200 računala direktno, ali ako bi se na taj jedan preklopnik povezalo umjesto računala, još preklopnika, pa na njih još preklopnika, mrežu se može skalirati koliko je potrebno. Nazivi VLAN-ova odgovaraju stvarnom svijetu i oznakama učionica u nekoj školi, a također oznake učionica i VLAN-ova odgovaraju podmrežama. Po tom principu logičkoga i smislenog povezivanja gradi se računalna mreža, ali i sustav IKT-a u cjelini.

Prilikom odražavanja računalne mreže bit će jasno da za problem u učionici 11 treba odmah gledati VLAN 11 i podmrežu 192.168.11.0/24.

```
SW1>enable
SW1#configure terminal
SW1(config)#vlan 11
SW1(config-vlan)#name Ucionica11
SW1(config-vlan)#exit
SW1(config)#vlan 21
SW1(config-vlan)#name Ucionica21
SW1(config-vlan)#exit
SW1(config)#vlan 31
SW1(config-vlan)#name Ucionica31
SW1(config-vlan)#exit
SW1(config)#interface fa 0/1
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 11
SW1(config-if)#exit
SW1(config)#interface fa 0/2
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 21
SW1(config-if)#exit
SW1(config)#interface fa 0/3
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 31
SW1(config-if)#
```

Slika 113. Konfiguracija VLAN-ova s njihovim nazivima i dodijeljenim sučeljem

Kada su konfigurirani potrebni VLAN-ovi i određeni im nazivi, treba izraditi L3 virtualna sučelja za svaki VLAN, jer će ona biti mrežni pristupnik (engl. *Gateway*) za računala u pojedinom VLAN-u.

```
SW1(config)#interface vlan 11
SW1(config-if)#ip address 192.168.11.254 255.255.255.0
SW1(config-if)#exit
SW1(config)#interface vlan 21
SW1(config-if)#ip address 192.168.21.254 255.255.255.0
SW1(config-if)#exit
SW1(config)#interface vlan 31
SW1(config-if)#ip address 192.168.31.254 255.255.255.0
SW1(config-if)#exit
SW1(config)#exit
SW1#write memory
SW1#
```

Slika 114. Konfiguracija virtualnih L3 sučelja za pojedini VLAN s pohranom konfiguracije — write memory

Kad bi se sada ručno konfigurirala sva računala s IP adresama i svako računalo koristilo svoj mrežni pristupnik (engl. *Gateway*) u vidu IP adrese L3 virtualnog VLAN sučelja, komunikacija bi bila moguća. Međutim, svrha DHCP protokola nije ručna konfiguracija IP postavki, osobito u velikim mrežama s tisućama računala. Na slici 115. vidljiv je princip smislenog povezivanja u gradnji računalne mreže, tako da se zna na koji element stvarnog svijeta se odnosi, u ovom slučaju Učionicu 11 i Učionicu 21.

```
SW1(config)#ip dhcp pool DHCP-POOL-Ucionica11
SW1(dhcp-config)#network 192.168.11.0 255.255.255.0
SW1(dhcp-config)#default-router 192.168.11.254
SW1(dhcp-config)#dns 8.8.8.8
SW1(dhcp-config)#exit
SW1(config)#ip dhcp pool DHCP-POOL-Ucionica21
SW1(dhcp-config)#network 192.168.21.0 255.255.255.0
SW1(dhcp-config)#default-router 192.168.21.254
SW1(dhcp-config)#dns 8.8.8.8
SW1(dhcp-config)#exit
SW1(config)#
SW1(config)#ip dhcp pool DHCP-POOL-Ucionica31
SW1(dhcp-config)#network 192.168.31.0 255.255.255.0
SW1(dhcp-config)#default-router 192.168.31.254
SW1(dhcp-config)#dns 8.8.8.8
SW1(dhcp-config)#exit
SW1(config)#
```

Slika 115. Konfiguracija DHCP raspona (engl. Pool) za dva VLAN-a na preklopniku

Treba imati na umu da računala koja se želi konfigurirati pomoću DHCP-a moraju biti dio L2 mreže u kojoj je DHCP servis uključen, a to znači da sučelja na preklopniku u koja su spojena računala moraju biti dio određenog VLAN-a za koji postoji DHCP konfiguracija. Tek nakon što su konfigurirani svi potrebni elementi računala, u VLAN-ovima će dobiti IP postavke od preklopnika.

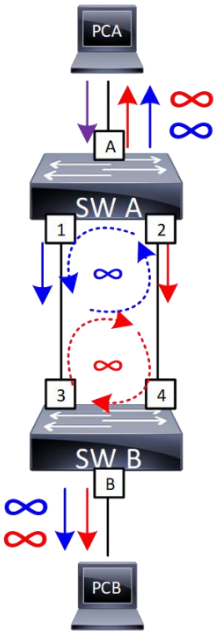
2.7. Spanning Tree Protocol — STP

STP (engl. *Spanning Tree Protocol*) je protokol koji služi za sprječavanje nastanka petlji u L2 mrežama koje imaju redundantne veze između preklopnika. To se odnosi na situacije kada je jedan preklopnik povezan na više preklopnika i sve veze čine „zatvoreni krug“, ali i kada su dva preklopnika povezana međusobno s više veza, kao što je vidljivo na slikama 116. i 117. Takve su mreže česte, osobito kada su hijerarhijske te imaju višestruke veze između preklopnika na različitim slojevima hijerarhije u svrhu veće propusnosti i otpornosti mreže u cjelini na otkaze pojedinih veza između preklopnika. Iako je korisno imati redundantne veze koje će u slučaju otkaza primarne veze preuzeti prosljeđivanje mrežnog prometa, to može uzrokovati preklopničke petlje (engl. *Switching Loops*), koje vrlo brzo mogu učiniti cijelu mrežu nedostupnom.

Osnovni principi rada preklopnika:

- Preklopnik održava tablice sa zapisima MAC adresa (engl. *MAC Address Table*) i sučelja kojima se može doći do pojedine MAC adrese;
- Okvire kojima je sveodredišna (engl. *Broadcast*) MAC adresa (FF:FF:FF:FF:FF:FF) šalje na sva sučelja, osim na ono na koje je okvir primljen. Taj se proces još naziva i *Flooding*;
- Ako preklopnik u svojoj tablici MAC adresa nema zapis za neku MAC adresu (jer u mreži ne postoji uređaj s tom MAC adresom), takav okvir se također tretira kao sveodredišni (engl. *Broadcast*) i šalje na sva sučelja, osim na sučelje na koje je okvir primljen.

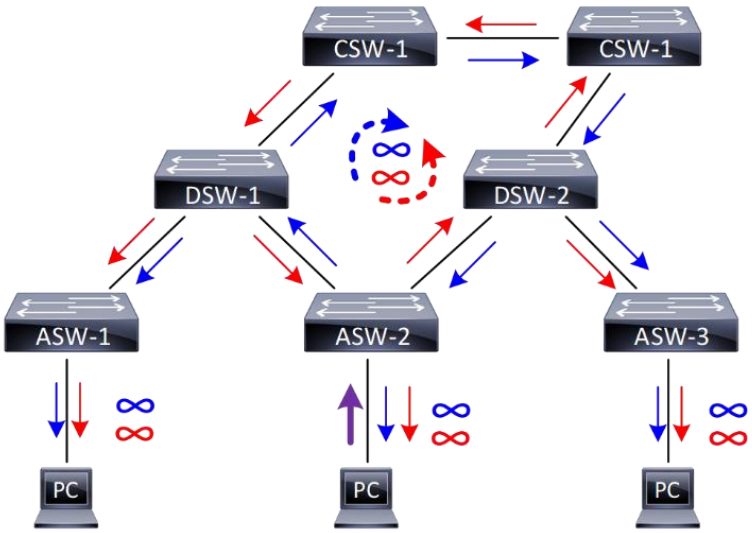
Funkcionalnost STP-a jasna je kada se razumije kako redundantno povezani preklopnici prosljeđuju okvire između računala bez STP-a, tj. kada nema blokiranih sučelja na preklopnicima. Za ilustraciju takve situacije može se promatrati što se događa sa sveodredišnim (engl. *Broadcast*) mrežnim prometom (npr. ARP zahtjevom) koji neko računalo pošalje u mrežu redundantno povezanih preklopnika. Uz pomoć slike 116. treba pretpostaviti da računalo PCA šalje okvir na sveodredišnu (engl. *Broadcast*) L2 MAC adresu (FF:FF:FF:FF:FF:FF), kao što je slučaj kada računala šalju poruku ARP zahtjeva. Preklopnik SWA će taj okvir primiti i upisati izvorišnu MAC adresu računala PCA u svoju tablicu MAC adresa.



Slika 116. Ilustracija beskonačne petlje između dva preklopnika zbog samo jednog sveodredišnog (engl. Broadcast) okvira

Budući da je sveodredišna (engl. *Broadcast*) MAC adresa, preklopnik će taj okvir proslijediti na sva sučelja osim na ono na koje je primio taj okvir, u ovom slučaju prosljeđuje okvire na sučelja 1 i 2 prema preklopniku SWB. Preklopnik SWB će primiti dva okvira koja imaju sveodredišnu (engl. *Broadcast*) MAC adresu na svoja sučelja 3 i 4, a izvorišnu MAC adresu računala PCA zapisat će u svoju tablicu MAC adresa kao da je na oba sučelja 3 i 4. Preklopnik SWB će prema računalu PCB poslati dva ista okvira koja dolaze od PCA. Vidljiv je problem dupliciranja okvira zbog višestrukih veza u mreži. Međutim, najveći je problem što će preklopnik SWB okvir koji primi od preklopnika SWA na svoje sučelje 2 proslijediti natrag prema SWA kroz svoje sučelje 4. Isto će tako preklopnik SWB okvir koji je primio od preklopnika SWA na svoje sučelje 4 proslijediti natrag prema SWA kroz svoje sučelje 3. Nakon toga SWA će te okvire primiti i napraviti isto prema SWB, SWB prema SWA i tako nastaje beskonačna petlja. Osim toga, svaki put kada neki od preklopnika primi okvire, poslat će ih i prema računalima, pa će svi uređaji u mreži primati ogromnu količinu sveodredišnoga (engl. *Broadcast*) prometa koji nije za njih. Što je više takvih sveodredišnih (engl. *Broadcast*) okvira, mreža će prije doći u stanje zagušenja i prestat će funkcionirati.

Za razliku od sveodredišnog (engl. *Broadcast*) prometa koji se od prvog okvira beskonačno prosljeđuje između preklopnika, u hipotetskom okruženju gdje bi postojao samo jednoodredišni (engl. *Unicast*) promet koji se šalje prema poznatoj MAC adresi, postojat će samo problem dupliciranja okvira, ali i to bi negativno utjecalo na performanse mreže i svakako trebalo spriječiti. U normalnom funkcioniranju mreže prije jednoodredišnoga uvijek se dogodi sveodredišni promet (npr. ARP zahtjev) te će i jedan i drugi završiti u beskonačnoj petlji, jer će oba preklopnika SWA i SWB imati na svim svojim sučeljima zapamćene MAC adrese oba računala, te će se i jednoodredišni promet „zavrtjeti“ u petlji i beskonačno dupliciratи. Na tom jednostavnom primjeru vidljiv je princip nastanka petlje, a u složenijim mrežama problem je još teži, i mreže su nedostupne dok god postoji petlja, čak i danima.



Slika 117. Prikaz preklopničke petlje na više redundantno povezanih preklopnika: onemogućena komunikacija mrežom

Posljedice postojanja redundantnih veza u mreži bez STP-a:

- **sveodredišna *(engl. Broadcast)* oluja** – ako preklopnik ne zna odredišnu MAC adresu, a to je posebno slučaj kada odredišna MAC adresa ne postoji u mreži, kada ne postoji u ARP tablici ili ako je sveodredišna MAC adresa, takvi okviri se prosljeđuju na sva sučelja i automatski nastaje beskonačna preklopnička petlja koja lako može učiniti mrežu nedostupnom. Svaki novi sveodredišni promet pogoršava stanje u mreži. Ovdje nije pitanje hoće li mreža postati nedostupna, već samo za koliko vremena, jer sav sveodredišni promet koji se generira, gomila se u mreži. Kako o sveodredišnom prometu inače ovisi normalno funkcioniranje mreže, takvog prometa u svakoj mreži ima jako puno. Za razliku od mrežnog sloja OSI modela, gdje u zaglavlju IP paketa postoji TTL polje koje služi za odbacivanje paketa kada im vrijednost TTL polja dođe do nule, na sloju podatkovne veze u *Ethernet* okviru nema takvog polja, i zato svi okviri koji završe u petlji, nikada neće biti odbačeni. Naravno, u praksi je situacija malo drugačija, pa će preklopnici blokirati i odbacivati promet jer jednostavno ne mogu obraditi tolike količine, ali tada je već mreža postala nedostupna, pa više i nije bitno što se neki okviri odbacuju.
- **primanje istog jednoodredišnog (engl. *Unicast*) okvira više puta** – također je posljedica redundancije u mreži. Primanje dupliciranih okvira u normalnoj funkcionalnoj mreži moguće je vidjeti samo u slučaju da prethodno nije bilo sveodredišnog prometa s računala koja komuniciraju, no u praksi toga nema. Stoga,

kada računalo vidi jedan duplicirani okvir, to je obično znak da je nastala preklopnička petlja i da će neprestano primati duplicirane okvire.

- **višestruki zapisi u MAC tablicama preklopnika** – zbog petlji stvorenih redundancijom okviri u preklopnik dolaze na sva sučelja povezana s nekim drugim preklopnicima i zbog toga preklopnicima izgleda da je odredišno računalo dostupno na svim sučeljima, te šalju okvire na sva ta sučelja, što dodatno pogoršava petlju.

Svi problemi s preklopničkim petljama nastaju samo u slučaju da STP mehanizam nije implementiran. STP, dakle, služi za sprječavanje nastanka preklopničkih petlji, što znači da unaprijed računa koje redundantne veze između preklopnika treba blokirati kako se preklopnička petlja ne bi ni pojavila. U praksi je STP mehanizam prisutan i aktivan u svim preklopnicima na kojima se može konfigurirati pa je preporuka takve i koristiti. Također je preporuka izbjeći ovisnost formiranja logičke mrežne topologije o STP-u korištenjem drugih mehanizama, ali nikako isključivati STP na preklopnicima. Postoje drugi „napredni“ mehanizmi koji omogućuju redundantnu i stabilnu mrežnu topologiju (npr. kombinacija agregacije veza i preklopnika u stogu, i slično) bez ovisnosti o STP mehanizmu. Međutim, treba uzeti u obzir da se preklopnici mogu povezati na način koji ti „napredni“ mehanizmi ne očekuju, a nemamo li STP u mreži, opet će nastati preklopnička petlja.

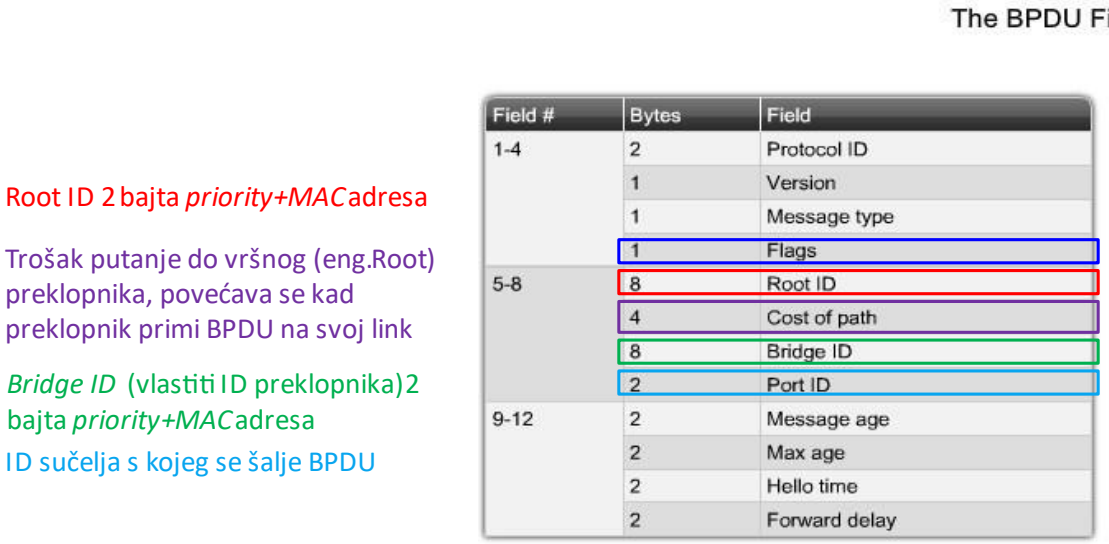
Vremenom je STP mehanizam evoluirao, i danas se u mrežama uglavnom koristi RSTP (engl. *Rapid Spanning Tree Protocol*), kao i verzije koje podržavaju okruženje s više VLAN-ova; MST (engl. *Multiple Spanning Tree*), PVST+ (engl. *Per VLAN Spanning Tree*), odnosno RPVST+ (engl. *Rapid Per VLAN Spanning Tree+*), o čemu će biti više riječi u nastavku.

STA ALGORITAM

Kako se u redundantno povezanoj mreži koju čine preklopnici ne bi dogodile preklopničke petlje, koristi se STP. STP za svoj rad koristi algoritam kojim preklopnici „nauče“ kako izgleda računalna mreža i koje su redundantne veze, a zatim određene veze po određenim pravilima blokiraju kako se ne bi dogodile preklopničke petlje. Dakle, iako je fizička topologija i dalje ista, odnosno preklopnici su i dalje povezani kako su bili prije STP protokola, logička topologija mreže izgleda drugačije zato što se neke veze između preklopnika ne koriste. Pod logičkom topologijom misli se na to kojim vezama između preklopnika će promet zaista biti prosljeđivan. One veze koje su

blokirane, neće biti korištene za prosljeđivanje prometa dok god su funkcionalne aktivne veze u logičkoj topologiji. STP protokol samo najbolje veze ostavlja aktivnima. Pitanje je kako STP „zna“ koje su veze najbolje? Već je rečeno da se termini kao „znati“, „misliti“, „vidjeti“ ili „utvrditi“ uopće ne odnose na računala, ali pojednostavnjuju priču o uređajima u mreži i njihovu radu. Dakle, STP neće ništa „znati“, ali će svi preklopnici koji koriste STP odraditi niz koraka kako bi „utvrdili“ koje veze između preklopnika u mreži privremeno blokirati, tako da logička topologija nije redundantna, već jasno definirana samo jednim aktivnim putem kroz mrežu. Na taj se način sprječava nastanak preklopničkih petlji. Međutim, ako se koja od tih aktivnih veza prekine, aktivira se neka od logički blokiranih fizičkih veza, i tako se mijenja i logička topologija. Cilj je da svi preklopnici u mreži „zaključe“ koji će preklopnik u mreži biti „prihvaćen“ kao referentna točka u odnosu na koju će se izgraditi stablasta struktura aktivnih veza koje čine logičku topologiju. Primarno se koristi kumulativna metrika temeljena na propusnosti sučelja na putu do referentnog preklopnika. Svi će ostali preklopnici „odlučiti“ koja sučelja će koristiti kao najbolja za slanje prometa u računalnu mrežu (vršna sučelja) i u dogovoru sa susjednim preklopnicima koja sučelja će biti neodabrana (engl. *Non-designatied*), a koja odabrana (engl. *Designated*).

Da bi STP mogao utvrditi stanje mreže i odrediti konačnu STP topologiju, nužna je međusobna komunikacija preklopnika i razmjena određenih informacija. Ta se komunikacija odvija pomoću BPDU (engl. *Bridge Protocol Datagram Unit*) okvira.



Slika 118. Prikaz ključnih vrijednosti u BPDU okviru

Da bi preklopnici koristeći STP u konačnici utvrdili konvergiranu i stabilnu STP topologiju, svi međusobno povezani preklopnici u mreži prolaze trima koracima konvergencije uspoređujući određene elemente i donose odluku o vršnom preklopniku i stanjima sučelja.

Predefinirana je konfiguracija na preklopnicima *Cisco* postojanje STP-a samo za VLAN 1 i koristi se način PVST+. Početni prioritet je 32768 i mijenja se u inkrementima od 4096, a prioritet na sučelju je 128 i mijenja se u inkrementima od 16. Trošak (engl. *Cost*) je podešen na 4 za 1000 Mbps veze, a 19 za 100 Mbps veze. Vrijeme između *Hello* poruka je 2 sekunde, *Forward-Delay* iznosi 15 sekundi, *Max. Aging* vrijeme je 20 sekundi. Sve ove vrijednosti mogu se mijenjati, ali nikako se ne smije dogoditi da različiti preklopnici u mreži imaju različite postavke jer će to uzrokovati nestabilnost u mreži.

Ključni koraci konvergencije topologije koje svaki preklopnik prolazi:

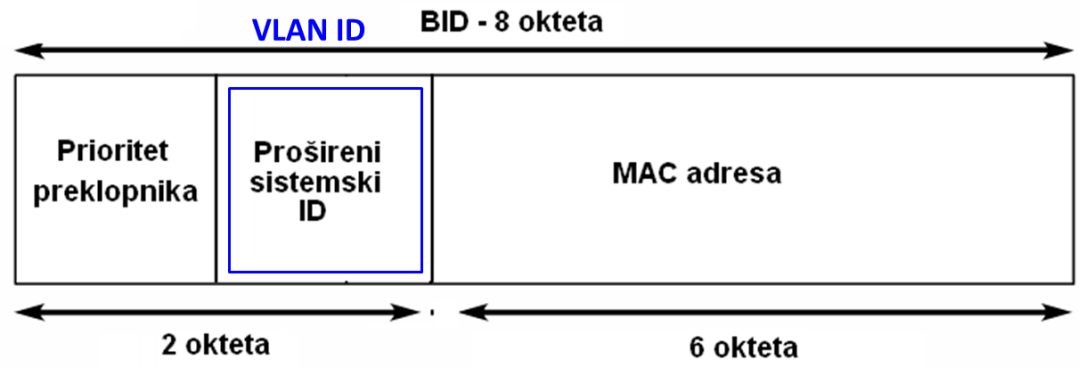
1. Odabir vršnog preklopnika (engl. *Root Switch*) u mreži
2. Odabir vršnih sučelja (engl. *Root Port*)
3. Odabir odabranih (engl. *Designated*) i neodabranih (engl. *Non-designated*) sučelja.

Proces odabira vršnog preklopnika u mreži:

Svaki preklopnik u proces definiranja logičke topologija kreće iz pozicije vršnoga preklopnika u mreži. Ključne informacije u potvrdi je li on vršni preklopnik u mreži ili ne, donosi *Root ID* i *Bridge ID*. Sva se komunikacija između preklopnika u mreži u kontekstu STP mehanizma odvija koristeći BPDU okvire, i u njima se nalazi informacija o tome što preklopnik koji šalje BPDU paket smatra da je *Root ID* i koji je njegov *Bridge ID*. Te okvire preklopnici šalju na sveodredišnu MAC adresu 01:80:C2:00:00:00. Kada preklopnik počinje proces odabira STP vršnog preklopnika, vrijednosti su u ta dva polja BPDU poruke iste. Svaki puta kada neki preklopnik dobije BPDU paket, on usporedi svoj *Bridge ID* s informacijom u polju *Root ID*. Zaključi li da je *Root ID* niži (bolji) od njegovog, tada prihvaća da on nije više vršni preklopnik i dalje u mrežu na odabrana sučelja (engl. *Designated Ports*) prosljeđuje BPDU okvire s informacijom u *Root ID* polju preklopnika za koji zna da ima manji *Bridge ID* – *BID*. Taj BPDU koji dolazi od vršnog preklopnika, svi ostali preklopnici modificiraju dodajući neke informacije poput

svog BID-a starosti poruke (eng *Message Age*) i vrijednosti troška (engl. *Cost*) koju ima do vršnog preklopnika i slično. U konačnici, svi će preklopnici uspoređivanjem *Root ID* i BID vrijednosti doći do zaključka koji je najniži BID u mreži, i koliki im je trošak do vršnog preklopnika. Dakle, to je iterativni proces uspoređivanja i ažuriranja *Root ID* polja, dok u konačnici svi imaju istu vrijednost u tom polju. Taj tren jasno je koji je preklopnik vršni u mreži i konvergencija STP topologije može se nastaviti određivanjem vršnog sučelja, odabranog i neodabranog sučelja na svakom preklopniku.

Bridge ID polje veličine 64 bita sastoji se od vrijednosti prioriteta preklopnika u rasponu od 0 do 61440 predefinirane vrijednosti 32768 i konfigurira se u inkrementima od 4096, zatim od proširenog sistemskog ID-a, što je zapravo VLAN ID, i MAC adrese. U originalnoj verziji BID polja nije bilo VLAN ID-a, već je polje Prioritet imalo 16 bita na raspolaganju, što je omogućavalo vrijednosti od 0 do 65535. To je bilo i više nego dovoljno za razlikovanje preklopnika po prioritetu. Kada su se pojavili VLAN-ovi kao koncept implementiran u mrežama, od tih 16 bitova, uzeto je 12 za VLAN ID, a za svaki od četiri preostala bita određena je vrijednost, tako da ovisno o kombinaciji bitova, najveća vrijednost može biti 61440. Ono što se dogodi ako ni jedan preklopnik nije konfiguriran, da svi preklopnici imaju isti predefinirani dio Prioritet + VLAN ID (npr. za VLAN 10 to će biti 32768 + 10 = 32778), i zato će u mreži za vršni preklopnik biti odabran onaj s najmanjom MAC adresom, jer je MAC adresa jedino što *Bridge ID* preklopnika čini različitim.



Slika 119. Sastav BID vrijednosti na preklopniku

Međutim, u praksi ne bi trebalo dopustiti odabir vršnog preklopnika na temelju MAC adrese, jer su preklopnici istog proizvođača s manjim MAC adresama obično proizvedeni prije, pa mogu biti stariji modeli sa slabijim performansama. Ako stari preklopnik postane vršni preklopnik, sav promet će ići kroz njega, i on može postati

usko grlo koje zagušuje mrežnu komunikaciju. Isto tako, ako vršni postane preklopnik koji se nalazi na pristupnom sloju mrežne hijerarhije koji je obično nižih performansi, to će usmjeriti velik dio prometa kroz taj preklopnik, što opet može uzrokovati probleme s karakteristikama mreže. Zato je važno da se za vršne preklopnike odaberu oni s najboljim karakteristikama i centralnim položajem u mreži u smislu količine ukupnoga mrežnog prometa koji kroz takve preklopnike prolazi, a to su obično preklopnici sloja jezgre.

Nakon što preklopnici odaberu vršni preklopnik u mreži, započinju proces utvrđivanja stanja pojedinih aktivnih sučelja kako bi odredili koje će sučelje na preklopniku biti vršno, koje će biti odabrano, a koje će biti neodabrano sučelje. Da bi neki preklopnik utvrdio vršno sučelje, on uspoređuje vrijednosti u BPDU paketima koje dobije na svako od sučelja, kandidata za vršno. Čim preklopnik zaključi koje je sučelje vršno, na tom koraku staje i dalje ne uspoređuje.

Koraci određivanja vršnog sučelja:

1. vršno sučelje imaju svi preklopnici osim vršnog preklopnika, a određuje se usporedbom kumulativnog troška koji preklopnik ima do vršnoga preklopnika
2. ima li više mogućih puteva preko različitih preklopnika do vršnog preklopnika, a svi imaju isti kumulativni trošak, onda će vršno sučelje biti ono koje ima bolju vrijednost BID polja (manje je bolje)
3. ako je preklopnik s više sučelja povezan na neki drugi preklopnik, koji je na putu prema vršnom preklopniku, tada će vršno sučelje biti ono koje je povezano s tim drugim preklopnikom u njegovo sučelje s najnižim prioritetom. Prioritet svih sučelja je predefiniran na vrijednost 128 i mijenja se u rasponu od 0 do 240 u inkrementima od 16
4. ako je preklopnik s više sučelja povezan na neki drugi preklopnik, koji je na putu prema vršnom preklopniku, a prioritet je sučelja isti na svim sučeljima, tada se koristi najnižu oznaku sučelja na susjednom preklopniku. Vršno sučelje postaje ono koje je povezano u sučelje s najnižom oznakom (npr. *FastEthernet* 0/5 bolje je od *FasteEthernet* 0/7).

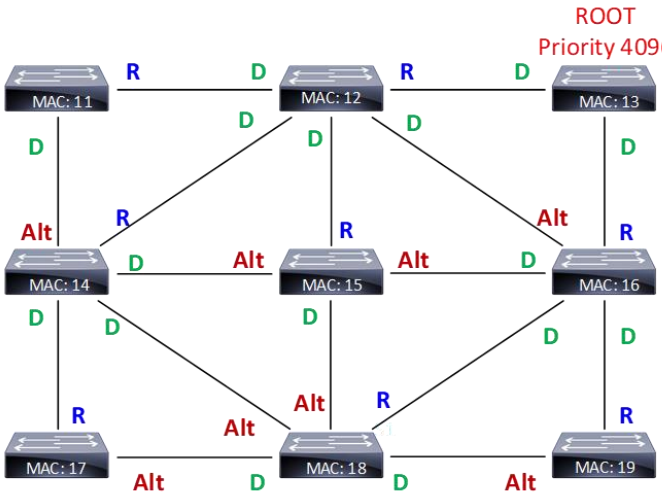
Nakon što preklopnici odrede vršno sučelje, a povezani su međusobno dodatnim vezama, moraju utvrditi odabrana (engl. *Designated*) i neodabrana (engl. *Non-designated*) sučelja. Za to koriste sličan proces kao i za određivanje vršnog sučelja:

1. Preklopnik koji ima veći kumulativni trošak do vršnog preklopnika mora svoje sučelje staviti u neodabrano stanje (engl. *Non-designated*) i takvo sučelje zapravo blokira sav promet i samo „osluškuje“ BPDU okvire kako bi „znalo“ da mora ostati u neodabranom stanju
2. Ako je kumulativni trošak do vršnog preklopnika za oba preklopnika isti, onda će preklopnik koji ima veći BID (Prioritet + VLAN ID + MAC) svoje sučelje staviti u neodabrano stanje (engl. *Non-designated*)
3. Ako je konfigurirani prioritet isti (npr. predefinirana vrijednost od 32768), onda će onaj preklopnik koji ima veću MAC adresu svoje sučelje staviti u neodabrano stanje (engl. *Non-designated*).

Nakon tih se koraka STP topologija smatra konvergiranom i stabilnom dok se ne dogodi neka promjena u fizičkoj (npr. spajanje ili odspajanje nekog kabela) ili logičkoj topologiji (npr. stvaranje ili brisanje nekog VLAN-a u PVST+ odnosno RPVST+ verziji STP-a).

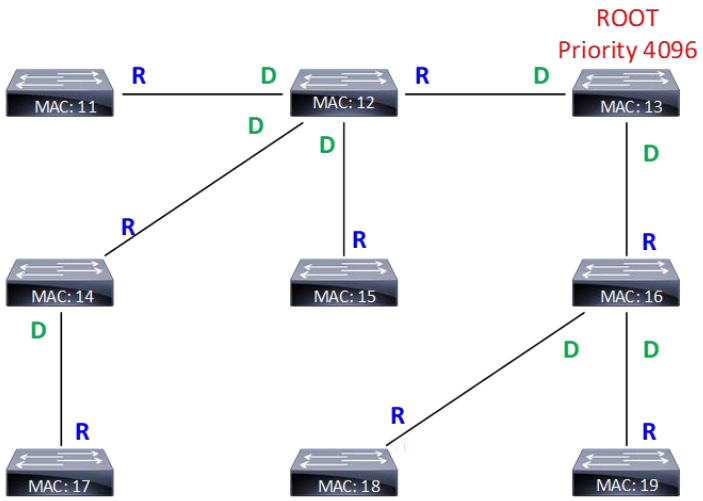
Danas se u mrežama koristi implementacija STP protokola koja podržava rad s VLAN-ovima. VLAN ID je sastavni dio BID-a koji se sastoji od prioriteta preklopnika, VLAN ID-a (engl. *Extended system ID*) i MAC adrese. U ovom priručniku neće biti upućivanja na „stariju“ osnovnu implementaciju STP-a jer se više ne koristi u praksi.

Princip je da se u mreži prvo odabere vršni preklopnik (engl. *Root Bridge*), a od vršnog se preklopnika kreira stablasta struktura do svih preklopnika bez petlji. Algoritam koji unutar STP protokola kreira stablastu strukturu preklopnika, zove se STA algoritam (engl. *Spanning Tree Algoritam*). Na slici 120. prikazana je mreža s redundantnim vezama.



Slika 120. Lokalna mreža s redundantnim vezama

Svaki je preklopnik na slici spojen višestrukim redundantnim vezama sa susjednim preklopnicima. Bez STP algoritma ta topologija nikada ne bi bila funkcionalna zbog višestrukih beskonačnih petlji koje se na njoj mogu dogoditi. STP koristeći STA algoritam (engl. *Spanning Tree Algoritam*) blokira redundantne veze tako da u konačnici ostanu aktivne samo one veze koje, iz perspektive svakog pojedinog preklopnika, predstavljaju najkraći put do vršnoga.



Slika 121. Aktivne veze u LAN mreži nakon što je STA konvergirao („izračunao najbolje“) veze

Vršni je preklopnik na slici s prioritetom 4096 (manje je bolje), a rezultat STA algoritma koji je svaki preklopnik iskoristio da za svoju poziciju odredi koja sučelja će biti vršna, a koja će biti odabrana odnosno neodabrana. Na slici 121. vidljivo je da je na svim preklopnicima, osim na preklopniku koji ima prioritet 4096, predefinirana vrijednost

prioriteta od 32768, što znači da se za usporedbu koristi MAC adresa preklopnika. To nije MAC adresa sučelja na preklopniku, već tzv. *Base Ethernet MAC address*, koja predstavlja cijeli preklopnik, i tu MAC adresu doznaje se naredbom „show version“ ili naredbom „show tech-support“.

2.7.1. Uloge sučelja — STP

Uloge sučelja u kontekstu STPa mogu biti razne ovisno o tome kako je topologija konvergirala. Bitno je znati da jedino vršni preklopnik neće imati vršna sučelja ni blokirana sučelja. Sva sučelja na vršnom preklopniku bit će odabrana (engl. *Designated*) sučelja i prosljeđivat će promet.

- **vršno sučelje** (engl. *Root Port*) – nalazi se samo na preklopnicima koji nisu vršni. To je sučelje s najboljim putem do vršnog preklopnika. Tim se sučeljem promet prosljeđuje prema vršnom preklopniku, a dopušteno je samo jedno takvo sučelje na preklopniku. Ta vrsta sučelja može ažurirati MAC tablicu;
- **odabrano sučelje** (engl. *Designated Port*) – ta se sučelja mogu nalaziti na svim preklopnicima, a na vršnima su sva sučelja u tom statusu. Kod preklopnika koji nisu vršni, preklopnici to sučelje primaju i prosljeđuju okvire prema vršnom preklopniku ako je to potrebno;
- **neodabrano sučelje** (engl. *Non-designated Port*) – to je blokirano sučelje, ne prosljeđuje okvire i ne ažurira MAC tablicu.

Ukoliko je sučelje na preklopniku isključeno (engl. *Disabled*), utoliko ne sudjeluje u STP procesu.
Na primjeru na slici 68. za vršni je preklopnik odabran preklopnik s prioritetom 4096. Njegova su sučelja prema ostalim preklopnicima odabrana sučelja (engl. *Designated Port*). Svi ostali preklopnici će u odnosu na vršni preklopnik na temelju prethodno spomenutih kriterija odrediti ulogu svojih sučelja.

Kada preklopnici počnu razmjenjivati BPDU okvire, STA algoritam kreira stablastu strukturu bez petlji. Taj proces zahtijeva određeno vrijeme. Ako neko sučelje postane neaktivno i dođe do promjene u topologiji mreže, također je potrebno određeno prijelazno razdoblje, odnosno vrijeme konvergencije.

U prijelaznim razdobljima sučelja mogu biti u sljedećim stanjima (stanja su različita od uloga koje sučelja mogu imati):

- **blokirano** (engl. *Blocking*) – sučelje u tom stanju neodabrano je sučelje (engl. *Non-designated Port*) i ne sudjeluje u prosljeđivanju okvira. Sučelje samo prima BPDU okvire da bi pratilo STP topologiju mreže, a na jednom je preklopniku uvijek povezano u odabrano sučelje na drugom preklopniku;
- **slušanje** (engl. *Listening*) – u tom stanju sučelje prima i šalje BPDU okvire. Slanjem BPDU okvira obavještava susjedne preklopnike da se priprema za sudjelovanje u aktivnoj topologiji;
- **učenje** (engl. *Learning*) – u tom stanju sučelje šalje i prima BPDU okvire te se priprema za prosljeđivanje podatkovnih okvira tako da počinje ažurirati MAC tablicu;
- **prosljeđivanje** (engl. *Forwarding*) – sučelje se smatra dijelom aktivne topologije. Prihvaća i prosljeđuje podatkovne okvire i ažurira MAC tablicu. Također prima i šalje BPDU okvire;
- **onemogućeno** (engl. *Disabled*) – sučelje je administrativno isključeno (engl. *Administratively Shut Down*). Ta sučelja nisu aktivna pa ne mogu ni sudjelovati u STP procesu.

Vrijeme koje će sučelje provesti u nekom od stanja ovisi o definiranim BPDU vremenskim brojačima (engl. *BPDU Timer*). Postoje sljedeći BPDU vremenski brojači:

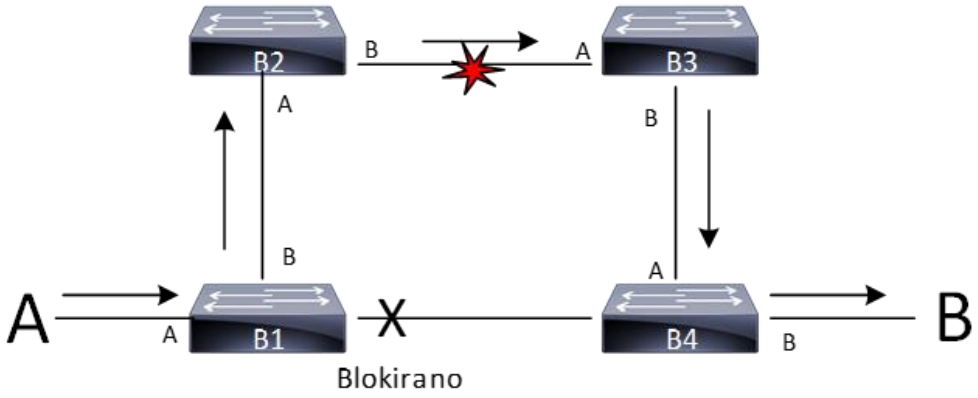
- ***hello time*** – vrijeme između slanja dvaju BPDU okvira. Predefinirano vrijeme je 2 sekunde;
- ***forward delay*** – vrijeme koje preklopnik čeka prije prelaska u novo stanje, npr. slušanja (engl. *Listening*) u učenje (engl. *Learning*), pa u prosljeđivanje (engl. *Forwarding*). Predefinirano vrijeme je 15 sekundi;
- ***maximum age*** – najdulje vrijeme u kojemu preklopnik čuva BPDU informacije dobivene sa sučelja, a da ne dobije novi BPDU okvir. Predefinirano je vrijeme 10 x *hello time*, što je 20 sekundi. Ako u intervalu od 20 sekundi ne dobije konfiguracijski BPDU na sučelju, smatra da je došlo do gubitka veze s izvorom BPDU okvira i BPDU se više ne smatra važećim.

Uzevši vrijednosti predefiniranih brojača konvergencija STP topologije, nakon otkaza neke značajne veze ili njezina dodavanja, koja utječe na promjenu topologije, može trajati i do 50 sekundi. Vrijednosti vremenskih brojača mogu se promijeniti. Promjene

se pridruže vršnom preklopniku (engl. *Root Bridge*), a vršni ih preklopnik pomoću BPDU okvira prosljeđuje ostalim preklopnicima u mreži.

Obavijest o promjeni topologije (engl. *Topology Change Mechanism*)

U slučaju da se dogodi promjena u STP topologiji, nužno je da se svi preklopnici na koje utječe ta promjena prilagode novoj STP topologiji. To uključuje i promjenu zapisa u tablicama MAC adresa koje preklopnici održavaju. Predefinirano je vrijeme pamćenja MAC adresa u tablici MAC adresa na *Cisco* preklopnicima 300 sekundi, a to je jako dugo za moderne mreže. Svrha je mehanizma obavijesti o promjeni topologije da se to vrijeme znatno smanji kako bi se komunikacija kroz mrežu brzo ponovno uspostavila.



Slika 122. Prikaz otkaza u jednostavnoj topologiji

Na slici 122. može se vidjeti da u slučaju otkaza primarne veze između B2 i B3, preklopnik B1 bi i dalje imao zapis o MAC adresi računala B na svom sučelju B, ali to više nema smisla, jer tim putem više nije moguće doći do računala B. Sve okvire koje preklopnik B1 pošalje koristeći svoju tablicu MAC adresa odbacit će preklopnik B2, koji više nema funkcionalnu vezu prema preklopniku B3. Zato se koristi mehanizam obavijesti o promjeni topologije kako bi preklopnik B1 obrisao svoju MAC tablicu, zatim odblokirao sučelje prema preklopniku B4 i na to sučelje naučio potrebne MAC adrese. Bitno je znati kako poruke obavijesti o promjeni topologije ne uzrokuju promjenu topologije i njezine ponovne izračune, već su one samo posljedica neke promjene. BPDU okvire šalje vršni preklopnik niz stablastu strukturu preklopnika, dok ih ostali preklopnici prosljeđuju dalje u mrežu, ali nikada ne šalju na vršna sučelja. Da bi BPDU okviri ipak došli do vršnog preklopnika, postoje posebni BPDU okviri naziva TCN (engl. *Topology Change Notification*) koji se šalju susjednom preklopniku u smjeru vršnog

preklopnika kroz vršno sučelje (engl. *Root Port*). To se događa kada se pojavi bolji put prema vršnom preklopniku ili ako dođe do otkaza nekog od vršnih sučelja u mreži. Susjedni preklopnik potvrđuje primitak TCA okvirom (engl. *Topology Change Ack*) i generira novi TCN koji šalje dalje prema vršnom preklopniku. Kada vršni preklopnik dobije informaciju o promjeni topologije u mreži, počinje slati u mrežu BPDU okvire s postavljenim TC (engl. *Topology Change*) bitom. Na taj način obavještava preklopnike u mreži da je došlo do promjene topologije. Svi preklopnici primaju TC obavijest na sva sučelja (blokirana sučelja i sučelja koja prosljeđuju okvire). Nakon dobivenog BPDU okvira s postavljenim TC bitom, preklopnici smanjuju vrijeme u kojemu čuvaju zapise u MAC tablici s predefiniranih 300 sekundi na vrijeme od 15 sekundi (predefinirano *Forward Delay* vrijeme). Razlog je potreba za bržim brisanjem starih zapisa u MAC tablici i kreiranjem novih zbog promjene topologije mreže. Nakon toga mreža konvergira u novu topologiju razmjenom konfiguracijskih BPDU okvira. Što je veća mreža i što je više računala u mreži, to je veća vjerojatnost da će se slati TCN poruke. Na primjer, ponovno pokretanje računala uzrokovat će pojavu TCN poruka jer je sučelje na preklopniku također doživjelo promjenu stanja. U ekstremnim slučajevima može se dogoditi da se mreža nađe u neprestanom slanju TCN poruka i u neprestanom stanju konvergencije, dakle može se dogoditi da mreža postane neupotrebljiva. Da bi se spriječilo generiranje TCN poruka, prilikom ponovnog pokretanja računala sučelja na koja su povezana računala treba konfigurirati kao *Portfast* sučelja naredbom: „spanning-tree portfast“ tako isključiti STP na tim sučeljima. To se radi samo ako je sigurno da se na neko sučelje nikada neće povezati preklopnik. Uz *Portfast* funkcionalnost obično se uključuje i mehanizam *BPDU Guard*, koji štiti mrežu od BPDU poruka. Kada na sučelje koje je konfigurirano kao *Portfast* dođe BPDU poruka, to sučelje će biti isključeno. Ta se osnovna verzija STP mehanizma danas praktički više nigdje ne koristi. Umjesto „običnog“ STP-a, češće se koristi RSTP, i to verzija koja podržava VLAN-ove, o čemu će biti riječi kasnije.

RSTP PROTOKOL

Problem STP protokola spora je konvergencija koja može trajati do 50 sekundi (STP je definiran u IEEE 802.1D) pa je STP protokol evoluirao u RSTP protokol (engl. *Rapid Spanning Tree Protocol*) koji je definiran u IEEE 802.1w. RSTP omogućava znatno bržu konvergenciju mreže – desetak puta brže i ne ovisi o vremenskim brojačima (engl. *Timers*) nakon promjene topologije u mreži.

Osnovne značajke RSTP protokola su:

- kompatibilan je sa STP protokolom (IEEE 802.1D);
- zadržava STP terminologiju, mnogi su parametri ostali nepromijenjeni (BPDU format i izbor vršnog preklopnika su isti);
- za ubrzanje konvergencije:
    - ne koristi IEEE 802.1D vremenske brojače,
    - redefinira tipove sučelja i njihova moguća stanja.

