Prijenos multimedijskog sadržaja koristeći mrežne simulatore

Martan, Fran

SVEUČILIŠTE SJEVER

SVEUČILIŠNI CENTAR VARAŽDIN

DIPLOMSKI RAD br. 095-MMD-2023

PRIJENOS MULTIMEDIJSKOG SADRŽAJA KORISTEĆI MREŽNE SIMULATORE

Fran Martan

Varaždin, rujan 2023.

SVEUČILIŠTE SJEVER

SVEUČILIŠNI CENTAR VARAŽDIN

Studij Multimedije

DIPLOMSKI RAD br. 095-MMD-2023

PRIJENOS MULTIMEDIJSKOG SADRŽAJA KORISTEĆI MREŽNE SIMULATORE

Student:
Fran Martan, 0081158346

Mentor:
izv. prof. dr. sc. Emil Dumić

Varaždin, rujan 2023.

Sveučilište Sjever
Sveučilišni centar Varaždin
104. brigade 3, HR–42000 Varaždin

Prijava diplomskog rada

Definiranje teme diplomskog rada i povjerenstva

ODJEL Odjel za multimediju

STUDIJ diplomski sveučilišni studij Multimedija

PRISTUPNIK Martan Fran | JMBAG 0081158346

DATUM 26.06.2023. | KOLEGIJ Multimedijska videotehnologija

NASLOV RADA Prijenos multimedijskog sadržaja koristeći mrežne simulatore

NASLOV RADA NA ENGL. JEZIKU Multimedia content transmission using network simulators

MENTOR Emil Dumić | ZVANJE izv.prof.dr.sc.

ČLANOVI POVJERENSTVA

1. doc. art. dr. sc. Mario Periša - predsjednik
2. izv. prof. dr. sc. Dean Valdec - član
3. izv. prof. dr. sc. Emil Dumić - mentor
4. doc. dr. sc. Andrija Bernik - zamjenski član
5.

Zadatak diplomskog rada

BROJ 095-MMD-2023

OPIS

U ovom radu će biti objašnjeni pojmovi vezani za mrežne tehnologije, te realizirane simulacije prijenosa multimedijskog sadržaja (zvuk, video) pomoću mrežnog simulatora.
Bit će objašnjene osnove mrežne tehnologije, implementacija LAN (Local Area Network), VLAN (Virtual LAN), WAN, Cloud mreža te STP (Spanning Tree Protocol) i RSTP (Rapid STP) protokoli. Opisat će se IPv4 adresiranje i usmjeravanje te IPv6. Također će biti objašnjeni OSPF (Open Shortest Path First) i EIGRP (Enhanced Interior Gateway Routing Protocol) protokoli usmjeravanja. Objasnit će se IP kontrolne liste za pristup (ACL, Access Control List). Opisat će se DHCP protokol (Dynamic Host Configuration Protocol). Opisat će se NAT (Network Address Translation) servis. Opisat će se razlika standardnih i SDN (Software Defined Networks) mreža te OpenFlow komunikacijski protokol. Opisat će se koncepti QoS (Quality of Service) i QoE (Quality of Experience). Bit će opisani i alati za simulaciju mrežne arhitekture, poput omnet++ simulatora.

U praktičnom dijelu rada koristit će se omnet++ mrežni simulator, pomoću kojeg će se simulirati slanje paketa (npr. za videozapis), u ovisnosti o različitim parametrima mreže, te o simuliranim gubicima paketa (PLR, Packet Loss Rate). Ovisnost kvalitete videozapisa o gubicima paketa će se potom usporediti sa subjektivnim rezultatima (MOS, Mean Opinion Score) od nekih postojećih istraživanja.

ZADATAK URUČEN 25.08.2023. | POTPIS MENTORA Emil Dumić

Sveučilište
Sjever

SVEUČILIŠTE SJEVER

IZJAVA O AUTORSTVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, FRAN MARTAN *(ime i prezime)* pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica ~~završnog~~/diplomskog *(obrisati nepotrebno)* rada pod naslovom PRIJENOS MULTIMEDIJSKOG SADRŽAJA KORISTEĆI MREŽNE SIMULATORE *(upisati naslov)* te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:
(upisati ime i prezime)

FRAN MARTAN

(vlastoručni potpis)

Sukladno čl. 83. Zakonu o znanstvenoj djelatnost i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Sukladno čl. 111. Zakona o autorskom pravu i srodnim pravima student se ne može protiviti da se njegov završni rad stvoren na bilo kojem studiju na visokom učilištu učini dostupnim javnosti na odgovarajućoj javnoj mrežnoj bazi sveučilišne knjižnice, knjižnice sastavnice sveučilišta, knjižnice veleučilišta ili visoke škole i/ili na javnoj mrežnoj bazi završnih radova Nacionalne i sveučilišne knjižnice, sukladno zakonu kojim se uređuje znanstvena i umjetnička djelatnost i visoko obrazovanje.

Zahvale

Zahvaljujem se svim profesorima Sveučilišta Sjever u Varaždinu na prenesenom znanju i trudu kroz diplomski studij. Studij multimedije, zanimljivi predmeti i relevantne informacije motivirale su me za daljnji napredak u životu.

Također se zahvaljujem mentoru izv. prof. dr. sc. Emilu Dumiću na uloženom trudu i pomoći pri pisanju diplomskog rada. Prijedlozi, smjernice i veliki broj literature omogućili su mi da otkrijem nove činjenice i alate u mrežnim sustavima, te da završim diplomski rad.

Posebno se zahvaljujem svojim roditeljima bez kojih moje studiranje uopće ne bi bilo moguće. Njihova neopisiva potpora u svim trenucima, omogućila mi je da uspješno završim svih pet godina studija na Sveučilištu Sjever.

Sažetak

Diplomski rad istražuje ključne komponente modernih komunikacijskih mreža, uključujući referentne modele, mrežne protokole, tehnologije i uređaje. Opisuje se adresiranje i usmjeravanje u IP verzijama 4 i 6, te se detaljno opisuju dinamično protokoli usmjeravanja. Rad se usredotočuje na virtualizaciju i analizu prijenosnih protokola poput „Transmission Control“ protokola i „User Datagram“ protokola, te njihov utjecaj na prijenos podataka. Fokus rada je na naprednom simulacijskom alatu OMNeT++ i programskom okviru INET. Kroz OMNeT++ simulacije, analizira se utjecaj gubitaka paketa na prijenos multimedijskog sadržaja.

Ključne riječi: mreža, Ethernet, protokol, kvaliteta, Omnet++, simulacija

Summary

The master's thesis explores the key components of modern communication networks, including reference models, network protocols, technologies, and devices. It describes addressing and routing in IP versions 4 and 6, with a detailed focus on dynamic routing protocols. The thesis emphasizes virtualization and analyzes transport protocols such as the „Transmission Control Protocol“ and „User Datagram Protocol“, and their impact on data transmission. The thesis centers on the advanced simulation tool OMNeT++ and the INET framework. Through OMNeT++ simulations, the impact of packet loss on multimedia content transmission is analzyed.

Keywords: network, Ethernet, protocol, quality, Omnet++, simulation

Sadržaj

1. Uvod

Danas u modernom svijetu, komunikacija između računala i mobilnih uređaja ključna je za funkciju društva. Vrlo je bitno razumijevanje načina na koji se podaci prenose, analiziraju i obrađuju unutar komunikacijskih mreža. Veliki broj mrežnih inženjera sve se više okreće simulacijama i simulacijskim alatima kako bi kvalitetnije analizirali različite mrežne scenarije, protokole i rješenja bez opasnosti.

Diplomski rad istražuje ključne komponente modernih komunikacijskih mreža poput referentnih modela mrežnog povezivanja, mrežnih protokola, mrežnih tehnologija i uređaja. Pobliže se opisuje virtualizacija i tehnologija oblaka. Analiziraju se funkcije prijenosnih protokola poput TCP i UDP, te kako one utječu na promet. Opisuju se procesi adresiranja i usmjeravanja uređaja i prometa u IP verzije 4 i IP verzije 6 mreža. Detaljno se opisuju funkcije i tehnologije koje su potrebne da bi funkcionirale mreže, te se istražuje razlika između lokalnih mreža i mreža šireg geografskog područja. Opisuju se ključni dinamični protokoli usmjeravanja koji su zaslužni za prijenos podataka od odredišta do destinacije.

Diplomski rad usredotočuje se na napredni simulacijski alat OMNeT++ i njegov programski okvir imenom INET, njihovu instalaciju, stvaranje mrežnih modula i konfiguraciju postavka za željene mrežne scenarije koji se prikazuju putem grafičkog sučelja simulacijskog alata. Detaljno se opisuje instalacija virtualnih operativnih sustava distribucije Linux.

Kroz OMNeT++ i INET programski okvir modeliraju se dvije različite mreže i nekoliko događaja kako bi se mogla stvoriti dublja analiza i utjecaj gubitaka paketa na prijenos multimedijskog sadržaja.

2. Referentni modeli mrežnog povezivanja

Referentni modeli mrežnog povezivanja konceptualni su okviri kako bi se lakše razumjeli i organizirali načini na koji različiti mrežni sustavi komuniciraju i razmjenjuju podatke. Referentni modeli obuhvaćaju različite slojeve ili razine koji posjeduju specifične funkcije, te rješavaju specifične ciljeve. Takve funkcije omogućuju jednostavniju i efikasniju implementaciju mrežnih tehnologija, te osiguravaju interoperabilnost među različitim uređajima, različitih proizvođača. Slojevi referentnih modela mrežnog povezivanja jasno razdvajaju odgovornosti pojedinih funkcija.

Trenutno postoje dva modela mrežnog povezivanja, no samo je jedan u upotrebi. Zastarjeli model koji se više ne koristi je OSI model, a zamijenio ga je TCP/IP model. Mnogi mrežni inženjeri još se uvijek referenciraju prema OSI modelu zbog jednostavnije komunikacije i razumijevanja.

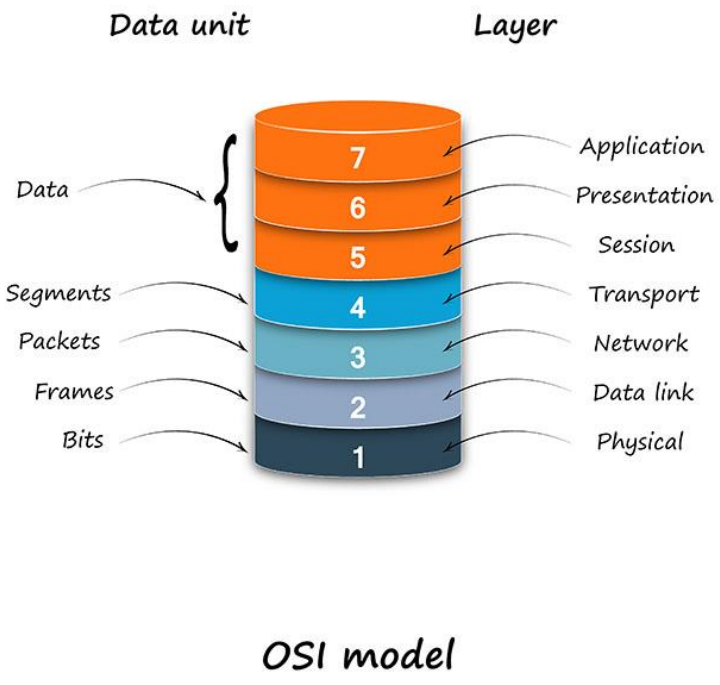
2.1. OSI model

Naglim razvojem mrežnih uređaja i mrežnih sustava 1970-ih godina potreba za standardizacijom i povezivanjem različitih mrežnih jedinica doveli su do nastanka OSI (Open Systems Interconnection) modela. OSI model razvila je međunarodna organizacija za standardizaciju, ISO (International Organization for Standardization) kako bi se zadovoljila potreba za interoperabilnošću među mrežnim uređajima koji se prije toga nisu mogli povezati jer nisu bili kompatibilni s različitim proizvođačima.

Stvaranje samog modela počelo je 1979. godine i prvi formalni prijedlog modela predstavljen je godinu poslije. Na početku su se razmatrali različiti načini kako pristupiti slojevima ili razinama, ali je na kraju prihvaćen model od sedam slojeva. Službena objava OSI modela za njegovu upotrebu predstavljena je 1984. godine kao ISO 7498 standard.

Standard 7498 definirao je sedam slojeva modela, no nije odredio pojedinačne protokole koji se koriste na svakom od slojeva. Takva fleksibilnost omogućila je da OSI model postane neovisan o različitim tehnologijama kako bi omogućio jednostavniju primjenu na mrežne sustave diljem svijeta. Iako je OSI model donio mnoge odlične promjene u svijetu tehnologije i mrežnih sustava nikada nije postao prevladavajući referentni model u produkciji. Uz razvoj OSI modela, razvijao se i TCP/IP (Transmission Control Protocol / Internet Protocol) model koji je u potpunosti preuzeo popularnost u akademskim i znanstvenim svrhama, pa tako i u komercijalnim.

OSI model i dalje posjeduje važnu ulogu u razumijevanju mrežnih sustava i komunikacija, te se mnogi različiti mrežni inženjeri i dalje referenciraju prema njemu. Slojevi prema kojima je raspoređen OSI model su: fizički sloj *(Physical layer),* sloj veze podataka *(Data-Link Layer)*, mrežni sloj *(Network layer)*, sloj prijenosa *(Transportation Layer)*, sloj sesije *(Session Layer)*, sloj prezentacije *(Presentation Layer)* i aplikacijski sloj *(Application Layer)* kako je prikazano na slici 2.1.



Slika 2.1 - OSI model – izvor: https://www.ionos.com/

Fizički sloj (eng. Physical Layer) najniži je sloj u OSI modelu koji omogućava fizički prijenos bitova preko različitih medija, kao što su kablovi ili bežični signali. Fizički sloj zaslužan je za funkcionalne karakteristike kablova i prijenosnih uređaja. Uređaji koji spadaju u fizički sloj su: repetitori, „hub-ovi“, mrežne kartice i bežični uređaji. Fizički sloj usko djeluje sa slojem veze podataka, jer mnoge tehnologije, poput Ethernet-a koriste funkcionalnosti fizičkog sloja i sloja veze podataka.

Sloj veze podataka (eng. Data-link layer) odgovoran je za ispravan prijenos podataka između povezanih uređaja. Također uključuje upravljanje greškama, primjenu kontrola pristupa i samo adresiranje uređaja. Upravljanjem pristupom mediju sloj veze podataka sprečava moguće sukobe i kolizije u mrežnom prometu. Sloj veze podataka sprema podatke iz viših slojeva u okvire (eng. Frame) kako bi se podaci u obliku bitova mogli prenijeti preko fizičke žice. „Frame“, odnosno okvir podataka sadrži izvornu i odredišnu adresu hardvera (fizička adresa). Takve adrese jedinstveno identificiraju računalo unutar mreža, te su zadane od strane samog proizvođača uređaja, a nazivaju se MAC (Media Access Control) adrese. MAC adresa veličine je 48 bitova, te je podijeljena u 6 različitih okteta **(00:0a:83:b1:c0:8e)**. Sam proizvođač najčešće određuje prvih 3 bajta, koji se nazivaju OUI (Organizational Unique Identifier). Sam sloj veze podataka sastoji se od dva podsloja: Logička kontrola veze (eng. LLC – Logical Link Control) i kontrole pristupa mediju (eng. MAC – Media Access Control). LLC podsloj funkcionira kao posrednik između fizičke veze i protokola viših slojeva, te omogućava protokolima poput Internet Protokola da ispravno rade bez obzira na korištenu fizičku tehnologiju. MAC podsloj upravlja pristupom fizičkom mediju, te služi kao posrednik ako se veći broj uređaja natječe za jednaku fizičku vezu. Kako bi to postigao, ethernet koristi tehnologiju CSMA/CD (Carrier Sense Multiple Access with Collision Detection). Uobičajene tehnologije koje koristi sloj veze podataka uključuju Ethernet, koja je najčešća tehnologija veze podataka za lokalne mreže, FDDI (Fiber Distributed Data Interface) tehnologija, 802.11 standard za bežične mreže, „Frame-Relay“ tehnologiju i ATM (Ansynchronous Transfer mode) tehnologiju. Uređaji koji spadaju u drugi sloj, odnosno sloj veze podataka su „prekidači“ (eng. Switch) i „mostovi“ (eng. Bridge).

Mrežni sloj zaslužan je za usmjeravanje podatkovnih paketa kroz mrežu unutar pa i izvan lokalnih mreža. Treći sloj, odnosno mrežni sloj određuje najpogodniji put za prijenos podataka od izvornog do odredišnog klijenta ili servera. Mrežni sloj koristi IP adrese kako bi identificirao različite uređaje na mreži. Postoje IP verzija 4 tehnologija (IPv4) koja koristi IPv4 adrese i IP verzija 6 tehnologija (IPv6) koja koristi IPv6 adrese. Sam IP sustav vrlo je veliki pojam koji će se detaljno opisati u kasnijem poglavlju.

Sloj prijenosa (eng. Transport layer) četvrti je sloj OSI modela. Unatoč svome nazivu, on zapravo ne prenosi podatke već je zaslužan za pouzdan prijenos, odnosno odgovoran je osigurati da podaci stignu na odredište u cijelosti i u ispravnom redoslijedu. Postoje dvije kategorije komunikacija na sloju prijenosa, „Connection-oriented“ (Uspostava veze) i „Connectionless“ (Bez uspostave veze). Uspostava veze zahtjeva da se prije slanja podataka mora uspostaviti veza prema dogovorenim parametrima, dok komunikacija bez uspostave veze ne zahtjeva uspostavu. Ključni protokoli na sloju prijenosa su TCP (Transmission Control Protocol) s uspostavom veze i UDP (User Datagram Protocol) bez uspostave veze. Protokoli koji zahtijevaju uspostavu veze pružaju dodatne funkcionalnosti kao što su segmentacija i numeriranje, potvrde o primitku podataka i kontrolu protoka pomoću veličine prozora (Window-size) koji pregovara o količini prijenosa podataka.

Sloj sesije koji je peti sloj OSI modela, omogućuje uspostavu, održavanje i prekid veze/sesije između komunikacija različitih aplikacija. Peti sloj zaslužan je za ispravnost sinkronizacije podataka, te kako bi se zadržala sesija pri prekidu veze. Tri su različite kategorije komunikacija u sloju sesija, puni-dupleks (eng. Full-Duplex), odnosno istovremena dvosmjerna komunikacija, polu dupleks (eng. Half-Duplex) gdje se dešava dvosmjerna komunikacija ali ne u istom vremenu i „Simplex“ – jednosmjernu komunikaciju.

Sloj prezentacije zaslužan je za format i sintaksu korisničkih podataka za aplikacijski sloj. Šesti sloj, odnosno sloj prezentacije omogućuje da podaci koje šalje jedna aplikacija mogu biti razumljivi drugoj aplikaciji koja prima podatke. U sloj prezentacije spadaju razni popularni standardi za formatiranje podataka, kao što su to slike, videozapis, tekst i zvuk.

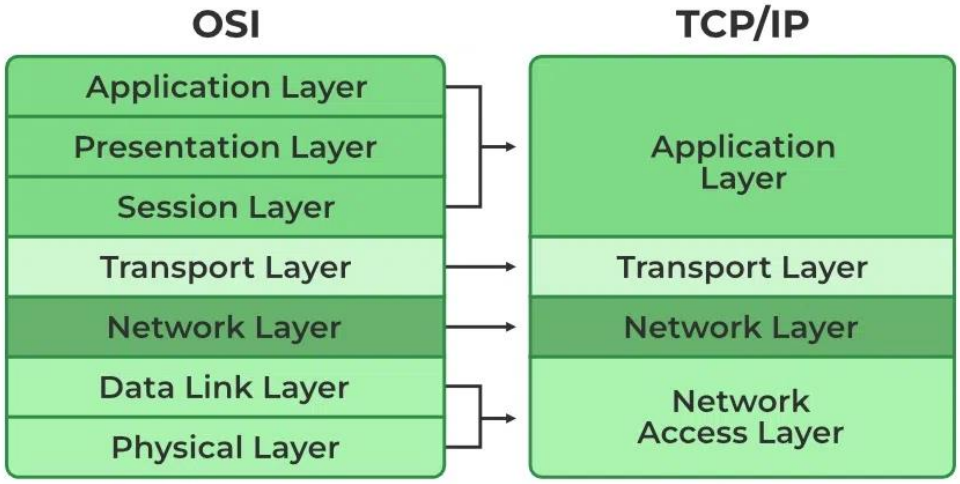
Neki od formata su: ASCII ili RTF za tekst, GIF, JPG i JPEG za slike, MIDI, MP3 ili WAV za zvuk ili MPEG, AVI i MOV za filmove. U slučaju da različiti uređaji ne podržavaju jednak format, sloj prezentacije pruža uslugu konverzije na format pogodan oba uređaja. Uz konverziju, sloj prezentacije može izvršiti kompresiju podataka prema potrebi, u nekim slučajevima čak i enkripciju.

Aplikacijski sloj je sedmi i najviši sloj OSI modela. Aplikacijski sloj pruža sučelje između aplikacije i mreže, kao što su to web preglednici ili sustavi za elektroničku poštu. Vrlo je bitno znati da se sama aplikacija ne nalazi u aplikacijskom sloju, već protokol koji je zaslužan za njihovu komunikaciju. Korisnik preko grafičkog sučelja komunicira s aplikacijom, a aplikacija s protokolom. Neki od protokola koji spadaju u sedmi, aplikacijski sloj su FTP (File Transfer Protocol), vrlo poznati HTTP i HTTPS (Hyper Text Transfer Protocol / Hyper Text Transfer Protocol Secure), POP3 (Post Office Protocol) i SMTP (Simple Mail Transfer Protocol) koji su zaslužni za elektroničku poštu. Sam aplikacijski sloj pruža razne usluge kao što su identifikacija komunikacijskih uređaja, dostupnost resursa i sinkronizaciju uređaja, te također usko djeluje s prezentacijskim slojem koji je ispod njega [1].

2.2. TCP/IP model

TCP/IP (Transmission Control Protocol/Internet Protocol) model je referentni model za mrežno povezivanje koji se razvio 1970-ih godina početkom ARPANET-a, prethodnika Interneta kakvog danas poznajemo. ARPANET je bio prvi veliki mrežni sustav koji je povezivao znanstvene institute i velika sveučilišta u Sjedinjenim Američkim Državama. Rastom ARPANET-a naglo je i porasla potreba za standardizacijom komunikacije između različitih uređaja. Uz suradnju s američkim ministarstvom obrane (DoD – Department of Defense) i mnoštvom znanstvenika s različitih sveučilišta nastao je TCP/IP referentni model. Razvoj TCP/IP modela nastavio se i kroz 1980-e godine, te je prolazio kroz mnoge promjene i napretke kako bi se uskladio s potrebama ARPANET-a i kasnije interneta.

Nastankom interneta TCP/IP postaje dominantni referentni model za mrežno povezivanje na globalnom tržištu. TCP/IP masivno se počinje koristiti u akademskim i znanstvenim ustanovama, velikom broju tvrtka i među potrošačima. U usporedbi s OSI modelom, TCP/IP ima vrlo praktičniju primjenu i bolju skalabilnost, te je postao globalno prihvaćeni zbog svoja četiri sloja. Slika 2.2 prikazuje slojeve koje koristi TCP/IP model: sloj pristupa mreži (eng. Network Access Layer / Link Layer), internetski sloj (eng. Internet Layer/Network Layer), sloj prijenosa (eng. Transport Layer) i Aplikacijski sloj (eng. Application Layer) [2].



Slika 2.2 - TCP/IP model i OSI model usporedba - preuzeto s https://www.geeksforgeeks.org/

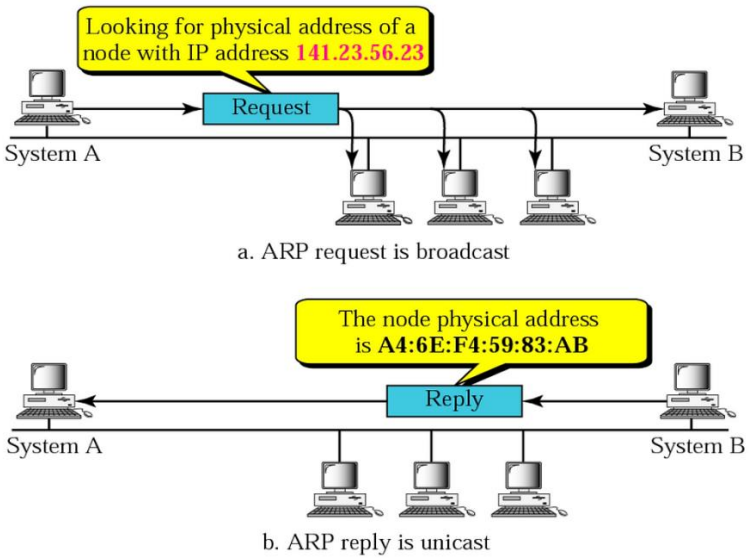
TCP/IP referentni model mrežnog povezivanja povezao je funkcije sedam slojeva OSI modela u četiri. Fizički sloj i sloj veze podataka spojeni su u TCP/IP sloj pristupa mreži (Network Access Layer), sloj mreže zadržao je svoje funkcije, no u modernom TCP/IP modelu često se naziva internetski sloj. Sloj prijenosa (Transport layer) zadržao je svoje funkcije i svoje ime. Tri viša sloja OSI modela, sesija, prezentacijski i aplikacijski u TCP/IP modelu spadaju u jedan sloj nazivom aplikacijski sloj. Aplikacijski sloj TCP/IP modela obavlja funkcije triju slojeva iz OSI modela. Zbog svoje fleksibilnosti postao je najefikasniji referentni model za daljnji razvoj mrežnih sustava.

Prvi i najniži sloj TCP/IP modela naziva se sloj pristupa mreži ili u nekim slučajevima samo sloj veze ili sloj veze podataka (eng. Link-Layer/Data-Link Layer). Njegova je zadaća povezivanje mrežnih sustava s fizičkim medijima i prijenos podataka preko istih. Sloj pristupa mreži obuhvaća funkcije dvaju slojeva, te zbog toga upravlja i načinom pristupa medijima i fizičkim adresiranjem mrežnih uređaja. Prvi sloj obuhvaća različite protokole, koji ovise o vrsti korištene mreže ili sustava. Najpoznatiji takav protokol je Ethernet koji se koristi za žičano povezivanje. Za bežično povezivanje koristi se Wi-Fi, a za direktne veze između dvaju uređaja koristi se PPP (Point-to-Point Protocol).

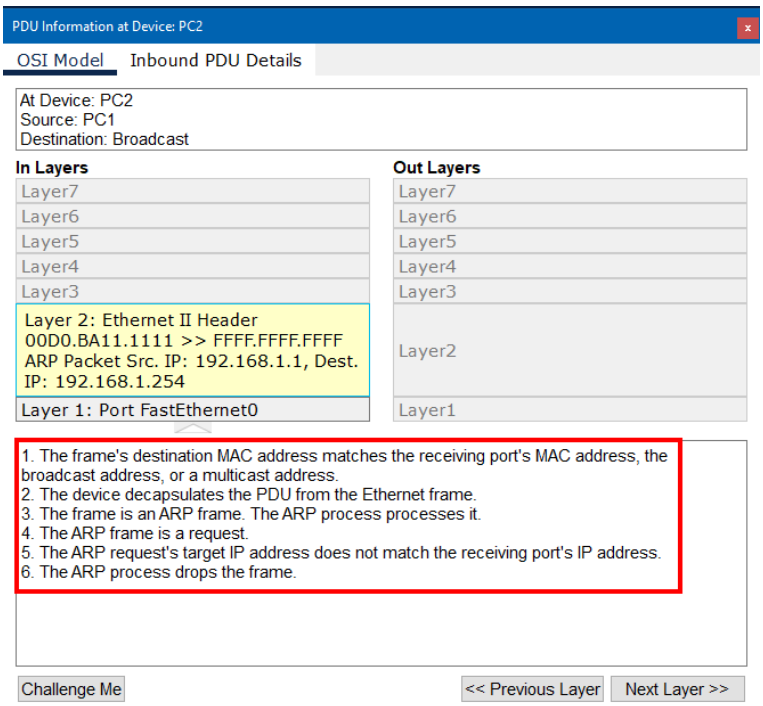
Drugi sloj često se naziva internetski sloj ili ponekad samo mrežni sloj po prijašnjem sloju iz OSI modela. Internetski sloj zaslužan je za usmjeravanje podatkovnih paketa kroz mreže i osigurava isporuku poslanog paketa na njegovo odredište, također služi kako bi usmjerio pakete izvan lokalne mreže u neke druge, bez obzira na udaljenost izvora i odredišta. Internetski sloj vrši usmjeravanje uz pomoć specifičnih mrežnih uređaja imenom „usmjerivači“ (eng. Router). Takvi uređaji stvaraju vlastite tablice usmjeravanja prema protoku podataka, kako bi lakše i brže odredili sljedeći skok (eng. Next Hop) i usmjerili paket prema odredištu. Ključni protokoli koji se nalaze na ovome sloju su ARP (Adress Resolution Protocol) i ICMP (Internet Control Message Protocol) koji služi za rješavanje problema vezanih uz pakete i njihov prijenos.