RSTP definira sljedeće tipove sučelja:

- **vršno sučelje** (engl. *Root Port*) – sučelje na svakom preklopniku koji nije vršni. Preklopnik može imati samo jedno vršno sučelje. To je sučelje s najboljim putem do vršnog preklopnika i njime se prosljeđuje promet prema njemu. U konvergiranoj mreži pretpostavlja stanje prosljeđivanja okvira (engl. *Forwarding State*);
- **odabrano sučelje** (engl. *Designated Port*) – to se sučelje može nalaziti na svim preklopnicima. Na vršnim su preklopnicima sva sučelja u tom statusu. U preklopnicima koji nisu vršni, to sučelje prima i prosljeđuje okvire prema vršnom preklopniku ako je to potrebno. U konvergiranoj mreži pretpostavlja stanje prosljeđivanja okvira (engl. *Forwarding State*);
- **alternativno sučelje** (engl. *Alternate Port*) – sučelje koje nudi alternativni put do vršnog preklopnika. Alternativno sučelje pretpostavlja odbačeno stanje (engl. *Discarding State*) u mreži koja je konvergirala;
- **rezervno sučelje** (engl. *Backup Port*) – to je sučelje s redundantnom vezom na segmentu. Rezervno sučelje pretpostavlja odbačeno stanje (engl. *Discarding State*) u mreži koja je konvergirala;
- **rubno sučelje** (engl. *Edge Port*) – to je sučelje koje nikada neće biti spojeno na drugi preklopnik. Odmah odlazi u stanje prosljeđivanja, bez da prolazi prijelazne faze. Kada rubno sučelje dobije BPDU, postaje normalno sučelje u kontekstu *Spanning Tree* algoritma.

Obavijest o promjeni topologije

Obavijest o promjeni topologije u RSTP protokolu od preklopnika na kojem je došlo do promjene ne ide samo prema vršnom preklopniku kako bi vršni preklopnik

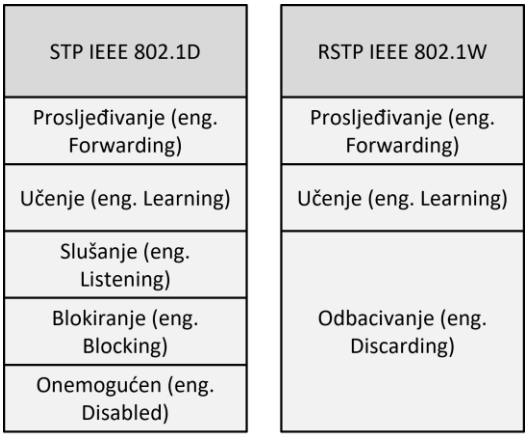
obavještavao sve ostale, već RSTP preklopnik odmah obavještava susjede, susjedi svoje susjede itd. Tako informacija o promjeni brže prođe mrežom.

RSTP definira sljedeća stanja sučelja:

- **odbacivanje** (engl. *Discarding*) – prihvaća i procesuira BPDU okvire. Ne prosljeđuje okvire (engl. *Frames*);
- **učenje** (engl. *Learning*) – u tom se stanju sučelje priprema za prosljeđivanje okvira i počinje ažurirati MAC tablicu;
- **prosljeđivanje** (engl. *Forwarding*) – sučelje se smatra dijelom aktivne topologije. Prihvaća i prosljeđuje okvire. Jednako tako prima i šalje BPDU okvire.

U svim stanjima sučelja prihvaćaju i procesuiraju BPDU okvire. Sučelja koja su u STP protokolu u stanjima „blokirano“, „slušanje“ i „onemogućeno“, ne prosljeđuju okvire. Ta su tri stanja u RSTP protokolu spojena u stanje „odbačeno“ (engl. *Discarding*).

U sljedećoj je tablici usporedba STP i RSTP stanja sučelja.



Slika 123. Usporedba STP i RSTP stanja na sučeljima

RSTP još definira granična (engl. *Edge Port*) i neograničena sučelja (engl. *Non-edge Port*). Granična sučelja preklopnika nisu spojena na sučelje drugog preklopnika. Nakon što su omogućena (engl. *Enabled*), odmah prelaze u stanje prosljeđivanja okvira (engl. *Forwarding State*). RSTP granična sučelja odgovaraju *Cisco Portfast* tehnici kod STP protokola pomoću koje sva sučelja koja su priključena na krajnje uređaje odmah odlaze u stanje prosljeđivanja, preskačući STP prijelazna stanja (slušanja i učenja), te tako ubrzavaju aktiviranje sučelja prema krajnjem uređaju. Za razliku od *Portfast* tehnike,

RSTP granična sučelja gube taj status ako dobiju BPDU okvir. Tada postaju normalno STP sučelje. Osim RSTP protokola, postoji i varijacija STP protokola namijenjena za rad u okruženjima s više VLAN-ova, a to su MST, PVST+ i RPVST+.

2.7.2. PVST+ i RPVST+

Te je verzije STP protokola osmislila tvrtka *Cisco* kako bi implementacija STP mehanizma u mreži bila fleksibilnija. U načelu PVST+ funkcionira isto kao STP, dok RPVST+ funkcionira isto kao RSTP, ali se logičke topologije u mreži izračunavaju za svaki VLAN posebno. Prednost je što konvergencija unutar jednog VLAN-a ne utječe na STP topologiju ostalih VLAN-ova. Također što se mogu različiti preklopnici konfigurirati da djeluju kao vršni za određene VLAN-ove i tako postići da se sve veze između preklopnika koriste za prosljeđivanje prometa. Neke su veze blokirane samo za dio VLAN-ova, dok su neke druge veze blokirane za drugi dio VLAN-ova. U slučajevima da u mreži postoji jako puno VLAN-ova, recimo 4000, tada bi preklopnici trebali izračunavati logičke topologije za njih 4000, pa je problem koji se može pojaviti toliko opterećenje preklopnika, da cijela mreža postane nefunkcionalna. Stoga postoji MST protokol koji omogućava kreiranje jedne instance STP protokola za više VLAN-ova, što znatno rasterećuje preklopnike. MST se koristi u velikim mrežama koje imaju više stotina ili čak tisuća VLAN-ova te nije predmet ovog priručnika.

2.8. Agregacija fizičkih veza u mreži — Etherchannel

Etherchannel je tehnologija koja omogućava povezivanje više fizičkih *Ethernet* veza na preklopnicima ili usmjernicima u jednu logičku *Ethernet* vezu. Najznačajnija prednost *Etherchannel* tehnologije tolerancija je na otkaze. Obično se češće koristi za veze između preklopnika. Otkaže li jedna od fizičkih veza unutar *Etherchanela*, promet će preuzeti preostale fizičke veze. *Etherchannel* može biti kreiran između dva do osam aktivnih *Ethernet* sučelja s dodatnih jedan do osam neaktivnih sučelja, koji se aktiviraju ako neko od aktivnih sučelja postane neaktivno. Konkretne specifikacije, naredbe i mogućnosti ovise o modelu uređaja i verziji operativnog sustava.

Sve veze koje čine jednu *Etherchannel* vezu, STP tretira kao jednu logičku vezu tako da ne može doći do preklopničkih petlji. Također, preklopnici će u kontekstu implementacije STP-a agregirane veze u *Etherchannelu* gledati kao veze s manjim

troškom od pojedinih *Ethercahnnel* veza, što znači da će *Ethercahnnel* utjecati na izgled STP topologije. *Etherchannel* može biti konfiguriran i kao VLAN TRUNK veza. Bez obzira što se može agregirati više fizičkih veza u jednu logičku vezu, brzina kojom bilo koji uređaj može komunicirati *Etherchannel* vezom može biti najviše jednaka brzini fizičkog sučelja *Etherchannel* veze, a nikako zbroju pojedinačnih brzina veza unutar nje. To se događa zato što *Etherchannel* veza ne raspoređuje promet (engl. *Load Balancing*) za jedno računalo preko svih dostupnih fizičkih veza, već koristi algoritam koji na temelju određenih parametara odlučuje na koju će fizički vezu slati promet pojedinog uređaja spojenog na preklopnik. Ukupno gledajući, rezultat je veća propusnost, ali ne za pojedino računalo, već za sva koja komuniciraju preko preklopnika povezanih *Etherchannel* vezom.

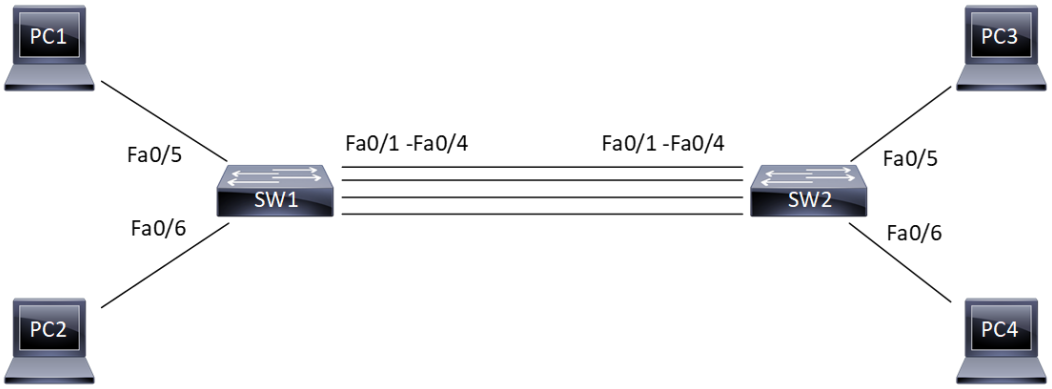
Slika 124. Primjer mogućnosti preklopnika Catalyst 3750/3560 za raspoređivanje prometa fizičkim vezama u Etherchannel veze

Za najbolju iskoristivost mogućnosti raspoređivanja prometa na agregirane fizičke veze preporuka je koristiti dvije srednje opcije. Predefinirano je na preklopnicima *Cisco* na temelju izvorišne MAC adrese.

PAgP je protokol u vlasništvu tvrtke *Cisco* čiji su inženjeri osmislili taj protokol. PAgP se koristi za agregaciju fizičkih veza između uređaja. PAgP osigurava da sva sučelja koja su pridružena zajedničkoj logičkoj vezi imaju jednaku konfiguraciju (brzina, tip veze, VLAN informacije i slično), te dodaje nove fizičke veze u logičku vezu i upravlja otkazima veza. STP algoritam tretira sučelja koja su povezana PAgP protokolom na preklopniku kao jedno sučelje. Sučelja možemo konfigurirati u modu *Desirable* i *Auto*. Važno je da barem jedna strana bude konfigurirana u modu *Desirable* tako da inicira pregovaranje.

LACP protokol definiran je IEEE 802.3ad specifikacijom. Kao i PAgP protokol, automatski pregovara o povezivanju više veza u jednu logičku vezu slanjem LACP paketa susjednim sučeljima. Budući da je LACP IEEE standardiziran, omogućava korištenje *Etherchannel* tehnologije s preklopnicima različitih proizvođača, za razliku od PAgP protokola koji je u vlasništvu tvrtke *Cisco*. Sučelja možemo konfigurirati kao aktivna (engl. *Active*) i pasivna (engl. *Passive*) i važno je da barem jedna strana bude konfigurirana kao aktivna tako da inicira pregovaranje.

Također, sučelja se mogu konfigurirati kao dio *Etherchannel* grupe bez pregovaranja, tako da se kompatibilna sučelja koja će biti agregirana na oba povezana uređaja, konfiguriraju u ON način rada.



Slika 125. Primjer primjene Etherchannel mehanizma za povećanje redundancije i ukupne propusnosti između dva preklopnika

Primjer konfiguracijskih naredbi za *Etherchannel* koristeći PAgP i LACP na slici je 125.

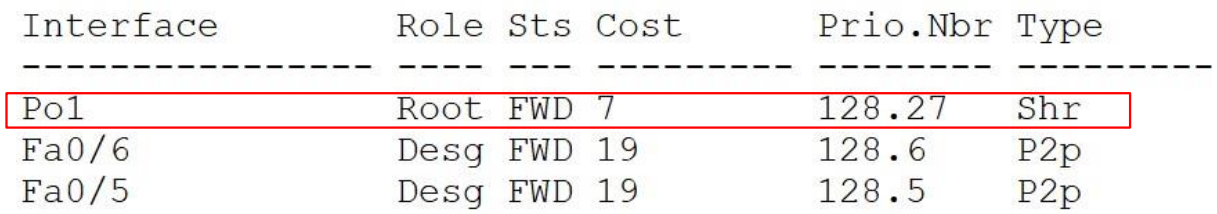
```
SW1>enable
SW1#configure terminal
SW1(config)#interface range FastEthernet 0/1-4
SW1(config-if-range)#channel-group 1 mode active
SW1(config-if-range)#exit
SW1(config)#interface Port-channel 1
SW1(config-if)#switchport trunk encapsulation dot1q
SW1(config-if)#switchport mode trunk
SW1(config-if)#end
SW1#show etherchannel summary
```

```
SW2>enable
SW2#configure terminal
SW2(config)#interface range FastEthernet 0/1-4
SW2(config-if-range)#channel-group 1 mode passive
SW2(config-if-range)#exit
SW2(config)#interface Port-Channel 1
SW2(config-if)#switchport trunk encapsulation dot1q
SW2(config-if)#switchport mode trunk
SW2(config-if)#end
SW2#
```

Slika 126. Konfiguracija dva međusobno povezana preklopnika za LACP u PAcketTracer simulatoru

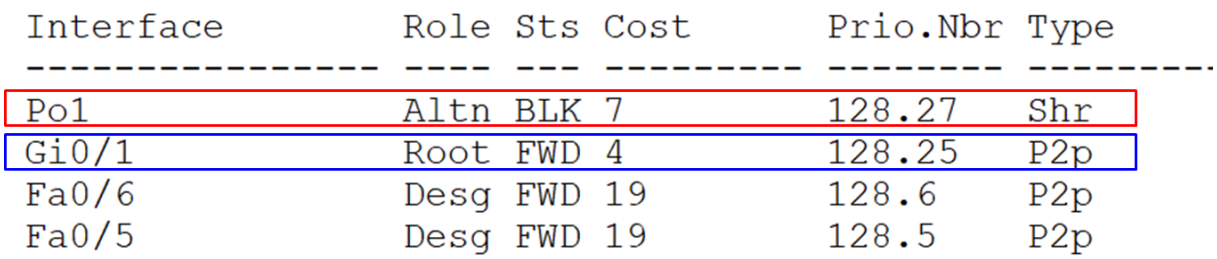
Nakon što je agregacija uspješno konfigurirana, STP više neće blokirati ni jedno fizičko sučelje, kao što je slučaj bez agregacije. STP agregirane veze tretira kao jedno sučelje

i smatra boljima od pojedinačnih veza, pa tako za četiri agregirane fizičke veze propusnosti 100Mbps, nakon agregacije dodjeljuje trošak 7, što je bolje od pojedinačnih veza koje imaju trošak 19. Međutim, doda li se sučelje brzine 1Gbps, tada će preklopnik agregirane veze blokirati i one neće biti korištene za prosljeđivanje prometa.



Interface	Role	Sts	Cost	Prio.Nbr	Type
Po1	Root	FWD	7	128.27	Shr
Fa0/6	Desg	FWD	19	128.6	P2p
Fa0/5	Desg	FWD	19	128.5	P2p

Slika 127. STP uloga sučelja za agregirane veze koja se sastoji od četiri fizičke veze propusnosti 100Mbps



Interface	Role	Sts	Cost	Prio.Nbr	Type
Po1	Altn	BLK	7	128.27	Shr
Gi0/1	Root	FWD	4	128.25	P2p
Fa0/6	Desg	FWD	19	128.6	P2p
Fa0/5	Desg	FWD	19	128.5	P2p

Slika 128. STP uloga sučelja agregirane veze uz dodano sučelje propusnosti 1Gbps

2.9. Redundancija izlaza iz LAN mreže

Kako bi računala mogla komunicirati s drugim računalima koja se nalaze izvan vlastite mreže ili na internetu, potrebno je imati funkcionalan mrežni pristupnik (engl. *Gateway*). Mrežni pristupnik jedan je od najvažnijih elemenata svake mreže i zbog toga je potrebno osigurati maksimalnu dostupnost takvog uređaja. Jedan od načina za to je učiniti ga redundantnim, što znači da postoje barem dva uređaja koja istovremeno obavljaju tu funkciju u mreži. Protokoli koje koristimo za osiguravanje redundancije mrežnog pristupnika zovu se *First Hop Redundancy Protocols*, i u tu skupinu spadaju HSRP, VRRP i GLBP. U ovom će poglavlju biti opisani HSRP i VRRP protokoli.

- HSRP (engl. *Hot Standby Router Protocol*)
- VRRP (engl. *Virtual Router Redundancy Protocol*)
- GLBP (engl. *Gateway Load Balancing Protocol*)

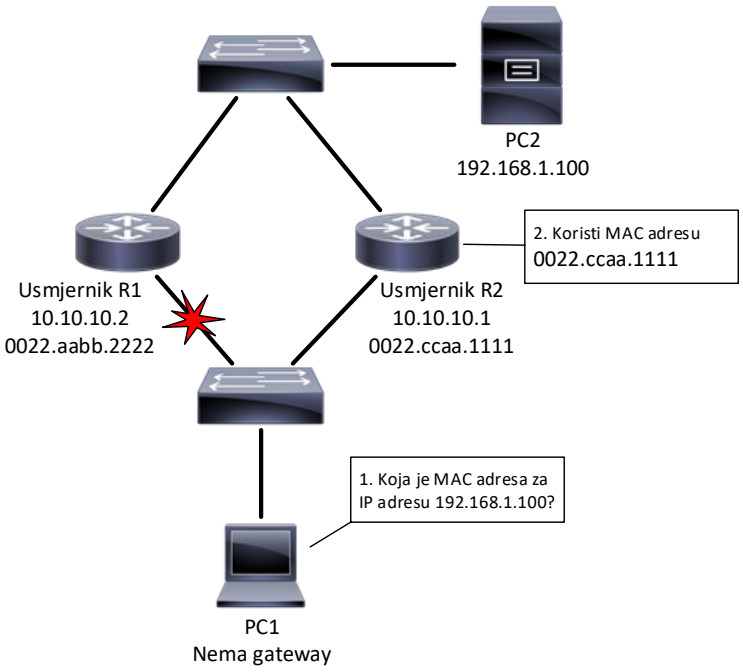
Koristeći protokole HSRP ili VRRP, redundanciju se osigurava tako što jedan uređaj funkcionira kao aktivni mrežni pristupnik, dok su mu ostali uređaji u grupi podrška. U slučaju otkaza aktivnog mrežnog pristupnika, jedan će od pomoćnih uređaja preuzeti ulogu aktivnoga mrežnog pristupnika.

Za razliku od HSRP i VRRP protokola, u GLBP protokolu može istovremeno biti više aktivnih mrežnih pristupnika koji međusobno dijele opterećenje (engl. *Loadbalance*).

Prije pojave gore navedenih protokola mreže su se oslanjale na *ProxyARP* mehanizam i statičku konfiguraciju mrežnih pristupnika.

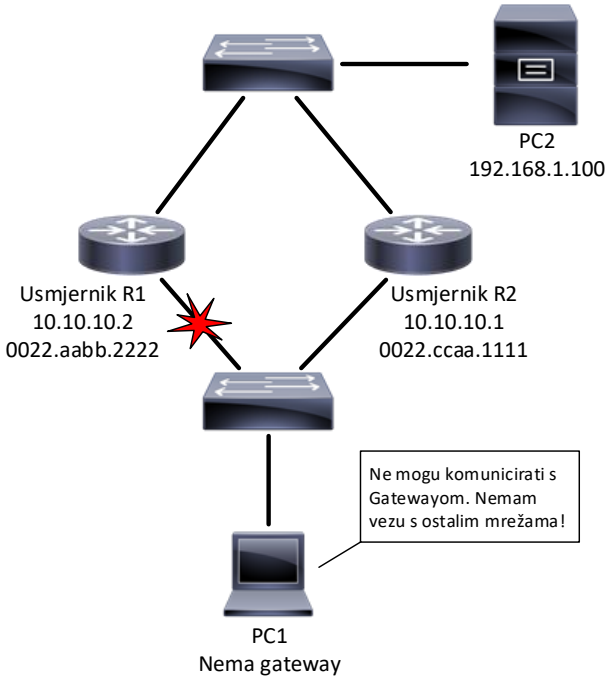
2.9.1. Proxy ARP

Prije nego je na većini računala bilo moguće konfigurirati IP adresu mrežnog pristupnika (engl. *Gateway*), mreže su koristile mehanizam *ProxyARP*. Usmjernik koji je bio mrežni pristupnik (engl. *Gateway*) za određeni segment mreže je, koristeći mehanizam *ProxyARP*, omogućavao računalima komunikaciju s drugim računalima izvan svoje mreže. Kao što je prikazano na slici 129., ako bi usmjernik R1 primio ARP zahtjev od nekog računala PC1 za IP adresu računala PC2 koje nije na istom sučelju na koje je spojeno računalo PC1, tada usmjernik R1 kao odgovor pošalje svoju MAC adresu sučelja Fa0/0, i na taj način se usmjernik zapravo predstavlja kao odredište (PC2). U nastavku usmjernik prosljeđuje promet i djeluje samo kao posrednik u komunikaciji između PC1 i PC2, no toga računala nisu svjesna, već misle da komuniciraju direktno na lokalnoj mreži. Ako otkaže usmjernik R1, računalo PC1 nastavit će slati promet na njegovu MAC adresu jer mu je ta informacija u ARP memoriji. Nakon što računalo PC1 iz ARP memorije izbriše unos MAC adrese usmjernika R1 koji je otkazao, ponovno će uspostaviti komunikaciju posredstvom usmjernika R2, tako što će poslati ARP zahtjev kao na početku komunikacije. Dok taj prekid traje, ni jedno računalo ne može komunicirati izvan svoje mreže.



Slika 129. Princip funkcioniranja Proxy ARP mehanizma

Zbog nedostataka *Proxy ARP* mehanizma rabi se dedicirani mrežni pristupnik (engl. *Gateway*) uređaja koji koristi neko računalo. U tom se slučaju informacija o mrežnom pristupniku (engl. *Gateway*) na računalu konfigurira ručno, pa nije potreban *ProxyARP* mehanizam. Problem je što svako računalo može imati samo konfiguriran jedan mrežni pristupnik (engl. *Gateway*) bez obzira koliko ih je u mreži. Na slici 130. prikazano je što bi se dogodilo ako otkaže mrežni pristupnik (engl. *Gateway*) koji su računala koristila za izlaz iz svoje mreže. Bez obzira što se cijela mreža može ponovno konvergirati koristeći dinamički protokol usmjeravanja (npr. OSPF) i znati gdje se nalazi podrmreža (engl. *Subnet*) u kojem je računalo PC1, problem PC1 je što njegov mrežni pristupnik (engl. *Gateway*) više nije dostupan i više nema načina da pošalje promet izvan svoje mreže. Ono što se može napraviti jest promijeniti mrežni pristupnik koji PC1 koristi (umjesto R1 konfigurirati R2), ali kada bi umjesto PC1 imali nekoliko stotina ili tisuća računala, tada bi promjena mrežnog pristupnika na svim računalima trajala predugo ili bila praktički nemoguća. Zbog toga je mrežni pristupnik (engl. *Gateway*) evoluirao u FHRP mehanizam.



Slika 130. Primjer otkaza veze s mrežnim pristupnikom (engl. Gateway)

2.9.2. HSRP

HSRP je protokol koji je razvio *Cisco* i služi za osiguravanje redundancije mrežnog pristupnika (engl. *Gateway*) bez potrebe za dodatnom konfiguracijom na krajnjim uređajima. Računala će imati ili statičku konfiguraciju IP postavki ili konfiguraciju posredstvom DHCP protokola.

Dvije su verzije HSRP protokola:

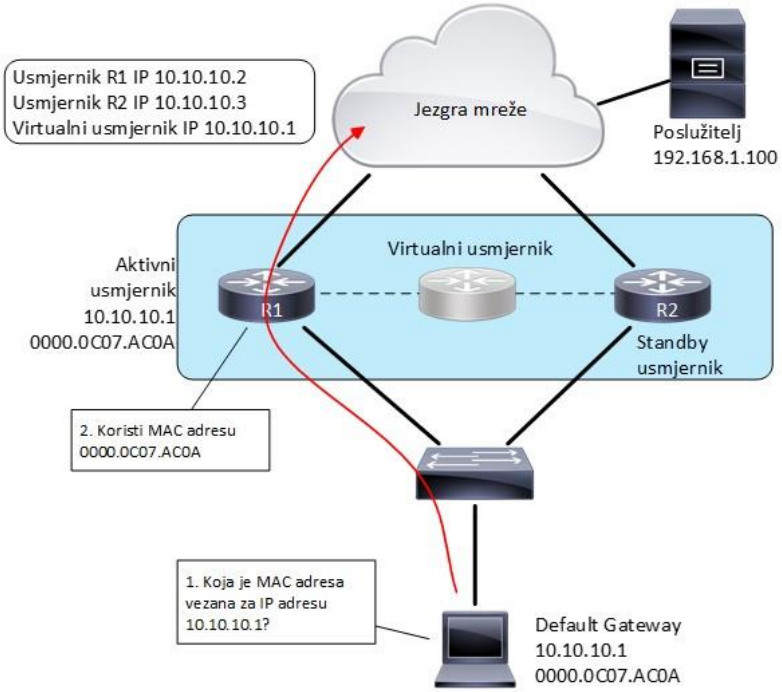
HSRP v1 omogućava konfiguraciju oznaka grupa od 0 do 255 na jednom sučelju, s tim da na dugom sučelju možemo ponovno koristiti iste brojeve. Virtualna se MAC adresa ne konfigurira, već je ona određena automatski u obliku 0000.0C07.ACXX, gdje je X broj grupe koji odabiremo prilikom konfiguracije HSRP protokola. HSRP v1 šalje *Hello* poruke na višeodredišnu (engl. *Multicast*) IP adresu 224.0.0.2 (što predstavlja sve usmjernike na L2 segmentu mreže).

HSRP v2 omogućava konfiguraciju od 0 do 4095 grupa što nam omogućava da HSRP grupe slažemo prema oznakama VLAN-ova. U HSRPv2 MAC adresa se također ne konfigurira, već je automatski generirana u obliku 0000.0C9F.FXXX, gdje je X broj HSRP grupe. *Hello* poruke šalju se na višeodredišnu (engl. *Multicast*) adresu 224.0.0.102

HSRP v1 je predefinairano uključen na *Cisco* usmjernicima i da bi svi usmjernici uspješno komunicirali, treba koristiti istu verziju HSRP protokola.

HSRP podržava i autentikaciju potrebnu kao osiguranje od napada na HSRP grupu, pa će u njoj moći sudjelovati samo oni usmjernici sa zajedničkom lozinkom. Autentikacija može biti u tekstualnom obliku ili posredstvom MD5 algoritma, i na svim usmjernicima u HSRP grupi mora biti na isti način konfigurirana.

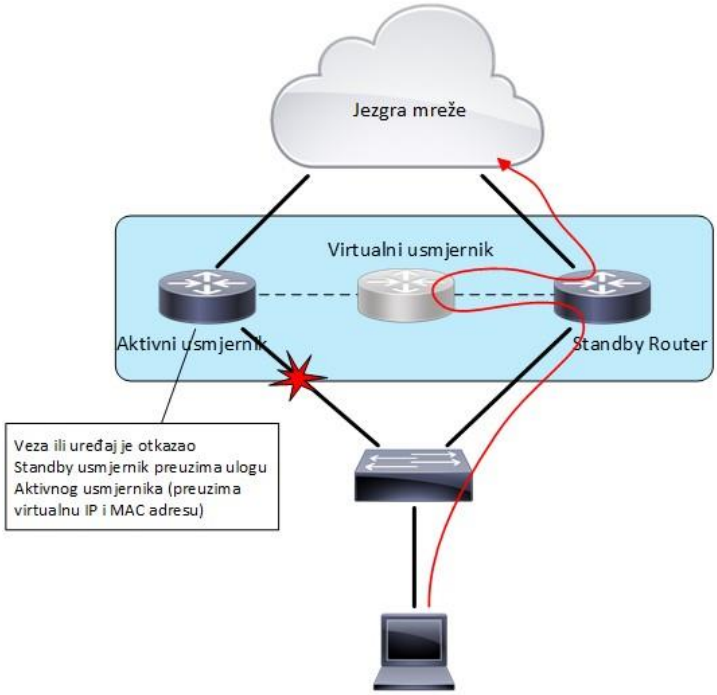
HSRP se konfigurira između dva ili više usmjernika koji služe kao izlaz iz mreže. Svi usmjernici na nekom L2 segmentu predstavljaju se računalima s jednom IP i jednom MAC adresom koju zovemo virtualna IP adresa, odnosno virtualna MAC adresa. Skupina usmjernika koja dijeli virtualnu IP i MAC adresu zove se još i „virtualni usmjernik“. Onaj usmjernik koji je konfiguriran kao aktivni mrežni pristupnik (engl. *Gateway*), odgovoran je za prosljeđivanje prometa koji dolazi na tu virtualnu IP adresu. Ako taj aktivni usmjernik otkaže, tada njegovu ulogu preuzima slijedeći usmjernik konfiguriran za tu ulogu. Osim što ima konfiguriranu virtualnu IP adresu, svaki usmjernik ima i svoju jedinstvenu IP adresu koja je u istoj podmreži kao virtualna IP adresa. Sva će računala na određenom L2 segmentu mreže za predefinirani mrežni pristupnik (engl. *Default Gateway*) koristiti virtualnu IP adresu konfiguriranu na svim usmjernicima u nekoj HSRP grupi. Da bi saznala virtualnu MAC adresu, računala koriste ARP kao i prije.



Slika 131. Prikaz virtualnog usmjernika koji računala koriste kao svoj mrežni pristupnik (engl. Gateway)

Kako bi se osigurala redundancija mrežnog pristupnika u svim slučajevima, HSRP može odrediti koji usmjernik treba preuzeti aktivnu ulogu i u kojem slučaju bi tu ulogu trebao prepustiti nekom drugom usmjerniku. Prelazak s jednog aktivnog usmjernika na drugi potpuno je transparentno računalima u L2 mreži.

Na slici 132. prikazan je otkaz L2 veze kojom su dva usmjernika međusobno razmjenjivala HSRP *Hello* poruke da bi znali koji treba biti aktivni mrežni pristupnik. U takvom slučaju usmjernik R2 više neće primati *Hello* poruke od usmjernika R1 i nakon predefiniranog vremena preuzet će ulogu aktivnog mrežnog pristupnika. *Hello* poruke služe za komunikaciju svih usmjernika u istoj HSRP grupi i svi usmjernici moraju biti povezani na zajednički L2 segment mreže.



Slika 132. Primjer otkaza veze prema glavnom pristupniku

Uloge koje usmjernici mogu zauzeti u HSRP grupi su:

Virtualni usmjernik (engl. *Virtual Router*): predstavlja par virtualne IP i MAC adrese i za svaku HSRP grupu definira se jedan; takav virtualni usmjernik zapravo ne postoji u fizičkom obliku (nije uređaj).

Aktivni usmjernik (engl. *Active Router*): jedan je od uređaja u HSRP grupi i jedini je. Njegova je uloga prosljeđivati okvire koji se šalju na virtualnu MAC adresu virtualnog usmjernika. Dakle, pošalje li bilo koje računalo okvire na virtualnu MAC adresu, on će ih obraditi; pošalje li ARP zahtjev, koji sadrži IP adresu virtualnog usmjernika, on će na nj odgovoriti slanjem virtualne MAC adrese.

Zamjenski usmjernik (engl. *Standby Router*): osluškuje periodičke *Hello* poruke i prestane li ih primati, preuzima ulogu aktivnog usmjernika. Postoji samo jedan takav u HSRP grupi.

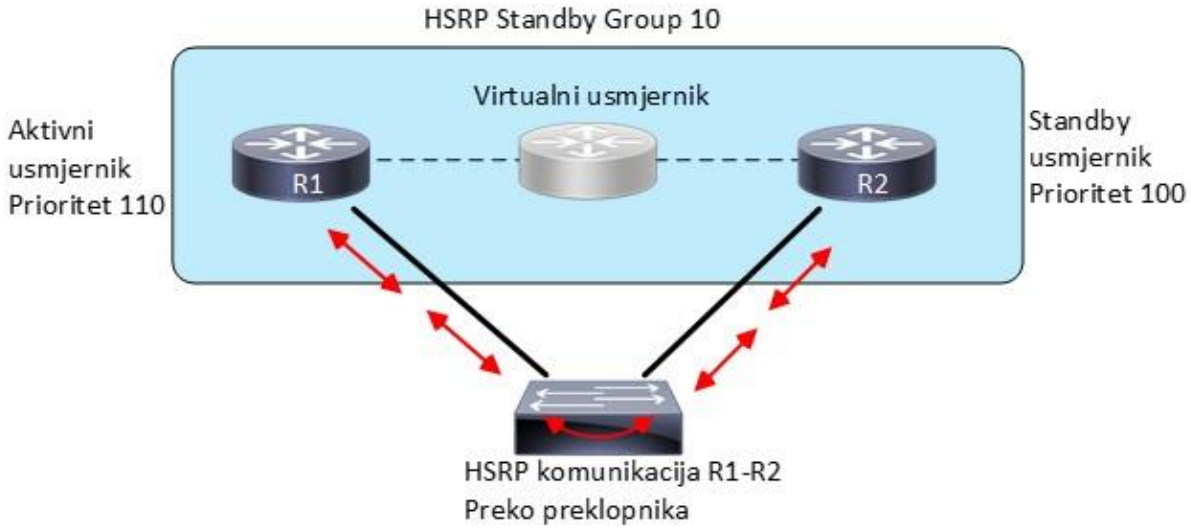
Ostali usmjernici (engl. *Other Routers*): u HSRP grupi može biti više od dva usmjernika, ali postoje samo jedan aktivni i jedan zamjenski. Svi su ostali usmjernici u početnom stanju (engl. *Initial State*) i tek nakon što otkažu i aktivni i zamjenski

usmjernik, mogu se „natjecati“ za njihove uloge. Ispod su prikazana stanja kroz koja prolazi HSRP usmjernik:

Početno stanje (engl. *Initial*)	Početno stanje kada HSRP još nije u funkciji; usmjernik se u tom stanju nalazi prilikom konfiguracije ili u trenutku uključivanja sučelja na kojem se konfigurira HSRP.
Stanje slušanja (engl. *Listen*)	Usmjernik zna svoju virtualnu IP adresu, ali nije ni aktivan ni zamjenski usmjernik, već samo sluša poruke koje oni šalju.
Stanje odabira (engl, *Speak*)	Usmjernik šalje *Hello* poruke i sudjeluje u odabiru aktivnoga i zamjenskog usmjernika; ne može bez konfigurirane virtualne IP adrese.
Stanje zamjenskog usmjernika (engl. *Standby*)	Usmjernik je kandidat za preuzimanje uloge aktivnog usmjernika i šalje *Hello* poruke; samo jedan usmjernik u HSRP grupi može biti zamjenski.
Stanje aktivnog usmjernika (engl. *Active*)	Usmjernik je zadužen za prosljeđivanje prometa koji dolazi na virtualnu MAC adresu; šalje *Hello* poruke. Samo jedan usmjernik u HSRP grupi može biti aktivan.

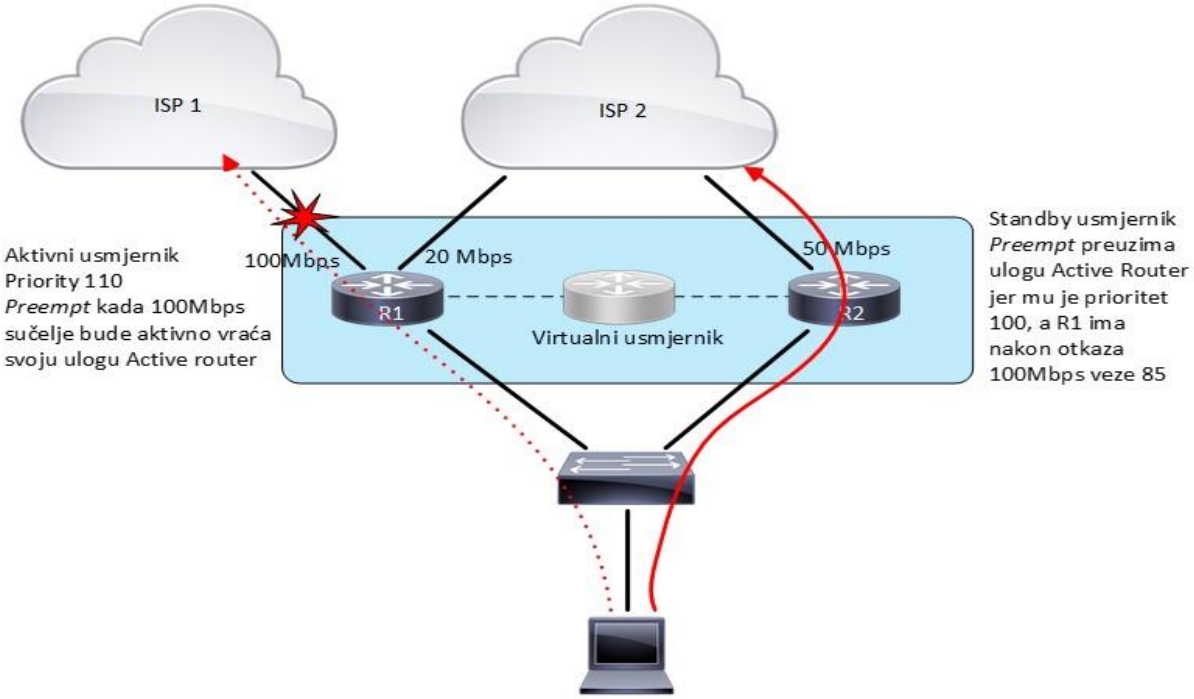
2.9.3. HSRP prioritet (engl. *Priority*), praćenje sučelja (engl. *Tracking*) i preuzimanje (engl. *Preempt*)

Utjecati na odabir aktivnoga i zamjenskog usmjernika može se konfiguriranjem prioriteta. Početna vrijednost prioriteta je 100, ako se ne promijeni, tada će onaj s najvećom IP adresom postati aktivni usmjernik. Prioritet možemo konfigurirati u rasponu od 0 do 255, a veća je vrijednost bolja.



Slika 133. Komunikacija HSRP usmjernika preko L2 segmenta

Želimo li da aktivni usmjernik prepusti svoju ulogu zamjenskome, tada se može konfigurirati praćenje bitnih sučelja (poput veza s internetom). U tom slučaju usmjernik će automatski prilagoditi vrijednost prioriteta u slučaju otkaza sučelja. Npr. aktivni usmjernik ima konfiguriran prioritet 110. Otkaže li mu jedna (bolja) od dvije internetske veze, aktivni usmjernik si smanjuje prioritet za 25, što čini 85, i to je sada manja vrijednost prioriteta od one koju ima zamjenski usmjernik (predefiniranu vrijednost 100). U tom slučaju zamjenski usmjernik, ako ima konfigurirano preuzimanje (engl. *Preempt*), postaje aktivni usmjernik.



Slika 134. Prikaz prepuštanja aktivne uloge u slučaju otkaza praćenog sučelja

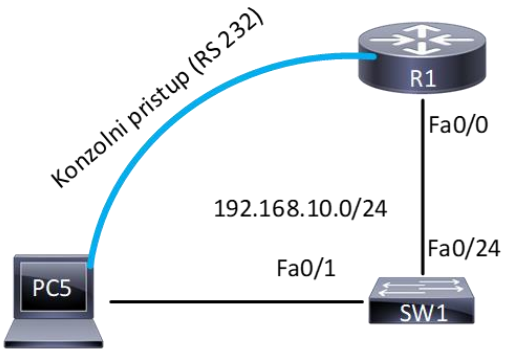
Usmjernik također može pratiti više sučelja i smanjivati si prioritet u više navrata dok ne dođe ispod granice kada zamjenski usmjernik preuzima aktivnu ulogu.

Naredba preuzimanja (engl. *Preempt*) važan je element konfiguracije jer pomoću nje zamjenski preuzima ulogu aktivnog usmjernika, ako si ovaj smanji prioritet zbog otkaza jednog od sučelja koje prati. Ta naredba također omogućava usmjerniku koji je bio aktivan do otkaza jednog od njegovih sučelja ponovno preuzimanje aktivne uloge kada to sučelje postane funkcionalno. U tom slučaju vraća si vrijednost prioriteta na onu koja je bila prije otkaza sučelja i u kombinaciji s naredbom preuzimanja (engl. *Preempt*) opet postaje aktivni usmjernik. Bez te naredbe to se neće dogoditi i korisnici neće imati prijašnji pristup resursima.

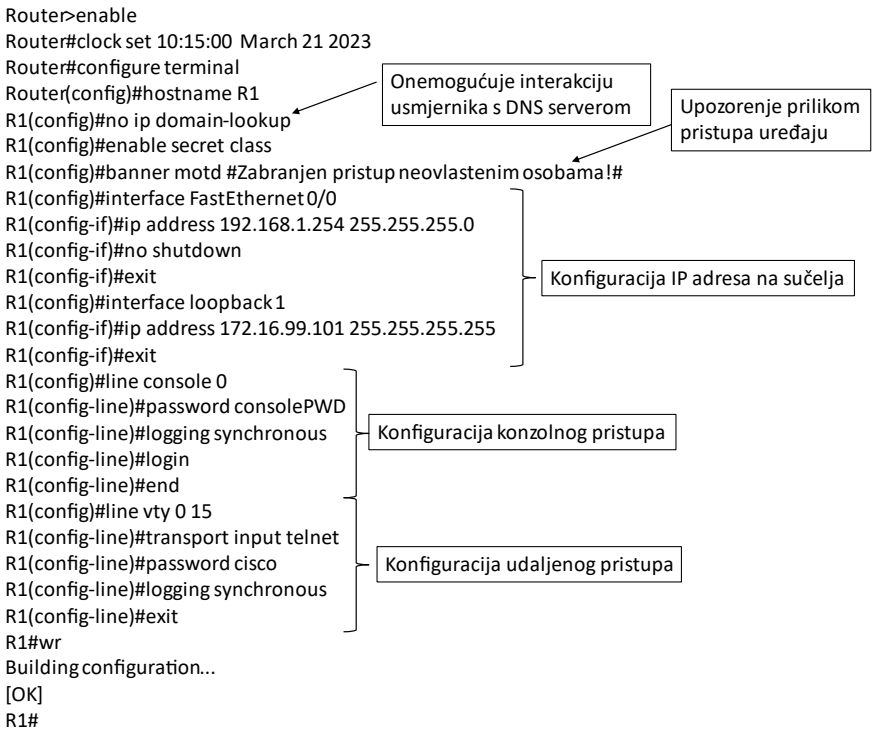
Također, aktivira li se neki usmjernik znatno brže od ostalih u HSRP grupi, on će postati aktivni usmjernik bez obzira na konfigurirani prioritet. U tom je slučaju potrebna naredba preuzimanja (engl. *Preempt*) tako da usmjernik koji zaista ima najveći prioritet, preuzme ulogu aktivnog usmjernika u HSRP grupi.

2.10. Usmjernik kao uređaj

Usmjernik je osnovni uređaj u računalnim mrežama, čija je uloga usmjeravanje mrežnog prometa (paketa) od izvorišta do odredišta. Obično je to dio koji krajnji korisnici ne vide, niti su svjesni da postoji, a ključan je za povezivanje udaljenih podmreža. Jedan je od način potvrde postojanja usmjernika naredbom *Tracert* koja daje ispis IP adrese usmjernika koji su na putu do odredišta. Gotovo sve što je opisano u poglavlju „Osnovna konfiguracija preklopnika“ o pristupu preklopniku i implementaciji osnovne konfiguracije za moguć udaljeni pristup, vrijedi i za usmjernik (engl. *Router*). Razlika je što se na usmjerniku neće konfiguirirati VLAN L3 sučelje, već će se IP adrese konfigurirati na fizička sučelja (npr. *FastEthernet 0/0*), podsučelja (npr. *FastEthernet 0/0.10*) ili na virtulano *Loopback* sučelje (npr. *Loopback 1*).

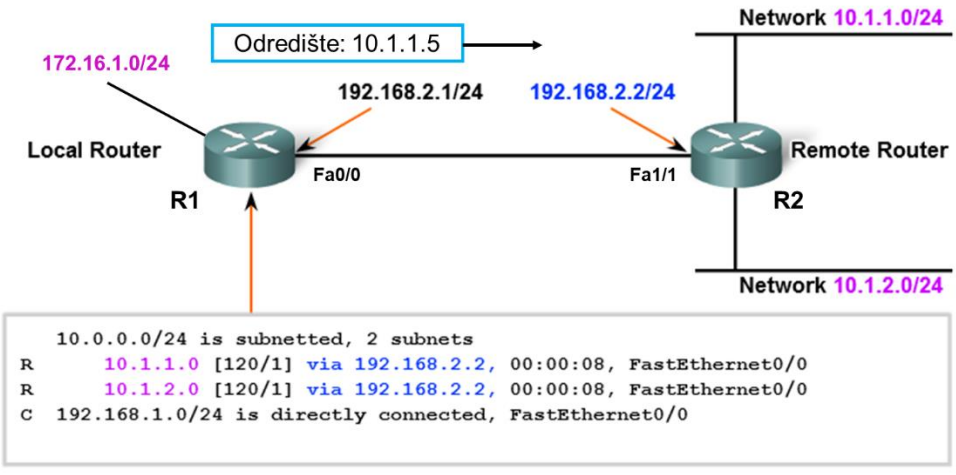


Slika 135. Povezivanje računala i usmjernika serijskim/konzolnim kabelom



Slika 136. Primjer osnovne konfiguracije usmjernika

Usmjernik je uređaj koji primarno radi na mrežnom sloju OSI modela, a to znači da za odluku o tome kako proslijediti pakete donosi na temelju odredišne IP adrese u zaglavlju IP paketa i svoje tablice usmjeravanja. Tablicu usmjeravanja kreira upisom informacija o direktno povezanim sučeljima, ručnim administratorovim upisom putanje ili razmjenom informacija o podmrežama pomoću protokola usmjeravanja koje je konfigurirao administrator.



Slika 137. Primjer uloge usmjernika u povezivanju udaljenih podmreža

Na slici 139. primjer je tablice usmjeravanja (naredba „show ip route“) jednog usmjernika koji ima više izvora informacije o podmrežama.

```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

Slika 138. Objašnjenja kratica za izvor informacije o mrežnoj putanji

```
     8.0.0.0/24 is subnetted, 1 subnets
C       8.8.8.0 is directly connected, Loopback0
     10.0.0.0/24 is subnetted, 5 subnets
S       10.1.10.0 [1/0] via 172.16.1.5
S       10.1.11.0 [1/0] via 172.16.1.5
C       10.1.20.0 is directly connected, FastEthernet0/0.20
C       10.1.21.0 is directly connected, FastEthernet0/0.21
C       10.1.99.0 is directly connected, FastEthernet0/0.99
     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.1.4/30 is directly connected, FastEthernet0/0.100
C       172.16.1.8/29 is directly connected, FastEthernet0/0.114
R    192.168.1.0/24 [120/1] via 172.16.1.9, 00:00:21, FastEthernet0/0.114
S    192.168.10.0/24 [1/0] via 10.1.10.100
S*   0.0.0.0/0 is directly connected, Loopback0
```

Slika 139. Primjer tablice usmjeravanja

Svi mogući izvori informacija o podmrežama vidljivi su u legendi prije same tablice usmjeravanja. Ovisno o tome kako je neka putanja upisana u tablicu usmjeravanja, imat će oznaku izvora. Administratori se time koriste za optimalno usmjeravanje u računalnoj mreži.

```
R1#show ip route

D    192.168.0.0/20[90/3196416] via 2.2.2.2, 00:03:52, Serial0/0/1
```

- Izvor informacije (eng. route source)
- Odredišna mreža (eng. destination network)
- Administrativna udaljenost/mjera kvalitete puta (eng. administrative distance/metric)
- Adresa slijedećeg usmjernik ana putu do odredišta (eng. next hop address)
- Izlazno sučelje lokalnog usmjernika (eng. exit interface)

Slika 140. Objašnjenje elemenata specifične putanje

Osim osnovne zadaće usmjeravanja IP paketa računalnom mrežom do odredišta, usmjernici pružaju i razne druge mogućnosti poput kvalitete usluge u mreži (engl. *Quality of Service – QoS*), filtracije mrežnog prometa na temelju IP adresa ili TCP/UDP ulaza i slično.

Glavni je zadatak usmjernika u mreži usmjeravanje, odnosno određivanje puta do odredišta (engl. *Routing*, *Path Determination*) i prosljeđivanje paketa (engl. *Packet Forwarding*). Razlika je između te dvije funkcije što su u usmjeravanju u mreži svi usmjernici međusobno ovisni kako bi komunikacija uopće bila moguća. Svaki usmjernik treba proslijediti pakete prema drugom usmjerniku tako da u konačnici ti paketi dođu do odredišta. S druge strane, prosljeđivanje je proces prijenosa paketa od ulaznog do izlaznog sučelja (engl. *Interface*) na lokalnom usmjerniku, i to na temelju tablice usmjeravanja i nekih drugih mehanizama u samom uređaju poput QoS oznaka, što nije obuhvaćeno priručnikom. Svi usmjernici na putu od ishodišta do odredišta ponavljaju isti postupak dekapsulacije na ulazu, traženja izlaznog sučelja u tablici usmjeravanja i ponovne enkapsulacije paketa na izlazu. U koji će format okvira biti enkapsuliran paket, ovisi o vrsti izlaznog sučelja.

2.11. Načela usmjeravanja i korištenje putanja u tablici usmjeravanja

Nakon što je poznata uloga usmjernika, uloga tablice usmjeravanja i njezina ažuriranja, nameće se pitanje kako usmjernik odlučuje koje putanje će zaista koristiti za prosljeđivanje prometa prema odredištu. Treba krenuti od načela usmjeravanja za razumijevanje kada će usmjernik uopće donijeti odluku da neki paket pošalje prema odredištu. Tri su osnovna načela da bi usmjeravanje prometa u mreži bilo uspješno:

- **Načelo 1.**: svaki usmjernik donosi odluku o usmjeravanju paketa neovisno, na temelju informacija koje ima u svojoj tablici usmjeravanja. Pri donošenju odluke o preusmjeravanju paketa, usmjernik ne konzultira tablice usmjeravanja ostalih usmjernika;
- **Načelo 2.**: činjenica da jedan usmjernik ima informaciju o nekoj mreži u svojoj tablici ne znači da tu istu informaciju imaju i ostali usmjernici; usmjernik ne zna putanje u tablicama ostalih usmjernika;

- **Načelo 3.**: informacija o nekom putu od jedne mreže do druge ne znači da postoji i informacija o obrnutom putu. Većina je razmjena podataka između krajnjih uređaja u mreži dvosmjerna. To znači da paketi putuju u oba smjera. Dakle, nije dovoljno imati putanje od ishodišta do odredišta, već i obrnuto. To treba stalno imati na umu pri konfiguriranju mreže.

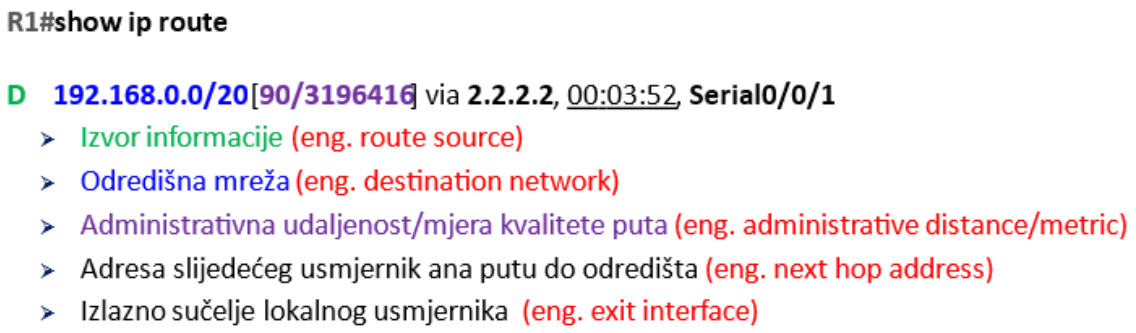
To su načela svih usmjernika u bilo kojem dijelu mreže, a nadalje je pitanje kriterija prema kojima usmjernici utvrđuju koju će od dostupnih putanja zaista koristiti u tablici usmjeravanja. Kada ima više putanja koje može koristiti za prosljeđivanje paketa prema odredišnoj mreži, usmjernik te putanje uspoređuje po dolje navedenim kriterijima, i onu koja se pokaže bolja, koristit će u tablici usmjeravanja.

1. **Dostupnost *Next-hop* adrese**: ako u nekoj putanji nije dostupna IP adresa sljedećeg usmjernika na putu do odredišta (engl. *Next-hop address*), ta se putanja uopće ne uzima u obzir;

2. **Administrativna udaljenost** (engl. *Administrative distance*): ako usmjernik ima identičnu putanju koju je dobio od dva različita izvora (npr. statički konfigurirana putanja i putanja koju je naučio kroz neki protokol usmjeravanja) tad će koristiti administrativnu udaljenost za odabir najbolje putanje. Najbolja je ona s najmanjom vrijednošću administrativne udaljenosti;

3. **Mjera kvalitete putanje** (engl. *Metric*): ako usmjernik dobije dvije putanje od istog izvora i administrativna udaljenost je ista, tada koristi mjeru kvalitete puta i odabire onu putanju koja ju ima manju;

4. **Duljina prefiksa** (engl. *Prefix Length*): za prosljeđivanje prometa usmjernik uvijek koristi putanju u tablici usmjeravanja koja ima najviše istih bitova kao i odredišna IP adresa, bez obzira na izvor putanje (statički konfigurirana ili posredstvom nekog usmjerničkog protokola). Moguće je da u tablici usmjeravanja ima više putanja koje vode prema odredištu, no moraju imati različite mrežne maske da bi bile u tablici usmjeravanja, i samim time ih usmjernik gleda kao različite.



Slika 141. Elementi jedne putanje u tablici usmjeravanja

Dakle, u tablici usmjeravanja nalaze se samo najbolje putanje prema određenoj podmreži, a unatoč tome što su naizgled iste, no imaju različitu mrežnu masku, usmjernik ih smatra različitima. Primjer na slici 142. pokazuje putanje koje usmjernik smatra različitima i sve će tri biti u tablici usmjeravanja, ali zbog preklapanja, dogodit će se neoptimalno i štetno usmjeravanje prometa. Ako bi promet za IP adresu 192.168.1.250 došao na usmjernik koji ima te putanje u tablici usmjeravanja, promet bi bio proslijeđen prema 10.1.2.2, međutim ako bi promet za IP adresu 192.168.1.10 došao na isti usmjernik, taj promet će biti proslijeđen prema 10.1.4.2. Ovo će vrijediti iako postoji samo jedna mreža 192.168.1.0/24 u sustavu. Zato treba paziti da se ne dogode takva preklapanja u usmjerničkim tablicama, gdje je jedna podmreža dio neke druge podmreže. Preduvjet da se to ne dogodi ispravno je definirana IPv4 adresna shema.

```
ip route 192.168.1.0 255.255.255.0 10.1.2.2
ip route 192.168.1.0 255.255.255.128  10.1.3.2
ip route 192.168.1.0 255.255.255.192  10.1.4.2
```

Slika 142. Primjer loše konfiguriranih statičkih putanja

Ako usmjernik ne pronađe preciznu putanju, nadalje će koristiti općenitije, poznate i kao sažete putanje. Ne može li ni tada proslijediti promet, koristit će najopćenitiju moguću, poznatu i kao predefinirana putanja 0.0.0.0/0.

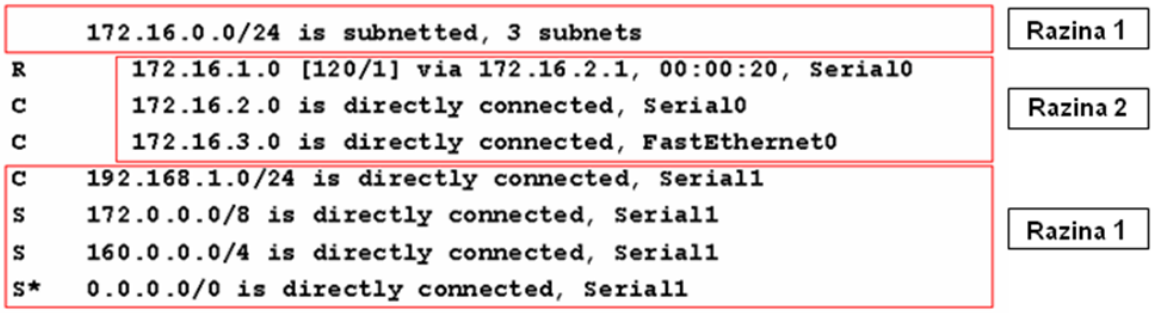
2.12. Vrste putanja u tablici usmjeravanja

Razumijevanje ovog dijela važno je za usvajanje terminologije tablice usmjeravanja na nekom usmjerniku. Tablica usmjeravanja ima dvije razine putanja, kao što je prikazano na slici 143. Razina 2 je podrazina razine 1. Da bi usmjernik neku putanju smatrao razinom 1 ili 2, trebaju biti zadovoljeni određeni uvjeti.

Treba se prisjetiti da postoji pet klasa IP adresa (od kojih koristimo tri) određenih kombinacijom bitova u prvom bajtu.

- Klasa A je u rasponu od 0 do 127 u prvom bajtu
- Klasa B je u rasponu od 128 do 191 u prvom bajtu
- Klasa C je od 192 do 223 u prvom bajtu.

U kontekstu usmjeravanja prometa i korištenja tablice usmjeravanja, mrežne maske klasa su takve da A klasa ima mrežnu masku /8, B klasa ima mrežnu masku /16 i C klasa ima mrežnu masku /24.



Slika 143. Podjela putanja u tablici usmjeravanja

Putanje razine 1 (engl. *Level 1 Route*) – putanje čija je mrežna maska (engl. *Subnet Mask*) jednaka ili kraća od mrežne maske klase u kojoj se ta mreža nalazi.

Putanje razine 1 mogu biti:

- predefinirana putanja (engl. *Default Route*),
- sažeta putanja (engl. *Supernet Route ili Summary Route*)
- precizna mrežna putanja (engl. *Network Route*).

Predefinirana putanja dogovorom je određeno da bude 0.0.0.0/0 i s obzirom na to da je mrežna maska /0, što je manje od /8, jasno je da će ovakva putanja biti na razini 1.

Primjer je sažete putanje: 192.168.0.0/20, koja će biti na razini 1 jer je mrežna maska /20 kraća od /24. Primjer precizne mrežne putanje bila bi putanja 172.16.0.0/16, koja ima mrežnu masku jednaku onoj za klasu u kojoj se adresa mreže nalazi.

Putanje razine 2 (engl. *Level 2 Route*) – putanje čija je mrežna maska (engl. *Subnet Mask*) dulja od mrežne maske klase u kojoj se mreža nalazi, odnosno putanje koje su podmreže neke od mreža klase A, B ili C. Primjer putanja na razini 2 na slici je 143. Odredišna mreža 172.16.1.0 je u podmreži (/24) i u mreži klase B (/16). Dakle, udovoljeno je uvjetu da ta putanja bude na razini 2 jer se izvodi iz mreže klase B 172.16.0.0 /16.

Konačne putanje (engl. *Ultimate Routes*) – putanje koje, uz adresu odredišne mreže, sadrže još i oznaku izlaznog sučelja (engl. *Exit Interface*) ili adresu susjeda na drugom kraju veze (engl. *Next Hop*). Konačne putanje mogu biti na razini 1 ili na razini 2. Na primjeru na slici 144. vidi se da su sve putanje, osim prve, konačne putanje. Samo prva putanja nema izlazno sučelje ili adresu susjeda.

```
     172.16.0.0/24 is subnetted, 3 subnets
R       172.16.1.0 [120/1] via 172.16.2.1, 00:00:20, Serial0
C       172.16.2.0 is directly connected, Serial0
C       172.16.3.0 is directly connected, FastEthernet0
C    192.168.1.0/24 is directly connected, Serial1
S    172.0.0.0/8 is directly connected, Serial1
S    160.0.0.0/4 is directly connected, Serial1
S*   0.0.0.0/0 is directly connected, Serial1
```

Slika 144. Prikaz konačnih putanja

Vršne i slijedne putanje (engl. *Parent and Child Routes*) – unatoč tome što su konačne putanje, moguće je da putanja razine 1 nije konačna putanja, odnosno da nema definirano izlazno sučelje. Takve se putanje zovu vršne putanje (engl. *Parent Routes*). Vršne putanje prikazuju samo mrežni dio klase i prefiks koji označava koliko je bitova mrežni dio. Kreiraju se automatski kada putanja ima masku podmreže veću od maske klase. Vršne su putanje razina 1 za slijedne putanje (engl. *Child Routes*) koje su na razini 2. Ako putanje na razini 2 nisu varijabilno subnetirane (VLSM), što znači da sve imaju istu mrežnu masku, vršne će putanje sadržavati jednu masku koju sve slijedne putanje dijele. Na slici 145. vidi se vršna putanja 172.16.0.0 s mrežnom maskom /24. Odredišne mreže na razini 2 su 172.16.1.0,

172.16.2.0 i 172.16.3.0 s maskom zapisanom u vršnoj putanji. U ovom su slučaju iz mreže klase B 172.16.0.0 napravljene tri podmreže s maskom podmreže /24.

```
     172.16.0.0/24 is subnetted, 3 subnets
R       172.16.1.0 [120/1] via 172.16.2.1, 00:00:20, Serial0
C       172.16.2.0 is directly connected, Serial0
C       172.16.3.0 is directly connected, FastEthernet0
C    192.168.1.0/24 is directly connected, Serial1
S    172.0.0.0/8 is directly connected, Serial1
S    160.0.0.0/4 is directly connected, Serial1
S*   0.0.0.0/0 is directly connected, Serial1
```

Slika 145 Primjer vršnih i slijednih putanja

Međutim, ako slijedne putanje koriste različite mrežne maske, dakle korišten je VLSM koncept, tada će vršna putanja koristiti prefiks klase (npr. A, B ili C), kao što je prikazano na slici 146.

```
     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
C       172.16.1.4/30 is directly connected, Serial0
C       172.16.1.8/30 is directly connected, Serial1
C       172.16.3.0/24 is directly connected, FastEthernet0
```

Slika 146. Prikaz varijabilno subnetiranih slijednih putanja

2.12.1. Proces pretraživanja tablice usmjeravanja

Ovdje će biti opisan proces koji usmjernici koriste u odluci kamo proslijediti određeni IP paket. Kada usmjernik primi paket na jedno od svojih sučelja, pročitat će odredišnu IP adresu u zaglavlju IP paketa i započinje proces pretraživanja tablice usmjeravanja. Usmjernik uspoređuje binarni zapis odredišne IP adrese u zaglavlju IP paketa i zapis u svojoj tablici usmjeravanja u binarnom obliku. Traži onaj zapis koji se binarno najviše podudara s odredišnom IP adresom IP paketa. To binarno uspoređivanje usmjernik radi slijeva nadesno do bita koji je različit. Iz toga se može zaključiti da će usmjernik najprije tražiti precizne mrežne putanje, ako njih nema, tada će tražiti sažete putanje, a ako ni njih nema, tada će koristiti predefiniranu putanju. Ako nema ni predefinirane putanje, usmjernik će odbaciti IP paket.

Proces pretraživanja tablice može se opisati u nekoliko koraka:

- **korak 1.** – usmjernik počinje pretraživanje tražeći putanje na razini 1 (mrežnu putanju, konačnu putanju ili vršnu putanju koja se slaže s odredišnom IP adresom u paketu);
- **korak 2.** – ako se odredišna IP adresa slaže s konačnom putanjom (engl. *Ultimate Route*) – a ona nema podmreže, ta će se putanja koristiti za prosljeđivanje paketa;
- **korak 3.** – ako je putanja koja se slaže s odredišnom IP adresom vršna putanja (engl. *Parent Route*), pretraživanje se nastavlja na razini 2 unutar podmreža;
- **korak 4.** – ako se ne nađe odgovarajuća putanja u koraku 2., a nađe se u koraku 3., nastavak pretraživanja ovisit će o tome koristi li usmjernik klasificirano (engl. *Classful*) ili neklasificirano (engl. *Classless*) pretraživanje. O tome će biti riječi malo dalje.

Kako usmjernik zna da je našao odgovarajuću putanju? Proces je pretraživanja uspoređivanje bitova u odredišnoj IP adresi s bitovima u putanjama u tablici usmjeravanja slijeva nadesno. Mrežna maska odredišne mreže u tablici usmjeravanja određuje koliko se najmanje bitova treba poklapati da bi putanja odgovarala. Ako se poklapaju dvije ili više putanja, odabire se putanja u kojoj se više bitova poklapa slijeva nadesno (engl. *Longest Match*). Bitno je znati da u samom zaglavlju IP paketa nema mrežne maske, već samo IP adrese izvorišta i odredišta paketa.

Slijedi nekoliko primjera tablice usmjeravanja na slici 147.

```
     172.16.0.0/24 is subnetted, 3 subnets
R       172.16.1.0 [120/1] via 172.16.2.1, 00:00:20, Serial0
C       172.16.2.0 is directly connected, Serial0
C       172.16.3.0 is directly connected, FastEthernet0
C    192.168.1.0/24 is directly connected, Serial1
S    172.0.0.0/8 is directly connected, Serial1
S    160.0.0.0/4 is directly connected, Serial1
S*   0.0.0.0/0 is directly connected, Serial1
```

Slika 147. Primjer tablice usmjeravanja

Primjer 1: Pretpostavka je da usmjernik treba proslijediti paket s odredišnom IP adresom 192.168.1.15. Usmjernik tablicu pretražuje od prvog do zadnjeg zapisa

redom. Prvi zapis u tablici usmjeravanja na slici je vršna putanja (engl. *Parent Route*) 172.16.0.0. Usmjernik uspoređuje prvih 16 bitova u putanji 172.16.0.0/24 u tablici usmjeravanja s prvih 16 bitova u odredišnoj IP adresi 192.168.1.15. Uspoređuje prvih 16 bitova zato što se kod vršnih putanja podrazumijeva mrežna maska klase, a to je u ovom slučaju B klasa, koja ima mrežnu masku 16. Budući da se prvih 16 bitova u putanji 172.16.0.0/24 i 192.168.1.15 ne poklapa, usmjernik ne ulazi u pretraživanje razine 2, odnosno podmreža 172.16.1.0, 172.16.2.0 i 172.16.3.0 s prefiksom 24, već nastavlja na sljedeću putanju razine 1. To je adresa 192.168.1.0 /24. Sada usmjernik uspoređuje prva 24 bita putanje 192.168.1.0/24 u tablici usmjeravanja i prva 24 bita odredišne IP adrese 192.168.1.15. Zaključak je da su prva 24 bita slijeva nadesno u adresama 192.168.1.15 i 192.168.1.0 ista, tako da će usmjernik koristiti upravo tu putanju za prosljeđivanje paketa. Kada putanja 192.168.1.0/24 ne bi bila konačna putanja, već vršna putanja, tada bi usmjernik pretraživao putanje razine 2 i tražio najbolje preklapanje bitova.

Primjer 2: Pretpostavka je da usmjernik treba proslijediti paket s odredišnom IP adresom 172.16.2.1. Usmjernik uspoređuje prvih 16 bitova u putanji 172.16.0.0 u tablici usmjeravanja s prvih 16 bitova u odredišnoj IP adresi. Prvih 16 bitova adrese u putanji 172.16.0.0 slaže se s prvih 16 bitova odredišne IP adrese 172.16.2.1. Ako se vršna putanja slaže s IP adresom odredišta, ulazi se u pretraživanje razine 2, odnosno podmreža mreže 172.16.0.0 u klasi B. Prefiks u vršnoj putanji na razini 1 određuje koliko se bitova treba poklopiti slijeva nadesno na razini 2. U ovom slučaju to su 24 bita. Uspoređujući prvu putanju druge razine s IP adresom odredišta, usmjernik zaključuje da nema preklapanja i pretraživanje se nastavlja. Provjerava se putanja 172.16.2.0 i ona se slaže sa 172.16.2.1 u prva 24 bita, te se ta putanja koristi za prosljeđivanje paketa.

Primjer 3.: Pretpostavka je da usmjernik treba proslijediti paket s odredišnom IP adresom 34.24.11.112. Sada se već na prvi pogled u tablici usmjeravanja vidi da se IP adresa neće složiti ni s jednom mrežom na razini 1. U takvim slučajevima usmjernici koriste predefiniranu putanju 0.0.0.0 0.0.0.0 ali tek ako nema ni jedne bolje u tablici usmjeravanja, dakle zadnju će pregledati. Paket se preusmjerava na izlazno sučelje

povezano s predefiniranom putanjom. Ako ne postoji predefinirana putanja, tada se paket odbacuje, odnosno uništava.

Postavlja se važno pitanje: što ako se uđe u pretraživanje na razinu 2, a nijedna se odredišna mreža na toj razini ne slaže s odredišnom IP adresom? Sljedeći primjer to pokazuje.

Primjer 4.: Pretpostavka je da je odredišna IP adresa 172.16.23.11.

- 16 bitova od 172.16.0.0 slaže se sa 172.16.23.11 i ulazi se u pretraživanje putanja na razini 2 (engl. *Child Route*);
- 24 bita od 172.16.1.0 se ne poklapaju, nastavlja se pretraživanje;
- 24 bita od 172.16.2.0 se ne poklapaju, nastavlja se pretraživanje;
- 24 bita od 172.16.3.0 se ne poklapaju, što dalje?

Ako je riječ o klasificiranom (engl. *Classful*) pretraživanju tablice usmjeravanja, sažete putanje i predefinirane putanje neće se provjeravati (ako postoje) i paket će biti odbačen. Ako je riječ o neklasificiranom pretraživanju (engl. *Classless*), pretraživanje će se nastaviti uključujući sažete i standardne putanje. Važno je razlučiti da klasificirano i neklasificirano pretraživanje tablice usmjeravanja nema nikakve veze s klasificiranim i neklasificiranim protokolima usmjeravanja. Protokoli usmjeravanja brinu se kako razmijeniti informacije o postojećim podmrežama u IKT sustavu između usmjernika, a ne kako prosljeđivati pakete pomoću tablice usmjeravanja. Klasificirano i neklasificirano se odnosi na način na koji usmjernik čita tablicu usmjeravanja. Danas je uobičajeno da svi usmjernici rade na neklasificiran način, što uvijek omogućava korištenje sažetih putanja i predefinirane putanje. Ako neki usmjernik ipak radi na klasificirani način, treba ga konfigurirati da radi na neklasificiran, kako ne bi bilo neoptimalno usmjeravanje u mreži. Ako usmjernik ne podržava neklasificiran način rada, onda ga treba zamijeniti nekim koji podržava.

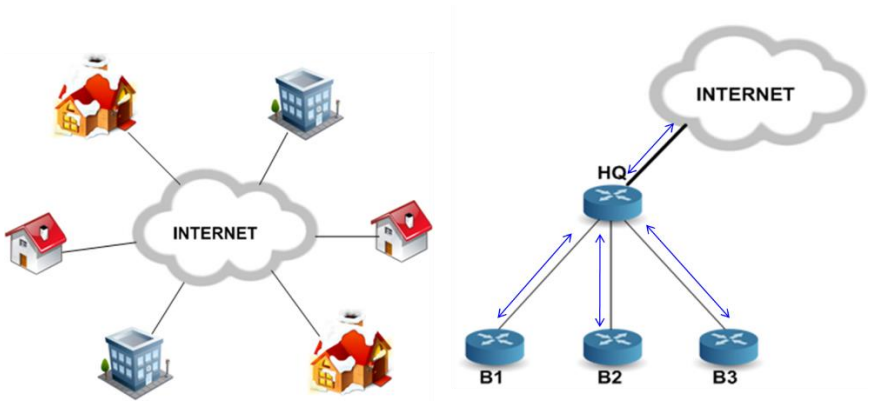
2.13. Ažuriranje tablice usmjeravanja

Ranije je spomenuto da su informacije u tablici usmjeravanja ključne kako bi usmjernik mogao ispravno proslijediti pakete prema odredišnoj podmreži. Zato je važno da svaki

usmjernik ima aktualne i točne informacije u tablici usmjeravanja. Dva su osnovna načina ažuriranja tablice usmjeravanja na usmjernicima, a to su:

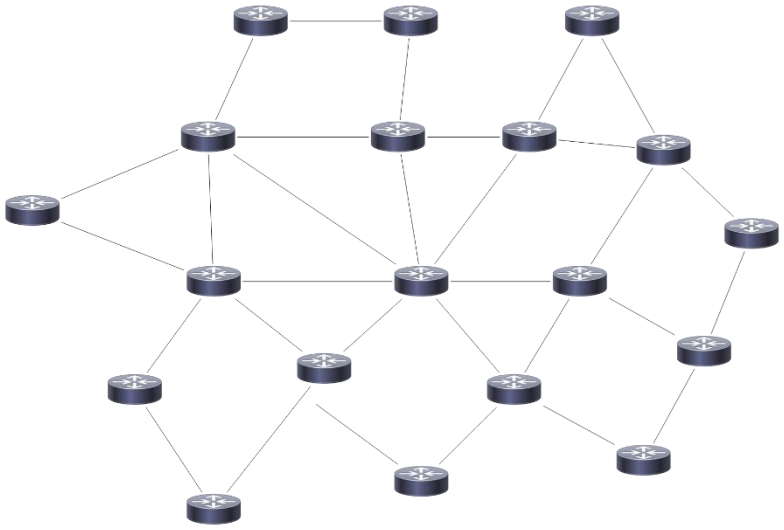
- **Statičko ažuriranje** – podrazumijeva da administrator mreže ručno konfigurira dio ili sve statičke putanje u tablici usmjeravanja usmjernika
- **Dinamičko ažuriranje** – podrazumijeva da usmjernici koriste jedan od protokola usmjeravanja kako bi između sebe razmijenili informacije o postojećim mrežama.

U praksi se koristi i jedan i drugi način ovisno o konkretnoj situaciji i mogućnostima uređaja u mreži. Bitno je uvijek paziti da u konačnici svi uređaji u mreži imaju tablice usmjeravanja koje će omogućiti optimalno usmjeravanje. Nije moguće reći kako doći do tog stanja, ali postoji nekoliko principa koje svakako treba poštivati kada se definira kako će usmjeravanje u nekoj mreži izgledati. Prvi je, i najvažniji, ispravna definicija adresne sheme koja odgovara logičkoj segmentaciji računalne mreže, a to u praksi znači da nema mreža koje se preklapaju i da adresna shema odgovara geografskoj distribuciji lokacija organizacije. To je neophodno da bi se mogle koristiti sažete putanje i predefinirane putanje koje će omogućiti manje usmjerničke tablice, a samim time i efikasnije usmjeravanje prometa u mreži. Također je važno treba li mreža biti skalabilna i lako upravljiva, što uključuje protokol usmjeravanja, a ne ručno konfiguriranje putanja, koje je podložno greškama, te bi dodatkom podmreže trebalo napraviti puno izmjena na svim usmjernicima u mreži.



Slika 148. Prikaz tzv. Hub and Spoke topologija gdje se koristi statičko ažuriranje usmjerničkih tablica

Na slici 149. prikazan je ekstremni primjer mreže s više usmjernika u kojoj treba koristiti dinamičko ažuriranje usmjerničkih tablica. Statičko ažuriranje bi u ovom slučaju bilo previše teško za održavanje zbog grešaka u konfiguraciji.

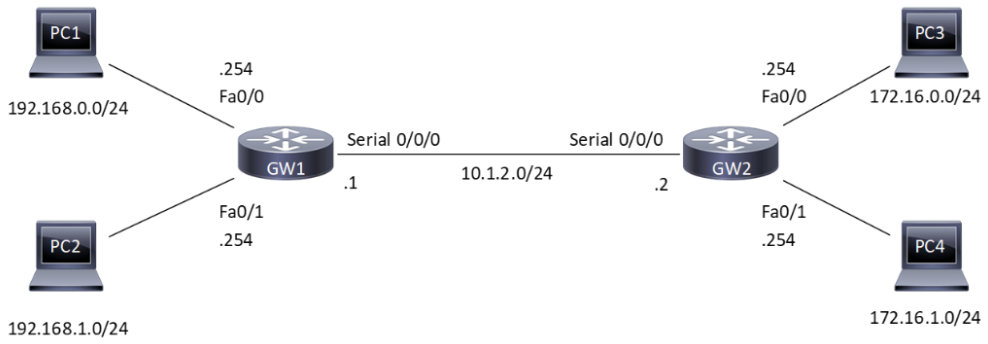


Slika 149. Primjer mreže u kojoj treba koristiti dinamičko ažuriranje usmjerničkih tablica

U specifičnim situacijama koriste se statičke putanje i u prikazanom okruženju, ali nisu u opsegu ovog priručnika.

2.14. Statičko ažuriranje tablice usmjeravanja

Iako se u praksi češće sreće termin „statičko usmjeravanje“ nego „statičko ažuriranje tablice usmjeravanja“, ispravno je potonje. Dakle, radi se o unosu zapisa administratora računalne mreže u tablicu usmjeravanja. Takav način ažuriranja tablice usmjeravanja puno je sporiji od ažuriranja pomoću dinamičkih protokola usmjeravanja, i podložan je greškama. Čak ako administrator uspije sve zapise unijeti bez greške, mreže u kojima se usmjeravanje prometa temelji na statičkom ažuriranju usmjerničkih tablica nisu skalabilne, jer svaka nova podmreža u sustavu uglavnom znači da treba velik dio ili čak sve postojeće usmjernike ažurirati s novim zapisima u usmjerničkim tablicama. Naravno, to ovisi o samoj mrežnoj topologiji. U slučaju *Hub-and-Spoke* topologije to nužno ne vrijedi ako se na svaku od lokacija (engl. *Spoke*) konfigurira jedna predefinirana putanja tako da se više nikad ne mora ništa unositi u tablice usmjeravanja. Međutim, u praksi su ovakve mreže rijetke, pogotovo u kompleksnim i dinamičnim IKT sustavima gdje se podmreže mijenjaju, i gdje ima puno korisnika, kao što su podatkovni centri ili okruženje telekomunikacijskih operatora.



Slika 150. Prikaz male mreže sa statičkim putanjama

Na topologiji prikazanoj na slici 150. vidljiva je veza između usmjernika GW1 i GW2. Kada je samo jedna mogućnost za prosljeđivanje prometa između odredišnih mreža, to je situacija u kojoj se može koristiti statičke putanje bez straha da će usmjeravanje paketa biti neoptimalno ili da mreža neće biti skalabilna. Usmjeravanje između lijeve i desne strane topologije može se postići na nekoliko načina koristeći statičke putanje:

1. Način – konfiguriraju se precizne statičke putanje na svakom od usmjernika za sve mreže koje taj usmjernik nema u tablici usmjeravanja. Usmjernik će predefinirano u tablici usmjeravanja imati samo direktno povezane mreže;
2. Način – konfiguriraju se sažete putanje (engl. *Supernet*) kako bi usmjernici imali manje tablice usmjeravanja, što znači efikasnije pretraživanje tih tablica;
3. Način – korištenjem predefinirane putanje tako da i GW1 i GW2 koriste samo predefiniranu putanju za slanje paketa prema odredištu.

Prvi način je vrlo jasan i precizan, dok u korištenju sažetih putanja i predefiniranih putanja treba oprez, da ne bi zbog nepravilne primjene takvih putanja, bilo neoptimalno usmjeravanje ili čak usmjernička petlja, što je vrlo opasno za računalnu mrežu. Na slici 151. prikazan je način konfiguracije preciznih statičkih putanja na usmjernicima GW1 i GW2.

```
GW1>enable
GW1#configure terminal
GW1(config)#ip route 172.16.0.0 255.255.255.0 10.1.2.2
GW1(config)#ip route 172.16.1.0 255.255.255.0 10.1.2.2
GW1(config)#exit
GW1#
```

```
GW2>enable
GW2#configure terminal
GW2(config)#ip route 192.168.0.0 255.255.255.0 10.1.2.1
GW2(config)#ip route 192.168.1.0 255.255.255.0 10.1.2.1
GW2(config)#exit
GW2#
```

Slika 151. Konfiguracija preciznih statičkih putanja na GW1 i GW2 (v. sliku 150.)

```
GW1#show ip route static
     172.16.0.0/24 is subnetted, 2 subnets
S       172.16.0.0 [1/0] via 10.1.2.2
S       172.16.1.0 [1/0] via 10.1.2.2
```

Slika 152. Ispis statičkih putanja na usmjerniku GW1

```
GW2#show ip route static
S    192.168.0.0/24 [1/0] via 10.1.2.1
S    192.168.1.0/24 [1/0] via 10.1.2.1
```

Slika 153. Ispis statičkih putanja na usmjerniku GW2

Moguće je koristiti i sažete putanje za usmjeravanje u prikazanoj mreži jer je adresna shema tako definirana i podmreže prikladno raspoređene. Kada bi podmreže računala PC1 i PC3 zamijenile mjesta, ne bi bilo moguće koristiti sažete putanje jer se nema što sažeti ni na jednoj strani. Kada bi se koristila predefinirana putanja na oba usmjernika, komunikacija između svih računala bi uredno radila, međutim dogodila bi se petlja za sve pakete kojima je odredište IP adresa koja nije dio ni jedne postojeće podmreže na topologiji. Na slici 154. prikazana je usmjernička petlja gdje paket kojem je odredište nepostojeća IP adresa 10.10.10.10 zaglavi u usmjerničkoj petlji između usmjernika GW1 i GW2. Razlog leži u tome što odredišna mreža ne postoji, a oba usmjernika imaju predefiniranu putanju koju koriste za slanje prometa jedan drugome. To će raditi očekivano u svim mrežama koje postoje iza ovih usmjernika, ali čim pakete treba slati

prema mreži koja ne postoji, nastaje petlja. Takvog prometa u mrežama uvijek ima jer postoje razne aplikacije koje pokušavaju kontaktirati poslužitelje koji više ne postoje ili je takav promet generirao maliciozni softver, i slično. Najvažnije je, dakle, usmjeravanje u mreži osmisliti i implementirati tako da se isključi mogućnost nastanka petlji.

```
C:\>tracert 10.10.10.10

Tracing route to 10.10.10.10 over a maximum of 30 hops:

  1    0 ms      0 ms      0 ms      172.16.0.254
  2    1 ms      1 ms      1 ms      10.1.2.1
  3    0 ms      1 ms      3 ms      172.16.0.254
  4    1 ms     12 ms     10 ms      10.1.2.1
  5    0 ms      1 ms      4 ms      172.16.0.254
  6    2 ms     23 ms      0 ms      10.1.2.1
```

Slika 154. Primjer usmjerničke petlje. ping je poslan s računala PC3 (v. sliku 113.)

Ako se u mreži koriste samo statičke putanje, a mreža se sastoji od više usmjernika između izvorišne i odredišne podmreže, svi usmjernici moraju imati putanje u svojoj tablici usmjeravanja za svaku podmrežu koja nije direktno povezana na usmjernik (treba se sjetiti načela usmjeravanja). Ako samo jedan usmjernik nema jednu od potrebnih putanja, udaljene podmreže neće moći komunicirati.

2.15. Dinamičko ažuriranje tablice usmjeravanja

Za razliku od statičkog ažuriranja tablice usmjeravanja, u ovom slučaju koriste se protokoli usmjeravanja koji automatski razmjenjuju informacije o postojećim podmrežama kako bi, u konačnici, svi usmjernici znali kako doći do bilo koje podmreže u sustavu. U najkraćim crtama, svaki usmjernik koristeći određeni protokol usmjeravanja šalje informaciju o tome koje mreže su na njega direktno povezane i koliko su „daleko“ od njega, a kako ta informacija putuje kroz mrežu, svaki usmjernik koji primi tu informaciju dodaje svoju vrijednost o „udaljenosti“ do te pojedine podmreže. Kada svi usmjernici saznaju za neku podmrežu, imat će podatak o ukupnoj „udaljenosti“ do te mreže iz vlastite perspektive i znat će preko kojeg je susjednog usmjernika „najkraći“ put do te podmreže.

Tri su ključne uloge protokola usmjeravanja u mrežama:

1. Razmjenjuju informacije o podmrežama
2. Koriste algoritam za određivanje najbolje putanje
3. Detektiraju i reagiraju na promjene u mrežnoj topologiji.

Neki od uobičajenih protokola usmjeravanja koji se danas koriste u mrežama su RIP (engl. *Routing Information Protocol*) , EIGRP (engl. *Enhanced Interior Gateway Routing Protocol*), OSPF (engl. *Open Shortest Path First*) i BGP (engl. *Border Gateway Protocol*)

Svaki protokol usmjeravanja koristi određeni format poruke kako bi mogao razmjenjivati informacije o podmrežama u sustavu s drugim usmjernicima. Također svaki protokol usmjeravanja koristi određeni algoritam koji služi usmjerniku za odabir najboljih putanja do svake podmreže u sustavu (u usmjerničkoj tabilici nalaze se samo najbolje putanje).

Tri su ključne vrste protokola usmjeravanja:

1. unutarnji (engl. *Interior Gateway Protocols–IGP*) i vanjski (engl. *Exterior Gateway Protocols–EGP*)
2. klasificirani (engl. *Classful*) i neklasificirani (engl. *Classless*) protokoli
3. Protokoli vektora udaljenosti (engl. *Distance Vector*) i protokoli stanja veze (engl. *Link-State*)

Unutarnji protokoli namijenjeni su razmjeni informacija o podmrežama u sustavu koji je pod jednom administrativnom kontrolom i nema doticaja s drugim sustavima u smislu razmjene putanja. Od unutarnjih protokola usmjeravanja danas se koriste RIP, EIGRP i OSPF za IPV4, a od vanjskih u primjeni je jedino BGP.

Podjela na klasificirane i neklasificirane odnosi se na način na koji protokoli usmjeravanja gledaju na podmreže u smislu mrežne maske. Tako RIPv1 uopće ne šalje informaciju o mrežnoj maski, već pretpostavlja da sve podmreže u sustavu koriste mrežne maske pojedine klase. Dobije li, npr., usmjernik informaciju o podmreži 10.12.10.0, on zaključuje da je to zapravo podmreža 10.0.0.0/8 iako u poruci nije bila

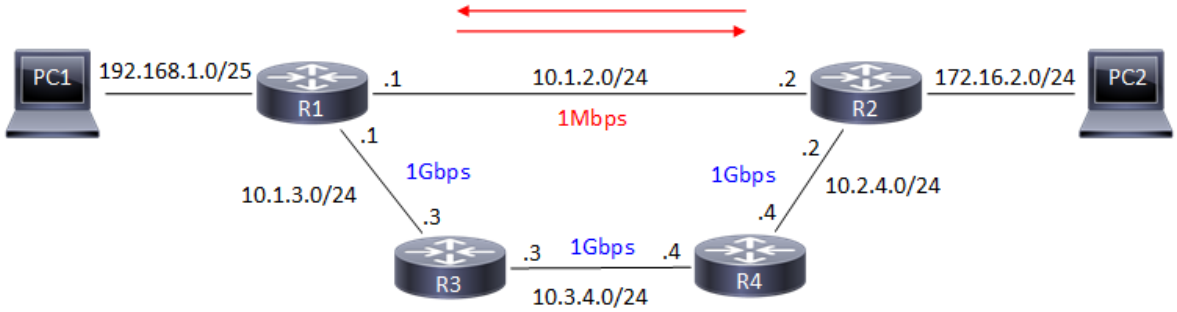
informacija o mrežnoj maski. Danas to nema smisla i nije primjenjivo u praksi, pa se protokoli koji su klasificirani više uopće ne koriste.

Podjela na protokole vektora udaljenosti i stanja veze odnosi se na način na koji usmjernici gledaju na ukupnu topologiju mreže. Protokoli vektora udaljenosti ne gledaju mrežu kao cjelinu, već vjeruju informacijama koje dobiju od svojih susjeda, a to su: podmreža, smjer i udaljenost do te podmreže. Kada usmjernici dobiju novu informaciju o nekoj podmreži, usporedit će je s postojećom informacijom u tablici usmjeravanja i ako nova putanja ima manju „udaljenost“, oni će jednostavno ažurirati svoju tablicu usmjeravanja. Takav način razmjene informacija o putanjama, gdje svaki usmjernik samo prihvaća što mu susjedni usmjernik pošalje, naziva se još i usmjeravanje na temelju glasina (engl. *Routing by Rumor*). Za razliku od protokola vektora udaljenosti, protokoli stanja mreže najprije razmijene informacije o svim podmrežama u sustavu, a te podmreže u ovom kontekstu nazivamo stanjima veza (engl. *Link State*), odakle i naziv takvih protokola. Kada usmjernik ima informacije o stanjima svih veza u sustavu, pokreće algoritam kojim gradi topologijsku tablicu na temelju koje odabire najbolje putanje do svake podmreže i upisuje ih u tablicu usmjeravanja. Protokole vektora udaljenosti može se usporediti s raskrižjem na kojem usmjernik promatra odredišta s putokazima i odlučuje koji je najbolji put na temelju informacije o smjeru i udaljenosti, dok se protokole stanja veze može usporediti s jednakom kartom i informacijama, koju svaki usmjernik ima kao i ostali usmjernici u mreži, i svaki od njih na temelju te karte odlučuje koji je najbolji put za njega. Od protokola vektora udaljenosti danas se koriste RIP i EIGRP, a OSPF je protokol stanja mreže, koji je u širokoj primjeni, posebno u telekomunikacijskim mrežama jer podržava hijerarhijski dizajn usmjerničke domene, što je prednost u jako velikim mrežama.

2.16. Protokol usmjeravanja RIP

Postoje dvije verzije RIP protokola za IPv4, ali u ovom priručniku u središtu je samo RIPv2. Zbog svojih karakteristika i ograničenja RIPv1 danas nema nikakvu praktičnu primjenu. Dalje u tekstu kada se spominje RIP, misli se na RIPv2. U odnosu na EIGRP ili OSPF, RIP je jednostavniji protokol i ujedno ima najmanje mogućnosti, tako da nije iskoristiv u većim i kompleksnijim računalnim mrežama. Jedno je od glavnih ograničenja što RIP koristi broj usmjernika (engl. *Hop Count*) na putu do odredišne

podmreže kao glavni kriterij prilikom odabira najbolje putanje. To znači da ne uzima u obzir brzinu veza između usmjernika, pa se često može dogoditi neoptimalno usmjeravanje prometa kroz mrežu. Tako bi na slici 155. promet između PC1 i PC2 išao preko sporije veze od 1Mbps, iako postoji tisuću puta brža veza, a samo zato što je manji broj usmjernika tim putem. Iako se možda na prvi pogled čini da broj usmjernika igra nekakvu ulogu u prosljeđivanju prometa, pa je manje bolje, danas su usmjernici toliko brzi i efikasni da broj usmjernika na putu do odredišta za većinu prometa nema nikakvu ulogu. Osim toga, mreže su uglavnom optimalno dizajnirane, pa se neće dogoditi mogućnost da promet ide preko tisuću puta sporije veze. Danas je važna stabilnost mreže, brza reakcija na promjenu i da usmjernici uvijek odabiru optimalnu putanju mrežom temeljenu na kvalitativnim parametrima poput propusnosti.



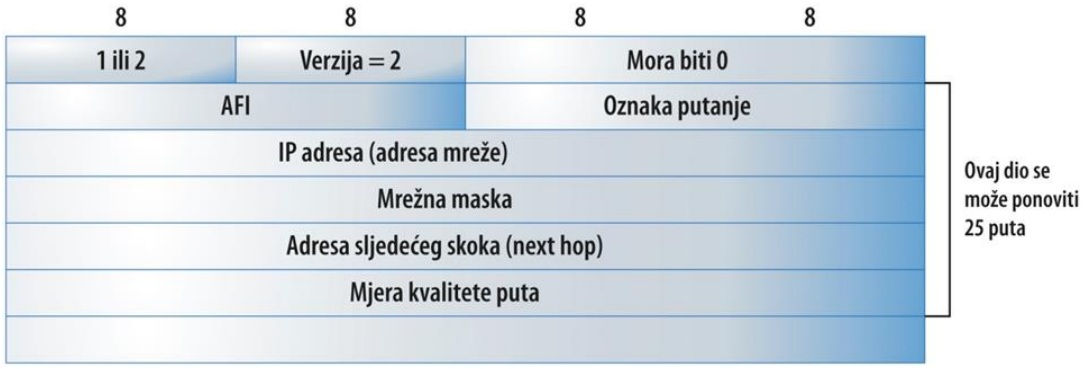
Slika 155. Ilustracija neoptimalnog usmjeravanja vezom od 1Mbps zbog RIP metrike

Pogleda li se tablicu usmjeravanja usmjernika R1 jasno je da je neoptimalno usmjeravanje. Sav promet između računala PC1 i PC2 ide najsporijom vezom u mreži. Razlog je u tome što RIP ne mari za brzinu veza ili neke druge kvalitativne parametre, već jedino uzima u obzir broj usmjernika do odredišne mreže, tako da je na topologiji na slici 155. najkraći put upravo najsporijom vezom. Na usmjerniku R2 ista je situacija.