ARP (Adress Resolution Protocol) ključan je mrežni protokol koji se koristi u mrežnim sustavima kako bi se povezale logičke (IP) adrese s fizičkim (MAC) adresama uređaja, zbog lakšeg prijenosa podataka. ARP je potreban kako bi se omogućila efikasna komunikacija između uređaja unutar iste mreže. Uređaji na mreži stvaraju vlastite ARP tablice koje služe kao popis IP adresa i njihovih pripadajućih fizičkih (MAC) adresa. Obično kada se mreže prvi puta podignu, ARP tablice su prazne, te računalo šalje ARP zahtjev (eng. ARP request) kroz lokalnu mrežu kako bi pronašlo odgovarajuću adresu kao što je prikazano na slikama 2.3 i 2.4. Kada se tražena adresa pronađe, odredišno računalo šalje ARP odgovor (eng. ARP reply) koji sadrži vlastitu IP i MAC adresu. Nakon ARP odgovora, pošiljatelj dodaje traženu IP adresu u svoju ARP tablicu, kako bi pri sljedećem

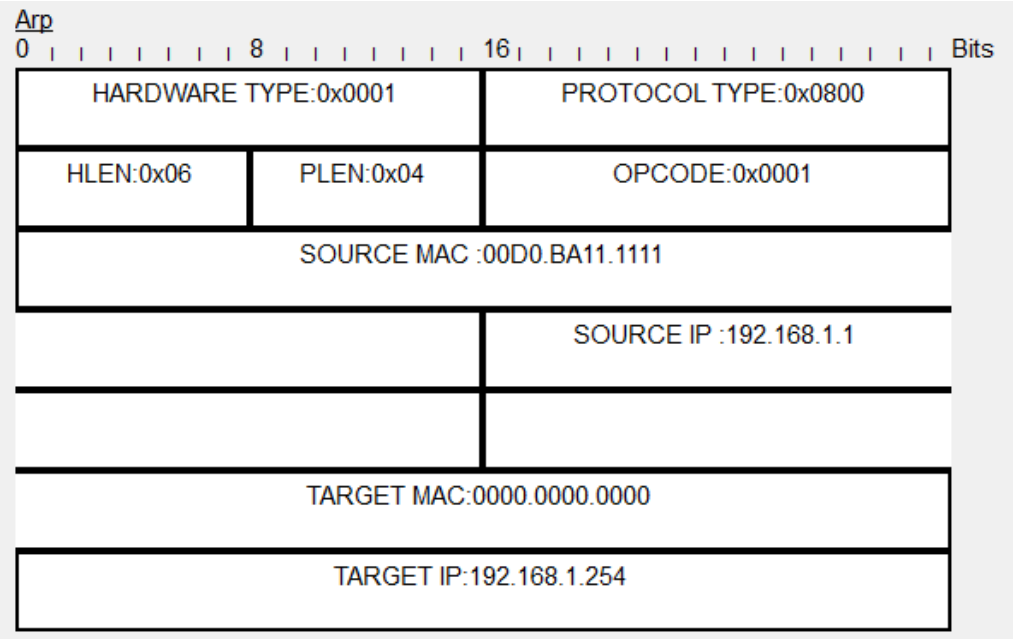
prijenosu podataka mogao brže pronaći konačno odredište. Podatkovna jedinica (PDU) ARP protokola prikazana je na slici 2.5.



Slika 2.3 - ARP - izvor: https://www.hackers-arise.com/



Slika 2.4 - ARP zahtjev - izvor: autor - Cisco Packet Tracer



Slika 2.5 - ARP PDU - izvor: autor - Cisco Packet Tracer

Treći sloj, imenom sloj prijenosa (eng. Transport Layer) osigurava pouzdanu komunikaciju između aplikacija na različitim računalima. Podaci na sloju prijenosa prenose se u obliku segmenata. Dva ključna protokola koja se nalaze na sloju prijenosa nazivaju se TCP (Transmission Control Protocol) prema kojem je i referentni model dobio i ime, te UDP (User Datagram Protocol). TCP pouzdani je protokol koji osigurava da svi segmenti stignu na odredište, te da nisu oštećeni tijekom prijenosa, dok je UDP nepouzdani protokol koji se koristi kada je bitna brzina prijenosa, te ne osigurava pravilan redoslijed ili izgubljenost podataka. Kako bi aplikacije mogle komunicirati jedna s drugom, koriste specifične portove sloja prijenosa koji identificiraju podatke aplikacija.

Najviši, četvrti sloj TCP/IP modela je aplikacijski sloj. Aplikacijski sloj omogućuje pristupanje različitim uslugama i aplikacijama na internetu, uključujući i web preglednike, elektroničku poštu i još mnoge različite protokole kao što su SSH (Secure Shell), FTP (File Transfer Protocol) koji se koristi za prijenos datoteka ili DNS (Domain Name System) koji se koristi za prevođenje domena („google.com“) u njihove pripadajuće IP adrese (8.8.8.8) [3].

3. Mrežne tehnologije

Mrežne tehnologije obuhvaćaju različite funkcije, protokole, uređaje i arhitekture koje osiguravaju nesmetano povezivanje i komunikaciju između jedinica u mrežnim sustavima. Mrežne tehnologije počele su se razvijati paralelno s razvojem telekomunikacija i računala. Naglim porastom interneta, rasla je i potreba za velikim brzinama i pouzdanim prijenosom podataka što je uvelike utjecalo na napredak mrežnih tehnologija. Uređaji su postali sve brži i pouzdaniji. Mrežne tehnologije doživjele su nagli porast dolaskom Ethernet tehnologije koja je omogućila efikasniji razvoj različitih lokalnih mreža (eng. LAN – „Local Area Network) i mreža širih područja (eng. WAN – Wide Area Network). Početkom 2000-ih godina, nagli razvoj doživjele su i bežične (eng. Wireless) mreže koje su omogućile napredniju mobilnu telekomunikaciju i brži pristup internetu putem različitih uređaja, poput pametnih telefona. Tehnologije poput 3G, 4G i 5G mobilnih mreža postaju nezamjenjive u modernom svijetu.

3.1. Ethernet i Ethernet okvir (Ethernet Frame)

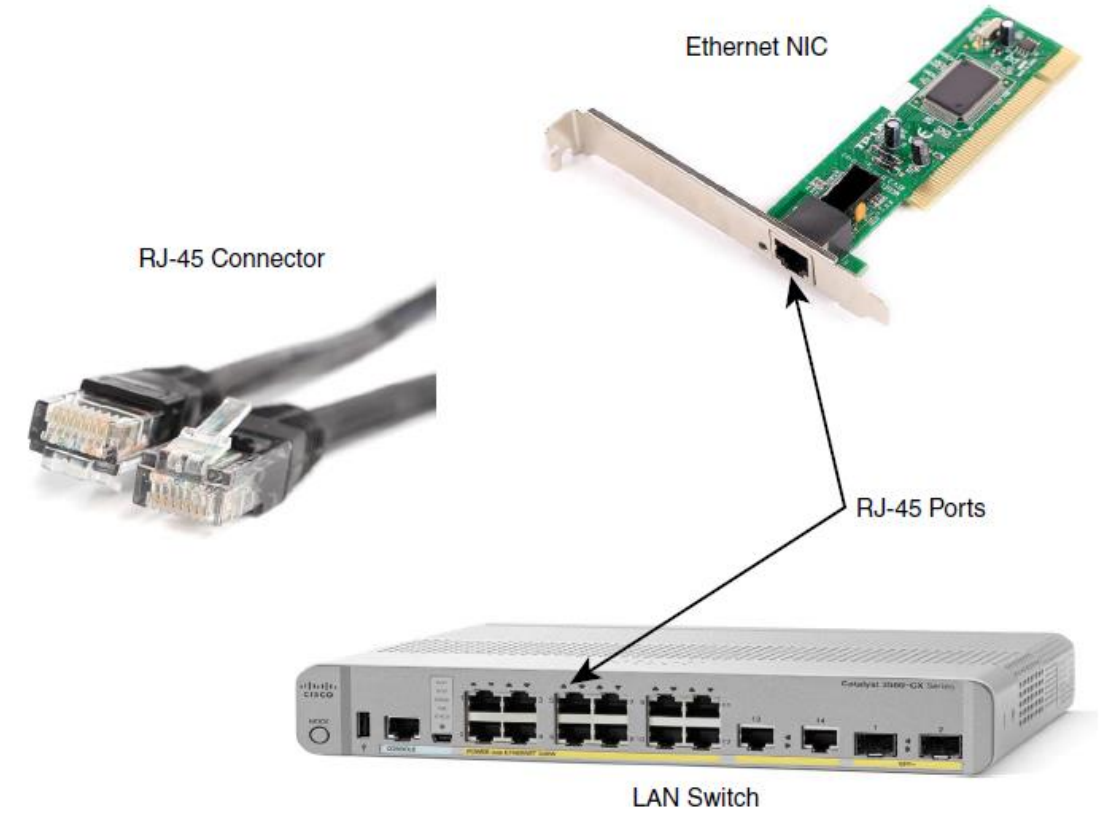
Sam izraz Ethernet odnosi se na skup standarda lokalnih mreža koji zajedno određuju fizički sloj i sloj veze podataka koje koristi najpopularnija žičana LAN tehnologija. Ethernet standarde odredio je Institut za elektrotehniku i elektroniku (eng. IEEE – Institute of Electrical and Electronics Engineers). Standardi uključuju vrste kablova, konektora od kojih je najpoznatiji „RJ-45“, pa sve do pravila mrežnih protokola kako bi se lakše primijenili Ethernet LAN-ovi u mrežne sustave. U današnje vrijeme Ethernet podržava vrlo veliki izbor različitih fizičkih Ethernet veza, te uključuje mnoštvo standarda za razne vrste bakrenih ili optičkih kablova. Brzine danas dostižu od 10 megabita po sekundi (eng. Mbps), pa sve do 400 gigabita po sekundi (eng. Gbps). Standardi se mogu razlikovati ovisno o vrsti ili duljini Ethernet kabla [3].

Izbor Ethernet kabla odnosi se na materijal koji se koristi unutar samog kabla za efikasniji prijenos bitova, a materijal može biti bakrena žica ili staklena vlakna. Mnogi uređaji koriste UTP (Unshielded Twisted Pair) kablove koji prenose podatke pomoću bakrenih žica. Postoje i optički kablovi koji se sve više koriste u današnjem svijetu i oni omogućuju slanje podataka svjetlošću putem staklenih vlakna. Optički kablovi puno su skuplji, no omogućuju puno veće udaljenosti, od 500 metara pa sve do 70 kilometara i više. Institut za elektrotehniku i elektroniku odredio je standarde fizičkog sloja Ethernet-a pod različitim nazivnim konvencijama. Formalna imena standarda počinju s 802.3, te poslije mogu slijediti sufiksna slova. Skraćena imena standarda također ukazuju na brzinu ili materijal koji koristi Ethernet veza.

Speed	Common Name	Informal IEEE Standard Name	Formal IEEE Standard Name	Cable Type, Maximum Length
10 Mbps	Ethernet	10BASE-T	802.3	Copper, 100 m
100 Mbps	Fast Ethernet	100BASE-T	802.3u	Copper, 100 m
1000 Mbps	Gigabit Ethernet	1000BASE-LX	802.3z	Fiber, 5000 m
1000 Mbps	Gigabit Ethernet	1000BASE-T	802.3ab	Copper, 100 m
10 Gbps	10 Gig Ethernet	10GBASE-T	802.3an	Copper, 100 m

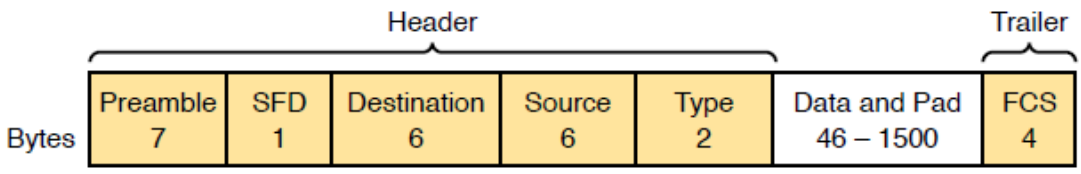
Slika 3.1 - Vrste Ethernet veza(kablova) - izvor: CCNA 200-301 Official Cert Guide Library, Wendell Odom

Ethernet veza odnosi se na bilo koji fizički Ethernet kabel između dvije jedinice. U Ethernet vezu spadaju sam kabel, konektori na njegovim krajevima i odgovarajuće utičnice (eng. Port) na uređajima koji primaju Ethernet veze. Standardi 10BASE-T i 100BASE-T koriste dva para žica, dok ostali zahtijevaju četiri para žica. Svaka žica ima svoju boju, a mnogi Ethernet UTP kablovi koriste „RJ-45“ konektore na svojim krajevima. Konektor „RJ-45“ posjeduje osam otvora nazivom „pin“ u koji ulaze svih osam žica Ethernet veza. Kako bi se pravilno povezali uređaji moraju posjedovati i „RJ-45“ Ethernet utičnice („Port-ove“) koji se podudaraju s „RJ-45“ konektorima. Mnoga računala sadrže Ethernet „RJ-45“ utičnice na svojim mrežnim karticama (eng. NIC – Network Interface Card). Slika 3.2 prikazuje RJ-45 utičnice i konektore na Cisco-vim uređajima.



Slika 3.2 - Ethernet Tehnologija - izvor: CCNA 200-301 Official Cert Guide Library, Wendell Odom

U računalnim mrežama, Ethernet okvir (eng. Ethernet Frame) je podatkovna jedinica protokla (eng. PDU – Protocol Data Unit) u sloju veze podataka. PDU preko Ethernet veze prenosi podatke s Ethernet okvirom kako bi mrežni uređaji mogli prenositi informacije vezane uz podatke koji se prenose preko lokalne mreže. Ethernet okvir sastoji se od zaglavlja na početku, zapakiranih podataka u sredini i Ethernet kraja (eng. Trailer). Slika 3.3. prikazuje izgled polja Ethernet zaglavlja.



Slika 3.3 - Ethernet okvir - izvor: CCNA 200-301 Official Cert Guide Library, Wendell Odom

Prvo polje Ethernet zaglavlja „Preamble“ sadrži uzorak od 56 bitova (7 bajtova) s izmjenjujućim nulama i jedinicama. „Preamble“ omogućuje mrežnim uređajima lakšu sinkronizaciju na razini bitova.

Nakon „Preamble-a“, drugo polje Ethernet zaglavlja je SFD (Start Frame Delimiter) koji pomaže u sinkronizaciji na razini bajtova, te označava kraj „Preamble-a“ i dolazak novog okvira. SFD služi kako bi prekinuo uzorak bitova „Preamble-a“ i na taj način označio njegov kraj i početak samog okvira. „Preamble“ binarni slijed izleda kao 10101010 10101010 10101010, te na kraju dolazi SFD čiji je binarni slijed 10101011.

Izvor (eng. Source) ili izvorna MAC adresa (eng. Source Mac Address) označava pošiljatelja okvira i veličine je 6 bajtova.

Odredište (eng. Destination) ili odredišna MAC adresa (eng. Destination Mac Address) označava primatelja okvira i veličine je 6 bajtova. Polja izvorne i odredišne MAC adrese imaju važnu ulogu u funkcionalnostima Ethernet lokalnih mreža. Pošiljatelj u polje izvorne Mac adrese stavlja svoju fizičku adresu, a u polje odredišta stavlja fizičku adresu primatelja i šalje okvir očekujući da će podaci stići na odredište.

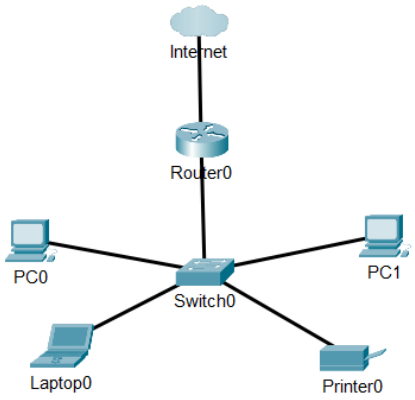
Polje tipa u Ethernet zaglavlju može se koristiti u dvije različite svrhe. Ako polje sadrži vrijednost manju od 1500 znači da označava veličinu podataka u bajtovima, a ako sadrži vrijednost veću od 1536 znači da se označava koji se protokol koristi u Ethernet okviru. Najčešće su to protokoli IP verzija 4 ili IP verzija 6.

„Data and Pad“ polje sadrži podatke iz višeg sloja, obično trećeg sloja (mrežni sloj) koji su najčešće IP verzija 4 ili IP verzija 6 paketi. Minimalna duljina polja koju pošiljatelj mora popuniti je 46 bajtova.

Ethernet kraj (eng. Trailer) naziva se „Frame Check Sequence“ i dolazi nakon zapakiranih podataka. „Frame Check Sequence“ koristi CRC (Cyclic Redundancy Check) algoritam kako bi uspio prepoznati oštećenja u cijelom okviru [3].

3.2. LAN (Local Area Network) i VLAN (Virtual Local Area Network)

Lokalna mreža je mreža koja se provlači kroz malo geografsko područje, kao što su to dom, ured, škole ili manji poslovni objekti. Jedna od najpopularnijih vrsta lokalnih mreža je SOHO LAN (Small Office/Home Office) koja koristi isključivo Ethernet LAN tehnologiju. Takvoj mreži potreban je mrežni uređaj nazivom Ethernet LAN prekidač (eng. Switch) koji pruža veliki broj fizičkih utičnica u koje se pripajaju Ethernet kablovi kao što je prikazano na slici 3.4.

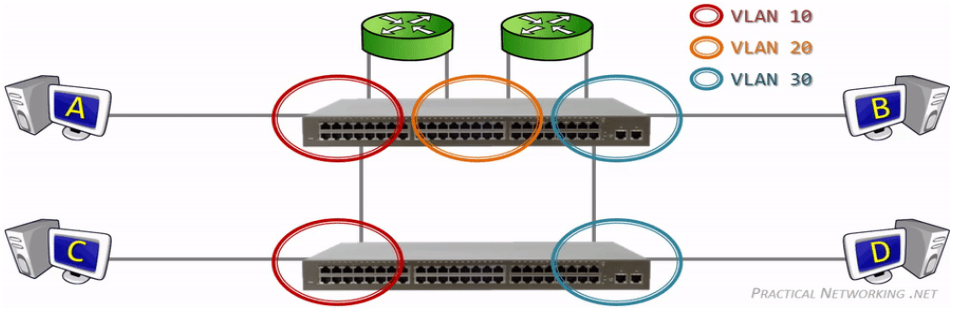


Slika 3.4 - tipična SOHO Ethernet mreža - izvor:autor - Cisco Packet Tracer

U mnogim modernim SOHO lokalnim mrežama prekidač (eng. Switch) i usmjerivač (eng. Router) čine jedan uređaj koji obično nazivamo „kućni ruter“. Mnogi proizvođači danas prodaju takve integrirane mrežne uređaje kako bi se olakšala implementacija samih mreža. Uz funkcije prekidača i usmjerivača, integrirani mrežni uređaji izvađaju još i druge korisne funkcije. Današnje SOHO mreže podržavaju i bežične LAN tehnologije, te je moguće koristiti i žičanu i bežičnu Ethernet LAN tehnologiju u istoj SOHO mreži. Institut za elektrotehniku i elektroniku definirao je standard 802.11 za bežične LAN mreže. Takve mreže koriste radio valove kako bi slale podatke u obliku bitova među uređajima. Kako bi se koristila bežična LAN tehnologija potrebno je u trenutnu mrežu dodati pristupnu točku (eng. Access Point). Pristupna točka mrežni je uređaj koji funkcionira slično

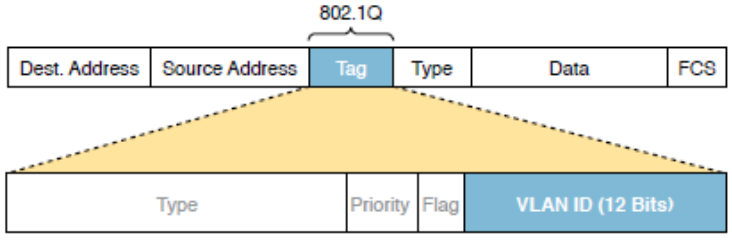
Ethernet prekidaču na kojeg se spajaju korisnici bežične mreže. Mnogi integrirani mrežni uređaji posjeduju i već integriranu pristupnu točku, što umanjuje potrebu za većom količinom različitih uređaja. Veća poduzeća i poslovni objekti imaju slične potrebe SOHO mrežama, no u većim omjerima. Ovisno o broju katova, takva poduzeća mogu imati jednu ili više posebnih prostorija u kojima se nalaze Ethernet prekidači. Kvalificirano osoblje provlači Ethernet veze/kablove iz takvih prostoriju prema ostalim prostorijama u kojima se nalaze uređaji kako bi se mogli povezati na lokalnu mrežu poduzeća. Veliki broj poslovnih objekata podržava i bežičnu 802.11 tehnologiju koja omogućava jednostavno povezivanje na mrežu kako bi se osoblje moglo kretati u prostorijama uz konstantnu povezanost. Koristeći Ethernet tehnologiju lokalne mreže moguće je povezati na mreže širokog područja (eng. WAN – Wide Area Network).

U lokalnu mrežu obično pripadaju svi uređaji koji se nalaze unutar jednake emitirane domene (eng. Broadcast Domain). U emitiranu domenu spadaju svi uređaji koji će primiti kopiju emitiranog okvira (eng. Broadcast Frame) koji je neki drugi uređaj na mreži poslao, što znači da je lokalna mreža jedna emitirana domena. Kada Ethernet prekidač na lokalnoj mreži primi okvir emitiranja na jednu utičnicu, prekidač će poslati okvir van na sve ostale utičnice osim na onu na kojoj je primljen. Zbog toga kako bi stvorili dvije lokalne mreže potrebno je bilo kupiti dva Ethernet prekidača, sve do dolaska virtualnih ili logičkih lokalnih mreža (eng. VLAN – Virtual Local Area Network). Korištenjem VLAN tehnologije, jedan Ethernet prekidač može stvoriti dvije ili više lokalnih mreža, odnosno domena emitiranja, na način da se utičnice prekidača rasporede u željene zone emitiranja kao što je prikazano na slici 3.5.



Slika 3.5 - Prijenos podataka preko virtualnih lokalnih mreža - izvor: practicalnetworking.net

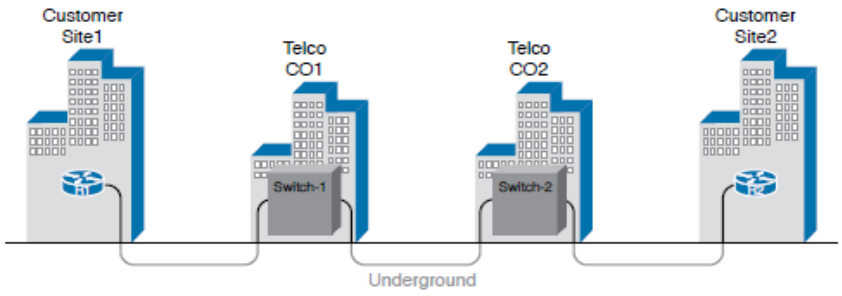
Implementacija virtualnih lokalnih mreža uvelike pomaže u poboljšanju mreže na razne načine. Emitirani okvir poslan od strane računala u VLAN-u 10 biti će primljen od svih ostalih računala ili uređaja u VLAN-u 10, dok uređaji u VLAN-ovima 20 i 30 neće primiti kopije okvira. Na taj način smanjuje se broj računala koja gube energiju obrađujući nebitne podatke, te se smanjuju sigurnosni rizici jer manji broj uređaja vidi okvire drugih uređaja. Uz nabrojene dvije prednosti, VLAN tehnologija smanjuje opterećenje procesora uređaja, te omogućava fleksibilniji raspored uređaja u mreži. Konfiguracija virtualnih lokalnih mreža na jednom prekidaču vrlo je jednostavna. Potrebno je svakoj utičnici priložiti pripadajući VLAN i konfiguracija je gotova. Ukoliko lokalna mreža koristi više Ethernet prekidača, proces konfiguracije je nešto zahtjevniji. U takvoj mreži potrebno je koristiti VLAN trunkiranje (eng. Trunking) na vezama između prekidača. Trunkiranje logički raspodjeljuje promet na vezama pomoću VLAN oznake (eng. VLAN tagging) koja je definirana 802.1Q standardom instituta za elektrotehniku i elektroniku, te se često naziva „dot1q“ oznaka (Slika 3.6). Prekidač koji šalje okvir na njega stavlja dodatno zaglavlje dot1q odmah nakon polja izvorne MAC adrese u Ethernet okviru (eng. Ethernet Frame). Dodana VLAN oznaka sadrži polje VLAN identifikatora (VLAN ID) koje omogućava prekidaču pošiljatelju povezivanje okvira s određenim VLAN identifikatorom, kako bi prekidač koji prima okvir mogao prepoznati VLAN kojem pripadaju podaci. Standard 802.1Q uvodi i poseban VLAN imenom izvorni VLAN (eng. Native VLAN) koji ne stavlja dodatnu VLAN oznaku na okvire. Izvorni VLAN omogućava konfiguracija jednog VLAN-a na prekidačima kao izvornog, bez oznake, što povećava brzinu prijenosa podataka. U slučaju da prekidač primi okvir koji ne sadrži oznaku, odmah zna da okvir i promet pripadaju izvornom VLAN-u [3].



Slika 3.6 - VLAN oznaka i njezina polja u Ethernet okviru - izvor: 200-301 CCNA Official Cert Guide Library, Wendell Odom

3.3. WAN (Wide Area Network)

Tehnologije mreža širokog područja (eng. WAN – Wide Area Networks) označuju standarde prvog fizičkog sloja i protokole koji su potrebni za komunikaciju na velikim udaljenostima. Ključne dvije WAN tehnologije su najamne linije (eng. Leased Lines) i Ethernet mreže širokog područja (eng. Ethernet WAN). Ethernet veze širokog geografskog područja koriste iste protokole za prijenos podataka kao i Ethernet lokalne mreže, no koriste neke dodatne pogodnosti kako bi što efikasnije djelovale na velikim udaljenostima. Najamne linije koriste se već dugo vremena, no u danas postaju sve manje uobičajene. Usluge koje nude najamne linije dostavljaju podatke u oba smjera, logikom punog dupleksa i brzinom koja je unaprijed dogovorena. Najamne linije koriste dva para žica, od kojih je svaki par zaslužan za jedan smjer slanja prometa. Kako bi se stvorile veze masivnih duljina, najamne linije nisu samo jedan dugi kabel, već telekomunikacijske tvrtke koje su proizvele najamne linije stvaraju velike računalne mreže koje djeluju kao „kablovi“ između dviju velikih udaljenosti. Fizička infrastruktura takvih mreža sakrivena je od krajnjeg korisnika kao što je prikazano na slici 3.7. Poduzeća ili tvrtke koje koriste najamne linije plaćaju najamnine za njihovo korištenje. Najamne linije pružaju usluge isključivo prvog sloja, odnosno osigurava prijenos bitova između dviju točaka spojenih s najamnom linijom. Zbog tog su razloga mnoge organizacije stvorile standardne protokole sloja veze podataka kako bi se mogle kontrolirati najamne linije. Trenutno dva najpoznatija protokola koji se koriste za najamne linije su High-Level Dana Link Control (HDLC) i Point-to-Point Protocol (PPP).



Slika 3.7 - primjer povezanih najamnih linija - izvor: Cisco.com

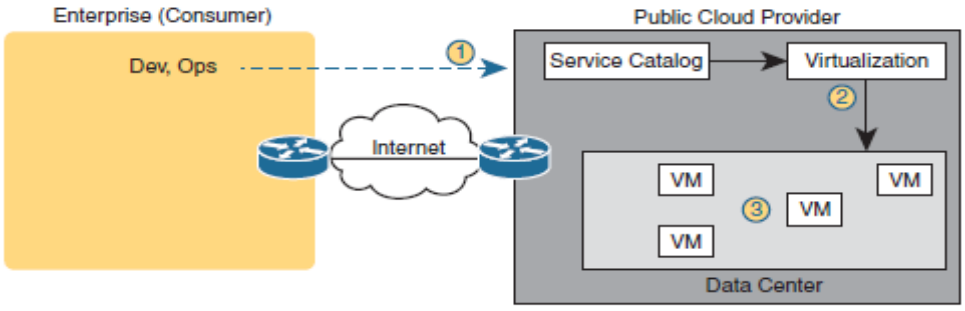
U početku Ethernet je bio primjeren isključivo za lokalne mreže. Duljine kablova bile su ograničene, te su mogle maksimalno podržavati lokalne mreže kampusa na udaljenostima od par kilometara. Naglim razvojem tehnologije, institut za elektrotehniku i elektroniku vrlo je brzo razvio Ethernet koji je podržavao i WAN tehnologiju. Danas postoje i optički kablovi koji mogu dostići duljine do čak 70 kilometara, te je Ethernet postao korisnija opcija za mreže šireg geografskog područja.