```
R1#show ip route rip
-dio ispisa nije prikazan-
     10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
R       10.2.4.0/24 [120/1] via 10.1.2.2, 00:00:15, Serial6/0
R       10.3.4.0/24 [120/1] via 10.1.3.3, 00:00:11, GigabitEthernet1/0
     172.16.0.0/24 is subnetted, 1 subnets
R       172.16.2.0 [120/1] via 10.1.2.2, 00:00:15, Serial6/0
R1#
```

Slika 156. Tablica neoptimalnog usmjeravanja temeljenoga na RIP metrici

Maksimalno iskoristiva metrička vrijednost za RIP je 15, što znači da podmreže koje su udaljene više od 15 usmjernika, ne mogu komunicirati bez posebnog manipuliranja putanjama u mreži, a to nije dobra praksa u kontekstu RIP protokola. Usmjernici koriste i metričku vrijednost 16 za razne potrebe, no takvu metriku usmjernik smatra beskonačnom, pa takve putanje ne koristi. Svi protokoli usmjeravanja, pa tako i RIP, koriste poruke za razmjenu informacija o putanjama. Usmjernik koji koristi RIP protokol, svakih 30 sekundi šalje poruku s popisom do 25 putanja koje ima u svojoj tablici usmjeravanja, a koje su oglašene ili ih je saznao pomoću RIP-a. Te poruke u obliku UDP datagrama usmjernici šalju na višeodredišnu IP adresu 224.0.0.9 na odredišni UDP port 520. Najveća veličina RIP-a također koristi i vrijednost administrativne udaljenosti koja iznosi 120. Tu vrijednost usmjernik koristi da bi znao koji je protokol usmjeravanja vjerodostojniji izvor informacije ili koji je protokol usmjeravanja, može se reći, bolji. Ta se vrijednost, kao i metrička vrijednost, može mijenjati po želji, ali treba paziti da rezultat ne bude neoptimalno usmjeravanje u mreži.



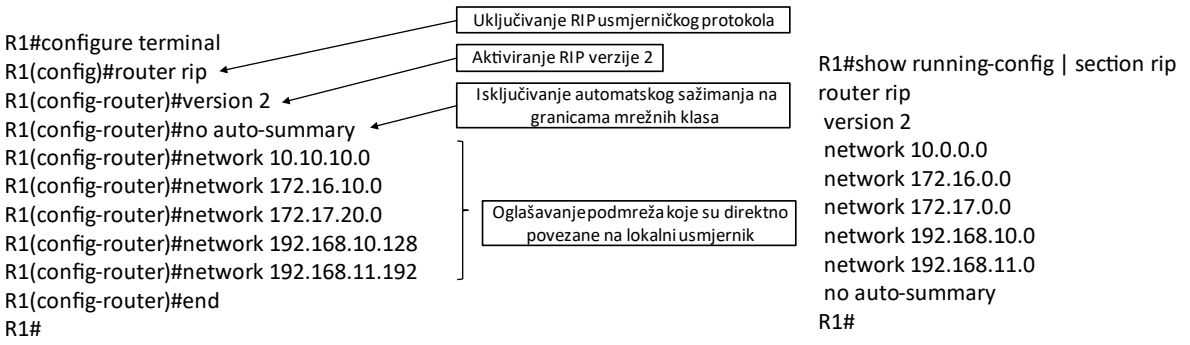
Slika 157. Format RIPv2 poruke

RIPv2 poruka sastoji se od sljedećih polja:

- **tip poruke** (engl. *Command*) – tip poruke može biti zahtjev ili odgovor. Zahtjev = 1 (engl. *Request*), odgovor = 2 (engl. *Response*);
- **verzija** (engl. *Version*) – verzija RIP protokola (RIPv1 = 1 ili RIPv2 = 2);
- **mora biti nula** – mjesto za buduće širenje protokola;
- **AFI** (engl. *Address Family Identifier*) – postavlja se 2 za IP;
- **Oznaka putanje** (engl. *Route Tag*) – koristi se za označavanje i filtriranje putanja redistribuiranih u RIP
- **IP adresa** – odredišna adresa (može biti mreža, podmreža ili IP adresa);
- **mrežna maska** (engl. *Subnet Mask*) – određuje mrežni dio IP adrese;

- **adresa sljedećeg skoka** (engl. *Next Hop*) – IP adresa sljedećeg skoka;
- **mjera kvalitete puta** (engl. *Metric*) – broj skokova do cilja (engl. *Hop Count*).

Primjer naredbi koje su potrebne za konfiguraciju RIP protokola prikazan je na slici 158.



Slika 158. Primjer konfiguriranja RIPv2 usmjerničkog protokola i ispisa trenutne konfiguracije

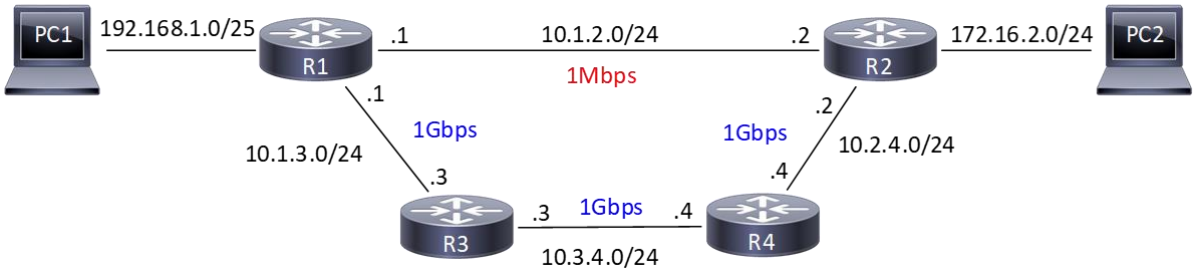
Kao što je vidljivo na slici 158., gdje je konfiguracija izvedena na usmjerniku *Cisco* serije 2800, unutar RIP protokola nije moguće upisati mrežnu masku. Ono što usmjernik koristi predefinirane su mrežne maske za svaku klasu, pa je zato mreža 172.16.10.0 u konfiguraciji vidljiva kao da je mrežna maska /16 iako u stvarnosti koristi mrežnu masku /24. To nije nužno problem osim u slučaju da postoji nekoliko mreža koje počinju s 172.16.X.X jer će u tom slučaju sve one biti oglašene iako se to možda ne želi. Bit će potrebni mehanizmi za filtriranje podmreža koje se ne želi oglasiti prema ostalim usmjernicima. Međutim, to ponašanje ovisit će o implementaciji RIP protokola na pojedinom usmjerniku, što ovisi o modelu i proizvođaču usmjernika. Princip vrijedi za sve tehnologije, no unatoč tome mogu imati različite opcije konfiguracije ovisno o proizvođaču opreme i konkretnom modelu uređaja.

2.17. Protokol usmjeravanja OSPF

OSPF (engl. *Open Shortest Path First*) kao protokol stanja veze usmjeravanja podrazumijeva da svi usmjernici unutar istog OSPF područja (engl. *Area*) imaju bazu sa zapisima o stanjima veza u tom području. Kada svi usmjernici razmijene informacije o svim vezama u mreži, svaki za sebe pomoću *Djikstra* algoritma izračuna koji je

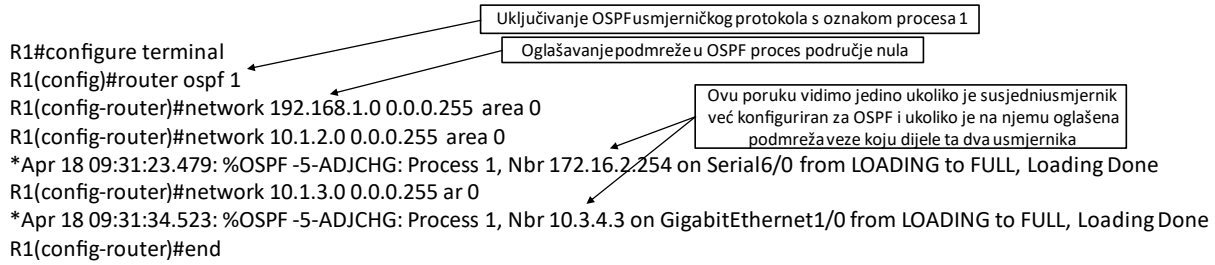
najbolji put do odredišta. OSPF predefinirano koristi vrijednost propusnosti na nekoj vezi. Predefinirana propusnost od 100 Mbps dat će metriku 1 u OSPF izračunu. Metrika se u OSPF protokolu naziva „Cost“ i također se može upisati ručno na svakom sučelju. Tako OSPF ne računa „Cost“ za tu vezu, već samo prihvati što je administrator upisao. Također se može mijenjati i administrativnu distancu za OSPF, koja je na usmjernicima *Cisco* predefinirano 110, a može se mijenjati i formula za izračunavanje metrike, pa u slučaju da u mreži ima veza propusnosti veće od 100Mbps, OSPF može točno izračunati stvarnu metriku. OSPF je hijerarhijski protokol koji podržava koncept više područja kako bi se u velikim mrežama moglo postići optimalnije usmjeravanje prometa i manje tablice usmjeravanja na usmjernicima. Kada se konfigurira OSPF, obavezno treba krenuti u konfiguraciji od područja nula. Područje nula naziva se još i područje okosnice (engl. *Backbone Area*) i sva komunikacija između drugih područja jedino može ići područjem nula. Dogodi li se situacija da recimo na područje 5, koje je povezano s područjem nula, povezuje neko područje 6, koje jedino ima vezu s područjem 5, područje 6 neće moći komunicirati s ostatkom mreže. Način na koji se takva situacija može privremeno riješiti jest konfiguracija tzv. virtualne veze iz područja koje nije povezano na područje nula prema rubnom usmjerniku između područja 5 i područja nula. Postoje razni tipovi OSPF područja koja se koristi za optimizaciju usmjeravanja u mreži, ali područje nula uvijek treba koristiti. Tehnički gledano, u slučaju male mreže koja se nikada ne planira širiti i povezivati s drugim mrežama, moglo bi se koristiti područje koje nije nula, ali tako bi se ograničio sav budući razvoj takve mreže, što svakako nije preporučljivo.

Za prikaz konfiguracije OSPF protokola koristit će se topologija na slici 159. Brzine sučelja su kako je prikazano na slici. U RIP protokolu nije bila važna propusnost na sučelju, već samo koliko je usmjernika do odredišta, a to potencijalno može rezultirati neoptimalnim usmjeravanjem.



Slika 159. Topologija za konfiguraciju OSPF protokola

Može se primijetiti na slici 160. da se prilikom oglašavanja mreže u OSPF koristi tzv. „Wildcard“ masku, isto kao i u EIGRP protokolu. To će također ovisiti o operativnom sustavu usmjernika, pa neki podržavaju i „Wildcard“ masku i mrežnu masku.



Slika 160. Konfiguracija OSPF protokola

„Wildcard“ maska dobije se promjenom vrijednosti svim bitovima u mrežnoj maski pa bit 1 postaje 0, i obratno. Na primjer, mrežna maska 255.255.255.0 tako postaje „Wildcard“ maska 0.0.0.255.

Max. Vrijednost mrežne maske	255.255.255.255	255.255.255.255	255.255.255.255
Mrežna maska za podmrežu	- 255.255.255.0	- 255.255.224.0	- 255.255.255.128
„Wildcard“ maska	000.000.000.255	000.000.031.255	000.000.000.127
Max. Vrijednost mrežne maske	255.255.255.255	255.255.255.255	255.255.255.255
Mrežna maska za podmrežu	- 255.255.255.252	- 255.255.255.192	- 255.255.255.240
„Wildcard“ maska	000.000.000.003	000.000.000.063	000.000.000.015

Slika 161. Usporedba „Wildcard“ maske s „običnom“ mrežnom maskom

Nakon što je pojedina podmreža oglašena u OSPF-u, na slici 161. vidi se da su uspostavljeni susjedski odnosi koji su potrebni kako bi usmjernici mogli međusobno razmijeniti informacije o stanju veza u mreži. Poruke su vidljive stoga što su susjedni usmjernici već prethodno konfigurirani za OSPF i veze su s usmjernikom R1 također

oglašene u OSPF-u. Čim je podmreža oglašena u OSPF-u, usmjernici automatski na tu vezu počinju slati OSPF *Hello* pakete u predefiniranim vremenskim razmacima od 10 sekundi i uspostavljaju susjedske odnose. Broj OSPF procesa lokalna je oznaka i može se razlikovati na usmjernicima, iako ima smisla da bude isti. Broj područja mora biti isti na oba usmjernika za podmrežu veze koju dijele međusobno. Na slici 162. prikazan je dio tablice usmjeravanja samo s OSPF putanjama gdje je vidljivo da OSPF, za razliku od RIP protokola, ne koristi naizgled najkraći put, odnosno put posredstvom manje usmjernika. Budući da OSPF koristi propusnost kao osnovu za izračun metrike po formuli *Metrika = Referentna propusnost (100Mbps) / propusnost sučelja* pa izračun bude manji od 1 (recimo ako su u mreži sučelja brzine 1 Gbps), usmjernik zaokružuje na 1 (vidi sliku 162.). U konačnici, kada usmjernik na temelju svih dostupnih informacija izračuna metriku u prikazanoj topologiji na slici 159., rezultat je znatno bolja metrika gigabitnim sučeljima nego serijskom vezom. Ta se propusnost na sučelju može mijenjati ručno, ali takva ručno konfigurirana propusnost utječe jedino na izračun metrike protokola usmjeravanja, ali nikako na stvarnu propusnost veze, za to postoje druge naredbe.

```
R1(config)#interface serial 6/0
R1(config-if)#bandwidth 1000
R1(config-if)#
```

Slika 162. Prikaz konfiguracije bandwith vrijednosti na sučelju u kilobitima po sekundi (Kbps)

```
R1#show ip route ospf
-dio ispisa nije prikazan-
     10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
O       10.2.4.0/24 [110/3] via 10.1.3.3, 00:00:07, GigabitEthernet1/0
O       10.3.4.0/24 [110/2] via 10.1.3.3, 00:00:07, GigabitEthernet1/0
     172.16.0.0/24 is subnetted, 1 subnets
O       172.16.2.0 [110/4] via 10.1.3.3, 00:00:07, GigabitEthernet1/0
R1#
```

Slika 163. Prikaz OSPF putanja u tablici usmjeravanja

Kako bi se OSPF prilagodio konkretnom okruženju te bio realan odraz brzina pojedinih veza u samoj metrici koju usmjernik koristi, prilagodit će se referentna propusnost u formuli za izračunavanje metrike, kako je prikazano na slici 164.

```
R1(config)#router ospf 1
R1(config-router)#auto-cost reference-bandwidth ?
  <1-4294967>  The reference bandwidth in terms of Mbits per second
R1(config-router)#auto-cost reference-bandwidth 10000
% OSPF: Reference bandwidth is changed.
      Please ensure reference bandwidth is consistent across all routers.
R1(config-router)#
```

Slika 164. Promjena referentne propusnosti za izračunavanje OSPF metrike na usmjerniku

Nakon što je to napravljeno na svim usmjernicima u svrhu konzistentnog izračuna u cijeloj mreži, tablica usmjeravanja to reflektira, i sada usmjernici imaju točan izračun metrike. U suprotnom, izračun metrike za vezu od 100Mbps i 1Gbps bio bi isti i iznosio bi 1, što bi uzrokovalo probleme u mreži, između ostalog neoptimalno usmjeravanje i neželjen raspored opterećenja (engl. *Loadbalance*). Postavljanjem referentne propusnosti na 10000, „Cost“ za veze od 1Gbps bit će 10, a za veze od 100Mbs bit će 100.

```
R1#show ip route ospf
-dio ispisa nije prikazan-
     10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
O       10.2.4.0/24 [110/21] via 10.1.3.3, 00:00:09, GigabitEthernet1/0
O       10.3.4.0/24 [110/20] via 10.1.3.3, 00:00:09, GigabitEthernet1/0
     172.16.0.0/24 is subnetted, 1 subnets
O       172.16.2.0 [110/121] via 10.1.3.3, 00:00:09, GigabitEthernet1/0
R1#
```

Slika 165. Prikaz dijela tablice usmjeravanja nakon promjene referentne vrijednosti bandidtha u OSPF-u

Da bi OSPF protokol u mreži radio očekivano, nužno je zadovoljiti nekoliko preduvjeta bez kojih OSPF usmjernici ne mogu uspostaviti susjedske odnose i ne mogu razmjenjivati informacije o podmrežama. Preduvjeti koji moraju biti zadovoljeni da bi dva OSPF usmjernika uspostavila susjedske odnose sljedeći su:

- podmreža na vezi koja povezuje dva usmjernika mora biti ista
- *Hello* i *Dead* vremena za OSPF moraju biti ista na oba usmjernika na vezi koja ih povezuje
- tip mreže na vezi koja povezuje dva usmjernika mora biti isti (engl. *Network Type*)

- prilikom oglašavanja podmreže veze koja povezuje dva usmjernika mora biti korišten isti broj područja
- prilikom oglašavanja treba točno upisati NETWORK naredbu
- treba koristiti istu vrstu provjere identiteta i iste lozinke.

Do sada je prikazana osnovna konfiguracija OSPF protokola i jednostavno mrežno okruženje. Za velike i kompleksne mreže OSPF podržava koncept više područja koja mogu imati različite karakteristike ovisno o tome što se želi postići u mreži. Taj dio izvan je opsega ovog priručnika.

2.18. Protokol usmjeravanja EIGRP

Protokol usmjeravanja EIGRP razvila je tvrtka *Cisco*. Po karakteristikama sličniji je OSPF protokolu, jer koristi i propusnost kao osnovu za izračunavanje metrike, no ona je puno kompleksnija nego u OSPF-u. Predefinirano EIGRP koristi najmanju propusnost na putu do odredišne podmreže i ukupno kašnjenje (engl. *Delay*) na vezama na putu do odredišta, no ne uzima stvarnu propusnost i ukupno kašnjenje, već vrijednosti koje su konfigurirane na sučelju. Tako se na izračun metrike može lako utjecati mijenjanjem tih vrijednosti. Isto kao u OSPF-u, vrijednosti koje se konfiguriraju, utječu isključivo na izračun EIGRP metrike i ne mijenjaju stvarno stanje na sučelju. Bitno je naglasiti da je vrijednost kašnjenja promjenjiva i ovisi o puno čimbenika. Preporuka tvrtke *Cisco* je koristiti predefinirane vrijednosti kako ne bi došlo do neoptimalnog usmjeravanja prometa. EIGRP protokol nema ugrađenu mogućnost hijerarhije unutar jednog autonomnog sustava, kao što OSPF ima, no svejedno se može implementirati kao hijerarhijski protokol koristeći različite autonomne sustave. Autonomni sustav je brojčana oznaka EIGRP procesa, koja mora biti ista na svim usmjernicima koji trebaju biti EIGRP susjedi. Za razliku od OSPF-a, gdje se koristi lokalno značajna brojčana oznaka procesa, u EIGRP protokolu broj autonomnog sustava mora biti isti na svim usmjernicima koji su dio cjeline.

EIGRP koristi DUAL (engl. *Diffusing Update Algorithm*) algoritam za izračun koje putanje će biti spremljene u tablicu topologije, a koje će ići i u tablicu usmjeravanja. Najbolje putanje idu u tablicu usmjeravanja, a rezervne najbolje putanje bit će u tablici topologije. To daje na raspolaganje usmjerniku, u slučaju da se najbolja putanja više ne može koristiti, već izračunatu drugu najbolju putanju, što znatno ubrzava

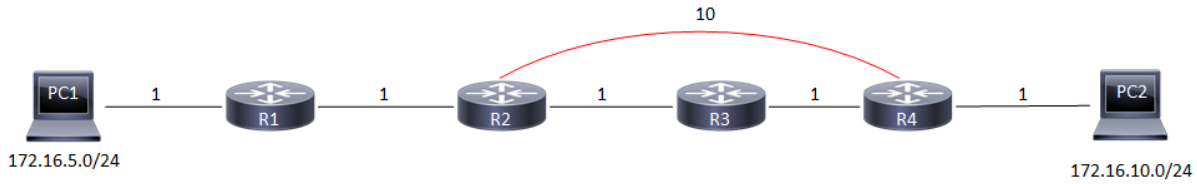
konvergenciju mreže. DUAL algoritam omogućava usmjernicima korištenje primarnih i rezervnih putanja koje ne mogu uzrokovati usmjerničke petlje i brzu konvergenciju mreže. Važni termini u EIGRP-u su *Successor* i *Feasible Successor.*

Successor je susjedni usmjernik posredstvom kojega je lokalni usmjernik izračunao da je najbolji put do odredišne podmreže (engl. *Feasible Distance)*.

Feasible successor je najbolji susjedni usmjernik na putu prema odredišnoj podmreži i nalazi se u tablici topologije. Da bi neki usmjernik mogao biti odabran za *Feasible Successor*, nužno je da bude zadovoljen uvjet *Feasibility Condition* koji kaže da vrijednost *Reported Distance*, za neku putanju koju šalje potencijalni *Feasible Successor* usmjernik, mora biti manja od vrijednosti *Feasible Distance* trenutne putanje koju lokalni usmjernik već ima.

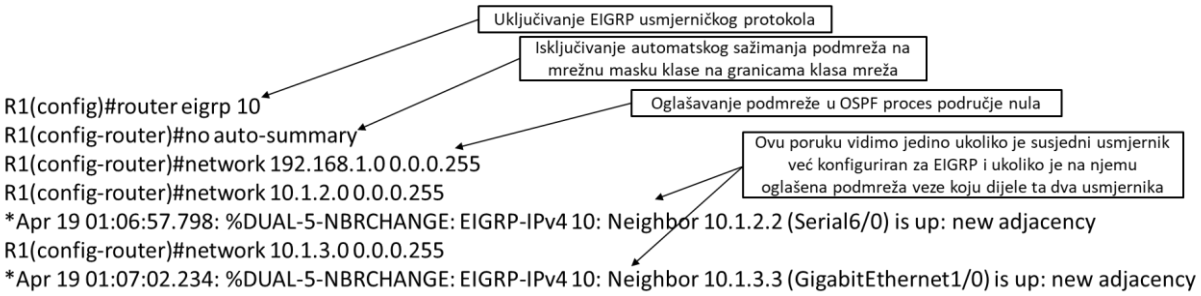
Dakle *Advertised Distance* usmjernik uči od susjeda (koliko je njima udaljena neka podmreža), a *Feasible Distance* je vrijednost koju usmjernici računaju sami za sebe. Na primjeru na slici 166. pretpostavka je da sve veze imaju metriku 1, osim veze između R2 i R4, koja ima metriku 10. R2 izračuna *Feasible Distance* 2 i oglašava to prema R3 i R4. Usmjernik R3 je dobio *Reported Distance* od R2, koja iznosi 2 i izračunava svoj *Feasible Distance* u iznosi 3. R3 to oglašava dalje prema R4 i usmjernik R4 na temelju *Reported Distance* od usmjernika R3 izračunava svoj *Feasible Distance* do podmreže 172.16.5.0/24 u iznosu 4. Usmjernik R4 oglašava tu podmrežu prema usmjerniku R2 i usmjernik R2 izračunava da preko usmjernika R4 *Feasible Distance* iznosi 14. Također, usmjernik R2 oglašava podmrežu 172.16.5.0/24 prema usmjerniku R4 na temelju čega R4 izračunava *Feasible Distance* u iznosu 12. Dok je sve stabilno u mreži, jasno je da se veza između R2 i R4 neće koristiti jer postoji bolji put. Međutim, otkaže li veza između R1 i R2, usmjernik će R2 izgubiti mogućnost korištenja trenutne *Successor* putanje i treba je zamijeniti nekom drugom. S obzirom na to da postoji putanja koju usmjernik R2 dobiva od usmjernika R4, pitanje je treba li koristiti tu putanju? Odgovor je ne, ali kako će to usmjernik zaključiti? Tu dolazi do izražaja *Feasibility Condition* koji kaže da vrijednost *Reported Distance*, za neku putanju koju šalje potencijalni *Feasible Successor* usmjernik, mora biti manja od vrijednosti *Feasible Distance* trenutne putanje koju lokalni usmjernik već ima. U prikazanom slučaju, usmjernik R2 uspoređuje svoj trenutni *Feasible Distance* od 2 i *Reported Distance* od usmjernika R4 koji iznosi 4, i zaključuje kako uvjeti nisu

zadovoljeni da bi koristio tu putanju jer bi mogla nastati petlja. Dogodi se da R2 briše svoju putanju za podmrežu 172.16.5.0/24 iz memorije, a samim time je više ne oglašava drugim usmjernicima i ta putanja nestaje iz cijele mreže.



Slika 166. Primjer Feasibility Condition u maloj mreži

Sama konfiguracija EIGRP protokola jednostavna je i slična konfiguraciji RIP i OSPF protokola. Isto kao RIP protokol, EIGRP na nekim usmjernicima predefinirano sažima podmreže na granicama klasa mreža, što se možda i ne koristi u modernim mrežama, već noviji usmjernici to obično imaju onemogućeno u konfiguraciji. Vidljivo je na slici 167. da se prilikom oglašavanja mreže u EIGRP-u koristi tzv. „Wildcard“ maska isto kao i u OSPF-u. To ovisi također o operativnom sustavu usmjernika, pa neki podržavaju i „Wildcard“ masku i mrežnu masku.



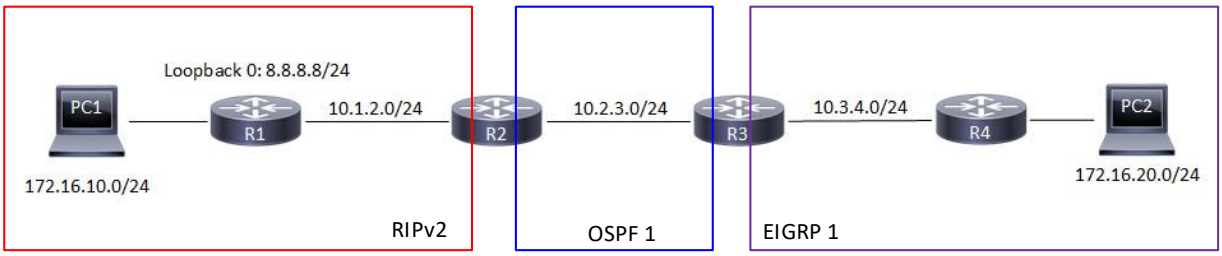
Slika 167. Primjer konfiguracije EIGRP protokola

```
R1#show ip route eigrp
-dio ispisa nije prikazan-
     10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
D       10.2.4.0/24 [90/3328] via 10.1.3.3, 00:00:25, GigabitEthernet1/0
D       10.3.4.0/24 [90/3072] via 10.1.3.3, 00:00:25, GigabitEthernet1/0
     172.16.0.0/24 is subnetted, 1 subnets
D       172.16.2.0 [90/28928] via 10.1.3.3, 00:00:25, GigabitEthernet1/0
```

Slika 168. Primjer EIGRP putanja u tablici usmjeravanja

2.19. Redistribucija između protokola usmjeravanja

Do sada je bila vidljiva konfiguracija usmjernika za korištenje pojedinih protokola usmjeravanja da bi jednostavnije mogli razmjenjivati informacije o podmrežama u sustavu i odrediti najbolji put do svake od njih. Međutim, što uraditi u situaciji kada se dvije mreže koje koriste različite protokole usmjeravanja trebaju povezati? Moguć je odgovor zamjena protokola usmjeravanja u jednoj mreži tako da bude jedinstven u cijelom sustavu, međutim to nije uvijek moguće iz raznih razloga. Zato postoji mogućnost redistribucije informacija između protokola usmjeravanja. To može biti vrlo zahtjevno ako se podmreže preklapaju pa treba filtrirati putanje prilikom redistribucije i slično, iako je u osnovi sama konfiguracija redistribucije jednostavna. Prilikom konfiguracije redistribucije, a ovisno o tome između kojih protokola usmjeravanja se ona radi, treba paziti na ispravnost implementacije. Na slici 169. primjer je jedne jednostavne topologije na kojoj je potrebno implementirati redistribuciju između protokola usmjeravanja kako bi u konačnici računala PC1 i PC2 mogla međusobno komunicirati. Nije to uobičajena situacija u računalnim mrežama imati više točaka redistribucije u sustavu. Preporuka je imati što manje takvih točaka kako bi usmjeravanje u mreži bilo jasnije i lakše za upravljanje.



Slika 169. Topologija s redistribucijom

Prilikom redistribucije može se u bilo koji od protokola usmjeravanja redistribuirati bilo koji drugi protokol usmjeravanja, direktno povezane podmreže, predefiniranu putanju i statičke putanje. Ovisno o usmjerničkom protokolu postoje neke specifičnosti.

Uobičajeno je da se sve željene podmreže protokolom usmjeravanja oglašavaju naredbom „network“. U specifičnim situacijama za oglašavanje odnosno redistribuciju svih direktno povezanih podmreža koristi se naredba „redistribute connected“. Međutim, najčešće ne jer se na taj način oglase sve direktno povezane podmreže, a

to se obično ne želi. Uvijek je potrebna kontrola nad procesima, pa tako i u ovom slučaju.

Također, moguće je redistribuirati statičke putanje naredbom „redistribute static“, ali ako je topologija kompleksna, to može uzrokovati poteškoće u usmjeravanju prometa, pa čak i petlje, tako da je potreban oprez i dobro znati što se želi postići.

Uobičajeno je u neki protokol usmjeravanja prebaciti putanje iz drugog usmjerničkog protokola. Da bi to bilo moguće, nužno je da jedan usmjernik koristi oba usmjernička protokola uz određene naredbe.

Prilikom redistribucije u RIP i EIGRP protokole usmjeravanja nužno je upisati i metriku, jer se u suprotnom te putanje neće redistribuirati pod pretpostavkom da je metrika beskonačna, što znači da se takve putanje neće oglašavati dalje.

```
R2(config-router)#redistribute ?
  bgp           Border Gateway Protocol (BGP)
  connected     Connected
  eigrp         Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis          ISO IS-IS
  iso-igrp      IGRP for OSI networks
  lisp          Locator ID Separation Protocol (LISP)
  mobile        Mobile routes
  odr           On Demand stub Routes
  ospf          Open Shortest Path First (OSPF)
  ospfv3        OSPFv3
  rip           Routing Information Protocol (RIP)
  static        Static routes
```

Slika 170. Primjer redistribucije u RIP — vrlo slično je i u ostalim protokolima usmjeravanja

Ovisno o operativnom sustavu i o mogućnostima usmjernika, naredbe će biti različite, kao i mogućnosti redistribucije, no princip ostaje isti bez obzira o kojem se protokolu usmjeravanja radi.

```
R2(config)#router rip
R2(config-router)#redistribute ospf 1 metric 15
R2(config-router)#exit
R2(config)#
```

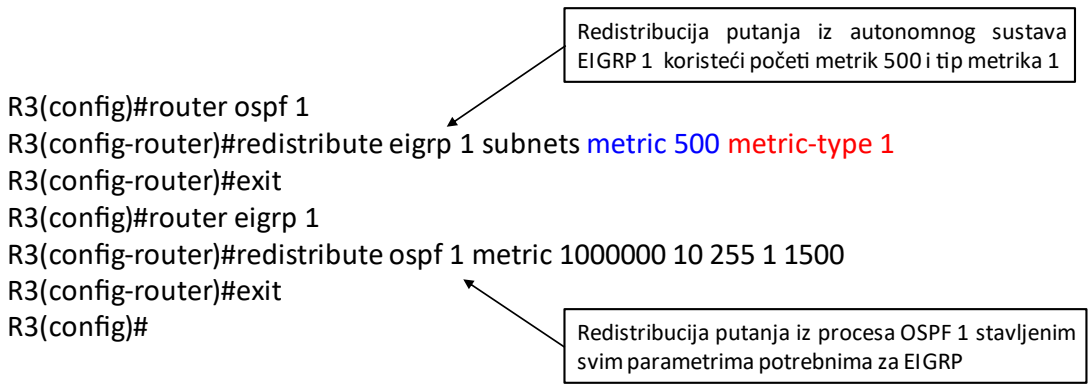
Slika 171. Redistribucija OSPF protokola u RIP s maksimalno korisnom metrikom

Na slici 171. prikazana je redistribucija OSPF putanja u RIP protokol, s početno postavljenom maksimalno iskoristivom metrikom za neku putanju u RIP protokolu. Kako 16 znači nedostupnu mrežu, a metrika se akumulira, već na sljedeći usmjernik na putu više neće prosljeđivati informacije. To, naravno, nije poželjno u mreži, pa se metriku prilikom redistribucije postavlja na puno manji broj (na primjer jedan) i tako omogućuje informacijama da se oglašavaju najdalje u mrežu. Kako bi redistribucija imala smisla, obično je dvosmjerna, pa se tako na usmjerniku R2 s topologije na slici 169. RIP putanja redistribuira u OSPF protokol. Kako se to radi vidljivo je na slici 172.

```
R2(config)#router ospf 1
R2(config-router)#redistribute rip subnets metric 20 metric-type 2
R2(config-router)#exit
R2(config)#
```

Slika 172. Primjer redistribucije RIP putanja u OSPF protokol usmjeravanja

Bitno je znati da prilikom redistribucije u OSPF nije nužno upisivati metriku zato što su te vrijednosti već predefinirane, osim ako se želi koristiti neke druge. Dakle, predefinirano sve putanje koje se iz RIP ili iz EIGRP protokola usmjeravanja redistribuiraju u OSPF već imaju postavljenu metriku tipa 2, što znači da se ona neće mijenjati razmjenom informacija između usmjernika u OSPF usmjerničkoj domeni. Također predefiniran iznos metrike prilikom redistribucije u OSPF je 20, tako da i taj dio neće imati utjecaja. Kako postoji metrika tipa 2, tako postoji metrika tipa 1, a to bi trebalo ručno konfigurirati ako se želi da se metrika za neku putanju, koja je redistribuirana u OSPF, mijenja kako se informacija o toj putanji razmjenjuje između usmjernika u OSPF usmjerničkoj domeni. Sve to upućuje na zaključak da bi redistribucija RIP putanja u OSPF bila uspješna da je samo korištena naredba „redistribute RIP subnets“. Bez dijela naredbe „subnets“ usmjernik bi u OSPF redistribuirao samo podmreže koje koriste klasnu mrežnu masku npr. 10.0.0.0/8 ili 172.17.0.0/16, a takvih podmreža u modernim mrežama praktički i nema, pa bi se dogodilo da ni jedna varijabilno subnetirana podmreža ne bi bila redistribuirana. Preciznije, podmreže iz klase C privatnih IP adresa (192.168.0.0/16) obično su podmrežene mrežnom maskom /24, što će proći prilikom redistribucije, ali mreže 192.168.10.0/26, 10.10.10.0/18 ili 172.16.0.0/23 neće proći, jer ne koriste klasnu mrežnu masku. Sve to vrijedi i za redistribuciju EIGRP putanja u OSPF.



Slika 173. Primjer redistribucije između OSPF i EIGRP protokola usmjeravanja

Kao što je vidljivo na slici 173., prilikom redistribucije EIGRP putanja u OSPF protokol usmjeravanja korištena je metrika tipa 1, što nije predefnirano. Bitno je znati da se za OSPF koristi termin „proces“, dok za EIGRP termin „autonomni sustav“. Na jednom usmjerniku može biti konfigurirano više OSPF procesa i više EIGRP autonomnih sustava, ali to nije dobra praksa u većini slučajeva.

Prilikom redistribucije u EIGRP nužno treba unijeti sve parametre koje on koristi u formuli za izračun vrijednosti metrike, a to su najniža propusnost (engl. *Bandwidth*), kašnjenje (engl. *Delay*), pouzdanost (engl. *Reliability*) i opterećenje (engl. *Load*), uz još vrijednost *MTU* (engl. *Maximum Transmission Unit*). Iako se za izračun metrike u EIGRP-u predefinirano koristi samo propusnost i *delay*, svejedno treba upisati sve početne parametre, inače se redistribucija neće dogoditi. Svi se elementi konfiguracije spomenutih protokola mogu mijenjati, iako to nije preporuka.

Na slici 174. može se vidjeti koji se sve parametri, i na koji način, nekog izvora informacije o putanjama, na primjer statičke putanje koje je upisao administrator, trebaju upisati prilikom redistribucije u EIGRP protokol.

Preporuka je tvrtke *Cisco*, koja je osmislila EIGRP protokol, da se koriste samo predefinirane vrijednosti i da se ne mijenja njihov težinski faktor u formuli za izračunavanje metrike. Težinski su faktor u EIGRP-u tzv. „k“ vrijednosti, dodijeljene svakom elementu a određuju njihov utjecaj prilikom izračuna metrike za neku putanju. Taj dio nije u opsegu ovog priručnika, no važno je znati da promjenu tih „k“ vrijednosti treba provesti na svim EIGRP usmjernicima jer u suprotnom EIGRP protokol neće raditi.

```
R3(config)#router eigrp 1
R3(config-router)#redistribute static metric ?
  <1-4294967295>  Bandwidth metric in Kbits per second
R3(config-router)#redistribute static metric 100000 ?
  <0-4294967295>  EIGRP delay metric, in 10 microsecond units
R3(config-router)#redistribute static metric 100000 10 ?
  <0-255>  EIGRP reliability metric where 255 is 100% reliable
R3(config-router)#redistribute static metric 100000 10 255 ?
  <1-255>  EIGRP Effective bandwidth metric (Loading) where 255 is 100% loaded
R3(config-router)#redistribute static metric 100000 10 255 1 ?
  <1-65535>  EIGRP MTU of the path
R3(config-router)#redistribute static metric 100000 10 255 1 1500 ?
  route-map  Route map reference
  <cr>
R3(config-router)#redistribute static metric 100000 10 255 1 1500
```

Slika 174. Prikaz parametara koji se unose prilikom redistribucije nekog izvora informacije u EIGRP protokol

Sada kada su upisane sve naredbe za redistribuciju između protokola usmjeravanja za topologiju prikazanu na slici 169., treba pogledati tablice usmjeravanja na krajnjim usmjernicima R1 i R4. Ono što je vidljivo — usmjernici R1 i R4 znaju za sve podmreže koje nisu direktno povezane na te usmjernike, što je i svrha protokola usmjeravanja. Također, na usmjerniku R1 postoje dvije putanje s metrikom 15, najvećom iskoristivom, koju koriste RIP protokol. Svi usmjernici kojima bi usmjernik R1 slao informacije o tim putanjama, više ih ne bi mogli koristiti, jer bi metrika za te putanje bila 16, a to je vrijednost koju RIP smatra „beskonačnom“, što znači da takve putanje ne koristi.

```
R1#show ip route
-dio ispisa nije prikazan-
R       10.2.3.0/24 [120/1] via 10.1.2.2, 00:00:15, GigabitEthernet1/0
R       10.3.4.0/24 [120/15] via 10.1.2.2, 00:00:15, GigabitEthernet1/0
     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
R       172.16.20.0/24 [120/15] via 10.1.2.2, 00:00:15, GigabitEthernet1/0
```

```
R4#sh ip route eigrp
-dio ispisa nije prikazan-
     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
D EX    10.1.2.0/24 [170/5376] via 10.3.4.3, 01:26:13, GigabitEthernet1/0
D EX    10.2.3.0/24 [170/5376] via 10.3.4.3, 01:26:13, GigabitEthernet1/0
     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
D EX    172.16.10.0/24 [170/5376] via 10.3.4.3, 01:26:13, GigabitEthernet1/0
```

Slika 175. Prikaz usmjerničkih tablica na usmjernicima R1 i R4

Na usmjerniku R4 EIGRP putanje imaju oznaku „D EX“, što znači da su redistribuirane u EIGRP. Kada bi imale samo oznaku „D“, to bi značilo da su u EIGRP oglašene korištenjem naredbe „network“ i da su tako došle do usmjernika R4. U ovom primjeru nema takvih putanja, ali kada bi se na usmjernik R3 dodalo logičko sučelje *Loopback 2*, konfiguriralo na njega IP adresu i tu podmrežu oglasilo u EIGRP naredbom „network“, tada bi se na R4 to vidjelo kao putanja s oznakom „D“. Oznaka za EIGRP je „D“ jer je to početno slovo algoritma koji je u pozadini EIGRP protokola.

Još je važno vidjeti kako se u svaki od protokola usmjeravanja redistribuira predefinirana putanja. U RIP i OSPF to se radi naredbom „default-information originate“, dok u EIGRP naredbom „redistribute static“. Međutim, prilikom redistribucije u EIGPR treba upisati i vrijednosti za početnu metriku, dok je u RIP i OSPF protokolu početna metrika predefinirana. U RIP protokolu iznosi 1, a OSPF protokolu iznosi 20 i koristi se metrika tipa 2. Prilikom redistribucije u RIP protokol naredba „default-information originate“ redistribuirat će predefiniranu putanju, čak i ako ona ne postoji na usmjerniku. To možda neće imati smisla s aspekta usmjeravanja prometa, jer je uobičajeno da usmjernik koji redistribuira predefiniranu putanju zaista i ima tu putanju, i ona nečemu služi, npr. za pristup internetu. Vrlo važno je zapamtiti da u OSPF protokolu, čak i ako redistribuira RIP protokol i ako se u tom RIP protokolu već nalazi predefinirana putanja, ona neće biti redistribuirana u OSPF bez naredbe „default-information originate“.

Nakon redistribucije predefinirane putanje na usmjerniku R1, i nakon što je na usmjerniku R2 u OSPF procesu dozvoljena redistribucija predefinirane putanje naredbom „default-information originate“, usmjernik R4 je saznao za predefiniranu putanju i računalo PC2 može komunicirati s IP adresom sučelja *Loopback 0* na usmjerniku R1.

3. Sigurnost u računalnoj mreži

U OVOJ CJELINI NAUČIT ĆETE:

- implementirati liste za kontrolu pristupa (ACL)
- implementirati NAT mehanizam
- implementirati IPSec tunele
- implementirati L2 sigurnosne mehanizme
- implementirati mehanizme za očuvanje stabilnosti STP topologije.

Sigurnost je karakteristika nekog sustava koja se ne može postići samo jednom tehnologijom, već je rezultat holističkog pristupa u osmišljavanju i primjeni sigurnosnih politika i procedura prema aktualnim prijetnjama, te kontinuirane edukacije korisnika IKT sustava.

U kontekstu IKT sustava sigurnost je trajni proces zaštite digitalnih podataka i IKT resursa od unutarnjih i vanjskih zlonamjernih ili slučajnih prijetnji. Taj proces obuhvaća detekciju, prevenciju i odgovor na prijetnje korištenjem adekvatne tehnologije i uređaja, programskih alata i servisa, te sigurnosnih politika i procedura.

U nastavku će biti obrađeni osnovni elementi koji se koriste za osiguranje pristupa IKT sustavu i filtriranje mrežnog prometa. Također će biti prikazan uobičajen način primjene tih mehanizama.

3.1. Liste za kontrolu pristupa

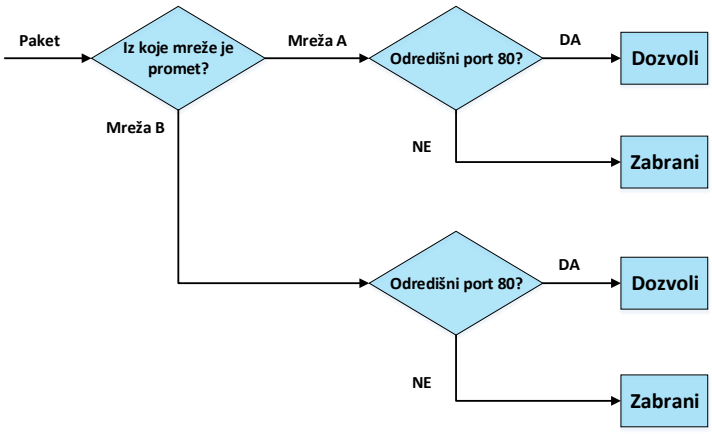
Lista za kontrolu pristupa (engl. *Access Control List — ACL*) lista je pravila koja se konfigurira na usmjerniku i koju usmjernik koristi kako bi provjerio svaki paket koji primi na sučelje. Usmjernik, na temelju pravila koja su konfiguriran u listi za kontrolu pristupa, odlučuje treba li neki paket propustiti ili odbaciti. Sva pravila u listi za kontrolu pristupa provjeravaju se redom dok se ne dođe do pravila koje odgovara pojedinom paketu, nakon toga usmjernik više ne provjerava listu za taj paket. Predefinirano usmjernik nema ni jednu listu za kontrolu pristupa, pa propušta sav promet koji odgovara zapisima u tablici usmjeravanja, a to znači da propušta i potencijalne napade na mrežu.

Ako se na usmjerniku konfigurirana lista za kontrolu pristupa nalazi na putanji paketa, usmjernik će osim gledanja u tablicu usmjeravanja, provjeriti i listu za kontrolu pristupa. U tom slučaju usmjernik može neki promet odbaciti ili propustiti ovisno o zapisima u listi za kontrolu pristupa.

Osim uloge filtriranja, liste za kontrolu pristupa u velikoj se mjeri koriste za klasifikaciju prometa na temelju čega se može konfigurirati NAT (engl. *Network Address Translation*) ili filtrirati putanje prilikom redistribucije, manipulirati BGP (engl. *Border Gateway Protocol*) atributima za određene putanje u mapama usmjeravanja i slično.

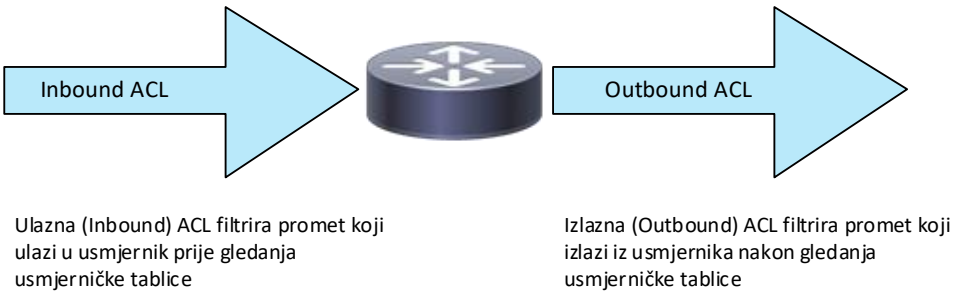
3.2. Princip rada liste za kontrolu pristupa

Kada dva računala žele komunicirati mrežom, za to, između ostalog, koriste IP adrese i TCP/UDP ulaze (mrežni i transportni sloj u OSI modelu). Te elemente može se iskoristiti u listama za kontrolu pristupa za zabranu ili propuštanje nekakvog prometa između dva krajnja računala. Npr. može se dozvoliti samo podmreži administratora pristup mrežnom poslužitelju po ulazu 22, a svim ostalim podmrežama u tvrtki zabraniti ulaz 22, ali dozvoliti ulaz 80, što omogućava sigurniju konfiguraciju i održavanje poslužitelja dok je istovremeno omogućeno korisnicima pristupati mrežnim stranicama. Također može se kontrolirati komunikacija između dvije podmreže propuštanjem samo komunikacije po određenim ulazima koji su potrebni, a sve ostalo zabraniti, i tako smanjiti površinu napada na računala. Na slici 176. prikazano je načelo rada liste za kontrolu pristupa.



Slika 176. Princip rada liste za kontrolu pristupa

Kako će lista za kontrolu pristupa (ACL) djelovati na određeni promet i hoće li uopće djelovati, ovisi na koji je uređaj u mreži i u koji smjer na sučelju postavljena. ACL mora biti postavljen na uređaj kroz koji prolaze paketi koje se želi filtrirati. Nema ga smisla postavljati na uređaj kroz koji zanimljiv promet ne prolazi, tako se može samo poremetiti komunikacija u mreži. Smjer djelovanja može biti na pakete koji ulaze u sučelje „inbound“ ili na pakete koji izlaze iz sučelja „outbound“. Također na jednom sučelju može se postaviti dvije liste za kontrolu pristupa, ali ne mogu biti u istom smjeru.



Slika 177. Smjer djelovanja liste za kontrolu pristupa na sučelju

Pravilo je da se na usmjerniku može primijeniti jedna listu za kontrolu pristupa po sučelju, smjeru i protokolu. To znači da na jednom sučelju u istom smjernu ne mogu biti dvije liste za kontrolu IPv4 prometa, ali bi mogle biti po jedna za IPv4 i za IPv6:

- Jedna lista za kontrolu pristupa po sučelju (npr. Fa0/0)
- Jedna lista za kontrolu pristupa po smjeru (ulazni promet – *Inbound*, izlazni promet – *Outbound*)
- Jedna lista za kontrolu pristupa po protokolu (npr. IPv4, IPv6, IPX, …).

Prednost je ACL-a postavljenih kada ulaze u sučelje usmjernika (*Inbound*) što usmjernik ne mora gledati tablicu usmjeravanja za svaki paket, već samo za one koji su propušteni kroz ACL. U slučaju izlaznog (*Outbound*) ACL-a, usmjernik bi gledao tablicu usmjeravanja za svaki paket, bez obzira što će ih možda 90% biti odbačeno kad dođu na izlazni ACL, što predstavlja nepotrebno trošenje resursa uređaja i usporavanje komunikacije mrežom.

Na kraju svakoga konfiguriranog ACL-a postoji skrivena tvrdnja „Implicit Deny“ a zabranjuje sav promet koji nije odgovarao ni jednoj tvrdnji prije. Kada bi se na sučelje primijenio ACL koji nema ni jednu tvrdnju, tada bi sav promet bio zabranjen zbog „Implicit Deny“ tvrdnje, i na to treba paziti prilikom konfiguracije.

3.3. Vrste lista za kontrolu pristupa

Postoji nekoliko vrsta lista za kontrolu pristupa (ACL), a obradit će se standardni i prošireni ACL-ovi, jer pomoću njih se može napraviti osnovno štićenje mreže i klasifikacija prometa za potrebe redistribucije putanja, prevođenja adresa (NAT) ili uspostavu tunela između tvrtki (GRE ili IPSec VPN).

STANDARDNI: Standardni ACL-ovi mogu filtrirati promet samo po izvorišnoj adresi, dok se ulazi i odredišna adresa uopće ne koriste. Prilikom konfiguracije na usmjerniku koriste se brojevne oznake od 1 do 99 i 1300 do 1999. za lakše pamćenje ili prepoznavanje čemu služi pojedini ACL, što može biti vrlo korisno, posebno kad ih postoji više na uređaju.

PROŠIRENI: Prošireni ACL-ovi koriste brojčane oznake od 100 do 199 i 2000 do 2699. Također, mogu se koristiti nazivi u istu svrhu kao i za standardne. Prošireni ACL-ovi mogu filtrirati promet po sljedećim elementima:

- Protokol (ICMP,ESP, GRE, OSPF, EIGRP, …)
- Izvorišna IP adresa (engl. *Source IP*);
- Odredišna IP adresa (engl. *Destination IP*);
- Izvorišni TCP ili UDP ulaz (engl. *Source Port*);
- Odredišni TCP ili UDP ulaz (engl. *Destination Port*).

Prilikom konfiguracije liste za kontrolu pristupa koristi se „Wildcard“ maska. Za razliku od mrežne maske, gdje je važan samo onaj dio gdje su bitovi 1, u „Wildcard“ maski važan je samo onaj dio gdje su bitovi 0. Prema tablici na slici 178. ACL će djelovati na sve IP adrese koje počinju sa 192.168.X.X (gdje je X bilo koji broj od 0 do 255).

	Decimalni zapis	Binarni zapis
IP adresa koja se provjerava u ACL	192.168.1.25	11000000.10101000.00000001.00011001
Wildcard maska	0.0.255.255	00000000.00000000.11111111.11111111
IP adresa na koju će ACL djelovati	192.168.0.0	11000000.10101000.00000000.00000000

Slika 178. Primjer djelovanja „wildcard“ maske

Ako se želi da lista za kontrolu pristupa filtrira samo jednu IP adresu, tada je „Wildcard“ maska 0.0.0.0, što se može kraće napisati **„host“** i predstavlja točno tu IP adresu s koje promet dolazi. Kada bi se htjelo filtrirati sve moguće IP adrese, tada „Wildcard“ maska izgleda ovako: 255.255.255.255, što se može kraće napisati **„any“.** Npr. želi li se zabraniti jednom računalu s IP adresom 192.168.0.10, a svim ostalim računalima u njegovoj mreži dozvoliti komunikaciju IP protokolom prema svim ostalim mrežama, tvrdnja bi u ACL-u izgledala ovako:

access-list 101 deny ip 192.168.0.10 *0.0.0.0* 0.0.0.0 0.0.0.0

access-list 101 permit ip 192.168.0.0 0.0.0.255 0.0.0.0 0.0.0.0

ili kraće:

access-list 101 deny ip *host* 192.168.0.10 any

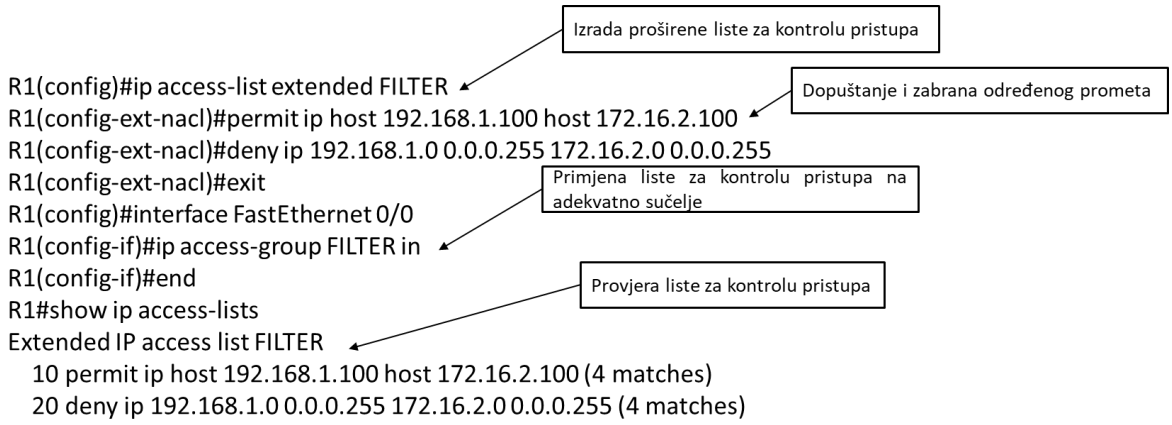
access-list 101 permit ip 192.168.0.0 0.0.0.255 any

To je moguće zato što se lista za kontrolu pristupa (ACL) ponaša kao skripta i usmjernik izvršava slijedno svaku tvrdnju u listi dok ne nađe odgovarajuću (engl. *Match*). Kada dođe do tvrdnje koja odgovara podacima u paketu koji provjerava (IP adrese i/ili ulazi) ne gleda ostale tvrdnje u listi, nego propusti ili odbaci paket. Zato treba paziti na redoslijed tvrdnji u ACL-u. Prema prethodnom zapisu prva tvrdnja zabranjuje komunikaciju samo IP adresi 192.168.0.10. Ako bi paket došao od računala s IP adresom 192.168.0.11, to će odgovarati drugoj tvrdnji. Ako bi slučajno paket došao

iz mreže koja nije navedena u tom ACL-u, npr. 192.168.20.0/24, tada će taj promet biti odbačen po zadnjoj nevidljivoj „Implicit Deny“ tvrdnji koju svaka lista za kontrolu pristupa ima.

3.4. Konfiguracija lista za kontrolu pristupa

Kao što je već rečeno, standardne liste za kontrolu pristupa mogu filtrirati promet samo po izvorišnoj IP adresi i najviše se koriste prilikom filtriranja putanja prilikom redistribucije između protokola usmjeravanja. Za opću primjenu puno su korisnije proširene liste za kontrolu pristupa, pa će one biti u središtu obrade. Proširene liste za kontrolu pristupa omogućavaju filtriranje na temelju izvorišne i odredišne IP adrese, izvorišnog i odredišnog ulaza i na temelju protokola.



Slika 179. Konfiguracija proširene liste za kontrolu pristupa

Postavlja se pitanje je li druga tvrdnja zaista potrebna u našem ACL-u? Što se tiče funkcionalnosti prikazanog ACL-a, i nije. Nakon prve tvrdnje gdje je dopuštena željena komunikacija, zadnja je tvrdnja u svakom ACL-u „Implicit Deny“, tako da bi sva ostala komunikacija između dvije mreže ionako bila zabranjena. Razlog zašto ipak treba druga tvrdnja je stoga da se vidi pokušava li netko iz mreže 192.168.1.0/24 komunicirati prema mreži 172.16.2.0/24, pa se na temelju toga može istražiti razlog i zaključiti treba li još nekom računalu dozvoliti komunikaciju ili možda postoje pokušaju nelegitimne komunikacije (napad). Čest je to slučaj prilikom konfiguracije ACL-a u nepoznatom okruženju. Alarm bi se svakako trebao upaliti ako prema „Deny“ tvrdnji postoji puno više komunikacije nego prema „Permit“ tvrdnji, to može biti znak da su

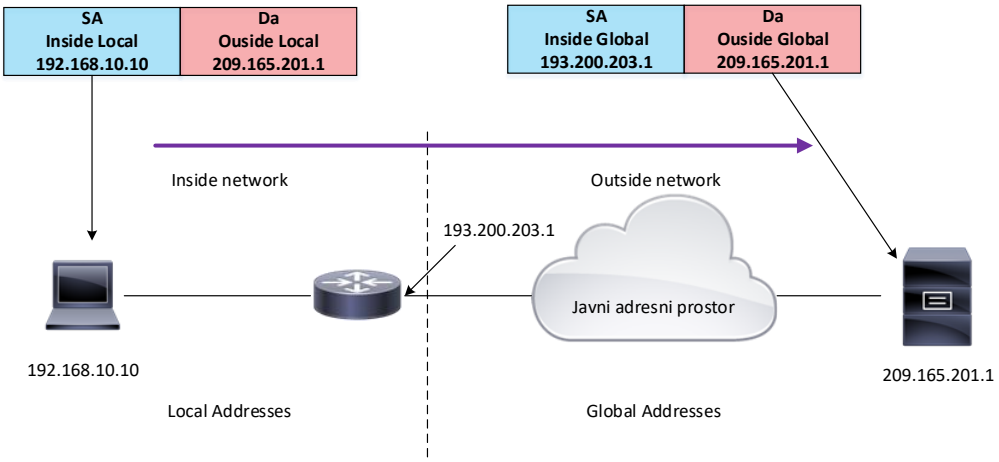
zaražena računala u mreži ili je zaboravljeno propustiti nešto potrebno za njezino poslovanje.

3.5. Primjena listi za kontrolu pristupa u konfiguraciji NAT mehanizma

Kao što je do sada naučeno, za komunikaciju u mrežama koristi se IP protokol, a još uvijek je najzastupljenija verzija IPv4. Problem IPv4 adresnog prostora je što je premalen za sve veće potrebe za IP adresama na internetu i nema ih dovoljno jedinstvenih za svaki uređaj povezan na mrežu. Povezivanje sve većeg broja uređaja na mrežu i sve više servisa samo pogoršava taj problem. Privremeno rješenje korištenje je privatnih adresnih prostora unutar privatnih mreža tvrtki i pretvaranje tih IP adresa u jednu ili više javnih IP adresa, što je omogućilo da se stotine IP adresa iz privatne mreže predstavljaju na internetu kao jedna javna IP adresa. To prevođenje IP adresa naziva se NAT (engl. *Network Address Translation*). Korištenjem NAT-a računalima je omogućena komunikacija internetom, uz istovremenu uštedu IPv4 adresnog prostora. I to ima svoje granice iskoristivosti, ali i neke komunikacijske probleme, poput *end-to-end* vidljivosti, tuneliranja, uspostavljanja TCP veza i slično. Osim nedostataka, NAT ima i prednosti, poput povećane sigurnosti mreže, veće fleksibilnosti povezivanja na javnu mrežu, konzistenciju unutarnje adresne sheme i slično. Bez korištenja NAT-a danas nije moguća komunikacija računala iz privatne mreže s računalima na internetu. Unutar privatne mreže bi tekla normalno, što može odgovarati jednom dijelu poslovanja, ali kako je danas internet u svim životnim aspektima, velika većina posla se ne bi mogla obaviti. Iako usmjernik, koji je jednim svojim dijelom na javnoj mreži, a drugim na unutarnjoj privatnoj mreži, može komunicirati i s jednom i drugom, bez NAT-a ne bi mogao biti most u komunikaciji računala u privatnoj mreži s internetom. Razlog je u činjenici što privatni adresni prostori nisu oglašeni u protokol usmjeravanja pružatelja usluga za usmjeravanje prometa na internetu. Sav promet koji ima za odredišnu mrežu nešto iz privatnog adresnog prostora , a nađe se na internetu, u konačnici će biti odbačen. Zbog toga treba usmjernik koji veže privatnu mrežu tvrtke i javni adresni prostor interneta konfigurirati da prevodi privatne u javne IP adrese, i obratno. Iako se NAT najviše koristi za prevođenje privatnih u javne IP adrese, može se koristiti za prevođenje bilo kojih IP adresa, pa tako možemo i privatnih u privatne ili javnih u javne. Primjer

prevođenja privatnih u privatne IP adrese je prilikom konfiguracije VPN tunela između dvije lokacije koje imaju identične podmreže. U tom slučaju mrežu na jednoj lokaciji treba prevesti u neku drugu da bi komunikaciju između tih lokacija bila moguća. U tom slučaju *Outside Local* i *Outside Global* IP adrese nisu iste!

Kako je prikazano na slici 180., postoji nekoliko ključnih termina vezanih uz NAT, ovisno o tome jesu li IP adrese privatne ili javne i jesu li pod osobnom kontrolom ili ne. Terminologija za NAT određena je iz perspektive uređaja čija se IP adresa prevodi. Termin *Inside Address* vezan je za uređaj čija se IP adresa prevodi, a termin *Outside Address* odredišni je uređaj. Također, adrese u tom smislu dijele se na lokalne i globalne. Lokalna adresa je bilo koja adresa koja je povezana s *Inside Network* dijelom, a globalna adresa je bilo koja adresa u *Outside* dijelu. Termini *inside*, *outside*, *local* i *global* kombiniraju se za određivanje o kojoj se IP adresi radi.



Slika 180. Princip rada NAT mehanizma

***Inside Local Address*:** IP adresa računala koju treba prevesti i kako je vidljiva u lokalnoj mreži.

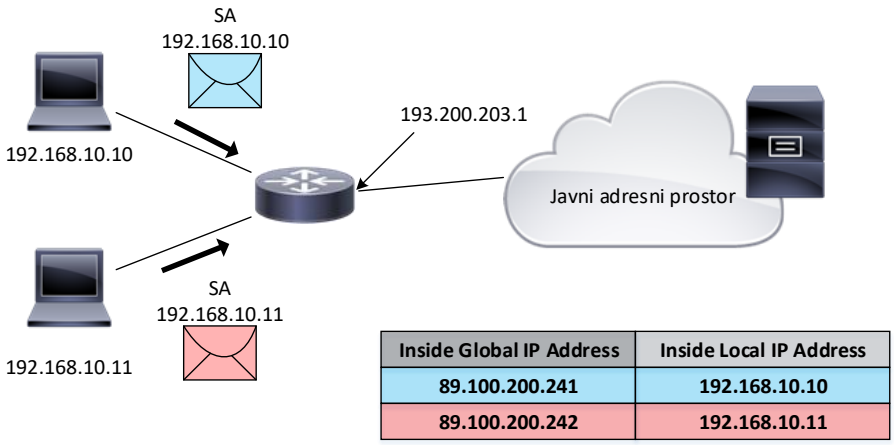
***Inside Global Address*:** javna IP adresa u koju se prevodi unutarnja IP adresa računala i kako je drugi vide izvan lokalne mreže.

***Outside Local Address*:** IP adresa odredišta kako je vidljiva u lokalnoj mreži.

Outside Global Address: IP adresa odredišta kakva zaista jest u odredišnoj mreži. Za računala s javnom IP adresom to je isto kao i *Outside Local Address.*

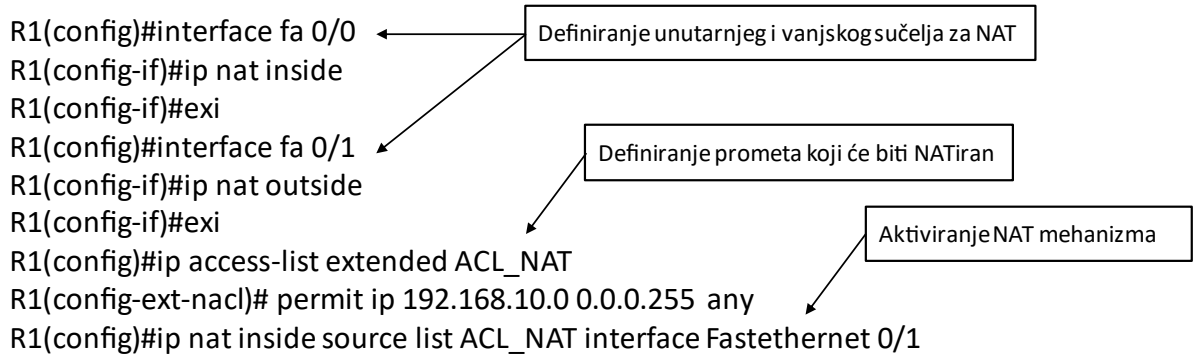
Postoji nekoliko načina konfiguracije NAT-a, pa su to čak i različite njegove vrste. U priručniku se obrađuje najkorištenija, a to je dinamički NAT. U njemu se više IP

adresa računala u lokalnoj mreži prevodi obično u jednu javnu IP adresu ili određeni raspon javnih IP adresa (engl. *Pool*).



Slika 181. Princip rada NAT mehanizma (dinamički NAT)

U klasifikaciji NAT prometa dovoljno je koristiti standardne liste za kontrolu pristupa za određivanje mreža koje će ući u NAT, međutim to u praksi nije dobar pristup, jer osim NAT mehanizma, postoje i razni drugi, kao što je IPSec VPN, tako da je uvijek bolje koristiti proširene liste. Njih svakako treba koristiti u slučaju da uz NAT, ima i VPN tunela, zato što promet koji ide u VPN tunel, ne smije ići u NAT, a za to je bitno i odredište prometa. Na slici 182. prikazane su sve potrebne naredbe za osnovnu konfiguraciju NAT mehanizma.



Slika 182. Konfiguracija NAT mehanizma

Naredbom „show ip nat translations“ provjerava se radi li NAT mehanizma očekivano. Za bilo kakav ispis prije upisivanja te naredbe nužno je stvoriti promet između privatne mreže i javnog adresnog prostora.

3.6. Primjena lista za kontrolu pristupa u konfiguracija IPSec VPN tunela

Protokol IPSec služi za zaštitu podataka u tranzitu. To je uobičajen način povezivanja udaljenih lokacija ili udaljenih korisnika s centralnom lokacijom tvrtke. Također uobičajeno je da se tvrtke međusobno povezuju korištenjem mehanizma IPSec, koji koristi javni internet za povezivanje, što ovo rješenje čini jeftinim i vrlo fleksibilnim. Uz navedeno, IPSec je idealan za borbu protiv lažiranja ARP-a i DNS-a (engl. *Spoofing*) i protiv većine napada presretanjem komunikacije (engl. *Man in the Middle* – MitM). Točnije, MitM napadi nisu izvedivi ako se na mreži koristi protokol IPSec. Kako ima prednosti, tako ima i mana, a jedna od najvećih dodatno je opterećenje koje se postavlja na mrežu (paketi su nešto veći zbog enkripcije) kao i opterećenje na procesor (enkripcija/dekripcija). Ako se planira uvođenje protokola IPSec na razini organizacije, preporučuje se potražiti mrežne adaptere koji imaju procesora za enkripciju/dekripciju protokola IPSec kako bi se rasteretio procesor računala (ovo se pogotovo odnosi na poslužitelje). Bez obzira na mane, prednosti su daleko veće.

IPSec (engl. *Internet Protocol Security*) skupina je protokola kojima je glavna namjena zaštititi IP komunikaciju između dva računala autentikacijom i enkripcijom svakog paketa koji se između njih razmjenjuje. Protokoli koji čine IPSec zajedno omogućuju razmjenu ključeva za sigurnu enkripciju podataka, zatim integritet, tajnost i neporecivost podataka, obostranu autentikaciju te zaštitu od zlućudnih odgovora (engl. *Antireplay*). Osnovni su protokoli koji omogućuju sve navedeno: *Security Association* (SA), *Authentication Header* (AH) i *Encapsulates Security Payload* (ESP).

Neki su od kriptografskih algoritama koji koristi IPSec:

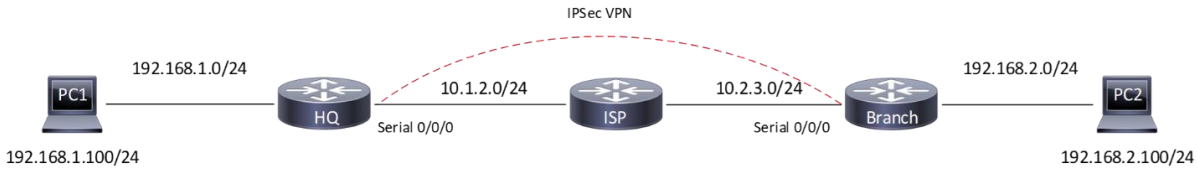
- HMAC-SHA1, koji se koristi za integritet i autentičnost podataka, tj. digitalni potpis te
- TripleDES-CBC ili AES-CBC koji se koriste za povjerljivost podataka, tj. enkripciju.

Osim spomenutih, ima i drugih algoritama koji se mogu koristiti, no koji god bio odabran, obje ga strane u komunikaciji moraju podržavati i biti konfigurirane za njegovo korištenje.

Protokol IPSec može raditi u transportnom ili tunelskom načinu rada. Svaki od njih ima svoje prednosti i mane te, naravno, primjenu. U ovom priručniku u središtu je tunelski način, najviše i korišten u praksi.

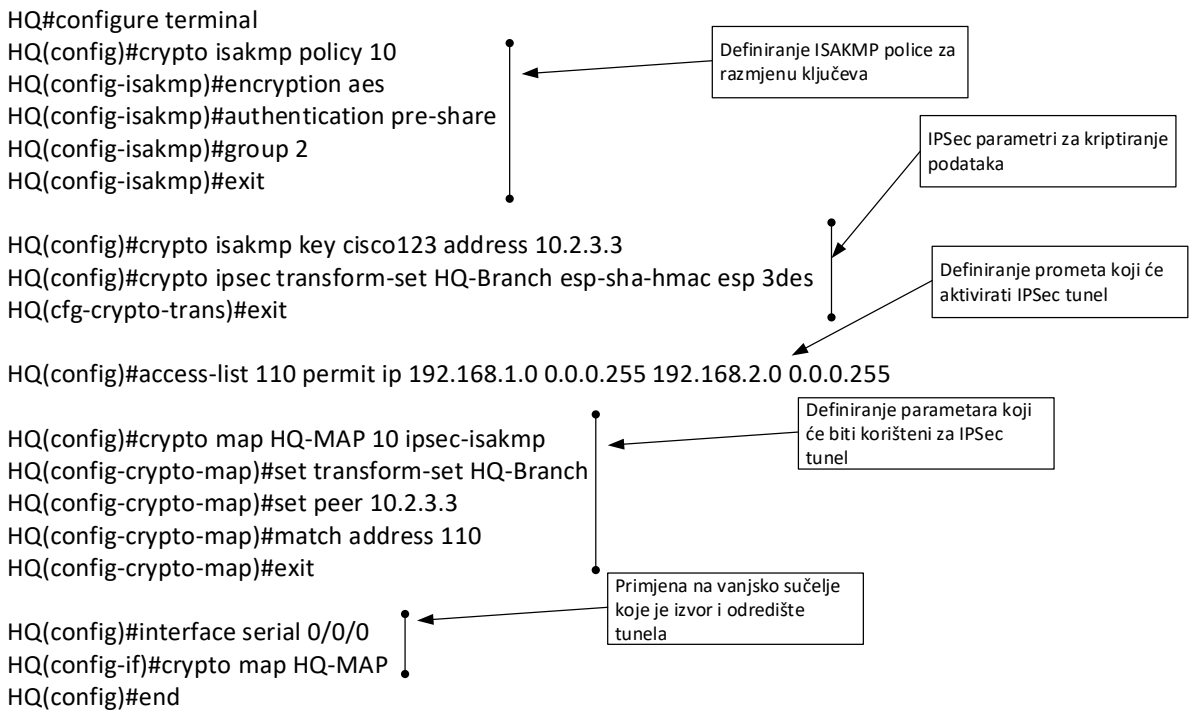
3.7. Konfiguracija IPSec S2S tunela

Prilikom konfiguracije S2S IPSec VPN tunela koji je danas neizostavni dio svake mreže, treba paziti na nekoliko ključnih elemenata. Za objašnjenje služi topologija prikazana na slici 183.



Slika 183. Ilustracija primjene IPSec tunela

Naredbe koje su potrebne za uspostavu IPSec tunela prikazane su na slici 184. Nužno je zapamtiti da lista za kontrolu pristupa na drugoj strani mora biti zrcalno simetrična. Također, ako se u mreži koristi NAT, što je danas uobičajeno, treba paziti da promet koji je namijenjen za IPSec tunel bude zabranjen u listi za kontrolu pristupa koja se koristi za NAT. Bitno je i konfigurirati statičku putanju (obično je to predefinirana putanja, engl. *Default*) koja usmjerava promet za odredišnu privatnu mrežu sučeljem na kojem je *Crypto Map*.



Slika 184. Naredbe za konfiguriranje IPSec tunela

Za potrebe istovremenog konfiguriranja NAT i IPSec mehanizma bilo bi dobro koristiti listu za kontrolu pristupa za NAT koja će uvijek sav promet koji bi trebao ići u IPSec VPN tunele zabraniti u NAT mehanizmu. Ta „univerzalna“ lista za kontrolu pristupa je prikazana na slici 185.