3.4. Virtualizacija i računalstvo u oblaku (Cloud Computing)

U današnje vrijeme sve više tvrtki stvara virtualni podatkovni centar, sav poslužiteljski hardver, procesorsku snagu i RAM (eng. Random Access Memory) stavlja i instalira u posebne prostorije i „ormare“ koji postaje kapacitet podatkovnog centra. Svaka komponenta operativnog sustava odvojena je od hardvera i postaje virtualna, zbog svoje odvojenosti. Hardver pokreće više instanci operativnog sustava, te se svaka takva instanca naziva virtualnom mašinom ili „virtualkom“ (eng. VM - Virtual Machine). Svaka virtualna mašina ima konfiguraciju koja određuje broj korištene procesorske snage i memorije. Na fizičkim poslužiteljima sustav virtualizacije pokreće virtualne mašine od kojih svaka koristi određeni dio procesorske snage i memorije podatkovnog centra. Kako bi virtualizacija mogla raditi, svaki fizički poslužitelj koristi „hipervizor“ (eng. Hypervisor). Hipervizor upravlja i dodjeljuje resurse hardvera fizičkim poslužiteljima. Mnoge tvrtke danas prodaju i potpune sustave virtualizacije, kao što su VMware, Microsoft HyperV ili Red Hat KVM. Takvi sustavi omogućuju inženjerima da mogu dinamički stvoriti virtualne mašine, pokretati ih ili premjestiti.

Računalstvo u oblaku pristup je pružanju IT usluga koristeći virtualizaciju. Korisniku se treba pružiti „samoposluga“ IT usluge, što znači da korisnik mora moći zatražiti i primiti uslugu bez čekanja, u realnom vremenu. Usluga mora biti dostupa bez potrebe razmatranja zahtjeva i potvrde. Nacionalni institut za standarde i tehnologiju (NIST) Sjedinjenih Američkih Država izdao je popis kriterija koje usluga računalstva u oblaku mora

poštovati. Popis je izveden iz same definicije o računalstvu u oblaku. Prvi bitan kriterij je „samoposluga“ na zahtjev korisnika, što znači da potrošač bira kada će početi ili prestati koristiti uslugu računalstva u oblaku bez da direktno komunicira s pružateljem usluge. Drugi kriterij je širok pristup mreži, što znači da usluga mora biti dostupna na različitim uređajima i mrežama, uključujući i internet. Treći kriterij je dijeljenje resursa, što znači da pružatelj usluge stvara zajednički spremnik resursa iz kojeg dinamično dodjeljuje resurse ovisno o potražnji potrošača. Četvrti kriterij je brza elastičnost kod kojeg se spremnik resursa potrošaču čini neograničen, te se zahtjevi mogu vrlo brzo procesirati. Posljednji kriterij je mjerenje usluge, što znači da pružatelj može mjeriti upotrebu oblaka i izvijestiti korisnika zbog transparentnosti u naplati. Mreže u oblaku imaju dvije grane, a to su privatni oblak (eng. Private Cloud) i javni oblak (eng. Public Cloud). Privatni oblak nudi uslugu unutar poduzeća ili tvrtke za interne korisnike, kako bi se to omogućilo tvrtka često mora proširiti alate za virtualizaciju, te dodati nove. Privatni oblak omogućuje internim korisnicima automatsko podizanje virtualnih mašina u roku od nekoliko minuta što osigurava platforme za razvoj aplikacija. U privatnom oblaku poslužitelj i korisnik su jednako poduzeće, što nije isto za javni oblak (eng. Public Cloud). Poslužitelj javnog oblaka je tvrtka koja nudi usluge oblaka drugim tvrtkama ili poduzećima. Poslužitelji takvih usluga podržavaju različite mogućnosti povezivanja mreža, te se povezuju s internetom kako bi korisnici i aplikacije mogli komunicirati s aplikacijama koje korisnik pokreće u mreži poslužitelja kako je prikazano na slici 3.8. Danas poslužitelji nude mnogo načina povezivanja, kao što su WAN tehnologije ili virtualne privatne mreže (eng. VPN – Virtual Private Network) [3].



Slika 3.8 - Usluga poslužitelja javnog oblaka kroz internet - izvor: 200-301 CCNA Official Cert Guide Library, Wendell Odom

4. Ključni mrežni protokoli

4.1. IP verzija 4 i IPv4 zaglavlje

Internet protokol verzija 4 je verzija IP-a koja se koristi za identifikaciju i adresiranje uređaja na internetu. Osnovni je protokol koji omogućuje uspostavljanje mrežne komunikacije između uređaja na internetu. Internet protokol koristi 32-bitne adrese koje se sastoje od četiri okteta odvojenih točkama (192.168.1.20). IP adrese koriste se kako bi uređaji imali jedinstvenu identifikaciju na internetu i mogli razgovarati jedni s drugima. Vrlo bitna zadaća Internet protokola verzije 4 je dodavanje novog IPv4 zaglavlja na okvire koji putuju kroz mrežu. IPv4 zaglavlje omogućuje uređajima efikasnije usmjeravanje podataka kroz i izvan IP mreža.

IPv4 header format

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
⋮	⋮																																
56	448																																

Slika 4.1 - izgled IPv4 zaglavlja na okvirima - izvor: geeksforgeeks.org

Slika 4.1 prikazuje polja IPv4 zaglavlja. Prvo polje IPv4 zaglavlja je verzija (eng. Version) koje je veličine 4 bita i označava verziju protokola, koja u slučaju IPv4 zaglavlja uvijek iznosi 4 (binarno 0100).

Drugo polje naziva se duljina zaglavlja (eng. IHL - Internet Header Length) i označava samu duljinu zaglavlja u 32-bitnim „riječima“. Veličina polja je 4 bita, a najmanja vrijednost koju može podržavati je 5 „riječi“ koje označavaju duljinu zaglavlja od 160 bitova (5 x 32 = 160 bitova). Najveća vrijednost koju polje podržava je 15 što označava duljinu zaglavlja od 480 bitova, odnosno 60 bajtova (15 x 32 = 480 bitova).

Treće polje je diferencirana točka koda usluge (eng. Differentiated Services Code Point), koje je je originalno definirano kao tip usluge (eng. ToS – Type of Service), te se koristi u kvaliteti usluge (eng. Quality of Service) kako bi označilo prioritet različitog prometa i njegovu kvalitetu. Veličine je 6 bitova.

Četvrto polje naziva se eksplicitna obavijest o zagušenju (eng. Explicit Congestion Notification) i služi kako bi pružilo obavijesti ako dođe do zagušivanja prometa bez da pritom mora odbaciti pakete i veličine je samo 2 bita.

Peto polje je ukupna duljina (eng. Total Length) veličine 16 bitova koje označava ukupnu duljinu cijelog IP paketa, uključujući i zaglavlje i podatke.

Šesto polje naziva se identifikator (eng. Identification) i veličine je 16 bitova. Koristi se u slučaju fragmentacije prevelikih paketa kako bi se mogli identificirati jednaki fragmenti istog paketa.

Sedmo polje naziva se zastave (eng. Flags) i veličine je 3 bita. Koristi se za kontrolu fragmenata paketa i koristi tri različite zastave. Bit 0: uvijek rezerviran, uvijek iznosi nula, Bit 1: „Don't Fragment“ koji označava da se paket ne smije fragmentirati i Bit 2: „More Fragments“ koji označava dolazak novih fragmenta iza trenutnog.

Osmo polje je odmak fragmenta (eng. Fragment Offset) koje označava položaj fragmenta u odnosu na originalni paket. Veličina polja je 13 bitova i koristi se i za ponovno sastavljanje paketa.

Deveto polje naziva se vrijeme života (eng. Time to Live), te služi kako bi se spriječilo beskonačno putovanje paketa kroz mrežu. Vrijednost polja „Time to Live“ smanjuje se za jedan nakon svakog sljedećeg skoka, odnosno prolaska kroz usmjerivač, kada vrijednost dostigne nulu, paket se odbacuje. Veličine je od 8 bitova.

Deseto polje je protokol (eng. Protocol) i veličine je 8 bitova, označava korišteni protokol koji se koristi u podatkovnom dijelu IP paketa, kao što su TCP, UDP, ICMP ili drugi.

Jedanaesto polje je kontrolna suma (eng. Header Checksum) od 16 bitova. Koristi se za provjeru grešaka u zaglavlju.

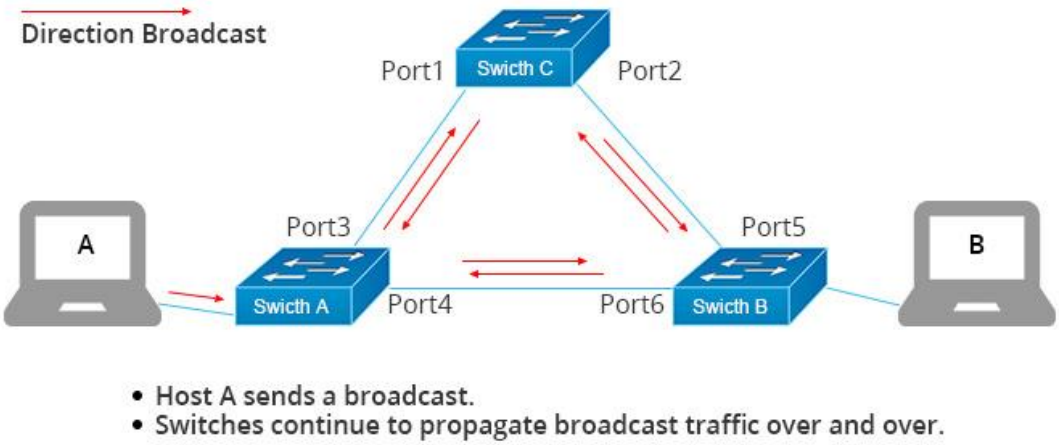
Dvanaesto i trinaesto polje su izvorišna i odredišna IP adresa, oboje su veličine 32 bita. Izvorišna IP adresa označava pošiljatelja podataka, dok odredišna IP adresa označava primatelja podataka.

Posljednje polje u IPv4 zaglavlju naziva se opcije (eng. Options). Opcionalno je polje koje može sadržavati dodatne informacije, kao što su vrijeme (Timestamps) ili zabilježene rute. Obzirom da je opcionalno polje njegova veličina može iznositi 0 bitova ili maksimalno 40 bajtova, odnosno 320 bitova. Ako je vrijednost duljine zaglavlja veća od 5 (6 – 15) znači da u zaglavlju postoji polje opcija, te da se mora uzeti u obzir pri prijenosu [4, 5].

4.2. OSI sloj 2 protokoli

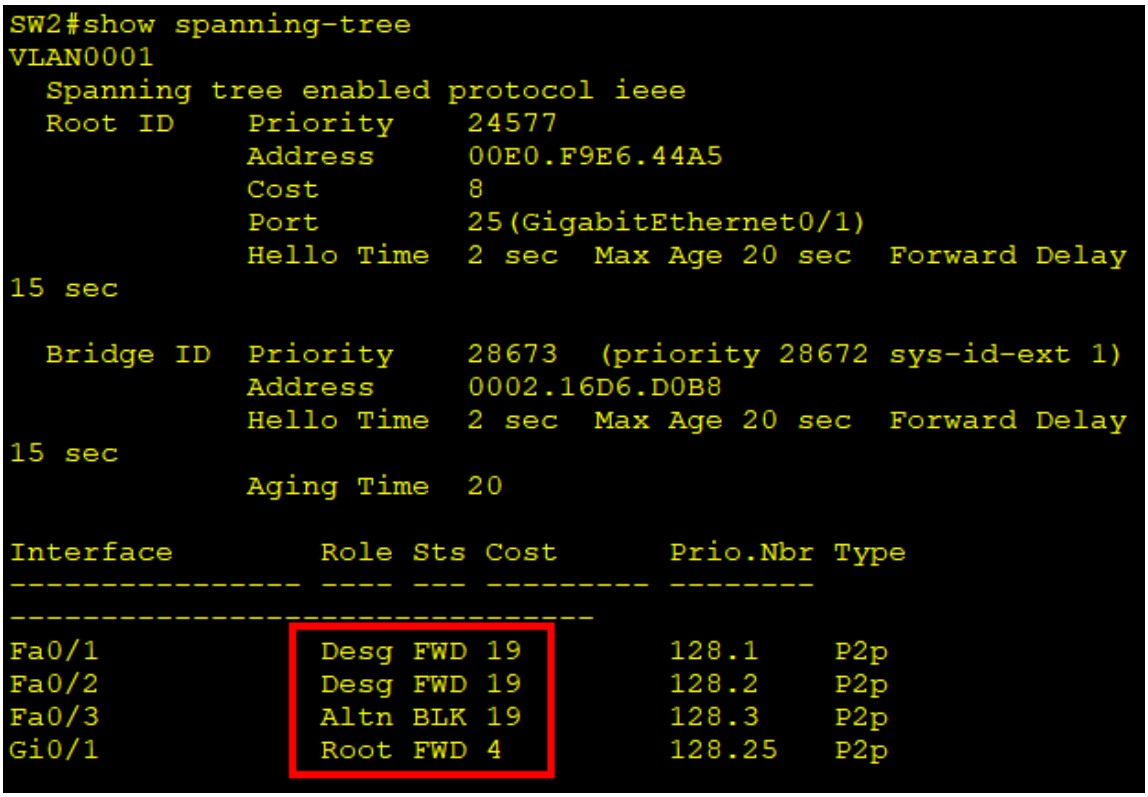
4.2.1. Spanning Tree Protocol (STP)

Spanning Tree Protokol nastao je kako bi spriječio tri uobičajena problema u Ethernet lokalnim mrežama. Sva tri problema proizlaze iz činjenice da bi se bez Spanning Tree protokola Ethernet okviri vrtjeli po mreži vrlo dugo pa čak i zauvijek ako bi uređaji zauvijek radili. Već jedan Ethernet okvir koji se vrti kroz mrežu uzrokuje pojavu nazivom „emitirajuća oluja“ (eng. Broadcast Storm). Emitirajuća oluja događa se kada Ethernet okvir bilo koje vrste beskonačno dugo vrti po lokalnoj mreži. Takve oluje zasićuju uređaje kopijama istog Ethernet okvira, što dovodi do znatno lošijih performansa jer računala obrađuju previše podataka. „Broadcast Storm“ događati će se tako dugo dok netko ne ugasi utičnicu ili sučelje ili ponovno pokrene prekidač. Isto tako emitirajuće oluje uzrokovati će nestabilnost MAC tablica (eng. MAC Table Instability), što znači da se tablice konstantno mijenjaju zbog nadolazećih okvira koji stižu na različite utičnice s jednakom MAC adresom.



Slika 4.2 - Broadcast Storm - izvor: fscommunity.com

Kako bi spriječio emitirajuće oluje, algoritam Spanning Tree protokola stvara jedinstveni put do i od svake Ethernet veze tako što odabire utičnice koje će biti postavljene u stanje prosljeđivanja, a ostala u stanju blokiranja. Spanning Tree određuje koje će utičnice prosljeđivati podatke, a to određuje prema tri kriterija. Prvo se bira glavni (eng. Root) prekidač na kojem su sve utičnice u stanju prosljeđivanja. Na ostalim se prekidačima odabire jedna od utičnica koja ima najmanji „glavni trošak“ (eng. Root Cost) između nje i glavnog prekidača. Odabrana utičnica postaje „glavna“ utičnica (eng. Root Port) i prelazi u stanje prosljeđivanja. Svaka veza može imati dva prekidača, te zbog toga prekidač s najmanjim „glavnim“ troškom (Root Cost) stavlja u stanje prosljeđivanja, a utičnica povezana s tim segmentom naziva se određeni port (eng. Designated Port). Uz stanja prosljeđivanja i blokiranja, Spanning Tree Protokol posjeduje još dva „prijelazna“ stanja (eng. Transitioning States) koja se nazivaju učenje (eng. Learning) i slušanje (eng. Listening). Nakon blokiranja utičnica prelazi u stanje slušanja, u ovom stanju utičnica ne prosljeđuje promet već ga samo „prisluškuje“ kako bi se pripremila za sljedeće stanje. Utičnica u stanju slušanja prima STP BPDU (Bridge Protocol Data Unit) poruke kako bi se uspostavila topologija mreže i spriječile petlje. Nakon slušanja utičnica prelazi u stanje učenja (eng. Learning) u kojem nastavlja prisluškivati promet, ali počinje i „učiti“ MAC adrese drugih uređaja u mreži kako bi ih spremila u svoje tablice [6].



Slika 4.3 - Spanning Tree Protocol, stanja utičnica - izvor:autor - Cisco CLI

4.2.2. Per – VLAN Spanning Tree Protocol (PVST+)

Per – VLAN Spanning Tree Protokol (PVST +) proizveo je Cisco, te ga je moguće koristiti isključivo na Cisco-vim uređajima. PVST je unaprijeđena verzija klasičnog Spanning Tree protokola koji se koristi u lokalnim mrežama. Proizveden je kako bi omogućio nezavisno održavanje različitih instanci protokola u svakoj virtualnoj lokalnoj mreži. Klasični Spanning Tree može održavati samo jednu instancu u lokalnoj mreži, neovisno o virtualnim lokalnim mrežama dok PVST + to može činiti za svaki zasebni VLAN. Na taj se način maksimalno iskorištavaju mrežni resursi i smanjuju opasnosti od zagušenja mreže. Svaka virtualna lokalna mreža posjeduje vlastite instance protokola koje djeluju u toj mreži kao klasični Spanning Tree protokol [3].

4.2.3. Rapid Spanning Tree Protocol (RSTP)

Napretkom tehnologije i mrežnih uređaja klasičan Spanning Tree protokol više nije bilo dovoljan kako bi zadovoljio potrebe velikih pa i manjih poslovnih objekata. Potreba za bržim performansama dovela je do stvaranja novog protokola imenom Rapid Spanning Tree protokol (RSTP). RSTP odlučuje o glavnom prekidaču (eng. Root Bridge) koristeći jednaka pravila kao i klasični STP. Glavne utičnice (eng. Root Ports) odabiru se također na jednake načine klasičnom STP-u, kao i određene utičnice (eng. Designated Ports). RSTP također koristi stanja prosljeđivanja i blokiranja, no u njegovom se slučaju blokiranje zove „odbacivanje“ (eng. Discarding). Klasični STP i RSTP su veoma slični protokoli, što omogućuje korištenje oba protokola u jednoj mreži. Prekidači koji podržavaju samo klasični Spanning Tree koristiti će STP, dok će moderniji uređaji koristiti funkcionalnosti Rapid Spanning Tree protokola. Glavni razlog stvaranja RSTP-a bila je brže podizanje protokola u slučaju promjena u mreži. RSTP prestaje koristiti prijelazno stanje slušanja (eng. Listening), te dodatno smanjuje vrijeme potrebno za prelazak s jednog stanja na drugo, što stvara veliku prednost nad klasičnim Spanning Tree protokolom kojem treba skoro minutu kako bi stanje došlo od blokirajućeg do prosljeđivanja, dok Rapid Spanning Tree treba samo desetak sekundi kako bi to dostigao. Uz brže vrijeme promjena stanja, RSTP uvodi dodatne funkcionalnosti kojima prekidač može zamijeniti trenutnu glavnu utičnicu (eng. Root Port) za neku drugu bez čekanja promjena stanja, te isto tako za određene utičnice (eng. Designated Port) [6].

4.2.4. Rapid Per – VLAN Spanning Tree Protocol (Rapid PVST +)

Rapid Per – VLAN Spanning Tree Protocol proizveo je Cisco, te se također može koristiti jedino na Cisco-vim mrežnim uređajima. Kao što to radi PVST, Rapid PVST također stvara zasebne instance protokola za svaku virtualnu lokalnu mrežu, što omogućuje nezavisno upravljanje svakom virtualnom mrežom, no on koristi Rapid Spanning Tree

protokol umjesto klasičnog. Rapid PVST pomaže pri boljem iskorištavanju mrežnih resursa i dodatno smanjuje vjerojatnost od stvaranja emitirajućih oluja (eng. Broadcast Storm). Rapid PVST koristi jednaka pravila i funkcionalnosti Rapid Spanning Tree protokola kako bi omogućio moderniju kontrolu mreže [3].

```
SW1#show spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID     Priority    32769
              Address     0005.5E4E.714B
              This bridge is the root
              Hello Time  2 sec  Max Age 20 sec  Forward Delay
15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     0005.5E4E.714B
              Hello Time  2 sec  Max Age 20 sec  Forward Delay
15 sec
              Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- --------
--------------------------------
Fa0/3            Back BLK 19        128.3    Shr
Fa0/1            Desg FWD 19        128.1    P2p
Fa0/2            Desg FWD 19        128.2    Shr
Fa0/24           Desg BLK 19        128.24   Shr
```

Slika 4.4 - Rapid Spanning Tree Protocol - izvor: autor - Cisco CLI, Cisco Packet Tracer

4.2.5. Multiple Spanning Tree Protocol (MSTP)

Najmoderniji oblik Spanning Tree protokola je Multiple Spanning Tree koji se obično koristi u velikim organizacijama. Nastao je kao protokol za upravljanje prometa na Ethernet mrežama koji omogućuje grupiranje više virtualnih lokalnih mreža u jednake instance Spanning Tree protokola kako bi se maksimalno optimizirala mreža. U organizacijama s velikim količinama virtualnih lokalnih mreža, loše je koristiti pojedinačnu instancu protokola za svaki VLAN jer to dovodi do prekomjernog korištenja mrežnih resursa. Zbog toga MSTP grupira slične virtualne mreže u jednake instance [3].

4.3. OSI sloj 3 protokoli

4.3.1. Open Shortest Path First (OSPF)

Open Shortest Path First (OSPF) je protokol usmjeravanja. On obuhvaća skup algoritama, pravila i poruka koje koriste usmjerivači (eng. Router) kako bi mogli stvoriti svoje tablice usmjeravanja i tako odredili najpovoljnije puteve kroz mreže. Takav proces uključuje analizu usmjerenog prometa i razmjenu informacija između usmjerivača. Svaki usmjerivač šalje posebne poruke kroz mrežu kako bi odredio topologiju mreže i pritom odredio najpovoljnije puteve do svake mreže ili podmreže koje pohranjuje u svoju tablicu usmjeravanja za lakšu orijentaciju pri sljedećem prijenosu podataka. Protokol usmjeravanja odnosi se na samu strukturu paketa i njegovu logičku adresu koji omogućuje prosljeđivanje paketa po mreži. Protokoli mogu se koristiti u IP verzija 4 mrežama i u IP verzija 6 mrežama. Putevi koje usmjerivači pohranjuju u svoje tablice mogu biti povezani, statički ili naučeni od strane dinamičkih protokola. Kako bi se stvorila jasna topologija mreže između usmjerivača, protokoli usmjeravanja omogućuju usmjerivačima učenje informacija o najboljim putevima do drugih podmreža od svojih susjednih usmjerivača. Isto tako usmjerivač može „reklamirati“ svoje puteve ostalim susjednim usmjerivačima, koji onda te informacije prosljeđuju dalje, te ukoliko se desi promjena na mreži usmjerivač može koristiti posebne poruke kako bi javio promjenu susjednim uređajima. Ukoliko usmjerivači nauče više različitih puteva prema istim odredištima, usmjerivač bira najbolji put prema posebnoj metrici, a najmanja vrijednost metrike je ujedno i najbolja. Različiti protokoli usmjeravanja koriste različite metrike kako bi izračunali najpovoljnije puteve, pa tako RIP (Routing Information Protocol) koristi ukupan broj sljedećih skokova i odabire najmanji za svoj najbolji put, dok EIGRP (Enhanced Interior Gateway Routing Protocol) koristi kompliciraniju formulu da bi odabrao najpovoljniji put do druge podmreže.

Jedna od najkorištenijih vrsta protokola usmjeravanja su protokoli temeljeni na stanju veze (eng. Link-State Routing Protocols) koji zahtijevaju konstantno oglašavanje svih informacija o mreži na svim usmjerivačima. Na taj način svi usmjerivači neke mreže posjeduju jednake informacije o istoj. Open Shortest Path First je najpoznatiji internetski protokol za usmjeravanje koji se temelji na stanju veze. OSPF sve svoje informacije organizira u posebnu bazu podataka koristeći posebne poruke LSA (Link-State Advertisements) koje šalje i preostalim usmjerivačima na mreži. Link-State Advertisement specijalne su podatkovne strukture koje nose informacije o topologiji mreže, te se spremaju u baze podataka imenom LSDB (eng. Link-State Database). Postupak oglašavanja LSA poruka omogućuje usmjerivačima kako bi posjedovali jednake kopije LSDB baza podataka, koje su izuzetno korisne, no one ne mogu izračunati najpovoljniji put za prijenos podataka do odredišta. Kako bi OSPF protokol mogao izračunati najpovoljniji put do odredišta, nizozemski računalni znanstvenik, Edsger Wybe Dijkstra osmislio je algoritam koji je i po njemu dobio ime Dijkstra's Shortest Path First algoritam. Algoritam računa najpovoljniji put tako da uzme unaprijed zadanu referentnu propusnost (eng. Reference Bandwidth), koju je moguće promijeniti i podijeli ju sa stvarnom propusnošću veze (eng. Link Bandwith), te dobije krajnji trošak (eng. Cost) kao što je prikazano formulom 4.1. Trošak koji rezultira dijeljenjem referentne propusnosti i propusnosti veze ne može biti manji od 1, te u slučaju posjedovanja bržih veza potrebno konfigurirati veću referentnu propusnost kako bi vrijednosti bile točne i kako bi se osigurala točnost pri odabiru najpovoljnijih puteva.

Trošak = ***Referentna propusnost / Propusnost veze*** **(4.1)**
(1 = 100 mbps / 100 mbps)

Ključna zadaća OSPF protokola je postizanje susjedstva s usmjerivačima sa svake strane kako bi se stvorile veze za razmjenu informacija iz LSDB baza podataka, te pohranjivanje najpovoljnijih puteva u tablice usmjeravanja. Slike 4.5 i 4.6 prikazuju „susjede“ i tablicu usmjeravanja na usmjerivaču. U malim mrežama kako bi se postavio OSPF, dovoljno ga je samo uključiti na svim usmjerivačima, no u većim mrežama koje koriste mnogo usmjerivača, potrebno je svrstati u različita OSPF područja (eng. OSPF area) kako bi OSPF mogao pravilno djelovati.

Dizajn OSPF područja mora poštovati dogovorena pravila kako bi funkcionalnost bila ispravna. Potrebno je odrediti utičnice koje će pripadati osnovnom (nultom) OSPF području (eng. Backbone Area – Area 0), te sve utičnice koje spadaju u jednaku podmrežu moraju biti u jednakom području. Utičnice usmjerivača ne smiju biti povezana s više od dva područja i sva područja koja nisu nulto područje moraju imati put do osnovnog (nultog) područja. Usmjerivači koji se nalaze na rubovima područja i povezuju dva područja, nazivaju se „granični usmjerivači“ (eng. Area Border Routers – ABR) [7].

```
R3#show ip ospf neighbor detail
 Neighbor 192.168.245.2, interface address 192.168.34.2
    In the area 0 via interface GigabitEthernet0/1
    Neighbor priority is 1, State is FULL, 6 state changes
    DR is 192.168.34.2 BDR is 192.168.34.1
    Options is 0x00
    Dead timer due in 00:00:36
    Neighbor is up for 00:01:33
    Index 1/1, retransmission queue length 0, number of
retransmission 0
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

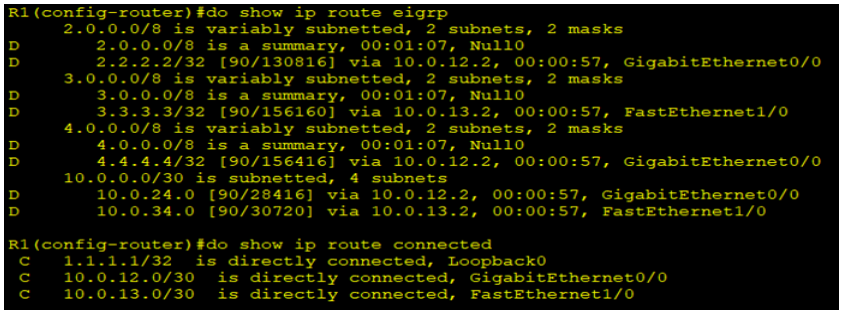
Slika 4.5 - OSPF "susjed" - izvor: autor - Cisco CLI, Cisco Packet Tracer

```
R2#show ip route
Codes: L - local, C - connected, S - static, 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

Gateway of last resort is not set

     192.168.34.0/30 is subnetted, 1 subnets
O       192.168.34.0/30 [110/2] via 192.168.245.2, 00:00:41, GigabitEthernet0/0
     192.168.245.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.245.0/29 is directly connected, GigabitEthernet0/0
L       192.168.245.1/32 is directly connected, GigabitEthernet0/0
```

Slika 4.6 - Tablica usmjeravanja koja prikazuje dodani OSPF najpovoljniji put - izvor: autor - Cisco CLI, Cisco Packet Tracer

4.3.2. Enhanced Interior Gateway Routing Protocol (EIGRP)

Enhanced Interior Gateway Routing Protocol izumio je Cisco. EIGRP je napredniji dinamički protokol usmjeravanja koji se koristi u IP mrežama. Nastao je kao unaprjeđenje IGRP (Interior Gateway Routing Protocol) protokola, te dodao mnoge modernije funkcionalnosti. EIGRP koristi funkcionalnosti „Distance Vector“ i „Link State“ protokola koje mu omogućuju puno brže podizanje i efikasnije upravljanje prometom i optimalnim putevima. Koristi algoritam koji računa na temelju propusnosti i kašnjenja (eng. Bandwidth and Delay) najpovoljniju rutu koju usmjerivači pohranjuju u svoje tablice usmjeravanja. EIGRP protokol usmjeravanja dijeli različite mreže na autonomne sustave (eng. Autonomous System) unutar kojih usmjerivače razmjenjuju tablice usmjeravanja samo s usmjerivačima iz istih autonomnih sustava, što dodatno pojačava performanse mreže i bolju iskorištenost resursa. EIGRP tablica usmjeravanja prikazana je na slici 4.7. Kako bi EIGRP zaštitio uređaje na mreži koristi SHA i MD5 ovjeru podataka kako bi spriječio neovlaštene ulaske u mrežu, gdje svaki usmjerivač provjerava valjanost informacija i identitet uređaja kako bi prihvatio vjerodostojne informacije usmjeravanja od ostalih mrežnih uređaja. Uz ovjeru i razdjeljivanje EIGRP podržava veoma brzo ažuriranje uređaja ukoliko se dogode promjene ili prekidi mreže, te je jedini protokol usmjeravanja koji omogućuje neuravnoteženo opterećenje između veza koje vode do istog odredišta. EIGRP razdjeljuje promet na više različitih puteva kako bi smanjio zagušivanje pojedinih veza ili mreže. EIGRP protokol često se koristi u srednjim i velikim mrežama zbog svojih naprednih mogućnosti [8, 9].



Slika 4.7 - EIGRP putevi u tablici usmjeravanja - izvor: autor - Cisco CLI, Cisco Packet Tracer

4.3.3. Dynamic Host Configuration Protocol (DHCP)

Dynamic Host Configuration Protocol jedna je od najčešće korištenih usluga na TCP/IP mrežama. Svaki korisnik koji se spaja na mrežu koristi DHCP protokol kako bi saznao potrebne IPv4 postavke, kao što je to IP adresa. Informacije se nalaze na DHCP poslužitelju i svaki korisnik uz pomoć posebnih DHCP poruka može saznati potrebne informacije kako bi se mogao koristiti mrežom, na taj se način trebaju ručno podesiti konfiguracije samo na poslužitelju, a ne na svakom korisniku posebno kao što je prikazano na slikama 4.8, 4.9 i 4.10. DHCP često dodjeljuje zamjenjive i privremene IP adrese kako bi se mogle ponovo koristiti u slučaju da neki od uređaja napusti mrežu. Dynamic Host Configuration Protocol također omogućuje laganu kretnju s uređajem jer korisnik može dobiti automatski potrebne informacije za mrežu bez da ih ručno upisuje pri svakom ulasku na novu mrežu. Na taj se način smanjuje vjerojatnost grešaka pri upisu podataka ili povezivanju na mrežu. Kada se korisnik prvi puta povezuje na mrežu ne posjeduje IP postavke. Korisnikov uređaj postaje DHCP klijent (eng. Client) koji od poslužitelja zatraži potrebne IP informacije kako bi se mogao povezati na mrežu. Poslužitelj korisniku dodjeljuje IPv4 adresu, masku, povoljne puteve i adresu DNS (Domain Name System) servera. Klijent i poslužitelj koriste četiri poruke kako bi proveli taj proces („DORA“).

Discover (hrv. Otkrij) – poruka poslana od strane klijenta kako bi mogao pronaći DHCP poslužitelja. Klijent šalje „Discover“ poruku na emitirajuću IP adresu (eng. Broadcast Address = 255.255.255.255) koja dostiže sve uređaje na toj mreži, tražeći poslužitelja DHCP informacija.

Offer (hrv. Ponudi) – poruka poslana od strane poslužitelja koja nudi slobodne IP adrese iz spremnika poslužitelja. Poruka ide direktno klijentu.

Request (hrv. Zatraži) – poruka poslana od strane klijenta kako bi preuzeo ponuđenu IP adresu [10].

Acknowledgement (hrv. Potvrdi) – Poslužitelj pohranjuje klijenta i adresu u spremnik.

```
R1(config-if)#
%DHCP-6-ADDRESS_ASSIGN: Interface GigabitEthernet0/0 assigned DHCP address 203.0.113.2,
mask 255.255.255.252, hostname R1


R1(config-if)#do show ip int br
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     203.0.113.2     YES DHCP   up                    up
GigabitEthernet0/1     192.168.1.1     YES manual up                    up
GigabitEthernet0/2     unassigned      YES unset  administratively down down
Vlan1                  unassigned      YES unset  administratively down down
```

Slika 4.8 - dodjela IP adrese klijentu - izvor: autor - Cisco CLI, Cisco Packet Tracer

```
R2#show ip dhcp binding
IP address       Client-ID/              Lease expiration        Type
                 Hardware address
203.0.113.2      0001.63B0.5601           --                     Automatic
```

Slika 4.9 - dodijeljenje adrese od strane DHCP - izvor: autor - Cisco CLI, Cisco Packet Tracer

```
R2#show ip dhcp pool

Pool POOL1 :
 Utilization mark (high/low)    : 100 / 0
 Subnet size (first/next)       : 0 / 0
 Total addresses                : 254
 Leased addresses               : 0
 Excluded addresses             : 3
 Pending event                  : none

 1 subnet is currently in the pool
 Current index        IP address range                    Leased/Excluded/Total
 192.168.1.1          192.168.1.1      - 192.168.1.254     0    / 3     / 254

Pool POOL2 :
 Utilization mark (high/low)    : 100 / 0
 Subnet size (first/next)       : 0 / 0
 Total addresses                : 254
 Leased addresses               : 0
 Excluded addresses             : 3
 Pending event                  : none

 1 subnet is currently in the pool
 Current index        IP address range                    Leased/Excluded/Total
 192.168.2.1          192.168.2.1      - 192.168.2.254     0    / 3     / 254

Pool POOL3 :
 Utilization mark (high/low)    : 100 / 0
 Subnet size (first/next)       : 0 / 0
 Total addresses                : 2
 Leased addresses               : 1
 Excluded addresses             : 3
 Pending event                  : none

 1 subnet is currently in the pool
 Current index        IP address range                    Leased/Excluded/Total
 203.0.113.1          203.0.113.1      - 203.0.113.2       1    / 3     / 2
```

Slika 4.10 - spremnici IP adresa na poslužitelju i dodjeljene adrese - izvor: autor - Cisco CLI, Cisco Packet Tracer

4.3.4. Network Address Translation Protocol (NAT)

Početkom interneta i nastankom IP adresa svako poduzeće moglo je otkupiti pojedine klasne IP adrese ovisno o broju korisnika. Nastale su posebne organizacije koje su kontrolirale i dodjeljivale IP adrese korisnicima, te je bilo potrebno osigurati da ni jedan korisnik ne dobije dupliciranu adresu. IP adrese stvorene su kao jedinstvene 32-bitne adrese, što je značilo da ih sveukupno postoji (2^{32} = 4 294 967 296). Naglim napretkom interneta i sve većim brojim uređaja na svijetu postalo je jasno već 90-ih godina prošlog stoljeća da će se sve IP verzija 4 adrese iskoristiti. Svijet je iskoristio sve svoje IP verzija 4 adrese 2019. godine, što je dovelo do stvaranja i korištenja IP verzija 6 adresi koje su 128-bitne, te je to jedino rješenje u budućnosti. Kako bi se spriječili trenutni problemi stvorene su privatne IP adrese i Network Address Translation protokol (NAT). Privatne IP adrese specijalne su adrese koje se koriste unutar privatnih računalnih mreža, kao što su kućne mreže, unutarnje poslovne mreže i lokalne mreže. Privatne IP adrese mogu biti jednake, te se ne razlikuju na globalnoj razini, te nisu izravno dostupne putem interneta, što znači da se uređaji spajaju na internet s različitom adresom od one privatne. Dijele se na dogovorena tri raspona.

Raspon 1: 10.0.0.0 – 10.255.255.255 → rezerviran za velike mreže.

Raspon 2: 172.16.0.0 – 172.31.255.255 → rezerviran za srednje mreže.

Raspon 3: 192.168.0.0 – 192.168.255.255 → rezerviran za male mreže.

Kako bi privatne mreže mogle stupiti u komunikaciju s internetom, potrebne su funkcionalnosti Network Address Translation protokola. NAT protokol omogućuje uređajima koji nemaju registriranu, važeću i globalno jedinstvenu IP adresu jednostavnu komunikaciju s internetom. NAT svojom funkcionalnošću „prevađanja“ (eng. Translation) omogućuje uređajima koji koriste privatne adrese da iste prevedu u važeću i registriranu globalnu IP adresu. Na taj način će globalno jedinstvena IP adresa predstavljati sve „unutarnje“ privatne IP adrese kako bi mogle pristupiti komunikaciji s internetom.

Kako bi NAT mogao pravilno funkcionirati, potrebno je konfigurirati pojedine IP adrese, ovisno o njihovom položaju u mreži kao što je prikazano na slikama 4.11, 4.12 i 4.13.

Unutarnja lokalna adresa: Lokalna (privatna) adresa uređaja koji se nalazi unutar tvrtke.

Unutarnja globalna adresa: Globalna adresa s kojom se uređaji povezuju na Internet preko NAT protokola.

Vanjska globalna adresa: Adresa uređaja izvan tvrtke.

Vanjska lokalna adresa: Korištenjem izvornog NAT protokola, adresa je jednaka vanjskoj, globalnoj adresi, a u slučaju korištenja odredišnog NAT protokola predstavlja uređaj izvan tvrtke dok njegovi paketi prolaze kroz lokalnu mrežu.

Postoje tri verzije NAT protokola koje se razlikuju u svojim funkcionalnostima. Različite se verzije koriste ovisno o potrebi mreže.

Statički NAT – najjednostavniji je oblik NAT protokola koji omogućuje direktno „jedan na jedan“ prevađanje unutarnje lokalne u unutarnju globalnu adresu. Prevađanje se događa „ručno“ od strane mrežnih inženjera.

Dinamički NAT – posjeduje neke sličnosti statičkom NAT protokolu kao što je prevađanje „jedan na jedan“ unutarnje lokalne u unutarnju globalnu adresu, no u ovoj verziji NAT protokola se dešava dinamički. Konfigurira se na način da se odrede skupovi raspoloživih unutarnjih globalnih adresa, te se definiraju kriteriji kako bi protokol znao koje unutarnje lokalne adrese treba prevesti.

PAT (Port Address Translation) / NAT „Overload“ – mnoge mreže trebaju omogućiti svim uređajima pristup internetu, što uvelike povećava broj potrebnih registriranih adresa. PAT ili NAT „Overload“ protokol omogućava prevođenje adresa na temelju „ulaza“ (eng. Port) aplikacije, na način da može koristiti jednake privatne adrese i razlikovati promet prema različitim „Port-ovima“ koje aplikacija koristi. NAT usmjerivač stvara NAT tablicu u koju pohranjuje povezane brojeve ulaza i adrese koje se koriste, kako bi mogao raspoznati koji promet pripada kojem uređaju [3].

```
R1(config)#int g0/1
R1(config-if)#ip nat ins
R1(config-if)#ip nat inside
R1(config-if)#int g0/0
R1(config-if)#ip nat ou
R1(config-if)#ip nat outside
R1(config-if)#exit
R1(config)#ip nat ?
  inside   Inside address translation
  outside  Outside address translation
  pool     Define pool of addresses
R1(config)#ip nat pool ?
  WORD  Pool name
R1(config)#ip nat pool POOL1 ?
  A.B.C.D  Start IP address
R1(config)#ip nat pool POOL1 100.0.0.1 ?
  A.B.C.D  End IP address
```

Slika 4.11 - NAT konfiguracija sučelja i stvaranje skupa globalnih adresa - izvor: autor - Cisco CLI, Cisco Packet Tracer (1)

```
R1(config)#ip nat pool POOL1 100.0.0.1 100.0.0.2 netmask 255.255.255.0
R1(config)#ip nat in
R1(config)#ip nat inside sour
R1(config)#ip nat inside source ?
  list    Specify access list describing local addresses
  static  Specify static local->global mapping
```

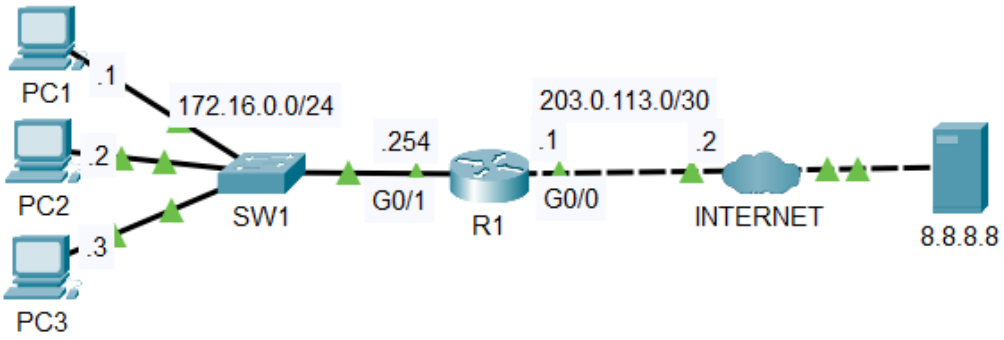
Slika 4.12 - NAT konfiguracija sučelja i stvaranje skupa globalnih adresa - izvor: autor - Cisco CLI, Cisco Packet Tracer (2)

```
R1(config)#access-list 1 permit 172.16.0.0 ?
  A.B.C.D  Wildcard bits
  <cr>
R1(config)#access-list 1 permit 172.16.0.0 0.0.0.255 ?
  <cr>
R1(config)#access-list 1 permit 172.16.0.0 0.0.0.255
R1(config)#ip nat ins
R1(config)#ip nat inside sou
R1(config)#ip nat inside source list
R1(config)#ip nat inside source list 1 ?
  interface  Specify interface for global address
  pool       Name pool of global addresses
R1(config)#ip nat inside source list 1 pool ?
  WORD  Name pool of global addresses
R1(config)#ip nat inside source list 1 pool POOL1 ?
  overload  Overload an address translation
  <cr>
R1(config)#ip nat inside source list 1 pool POOL1 overload ?
  <cr>
R1(config)#ip nat inside source list 1 pool POOL1 overload
```

Slika 4.13 - stvaranje liste kontrole pristupa za prevođenje u NAT protokolu i pokretanje PAT protokola - izvor: autor - Cisco CLI, Cisco Packet Tracer

```
R1#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 100.0.0.1:1       172.16.0.2:1       8.8.8.8:1          8.8.8.8:1
icmp 100.0.0.1:2       172.16.0.2:2       8.8.8.8:2          8.8.8.8:2
icmp 100.0.0.1:3       172.16.0.2:3       8.8.8.8:3          8.8.8.8:3
icmp 100.0.0.1:4       172.16.0.2:4       8.8.8.8:4          8.8.8.8:4
icmp 100.0.0.1:5       172.16.0.1:5       172.217.175.238:5  172.217.175.238:5
icmp 100.0.0.1:6       172.16.0.1:6       172.217.175.238:6  172.217.175.238:6
icmp 100.0.0.1:7       172.16.0.1:7       172.217.175.238:7  172.217.175.238:7
icmp 100.0.0.1:8       172.16.0.1:8       172.217.175.238:8  172.217.175.238:8
udp 100.0.0.1:1025     172.16.0.1:1025    8.8.8.8:53         8.8.8.8:53

R1#show ip nat sta
R1#show ip nat statistics
Total translations: 9 (0 static, 9 dynamic, 9 extended)
Outside Interfaces: GigabitEthernet0/0
Inside Interfaces: GigabitEthernet0/1
Hits: 11  Misses: 13
Expired translations: 4
Dynamic mappings:
-- Inside Source
access-list 1 pool POOL1 refCount 9
 pool POOL1: netmask 255.255.255.0
       start 100.0.0.1 end 100.0.0.2
       type generic, total addresses 2 , allocated 1 (50%), misses 0
```

Slika 4.14 - NAT tablica prevođenja i statistike korištenja - izvor: autor - Cisco CLI, Cisco Packet Tracer



Slika 4.15 - izgled simulirane mreže pri NAT (PAT) konfiguraciji - izvor: autor - Cisco Packet Tracer

4.4. OSI sloj 4 protokoli (prijenosni tok)

4.4.1. TCP (Transmission Control Protocol)

Dva ključna protokola četvrtog sloja OSI modela, odnosno prijenosnog sloja su TCP i UDP (User Datagram Protocol). Svaka TCP/IP aplikacija prema svojim zahtjevima odabire koristiti ili TCP ili UDP. TCP se bazira na trajanju veze i pruža oporavak od pogrešaka, no pritom troši više propusnosti od UDP-a. Transmission Control Protocol oslanja se na IP za krajnju isporuku podataka i rješavanju problema pri usmjeravanju, što znači da TCP izvodi samo dio funkcionalnosti potrebnih za isporuku i prijenos podataka između aplikacija. Svoje funkcije TCP izvodi jednako, bez obzira da li su dva uređaja na istoj Ethernet mreži ili su odvojena internetom. Oba protokola (TCP i UDP) koriste funkcionalnosti „multipleksiranja“ kako bi uređaj koji prima podatke znao predati primljene podatke određenoj aplikaciji. Kao i PAT protokol, TCP i UDP koriste brojeve ulaza aplikacija (eng. Port) kako bi promet stigao aplikaciji koja ga koristi. Multipleksiranje se temelji na konceptu koji se zove „Socket“ i sastoji se od tri ključna dijela – IP adresa, prijenosni protokol i broj ulaza. Ulazi aplikacija također se dijele na tri raspona, a sveukupno je 65 535 različitih „ulaza“ (eng. Port).

Dobro poznati, sistemski ulazi (eng. Well Known (System) Ports) – brojevi od 0 do 1023 koje koriste važne i često korištene aplikacije. Posjeduju mnogo stroži proces pri izdavanju novih ulaza. Tablica 4.1. prikazuje dobro poznate TCP ulaze.

Korisnički, registrirani ulazi (eng. User (Registered) Ports) – brojevi od 1024 do 49 151 koji posjeduju nešto slabiji proces pri izdavanju novih ulaza.

Privremeni, dinamički i privatni ulazi (eng. Ephemeral (Dynamic, Private) Ports) – preostali brojevi od 49 152 do 65 535. Ulazi koji se ne dodjeljuju, već su namijenjeni dinamičnoj alokaciji i privremenoj uporabi, ovisno o trajanju korištene aplikacije.

Popularne TCP aplikacije i njihovi korišteni dobro poznati ulazi

Broj ulaza	Protokol	Aplikacija
20	**TCP**	File Transfer Protocol (FTP) Data
21	**TCP**	File Transfer Protocol (FTP) Control
22	**TCP**	Secure Shell (SSH)
23	**TCP**	Teletype Network (Telnet)
25	**TCP**	Simple Mail Transfer Protocol (SMTP)
53	**TCP, UDP**	Domain Name System (DNS)
80	**TCP**	HyperText Transfer Protocol (HTTP)
110	**TCP**	Post Office Protocol 3 (POP3)
443	**TCP**	HyperText Transfer Protocol Secure (HTTPS) + Secure Socket Layer (SSL)

Tablica 4.1 - Dobro poznati TCP ulazi aplikacija - izvor: autor

DNS (Domain Name System) protokol koristi funkcionalnosti od oba prijenosna protokola (TCP i UDP). Ostale poznate aplikacije koriste TCP zbog svoje pouzdanosti i traženja potvrde veze prije samog prijenosa prometa [3].

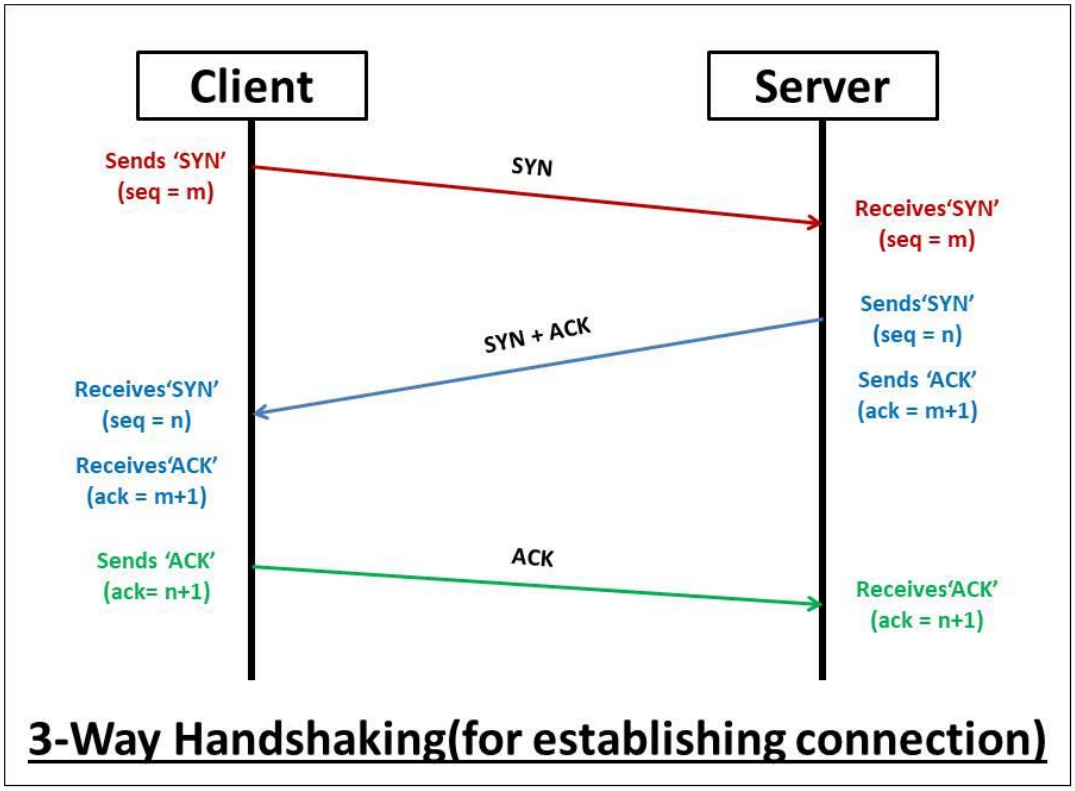
Prije nego što bilo koja svojstva Transmission Control protokola mogu pravilno prenositi promet, potrebna je potvrda uspostave veze između dviju točaka. Takav tok uspostave

veze naziva se „Three-Way Handshake“ jer koristi tri različite faze kako bi se potvrdilo pravilno djelovanje veze prikazano slikom 4.16.

Faza 1: **SYN** (Synchronize)

Faza 2: **SYN, ACK** (Synchronize and Acknowledge)

Faza 3: **ACK** (Acknowledge)



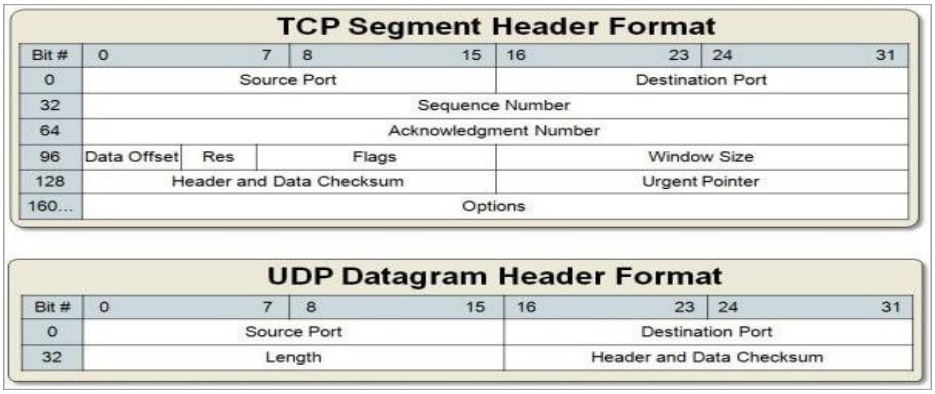
Slika 4.16 - proces sinkronizacije (Three-way Handshake) - izvor: afteracademy.com

TCP također kontrolira količinu protoka prometa koristeći koncept „prozora“ (eng. Window). Prozor omogućuje primatelju podataka da može obavijestiti pošiljatelja koliko podataka može primiti u nekom trenutku, na način da se uspori ili ubrza slanje podataka. Prozor na taj način može „kliziti“ gore i dolje, što se naziva „Sliding Window“ [11].

4.4.2. User Datagram Protocol (UDP)

User Datagram protokol pruža uslugu razmjene poruka između aplikacija. U usporedbi s Transmission Control protokolom, UDP počinje slati podatke odmah, bez prethodne potrebe za uspostavom veze. UDP ne pruža pouzdanost i ne posjeduje funkcionalnosti kao prozor, kako bi smanjio ili povećao količinu prijenosa podataka, te ne može segmentirati velike podatke za prijenos. Zbog nedostatka mnogo funkcionalnosti koje posjeduje TCP, UDP prenosi podatke većom brzinom i manjim opterećenjem. Protokol poput UDP-a povoljan je za aplikacije koje mogu tolerirati izgubljene podatke ili posjeduju svoje funkcionalnosti za oporavak. Prijenos audio prometa preko IP-a, VoIP (eng. Voice over IP – Discord, TeamSpeak, Skype) koristi isključivo UDP za svoje procese, jer kada bi koristio svojstva Transmission Control protokola izgubljeni audio podaci ponovno bi se slali što bi dovelo do velikih kašnjenja i potpuno nerazumljivog glasa. Uz VoIP, UDP koristi i Domain Name System (DNS) jer mu je potrebna veća brzina, a u slučaju lošeg DNS zahtjeva, korisnik će sam ponovno započeti proces. Dobro poznati ulazi UDP protokola prikazani su tablicom 4.2.

User Datagram protokol na okvire dodaje svoje jednostavno zaglavlje (Slika 4.17) koje je veličine 8 bajtova (64 bitova) u usporedbi s TCP-ovim 20-bajtnim zaglavljem. Zbog manjeg mjesta koje UDP zaglavlje zauzima, može dostići veće brzine i manje opterećenje od TCP-a [12].



Slika 4.17 - Razlike TCP i UDP zaglavlja - izvor: softwaretestinghelp.com

Broj ulaza	Protokol	Aplikacija
53	UDP, TCP	Domain Name System (DNS)
67	UDP	DHCP poslužitelj (Dynamic Host Configuration Protocol Server)
68	UDP	DHCP klijent (Dynamic Host Configuration Protocol Client)
69	UDP	Trivial File Transfer Protocol (TFTP)
161	UDP	Simple Network Management Protocol (SNMP)
514	UDP	System logging (Syslog)

Tablica 4.2 - Dobro poznati UDP ulazi aplikacija - izvor: autor

4.4.3. Real-time Transport Protocol (RTP)

Real-time Transport protokol stvoren je za prijenos multimedijskog sadržaja (audio i video) u realnom vremenu preko IP mreža. RTP osigurava pouzdanost i efikasnost kod prijenosa podataka kako bi se omogućila glatka reprodukcija i visoka kvaliteta sadržaja. Protokol usko surađuje s RTSP (Real Time Streaming Protocol) protokolom kako bi pružio kvalitetnu uslugu korisnicima. Real-time Transport Protocol zaslužan je za prijenos multimedijskog sadržaja i segmentaciju paketa na manje dijelove zbog lakšeg i bržeg prijenosa preko mreže. RTP isto tako omogućuje sinkronizaciju audio i video podataka kako bi se osigurala reprodukcija u realnom vremenu. Podržava kodeke različitih vrsta za kompresiju podataka kako bi se dodatno povećala učinkovitost prijenosa podataka, te ga je moguće koristiti u raznim mrežama, poput lokalnih, šireg geografskog područja i bežičnim mrežama. Koristi se vrlo često za VoIP, video konferencije, streaming usluge i video nadzor [13].

4.4.4. Real Time Streaming Protocol (RTSP)

Real Time Streaming protokol nastao je kako bi se lakše moglo upravljati prijenosom multimedijskog sadržaja kroz Real-time Transport protokol. Dok Real-time Transport protokol obavlja isključivo prijenos podataka, RTSP omogućuje korisnicima upravljanje nad reprodukcijom, pretraživanju i interaktivnim djelovanjem nad sadržajem. RTSP koristi TCP/IP funkcionalnosti za prijenos sadržaja [14].

5. IPv4 adresiranje i usmjeravanje

Internet protokol verzije 4 obuhvaća pet različitih razreda adresa (A, B, C, D i E) prikazanih u tablici 5.1. Prva tri razreda A,B i C predstavljaju „unicast“ adrese, dok razred D predstavlja „multicast“ adrese. Unicast adrese su IP adrese namijenjene za jednog primatelja, a multicast adrese služe kako bi prenijele podatke više primatelja od jednom. Razred E IP adresa na početku bio je namijenjen eksperimentalnim procesima, no kasnije su promijenjene na stanje rezerviranosti za buduću upotrebu. IP adresa veličine je 32 bita, te se sastoji od četiri okteta odvojenih točkama (10.125.200.1). Razred IP adresa može se odrediti prema prvom oktetu adrese.

Razred	Prvi oktet	Namjena
A	1 - 126	Unicast – velike mreže
B	128 - 191	Unicast – srednje mreže
C	192 - 223	Unicast – male mreže
D	224 - 239	Multicast
E	240 - 255	Rezervirano

Tablica 5.1 - IPv4 razredi adresa - izvor: autor

Razredi IP adresa definiraju fiksni prefiks koji označava broj mogućih uređaja u mreži. Razred A je najveći razred koji sadrži 8-bitni prefiks (10.0.0.0/8) koji označava rezervirani i nepromjenjivi mrežni identifikator (eng. Network ID). Obzirom da IP adresa razreda A ima prefiks 8, njezina je maska 255.0.0.0, što znači da su svih 8 bitova u prvom oktetu rezervirani, a preostalih 24 bita u 3 okteta mogu se koristiti za korisnike na mreži, te se oni nazivaju „host“ bitovi i prikazani su u tablici 5.2. Prema dogovorenom pravilu, prva je adresa u mreži rezervirana za identifikator mreže (eng. Network ID), dok je posljednja rezervirana za emitirajuću adresu (eng. Broadcast Address). Prema tome potrebno je od ukupnog broja adresa oduzeti dvije kako bi se dobila brojka raspoloživih adresa u mreži. Kako bi se izračunao broj raspoloživih adresa, koristi se prikazana formula 5.1.

$2^n - 2$ = broj raspoloživih adresa u mreži (5.1)

n → broj „host“ bitova

($2^{24} - 2 = 16\ 777\ 214$ raspoloživih adresa u IP adresi (10.0.0.0/8) razreda A)

Mrežni bitovi	„Host“ bitovi			Maska podmreže
10.	0.	0.	0	/8
11111111.	00000000	00000000	00000000	255.0.0.0

Tablica 5.2 - podjela IPv4 adrese razreda A - izvor: autor

Razred B ima prefiks /16, što znači da može imati 65 534 raspoloživih adresa, dok razred C ima prefiks /24 i raspoloživih adresa 256 – 2 = 254 .

Mrežni bitovi (eng. Network Bits), odnosno rezervirani bitovi adrese prikazuju koliko se različitih mreža može stvoriti iz jednog razreda. Za izračun broja raspoloživih mreža koristi se formula 5.2.

2^n = broj raspoloživih mreža (5.2)

($2^8 = 256$ raspoloživih mreža)

Razred A može koristiti 256 različitih mreža.

Rastom interneta i tehnologije i većom potrebom za IP adresama, razrede IP adresa zamijenio je standard imenom „CIDR“ (Classless Inter-Domain Routing) koji je nastao 1993. godine. CIDR je uveo varijabilnu veličinu mreža (Tablica 5.3), što je uvelike smanjilo veliki broj neiskorištenih adresa razreda (Organizacije koje su trebale više od 70 000 adresa, nisu mogle koristiti razred B, već su trebale koristiti razred A, što bi značilo da bi 16 707 216 adresa ostalo neiskorišteno. CIDR je uz pomoć varijabilnih veličina mreža omogućio dodjelu potrebnog broja adresa, što je osiguralo iskoristivost ostalih adresa i time produžio život IP verziji 4 adresama. Prefiksevi adresa sada mogu biti različiti, što omogućuje varijabilnu rezervaciju bitova i tako varijabilnu količinu raspoloživih adresa ili mreža [4].

Network Bits	Subnet Mask	Bits Borrowed	Subnets	Hosts/Subnet
8	255.0.0.0	0	1	16777214
9	255.128.0.0	1	2	8388606
10	255.192.0.0	2	4	4194302
11	255.224.0.0	3	8	2097150
12	255.240.0.0	4	16	1048574
13	255.248.0.0	5	32	524286
14	255.252.0.0	6	64	262142
15	255.254.0.0	7	128	131070
16	255.255.0.0	8	256	65534
17	255.255.128.0	9	512	32766
18	255.255.192.0	10	1024	16382
19	255.255.224.0	11	2048	8190
20	255.255.240.0	12	4096	4094
21	255.255.248.0	13	8192	2046
22	255.255.252.0	14	16384	1022
23	255.255.254.0	15	32768	510
24	255.255.255.0	16	65536	254
25	255.255.255.128	17	131072	126
26	255.255.255.192	18	262144	62
27	255.255.255.224	19	524288	30
28	255.255.255.240	20	1048576	14
29	255.255.255.248	21	2097152	6
30	255.255.255.252	22	4194304	2

Tablica 5.3 - Tablica podmreža - izvor: tutorialspoint.com

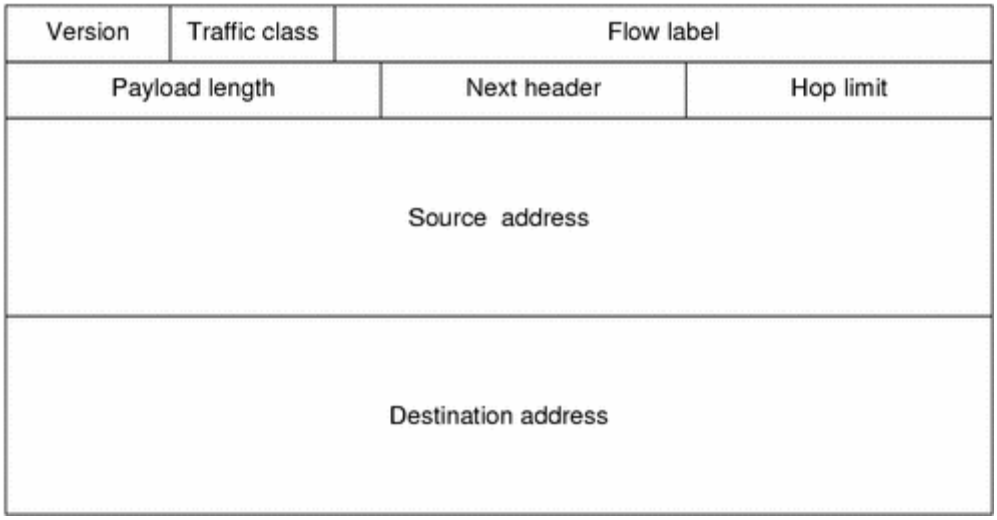
IP usmjeravanje proces je prosljeđivanja paketa kroz cijelu TCP/IP mrežu, od uređaja koji šalje podatke pa sve do uređaja koji će ih primiti. Postupak samog usmjeravanja oslanja se na funkcionalnosti sloja mreže (3. sloj) kako bi stvorio potrebne IP pakete i proslijedio ih na predviđena mjesta. Usmjeravanje se također oslanja i na neke funkcionalnosti prvog i drugog sloja OSI modela. Koristi Ethernet veze, serijske WAN veze i Ethernet LAN tehnologije, te bežične LAN mreže. Podaci se iz bitova enkapsuliraju u okvire sa zaglavljima korištenih protokola i prenose se preko TCP/IP mreže. Dinamički protokoli usmjeravanja odabiru najpovoljnije puteve usmjeravanja, no postoje i statički putevi, koje mrežni inženjeri trebaju unijeti ručno. Statički putevi konfiguriraju se u usmjerivačima pomoću globalne konfiguracijske naredbe **„ip route“**. Nakon naredbe ip route slijedi IP adresa odredišne mreže, obično mrežni identifikator (eng. Network ID) s prefiksom podmreže ili maskom. Uz ostale dijelove naredbe, potrebno je navesti upute za prosljeđivanje, kao što je izlazno sučelje ili adresa sljedećeg skoka, odnosno sljedećeg usmjerivača kao što je prikazano na slici 5.1. Informacije se pohranjuju u tablice usmjeravanja. Pohranjena statička ruta smatra se mrežnom rutom jer navedeno odredište u naredbi označava podmrežu ili cijeli razred A, B ili C [3].