```
HQ(config)#ip access-list extended ACL_NAT
HQ(config-ext-nacl)# deny ip any 10.0.0.0 0.255.255.255
HQ(config-ext-nacl)# deny ip any 172.16.0.0 0.15.255.255
HQ(config-ext-nacl)# deny ip any 192.168.0.0 0.0.255.255
HQ(config-ext-nacl)# permit ip any any
```

Slika 185. Univerzalni ACL za NAT uz IPSec VPN tunele

3.8. Sigurnost pristupa L2 računalnoj mreži

Iako se većina ciljanih zlonamjernih napada očekuje izvan okruženja lokalne mrežne tvrtke, nije neuobičajeno da napadi na njezinu mrežnu infrastrukturu dolaze iznutra. Takvi napadi mogu biti zlonamjerni, ali često su slučajni i nemaju namjeru naštetiti. Bez obzira na to jesu li namjerni ili nenamjerni, inicirani iznutra ili izvana, potrebno je u što većoj mjeri prevenirati takve napade. U ovom poglavlju bit će obrađeni neki od

najčešćih napada na L2 mrežnu infrastrukturu neke tvrtke. Iako ti napadi mogu imati devastirajuć učinak na poslovanje, relativno ih je lako spriječiti.

3.8.1. Mehanizam Port Security

Preplavljivanje MAC tablica (engl. *MAC Flooding Attack*) napad je na L2 uređaje koji čine mrežnu infrastrukturu tvrtke, i to korištenjem uobičajenog ograničenja koje ima svaki uređaj, a to su njegovi resursi. L2 preklopnici funkcioniraju u najvećoj mjeri tako što pamte MAC adrese drugih uređaja koji su spojeni na mrežu. Obično je to nekoliko tisuća do nekoliko desetaka tisuća MAC adresa koje današnji preklopnici mogu zapamtiti. Dakle, ako napadač generira velik broj MAC adresa, u nekom trenutku preklopnici u mreži neće više moći zapamtiti nove MAC adrese, a to znači polaganu „smrt“ L2 mreže. Zašto polaganu „smrt“? Zato što preklopnici imaju vremenski period u kojem će pamtiti MAC adresu uređaja koji ne komunicira na mreži. Nakon tog vremena, MAC adrese uređaja koji više ne komuniciraju na mreži (isključeni su ili odvojeni od mreže), bit će obrisane iz MAC tablice preklopnika, a umjesto njih će preklopnik zapamtiti MAC adrese koje generira napadač. Kada se legitimno računalo opet spoji na mrežu, preklopnici više nemaju memorije za njegovu MAC adresu, tako da mreža za njih više ne postoji, a to se odražava negativno na poslovanje tvrtke. Odgovor je na taj napad mehanizam *Port Security*.

Port Security je mehanizam koji se koristi za ograničavanje pristupa mreži uređajima na temelju njihovih MAC adresa. To može biti vezano za konkretnu MAC adresu ili jednostavno broj MAC adresa koje će biti dozvoljene za pamćenje preklopnicima po pojedinom sučelju ili VLAN-u. Prilikom konfiguracije mehanizma *Port Security* može se odabrati jedan od tri načina na koji preklopnik može učiti MAC adrese uređaja u mreži: dinamički, statički i ljepljiv (engl. *Sticky*).

- Dinamički: to je predefiniran način učenja MAC adresa na preklopnicima. U ovom slučaju preklopnik će dopustiti komunikaciju svim uređajima dok ne dođe do maksimalno dozvoljenog broja MAC adresa. MAC adrese se nakon unaprijed definiranog vremena (300 sekundi) brišu i tako se omogućava povezivanje novih uređaja. Također, ako se preklopnik ponovno pokrene, sve MAC adrese se brišu iz memorije. Vrijeme nakon kojega preklopnik briše MAC adrese uređaja koji više ne komuniciraju na mreži, moguće je konfigurirati u rasponu od 0 do 1440 minuta. Ako

je konfigurirana vrijednost 0, uređaj neće nikada obrisati MAC adresu. Osim što se može mijenjati vrijeme, može se mijenjati i način na koji se to vrijeme računa. U načinu *Inactivity* preklopnik računa vrijeme od zadnjeg primljenog okvira, a ako se konfigurira *Absolute* (što je unaprijed definirano), preklopnik računa vrijeme od prvog okvira koji je primio od nekog uređaja.

- Statički: ručno se konfiguriraju MAC adrese kojima je dozvoljen pristup određenom sučelju, a konfiguracija ostaje na sučelju dok administrator ručno ne izbriše neku MAC adresu iz konfiguracije ili ne konfigurira konkretno sučelje kao L3 sučelje.
- Ljepljivo (engl. *Sticky*): preklopnik se ponaša slično kao i u dinamičkom načina rada, ali sada MAC adrese sprema u „running-config“ i pohrani li se „running-config“ u „startup-config“, prilikom ponovnog pokretanja preklopnika tako naučene MAC adrese neće se obrisati, već ostaju trajno. Mogu se vidjeti naredbom „show running-config“ ili „show startup-config“. Ako je omogućen „Sticky“ način učenja MAC adresa, preklopnik više ne koristi dinamički način. Ne postoji vremensko ograničenje MAC adresa naučenih „Sticky“ načinom.

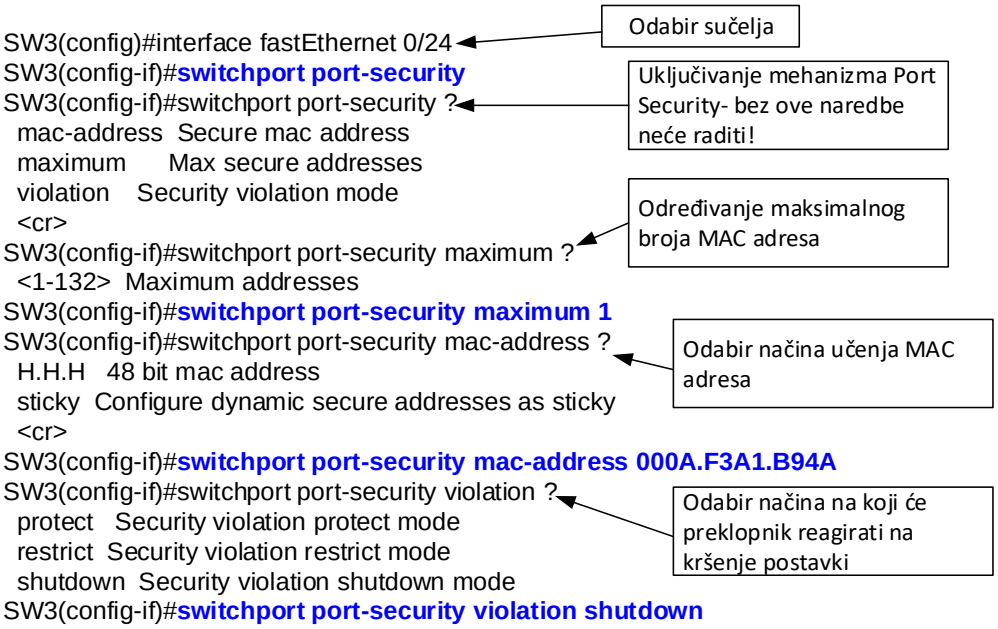
Kada se želi konfigurirati maksimalan broj MAC adresa koje preklopnik može pamtiti, treba uzeti u obzir maksimum samoga modela uređaja i njegove karakteristike, i taj se maksimum, naravno, ne može mijenjati. Unaprijed definirana vrijednost maksimuma za sučelja kada se uključi *Port Security* mehanizam 1 je MAC adresa, i promjenjiva je prilikom konfiguracije *Port Security* na tom sučelju. *Port Security* neće raditi ako se prije konfiguracije ne uključi na željenim sučeljima naredbom „switchport port-security“.

Preklopnik će reagirati na povredu postavki mehanizma *Port Security* ako se na sučelju gdje je uključen mehanizam pojavi više MAC adresa nego što je dozvoljeno konfiguracijom na tom sučelju, ako se na sučelju pojavi MAC adresa koja nije na listi dozvoljenih MAC adresa ili ako se ulazni promet s dozvoljenom MAC adresom na jednom sučelju pojavi na nekom drugom sučelju koje se nalazi u istom VLAN-u (engl. *MAC Move Violation*). Postoje i neki drugi uvjeti, ali nisu dio ovog priručnika Tri su načina rada u kojima mehanizam *Port Security* može reagirati na kršenje postavki u konfiguraciji.

- Zaštiti (engl. *Protect*): ako se dogodi kršenje konfiguracije, preklopnik nakon toga ne prihvaća nove MAC adrese na sučelje i odbacuje sav dolazni promet s nesigurnih MAC adresa, ali ne povećava „Security Violation Counter“ (vidljivo

naredbom „show portsecurity“) i ne generira nikakvu obavijest da se kršenje postavki dogodilo. Taj je način rada najmanje agresivan i načelno ga ima smisla koristiti u kombinaciji s dinamičkim načinom učenja MAC adresa, kada je sučelje u javno dostupnom prostoru gdje se povezuje mnogo uređaja u relativno kratkom vremenu.

- Ograniči (engl. *Restrict*): u slučaju kršenja postavljenih pravila, preklopnik odbacuje sav promet s nesigurnih MAC adresa, ne uči nove MAC adrese, povećava „Security Violation Counter“, generira *Syslog* poruku i *SNMP trap*.
- Isključi (engl. *Shutdown*): to je unaprijed definiran način rada koji, ako dođe do kršenja postavki, stavlja sučelje u stanje „Error-Disabled“, što je vidljivo i na samom fizičkom uređaju gdje svjetla na sučelju više ne svijetle. U tom načinu rada preklopnik povećava „Security Violation Counter“, generira *Syslog* poruku i *SNMP trap*. Sučelje se mora ručno resetirati na način da ga se prvo isključi naredbom „shutdown“, a zatim uključi naredbom „no shutdown“. Na slici 186. prikazan je primjer konfiguracije *Port Security* mehanizma.

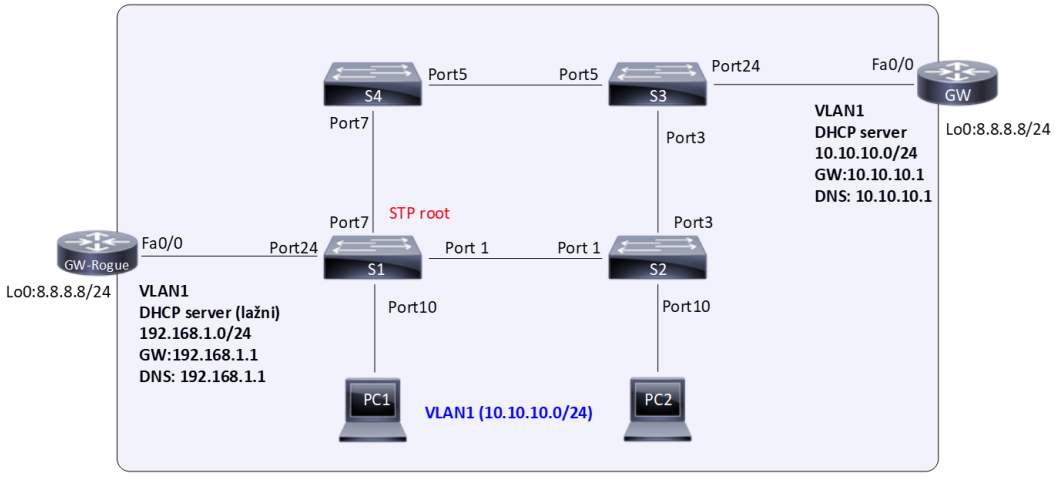


Slika 186. Primjer konfiguracije Port Security mehanizma

Za provjeru konfiguracije može se koristiti naredbe „show port-security“ ili, za pojedino sučelje, „show portsecurity interface fastethernet 0/24“.

3.8.2. DHCP snooping

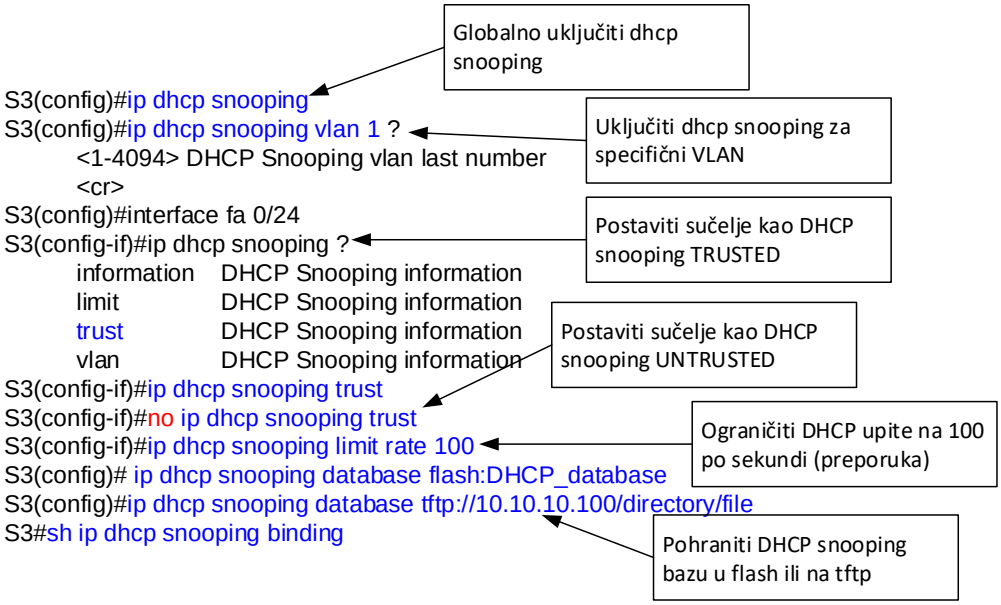
DHCP je vrlo važan servis u modernim mrežama sastavljenim od velikog broja uređaja koji imaju potrebu komunicirati računalnom mrežom. Zbog važnosti DHCP-a za ispravno funkcioniranje mreže nužno je osigurati mrežu od lažnih (engl. *Rogue*) DHCP poslužitelja. U DHCP lažiranju (engl. *Spoofing*) napadač se predstavlja kao legitimni DHCP poslužitelj i odgovara na DHCP zahtjeve uređaja u mreži nudeći im lažne IP postavke kako bi izvršio napad presretanjem komunikacije (engl. *Man-in-the-Middle*) ili jednostavno uskratio pristup mreži. Zbog prirode DHCP mehanizma to je lako izvesti u nezaštićenoj mreži. Naime, uređaji koji trebaju IP postavke prihvatit će ih od bilo kojeg DHCP poslužitelja (najčešće je to onaj koji prvi odgovori), bez obzira na to bio on legitimni ili lažni. Na slici 187. vidljivo je da je zbog topologije STP-a (pretpostavka je da je veza između S2 i S3 blokirana) računalima bliži lažni DHCP poslužitelj, tako da će umjesto postavki za mrežu od uređaja GW, dobiti potpuno druge IP postavke od uređaja GW-*Rogue*. Nakon toga računala više nisu dio legitimne mreže, već sve upite prema DNS poslužitelju zapravo šalju prema napadaču koji s tim upitima može raditi što god želi. Zbog toga je nužno pripremiti računalnu mrežu za odgovor legitimnim DHCP poslužiteljima na DHCP zahtjeve uređaja u mreži.



Slika 187. Prikaz topologija s lažnim DHCP poslužiteljem

Mehanizam koji štiti mrežu od lažnih DHCP poslužitelja naziva se *DHCP Snooping* i konfigurira se na preklopnicima za svaki VLAN posebno. Taj se mehanizam zasniva na određivanju sučelja na preklopnicima u mreži na koja će biti dopušteno preklopniku prihvatiti odgovore na DHCP zahtjeve klijentskih uređaja (uređaji koji trebaju IP postavke). Uloge koje sučelja u kontekstu mehanizma *DHCP Snooping* mogu imati su:

„TRUSTED“ i „UNTRUSTED“. Nakon odluke gdje će u mreži biti spojen legitimni DHCP poslužitelj, to sučelje će se konfigurirati kao „TRUSTED“ i jedino će na takvo sučelje preklopnik prihvatiti sve DHCP poruke koje šalje poslužitelj. Na sva ostala sučelja gdje nije spojen legitimni DHCP poslužitelj, a pretpostavka je da će to biti samo klijentska računala i da se ne smije pojaviti drugi DHCP poslužitelj (lažni), konfigurirat će se kao „UNTRUSTED“. Stanje svih sučelja, kada se koristi mehanizam *DHCP Snooping* unaprijed je definirano kao „UNTRUSTED“. Također, sučelja između preklopnika konfigurirat će se kao „TRUSTED“. Na slici 188. prikazan je primjer konfiguracije *DHCP Snooping* mehanizma na jednom preklopniku.

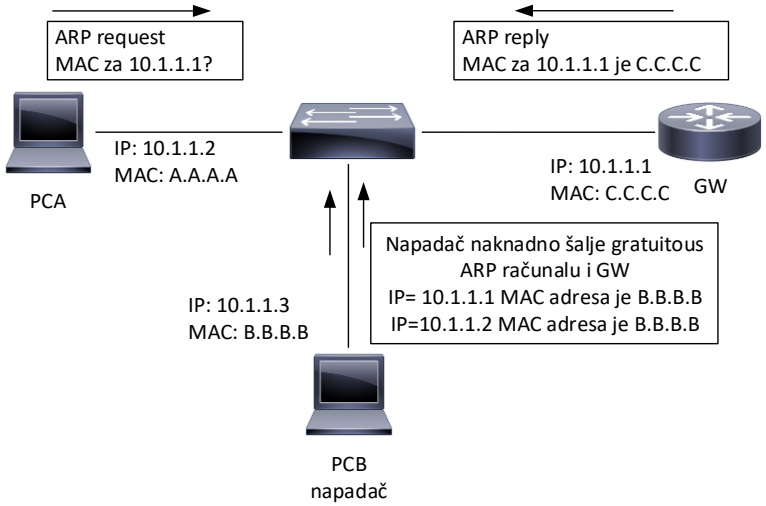


Slika 188. Primjer konfiguracije DHCP Snooping mehanizma

3.8.3. Mehanizam zaštite od lažnih ARP poruka

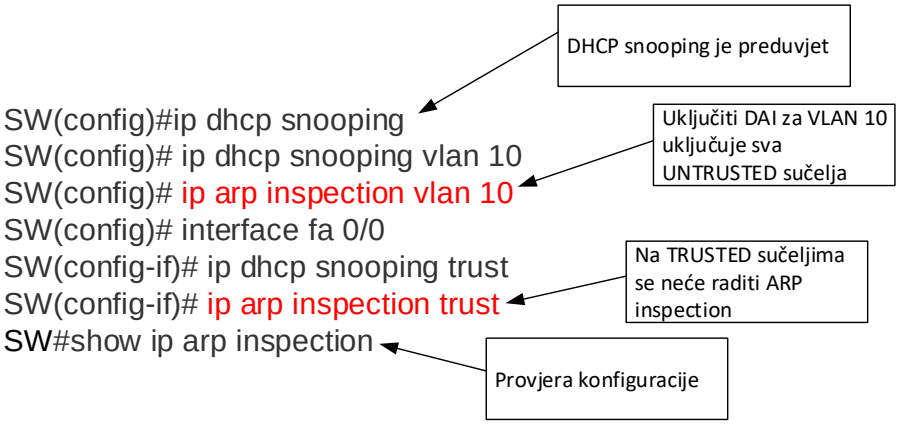
Cilj je napada lažiranjem ARP zahtjeva (engl. *ARP Spoofing*) učiniti mrežu nestabilnom i nefunkcionalnom i/ili preusmjeriti promet prema napadačkom računalu (engl. *Man-in-the-Middle Attack*). Ovdje se koristi uobičajen način funkcioniranja L2 lokalne mreže, a to je mehanizam ARP kojim računala u lokalnoj mreži otkrivaju MAC adrese ostalih računala u toj lokalnoj mreži kako bi s njima mogla komunicirati. Kada računalo sazna MAC adresu, npr. GW-a, privremeno pohrani tu MAC adresu u ARP tablicu. ARP zahtjevi su sveodredišni promet, i svaki takav vide sva računala u lokalnoj mreži. Ako napadač ima pristup lokalnoj mreži, može odgovoriti na svaki APR zahtjev umjesto legitimnog računala, a to znači da može reći ostalim računalima u mreži da je on GW

(što nije poželjno). Isti učinak može postići slanjem tzv. „gratuitous ARP“ koji se koristi kada neki uređaj želi sve ostale obavijestiti da za njega promijene MAC-IP zapis u svojim tablicama. Na taj način opet preusmjerava računala u lokalnoj mreži na sebe.



Slika 189. Princip ARP Spoofing napada

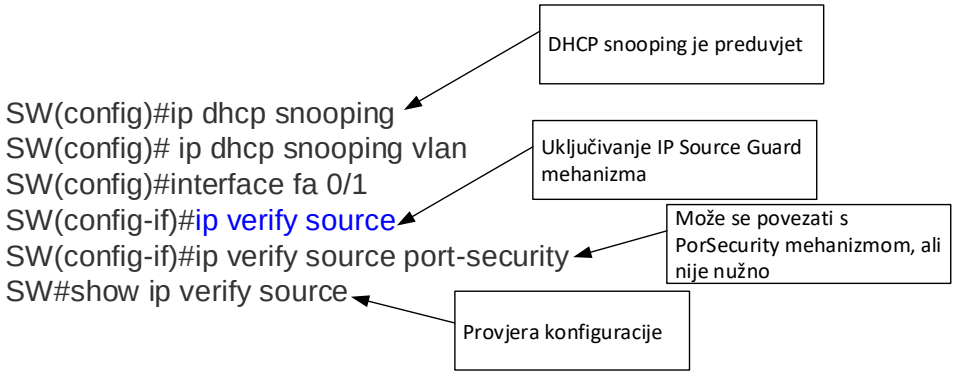
Mehanizam za zaštitu od lažiranja ARP zahtjeva dinamičko je pregledavanje ARP poruka (engl. *Dynamic ARP inspection — DAI*). Mehanizam DAI kao osnovu zahtijeva uključen mehanizam *DHCP Snooping* zato što se oslanja na tu bazu (mora imati nekakvu bazu) koja je prethodno izgrađena (iako može koristiti i statičke zapise i koristiti ARP ACL). DAI funkcionira na svim „UNTRUSTED“ sučeljima gdje pregledava sve ARP zahtjeve i ARP odgovore. Svaki se ARP paket pregledava i utvrđuje odgovara li IP-MAC zapis pošiljatelja u ARP paketu onome u bazi (*DHCP Snooping* bazi). Konfiguracija DAI mehanizma prikazana je na slici 190.



Slika 190. Konfiguracija mehanizma za zaštitu od lažnih ARP poruka

3.8.4. Mehanizam zaštite od lažiranja IP adresa

Lažiranje IP adresa još je jedan iz grupe napada lažiranjem (engl. *Spoofing*) kojima je cilj preoteti komunikaciju (engl. *Man-in-the-Middle Attack*) ili učiniti mrežu nedostupnom (engl. *Denial of Service — DoS*). U ovom je slučaju ideja preuzimanje IP adrese legitimnog uređaja i pokušaj ostvarivanja komunikacije koja je inače moguća samo uređaju koji je konfiguriran s određenom IP adresom. Zaštita od ovog napada je mehanizam *IP Source Guard* na preklopnicima. Taj mehanizam, isto kao i mehanizam DAI, oslanja se na bazu koju je izgradio mehanizam *DHCP Snooping*. *IP Source Guard* funkcionira tako što se na svim L2 sučeljima na kojima je omogućen, provjerava promet koji ulazi u sučelje i uspoređuje se IP i MAC adresa u paketu s onom koja se nalazi u *DHCP Snooping* bazi. Ako se IP-MAC adresa u paketu koji ulazi u sučelje ne podudara s IP-MAC adresom zapisanom u bazi za to sučelje, taj promet se odbacuje. Kada je omogućen mehanizam *IP Source Guard*, preklopnik na svakom sučelju stvara dinamičke VLAN liste za kontrolu pristupa (engl. *Per-port VLAN ACL-PVACL*) i tako filtrira sav promet koji ne odgovara *DHCP Snooping* zapisima. Na slici 191. prikazani su koraci konfiguracije.



Slika 191. Primjer konfiguracije mehanizma za zaštitu od lažiranja IP adresa

3.9. Zaštita stabilnosti topologije *Spanning Tree* Protokola (STP-a)

Kao što je ranije u priručniku rečeno, STP mehanizam je ključan za stabilnost L2 logičke mrežne topologije za svaki VLAN u mreži. Bez STP protokola mreže jednostavno ne bi mogle funkcionirati. Da bi STP logička topologija bila stabilna, nije dovoljno samo odrediti koji je preklopnik vršni preklopnik za pojedini VLAN, već je nužno osigurati da se u početku definirana topologija ne može narušiti. To je vrlo važno jer kada bi neki napadač imao mogućnost utjecati na promjenu u STP topologiji tako

da, na primjer, poveže preklopnik koji ima bolji prioritet od trenutnog vršnog preklopnika, cijela mreža bi se našla u stanju ponovnog izračunavanja i odabira vršnog preklopnika. U tom periodu mreža bi bila neupotrebljiva. Rezultat može biti neoptimalno preklapanje mrežnog prometa, ali i potpuno nedostupna mreža ako bi napadač odlučio držati je u stalnom stanju konvergencije mijenjanjem prioriteta svog preklopnika tako da jedan čas ima najbolji (najmanji) prioritet, a već drugi čas ga promijeni u lošiji (veći) od uobičajenog vršnog preklopnika u mreži.

3.9.1. Mehanizam *BPDU filtering*

Ta se funkcija može uključiti u cjelini, na preklopniku, ili za svako sučelje posebno. Na općoj razini koristi se naredba „spanning-tree portfast bpdufilter default“ koja sprječava da sučelja konfigurirana kao *PortFast* šalju i primaju BPDU pakete. Iako je slanje i primanje onemogućeno, sučelje ipak pošalje nekoliko BPDU paketa prilikom uključivanja, prije nego što ih preklopnik počne filtrirati. Ako sučelje ipak primi BPDU paket, ono gubi svoj *PortFast* status i BPDU filtriranje se isključuje. To znači da se sučelje vraća u unaprijed definirano stanje i obrađuje BPDU pakete normalno. BPDU filtriranje se može uključiti i na sučelju naredbom „spanning-tree bpdufilter enable“ bez uključivanja mehanizma *PortFast* na sučelju. Ta naredba sprječava sučelje da šalje i prima BPDU pakete. Treba biti pažljiv jer naredba na sučelju ima isti učinak kao isključenje protokola STP, i to može uzrokovati petlje u mreži.

3.9.2. Mehanizam *BPDU Guard*

Kada se uključi funkcija *BPDU Guard* na općoj razini naredbom „spanning-tree portfast bpduguard default“, to se odnosi na sva sučelja koja imaju uključen *PortFast*. STP se i dalje koristi na tim sučeljima koja će biti u stanju „up“ dok god ne prime BPDU paket. Treba imati na umu da u dobro složenoj mreži *PortFast* sučelja ne bi trebala primiti BPDU pakete. Ako se to dogodi, to znači da mreža nije dobro složena ili je spojen uređaj koji ne bi trebao biti u takvo sučelje i treba ispraviti pogrešku. *BPDU Guard* će, ako sučelje primi BPDU paket, staviti ga u stanje „error-disabled“, dakle, isključeno sučelje koje administrator mora ručno ponovno pokrenuti naredbama „shutdown“, a zatim „no shutdown“. To se obično koristi u konfiguraciji telekomunikacijskih mreža. Osim na općoj razini, funkcija se može uključiti za pojedina sučelja naredbom u sučelju „spanning-tree bpduguard enable“ i istovremenim uključivanjem funkcije *PortFast*.

Kada sučelje primi BPDU, poprima stanje „error-disabled“. *BPDU Guard* može se konfigurirati ako se koristi PVST+, RSTP ili MSTP.

3.9.3. Mehanizam *ROOT Guard*

U okruženjima telekomunikacijskih mreža, gdje se mnogo korisnika spaja na jednu L2 mrežu, postoji mogućnost da korisnički preklopnik preuzme ulogu vršnog preklopnika. Osim toga, zbog različito konfiguriranih preklopnika i promjena u mreži, ona bi mogla pasti u stanje neprestane konvergencije, što znači da nije funkcionalna. Da se to ne bi dogodilo, koristi se funkcija *Root Guard* koju će operater konfigurirati na svim sučeljima prema korisničkim mrežama. U slučaju da telekomunikacijski preklopnik na sučelje primi BPDU paket s informacijama da je vršni preklopnik u mreži korisnika, to sučelje stavlja u blokirano stanje koje se naziva „root-inconsistent state“. Konfigurira se naredbom na sučelju „spanning-tree guard root“.

4. Dizajn računalne mreže

U OVOJ CJELINI NAUČIT ĆETE:

- zašto je faza dizajna važna u računalnim mrežama
- što utječe na dizajn računalne mreže
- koja su ograničenja dizajna računalne mreže
- kako izgleda hijerarhijski dizajn računalne mreže
- koja je uloga pojedinog sloja u hijerarhijskom dizajnu računalne mreže.

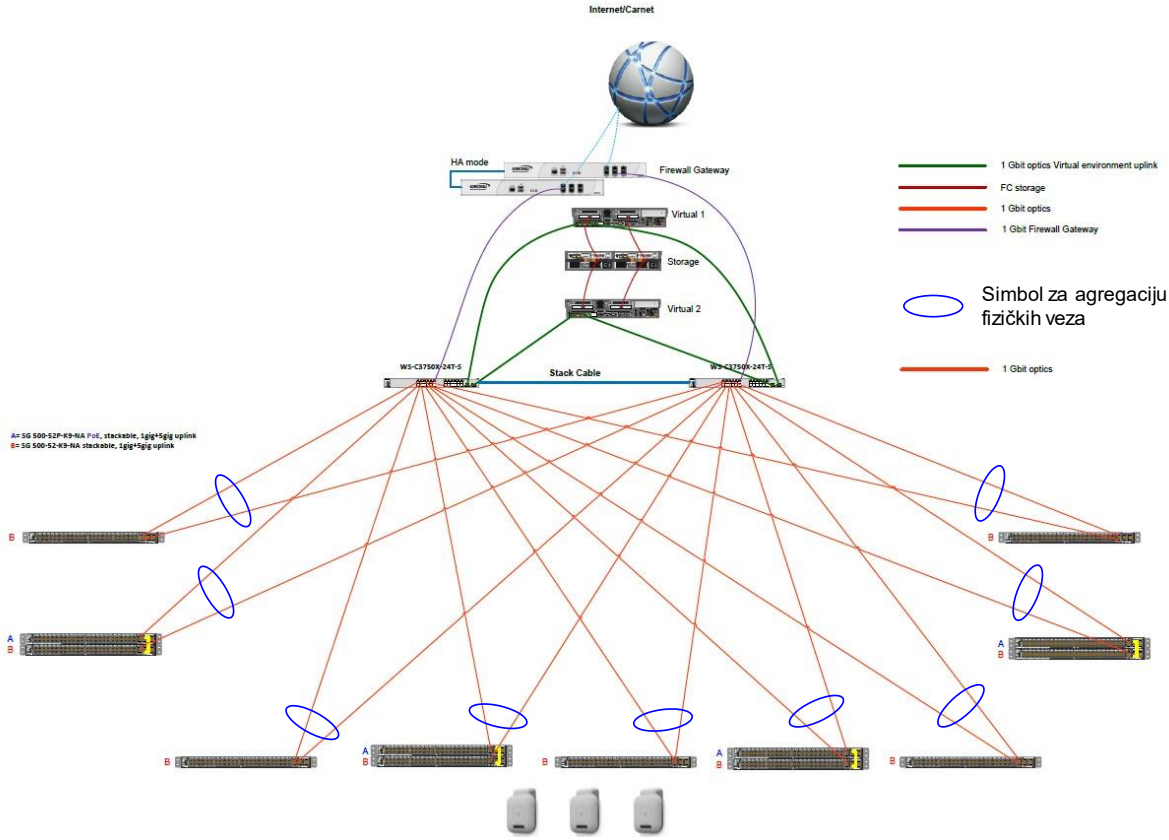
Jedna je od ključnih osobina IKT sustava da nisu svrha samima sebi, već biti podrška poslovnim procesima i osnovnom poslovanju tvrtke kojoj „služe“. To podrazumijeva da inženjeri u dizajnu IKT sustava moraju najprije razmišljati o poslovanju organizacije, a tek onda o tehnologijama koje mogu iskoristiti kako bi IKT sustav koji će dizajnirati, a zatim i implementirati, mogao adekvatno podržati poslovne procese organizacije. Takav je način promišljanja puno apstraktniji od konfiguriranja mrežnih i drugih uređaja, ali neophodno ga je prepoznati, razumjeti, a zatim i prihvatiti.

Područje dizajna IKT sustava vrlo je široko, a danas postaje interdisciplinarno jer se isprepliće promišljanje o sigurnosti, skalabilnosti, visokoj dostupnosti, interoperabilnosti i drugim karakteristikama, ne samo računalne mreže, već IKT sustava u cjelini. To podrazumijeva okruženje računarstva u oblaku, poslužiteljske infrastrukture, sustav za (sigurnosnu) pohranu podataka, jednostavnost održavanja, sustave podatkovnih centara, zakonodavni i regulatorni okvir i slično. Postoji mnogo pristupa u dizajnu neke računalne mreže ili nekog IKT sustava kao cjeline, a ovisno je

o mnogo faktora, između ostalog i o financijskoj raspoloživosti za ulaganje u IKT infrastrukturu. U ovom priručniku u središtu je troslojni hijerarhijski model računalne mreže koji omogućava jasno strukturiranje mrežne infrastrukture, kao i relativno jednostavnu „implementaciju“ skalabilnosti, sigurnosti i redundancije u računalnoj mreži uz optimalne troškove i jednostavnost održavanja infrastrukture. Navedene karakteristike nije moguće konfigurirati „jednim klikom“, već su one rezultat dizajna i mogu se nazvati „pojavljujućim svojstvima“ sustava (engl. *Emergent Properties*), slično kao što se može reći da je svijest pojavljujuće svojstvo materije koja je organizirana na određeni način. Pojavljujuća svojstva proizlaze iz interakcije više složenih komponenti unutar sustava, i ni jedna od tih komponenti nema sama svojstvo u cijelosti. Skalabilnost je očiti primjer pojavljujućeg svojstva računalne mreže u troslojnoj hijerarhiji.

4.1. Hijerarhijski dizajn računalne mreže

Standardni pristup je troslojni hijerarhijski model računalne mreže, koji se sastoji od pristupnog sloja (engl. *Access Layer*), distribucijskog sloja (engl. *Distribution Layer*) i sloja jezgre (engl. *Core Layer*). Takav pristup u dizajnu računalnih mreža omogućava jasnu modularnost i jednostavnu skalabilnost uz predvidiv trošak implementacije i održavanja mrežne infrastrukture. Ovisno o količini krajnjih uređaja koji se povezuju na računalnu mrežu, ponekad nije potreban troslojni dizajn, već se može koristiti tzv. dizajn urušene jezgre (engl. *Collapsed Core*), gdje su jezgreni i distribucijski sloj zapravo jedan sloj. To je slučaj u mrežama manjih organizacija, kao što su manje bolnice ili škole, koje imaju manje od 50-ak preklopnika pristupnog sloja na koje se povezuju krajnji uređaji. Ne postoji precizan broj preklopnika, već će to ovisiti o potrebama organizacije s obzirom na funkcionalnost mreže, potrebu za širenjem u budućnosti, raspoložive resurse i procjenu inženjera koji dizajniraju mrežu. Ima i drugih faktora koji utječu na odabir dizajna, ali to su važniji. Dizajn računalne mreže nije egzaktna znanost, već proces višestrukih konzultacija s korisnikom i rafiniranja prijedloga do prihvaćanja konačnog rješenja koje korisniku odgovara s obzirom na potrebe poslovanja i prihvatljiv financijski trošak implementacije i održavanja takvog dizajna. Trošak održavanja implementirane infrastrukture ponekad može biti odlučujući prilikom odabira tehnologije koja će se koristiti u računalnoj mreži. U dizajnu računalne mreže, dakle, prisutna je i ekonomija.



Slika 192. Primjer dizajna hijerarhijske računalna mreže u izvedbi „Collapsed Core“ — iz stvarne dokumentacije

Sloj jezgre računalne mreže treba veću propusnost kako bi omogućio brzu komunikaciju između uređaja unutar računalne mreže. Uređaji koji se koriste u sloju jezgre obično su vrlo brzi preklopnici koji obično koriste optička sučelja, tehnologije za visoku dostupnost poput višestrukih napajanja, mogućnosti rada u preklopničkom stogu (engl. *Switch Stack*), mogućnost zamjene ventilatora i napajanja u radu i slične napredne funkcionalnosti koje omogućuju otkaze i popravke na sloju jezgre bez gubitka komunikacije. Samim time ovakvi uređaji su znatno skuplji od „običnih“ preklopnika. Jezgra računalne mreže, osim što koristi uređaje s naprednim mogućnostima, dizajnirana je tako da ne postoji jedna točka otkaza (engl. *Single Point of Failure*), što se postiže višestrukim fizičkim vezama i prikladnom logičkom konfiguracijom koja se neprimjetno prilagođava otkazima, poput konfiguriranja preklopnika za rad u stogu (engl. *Stack*).

Distribucijski sloj računalne mreže koristi se u svim većim računalnim mrežama gdje je nužno koristiti posebne preklopnike kako bi se veliki broj preklopnika pristupnog sloja mogao povezati na sloj jezgre. Na primjer, veliki studentski kampusi koji imaju više od

tisuću preklopnika pristupnog sloja, moraju koristiti distribucijski sloj jer nije moguće povezati tolik broj pristupnih preklopnika direktno na preklopnike sloja jezgre. Uloga distribucijskog sloja može varirati od isključivo uloge sloja koji povezuje pristupne i preklopnike sloja jezgre, što je čest slučaj, do sloja koji obavlja funkcije poput usmjeravanja prometa između VLAN-ova, redistribuira putanje između protokola usmjeravanja, ima određenu razinu sigurnosti korištenjem lista za kontrolu pristupa i slično. Te funkcionalnosti se u većini mreža ipak obavljaju na uređajima koji su spojeni na sloj jezgre, kao što su usmjernici ili vatrozidi jer je tako jednostavnije i jasnije upravljanje cijelim sustavom.

Pristupni sloj računalne mreže nužan je i bez njega nema o njoj govora. Taj sloj uvijek postoji i karakteriziraju ga uređaji velike gustoće sučelja, obično 24 ili 48, u svrhu omogućavanja pristupa računalnoj mreži što većem broju krajnjih uređaja uz što manji trošak po sučelju. Na uređajima pristupnog sloja postoje i specifične funkcionalnosti, kao što su napajanje uređaja posredstvom *Etherneta* (engl. *Power over Ethernet*) i razni sigurnosni mehanizmi, kao što su *Port Security*, *DHCP Snooping, Dynamic ARP Inspection*, 802.1x provjera identiteta za pristup računalnoj mreži i slično. U nekim slučajevima na ovom se sloju radi i logička segmentacija i usmjeravanje između VLAN-ova, ali to će ovisiti o konkretnom slučaju i što neka organizacija želi postići u svojoj računalnoj mreži.

Troslojna hijerarhijska podjela u dizajnu računalne mreže nije rezervirana samo za mreže koje se baziraju na preklopnicima poznatima kao kampus mrežna infrastruktura (engl. *Campus Network Design*), već i za mreže koje koriste usmjernike za povezivanje brojnih korisničkih lokacija na razini podružnica, prema regionalnim središtima, pa u konačnici prema glavnim čvorištima u podatkovnim centrima na razini države ili šire. U tom slučaju hijerarhija će biti toliko „duboka“ koliko je potrebno kako bi se svi dijelovi računalne mreže povezali u jednu smislenu cjelinu učinkovito i financijski opravdano.

5. Bežična računalna mreža

U OVOJ CJELINI NAUČIT ĆETE:

- što je bežična računalna mreža i koja je uloga bežične računalne mreže
- koji elementi čine bežičnu računalnu mrežu
- kako se odvija komunikacija u bežičnoj računalnoj mreži
- koji su sigurnosni mehanizmi u bežičnoj računalnoj mreži
- koji mehanizmi se koriste za „izbjegavanje sudara“ u bežičnoj računalnoj mreži.

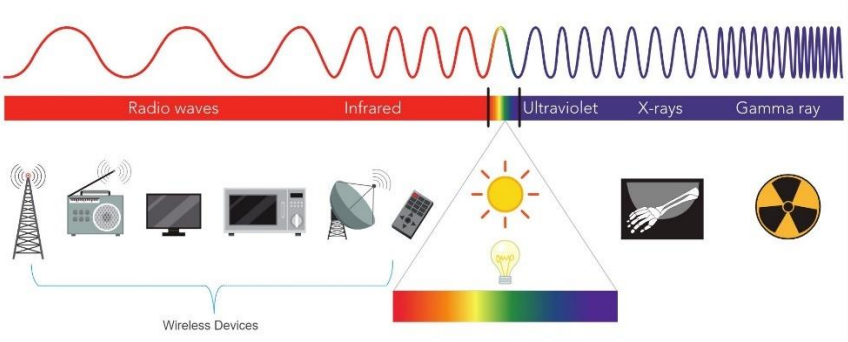
Bežična lokalna računalna mreža (engl. *Wireless Local Area Network — WLAN*) uobičajeno se koristi u poslovnim organizacijama i kućanstvima. Služi za povezivanje prijenosnih računala, tableta, pametnih telefona i drugih uređaja s ostatkom računalne mreže i s internetom. Postoje različite vrste bežičnih računalnih mreža, ali WLAN podrazumijeva tehnologije koje služe kao produžetak žičane mrežne infrastrukture da bi se na nju povezali uređaji koji nemaju mogućnost povezivanja *Ethernet* kabelom. Bežične su računalne mreže vrlo fleksibilne i lako ih je implementirati u vrlo kratkom vremenskom roku bez posebnih zahvata u infrastrukturu zgrade. Postoje različite bežične tehnologije i implementacije bežičnih lokalnih mreža, ali u ovom priručniku u središtu će biti samo WLAN mreže, i to one bazirane na IEEE 802.11 standardu. Taj standard definira kako se radijski frekvencijski spektar koristi za komunikaciju. Specifikacije podrazumijevaju brzine i frekvencije prijenosnog signala, signalizaciju,

modulaciju i druge aspekte bežične komunikacije. Postoje mnoge varijacije IEEE 802.11 standarda, koji se razvija godinama.

IEEE WLAN Standard	Frekvencija signala	Opis
802.11	2.4 GHz	Brzine do 2 Mbps
802.11a	5 GHz	Brzine do 54 Mbps
		Malo područje pokrivanja (domet)
		Slabo prolazi kroz prepreke
		nije kompatibilan s 802.11b i 802.11g standardima
802.11b	2.4 GHz	Brzine do 11 Mbps
		Veći domet od 802.11a
		bolji u prolasku kroz prepreke od 802.11a
802.11g	2.4 GHz	Brzine do 54 Mbps
		Kompatibilan s 802.11b ali sa smanjenom brzinom prijenosa podataka
802.11n	2.4 GHz 5 GHz	Brzina prijenosa podataka od 150 Mbps do 600 Mbps na udaljenost do 70 m
		Bežične pristupne točke i klijenti zahtijevaju više antena i koriste MIMO tehnologiju
		Kompatibilan s 802.11a/b/g ali sa smanjenom brzinom prijenosa podataka
802.11ac	5 GHz	Brzina prijenosa podataka od 450 Mbps do 1.3 Gbps (1300 Mbps) koristeći MIMO tehnologiju
		Podržava do osam antena
		Kompatibilan s 802.11a/n ali sa smanjenom brzinom prijenosa podataka
802.11ax	2.4 GHz 5 GHz	Najnoviji standard objavljen 2019.
		Poznat i kao Wi-Fi 6 ili High-Efficiency Wireless (HEW)
		Omogućava poboljšanu energetsku učinkovitost, više brzine prijenosa podataka, povećan kapacitet i može opsluživati mnogo uređaja istovremeno
		Trenutno radi na 2.4 GHz i 5 GHz, ali je u planu da koristi i frekvencije 1 GHz i 7 GHz kada budu dostupne

Slika 193. Popis IEEE 802.11 standarda

Svi bežični uređaji koriste elektromagnetsko zračenje, i to područje elektromagnetskog spektra poznato kao radiovalovi.



Slika 194. Elektromagnetski spektar; izvor: Freepik; https://www.freepik.com/premium-vector/science-electromagnetic-spectrum-diagram_2589131.htm#query=electromagentic%20spectrum&position=9&from_view=search&track=ais

Postoji nekoliko organizacija koje utječu na to kako će izgledati WLAN standardi, a to su ITU (engl. *International Telecommunication Union*), koji se brine o dodjeli radio-frekvencijskog spektra, IEEE (engl. *Institute of Electrical and Electronics Engineers*), koji se brine o modulaciji signala za prijenos podataka i Wi-Fi savez (engl. *Wi-Fi Alliance*), koji je neprofitna industrijska organizacija s ciljem promocije WLAN

tehnologije i interoperabilnosti uređaja raznih proizvođača te sukladnosti sa standardima i industrijskim normama.

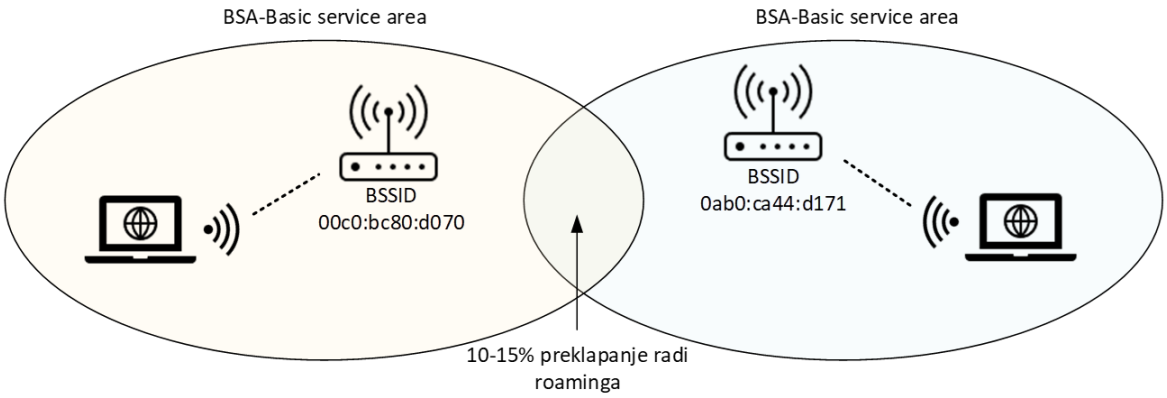
Osnovne komponente bežične lokalne računalne mreže krajnji su uređaji koji imaju bežični primopredajnik, odnosno bežičnu mrežnu karticu (engl. *Wireless Network Interface Card*), bežične pristupne točke (engl. *Wireless Access Points*) i bežične usmjernike (engl. *Wireless Routers*). Naravno, postoje i razne druge komponente, ali u ovom priručniku sagledat će se samo one nužne u implementaciji WLAN mreže kao proširenja žičane računalne mreže. Bežični usmjernici su uobičajeni uređaji u kućanstvima i malim uredima. To su uređaji koje pružatelji telekomunikacijskih usluga daju korisnicima na raspolaganje i ujedno daju pristup internetu. U velikim organizacijama ti se uređaji ne koriste, već se koriste bežične pristupne točke, ponekad i više stotina za potrebe svih bežičnih uređaja u mreži. To je slučaj u velikim studentskim kampusima, sveučilištima, bolnicama i sličnim organizacijama, gdje boravi velik broj ljudi. Bežične pristupne točke mogu biti samostalne (engl. *Standalone Wireless Access Points*), a može njima upravljati centralni uređaj koji se zove upravljač bežičnih pristupnih točaka (engl. *WLAN Controller*). Njime se može istovremeno upravljati s više stotina bežičnih pristupnih točaka, i to s jednog centralnog mjesta.

Uobičajeno je da se bežične pristupne točke povezuju na preklopnike i omogućuju krajnjim korisnicima povezivanje bežičnom mrežom u određeni VLAN na žičanoj infrastrukturi. Tu se otkriva sama svrha bežične mreže — uređajima koji nemaju mogućnost povezivanja kabelom, omogućiti iste funkcionalnosti. Većina bežičnih pristupnih točki u poslovnim okruženjima koristi posebne vanjske antene koje mogu biti različitih vrsta ovisno o potrebama organizacije. Tako postoje svesmjerne antene (engl. *Omnidirectional*) koje odašilju signal u prostor 360º oko antene, usmjerene antene (engl. *Directional*) koje zrače signal u određenom smjeru i MIMO antene (engl. *Multiple Input Multiple Output*) koje, kako im samo ime kaže, koriste skup od više antena za povećanje kapaciteta prijenosa podataka bežičnih lokalnih mreža baziranih na IEEE 802.11n/ac/ax standardima.

5.1. Funkcioniranje lokalne bežične računalne mreže

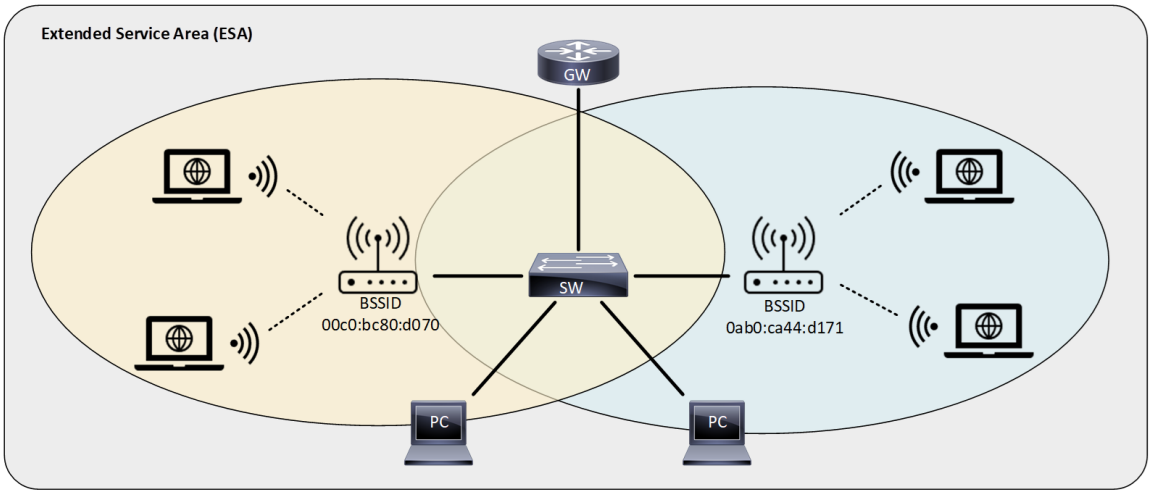
Više je načina na koje se može implementirati bežična lokalna računalna mreža, ali u ovom priručniku u središtu je tzv. infrastrukturni način, koji podrazumijeva postojanje

IKT sustava s preklopnicima, usmjernicima i drugim uređajima, a bežična mreža se implementira kao proširenje toga IKT sustava i služi za povezivanje bežičnih klijenata na postojeću žičanu računalnu mrežu. Postoje dvije osnovne bežične topologije, a to su: osnovni set bežičnog pristupa (engl. *Basic Service Set — BSS*) i prošireni set bežičnog pristupa (engl. *Extended Service Set*). BSS je topologija koja se sastoji od jedne bežične pristupne točke i svih bežičnih uređaja koji se povezuju na tu pristupnu točku. Područje signala koje pokriva BSS naziva se BSA — *Basic Service Area*. Ako ima više takvih BSA područja i želi se omogućiti prijelaz klijenata (prijenosnih računala, pametnih telefona i slično) između pojedinih BSA bez prekida signala (engl. *Roaming*), nužno je da se BSA područja signala preklapaju 10—15%. BSSID je identifikator za BSS, i to je MAC adresa bežične pristupne točke, te je uvijek povezan samo s jednom bežičnom pristupnom točkom.



Slika 195. Prikaz osnovnog područja pokrivanja BSA — Basic Service Area

Ako je potrebno pokriti veće područje bežičnim signalom i to nije izvedivo korištenjem jednog BSS-a, tada se kreira ESS. ESS je skup više BSS-ova povezanih posredstvom distribuiranog sustava. Svaki ESS je identificiran svojom SSID oznakom (engl. *Service Set ID*) koju obično konfigurira administrator sustava i ona predstavlja „naziv“ bežične mreže na koju se korisnik povezuje. Svi uređaji povezani na bežičnu mrežu mogu međusobno komunicirati koristeći taj ESS i mogu neprimjetno prelaziti između pojedinih BSS-ova koji su dio istog ESS-a (opcija engl. *Roaming*). To se područje naziva ESA (engl. *Extended Service Area*).



Slika 196. Prikaz proširenog područja pokrivanja ESA — Extended Service Area

5.1.1. CSMA/CA — *Carrier Sense Multiple Access With Collision Avoidance*

WLAN mreže su neistovremene (engl. *Half-duplex*) mreže, gdje svi uređaji dijele zajednički medij za prijenos podataka koji se može nazvati „eter“. Neistovremena komunikacija znači da samo jedan uređaj može odašiljati ili primati podatke u nekom trenutku. Problem u komunikaciji nastaje zato što bežični uređaj ne može „slušati“ dok odašilje podatke, odnosno ne može detektirati koliziju ako se ona dogodi. Kao rješenje za to WLAN mreže koriste CSMA/CA mehanizam za utvrđivanje kako i kada odašiljati podatke u „eter“. Način je funkcioniranja sljedeći:

1. Bežični uređaj osluškuje kanal je li slobodan, što znači da trenutno ne detektira drugi promet na kanalu. Kanal se također naziva nosač (engl. *Carrier*).
2. Bežični uređaj zatim šalje zahtjev za slanje poruke (engl. *Request to Send-RTS*) bežičnoj pristupnoj točki čime traži pristup mediju.
3. Bežična pristupna točka šalje poruku klijentu da je slobodno za slanje (engl. *Clear to Send-CTS*).
4. Ako bežični klijent ne primi CTS poruku, čeka nasumično vrijeme prije ponovnog pokretanja procesa.
5. Nakon što primi CTS, klijent šalje podatke.
6. Svi se prijenosi podataka potvrđuju. Ako bežični klijent ne primi potvrdu, pretpostavlja da je došlo do kolizije i ponovno pokreće proces.

5.1.2. Sigurnost bežične računalne mreže

Kako je bežično okruženje dostupno svima unutar dosega signala, bilo tko može pokušati pristupiti bežičnoj mreži ili pokušati izvesti određeni kibernetički napad. Često napadači ne moraju ni ući na posjed tvrtke, već je dovoljno da se približe zgradi i imaju pristup bežičnom signalu, što im omogućava kibernetički napad. Bežične mreže posebno su podložne sljedećim napadima:

1. Presretanje podataka (engl. *Man-in-the-Middle Attack — MITM*): bežični podaci trebaju biti šifrirani kako bi se spriječilo čitanje stranim napadačima.
2. Bežični uljezi: neovlašteni korisnici koji pokušavaju pristupiti mrežnim resursima mogu se odvratiti učinkovitim tehnikama provjere identiteta i prava pristupa.
3. Napadi uskraćivanja usluge (engl. *Denial of Service — DoS*): pristup uslugama WLAN-a može biti ugrožen slučajno ili namjerno. Postoje različita rješenja ovisno o izvoru DoS napada.
4. Neovlašteni AP-ovi: instalirani dobronamjerno ili zlonamjerno mogu se otkriti pomoću softvera za upravljanje bežičnim AP-ovima.

Kako bi se osigurala bežična računalna mreža, nužno je implementirati određene mehanizme. S obzirom na to da bežični signali prolaze kroz zidove i vidljivi su izvan učionica i zgrada, potrebno je onemogućiti da bežične pristupne točke odašilju svoj SSID, tako da svi korisnici koji se žele povezati moraju ručno upisati točan SSID za povezivanje na bežičnu mrežu.

Također, potrebno je implementirati mehanizme za provjeru identiteta koristeći nekakav dijeljeni ključ (lozinku) uz određenu razinu enkripcije pomoću dostupnih algoritama za autentikaciju, kao što su WEP (engl. *Wired Equivalent Privacy*), WPA (engl. *Wi-Fi Protected Access*), WPA2 ili WPA3, uz primjenu enkripcijskih algoritama poput *Temporal Key Integrity Protocol* (TKIP) ili *Advanced Encryption Standard* (AES). Današnji industrijski standard podrazumijeva korištenje WPA2 i AES algoritama.

6. IPv6 protokol i usmjeravanje u IPv6 mreži

U OVOJ CJELINI NAUČIT ĆETE:

- koji su razlozi za prelazak na IPv6 protokol
- kako izgleda struktura IPV6 adrese
- konfigurirati IPV6 adrese na usmjernicima
- implementirati usmjeravanje u IPV6 računalnoj mreži
- opisati glavne prijelazne metode s IPv4 na IPv6 protokol
- implementirati ručne IPv6 tunele.

IPv4 standard sve se teže nosi s izazovima modernog vremena, a najveći problem već danas, a pogotovo u budućnosti, veličina je adresnog prostora koju podržava. Kada se korištenje interneta naglo počelo širiti, postalo je jasno da klasno dijeljenje adresnog prostora jednostavno nema smisla i kao modifikacija u primjeni uvedene su tehnike kao što je VLSM (engl. *Variable Length Subnet Mask*), CIDR (engl. *Classless Interdomain Routing*), privatne adrese i NAT (engl. *Network Address Translation*). VLSM i CIDR povećavaju fleksibilnost dodjeljivanja IP adresa i optimiziranje usmjerničkih tablica, a NAT smanjuje potrebu za javnim IP adresama. Te tehnike pomažu, ali nikako nisu dugoročno rješenje problema u primjeni IPv4 protokola. IPv4 protokol ima i drugih karakteristika koje jednostavno nisu dorasle aktualnom vremenu, ali to nije tema ovog priručnika. Adresni je prostor dovoljan razlog za zamjenu IPv4 protokola pa za tu potrebu nova IPv6 verzija ima 128 bita za IP adrese, što bi trebalo

biti dovoljno daleko u budućnosti. Glavni razlog prijelaza na IPv6, dakle, proširenje je adresnog prostora i poboljšano adresiranje, ali iskorištena je prilika i za dodatna unaprjeđenja, pa je povećana učinkovitost u prosljeđivanju paketa i fleksibilnost za nove promjene, pojave li se u budućnosti. Iako je IPv6 protokol u primjeni, potpuni će se prelazak dogoditi postepeno kada sve organizacije koje koriste računalne mreže zaključe da im je to u interesu poslovanja. Dok god IPv4 radi i ne stvara probleme u poslovanju, do tada će biti u primjeni. Trenutno postoji više tehnika koje se mogu koristiti kako bi IPv4 i IPv6 protokoli koegzistirali.

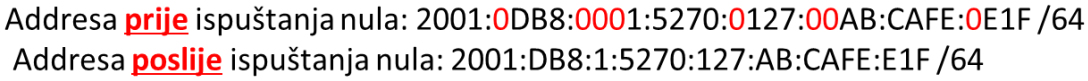
6.1. Struktura IPv6 adrese

IPv6 adresa je 128-bitni niz predstavljen korisnicima i administratorima IKT sustava kao heksadecimalni zapis. Na slici 197. primjer je IPv6 adrese u binarnom i u heksadecimalnom obliku.

Slika 197. Primjer IPv6 adrese u binarnom i heksadecimalnom obliku

IPv6 adresa sastoji se od 8 heksadecimalnih polja, koji se nazivaju heksteti, odvojenih dvotočkama, što bi odgovaralo skupinama od po 16 bitova. Svaka heksadecimalna znamenka sastoji se od 4 bita a vrijednost ta 4 bita redom je: 8, 4, 2, 1 — što znači da raspon heksadecimalnih znameni ide od 0 do 9, a zatim se koriste slova od A do F. Vrijednosti koje se mogu pojaviti u IPv6 adresi su: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Budući da su IPv6 adrese vrlo velike, postoje pravila po kojima se mogu skraćivati. U svakom hekstetu moguće je izostaviti vodeće nule u zapisu, pa tako hekstet koji bi se mogao pisati 000A, može se napisati samo kao A.



Slika 198. Primjer skraćivanja IPv6 adrese

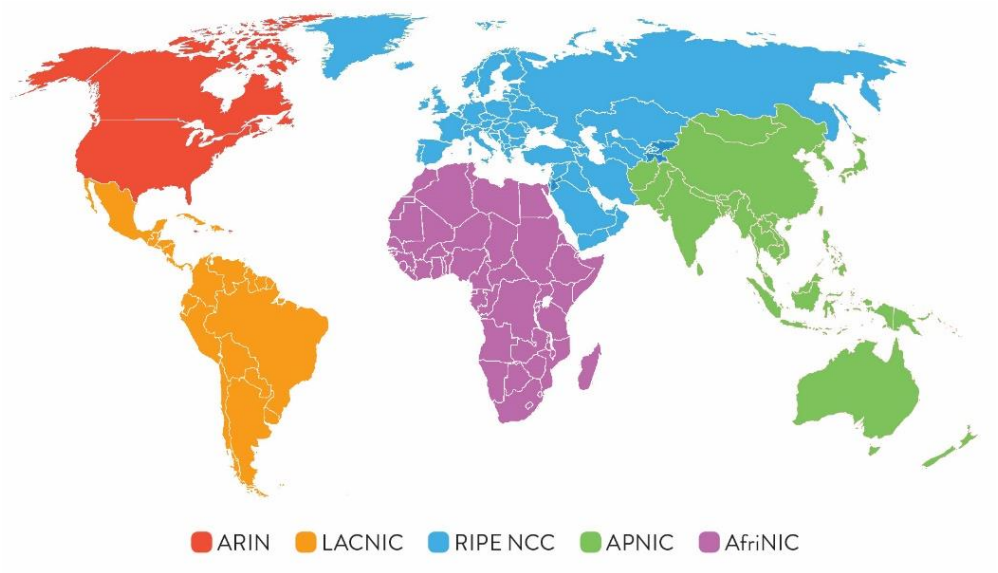
Također je moguće samo jednom u cijeloj IPv6 adresi skratiti neprekinuti niz heksteta u kojima su sve znamenke 0 i predstaviti taj niz kao duplu dvotočku. Na slici 199. vidljivo je skraćivanje vodećih nula i neprekinutih heksteta koji se sastoje od nula.

Originalna IPv6 adresa: 2001:0DB8:0000:0000:0127:0000:0000:0E1F
Skraćena IPv6 adresa: 2001:DB8::127:0:0:E1F

Slika 199. Skraćivanje niza heksteta sa znamenkama nula

Razlog zašto se ne može dva puta raditi to skraćivanje jest što bi računalu ili usmjerniku bilo nepoznato koliko je heksteta predstavljeno duplom dvotočkom ako ih ima dvije u IPv6 adresi. Stoga je na slici 198. prvi niz od dva heksteta koji se sastoje samo od nula predstavljen kao dupla dvotočka, dok je drugi niz skraćen ispuštanjem vodećih nula u svakom hekstetu.

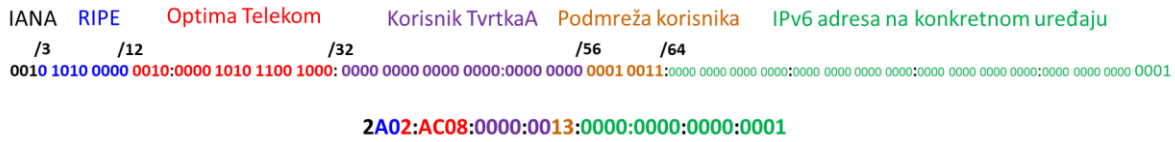
Postoji i hijerarhija dodjeljivanja IPv6 adresa, a kreće od krovne organizacije za dodjelu raznih brojeva koji se koriste na internetu, IANA-e (engl. *Internet Assigned Numbers Authority*). Ona regionalnim internetskim registrima (engl. *Regional Internet Registries — RIR*) dijeli blokove /12. Na svijetu postoji 5 velikih regionalnih registara, kao što je prikazano na slici 200.



Slika 200. Podjela svijeta na regionalne registre za IPV6 adrese

Regionalni registri dijele velikim korisnicima, primarno telekomunikacijskim operatorima blokove veličine/32 do /48, zatim oni svojim korisnicima blokove od /56.

Vizualno to izgleda kao na slici 201. Odlukom IANA-e trenutno je u upotrebi raspon 2000::/3, što je raspon od 2000-3FFF u prvom hekstetu i taj raspon je raspodijeljen na sve regionalne registre.



Slika 201. Primjer podmrežavanja IPV6 adresnog prostora

IPv6 podmrežavanje radi se po istom principu kao i u IPv4 protokolu, pa tako korištenjem mrežne maske /120 dobije se 256 kombinacija za IP adrese ili ako se koristi /118, dobivaju se 1024 kombinacije za IPv6 adrese. Preporuka je da se uvijek koristi /64 mrežna maska za svaku podmrežu kada se radi adresna shema. Također kada telekomunikacijski operatori alociraju IPv6 adresni prostor korisnicima, preporuka je da koriste /56.

IPv6 protokol podržava jednoodredišni, višeodredišni i promet prema najbližem odredištu, a sveodredišni više ne postoji. Također, postoje različiti tipovi IPv6 adresa, ali u ovom priručniku obradit će se *Global unicast* i *Link-local* IPv6 adrese.

Sve IPv6 adrese koje su općenito dostupne za direktnu komunikaciju, a to su IPv6 adrese iz raspona 2000::/3 nazivaju se *Global Unicast* IPv6 adrese. Te se IPv6 adrese dodjeljuju računalima, usmjernicima i drugim uređajima u IKT sustavu u privatnim organizacija i na javnom internetu. Korištenjem IPv6 adresa nema potrebe za NAT mehanizmom, a mreže privatnih organizacija se štite vatrozidima od vanjskih napada.

Link-local IPv6 adrese koriste se samo na jednoj vezi, npr. u LAN mreži ili na vezi između dva usmjernika. Te adrese su u predefiniranom rasponu FE80::/10. Kada usmjernici prosljeđuju promet između sebe, oni koriste upravo *Link-local* IPv6 adrese. Usmjernik koji ima više sučelja, može imati istu *Link-local* IPv6 adresu na svakom sučelju. To je korisno, na primjer, u edukaciji kada uređaj koji s oznakom „R5“ na svim sučeljima koristi IPv6 adresu FE80::5/64, pa je lakše pregledavati tablice usmjeravanja. *Link-local* IPv6 adrese mogu se konfigurirati ručno, ali u suprotnom uređaj će sam sebi generirati *Link-local* IPv6 adresu na temelju svoje MAC adrese, što zahtjeva proceduru koja se neće objašnjavati u ovom priručniku.

Može se konfigurirati više od jedne IPv6 adrese na nekom sučelju, što je važno u slučaju pogreške u konfiguraciji. Najprije, dakle, treba obrisati pogrešnu IPv6 adresu da bi se konfigurirala ispravna jer uređaj će prihvatiti sve upisano. Također IPv4 i IPV6 adrese mogu biti konfigurirane na ista sučelja i mogu biti paralelni sustavi, jedan baziran na IPv4, a drugi na IPV6. To je još poznato kao dvostruki stog (engl. *Dual Stack*). Na slici 202. prikazan je primjer konfiguracije IPv6 adrese na sučelju usmjernika.