```
R1(config)#ip route 192.168.13.0 255.255.255.0 192.168.12.2
R1(config)#do show ip route
Codes: L - local, C - connected, S - static, 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

Gateway of last resort is not set

     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/0
L       192.168.1.254/32 is directly connected, GigabitEthernet0/0
S    192.168.3.0/24 [1/0] via 192.168.12.2
     192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.12.0/24 is directly connected, GigabitEthernet0/1
L       192.168.12.1/32 is directly connected, GigabitEthernet0/1
S    192.168.13.0/24 [1/0] via 192.168.12.2
```

Slika 5.1 - umetanje statičnog puta u tablicu usmjeravanja - izvor: autor - Cisco CLI, Cisco Packet Tracer

6. IPv6 adresiranje i usmjeravanje

Internet protokol verzije 6 nastao je zbog nedostataka adresa verzije 4. Adresa IP verzije 6 duljine je 128 bitova (**2345:1:2:3::1/64**), što znači da IPv6 adresa postoji $7.9228 * 10^{28}$ puta više nego verzije 4. Novija verzija Internet protokola koristi i drugačija polja zaglavlja, drugačijih veličina. Adresiranje se odvija po drugačijim pravilima, te se koriste mnogi drugi protokoli i funkcije namijenjene isključivo za verziju 6 Internet protokola. Kako bi podržali IPv6 sustav, na usmjerivačima je potrebno aktivirati IPv6 adresiranje i usmjeravanje kako bi usmjerivači mogli razumjeti IPv6 adrese i puteve. Zbog toga je migracija s IPv4 na IPv6 dosta zahtjevna, no moguće je primijeniti „dual-stack" funkcije na uređajima kako bi koristili obje verzije protokola i olakšali tranziciju na samo IPv6. „Dual-stack" mreža prikazana je na slici 6.2. Zbog različitih funkcionalnosti, IPv6 koristi i neke drugačije protokole od IPv4, kao što su modernija verzija 3 OSPF protokola stvorena isključivo za podršku u IPv6 sustavu. Internet Control Message Protocol (ICMP) je zamijenjen novijom verzijom ICMPv6. Također je i ARP (Address Resolution Protocol) zamijenjen modernijim protokol zvanim NDP (Neighbor Discovery Protocol). Zbog svoje velike duljine, IPv6 adrese vrlo je teško prikazati binarnim vrijednostima, te se zbog toga koriste heksadecimalni brojevi od kojih svaki iznosi 4 bita. U heksadecimalni brojevni sustav spadaju brojevi i slova: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Slova A, B, C, D, E, F služe kako bi zamijenila brojeve 10, 11, 12, 13, 14 i 15. IPv6 adrese sastoje se od 32 heksadecimalna broja koji iznose 128 bitova (**2001:0db8:76b4:0000:0000:0000:0362:7727**). Obzirom da su IPv6 adrese i vizualno velike, postoji skup dogovorenih pravila kojima se „krate" IPv6 adrese kako bi bile lakše čitljive i pamtljive. Jedno od tih pravila je pisati i podijeliti IPv6 adresu na osam dijelova (kvarteta) po četiri heksadecimalna broja. Zaglavlje IPv6 protokola nešto je veće, no mnogo je jednostavnije od IPv4 zaglavlja koje se enkapsulira na okvir. IPv6 zaglavlje prikazano je na slici 6.1.

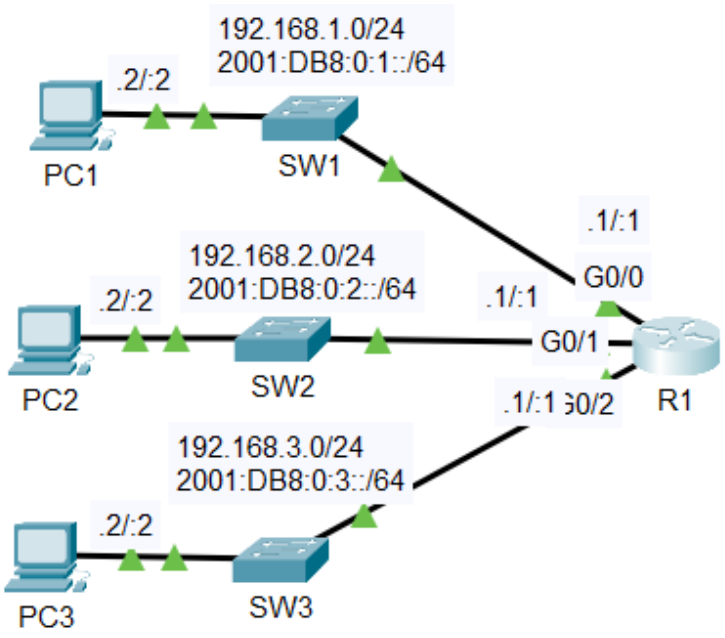


Slika 6.1 - izgled IPv6 zaglavlja - izvor: docs.oracle.com

Dva dogovorena standarda skraćivanja IPv6 adresa uključuju uklanjanje vodećih nula (0 s lijeve strane kvarteta) unutar svakog kvarteta od četiri heksadecimalna broja, što bi značilo da će IPv6 adresa (**FE80:0000:0000:0000:1234:5678:0000:1234**) nakon kraćenja izgledati: (**FE80:0:0:0:1234:5678:0:1234**). Drugo dogovoreno pravilo je da se jedan od skupova uzastopnih nula (0000:0000:0000) može zamijeniti znakom duplih dvotočaka (**::**). Primjenjujući drugo pravilo na prijašnju IPv6 adresa dobio bi se drugačiji izgled adrese: (**FE80::1234:5678:0:1234**). IPv6 sustav omogućava kao i IPv4 sustav, javno i privatno adresiranje (eng. Public and Private Unicast Addressing). Globalne, odnosno javne adrese dodjeljuju se od strane IANA organizacije (Internet Assigned Numbers Authority) kako bi se spriječio problem dupliciranih adresa. Svaka tvrtka može kupiti dogovoreni blok IPv6 adresa koje se mogu podijeliti na mnoštvo različitih mreža. IANA dodjeljuje najčešće blokove adresa s prefiksom /48 što omogućava veliki broj različitih podmreža i korisnika. IPv6 sustav također omogućuje korištenje privatnih adresa koje se nazivaju jedinstvene lokalne (Unique local) adrese, te djeluju poput IPv4 privatnih adresa. Jedinstvene lokalne adrese vrlo su slične globalnima, no razlikuju se po adresi jer

jedinstvene lokalne adrese prema dogovoru počinju s heksadecimalnim znamenkama FD (FD00::/8) [3].

Svaki usmjerivač koji usmjerava IPv4 ili IPv6 promet automatski stvara dva puta u svojim tablicama usmjeravanja, povezani put (eng. Connected Route) koji označava put do fizički povezanih mreža s uređajem, te lokalni put koji označava put do specifične IPv6 adrese konfigurirane na ulazu usmjerivača (duljina prefiksa je uvijek /128 jer lokalna adresa označava isključivo domaćina, eng. Host). Vrlo se često za ostale dinamičke puteve koriste dinamični protokoli usmjeravanja kako bi usmjerivači mogli popuniti svoje tablice. Uz dinamičke puteve, kao i u IPv4, postoje statički putevi koje manualno konfigurira mrežni inženjer. Kako bi se konfigurirale statičke rute, potrebno je koristiti naredbu „**ipv6 route**“ i naredbu „**ipv6 unicast-routing**“ kako bi se aktiviralo IPv6 usmjeravanja na usmjerivaču [4]. Naredba „ipv6 route“ koristi prefiks, njegovu duljinu i informacije o sljedećem skoku ili smjeru kao što je prikazano na slikama 6.3. i 6.4.



Slika 6.2 - jednostavna "dual-stack" mreža za IPv6 adresiranje - izvor: autor - Cisco Packet Tracer

```
R1>
R1>en
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip
R1(config)#ipv
R1(config)#ipv6 uni
R1(config)#ipv6 unicast-routing
R1(config)#int g0/0
R1(config-if)#ipv6 add
R1(config-if)#ipv6 address 2001:DB8:0:1::1/64
R1(config-if)#int g0/1
R1(config-if)#ipv6 add
R1(config-if)#ipv6 address 2001:DB8:0:2::1/64
R1(config-if)#int g0/2
R1(config-if)#ipv6 add
R1(config-if)#ipv6 address 2001:DB8:0:3::1/64
```

Slika 6.3 - aktivacija IPv6 usmjeravanja i adresiranje sučelja na usmjerivaču - izvor: autor - Cisco CLI, Cisco Packet Tracer

```
R1#show ipv6 int br
GigabitEthernet0/0         [up/up]
    FE80::201:97FF:FE9A:AC01
    2001:DB8:0:1::1
GigabitEthernet0/1         [up/up]
    FE80::201:97FF:FE9A:AC02
    2001:DB8:0:2::1
GigabitEthernet0/2         [up/up]
    FE80::201:97FF:FE9A:AC03
    2001:DB8:0:3::1
Vlan1                      [administratively down/down]
    unassigned
```

Slika 6.4 - ulazi usmjerivača i njihove konfigurirane IPv6 adrese - izvor: autor - Cisco CLI, Cisco Packet Tracer

7. IP liste kontrole pristupa

IP liste kontrole pristupa (eng. ACL – Access Control List) omogućuju mrežnim inženjerima kako bi mogli identificirati i upravljati različitim vrstama prometa. Konfiguracija lista kontrole pristupa sadržava vrijednosti koje pojedini usmjerivač može pregledati u procesu de-enkapuslacije iz pojedinih zaglavlja, kao što su IPv4, IPv6, TCP, UDP ili drugi. Kontrolne liste mogu identificirati pakete prema izvornim ili odredišnim IP adresama, ulazima aplikacija ili protokola. Kontrolne liste pristupa najčešće se primjenjuju kao „filtri“ paketa koji prolaze kroz pojedine usmjerivače. Pri konfiguraciji liste kontrole pristupa i njezinoj primjeni na usmjerivač, usmjerivaču može prema listi odrediti hoće li proslijediti ili odbaciti pakete. Liste se koriste i pri kvaliteti usluge (eng. QoS – Quality of Service) kako bi se mogli odvojiti paketi ovisno o značajki kvalitete usluge, pri čemu se paketima pridodaje manja ili veća važnost kako bi se osigurale bolje performanse ključnih aplikacija. Postoje tri različite vrste lista kontrole pristupa, a to su standardne (eng. Standard), proširene (eng. Extended) i imenovane (eng. Named) liste.

7.1. Standardne liste kontrole pristupa

Standardne liste kontrole pristupa najjednostavniji je oblik kontrolnih lista koji može identificirati pakete samo prema izvornoj IP adresi (eng. Source IP address) i koristi standardne brojeve za naziv liste (brojevi od 1 – 99, a kasnijim unaprjeđenjem dozvoljeno je korištenje i brojeva od 1300-1999). Standardne liste konfiguriraju se pomoću „access-list“ naredbe koja uključuje broj liste, dopuštenje ili odbijanje prometa i odabrane parametre prikazane na slici 7.1.

```
access-list access-list number permit/deny source IP [source-wildcard mask]
access-list 10 permit 192.168.1.0 0.0.0.255
```

Slika 7.1 - kod za stvaranje standardne liste kontrole pristupa - izvor: autor

U ovom slučaju naredba nazivom/brojem 10 dopušta sav promet iz mreže 192.168.1.0/24. Moguće je i konfigurirati više kontrolnih listi, a tad identifikacija funkcionira na principu „prvog pronalaska“ (eng. First Match), odnosno prvoj listi s kojom se promet podudara. Liste kontrole pristupa zahtijevaju korištenje „zamjenskih“ maski (eng. Wildcard Mask) umjesto običnih, te je iz tog razloga maska zapisana kao 0.0.0.255. Obična maska duljine prefiksa /24 piše se decimalno s točkom kao 255.255.255.0, gdje 255 označava rezervirane bitove za mrežni identifikator (eng. Network ID). Zamjenska maska jednostavno se dobije na način da se od obične maske 255.255.255.0 oduzme 255.255.255.255, te se dobije 0.0.0.255 zamjenska maska. Zamjenske maske koriste se iz povijesnih razloga, kao što su lakše kodiranje, no i dalje su ostale u upotrebi pa ih tako koriste i liste kontrole pristupa i OSPF protokol usmjeravanja. Standardne liste kontrole pristupa moraju biti postavljene čim je bliže odredištu moguće kako ne bi spriječile pogrešan promet koji prolazi kroz usmjerivač. Postavljaju se na sučelja u smjeru ulaza ili izlaza prometa (eng. Inbound/Outbound) pomoću naredbe „**ip access group (broj liste) out/in**“ - (**ip access group 10 out**).

7.2. Proširene liste kontrole pristupa

Proširene liste kontrole pristupa razlikuju se od standardnih po tome što mogu koristiti veći broj zaglavlja paketa za bolje usklađivanje, što znači da se promet može identificirati prema protokolu koji koristi, izvornoj i odredišnoj adresi i izvornom i odredišnom ulazu (eng. Port). Proširene liste kontrole pristupa koriste brojeve od 100-199, te je kasnije dozvoljeno i korištenje brojeva od 2000-2699. Slike 7.2 i 7.3 prikazuju potreban kod kako bi se stvorile proširene liste kontrole pristupa.

```
access-list access-list number permit/deny protocol source source-wildcard destination destination-wildcard


access-list 120 deny tcp 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255 eq 80
```

Slika 7.2 - kod za stvaranje proširene liste kontrole pristupa - izvor: autor

```
R1(config)#access-list 120 deny tcp 172.16.1.0 0.0.0.255 172.16.2.0 0.0.0.255 eq 80
R1(config)#do show acc
R1(config)#do show access-list
Extended IP access list 120
    10 deny tcp 172.16.1.0 0.0.0.255 172.16.2.0 0.0.0.255 eq www
```

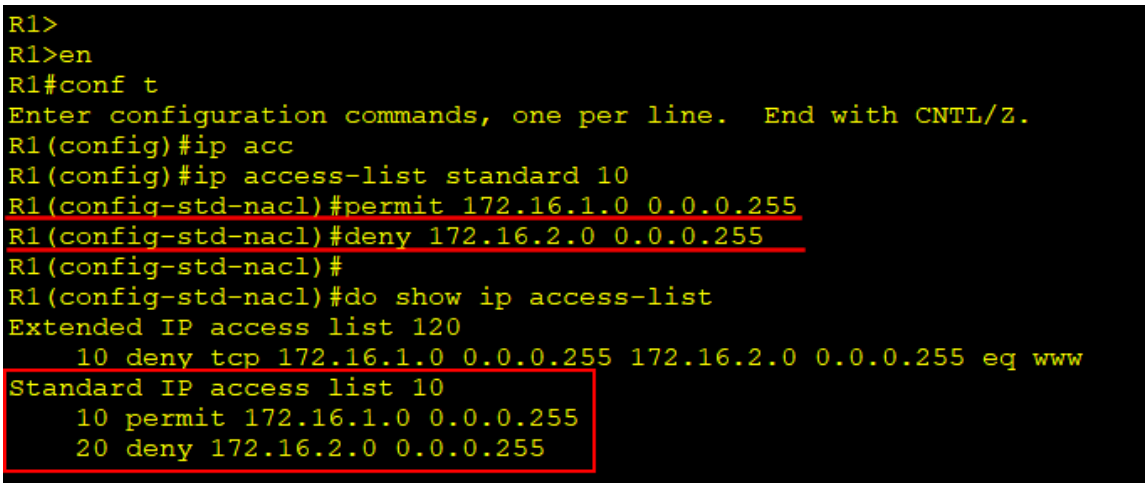
Slika 7.3 - konfiguracija proširene liste kontrole pristupa i odabranog protokola - izvor: autor - Cisclo CLI, Cisco Packet Tracer

U proširenim listama kontrole pristupa moguće je identificirati promet prema samom ulazu aplikacije koji može biti izvorni ili odredišni. U slučaju naredbe na slici, odbija se sav TCP promet iz mreže 172.16.1.0/24 do mreže 172.16.2.0/24 na ulazu 80, odnosno ulazu web prometa (http), što znači da korisnici mreže 172.16.1.0/24 ne mogu pristupiti web sadržaju putem http-a. Skraćenica eq dolazi od engleske riječi „equal“, te se koristi kako bi se identificirao točan „port“. Umjesto eq, moguće je koristiti i skraćenice „gt“ ili „lt“ koje znače „više od“ (eng. Greater than) i „manje od“ (eng. Lower than), te će označiti sve ulaze veće ili manje od navedenog u naredbi.

7.3. Imenovane liste kontrole pristupa

Imenovane liste kontrole pristupa imaju mnogo sličnosti s numeriranim IP listama kontrole pristupa, te se jednako koriste za filtraciju paketa ili kvalitetu usluge. Imenovane liste najmodernije su liste koje je moguće koristiti i kao standardne i kao proširene liste kontrole pristupa. Razlika je što imenovane liste koriste imena umjesto brojeva za identifikaciju liste, no moguće je i koristiti broj kao ime. Veliki napredak nad numeriranim listama je to što imenovane liste omogućuju mrežnim inženjerima posebnu konfiguraciju liste, na način da je moguće brisati ili uređivati pojedine linije lista, pa tako i dodati nove [3, 15]. Imenovane liste ne konfiguriraju se globalno, već koristeći ACL „podnaredbe“. Imenovane liste koriste „ip access-list“ naredbu kako bi korisnik ušao u ACL konfiguraciju prikazanu na slici 7.4.

```
ip access-list standard/extended name/number
```



Slika 7.4 - konfiguracija imenovane liste kontrole pristupa u obliku standardne - izvor: autor - Cisco CLI, Cisco Packet Tracer

8. Kvaliteta usluge (QoS – Quality of Service)

Moderne mreže prijenose veliku količinu podataka različitih aplikacija i usluga cijelo vrijeme, neki dijelovi prometa važniji su od ostalog, te je zbog toga potrebna konstantna kontrola kvalitete usluga. Kvaliteta usluge (eng. QoS – Quality of Service) odnos se na sve alate koje mrežni uređaji koriste kako bi primijenili različite oznake važnosti na pojedine pakete dok prolaze kroz uređaj na mreži. Mnogi WAN usmjerivači moraju postaviti pakete u red čekanja dok WAN ulaz ne postane dostupan. Usmjerivači mogu koristiti i različite algoritme koji služe kako bi usporedili pakete unutar reda čekanja i pritom odredili koje pakete treba proslijediti prije ostalih, što određuje nekim paketima bolju, a nekim lošiju uslugu. Alati kvalitete usluge omogućuju upravljanje četiri karakteristike prometa na mreži, propusnost (eng. Bandwidth), kašnjenje (eng. Delay), varijanca kašnjenja (eng. Jitter) i gubitak (eng. Loss) [3].

8.1. Propusnost (Bandwidth)

Propusnost se odnosi na sam kapacitet veze, odnosno koliko se bitova može poslati u mreži po sekundi, te se mjeri u bitovima po sekundi (eng. Bps – Bits Per Second). Alati kvalitete usluge omogućavaju osiguravanje dijela kapaciteta mreže za pojedini promet, što znači da je moguće odrediti 50% propusnosti veze kritičnim aplikacijama i ostale dijelove rasporediti po preostalim vrstama prometa. Algoritmi koji određuju red čekanja na taj način mogu propustiti važan promet kroz vezu prije manje bitnog. Glasovni pozivi preko IP-a obično zahtijevaju malo propusnosti, te je ona konstantna. Videozapis zahtijeva mnogo više propusnosti, te mu je često potrebna bolja kvaliteta usluge. Cisco preporuča svojim korisnicima kvalitetu usluge propusnosti za video od 384 Kbps (eng. Kilobits Per Second) pa do više od 20 Mbps (eng. Megabits Per Second), što ovisi o korištenoj kompresiji i rezoluciji videozapisa.

8.2. Kašnjenje (Delay)

U kvaliteti usluge, kašnjenje se može računati na dva načina, jednosmjerno kašnjenje ili kašnjenje u oba smjera. Jednosmjerno kašnjenje označava potrebno vrijeme između prijenosa jednog paketa od izvora do odredišta. Kašnjenje u oba smjera označava jednosmjerno kašnjenje i vrijeme kojem paketu treba za povratak od odredišta do izvora. Cisco preporuča za nesmetani prijenos podataka kašnjenje u jednom smjeru od 150 ms (eng. Miliseconds) ili manje za VoIP promet, dok se za video promet preporuča 200 do 400 milisekundi kašnjenja.

8.3. Varijanca kašnjenja (Jitter)

„Jitter“ se odnosi na varijaciju u jednosmjernom kašnjenju između uzastopno poslanih paketa iste aplikacije. Ako aplikacija šalje nekoliko stotina paketa određenom uređaju i kašnjenje prvog paketa iznosi 200 milisekundi (0,2 sekunde), pa tako i svakom ostalom paketu, tada u tome slučaju nema varijacije u kašnjenju. Ukoliko drugi paket kasni 220 milisekundi i treći 230 milisekundi, nastaje varijacija u kašnjenju koja bi iznosila 20 milisekundi između prvog i drugog paketa, te 10 milisekundi između drugog i trećeg paketa. Cisco preporuča držati varijancu kašnjenja na 30 milisekundi ili manje za VoIP promet, dok se za video preporuča varijanca od 30 do 50 milisekundi.

8.4. Gubitak (Loss)

Gubitak (eng. Loss) označava broj izgubljenih paketa, te se obično izražava u postotcima izgubljenih paketa. Gubitak se često javlja u trenucima prijenosa velikih količina prometa, kada se napune redovi čekanja, te zbog toga sustav počinje izbacivati pojedine pakete, što automatski uzrokuje gubitke pri prijenosu. Gubitak također može biti uzrokovan neispravnim vezama ili lošijim WAN uslugama. Cisco preporuča maksimalan gubitak paketa za VoIP promet od 1% ili manje, dok se za video promet preporuča 0,1% do 1% gubitaka.

8.5. Red čekanja (Queueing)

Najpoznatiji algoritam za raspoređivanje redova čekanja je „Round-Robin Scheduling“. Round-Robin redoslijedno prolazi kroz redove čekanja uzimajući podatke iz svakog reda. U svakom ciklusu procesa, algoritam uzima jedan paket ili određeni broj bajtova/bitova

iz svakog reda kako bi oslobodio prolaz za ostali promet. Algoritam Round-Robin Scheduling uključuje i funkcije „Weighted Round-Robin“ kako bi mogao uzeti različiti broj paketa iz različitih redova čekanja. Mnogi moderni usmjerivači koriste alat imenom CBWFQ (eng. Class-Based Weighted Fair Queueing) koji omogućava svakoj klasi minimalnu potrebnu propusnost tijekom zagušenja, kako bi pojedine aplikacije ili usluge mogle nastaviti raditi. U modernim mrežama, posebno za audio i video promet preko IP-a koristi se LLQ algoritam (eng. Low Latency Queueing). LLQ je korisniji od Round-Robin algoritma za audio i video promet, jer ponekad audio ili video promet trebaju prednost pred ostalim redovima čekanja, a Round-Robin mora proći kroz ostale redove, što dovodi do kašnjenja i velikih gubitaka pri prijenosu. LLQ zbog toga može odrediti pojedine redove čekanja kao „važne redove“, te će uzimati pakete iz tih redova prije ostalih, što omogućuje nesmetani prijenos audio i video prometa. Kako bi QoS mogao pripojiti oznake na promet, koristi polje DSCP (Differentiated Services Code Point) u IPv4 zaglavlju ili CoS (Class of Service) polje u Ethernet okviru [3, 16].

9. Kvaliteta Iskustva (QoE – Quality of Experience)

Kvaliteta iskustva (eng. QoE – Quality of Experience) odnosi se na zadovoljstvo korisnika i performansama pri korištenju usluge ili aplikacije. Na kvalitetu iskustva utječu čimbenici kao vrijeme odziva (eng. Latency), brzina prijenosa (eng. Bitrate), broj slika u sekundi (eng. Frame Rate) i gubitak paketa (eng. Packet Loss Rate). Kvaliteta iskustva promatra utjecaj na efikasnost mreže kod krajnjeg korisnika, te razne nezapažene nepravilnosti u mreži. Razne aplikacije ili usluge mogu podnesti različite vrijednosti gubitka paketa, pa tako u nekim mrežama gubitak paketa od 3% može biti zanemariv, dok u drugima gubitak paketa od samo 0,3% može uzrokovati probleme pri prijenosu podataka. Kvaliteta iskustva zahtijeva kvalitetniju analizu prijenosa prometa kroz mrežu, te prikupljanje i pohranjivanje ključnih podataka vezanim za rad i performanse mreže.

9.1. Vrijeme odziva (Latency)

Vrijeme odziva ili latencija vrlo je važna u iskustvu krajnjeg korisnika, odnosi se na vremenski interval kojem je potrebno da pošiljatelj primi odgovor na poslani paket. U mrežama je poželjna vrlo niska latencija koja ukazuje na brzinu mreže, odnosno reakciju. Nisko vrijeme odziva omogućuje nesmetanu komunikaciju koja je vrlo potrebna za aplikacije koje komuniciraju u stvarnom vremenu (video i glasovni pozivi, video igre). U slučajevima visokog vremena odziva promet u mrežama često kasni, te nastaje zastoj koji može dovesti do prestanka rada mreže. Problemi s visokom latencijom mogu uvelike narušiti kvalitetu iskustva kod krajnjeg korisnika.

9.2. Brzina prijenosa i broj slika u sekundi

Brzina prijenosa označava količinu podataka koji se prenose preko mreže u jednoj sekundi, te se najčešće mjeri u bitovima po sekundi (Bps) ili kilobitima po sekundi (kbps). Viši „bitrate“ obično uvjetuje boljoj kvaliteti audio i video prometa jer se prenosi veća količina podataka u sekundi. Niži „bitrate“ može uvjetovati zamućenjem ili „pikselizacijom“ slike, te nejasnim zvukom.

Broj slika u sekundi (eng. Frame Rate) označava broj „sličica“ (eng. Frame) koje se prikažu svake sekunde u nekom videu ili animaciji. Frame Rate mjeri se u slikama po sekundi (eng. FPS – Frame Per Second), te ima značajan utjecaj na kvalitetu iskustva u mrežama. Nizak Frame Rate često stvara probleme kod video igara, narušavajući kvalitetu iskustva korisnika. Visok Frame Rate osigurava glatko i nesmetano korištenje, te umanje pojave „trzanja“ slike (eng. Stuttering). Moderni standardi za kvalitetan video sadržaj zahtijevaju frame rate od 30 fps (eng. Frame per Second) ili više kako bi se osigurala maksimalna kvaliteta iskustva krajnjem korisniku [17].

10. SDN mreže i automatizacija

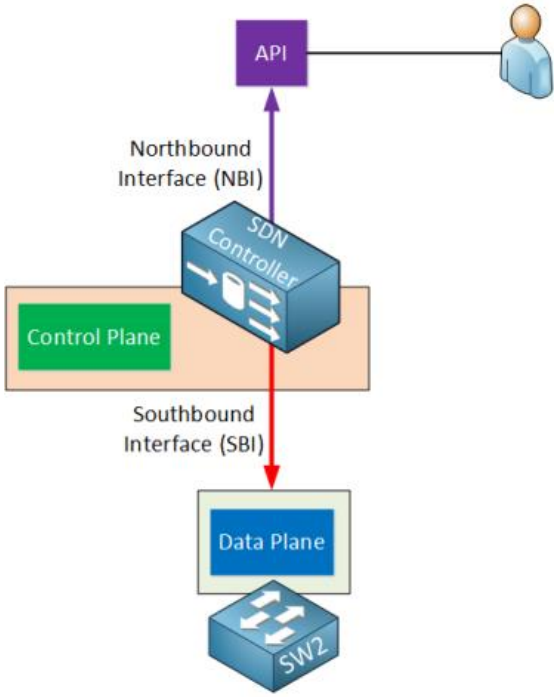
Podaci se preko mreže prenose u obliku okvira drugog sloja (Ethernet Frame). SDN mreže (eng. Software Defined Networks) putem softverskih alata analiziraju komponente i performanse mreže, te primjenjuju načine nadograđivanja kako bi unaprijedili mreže za današnje potrebe. SDN mreže koriste automatizaciju kako bi dinamički primijenili konfiguracije na pojedine uređaje. Uređaji između sebe šalju poruke i pohranjuju razloge promjena. Ključni dio SDN mreža je centralizirani upravitelj (eng. Centralized Controller) koji stvara arhitekturu za lakšu programibilnost i kontrolu SDN mreža. Arhitektura SDN mreža uključuje tri različite „ravnine“ (eng. Planes) koje se koriste za opisivanje načina rada programibilnosti: podatkovna ravnina (eng. Data Plane), kontrolna ravnina (eng. Control Plane) i upravljačka ravnina (eng. Management Plane).

Podatkovna ravnina (eng. Data Plane) označava sve funkcije koje mrežni uređaj obavlja kako bi proslijedio podatke. U podatkovnu ravninu spadaju sve funkcije vezane uz primanje i slanje podataka, te njihovu obradu. Kako bi pravilno funkcionirala podatkovna ravnina, usmjerivačima su potrebni IP putevi u tablicama usmjeravanja kako bi ravnina mogla proslijediti promet. Prekidačima i sloj 2 protokolima potrebne su MAC adrese kako bi se promet mogao usmjeriti kroz točan ulaz.

Kontrolna ravnina označava svaku funkciju i protokole koji upravljaju podatkovnom ravninom. Takve funkcije uključuju proces stvaranja tablica usmjeravanja kojom se koristi podatkovna ravnina ili ARP (eng. Address Resolution Protocol) tablica. Upravljačka ravnina upravlja podacima u tablicama, može ih ukloniti ili promijeniti.

Upravljačka ravnina obavlja funkcije koje ne utječu direktno na rad podatkovne ravnine, već uključuju protokole koji omogućuju mrežnim inženjerima upravljanje mrežnim uređajima. Najpoznatiji takvi protokoli su SSH (Secure Shell) i Telnet (Teletype Network) koji omogućuju pristup uređajima s udaljenih mjesta bez direktnog, fizičkog povezivanja. SSH podatke koje prosljeđuje „enkriptira“ što ga čini najkorištenijim protokolom danas.

Centralizirani upravitelj (eng. Centralized Controller / SDN Controller) omogućuje centralnu kontrolu nad mrežnim uređajima. Upravitelj može zamijeniti kontrolnu ravninu svakog pojedinog uređaja, te preuzeti sve funkcije kontrolne ravnine, no isto je moguće koristiti upravitelj za analizu uređaja i ravnina bez promjene njihovog načina djelovanja.



Slika 10.1 - arhitektura SDN mreže s centraliziranim upraviteljem - izvor: netfv.wordpress.com

Kako bi mreža funkcionirala, upravitelj mora moći komunicirati s mrežnim uređajima. Na vizualnim modelima mreže uređaji se obično nalaze ispod upravitelja, te je zbog toga sučelje (eng. Interface) između upravitelja i mrežnih uređaja nazvano „Južno sučelje“ (eng. SBI - Southbound Interface).

„Sjeverno sučelje“ (eng. NBI – Northbound Interface) omogućuje podacima i funkcijama upravitelja komunikaciju s drugim aplikacijama ili programima, osiguravajući kvalitetnu programibilnost kao što je prikazano na slici 10.1. Podaci se prenose pomoću API (Application Technology Interface) tehnologije kako bi se omogućio tok podataka od korisnika do uređaja [3].

Jedan od oblika SDN mreža stvorila je „Open Networking Foundation“ (ONF) organizacija, te se naziva „Open SDN“. Takav oblik obuhvaća protokole južnih i sjevernih sučelja i alate potrebne za implementaciju SDN mreža, te centralizira većinu funkcija kontrolne ravnine. Najpoznatiji Open SDN protokol južnog sučelja je „OpenFlow“.

OpenFlow omogućuje programibilnost i centralizirano upravljanje u SDN mrežama, te odvaja kontrolnu ravninu od podatkovne. Upravitelj putem OpenFlow protokola šalje posebne poruke s kojima može djelovati na ponašanje mrežnih uređaja, te pritom omogućava fleksibilniju prilagodbu mrežnog prometa prema potrebama korisnika. OpenFlow poruke prosljeđuju naredbe mrežnim uređajima kako bi ih mogli primijeniti na podatkovne pakete, poruke uključuju naredbe različitih filtracija i promjena svojstava podatkovnih paketa. OpenFlow model također omogućuje korištenje različitih API tehnologija s sjevernog sučelja koje su podržane na platformi upravitelja kako bi se odredili unosi puteva u tablice usmjeravanja. Prekidači moraju podržavati OpenFlow.

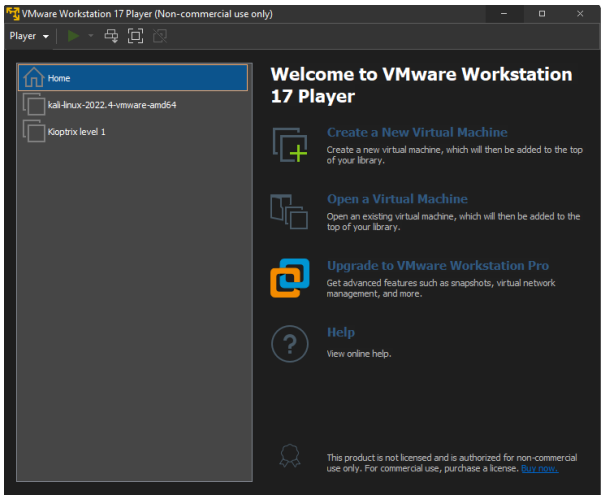
Za automatizaciju konfiguracija uređaja postoje mnogi različiti alati, no najkorišteniji su „Ansible“, „Puppet“ i „Chef“. Ansible koristi „agentless“ arhitekturu za upravljanje mrežnim uređajima, što znači da ne ovisi o kodu koji se izvodi na mrežnom uređaju, već uobičajenim protokolima poput SSH ili NETCONF pomoću kojih izvodi promjenu ili izvlači informacije iz uređaja [18, 19].

11. Prijenos multimedijskog sadržaja koristeći OMNeT++ alat

11.1. Virtualizacija

11.1.1. VMWARE Workstation Player

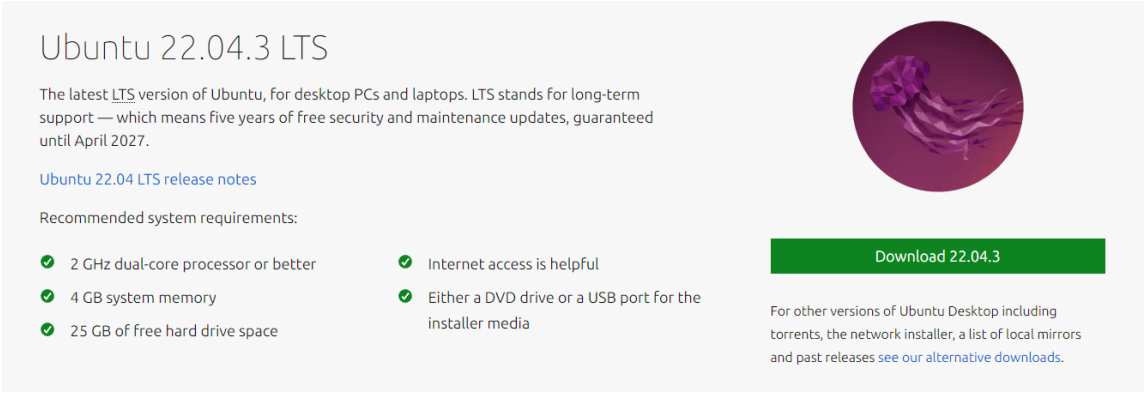
Jedna od najpopularnijih tvrtka u industriji virtualizacije i infrastrukture oblaka je VMware. VMware omogućuje mnoštvo različitih rješenja za virtualizaciju, upravljanje ili automatizaciju sustava. Tvrtka također nudi besplatni softverski alat, odnosno virtualni „Hypervisor“ (softver koji omogućuje stvaranje više gostujućih, virtualnih operativnih sustava na jednom operativnom sustavu „domaćinu“) za stvaranje i pokretanje virtualnih mašina na osobnim računalima čije je sučelje prikazano na slici 11.1. Softver omogućuje korisnicima izolaciju operativnih sustava i podataka koji se koriste za testiranje ili razvoj. VMware Workstation Player posjeduje i „Snapshot“ funkcionalnosti koje omogućuju pohranu trenutnog stanja virtualne mašine kako bi se korisnik u slučaju problema mogao vratiti na prethodno, ispravno stanje.



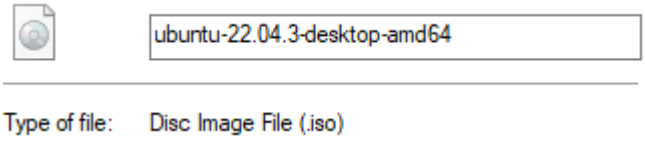
Slika 11.1 - izgled sučelja VMware Workstation Player alata - izvor: autor

11.1.2. Instalacija virtualnog operativnog sustava Ubuntu(Linux) 22.04

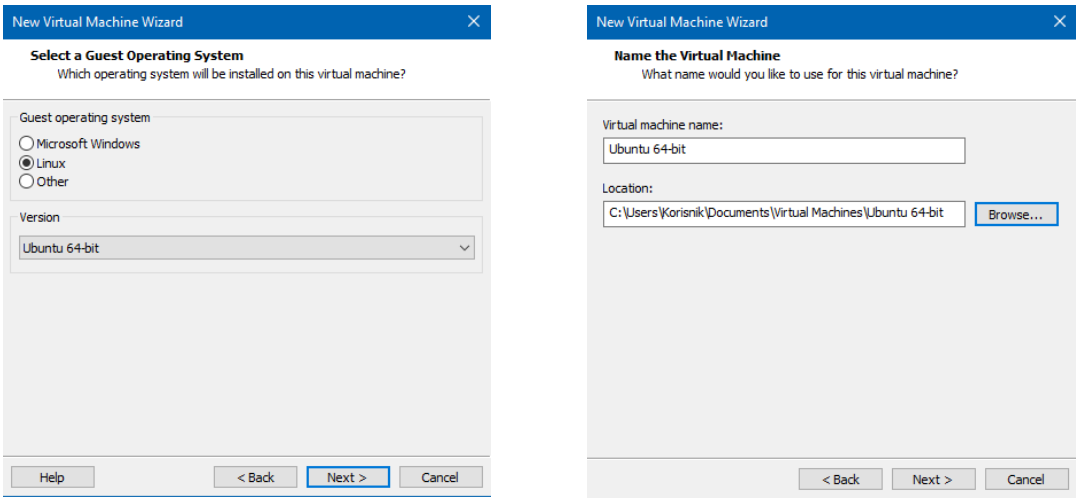
Kako bi se instalirala Ubuntu inačica operativnog sustava Linux kao virtualna mašina na VMware alatu osobnog računala potrebno je preuzeti relevantnu (.iso) datoteku sa službenih stranica Ubuntu operativnog sustava prikazanu na slici 11.2. ISO format datoteke služi kako bi točne kopije podataka optičkih diskova arhivirao u jednu datoteku, te je pogodan za prijenos „slika“ operativnih sustava.



Slika 11.2 - mjesto preuzimanja Ubutnu verzije 22.04 u obliku .iso datoteke - izvor: ubuntu.com

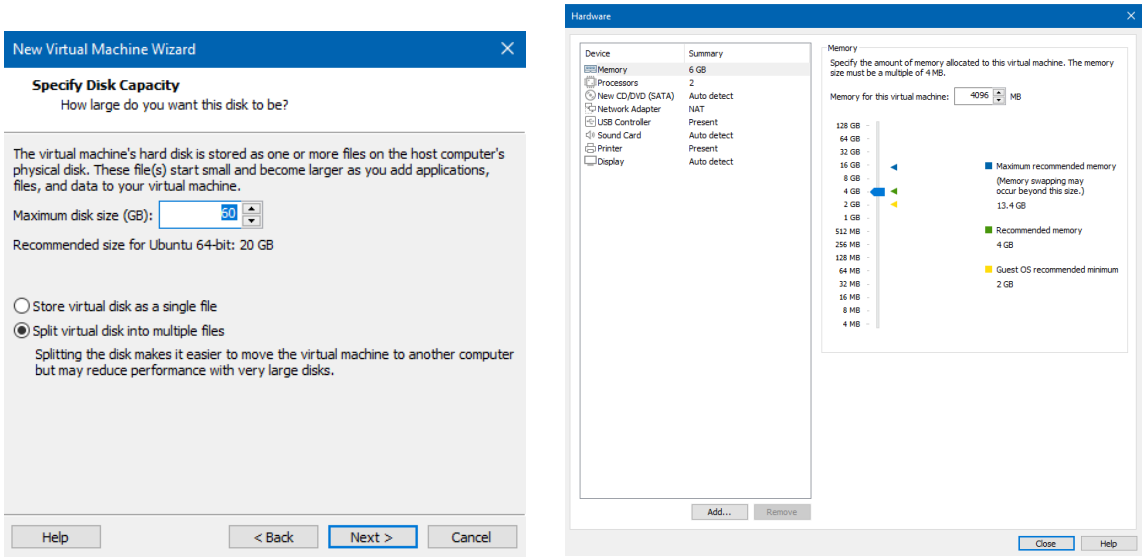


ISO datoteka potrebna je kako bi mogli stvoriti virtualnu mašinu inačice Ubuntu 22.04, te instalirati odabrani operativni sustav. Pritiskom na poveznicu „Create new Virtual Machine“ u VMware sučelju, pokrenuti će se proces stvaranja virtualne mašine, gdje je potrebno navesti relevantne informacije i korištenje resursa virtualnog operativnog sustava. Slike 11.3 – 11.7 prikazuju proces instalacije virtualnog operativnog sustava. Instalacija od korisnika prvo traži informacije o željenom operativnom sustavu i mjestu na kojem će se nalaziti. Korisnik unosi željeno ime virtualne mašine.



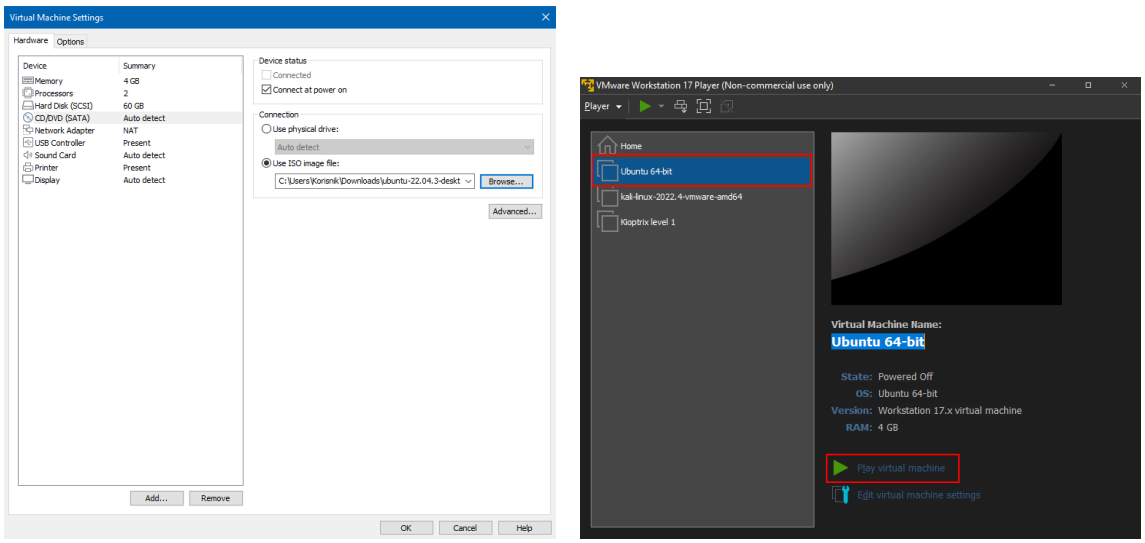
Slika 11.3 - instalacija Ubuntu operativnog sustava - izvor: autor - VMware Workstation Player

Drugi korak instalacije zahtjeva navođenje veličine virtualnog diska, te VMware Workstation Player omogućuje odabir količine resursa hardvera kojima će se koristiti virtualni operativni sustav (ovisno o mogućnostima osobnog računala).



Slika 11.4 - dodjela resursa virtualnom sustavu - izvor: autor - VMware Workstation Player

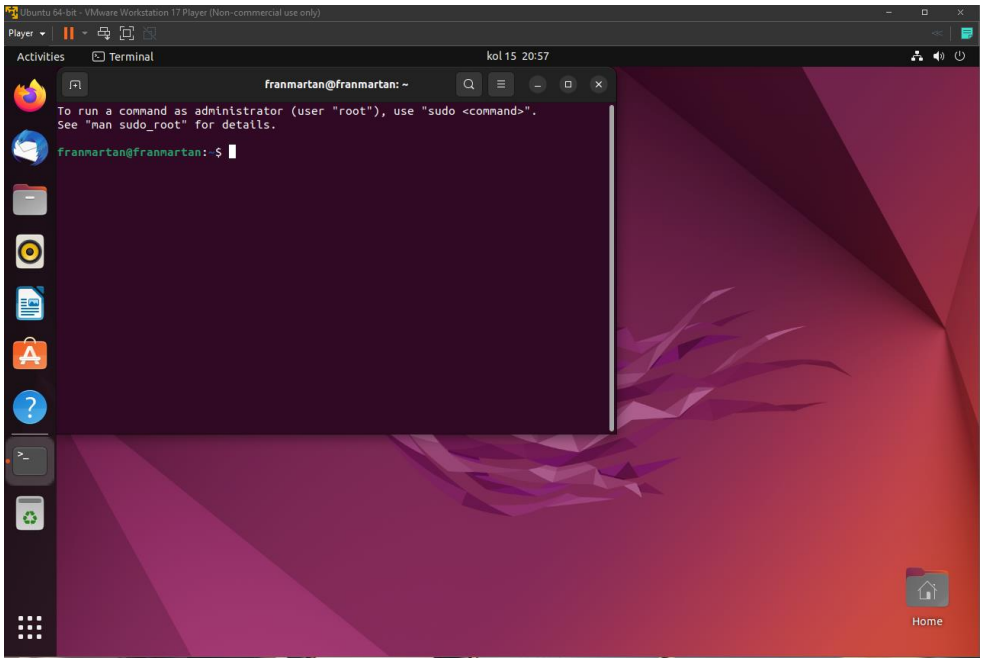
Pri alociranju resursa hardvera potrebno je pod opcijom „New CD/DVD (SATA)“ odabrati opciju povezivanja „Use ISO image file“ i priložiti prethodno preuzetu ISO datoteku, te je tada moguće pokrenuti virtualnu mašinu i instalaciju operativnog sustava.



Slika 11.5 - ISO datoteka i pokretanje virtualnog sustava - izvor: autor - VMware Workstation Player

Pokretanjem virtualne mašine, otvara se proces za instalaciju Ubuntu operativnog sustava. Potrebno je odabrati opciju „Install Ubuntu“ i proći kroz instalaciju, nakon čega će korisnik moći koristiti operativni sustav.

Slika 11.6 - instalacija Ubuntu operativnog sustava - izvor: autor - VMware Workstation Player



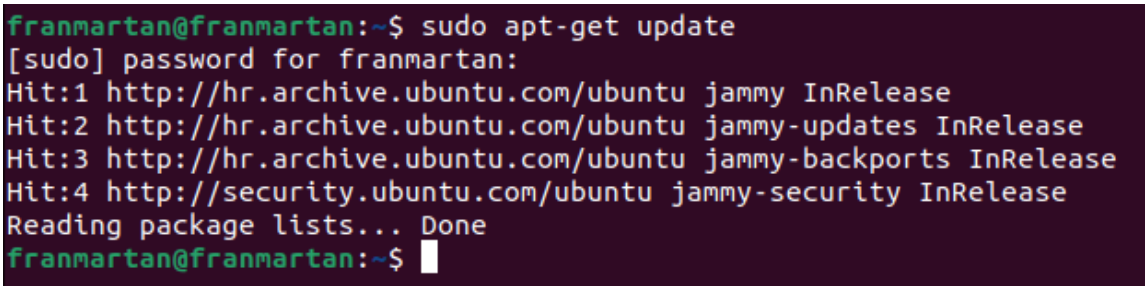
Slika 11.7 - izgled operativnog sustava Ubuntu - izvor: autor - VMware Workstation Player

11.2. OMNeT++ simulacijski alat

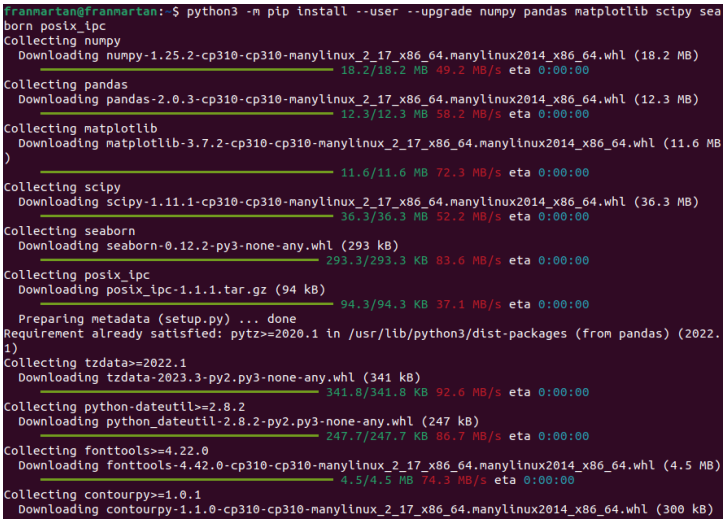
OMNeT++ je modularni simulacijski alat zasnovan na C++ simulacijskoj „knjižnici“ stvoren za stvaranje žičanih ili bežičnih komunikacijskih mrežnih simulacija. OMNeT++ omogućuje specifično modeliranje simulacija za određena područja, kao što su bežične „ad hoc“ mreže, mreže senzora ili različitih protokola. OMNeT++ stekao je veliku popularnost u znanstvenim zajednicama i mnogim industrijskim okruženjima. Kako bi stvorio razne simulacije, OMNeT++ koristi različite module kodirane u C++ programskom jeziku, te ih sastavlja u veće komponente koristeći specifičan jezik NED – Network Description. Moduli omogućuju inženjerima precizno modeliranje pojedinih dijelova sustava, te ih je moguće koristiti na više mjesta. OMNeT++ koristi i vlastito grafičko sučelje koje omogućava korisnicima vizualni prikaz rezultata mrežnih simulacija, što omogućuje efikasniju analizu podataka. Alat omogućuje korisnicima definiranje različitih parametara simulacija.

11.2.1. Instalacija OMNeT++ simulacijskog alata na Linux virtualnom operativnom sustavu

Kako bi instalacija OMNeT++ simulacijskog alata bila uspješna potrebno je prethodno instalirati i ažurirati ključne pakete koji sadrže potreban C++ kompajler (eng. Compiler) i ključne programe i knjižnice. Prije same instalacije potrebno je ažurirati već postojeću bazu paketa Ubuntu sustava pomoću naredbe „*sudo apt-get update*“ kao što je prikazano na slikama 11.8 i 11.9.

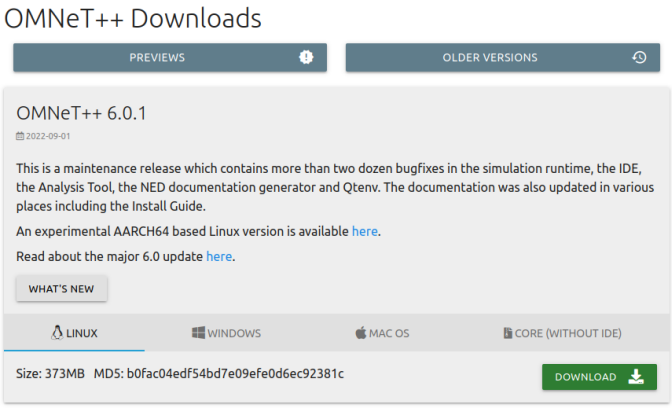


Slika 11.8 - ažuriranje baze paketa Ubuntu sustava - izvor: autor - Ubuntu Terminal



Slika 11.9 - instalacija potrebnih paketa i Python3 knjižnica - izvor: autor – Ubuntu Terminal

Nakon preuzimanja, instalacije i ažuriranja neophodnih paketa za instalaciju OMNeT++ alata, potrebno je preuzeti ispravnu komprimiranu datoteku OMNeT++ alata sa službenih stranica, te dekomprimirati i zalijepiti na željeno mjesto u sustavu kao što je prikazano na slici 11.10. Među datotekama nalazi se i „*setenv*“ skripta koja omogućuje automatsku konfiguraciju „varijable okruženja“ (eng. Envirnoment Variables). Nakon prethodnih koraka, naredbe „*./configure*“ i „*make“* pokrenut će instalaciju prikazanu na slici 11.11.



Slika 11.10 - mjesto preuzimanja alata - izvor: omnetpp.org

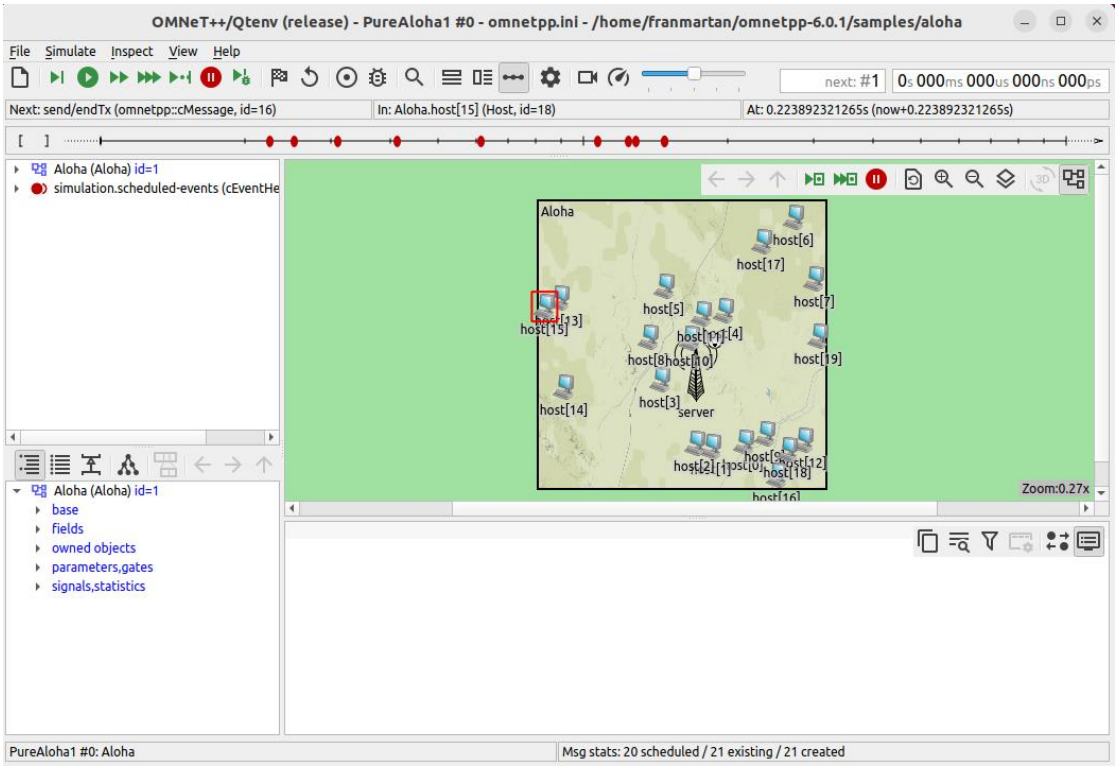
```
franmartan@franmartan:~$ cd omnetpp-6.0.1/
franmartan@franmartan:~/omnetpp-6.0.1$ source setenv
Environment for 'omnetpp-6.0.1' in directory '/home/franmartan/omnetpp-6.0.1' is ready.