```
R1(config)#interface loopback1
R1(config-if)#ipv6 address 2a02:0ac8:0001:000A:0000:0000:0000:0001/64
R1(config-if)#interface loopback2
R1(config-if)#ipv6 2a02:0ac8:0001:000B:0000:0000:0000:0001/64
R1(config-if)#exit
R1(config)#interface GigabitEthernet 1/0
R1(config-if)#ipv6 address 2a02:0ac8:0012:0000:0000:0000:0000:0001/64
R1(config-if)#ipv6 address Fe80::4 link-local
```

Slika 202. Primjer konfiguracije IPv6 adresa

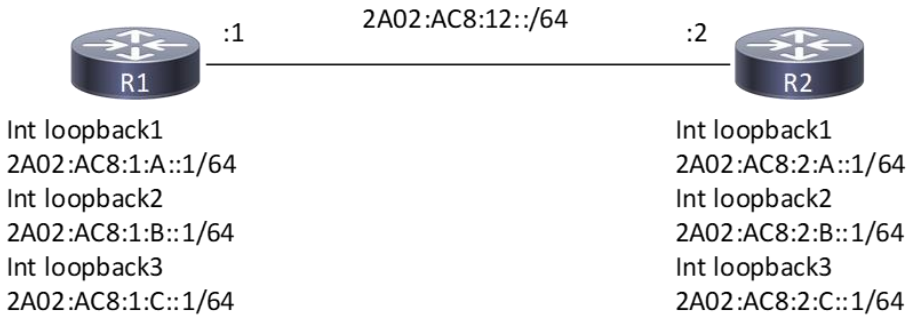
Može se primijetiti da je konfiguracija IPv6 adresa izvedena u neskraćenoj verziji. Konfiguracija IPv6 adresa može se provjeriti naredbom „show ipv6 interface brief“. Naredbe na usmjerniku gotovo su iste, osim što sada treba napisati „ipv6“. Na slici 203. vidljivo je da je usmjernik automatski napravio skraćivanje IPv6 adresa prema iznesenim pravilima. Također se može vidjeti razlika između *Link-local* IPv6 adresa koje konfigurira administrator na sučelju *GigabitEthernet1/0* i onih koje usmjernik sam stvara na ostalim sučeljima.

```
R1#sh ipv6 interface brief
GigabitEthernet1/0          [administratively down/down]
   FE80::4
   2A02:AC8:12::1
Loopback1                   [up/up]
   FE80::C804:8FF:FEC4:0
   2A02:AC8:1:A::1
Loopback2                   [up/up]
   FE80::C804:8FF:FEC4:0
   2A02:AC8:1:B::1
```

Slika 203. Prikaz sučelja na usmjerniku konfiguriranih IPv6 adresama

6.2. Usmjeravanje prometa u IPv6 mreži koristeći statičke putanje

Usmjeravanje prometa koristeći IPv6 adrese isto je kao u IPv4 adresama. Razlika je, naravno, u veličini IPv6 i teže ih je ručno konfigurirati. Također postoji razlika i u protokolima usmjeravanja koji su posebno razvijeni za razmjenu informacija o IPv6 podmrežama i puno su veći zapisi. Na slici 204. prikazana je mala topologija na kojoj je objašnjeno statičko i dinamičko usmjeravanje.



Slika 204. Jednostavna topologija za prikaz usmjeravanja u IPv6 mreži

Kada se konfiguriraju IPv6 statičke putanje, one mogu biti precizne, sažete ili predefinirane putanje, isto kao i u IPV4 protokolu. Razlika je što se u IPv6 protokolu za argument sljedećeg koraka (engl. *Next-hop*) može koristiti više IPv6 adresa na sučelju susjednog usmjernika. Statička predefinirana putanja jednostavna je i piše se u konfiguraciji *::/0,* što je isto kao u IPv4 *0.0.0.0/0*. Na slici 205. prikazano je kako se na jednom usmjerniku kao sljedeći korak (engl. *Next-hop*) koristi *Global Unicast* IPv6 adresa, dok se na drugom koristi *Link-local* IPv6 adresa. Bez obzira koju koristili, usmjernik će moći koristiti obje statičke putanje. Koristi li se *Link-local* IPv6 adresa, usmjernik mora znati kroz koje sučelje je ta adresa dostupna.

```
R1(config)#ipv6 route 2a02:ac8:1:a::1/64 GigabitEthernet 1/0 fe80:: 2
R2(config)#ipv6 route 2a02:ac8:1:a::1/64 2a02:ac8:12::1
R2#ping 2a02:ac8:1:a::1 source loopback 1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2A02:AC8:1:A::1, timeout is 2 seconds:
Packet sent with a source address of 2A02:AC8:2:A::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/17/24 ms
```

Slika 205. Primjer konfiguracije statičkih putanja za IPv6 podmreže

6.3. Usmjeravanje prometa koristeći dinamičke protokole usmjeravanja

U IPV6 koriste se isti dinamički protokoli, samo unaprijeđeni, RIPng, EIGRP i OSPFv3, a rade na istom principu i slično se konfiguriraju. Prije konfiguracije dinamičkih protokola usmjeravanja treba aktivirati usmjernik za IPv6 jednoodredišni prometu naredbom „ipv6 unicast routing“. Također, kada se konfiguriraju EIGRP i OSPF protokoli, oni i dalje koriste IPv4 format za oznaku usmjernika (engl. *Router-ID*), tako da je potrebno ili konfigurirati jedno IPV4 sučelje, ili ručno konfigurirati oznaku usmjernika. Prilikom oglašavanja podmreža u neki od IPv6 protokola usmjeravanja, to se više ne radi za vrijeme samog protokola, već na razini sučelja naredbama specifičnim za svaki od protokola usmjeravanja. Na slici 206. prikazane su naredbe kojima se konfigurira protokol usmjeravanja RIPng i način na koji se oglašavaju pojedine podmreže. RIPng sada koristi naziv instance protokola i to je lokalno značajno. Zato se na usmjernicima R1 i R2 na slici 204. mogu konfigurirati različiti nazivi a razmjena informacija o putanjama i dalje će biti moguća. Tablica usmjeravanja provjerava se naredbom „show ipv6 route“.

```
R1(config)#ipv6 router rip RIP-test
% IPv6 routing not enabled
R1(config)#ipv6 unicast-routing
R1(config)#ipv6 router rip RIP-test
R1(config-rtr)#exit
R1(config)#interface loopback 1
R1(config-if)#ipv6 rip RIP-test enable
R1(config-if)#exit
R1(config)#interface GigabitEthernet 1/0
R1(config-if)#ipv6 rip RIP-test enable
```

```
R2(config)#ipv6 router rip RIPng
R2(config-if)#ipv6 rip RIPng enable
R2(config-rtr)#interface gigabitEthernet 1/0
R2(config-if)#ipv6 rip RIPng enable
R2(config-if)#interface loopback 1
```

Slika 206. Primjer konfiguracije RIPng usmjerničkog protokola

Na slici 207. prikazan je primjer konfiguracije EIGRP usmjerničkog protokola za IPv6. Sada protokol usmjeravanja EIGRP zahtijeva aktiviranje naredbom „no shutdown“.

```
R2(config)#ipv6 router eigrp 12
R2(config-rtr)#router-id 1.1.1.1
R2(config-rtr)#no shutdown
R2(config-rtr)#exit
R2(config)#interface loopback 2
R2(config-if)#ipv6 eigrp 12
R2(config-if)#exi
R2(config)#int GigabitEthernet 1/0
R2(config-if)#ipv6 eigrp 12
```

```
R1(config)#ipv6 router eigrp 12
R1(config-rtr)#router-id 2.2.2.2
R1(config-rtr)#no shutdown
R1(config-rtr)#exit
R1(config)#interface loopback 2
R1(config-if)#ipv6 eigrp 12
R1(config-if)#exit
R1(config)#interface GigabitEthernet 1/0
R1(config-if)#ipv6 eigrp 12
```

Slika 207. Primjer konfiguracije EIGRP usmjerničkog protokola za IPv6

Na slici 208. prikazan je primjer konfiguracije usmjerničkog protokola OSPFv3. Bitno je primijetiti da se i oznaka područja sada konfigurira na sučeljima.

```
R1(config)#ipv6 router ospf 1
R1(config-rtr)#router-id 1.1.1.1
R1(config-rtr)#exi
R1(config)#interface GigabitEthernet 1/0
R1(config-if)#ipv6 ospf 1 area 0
R1(config-if)#exit
R1(config)#interface loopback 3
R1(config-if)#ipv6 ospf 1 area 0
```

```
R2(config)#ipv6 router ospf 1
R2(config-rtr)#router-id 2.2.2.2
R2(config-rtr)#exi
R2(config)#interface loopback 3
R2(config-if)#ipv6 ospf 1 area 0
R2(config-if)#exit
R2(config)#interface GigabitEthernet 1/0
R2(config-if)#ipv6 ospf 1 area 0
```

Slika 208. Primjer konfiguracije OSPFv3 usmjerničkog protokola

Nakon što su sva tri usmjernička protokola konfigurirana, na slici 209. može se vidjeti dio ispisa tablice usmjeravanja na oba usmjernika s topologije prikazane na slici 204.

```
R1#show ipv6 route
-dio ispisa nije prikazan-
R   2A02:AC8:2:A::/64 [120/2]
     via FE80::2, GigabitEthernet1/0
D   2A02:AC8:2:B::/64 [90/130816]
     via FE80::2, GigabitEthernet1/0
O   2A02:AC8:2:C::1/128 [110/1]
     via FE80::2, GigabitEthernet1/0
```

```
R2#show ipv6 route
-dio ispisa nije prikazan-
R   2A02:AC8:1:A::/64 [120/2]
     via FE80::1, GigabitEthernet1/0
D   2A02:AC8:1:B::/64 [90/130816]
     via FE80::1, GigabitEthernet1/0
O   2A02:AC8:1:C::1/128 [110/1]
     via FE80::1, GigabitEthernet1/0
```

Slika 209. Prikaz tablice usmjeravanja s putanjama svih konfiguriranih protokola usmjeravanja

Sva su tri usmjernička protokola konfigurirana istovremeno, ali samo po jedno *Loopback* sučelje oglašeno je u svaki protokol usmjeravanja. Usmjernici mogu istovremeno koristiti više protokola usmjeravanja iako to nije uobičajeno.

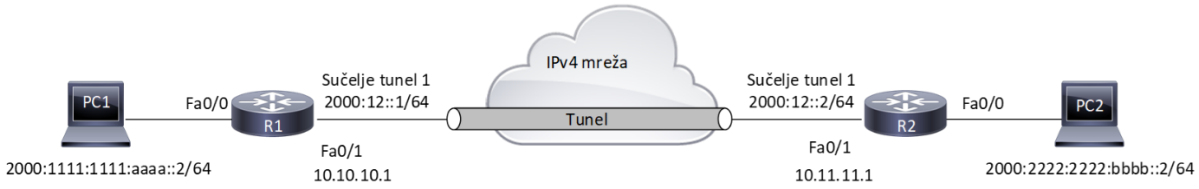
Redistribucija između protokola usmjeravanja radi se na usmjernicima koji koriste dva ili više protokola usmjeravanja, na isti način kao i u IPv4 protokolu.

6.4. Prijelazne metode u IPv6 komunikaciju pomoću IPv4 mreže

Danas još uvijek velik broj mreža koristi IPv4 protokol. Tvrtke neće prijeći na IPv6 dok to ne bude opravdano za poslovanje, bilo da će im prelazak omogućiti dodatne funkcionalnosti i povećati efikasnost IKT sustava, bilo da su uvjetovane regulatornim zahtjevima. Idealna bi bila jednostavna i neprimjetna tranzicija bez negativnih utjecaja na poslovanje tvrtke, što je rijetko moguće. Pozitivna je strana te tranzicije što ne mora odjednom ići u cijelosti i svi uređaji istovremeno, već taj proces može biti u skladu s poslovnim potrebama i raspoloživim resursima tvrtke. Za sam taj proces dostupni su različiti prijelazni mehanizmi koje tvrtke mogu implementirati kao privremeno rješenje dok cijelu IKT infrastrukturu ne prenesu u IPV6. Ovdje će biti opisana dva uobičajena i vrlo jednostavna prijelazna mehanizma, a to su „ručni tuneli“ (engl. *Manual IPV6 Tunnels*) i automatski „6 u 4 tuneli“ (engl. *Automatic 6 to 4 Tunnels*).

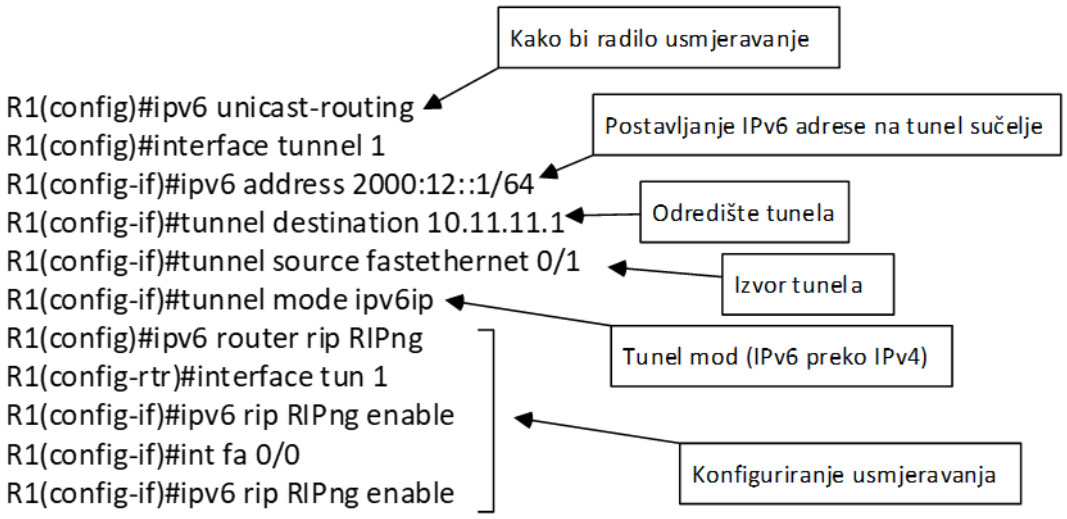
6.4.1. Ručni IPv6 tuneli

Ručno konfigurirane tunele podržava velika većina usmjernika koji se danas koriste, a opisani su u dokumentu RFC 4213. Korištenjem ručno konfiguriranih tunela ostvaruje se komunikacija između dvije ili više izoliranih IPv6 mreža postojećom IPv4 mrežom. Oznaka je protokola koji se koristi za enkapsulaciju 41, što je potrebno znati ako postoji vatrozid koji treba propustiti taj promet. Usmjernici na kojima se konfiguriraju takvi tuneli, moraju biti dvostrukoga stoga (engl. *Dual-Stack*), što znači da podržavaju i IPV4 i IPV6 protokole. Problem u ovakvim tunelima ograničena je skalabilnost, jer se kompleksnost mreže povećava eksponencijalno s brojem lokacija koje na taj način treba povezati. Prednost je ručnih tunela što su vrlo jednostavni za konfiguraciju i što se pomoću njih mogu koristiti protokoli usmjeravanja za razmjenu informacija o mrežama. Na slici 210. bit će prikazana konfiguracija ručnih IPv6 tunela između dvije izolirane IPV6 mreže koje će biti povezane postojećom IPV4 mrežom.



Slika 210. Princip rada ručnih IPv6 tunela

Na topologiji je vidljivo da su usmjernici R1 i R2 povezani IPv4 mrežom. Da bi mogli uspostaviti IPv6 tunele, nužno je da usmjernici mogu međusobno komunicirati IPV4 mrežom jer će sučelja Fa0/1 na R1 i Fa1/0 na R2 biti izvorište i odredište tunela. Na slici 211. prikazane su naredbe za konfiguraciju ručnih IPv6 tunela između R1 i R2. Konfiguracija na usmjerniku R2 je slična, ali treba prilagoditi IPv4 i IPv6 adrese koje se koriste.

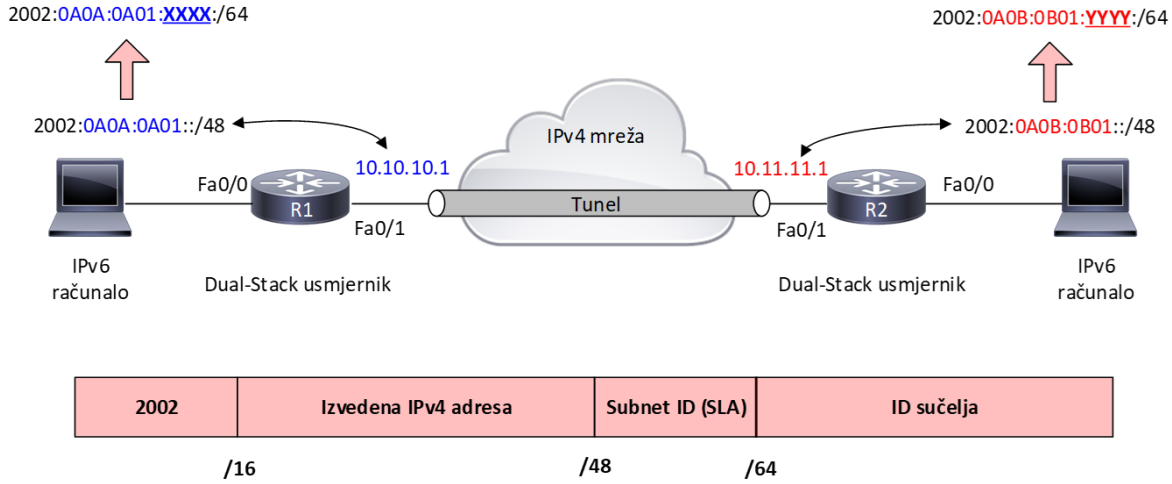


Slika 211. Primjer konfiguracije ručnih IPv6 tunela na usmjerniku R1

6.4.2. Automatski 6 u 4 tuneli

To je automatska metoda uspostave tunela opisana u dokumentu RFC 3056, koja se implementira na rubnim usmjernicima dvostrukoga stoga (engl. *Dual-Stack*) u mreži i može povezivati velik broj mreža istovremeno (engl. *Point-to-Multipoint*). U automatskom načinu tuneliranja nema puno konfiguracije i moguće je vrlo brzo povezati distribuirane IPv6 mreže tvrtke bez potrebe za velikim izmjenama. Ako se dobro isplanira, konfiguracija se samo replicira na sve usmjernike koje treba povezati. Odredište je tunela IPv4 adresa koja mora biti na neki način dostupna IPv4 mrežom, za što se mogu koristiti statičke putanje ili dinamički protokoli usmjeravanja.

Prema standardu svaka IPv6 mreža koja se povezuje mora početi prefiksom 2002::/16. Na prefiks 2002::/16 dodaje se IPv4 adresa koja će služiti kao izvor i odredište automatskih tunela u heksadecimalnom obliku. U mreži pod jednom administrativnom kontrolom može se koristiti bilo koje privatne IPv4 adrese ili od ISP-a zakupljena javna IPv4 adresa kako bi se internetom povezale sve lokacije. Koristi li se ISP-ova javna IPv4 adresa, od nje treba izvesti IPv6 mrežu iz rubnog usmjernika. Za primjer konfiguracije „6 u 4“ tunela koristit će se topologija na slici 212. Na topologiji je vidljivo da su IPv6 podmreže na usmjernicima izvedene od IPv4 adresa na vanjskim sučeljima (prema oblaku).



Slika 212. Topologija za "6 u 4" tuneliranje

U tablici na slici 213. može se vidjeti kako se pretvaraju IPv4 adrese s topologije u IPV6 adrese koje će biti korištene za lokalne IPV6 mreže iza usmjernika R1 i R2. Najprije treba odabrati IPv4 adrese, a tek onda se mogu izvesti IPV6 adrese podmreža koje će biti korištene.

Decimalno	10	10	10	1
Binarno	0000 1010	0000 1010	0000 1010	0000 0001
Hexadecimalno	0 A	0 A	0 A	0 1

Decimalno	10	11	11	1
Binarno	0000 1010	0000 1011	0000 1011	0000 0001
Hexadecimalno	0 A	0 B	0 B	0 1

Slika 213. Primjer pretvaranja IPv4 adrese u heksadecimalni zapis potreban za „6 u 4“ tuneliranje

Oznake XXXX i YYYY na slici 212. predstavljaju proizvoljne heksadecimalne znakove koji će ovisiti o načinu na koji se upravlja vlastitom mrežom, pa su zamjenjivi oznakom VLAN-a, oznakom zgrade i kata gdje se mreža nalazi ili nekim drugim znakovima. Kad se upiše još tih 16 bitova, dobiva se IPv6 adresa /64, što će zadovoljiti sve potrebe za IP adresama, isto kao i preporukama za izradu IPv6 podmreža. Na primjer, ako je podmreža iza usmjernika R1 zapravo lokacija tvrtke 3 i povezana s oznakom VLAN 135, tada bi XXXX mogao biti 3135.

Na slici 214. može se vidjeti primjer konfiguracije usmjernika R1 i R2 za dinamičke „6 u 4“ tunele. Konfiguracija je gotovo identična, što te tunele čini vrlo jednostavnim za konfiguraciju ako su oznake sučelja na svim usmjernicima iste. Ta bi se konfiguracija ponovila na svim lokacijama koje treba povezati tim načinom.

```
R1(config)#interface tunnel 1
R1(config-if)#tunnel source fastethernet 0/1
R1(config-if)#tunnel mode ipv6ip 6to4
R1(config-if)#ipv6 enable
R1(config-if)#exit
R1(config)#ipv6 route 2002::/16 tunnel 1
```

```
R2(config)#interface tunnel 1
R2(config-if)#tunnel source fastethernet 0/1
R2(config-if)#tunnel mode ipv6ip 6to4
R2(config-if)#ipv6 enable
R2(config-if)#exit
R2(config)#ipv6 route 2002::/16 tunnel 1
```

Slika 214. Primjer konfiguracije usmjernika za dinamičke "6 u 4" tunele

7. Nadzor i održavanje računalne mreže

U OVOJ CJELINI NAUČIT ĆETE:

- zašto je važno nadzirati i održavati računalnu mrežu
- kako se koriste CDP i LLDP mehanizmi
- što je NTP i kako se koristi u računalnoj mreži
- što je SNMP i kako se koristi
- što je *Syslog* i kako se koristi
- kako se sigurnosno pohranjuje konfiguracija usmjernika
- što je „Baseline“
- koji se pristupi koriste za otkrivanje i uklanjanje uzroka problema u računalnim mrežama.

U današnje vrijeme računalne mreže mogu biti vrlo velike i kompleksne i zato je nužno nadzirati ih i učinkovito njima upravljati. Osim što administratori računalne mreže moraju znati što se događa u njihovoj mreži i ima li kakvih problema ili sigurnosnih ugroza, nadzor pomaže i planiranju budućeg razvoja računalne mreže kako bi bila u skladu s potrebama poslovanja organizacije. U tu se svrhu primjenjuje strukturirani pristup upravljanju uređajima u računalnoj mreži, što podrazumijeva upravljanje pristupnim uređajima, promjenama konfiguracija na uređajima, sigurnosnim pohranama konfiguracija, licencama i raznim drugim elementima koji čine funkcionalnu

računalnu mrežu. Za nadzor i upravljanje računalnom mrežom koriste se različiti alati i servisi koji administratorima znatno olakšavaju upravljanje računalnom mrežom.

7.1. CDP i LLDP

Cisco Discovery Protocol (CDP) i *Link-layer Discovery Protocol* (LLDP) dva su protokola kojima se mogu otkrivati informacije o direktno povezanim mrežnim uređajima koji podržavaju te protokole. CDP je predefiniran na *Cisco* uređajima, dok LLDP treba posebno konfigurirati. Zbog sigurnosnih razloga ti protokoli trebaju biti ograničeni samo na sučeljima koja su pod administratorskom kontrolom i na koja nisu povezani vanjski korisnici jer bi potencijalni napadači mogli saznati vrijedne informacije, poput IP adresa uređaja, modela uređaja i verzije operativnog sustava na mrežnom uređaju. Za provjeru statusa CDP protokola na uređaju može se koristiti naredba „show cdp“. CDP se može isključiti u potpunosti ili uključiti naredbama „no cdp run“ i „cdp run“ u općenitom konfiguracijskom modu. Moguće je i na pojedinom sučelju mijenjati status CDP protokola naredbama „cdp enable“ i „no cdp enable“. Za uvid u informacije o susjednim uređajima koristi se naredba „show cdp neighbors“, a za stanje CDP protokola na sučeljima lokalnog uređaja naredba „show cdp interface“.

```
R3#show cdp neighbors
Capability Codes:  R - Router, T - Trans Bridge, B - Source Route Bridge
                   S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                   D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
  R2               Ser 6/0           174          R       7206VXR   Ser 6/0
  R1               Fas 0/0           170          R       7206VXR   Fas 0/0
```

Slika 215. Primjer ispisa naredbe „show cdp neighbors“

Na slici 215. u ispisu naredbe „show cdp neighbors“ vide se informacije o susjednim uređajima, može saznati naziv udaljenog uređaja (engl. *Device ID*), sučelje na lokalnom uređaju na koje je povezan susjedni uređaj (engl. *Local Intrfc*), mogućnosti susjednog uređaja (engl. *Capability*), model susjednog uređaja (engl. *Platform*) i sučelje na susjednom uređaju koje je povezano s lokalnim uređajem (engl. *Port ID*). Ako se želi saznati više detalja o pojedinom susjednom uređaju, na primjer njegovu IP adresu, koristi se naredba „show cdp neighbors detail“. Kad je poznata IP adresa

susjednog uređaja, može se povezati na taj uređaj te koristeći CDP protokol, otkrivati druge uređaje u mreži i na taj način izraditi topologiju mreže.

Za razliku od CDP protokola, LLDP protokol dostupan je na uređajima različitih proizvođača, ali ga podržava i *Cisco*. Ovisno o uređaju, LLDP predefinirano može biti uključen ili isključen, a može se uključiti ili isključiti općenito za cijeli uređaj naredbom „lldp run“ u općenitom konfiguracijskom modu ili na razini sučelja naredbama „lldp transmit“ i „lldp recieve“, a obje naredbe treba koristiti. Naredbe za otkrivanje susjednih uređaja slične su kao u CDP-u, pa se tako umjesto „show cdp neighbors“ koristi „show lldp neighbors“ ili, za detaljnije informacije, naredba „show lldp neighbors detail“.

7.2. NTP – Network Time Protocol

Kako bi cijeli proces nadzora, upravljanja i održavanja računalne mreže bio pravovremen i smislen, nužno je imati točno vrijeme na svim uređajima u računalnoj mreži. Ako ne bi bilo točno vrijeme, moglo bi se dogoditi da bude prilikom čitanja zapisa (engl. *Log*) o događajima, umjesto današnjeg datuma, datum od prije nekoliko dana ili čak godina, a to bi sigurno otežalo otkrivanje uzroka problema u mreži. Primarno mrežni uređaji koriste svoj vlastiti sat, ali to znači da može biti znatnih odstupanja u vremenu, zato je važna sinkronizacija vremena na svim uređajima. Umjesto ručne konfiguracije koristi se *Network Time Protocol* (NTP), koji omogućava mrežnim uređajima sinkronizaciju vlastitoga vremena s glavnim mjerodavnim izvorom vremena u IKT sustavu. Taj je protokol osobito važan u velikim mrežama gdje ručno konfiguriranje vremena nema nikakvog smisla. NTP protokol koristi UDP ulaz 123 na strani poslužitelja, dok klijenti koriste ulaz 1023 za rad, tako da je to potrebno uzeti u obzir i taj promet propustiti kroz vatrozide.

NTP je organiziran hijerarhijski gdje su NTP poslužitelji povezani u hijerarhiju s više razina. Te se razine nazivaju „stratum“, a stratum 0 najviša je razina u hijerarhiji. NTP poslužitelji niže u hijerarhiji obično se sinkroniziraju s NTP poslužiteljima koji su na višim razinama, na primjer NTP poslužitelj na razini stratum 3 sinkronizira se s NTP poslužiteljem stratum 2 ili stratum 1. Na razini stratum 0 autoritativni su izvori točnog vremena, i to su atomski satovi ili GPS satovi, kao najtočniji izvori vremena. Ti uređaji nisu mrežni uređaji. Stratum 1 mrežni su uređaji koji svoje vrijeme sinkroniziraju direktno sa stratum 0 uređajima i služe kao izvor točnog vremena za stratum 2 mrežne

uređaje pomoću NTP protokola. Uređaji na razini stratum 2 i niže povezani su sa stratum 1 uređajima računalnom mrežom i to su obično NTP klijenti. Najniža stratum razina je 16, što znači da uređaj nije sinkroniziran, dakle nema točno vrijeme.

```
R1#clock set 10:10:10 1 january 2019
R1#show clock detail
10:10:12.463 UTC Tue Jan 1 2019
Time source is user configuration
R1#configure terminal
R1(config)#ntp server 216.239.35.4
R1(config)#exit
R1#show clock detail
*05:33:35.824 UTC Sat Apr 29 2023
Time source is NTP
R1#
```

Slika 216. Primjer konfiguracije NTP servisa na uređaju

```
R1#sh ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~216.239.35.4    .GOOG.           1     23     64     1 68.332   1.285 187.53
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
R1#
```

Slika 217. Prikaz veze s NTP poslužiteljem

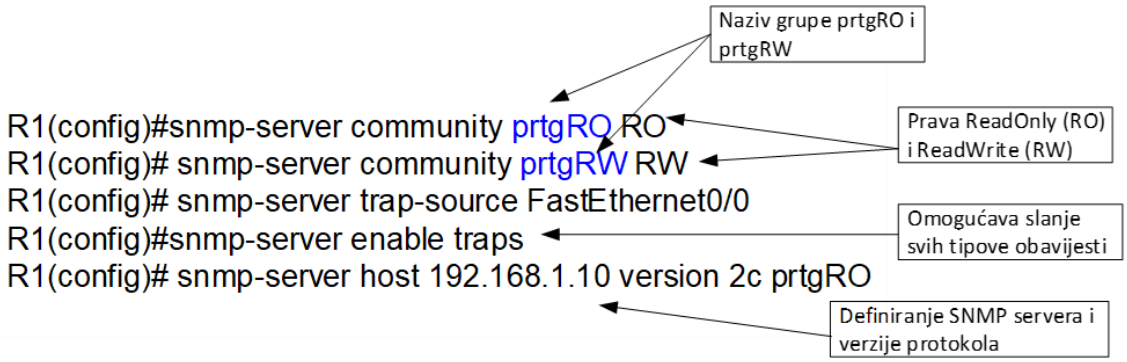
Za detaljnije informacije o statusu NTP poslužitelja i sinkroniziranosti koristi se naredba „show ntp status“.

7.3. SNMP protokol

SNMP (engl. *Simple Netwrok Management Protocol*) je protokol koji služi za prikupljanje parametara o stanju i performansama uređaja u mreži. Za komunikaciju koristi UDP ulaz 161, i to treba imati na umu prilikom konfiguracije vatrozida. SNMP se sastoji od SNMP upravitelja (obično softver instaliran na poslužitelju) i SNMP agenta (mrežna oprema i uređaji u mreži). SNMP može nadzirati i slati informacije o velikom broju elemenata u mreži. Na primjer, protokolom SNMP može se saznati koliko je sučelje na usmjerniku opterećeno, koliko koristi memorije, procesora, kolika je temperatura uređaja, broj okretaja ventilatora i slično. Kako bi SNMP mogao slati informacije o svim tim parametrima, postoji hijerarhijska struktura oznaka vezanih za

svaki parametar koji je moguće nadzirati (to je odluka proizvođača opreme). Ta hijerarhijska stablasta struktura naziva se *Management Information Base* (MIB) i sadrži *Object Identifiers* (OID). Svaki je OID vezan za određeni parametar nad kojim je proizvođač opreme omogućio nadzor. Na primjer, želi li se nadzirati prosječno zauzeće procesora u intervalu od 5 minuta, to se može praćenjem OID-a .1.3.6.1.4.1.9.2.1.56. U velikoj većini slučajeva nije nužno znati točan OID zato što na uređajima to biva prevedeno u neki razumljiv izraz, poput „cpmCPUTotal5min“ i vidljivo je u sustavu za nadzor.

Postoji nekoliko verzija protokola SNMP. SNMPv1 zastarjela je i više se ne koristi, dok se verzija SNMPv2c koristi i radi na principu „zajednica“ (engl. *Community*) za administriranje i nadzor uređaja. SNMPv3 je danas standardna verzija koja omogućava provjeru integriteta SNMP poruka, autentikaciju i kriptiranje komunikacije mrežom. Kao i u svakoj vrsti komunikacije, preporučuje se da i SNMP komunikacija bude sigurna, tako da bi bilo dobro koristiti SNMPv3. Konfiguracija protokola SNMPv2c na usmjerniku *Cisco* prikazana je na slici 218.



Slika 218. Primjer konfiguracije SNMP-a korištenjem zajednice

Osim dijela konfiguracije, na mrežnim uređajima treba imati softver na poslužitelju koji će primati sve te obavijesti iz mreže. Za tu svrhu može se koristiti ugrađene mogućnosti operativnih sustava, ali i posebne alate koje su razvili proizvođači nadzornih alata. Jedan od njih nadzorni je alat PRTG. Jednostavan je za korištenje, ima većinu funkcija koje su potrebne za malu mrežu, a osim toga, besplatan je za korištenje u malim mrežnim okruženjima.

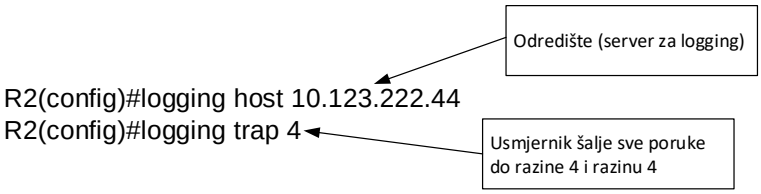
7.4. Prikupljanje sistemskih zapisa (engl. Log) — Syslog

Da bi administratori mogli primjereno reagirati na događaje u mreži, nužno je znati da se nešto dogodilo u stvarnom vremenu. Za to postoje razni mehanizmi, a jedan od njih je i *Syslog* protokol. Iako se pojam *Syslog* koristi za opis standarda, odnosi se i na protokol koji je razvijen na temelju njega. *Syslog* protokol koristi UDP ulaz 514, što znači da treba tu komunikaciju propustiti kroz vatrozid u mreži. Različiti mrežni uređaji podržavaju *Syslog* protokol, a na uređajima *Cisco* protokol je vidljiv prilikom konfiguracije, kada uređaj ispisuje na ekranu razne poruke da bi obavijestio administratora o događajima na njemu.

- 0 — *emergencies* (Sustav je neupotrebljiv.)
- 1 — *alerts* (Odmah treba poduzeti određene korake.)
- 2 — *critical* (Stanje je kritično.)
- 3 — *errors* (Poruka greške.)
- 4 — *warnings* (Poruka upozorenja.)
- 5 — *notifications* (Normalno, ali značajno stanje.)
- 6 — *informational* (Poruka je informativne prirode.)
- 7 — *debugging* (poruke o svim događajima na uređaju)

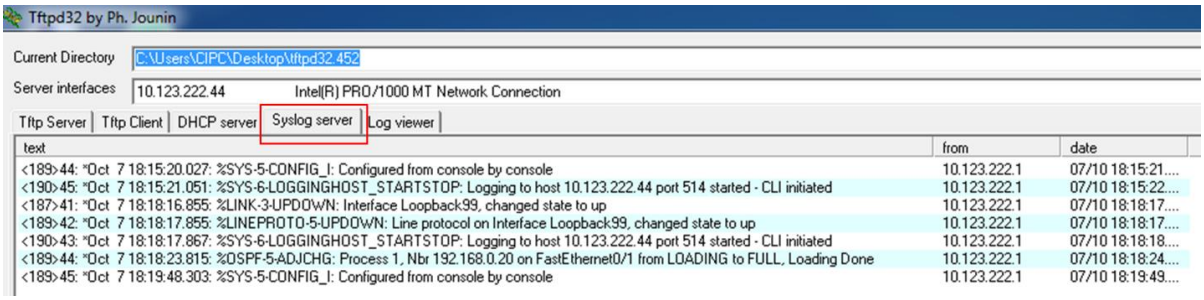
Slika 219. Razine važnosti pojedinih razina poruka na usmjerniku Cisco.

Pristup tim porukama ne mora biti direktno u konzoli uređaja, već se *Syslog* poruke mogu pohraniti na udaljenom poslužitelju i tako im se može pristupiti. Uobičajeno se *Syslog* poruke mogu vidjeti u *Logging Buffer* memorijskom prostoru na mrežnom uređaju, u konzolnoj liniji, u vty linijama i na *Syslog* poslužitelju. Mrežni administrator može odlučiti koje poruke želi prikazati i/ili poslati prema određenim odredištima, npr. može zabraniti da se u konzoli uređaja ispisuju poruke određene razine važnosti (engl. *Severity Level*). Na *Cisco* uređajima postoje poruke razine od 0 do 7, s tim da se poruke razine važnosti od 0 do 4 odnose na greške i kvarove u softveru i hardveru mrežnog uređaja, dok su poruke razine 7 rezultat korištenja pojedinih naredbi za traženje grešaka (engl. *Debug*). Želi li se *Syslog* poruke vidjeti na udaljenom poslužitelju, može se za to koristiti alat poput *Tftpd32* ili *Tftpd64*.



Slika 220. Primjer konfiguracije slanja Syslog poruka na udaljeni poslužitelj

Da bi *Syslog* poruke bile vidljive, alat na računalu ili poslužitelju treba biti pokrenut.



Slika 221. Prikaz Syslog poruka koje šalje mrežni uređaj na udaljeni poslužitelj

7.5. Sigurnosna pohrana konfiguracija i povrat konfiguracija uređaja

To je vrlo važan dio sustavnog održavanja mreže i sustava IKT-a općenito. U slučaju otkazivanja, a pogotovo katastrofalnih otkazivanja, nužno je imati zadnje funkcionalne konfiguracije koje će se koristiti na uređajima. Kako bi takve konfiguracije bile dostupne, treba redovito periodički te konfiguracije sigurnosno pohraniti. Osim konfiguracija, moguće je, i poželjno, sigurnosno pohraniti softver, firmver, operativne sustave i slično, a posebno prije različitih promjena i unaprjeđenja u mreži. Za sigurnosnu pohranu koriste se alati poput FTP, SCP, HTTP, HTTPS i slični protokoli. Sigurnosna pohrana može se raditi ručno, ali je uobičajenije automatski. Također, mora postojati poslužitelj za sigurnosnu pohranu a u tu se svrhu može koristiti ugrađena funkcija u operativnom sustavu ili poseban alat poput *Filezilla* koji je vrlo jednostavan za korištenje, a besplatan, što je za male mreže idealno rješenje.

```
R2#copy running-config ftp://cisco:class@10.123.222.44/R2_1
Address or name of remote host [10.123.222.44]?
Destination filename [R2_1]?
Writing R2_1 !
1510 bytes copied in 1.344 secs (1124 bytes/sec)
R2#

cisco =username kreiran na Filezilla serveru
class =lozinka kreirana za usera cisco na Filezilla serveru
10.123.222.44 = IP adresa FileZilla servera
R2_1 = ime konfiguracijske datoteke
```

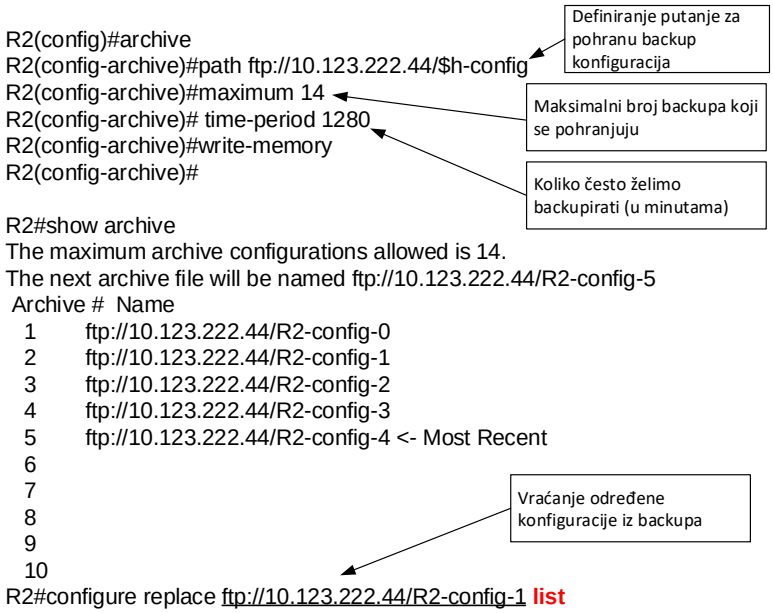
Slika 222. Primjer konfiguracije sigurnosne pohrane uz protokol FTP na usmjerniku Cisco za IOS

Želi li se pohraniti korisničko ime i lozinka da ih nije potrebno svaki puta upisivati, može se koristiti naredbe prikazane na slici 223.