Type "./configure" and "make" to build the simulation libraries.
When done, type "omnetpp" to start the IDE.
franmartan@franmartan:~/omnetpp-6.0.1$ ./configure
configure: Environment variables (PATH and PYTHONPATH) are correctly set.
configure: Reading configure.user for your custom settings.
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for clang... clang
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether clang accepts -g... yes
checking for clang option to accept ISO C89... none needed
checking for clang++... clang++
checking whether we are using the GNU C++ compiler... yes
checking whether clang++ accepts -g... yes
checking for clang++... clang++
checking for c++14 support... yes
checking for ranlib... ranlib
checking whether LLD linker is available... no
checking whether clang++ supports -fno-omit-frame-pointer... yes
checking whether clang++ supports -glldb... yes
checking whether clang++ supports -fstandalone-debug... yes
checking whether clang++ supports -Wl,--no-as-needed... yes
checking whether clang++ supports -Wl,--as-needed... yes
checking for swapcontext... yes
```

Slika 11.11 - instalacija OMNeT++ alata - izvor: autor - Ubuntu Terminal

```
franmartan@franmartan:~/omnetpp-6.0.1$ make

Building release and debug mode executables. Type 'make help' for further options.

***** Configuration: MODE=release, TOOLCHAIN_NAME=clang, SHARED_LIBS=yes, LIB_SUFFIX=.so ****
===== Checking environment =====
===== Compiling utils ====
===== Compiling common ====
YACC: expression.y
LEX: expression.lex
YACC: matchexpression.y
lcgrandom.cc
filelock.cc
filereader.cc
linetokenizer.cc
stringpool.cc
pooledstring.cc
stringtokenizer.cc
fnamelisttokenizer.cc
expression.cc
lex.expressionyy.cc
expression.tab.cc
matchexpression.cc
matchexpressionlexer.cc
matchexpression.tab.cc
patternmatcher.cc
unitconversion.cc
displaystring.cc
```

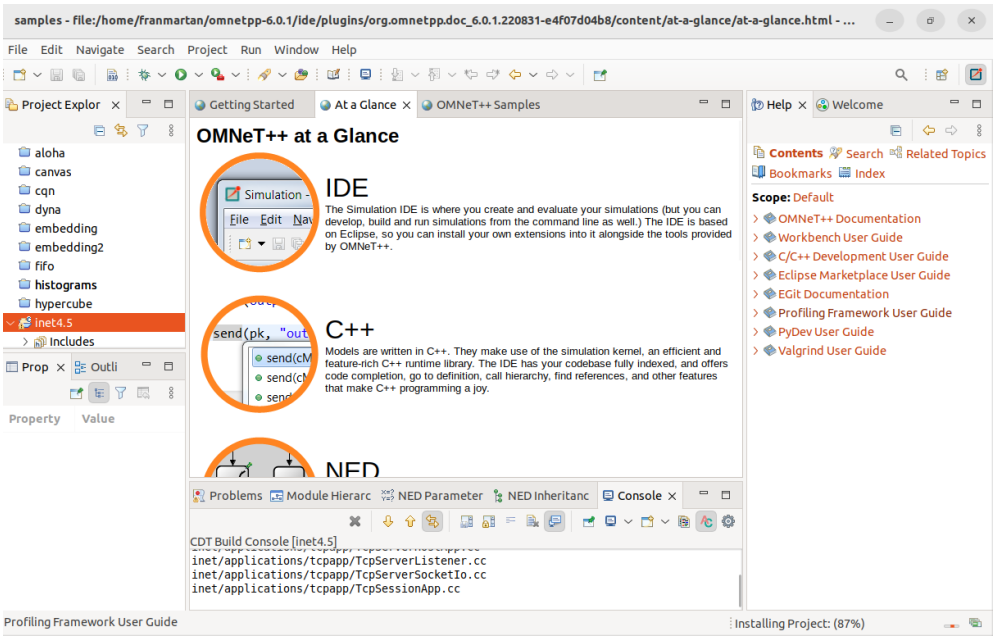
Slika 11.12 - naredba make i instalacija OMNeT++ alata - izvor: autor - Ubuntu Terminal

Nakon završetka instalacije, OMNeT++ omogućuje provjeru instalacije pomoću primjera „*./aloha*“, te će pokrenuti *Qtenv* okolinu s početnim primjerom prikazanim na slici 11.13.



Slika 11.13 - qtenv/OMNeT++ primjer - izvor:autor

OMNeT++ alata posjeduje i svoje integrirano razvojno okruženje (eng. IDE – Integrated Development Environment) koje se pokreće pomoću naredbe „*omnetpp*“ i prikazano je na slikama 11.4.



Slika 11.14 - izgled OMNeT++ simulacijskog alata - izvor: autor

11.2.2. INET i OMNeT++ simulacija „Tail Drop“ gubitaka paketa koristeći UDP Video Streaming uslugu

OMNeT++ Arhitektura

Simulacija OMNeT++ sastoji se od više modula koji komuniciraju putem poruka. Jednostavni moduli napisani su u C++ programskom jeziku, te se mogu povezivati u složenije i veće module. Broj razina hijerarhije je neograničen, te je cijela simulacija jedan veliki složeni modul. Moduli poruke šalju putem veza kroz ulaze na uređajima, koji se u OMNeT++ alatu nazivaju „vrata“ (eng. Gates). Poruke sadržavaju različite podatke, poput vremenskog žiga i potrebnih informacija o korištenim tehnologijama ili aplikacijama. Vrata predstavljaju ulazna i izlazna sučelja modula koja mogu biti povezana različitim vezama, promet se šalje kroz izlazno sučelje, a prima na ulazno sučelje. Na vezama je moguće podesiti različite parametre, poput kašnjenja, brzine ili količine prijenos, te stopu bitova s greškom. OMNeT++ omogućuje stvaranje posebnih objekata imenom kanali, koji omogućuju korištenje jednakih parametara na više veza kako bi se smanjila količina napisanog koda. Kao i veze, parametre posjeduju i moduli koji omogućuju konfiguraciju prijenosa podataka, te prihvaćaju sve tipove podataka poput „string-ova“, „boolean-a“ ili numeričkih vrijednosti. Parametri modula, također su predstavljeni kao objekti, te mogu sadržavati konstante vrijednosti ili izvore nasumičnih brojeva. Simulacija može pri pokretanju tražiti od korisnika da navede pojedine parametre, ovisno o potrebi. Za detaljnu analizu rezultata, OMNeT++ posjeduje IDE (eng. Integrated Development Environment) sučelje koje grafički prikazuje samu mrežu. Napisan kod dodanih modula, automatski prikazuje grafički izgled mreže i njezinih komponenata. OMNeT++ IDE omogućuje korisnicima praćenje simulacije, te pohranjuje sve događaje (eng. Event) u mreži za kasniju analizu. Korisniku je omogućeno stvaranje analitičkih datoteka pomoću kojih može iščitati korisne podatke o mreži, kao što su gubitak paketa, brzina prijenosa, količina prometa ili broj sličica po sekundi. Alat omogućuje filtraciju podataka prema modulima, vratima, vezama ili pojedinim događajima u simulaciji. Kako bi se mogla opisati arhitektura modela i modula OMNeT++ koristi vlastiti NED (Network Description) jezik.

NED jezik

NED omogućuje deklaraciju jednostavnih aktivnih modula, te povezivanje istih u veće i složenije module. Svaki se složeni modul može razdijeliti na manje jednostavnije dijelove zbog bolje preglednosti, te kasnije povezati u složeni. Stvoreni moduli mogu se koristiti u više različitih simulacija kao parametri ili sami objekti, te se njihova deklaracija koristi kao opis ulaza i parametara ulaznih i izlaznih sučelja. NED jezik omogućuje unos metapodataka koji dodatno opisuju pojedine module kao što je prikazano slikom 11.15. NED uvelike olakšava stvaranje kompleksnih modela mrežnih simulacija, te zbog toga OMNeT++ alat dobiva veliku popularnost među mrežnim inženjerima.

```
network jednostavnaSimulacija
{
    @display("bgb=839,592");
    submodules:
        myHost: MyHost {
            @display("p=167,379");
        }
        server: MyHost {
            @display("p=542,389");
        }
        ipv4FlatNetworkConfigurator: Ipv4FlatNetworkConfigurator {
            @display("p=358,275");
        }
        radioMedium: RadioMedium {
            @display("p=612.688,150.192");
        }
    connections:
        myHost.ethg++ <--> Eth100M <--> server.ethg++;
}
```

Slika 11.15 - primjer NED datoteke - izvor: autor - OMNeT++

INET programski okvir alata OMNeT++

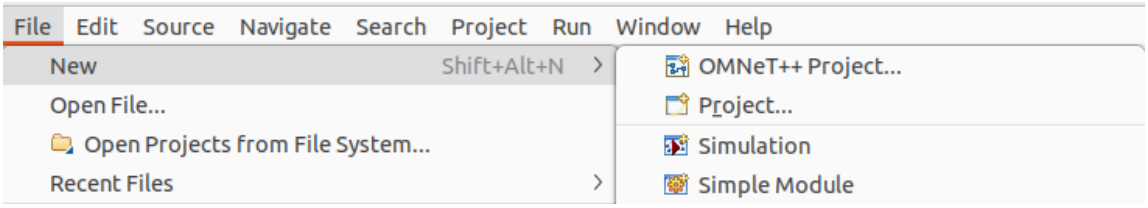
Programski okvir INET alata OMNeT++ otvorena je knjižnica modela i modula za simulacije. INET omogućuje simulacije protokola ili aplikacija pomoću unaprijed stvorenih modula koji omogućuju jednostavnije stvaranje simulacije putem grafičkog sučelja. Korišteni moduli automatski stvaraju potreban kod u NED datotekama. INET okvir podržava veliki broj različitih vrsta mreža u simulaciji, kao što su žičane ili bežične mreže, ad-hoc mreže, mobilne i senzorske mreže. Mreže podržavaju veliki broj protokola kao što su TCP, UDP, IPv4 ili IPv6, RTP, OSPF, BGP ili drugi. Korištene veze podržavaju Ethernet, PPP ili IEEE 802.11 tehnologije. Kako bi se stvoreni moduli mogli povezati, te im se dodati željeni parametri ili funkcije, INET koristi INI (.ini) datoteke. Parametri se mogu dodati i putem NED datoteka, no zbog preglednosti parametri se stvaraju u INI datotekama (Slika 11.16), te se putem „*omnetpp.ini*“ datoteke pokreće simulacija.

```
[General]
network = jednostavnaSimulacija

*.myHost.numApps = 1
*.myHost.app[*].typename = "UdpVideoStreamClient"
*.myHost.app[*].localPort = 1001
*.myHost.app[*].serverAddress = "server"
*.myHost.app[*].serverPort = 1001
*.myHost.app[*].startTime = 0s
*.myHost.app[*].endToEndDelay.statistic-recording = true
*.myHost.*.displayAddresses = true

*.server.numApps = 1
*.server.app[*].typename = "UdpVideoStreamServer"
*.server.app[*].localPort = 1001
*.server.app[*].videoSize = 10MiB
*.server.app[*].sendInterval = normal(10ms, 1ms)
*.server.app[*].packetSent.statistic-recording = true
```

Slika 11.16 - primjer .ini datoteke i parametara - izvor: autor - OMNeT++

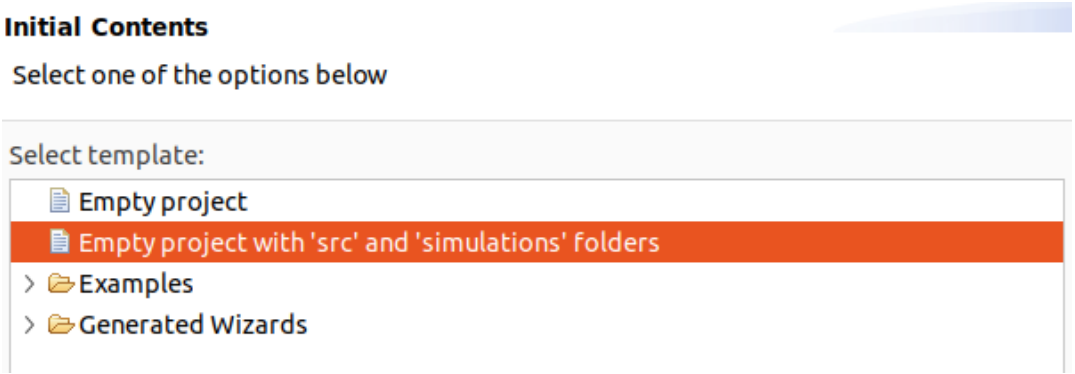
Stvaranje „Video Streaming“ simulacije putem UDP protokola koristeći INET okvir

Nakon pokretanja OMNeT++ alata, potrebno je u glavnom izborniku, pod opcijom „File“ i „New“ odabrati OMNeT++ projekt (eng. OMNeT++ Project).



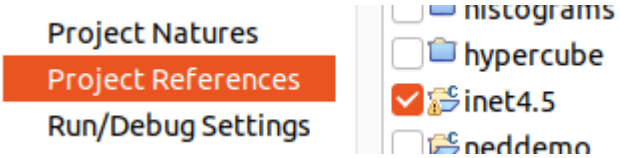
Slika 11.17 - stvaranje novog OMNeT++ projekta

Stvaranje novog projekta od korisnika zahtjeva ime projekta, te direktorij gdje korisnik želi pohraniti svoj projekt. Odabirom imena i mjesta pohrane potrebno je odabrati jednu od dvije opcije, „Empty Project“ ili „Empty project with 'src' and 'simulations' folders“. Druga opcija će odmah pri stvaranju projekta, stvoriti potrebnu NED i INI datoteku.



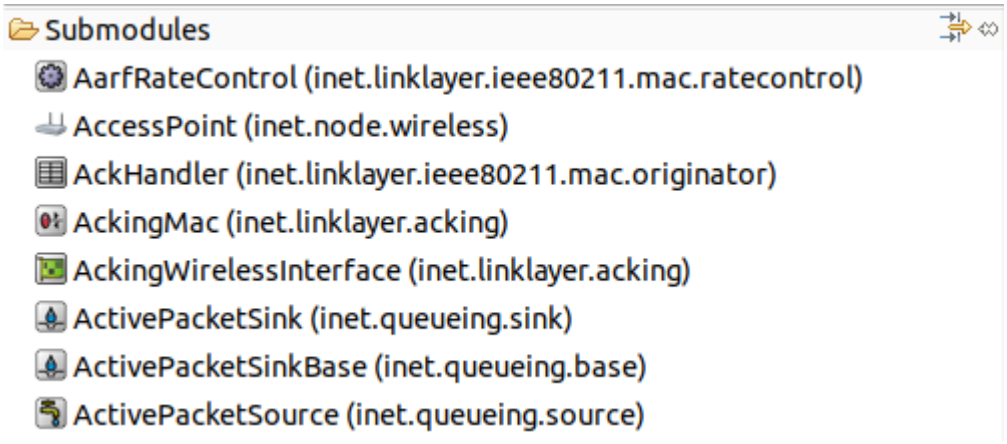
Slika 11.18 - odabir vrste projekta - izvor: autor - OMNeT++

Kako bi se omogućilo korištenje INET programskog okvira, potrebno je desnim klikom na stvoreni projekt, odabrati „Properties“, te pod kategorijom „Project References“ odabrati „inet 4.5“ opciju i pritisnuti „Apply and Close“.



Slika 11.19 - aktivacija INET okvira - izvor: autor, OMNeT++

Uključenjem INET okvira u postojeći projekt, korisniku se omogućuje izbornik velikog broja unaprijed stvorenih modula za različite potrebe simulacija iz INET-ove knjižnice.



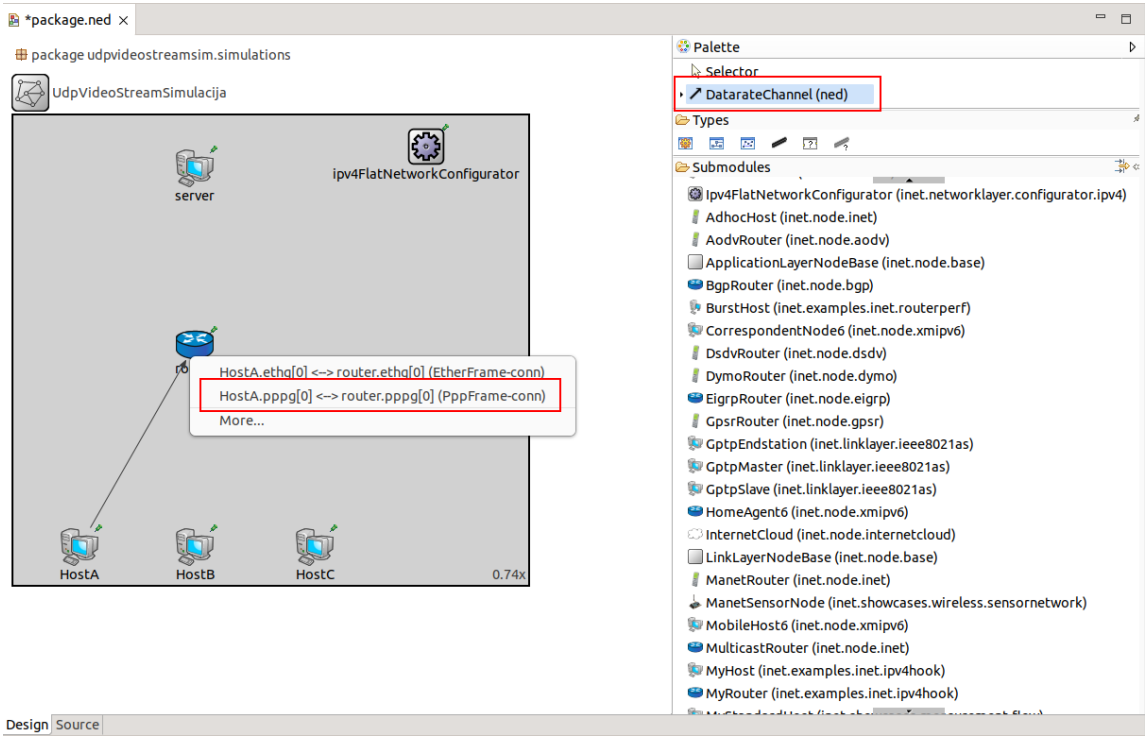
Slika 11.20 - izbornik modula INET okvira - izvor: autor - OMNeT++

Za početak simulacije potrebno je stvoriti područje mreže, odabirom odgovarajuće opcije iznad izbornika za module „Network – Create a Network Type“ ili ubacivanjem naredbe network „ime mreže“ u kodu NED datoteke.

Slika 11.21 - stvaranje područja mreže u NED datoteci - izvor: autor - OMNeT++

Na stvoreno područje mreže potrebno je, iz INET izbornika modula, odabrati željene module koji će se koristiti u simulaciji. Za potrebe simulacije koristit će se četiri različita modula imenom „StandardHost“ od kojih tri predstavljaju klijente na mreži, a preostali modul predstavlja server koji će pružati uslugu Video Streaming-a, te modul imenom „Router“ koji predstavlja usmjerivač na mreži kako bi usmjerio pakete pripadajućim klijentima. Svi će se moduli pomoću vrata povezati „Datarate Channel“ vezom koja omogućuje konfiguraciju važnih parametara, poput kašnjenja i stope „krivih“ paketa ili bitova. Desnim klikom na modul, moguće je promijeniti naziv modula.

Veze između pojedinih modula primjenjuju se pomoću opcije „Connection“ iznad izbornika za module. Grafičko sučelje omogućuje jednostavno povezivanje uređaja koristeći miš, no moguće je i napisati kod.



Slika 11.22 - stvaranje veze između modula, pomoću grafičkog sučelja - izvor: autor - OMNeT++

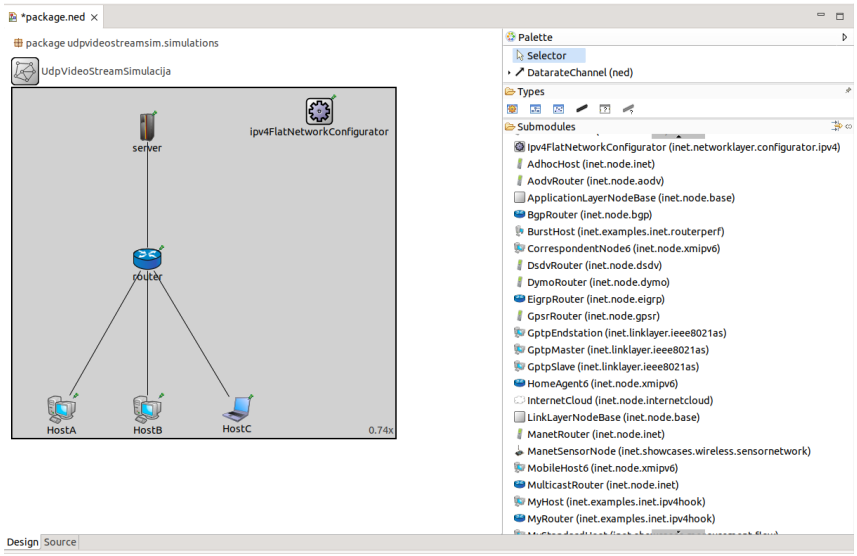
Veze „DatarateChannel“ povezuju se putem „ppp“ (Point-to-Point) protokola zbog dodatnih mogućnosti parametara kao što je prikazano na slici 11.22. Modul „ipv4FlatNetworkConfigurator“ koji se nalazi na području mreže, koristi se kako bi dinamički dodijelio IPv4 adrese i maske uređajima na mreži kako bi mogli komunicirati. Ipv4FlatNetworkConfigurator svrstava uređaje u lokalnu mrežu jednakih maski, odnosno podmrežu. Desnim klikom na pojedine veze moguće je odrediti parametre veze kao što su kašnjenje (eng. Delay), stopa prijenosa podataka (eng. Data Rate), stopa greške u bitovima (eng. ber – Bit Error Rate) i stopa krivih paketa (eng. per – Packet Error Rate). Parametre je također moguće dodati i putem koda u NED datoteci, dodavanjem „*{ delay = 100ms; }*” parametra. Sve promjene učinjene putem grafičkog sučelja bit će vidljive i u kodu.

Odabirom na „Source“ izbornik ispod područja mreže, prikazat će se trenutna mreža u obliku koda NED jezika kao što je prikazano na slici 11.23.

```
package udpvideostreamsim.simulations;

import inet.networklayer.configurator.ipv4.Ipv4FlatNetworkConfigurator;
import inet.node.inet.Router;
import inet.node.inet.StandardHost;
import ned.DatarateChannel;
@license(LGPL);

network UdpVideoStreamSimulacija
{
    @display("bgb=736.84,671.94403");
    submodules:
        HostA: StandardHost {
            @display("p=94.64001,617.864");
        }
        HostB: StandardHost {
            @display("p=259.584,617.864");
        }
        HostC: StandardHost {
            @display("p=431.28802,617.864;i=device/laptop");
        }
        router: Router {
            @display("p=259.584,328.536");
        }
        server: StandardHost {
            @display("p=259.584,75.712006;i=device/mainframe");
        }
        ipv4FlatNetworkConfigurator: Ipv4FlatNetworkConfigurator {
            @display("p=590.82404,44.616");
        }
    connections:
        HostA.pppg++ <--> DatarateChannel {  delay = 100ms; datarate = 64kbps; } <--> router.pppg++;
        HostB.pppg++ <--> DatarateChannel {  delay = 50ms; datarate = 64kbps; } <--> router.pppg++;
        HostC.pppg++ <--> DatarateChannel {  delay = 80ms; datarate = 64kbps; ber = 10e-5; per = 10e-5; } <--> router.pppg++;
        router.pppg++ <--> DatarateChannel {  delay = 50ms; datarate = 64kbps; } <--> server.pppg++;
}
```

Slika 11.23 - NED datoteka projekta - izvor: autor - OMNeT++

Dodavanjem „ber“ i „per“ parametara na vezi između klijenta „C“ i usmjerivača, stvorit će se greške u bitovima i paketima kako bi se simulirali gubici paketa zbog loše veze u prijenosu prometa.



Slika 11.24 - konačni izgled mreže - izvor: autor - OMNeT++

Na vezama je primijenjen parametar „datarate“ od 64kbps (Kilobita po sekundi). Za stvaranje aplikacija na modulima i njihovih funkcionalnosti, potrebno je napisati ključne parametre u INI (omnetpp.ini) datoteci koji su prikazani na slici 11.25.