```
R2(config)#ip ftp username cisco
R2(config)#ip ftp password class
R2#copy running-config ftp://10.123.222.44/R2_backup
Address or name of remote host [10.123.222.44]?
Destination filename [R2_backup]?
Writing R2_backup
!
1785 bytes copied in 0.948 secs (1883 bytes/sec)
R2#
```

Slika 223. Primjer konfiguracije sa stalno pohranjenim korisničkim podatcima

Dva prethodna primjera odnose se na ručno sigurnosno pohranjivanje, ali to uglavnom nije praktično, pogotovo u velikim mrežama. U tom slučaju uobičajeno je konfigurirati periodičku pohranu u određenim vremenskim intervalima. Vremenski interval će ovisiti o tome što se želi postići, odnosno koje su politike propisane. To je možda svaki dan ili svaki tjedan, ali period svakako mora biti dovoljno kratak da se u slučaju katastrofalnog otkazivanja može uspješno vratiti poslovanje u funkciju. Smjernica za sigurnosnu pohranu treba biti najnovija ispravna konfiguracija. Na slici 224. prikazano je kako se konfigurira periodička sigurnosna pohrana na *Cisco* usmjerniku korištenjem funkcije „archive“. Također, prikazano je na koji se način može vratiti određena konfiguracija iz pohrane. Osim što se možemo pohraniti na FTP poslužitelj, može i na druga odredišta, poput FLASH memorije na usmjerniku. Kada se dostigne maksimalni broj dozvoljenih pohrana, najstarija se pohrana briše. Uvijek ima onoliko zadnjih pohrana, koliko je konfigurirano naredbom „maximum“.



Slika 224. Konfiguracija periodičke pohrane i vraćanje pohranjene konfiguracije

7.6. Utvrđivanje uobičajenog stanja mreže

Kada u mreži postoji sustav za nadzor i kada administratori mreže imaju uvid u sve, moguće je pratiti kako se mreža ponaša u određenom vremenskom ciklusu kako bi se odredilo uobičajeno odnosno normalno stanje mreže. To uobičajeno stanje mreže naziva se *Baseline* i može se opisati kao „krvna slika“ sustava. Važno je prepoznati ključne dijelove mreže i sustava IKT-a kojima treba posvetiti posebnu pažnju kako ne bi negativno utjecali na poslovanje tvrtke. Da bi se utvrdio *Baseline*, period praćenja mreže mora biti minimalno jedan poslovni ciklus u kojem se mogu vidjeti svi aspekti njezina funkcioniranja. To može biti različito za različite periode u godini, npr. internetski dućan nema isti *Baseline* ljeti, za vrijeme godišnjih odmora, ili u prosincu, kada svi kupuju darove. U različitim periodima mreža i IKT sustav bit će različito opterećeni, a to je važno prepoznati prilikom definiranja *Baselinea*. Neki su elementi koji se pritom prate opterećenje sučelja, CPU, RAM, vrsta prometa u mreži, učestalost napada, performanse uređaja i slično. Koja je svrha *Baselinea*? Kada se ustanovi kako izgleda mreža i IKT sustav u normalnom stanju, s tim se mogu uspoređivati sve situacije i zaključiti je li nešto očekivano ili je izvan normalnih okvira. To daje kontrolu nad vlastitom mrežom i sigurnost. Problemi se puno lakše i brže detektiraju, a nakon uklanjanja njihova uzroka, može se vrlo jednostavno usporediti novo stanje s *Baseline* stanjem i vidjeti jesu li postupci bili uspješni ili ne. Osim u procesu TSHOOT-a, *Baseline*

je vrlo koristan za optimizaciju postojeće mrežne infrastrukture i planiranje proširenja u mreži.

Svako otkazivanje zahtijeva vrijeme koje je nužno kako bi se mreža vratila u funkcionalno stanje, dok posao tvrtke može biti ugrožen i stvarati velike gubitke, kako u financijskom smislu tako i u smislu njezine opstojnosti. Kada bi mrežne stranice tvrtke poput *eBaya* ili *Amazona* bile nedostupne nekoliko dana, to bi značilo velike financijske gubitke, a, osim toga, ugled tvrtke bio bi narušen te bi konkurencija to mogla iskoristiti. Kako bi IKT sustav bio što stabilniji i predvidljivo se ponašao, nužno je implementirati mehanizme za nadzor koji će administratorima omogućiti uvid u stanje IKT sustava i dati im vrijedne informacije koje će iskoristiti za planiranje održavanja, proširenja i obnavljanja elemenata IKT sustava u skladu s poslovnim potrebama tvrtke. Osim preventivnog djelovanja, sustav će nadzora u slučaju otkazivanja omogućiti administratorima puno brže pronaći uzrok i otkloniti ga na najefikasniji način. Također, sustav će nadzora omogućiti određivanje uobičajenih stanja u mreži, kao i onih izvanrednih na koje se mora reagirati prije nego izmaknu kontroli. Bitan element je svakako i adekvatno dokumentiranje sustava i održavanje te dokumentacije ažurnom kako bi sve odluke koje se donose vezano za IKT sustav bile temeljene na stvarnom stanju sustava. Kada postoji sustav nadzora i adekvatna dokumentacija, to je dobra pozicija da se svako otkazivanje koje se nije moglo predvidjeti, pronađe i efikasno ukloni. Ljudi koji to rade, moraju biti upoznati sa strukturiranim metodama otkrivanja i uklanjanja uzroka problema. Što je sustav složeniji, to je primjena strukturiranih metoda i implementacija procedura neophodnija za smanjivanje vremena potrebnog za rješavanje problema, ali i za umanjenje štete po poslovanje tvrtke.

7.7. Strukturirani proces otkrivanja i uklanjanja uzroka problema

Proces otkrivanja i uklanjanja uzroka problema u računalnim mrežama, u razgovoru poznat i kao TSHOOT, strukturirani je proces kojim administratori mogu efikasno uklanjati uzroke problema u IKT sustavu koji održavaju. Može se reći da je to proces postavljanja dijagnoze i pronalaska rješenja, odnosno uklanjanja uzroka problema ili, ako to nije moguće, smišljanja i implementacije privremenog rješenja (engl. *Workaround*). Naravno da će proces TSHOOT biti efikasniji ako su ljudi koji ga provode iskusni i upoznati s IKT sustavom koji održavaju, ali i uvježbani u korištenju propisanih

procedura. TSHOOT nije „egzaktna znanost“, jer se otkrivanju, pa čak i rješavanju problema može pristupiti na više različitih načina, ovisno o situaciji, topologiji, iskustvu i znanju inženjera, prioritetima i slično. Uvijek je dobro primjenjivati strukturirani pristup, jer se tako postiže kontinuirani napredak i uzroci problema se mogu ukloniti puno brže nego kada bi se koristio *ad hoc* pristup. Ako nije moguće ukloniti uzrok problema iz bilo kojeg razloga (možda bi uklanjanje uzroka utjecalo na neke druge sustave koji trenutno rade optimalno ili slično), traži se privremeno rješenje ili *Workaround*. Razlika je između otklanjanja uzroka i privremenog rješenja u ublažavanju simptoma i privremenom olakšanju, ali uzrok se ne otklanja. Važno je za stabilnost sustava da se *Workaround* što prije razriješi i IKT sustav dovede u optimalno stanje punoga raspona funkcionalnosti. Postoje različite verzije strukturiranih pristupa procesu TSHOOT, ovisno o korištenoj literaturi, o kakvom IKT sustavu ili o kakvom problemu se radi, ali bez obzira na različitosti, svi ti procesi dijele zajedničku osobinu stroge strukturiranosti. Budući da TSHOOT proces ovisi o različitim faktorima, veliku ulogu u njemu imaju ljudi. Bez iskusnih i stručnih administratora IKT sustava bilo koji proces, bez obzira koliko dobro bio razrađen i definiran, jednostavno neće sam ukloniti uzrok problema. Zato tvrtke ulažu velike napore u pronalaženje i školovanje ljudi koji rade u održavanju IKT sustava.

U nastavku su kratko opisani neki od često korištenih pristupa u procesu TSHOOT. Da bi se mogao početi koristiti bilo koji od navedenih pristupa, treba imati osnovu za njegov odabir. Ta se odluka donosi nakon jasnog definiranja problema i zaključivanja koji bi bio optimalan put za uklanjanje njegova uzroka. Ovdje je važno razlikovati problem koji percipira korisnik kada ga prijavljuje i stvarni problem koji je vezan za funkcioniranje IT sustava. Na primjer, korisnik prijavi da ne radi internet. Sigurno problem nije u internetu, već u nekom elementu između korisnika i interneta. Administrator detektira koji bi to mogao biti element i donosi odluku kako će uzrok problema ukloniti. Administrator je onaj koji poznaje IT sustav i funkcioniranje svih mehanizama i tehnologija u pozadini.

1. **Pristup od dolje prema gore (engl. *Bottom-Up Approach*):** od gore prema dolje misli se na OSI model i njegove slojeve. U ovom slučaju, u proces TSHOOT-a kreće se od najnižeg sloja, a to je fizički sloj. Cilj je provjeriti i potvrditi funkcioniranje nižih slojeva i na taj način suziti moguće uzroke

problema. Prednost toga pristupa je što nije potrebno odmah komunicirati s korisnikom ili se baviti aplikacijama i poslužiteljima (barem u početku). Taj pristup može biti problematičan u većim mrežama gdje jednostavno ima previše elemenata za provjeru i fizička infrastruktura nije tako lako dostupna jer se obično nalazi zaključana u podatkovnim centrima.

2. **Pristup od gore prema dolje (engl. *Top-Down Approach*):** proces TSHOOT-a započinje na aplikacijskom sloju te administrator provjerava aplikaciju i utvrđuje ponaša li se kako je očekivano. Taj se pristup obično koristi kada se problemi ponavljaju i postoji povijest problema s konkretnom aplikacijom. Nedostatak je što treba razgovarati s korisnikom, koji obično to doživljava kao smetnju u svojem poslu jer očekuje da sve radi bez greške. Prednost za administratore je to što je problem ograničen na konkretnu aplikaciju ili, još bolje, na konkretan uređaj. Kada se prepozna da je problem u nekoj aplikaciji, administratori ga mogu spriječiti na ostalim aplikacijama.
3. **Pristup podijeli pa vladaj (engl. *Divide and Conquer Approach*):** to je srednji pristup između dva prethodno spomenuta. Počinje se negdje unutar OSI modela (obično 2., 3. ili 4. sloj zbog naredbi koje se koriste, poput *ping*, *Telnet* po portu itd.), a zatim, ovisno o saznanju, kreće se ili prema gore ili prema dolje po OSI modelu. Efikasniji je od dva prethodno spomenuta pristupa.
4. **Pristup slijedi put (engl. *Follow-the-Path Approach*):** to je jedan od najjednostavnijih pristupa i obično se koristi uz neke od ostalih kao nadopuna. Bazira se na otkrivanju putanje prometa do odredišta, a zatim se usredotočuje samo na uređaje i veze na putu do odredišta. Cilj je eliminirati uređaje i veze koji nisu uzrok problema, i na taj način suziti područje istraživanja. Taj pristup mogu koristiti i oni koji ne znaju puno o IT sustavu da bi se s njime upoznali.
5. **Pristup usporedi konfiguracije (engl. *Compare-Configurations Approach*):** još jedan od najjednostavnijih pristupa, i možda najčešće korišten. Nadopuna je ostalima i bazira se na usporedbi konfiguracija, verzija softvera, hardvera; između uređaja koji radi normalno i uređaja koji ne radi kako se očekuje. Cilj je, na temelju uspoređivanja, zaključiti koji su elementi konfiguracije mogući uzrok problema. Prednost je što ne treba veliko znanje za korištenje ovog pristupa i može se brzo riješiti problem (uz kvalitetnu dokumentaciju i *Baseline*).

Nedostatak je što se mreža može vratiti u funkcionalno stanje, a da nije poznato zašto se to dogodilo, čime je potencijalno nastao još veći problem u budućnosti.

6. **Pristup zamjene komponenti (engl. *Swap-Components Approach*):** vrlo jednostavan pristup koji se koristi za izoliranje problema i, u načelu, to je metoda pokušaja i pogreške. Osnova mu je zamjena komponenata (sučelja, kabeli, uređaji, …) i promatranje što se događa s problemom (je li uklonjen ili se premjestio zajedno s uređajem). Nedostatak je što je ta metoda ograničena na samo jednu komponentu i podrazumijeva se da je jedan element u kvaru. Ako je više uređaja povezano uzrokom problema, to nije dobar pristup.

8. Virtualizacija i automatizacija računalne mreže

U OVOJ CJELINI NAUČIT ĆETE:

- što je virtualizacija programski definirane računalne mreže
- što je okruženje računarstva u oblaku
- koja je uloga i način rada SDN kontrolera
- koji su tipovi programibilnih računalnih mreža
- što je automatizacija u računalnim mrežama.

Već duže vrijeme dio poslovanja velikog dijela tvrtki, a negdje i cijelo poslovanje ovisi o virtualizaciji. Virtualizacija je tehnologija koja omogućava bolje iskorištenje fizičkih resursa, kao što je memorija, diskovni prostor i procesorska snaga, odvajajući operativni sustav od fizičkih resursa. Na fizičkom računalu većinu vremena ti resursi stoje neiskorišteni u punom kapacitetu, pa s osnove ulaganja to nije dobar povrat investicije. Tri računala koja koriste 30% svojih resursa u bilo kojem trenutku mogla bi raditi na jednom fizičkom uređaju i koristiti 90% resursa. Ovdje se ne računa vrijeme kada računala nisu uopće aktivna, pa tijekom noći, kada su sva računala u nekom uredu isključena, taj problem povrata ulaganja još je izraženiji. Zbog toga je osmišljena virtualizacija, odnosno virtualizacijski softver (engl. *Hypervisor*), koji sve fizičke resurse može predstaviti kao apstrakciju i na taj način na jednom fizičkom uređaju omogućava instalaciju više operativnih sustava sa svim funkcionalnostima kao da su na njemu direktno. Hipervizor je operativni sustav koji se instalira na hardver i zatim omogućava

instalaciju operativnih sustava u virtualnom okruženju u kojem su svi fizički resursi predstavljeni kao logički ili virtualni resursi. Tako instalirani operativni sustavi sa svim aplikacijama koje su potrebne nazivaju se virtualna računala. Svakom virtualnom računalu virtualizacijski softver dodjeljuje dio fizičke memorije, diskovnog prostora, procesora i ostale fizičke komponente kao što su mrežne kartice i slično. Na taj način fizički resursi se mogu znatno bolje iskoristiti, a što je više virtualnih računala, iskorištenje je bolje. Postoje hipervizori 1. i 2. razine. Hipervizori 1. razine instaliraju se direktno na fizički hardver i jedan primjer bio bi *VMware vSphere Hypervisor*. Hipervizori 2. razine instaliraju se kao dio operativnog sustava a primjer bi bio *Microsoft Hyper-V*. Virtualizacija je osnova na kojoj su se razvila okruženja računarstva u oblaku koja se danas koriste u velikoj mjeri.

8.1. Okruženje računarstva u oblaku

Računarstvo u oblaku (razgovorno samo: oblak) čine tri ključna elementa, a to su podatkovni centri, virtualizacijski softver i virtualna računala. Postoje i drugi elementi, ali to su osnovni bez kojih se ne može govoriti o računarstvu u oblaku. U oblaku postoje različite kategorije usluga, a najpoznatije su SaaS (engl. *Software as a Service*), gdje korisnici mogu platiti samo da se određena aplikacija pokrene u okruženju računarstva u oblaku, bez ikakvoga kontakta s bilo kakvim operativnim sustavom, a koja će biti dostupna korisnicima. Primjer su internetske stranice, e-pošta, komunikacijske aplikacije, *Office 365* i slično. Svu odgovornost za održavanje pristupa tim aplikacijama snosi pružatelj usluge. PaaS (engl. *Platform as a Service*) je okruženje računarstva u oblaku kompleksnije nego SaaS jer uključuje sve potrebne elemente za razvoj aplikacija, kao što su baze podataka i razni alati za razvojne inženjere, a pružatelj PaaS usluge odgovoran je za pristup tom razvojnom okruženju. IaaS (engl. *Infrastructure as a Service*) još je širi pojam i uključuje virtualne poslužitelje, virtualna računala, virtualne mreže i slično. Praktički moguće je zakupiti sve elemente infrastrukture koja bi inače bila potrebna nekoj tvrtki kada bi htjela imati svoj fizički podatkovni centar. Pružatelj IaaS usluge odgovoran je za omogućavanje pristupa svim elementima infrastrukture administratorima tvrtke koja je zakupila virtualno okruženje u oblaku. Također pružatelji usluga računarstva u oblaku nude i usluge podrške korisnicima toga okruženja, što je posebno korisno malim tvrtkama. Ta usluga se još naziva ITaaS (engl. *IT Support for Each of the Cloud Services*).

Neke od prednosti računarstva u oblaku su:

- Mogućnost pristupa resursima s bilo koje lokacije;
- Puno bolje iskorištenje resursa, jer se koriste samo kada je potrebno za virtualna računala koja ih trebaju;
- Smanjena potreba za osobljem koje će održavati okruženje računarstva u oblaku;
- Manji troškovi opreme, energije, prostora i radne snage;
- Lako je postići skalabilnost i visoku dostupnost za kritične aplikacije.

Važna je prednost računarstva u oblaku lako i brzo podizanje IKT infrastrukture sa svim elementima potrebnima za poslovanje nove tvrtke. Model plaćanja po potrebi vrlo je koristan tvrtkama koje pokreću svoje poslovanje pa im svi resursi trebaju a ne mogu si priuštiti kupnju skupe računalne opreme za vlastitu IKT infrastrukturu.

Postoji nekoliko modela okruženja računarstva u oblaku:

Javni oblak (engl. *Public Cloud*) — pristup javnom oblaku dostupan je svim korisnicima koji su spremni platiti za njegovo korištenje. Usluge u takvom oblaku ponekad se nude besplatno, ali i uz model „pay-per-use“, kao na primjer, plaćanje diskovnog prostora u oblaku. Pristup javnom oblaku uvijek je internetom.

Privatni oblak (engl. *Private Cloud*) — aplikacije, servisi i resursi privatnih oblaka namijenjeni su specifičnim organizacijama kao što su vlade, banke i slično, koje su i vlasnici opreme i sami ga održavaju i unaprjeđuju. Pristup tom oblaku privatnom je mrežom tvrtke koja je vlasnik oblaka, a održavanje je vrlo skupo i zahtjevno. Takav oblak također mogu održavati i vanjske organizacije, ali uz sigurnosna ograničenja.

Hibridni oblak (engl. *Hybrid Cloud*) — takav se oblak obično sastoji od dva ili više oblaka, kombinacija je javnog i privatnog oblaka koji su povezani i dijele komunikacijsku mrežu pa je potrebno svaki dio štititi da ne bi došlo do kibernetičkih napada. Administratori takvih oblaka imaju različite razine prava pristupa kako bi takvo okruženje bilo adekvatno osigurano.

Namjenski oblaci (engl. *Community Cloud*) — takve oblake koriste obično organizacije sličnih interesa i prilagođeni su određenoj vrsti poslovanja. Na primjer, organizacije kao što su bolnice i druge zdravstvene ustanove, koje podliježu

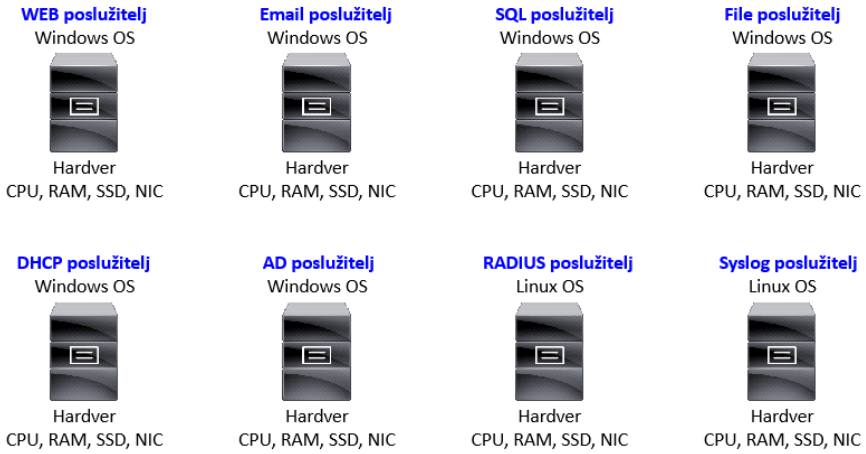
određenim zakonima i normama, dijele isti oblak jer je tako lakše i pružateljima usluga računarstva u oblaku, ali je bolje i tim organizacijama.

Neki od najpopularnijih oblaka su *Amazon Web Services — AWS*, *Microsoft Azure*, *Google Cloud*, ali ima i drugih.

8.2. Virtualizacija

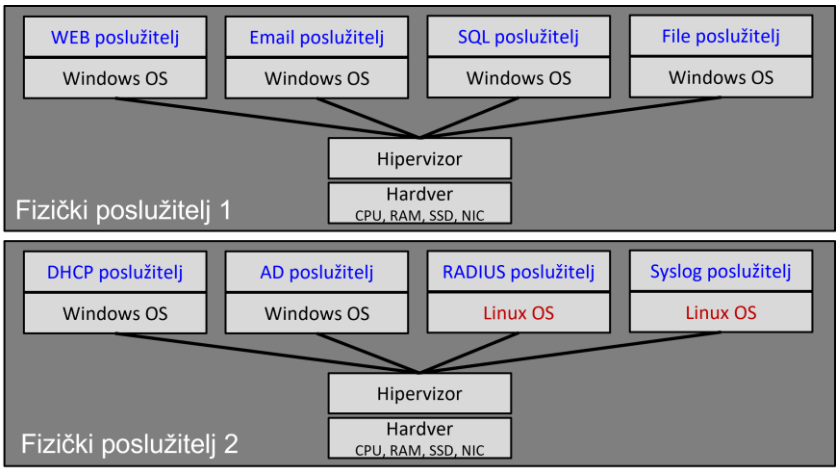
Termini računarstvo u oblaku i virtualizacija često se koriste kao sinonimi, ali znače različito i treba ih jasno razlikovati i razumjeti. Virtualizacijska tehnologija temelj je za okruženje računarstva u oblaku i bez virtualizacije ono nije moguće. Računarstvo u oblaku podrazumijeva automatizirano tehnološko okruženje u kojem postoji mnoštvo različitih virtualiziranih usluga. Također treba razlikovati virtualiziranog poslužitelja i virtualizacijskog poslužitelja. Virtualizirani poslužitelj je onaj koji je nekad bio fizički, a sada je virtualan, dok je virtualizacijski poslužitelj fizički poslužitelj na kojem je instaliran hipervizor i na njemu se nalazi više virtualnih poslužitelja.

Svrha virtualizacije razdvajanje je operativnog sustava od fizičkog hardvera. Prije virtualizacije svaki fizički poslužitelj u organizaciji imao je samo jednu ulogu, pa je tako bio poseban fizički poslužitelj poseban mrežne stranice, poseban baze podataka, poseban DHCP poslužitelj, poseban poslužitelj e-pošte i slično. To je značilo da svi resursi koje aplikacija ne koristi, kao što su RAM memorija, CPU, diskovni prostor i slično, ostaju neoptimalno iskorišteni. Kada se zbroje svi neiskorišteni resursi na svakom pojedinom fizičkom poslužitelju, zaključak je da su u svakom trenutku možda potrebna samo 3 fizička poslužitelja, a ne 10. Osim neoptimalne potrošnje resursa, velik je problem i otkaz jedne od ključnih komponenti, npr. u slučaju otkaza fizičkog poslužitelja, kao što je domenski kontroler ili DNS poslužitelj, organizacija ima velike probleme.



Slika 225. Namjenski fizički poslužitelji koji su neučinkovito korišteni

Prije virtualizacije nije bilo izbora, no sada se sve te servise koji su prije tražili poseban fizički poslužitelj, može se instalirati na samo njih nekoliko koji imaju virtualizacijski softver, i na taj način znatno uštedjeti na trošku fizičkih poslužitelja. To je postalo jasnije, što je organizacija imala veću infrastrukturu.



Slika 226. Princip virtualizacije servisa koji su prije zahtijevali fizički poslužitelj

Virtualna je okolina vrlo pogodna za jednostavnu implementaciju redundancije i visoke dostupnosti virtualnih računala za relativno mali trošak. Virtualno okruženje može biti redundantno tako da je na dva fizička poslužitelja po jedan virtualni DNS poslužitelj, pa otkaže li jedan iz bilo kojeg razloga, drugi je na raspolaganju. Također dva identična virtualna mrežna poslužitelja mogu raditi na dva različita hipervizora tako da u slučaju otkaza hipervizora ili fizičkog poslužitelja na kojem je hipervizor instaliran, korisnici i dalje mogu pristupiti mrežnom poslužitelju. Moguće je održavati i mijenjati fizičke

poslužitelje tako da se sva virtualna računala prebace na drugi fizički poslužitelj, a nakon popravka ili zamjene originalnog fizičkog poslužitelja, vrate. Virtualizacija uz najvažniju prednosti smanjenja ukupnih troškova IKT sustava, ima mnoge druge prednosti:

- Manje potrebne fizičke opreme i poslužitelja zbog činjenice da se više operativnih sustava s raznim aplikacijama i servisima nalazi na jednom fizičkom poslužitelju s hipervizorom;
- Manja potrošnja energije jer se koristi znatno manje fizičkih poslužitelja;
- Manje potrebnog prostora, što je posebno važno jer je prostor u podatkovnim centrima ograničen i vrlo skup;
- Fleksibilnost virtualne okoline i mogućnosti lakšeg oporavka od katastrofalnih otkaza;
- Brža implementacija servisa ključnih za poslovanje, poput DNS, DHCP, *Active Directory* i slično.;
- Laka priprema okoline za testiranje aplikacija, zakrpa i raznih konfiguracija potrebnih za unaprjeđenje ključnih servisa.

8.3. Virtualizacija mreže i programski definirane mreže

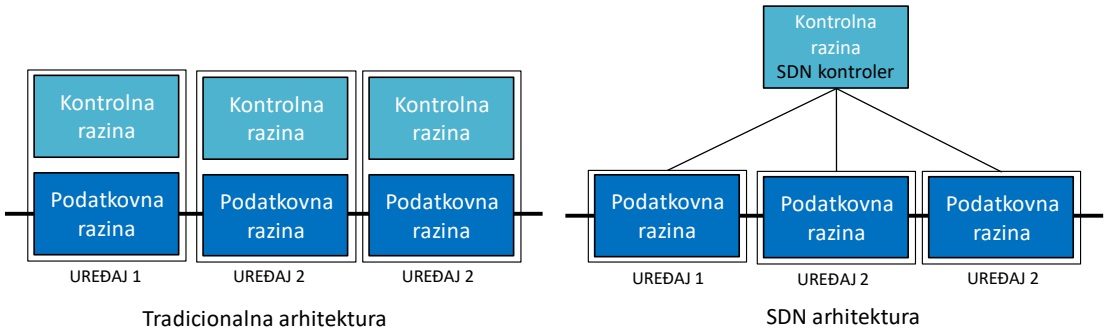
Svaki mrežni uređaj poput usmjernika i preklopnika koristi dvije razine kontrole prosljeđivanja podataka.

Kontrolna razina — to je „mozak“ uređaja i koristi se za donošenje odluka o tome kamo treba proslijediti mrežni promet. Kontrolnu razinu čine L2 i L3 mehanizmi, poput IPv4 i IPv6 usmjerničkih tablica koje se rade na temelju informacija razmijenjenih putem protokola usmjeravanja (RIP, OSPF, EIGRP), ARP tablica, tablica MAC adresa i slično. Te se informacije obrađuju u procesoru uređaja.

Podatkovna razina — ta se razina sastoji od elemenata u samoj elektronici koja je povezana sa sučeljima na uređaju. Na ovoj razini podatke obrađuju uglavnom posebne elektroničke komponente i mikroprocesori i ne obrađuje ih glavni CPU.

Programski definirane mreže rade na principu odvajanja upravljačke razine od podatkovne razine tako da se upravljačka razina centralizira na jednom glavnom upravljačkom uređaju a on upravlja velikim brojem ostalih uređaja na pristupnom sloju

mreže koji funkcioniraju samo na podatkovnoj razini. Uređaji na pristupu mreže praktički samo prosljeđuju podatke na temelju uputa od centralnog uređaja, što povećava efikasnost i sigurnost mreže i omogućava lakšu implementaciju raznih servisa, poput kvalitete usluge u mreži i slično.



Slika 227. Princip rada programski definirane mreže

Osim kontrolne i podatkovne razine postoji još i upravljačka razina koja služi administratorima da različitim protokolima poput SSH (engl. *Secure Shell*), FTP (engl. *File Transfer Protocol*) ili HTTPS (engl. *Secure Hypertext Transfer Protocol*) upravljaju cijelim sustavom.

Postoje različite tehnologije virtualizacije mreže, a njihov odabir i implementacija znatno će ovisiti o potrebama i preferencijama korisnika. Postoje i različite arhitekture virtualnih mreža koje načelno razvijaju pojedini proizvođači mrežnih tehnologija, kao što je tvrtka *Cisco* sa svojim rješenjem *Cisco Application Centric Infrastructure* — ACI. U ovom priručniku kratko će biti predstavljena generalna arhitektura programski definiranih mreža (engl. *Software-Defined Networking — SDN*). SDN komponente čini sljedeće:

OpenFlow — protokol je razvijen na sveučilištu Stanford i služi za upravljanje prometom između usmjernika, preklopnika, bežičnih pristupnih točaka i kontrolora, i kao takav osnova je za izgradnju SDN rješenja.

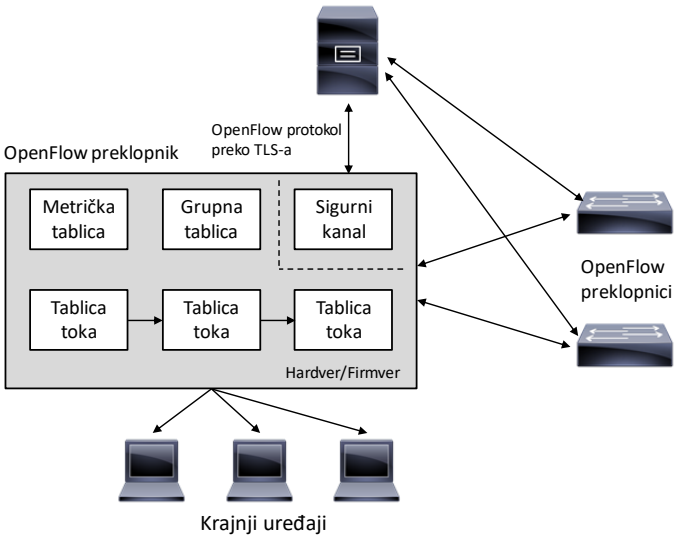
OpenStack — platforma je za virtualizaciju i orkestraciju dizajnirana za izgradnju skalabilnih okruženja računarstva u oblaku i osnova je za realizaciju IaaS rješenja. *OpenStack* se često koristi u kombinaciji s ACI arhitekturom *Cisco*-a. Orkestracija u kontekstu računalnih mreža proces je automatizacije implementacije mrežnih rješenja, kao što su poslužitelji, pohrana, preklopnici, usmjerivači i razni servisi.

Ostale komponente — uključuju sučelje prema sustavu usmjeravanja (engl. *Interface to Routing System* — I2RS), transparentno povezivanje više veza (engl. *Transparent Interconnection of Lots of Links — TRILL*), *Cisco FabricPath* (FP) i IEEE 802.1aq protokol za premošćivanje najkraćim putem (engl. *Shortest Path Bridging* — SPB).

Potpun prikaz načela rada programski definirane mreže prikazan je na slici 228. U okviru SDN-a koriste se programibilna aplikacijska sučelja (engl. *Application Programming Interfaces* — API). API je set standardiziranih koraka koji definiraju način na koji jedna aplikacija traži uslugu od druge aplikacije. U kontekstu SDN-a kontrolor koristi API za komunikaciju s vanjskim aplikacijama koje administratori koriste za upravljanje cijelim sustavom čiji je SDN kontrolor samo jedan dio. To su tzv. *Northbound* API i njih administratori koriste kako bi oblikovali mrežni promet i implementirali različite servise u mrežu. SDN kontrolor također koristi tzv. *Southbound* API kako bi definirao način rada preklopnika i usmjernika na podatkovnoj razini. *OpenFlow* je uobičajen i često korišten *Southbound* API.

8.4. Uloga i način rada SDN kontrolora

SDN kontrolor definira tokove podataka između centralizirane kontrolne razine i podatkovne razine na kojoj su pojedinačni mrežni uređaji. Svaki podatkovni tok (komunikacija) prije nego se uspostavi, mora dobiti odobrenje SDN kontrolora koji potvrđuje je li on u skladu s definiranim mrežnim politikama. Ako kontrolor odobri određeni podatkovni tok, određuje putanju mrežom i dodaje potrebne zapise za taj tok na sve uređaje u mreži koji rade na podatkovnoj razini. Sve napredne kompleksne radnje i izračune radi SDN kontrolor i tako kreira tablice podatkovnih tokova. SDN kontrolor komunicira s uređajima na podatkovnoj razini putem *OpenFlow* protokola posredstvom TLS mehanizma (engl. *Transport Layer Security*) kako bi komunikacija bila sigurna. Svaki *OpenFlow* preklopnik povezuje se s drugim *OpenFlow* preklopnicima, a mogu se povezati i direktno s korisničkim uređajima koji su dio podatkovnog toka.



Slika 2288. Princip rada SDN kontrolora koji upravlja preklopnicima putem OpenFlow protokola

U svakom preklopniku postoji niz tablica na razini hardvera ili firmvera koje se koriste za upravljanje tokovima podataka kroz preklopnik. Za preklopnik tok podataka je niz paketa koji odgovaraju određenom zapisu u tablici toka (engl. *Flow Table*). Postoje tri tipa tablica, kako je prikazano na slici 228.

Tablica toka (engl. *Flow Table*) — prema toj tablici uređaj prepoznaje pojedine pakete kao dio specifičnog toka podataka i definira što treba napraviti s tim paketima. Može biti više tablica toka koje se izvršavaju slijedno.

Grupne tablice (engl. *Group Table*) — tablica toka može neki podatkovni tok preusmjeriti na grupnu tablicu kako bi preklopnik pokrenuo niz predefiniranih radnji na jedan ili više podatkovnih tokova.

Metrička tablica (engl. *Meter Table*) — aktivira niz radnji vezanih za performanse pojedinog podatkovnog toka, poput ograničavanja brzine za taj podatkovni tok i slično.

8.5. Tipovi programibilnih računalnih mreža

Postoje tri tipa programibilnih računalnih mreža:

Device-based SDN — u ovom slučaju mrežni se uređaji programiraju u aplikaciji koja se nalazi na samom uređaju ili na nekom poslužitelju u mreži. Aplikacije se mogu programirati „C“ ili *Python* programskim jezikom, a s mrežnim uređajima komuniciraju putem *OpenFlow* protokola.

Controller-based SDN — taj tip programibilnih mreža podrazumijeva primjenu SDN kontrolora koji je povezan sa svim mrežnim uređajima. Razne aplikacije mogu se povezati sa SDN kontrolorom i tako se može upravljati prometom u mreži.

Policy-based SDN — taj je tip sličan kao *Controller-based SDN*, gdje je centralni SDN kontrolor povezan sa svim uređajima u mreži. Uključuje još jedan sloj mrežnih politika koji funkcionira na višoj razini apstrakcije. Koriste se ugrađene aplikacije koje automatiziraju napredne konfiguracijske zadatke korisniku prijateljskim grafičkim sučeljem. Vještine programiranja nisu potrebne.

8.6. Automatizacija u računalnim mrežama

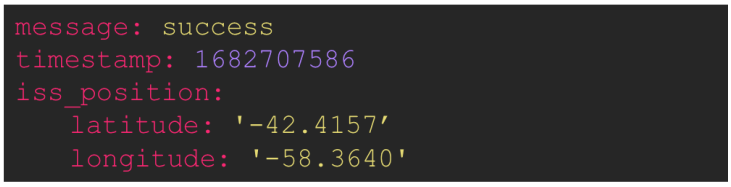
Automatizacija je bilo koji rutinski proces koji se odvija bez uplitanja čovjeka, što uvelike eliminira mogućnost pojave ljudskih grešaka. Neke su od prednosti automatizacije neprekinut rad uređaja i pogona, konzistentniji rezultati u proizvodnji, prikupljanje ogromne količine podataka brzo i kontinuirano, upravljanje opasnim procesima bez rizika za ljude i slično. Da bi se računalna mreža mogla uspješno automatizirati, koriste se različiti alati, ali je važno odabrati određeni format podataka koji će odabrani alat moći interpretirati i koje će mrežni uređaji moći iskoristiti. U svijetu mrežne automatizacije postoji nekoliko formata podataka koji se često koriste, a odabir će ovisiti o formatu koji podržava aplikacija ili alat za automatizaciju ili skripta koja se koristi za upravljanje konfiguracijama mrežnih uređaja.

JavaScript Object Notation (JSON)
eXtensible Markup Language (XML)
YAML Ain’t Markup Language (YAML)

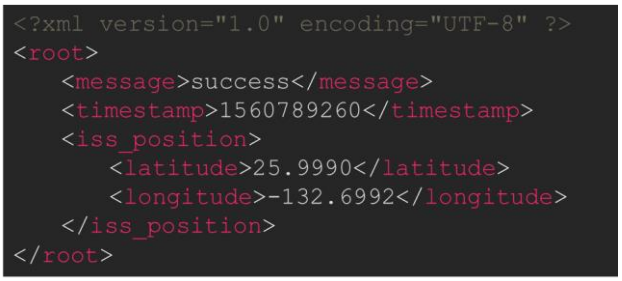
Slika 22929. Često korišteni formati podataka za automatizaciju i programiranje računalnih mreža

Na slikama 230. — 232. prikazani su isti podatci o poziciji međunarodne svemirske stanice koji se nalaze na poveznici http://api.open-notify.org/iss-now.json u sva tri formata kako bi se vidjela razlika u njima iako su sami podatci isti. Podatci na poveznici stalno će se mijenjati jer se međunarodna svemirska postaja kreće vrlo brzo i neće odgovarati stanju na slikama.

Slika 2300. Podatci u JSON formatu

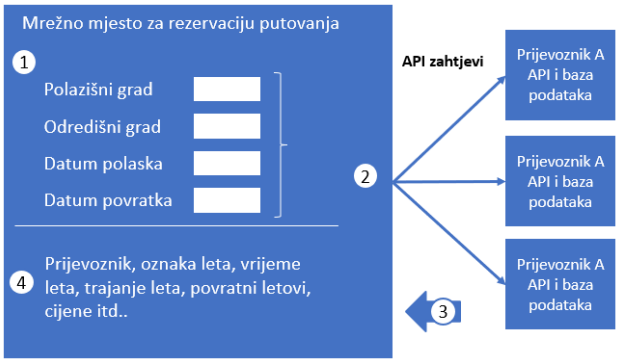


Slika 2311. Podatci u YAML formatu



Slika 2322. Podatci u XML formatu

Odabir prikladnog formata podataka bitan je kada se koristi aplikacijsko programsko sučelje (engl. *Application Programming Interface*), danas praktički sveprisutno u svim situacijama kada jedna aplikacija želi pristupiti podatcima neke druge aplikacije, na primjer kada se koristi internetska stranica za rezervaciju letova koja posredstvom API sučelja pristupa potrebnim podatcima u bazi podataka raznih avioprijevoznika. Da bi to bilo moguće, nužni su standardizirani načini komunikacije i standardizirani formati podataka koji se razmjenjuju između različitih aplikacija. Stoga što postoje API sučelja koje koriste aviokompanije, moguće je korisniku doći do informacija različitih avioprijevoznika, a sve jednim mrežnim sučeljem neke treće strane koja te podatke nema na svojim poslužiteljima, već do njih dolazi u realnom vremenu pomoću API sučelja.



Slika 2333. Princip rada API sučelja

Postoje tri vrste API sučelja.

Javna API sučelja (engl. *Open API* ili *Public API*) — to su javno dostupna API sučelja i mogu se koristiti bez ograničenja. Primjer dohvaćanja podataka o poziciji međunarodne svemirske stanice javno je dostupan API, još jedan primjer bilo bi korištenje *Google* karata za navigaciju i slično.

Privatna API sučelja (engl. *Internal* ili *Private API*) — to su API sučelja razvijena za primjenu u privatnim organizacijama, na primjer autorizirani pristup podatcima o prodaji tvrtke prodajnim predstavnicima na njihovim mobilnim telefonima.

Partnerska API sučelja (engl. *Partner API*) — ta se API sučelja koriste između partnerskih organizacija kako bi se olakšalo poslovanje partnerskih tvrtki. Primjer bi bila putnička agencija koja koristi API sučelja aviokompanija za rezervaciju letova.

Jedan od najzastupljenijih arhitekturnih stilova mrežnih API sučelja danas je tzv. REST (engl. *Representational State Transfer*) API, koji radi povrh HTTP protokola i definira set funkcija koje razvojni inženjeri mogu koristiti za slanje poruke koristeći HTTP protokol. Kada se API sučelje radi po principima REST arhitekture, tada se naziva „RESTful“. Ti su principi: da je API orijentiran kao klijentski-poslužitelj, da nema pohrane nikakvih klijentskih podataka na poslužitelju, već se sve pohranjuje na klijentu i da klijent može koristiti odgovore iz priručne memorije (engl. *Cache Memory*) u svrhu povećanja performansi.

RESTful API koriste uobičajene HTTP metode uključujući POST, GET, PUT, PATCH i DELETE, i te metode su mapirane na RESTful operacije prema tablici na slici 234.

HTTP Method	RESTful Operation
POST	Create
GET	Read
PUT/PATCH	Update
DELETE	Delete

Slika 2344. Mapiranje HTTP metoda i RESTful API operacija

Mnoge internetske stranice i aplikacije koriste API sučelja za pristup informacijama i pružanje raznih usluga korisnicima tih aplikacija. Za automatizaciju mreže bitno je da se API sučelja mogu pozvati i u programskim jezicima kao što je *Python,* koji se često koristi upravo za automatizaciju računalne mreže.

Konfiguracijski alati za upravljanje automatiziranom računalnom mrežom u znatnoj mjeri koriste upravo RESTful API zahtjeve za automatizaciju rutinskih zadataka u mrežama s tisućama uređaja. Više je razloga zašto je taj pristup bolji od klasičnog pristupa, u kojem jedan administrator svaki uređaj konfigurira posebno, a neki su: kontrola verzija softvera i konfiguracije na uređajima, konzistentnost atributa uređaja, konfiguracija i izmjene kompleksnih elemenata konfiguracije poput protokola usmjeravanja i lista za kontrolu pristupa.

Obično ti alati objedinjuju funkcionalnosti automatizacije svih zadataka vezanih za konfiguraciju uređaja i orkestracije svih automatiziranih zadataka u mreži kako bi se znalo kojim se redom što događa i kako bi se moglo pametno upravljati procesom izmjena u mreži.

Nekoliko je glavnih alata koji se koriste u automatiziranim mrežama, a to su *Ansible*, *Chef*, *Puppet* i *SaltStack*. Cilj je svih tih alata pojednostavljenje i smanjenje potrebnog vremena za konfiguraciju i održavanje velikih računalnih mreža koje sadrže tisuće uređaja.

Svi navedeni alati dolaze s detaljnom dokumentacijom za konfiguraciju RESTful API komunikacije. Svi podržavaju JSON i YAML format podataka, ali i neke druge.

Ansible i *SaltStack* temelje se na *Python* programskom jeziku, za razliku od *Chef* i *Puppet*, koji se temelje na *Ruby* programskom jeziku, sličnom *Pythonu*, ali nije tako jednostavan i intuitivan.

Upravljanje konfiguracijama uređaja može biti korištenjem agenata ili bez njih. Ako se koriste agenti, tada se konfiguracijama upravlja po principu povlačenja (engl. *Pull-Based*), što znači da se agenti instalirani na mrežne uređaje povremeno povezuju na glavni uređaj (engl. *Master*) kako bi povukli konfiguraciju. Promjene se rade na glavnom uređaju, agenti ih povlače i izvršavaju na ostalim uređajima. Ako se radi bez agenata, to je princip guranja (engl. *Push-Based*). Konfiguracijska skripta se pokreće na glavnom uređaju, glavni uređaj se povezuje na sve ostale uređaje i odradi sve zadatke u skripti. Od svih navedenih alata samo je *Ansible* u mogućnosti raditi bez agenata, što ga i čini toliko popularnim alatom.

Koristeći sve navedene alate administratori sustava zapravo rade niz instrukcija koje će biti izvršene. Svaki alat ima svoj naziv za taj niz ili set instrukcija, pa se tako koriste: *Playbook (Ansible)*, *Cookbok* (*Chef*), *Manifest (Puppet)* i *Pillar (SaltStack)*.

Svi logotipi i nazivi prikazani u ovom priručniku su vlasništvo respektivnih kompanija.

“Sadržaj publikacije/emitiranog materijala isključiva je odgovornost RCK ELPROS”

Projekt je sufinancirala Europska unija iz Europskog socijalnog fonda.