```
[General]
network = UdpVideoStreamSimulacija

*.HostA.numApps = 1
*.HostA.app[*].typename = "UdpVideoStreamClient"
*.HostA.app[*].localPort = 16384
*.HostA.app[*].serverAddress = "server"
*.HostA.app[*].serverPort = 16384  # ULAZ RTP PROTOKOLA
*.HostA.app[*].startTime = 0s

*.HostB.numApps = 1
*.HostB.app[*].typename = "UdpVideoStreamClient"
*.HostB.app[*].localPort = 16384
*.HostB.app[*].serverAddress = "server"
*.HostB.app[*].serverPort = 16384  # ULAZ RTP PROTOKOLA
*.HostB.app[*].startTime = 0s

*.HostC.numApps = 1
*.HostC.app[*].typename = "UdpVideoStreamClient"
*.HostC.app[*].localPort = 16384
*.HostC.app[*].serverAddress = "server"
*.HostC.app[*].serverPort = 16384  # ULAZ RTP PROTOKOLA
*.HostC.app[*].startTime = 0s

*.server.numApps = 1
*.server.app[*].typename = "UdpVideoStreamServer"
*.server.app[*].localPort = 16384
*.server.app[*].sendInterval = normal (10ms, 1ms)
*.server.app[*].packetLen = 250B
*.server.app[*].videoSize = 100MiB

*.server.ppp[*].queue.typename = "DropTailQueue"
*.server.ppp[*].queue.packetCapacity = 9000
**.displayAddresses = true
**.result-recording-modes = all
**.throughput.result-recording-modes = all
**.packets.result-recording-modes = all
```

Slika 11.25 - potrebni parametri za rad modula - omnetpp.ini - izvor: autor - OMNeT++

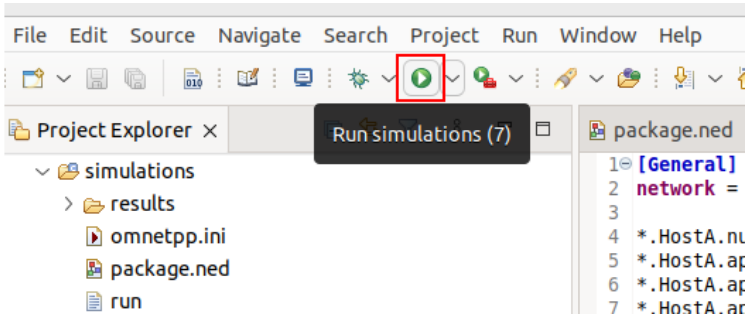
Na početku „omnetpp.ini“ datoteke, potrebno je povezati INI datoteku s mrežom iz NED datoteke (UdpVideoStreamSimulacija). Koristeći (*) označava se mreža „UdpVideoStreamSimulacija“. (*) je „wildcard“ znak koji označava sve elemente, u ovom slučaju element mreže (*UdpVideoStreamSimulacija.HostA.numApps = 1*). Nakon točke označava se ciljani modul (HostA), te se na njega primjenjuju odabrani parametri. „numApps“ parametar određuje broj aplikacija u pojedinom modulu. U ovom će slučaju prvi klijent posjedovati jednu aplikaciju imenom „UdpVideoStreamClient“ iz INET-ove knjižnice. Određivanjem imena aplikacije pomoću „typename“ parametra omogućuje se korištenje novih parametara, specifičnih za UDP Video Stream Aplikaciju. Prvi parametar koji se primjenjuje na aplikaciju je „localPort“ koji određuje izvorni ulaz same aplikacije. Drugi ključan parametar aplikacije je „serverAddress“ koji određuje adresu servera s uslugom. Adresa je „server“ zbog modula ipv4FlatNetworkConfigurator koji automatski određuje adrese uređaja. Treći parametar određuje ciljani ulaz aplikacije, odnosno ulaz na serveru, dok četvrti parametar određuje vrijeme kada će klijent zatražiti Video Streaming uslugu sa servera mreže. Sva tri klijenta posjeduju jednake parametre kako bi popunili „red čekanja“ (eng. Queue) na vratima usmjerivača.

Modulu koji predstavlja server na mreži određena je jedna aplikacija koja će slati UDP pakete Video prometa klijentima na mreži. Pomoću „typename“ parametra serveru je određena „UdpVideoStreamServer“ aplikacija iz INET-ove knjižnice. Kao i kod klijenata, prvi parametar aplikacije je broj ulaza, koji omogućuje klijentima povezivanje sa serverom. Aplikaciji je moguće odrediti vremenske intervale slanja paketa pomoću „sendInterval“ parametra. Parametar „packetLen“ označava duljinu paketa i zaglavlja, a parametar „videoSize“ označava veličinu videodatoteke.

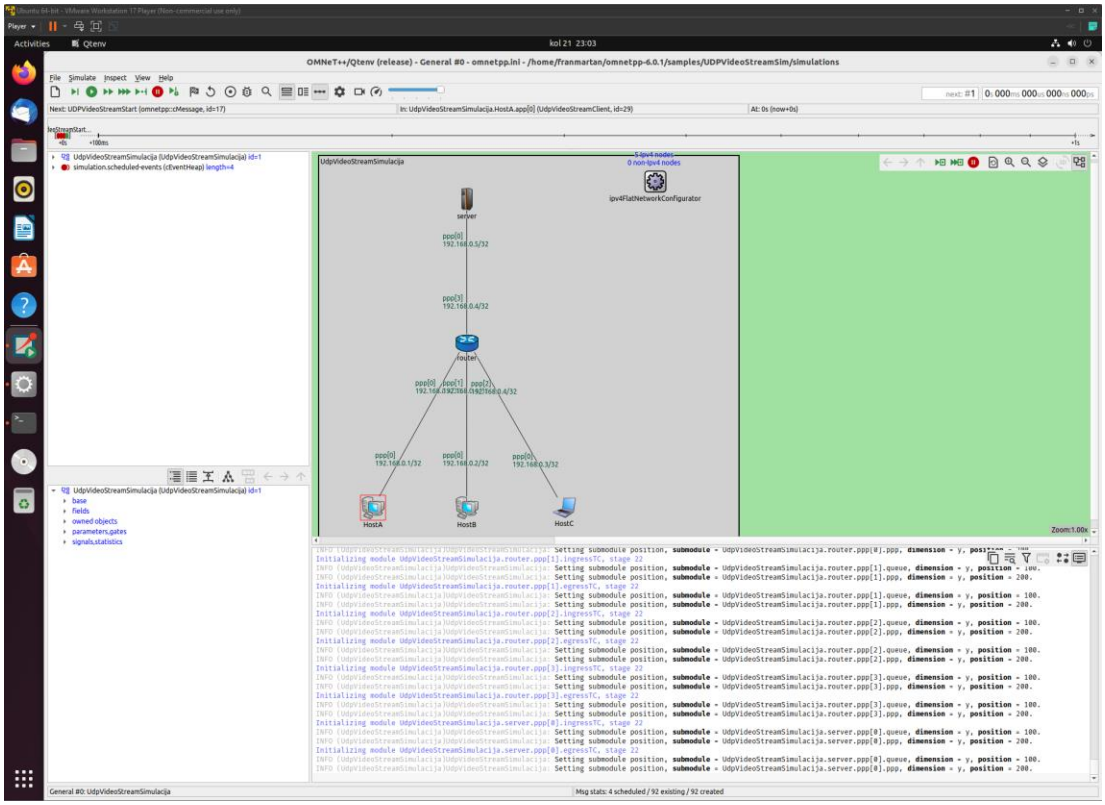
Na server, putem „ppp“ protokola moguće je odrediti parametre za red čekanja. Pomoću „typname“ parametra određuje se tip reda čekanja kao „TailDropQueue“, te kapacitet samog reda u broju paketa.

Pomoću „result-recording-modes“ parametara pohranjuju se rezultati mreže u posebne datoteke za analiziranje podataka. Podaci se mogu iščitavati po kategorijama.

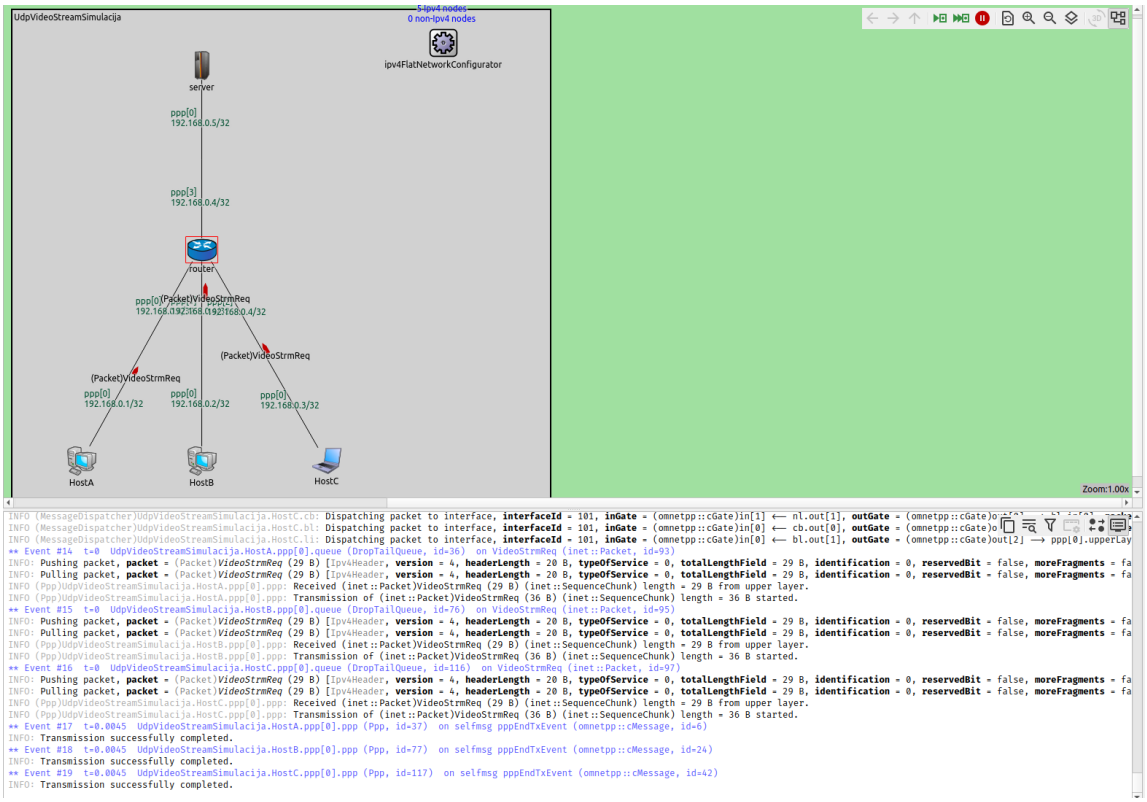
Klikom na „omnetpp.ini“ datoteku iz projekta, te pritiskom na gumb „Run“ pokreće se izgradnja (eng. Build) projekta i prikazuje se simulacija pomoću Qtenv grafičkog sučelja koje je prikazano na slikama 11.27 – 11.29.



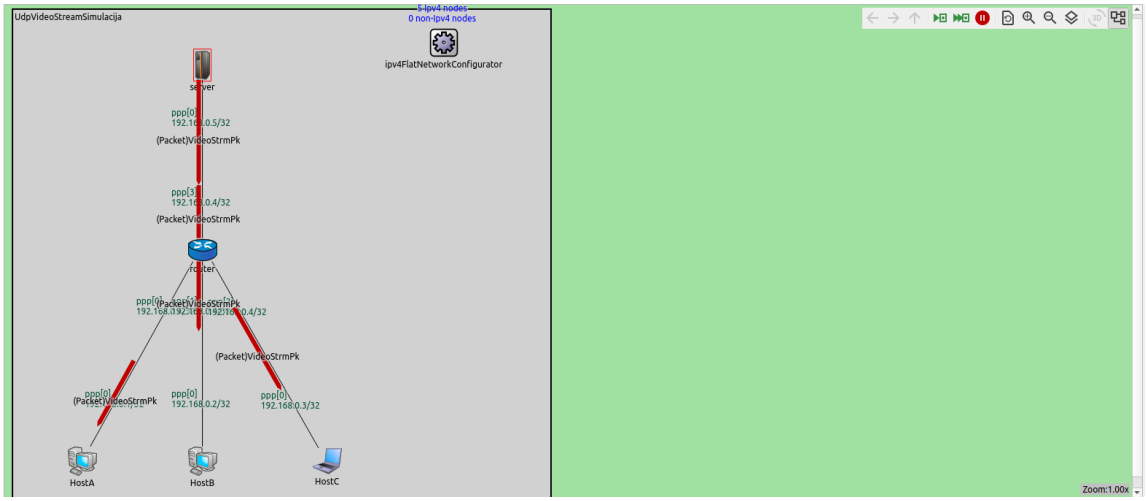
Slika 11.26 - pokretanje simulacije - izvor: autor - OMNeT++



Slika 11.27 - pokretanje qtenv grafičkog sučelja - izvor: autor - OMNeT++

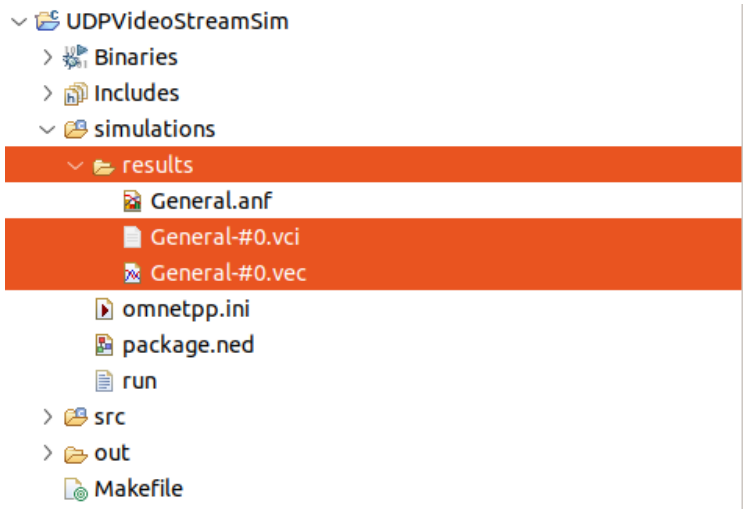


Slika 11.28 - zahtjev za Video Stream - simulacija - izvor: autor - OMNeT++ / qtenv



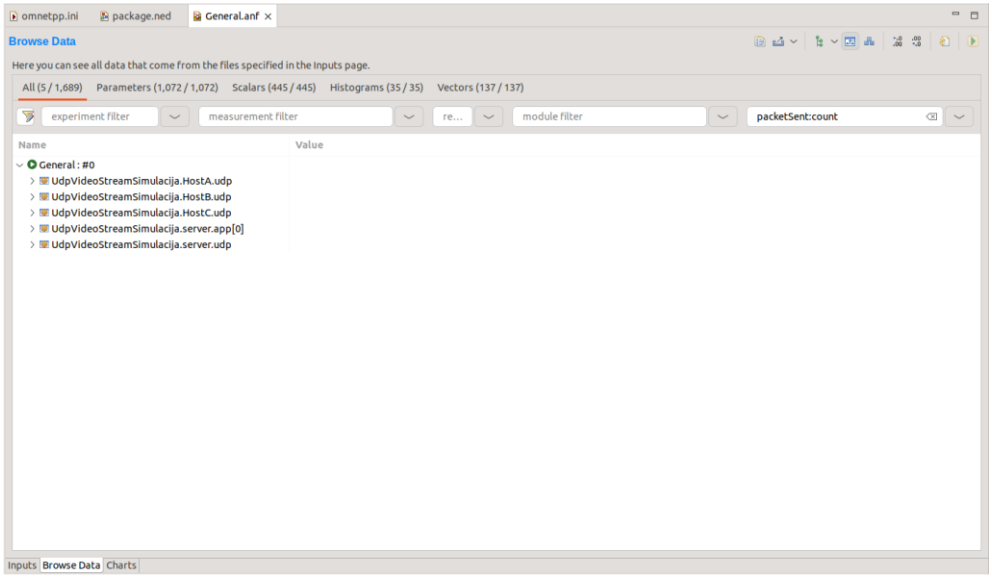
Slika 11.29 - prijenos Video paketa sa servera do klijenata - izvor: autor - OMNeT++ / qtenv

Na početku simulacije moduli pomoću ARP protokola pohranjuju IP i MAC adrese drugih modula u vlastite ARP tablice. Nakon završetka procesa ARP protokola, klijenti šalju video „streaming“ zahtjeve (eng. VideoStream Request) putem veza kroz usmjerivač, pa do modula server. Server obrađuje zahtjeve klijenata, te odmah šalje videopakete prema usmjerivaču koji usmjerava video promet prema destinaciji do klijenata. Aplikacija u server modulu, koristi UDP prijenosni protokol koji prosljeđuje podatke neovisno o uspostavi veze ili greškama u paketima. Simulacija nema određeno vrijeme trajanja, kako bi se prikupilo što više podataka o prijenosu prometa. Na svim je vezama definirano kašnjenje (eng. Delay) u različitim intervalima. UdpVideoStream aplikacija na server modulu slat će vrlo veliki broj paketa koji će brzo popunjavati red čekanja na serveru. Kako bi se otkrio poželjan kapacitet reda čekanja, testirati će se različita vrijednost parametra „packetCapacity“, te kroz rezultate analizirati utjecaj veličine reda čekanja na gubitak paketa u prometu. Zbog velikog broja poslanih paketa, za prvu simulaciju odabrana je vrijednost od 9000 paketa kao kapacitet reda čekanja (eng. Queue). Modul servera pomoću jedne aplikacije pruža uslugu triju različitih klijenata, što će dodatno stvoriti pritisak u redu čekanja servera, te narušiti samu kvalitetu prijenosa. Završetkom simulacije i pokretanjem „finish()“ funkcije, OMNeT++ alat pohranjuje dobivene rezultate u obliku .sca (scalar) i .vec (vector) datoteka u direktorij projekta kao što je prikazano na slici 11.30.



Slika 11.30 - rezultati simulacije - izvor: autor - OMNeT++

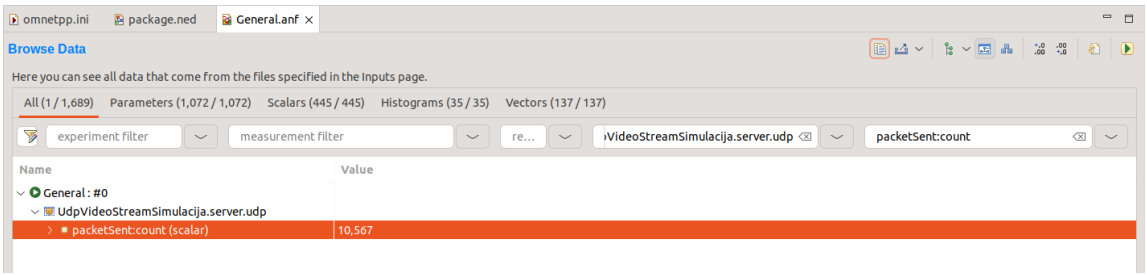
Desnim klikom na projekt i odabirom opcije „New“, potrebno je odabrati „Analysis File (anf)“ i stvoriti novu datoteku analize (.anf). Pokretanjem datoteke „general.anf“ otvara se novi prozor za pregled analize prikazan na slici 11.31.



Slika 11.31 - datoteka analize - izvor: autor - OMNeT++

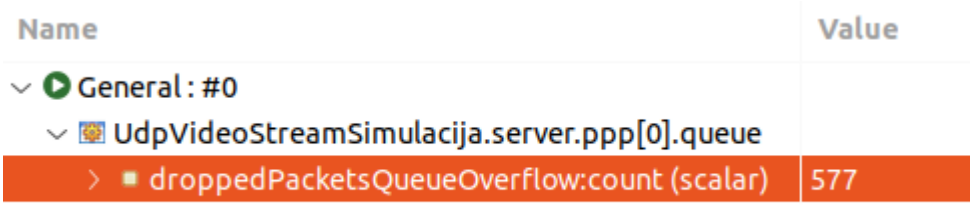
Primjenom typename = „DropTailQueue“ parametra, modulu se omogućuje izbacivanje viška paketa ukoliko se red čekanja maksimalno napuni paketima. Tail Drop na taj način smanjuje pritisak na mrežu, no narušava samu kvalitetu prijenosa multimedijskog sadržaja. Aplikacije koje nude prijenos videosadržaja mogu podnijeti određeni postotak, odnosno stopu izgubljenih paketa do određene mjere, obično do 5% ovisno o aplikaciji. Količine izgubljenih paketa iznad 5% uvelike narušavaju kvalitetu videosadržaja i uzrokuju negledljivost sadržaja. Cisco preporuča 0.1% - 1% stopu izgubljenih paketa, kako bi se osiguralo nesmetano gledanje videosadržaja korisnicima.

U trajanju prve simulacije modul servera poslao je nešto više od 10 000 paketa (10 567) prema svojim klijentima. Mala brzina prijenosa podataka od 64kbps na svim vezama i „Tail Drop“ red čekanja uvjetovali su gubicima paketa u prijenosu videosadržaja na simuliranoj mreži „UdpVideoStreamSimulacija“. Pomoću „general.anf“ datoteke analize moguće je iščitati točne podatke o gubicima paketa zbog Tail Drop funkcije ili greške na bitovima koja je moguća na vezi između klijenta C i usmjerivača.



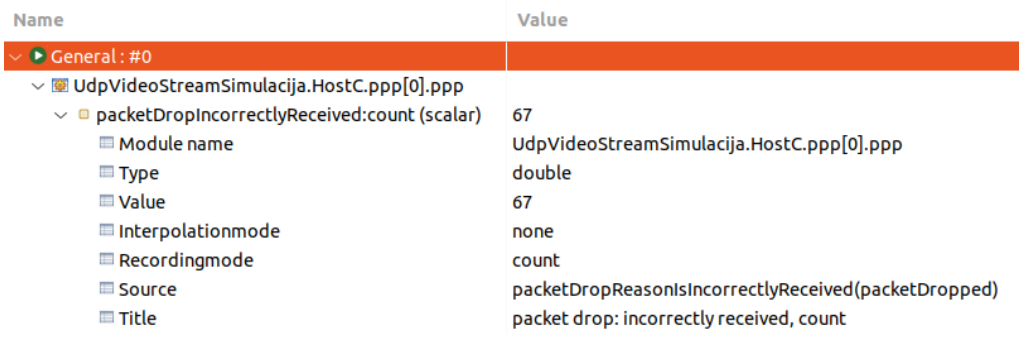
Slika 11.32 - rezultati poslanih paketa od servera prema klijentima putem UDP - izvor: autor - OMNeT++

Među filtracijom podataka potrebno je odabrati opciju „droppedPacketsQueueOverflow“ i odabrati modul „*.server.udp.ppp“. Queue Overflow označava pakete koji su odbačeni zbog punog reda čekanja (izlijevanja paketa). Prikazom analize simulacija očitava 577 odbačenih paketa u redu čekanja na serveru.



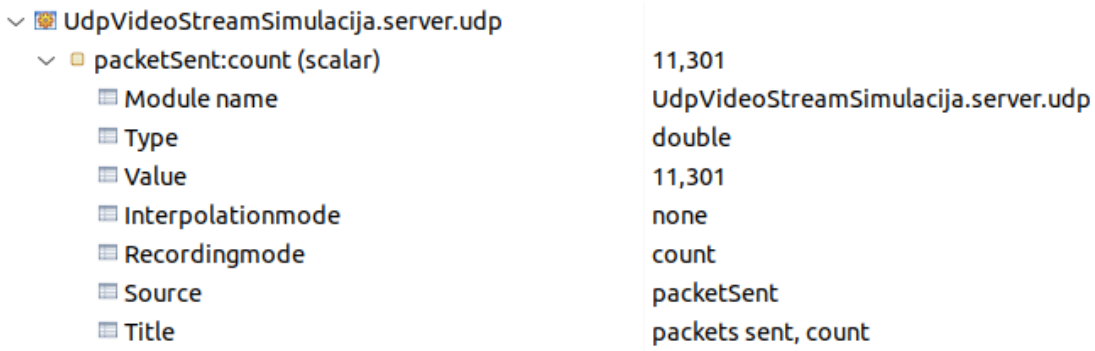
Slika 11.33 - broj odbačenih paketa iz reda čekanja na serveru - izvor: autor - OMNeT++

U prvoj simulaciji stopa izgubljenih paketa iznosi 5,46% što je mnogo više od poželjnog. Takva količina izgubljenih paketa veoma bi narušila kvalitetu videosadržaja i u konačnici ukupno korisničko iskustvo. Na vezi između klijenta C i usmjerivača konfigurirane su moguće greške u paketima koje su dodatno uvjetovale gubicima paketa na mreži. Pomoću analitičkog alata i datoteke analize pod opcijama moguće je očitati broj paketa koji su stigli netočni na modul klijenta C. Potrebno je odabrati filtracijsku opciju imenom „packetDropIncorrectlyReceived:count“ i primijeniti ju na vezu između klijenta C i usmjerivača. Pregledom analize, vidljivo je da je klijent C primio 67 netočnih paketa koji su dodatno narušili korisničko iskustvo klijenta. Videosadržaj klijenta C bio je negledljiv.



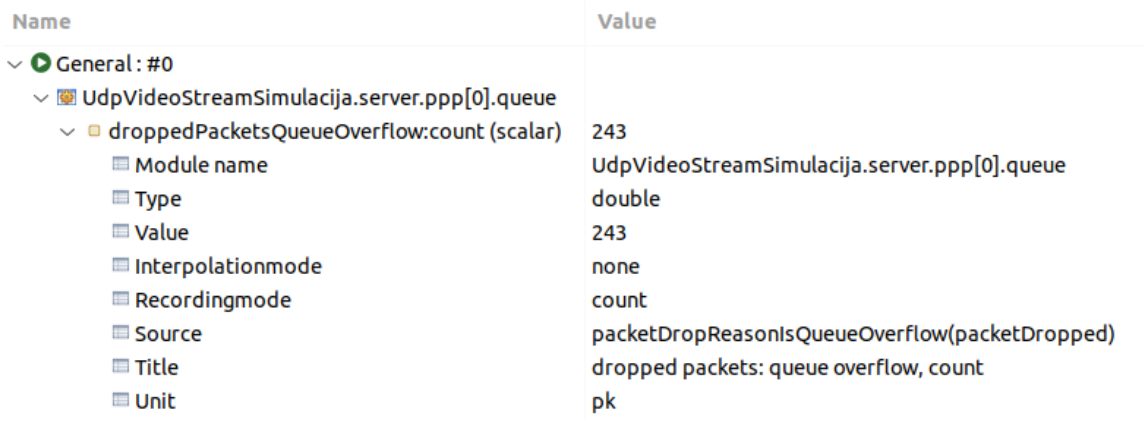
Slika 11.34 - broj netočno primljenih paketa - izvor: autor - OMNeT++

Povećanje kapaciteta reda čekanja na serveru uvjetovalo bi kvalitetnijim sadržajem jer bi se smanjio broj odbačenih paketa zbog Tail Drop-a. Promjenom parametra „packetCapacity“ na vrijednost od 10 000 očitat će se rezultati druge simulacije. Iako je kapacitet u drugoj simulaciji veći za 1000, zbog velike količine prometa potrebne za tri klijenta došlo je do nešto manjih gubitaka paketa zbog odbacivanja. U drugoj je simulaciji modul servera pomoću UDP aplikacije poslao nešto više od 11 000 paketa (11 301) kao što je prikazano na slici 11.35.



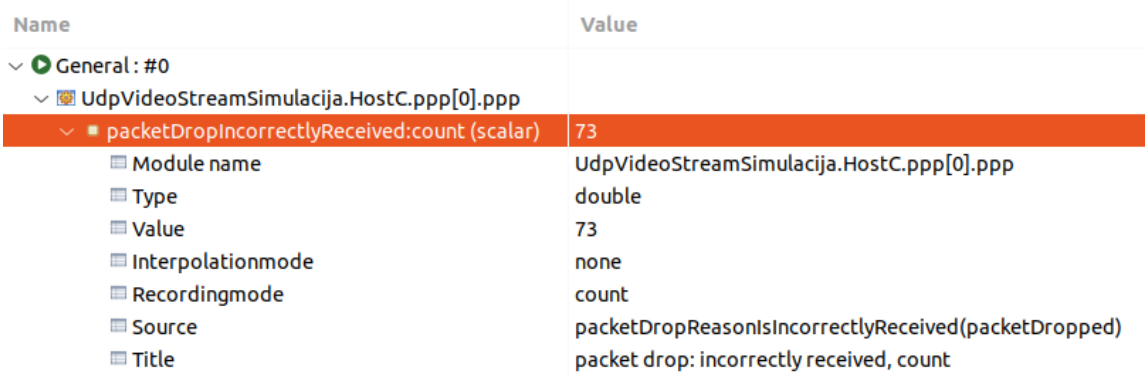
Slika 11.35 - broj poslanih paketa pomoću UDP aplikacije u drugoj simulaciji - izvor: autor - OMNeT++

Veliki broj paketa i u ovom je slučaju izazvao odbacivanje na redu čekanja servera, koje je iznosilo 243 paketa. Gubici od 243 paketa uvjetovali su stopu izgubljenih paketa od 2,15%. Takva količina gubitaka paketa mogla bi biti podnošljiva ovisno o aplikaciji, ali bi bila zamjetna, te narušavala korisničko iskustvo. Greške u videosadržaju bile bi vidljive.

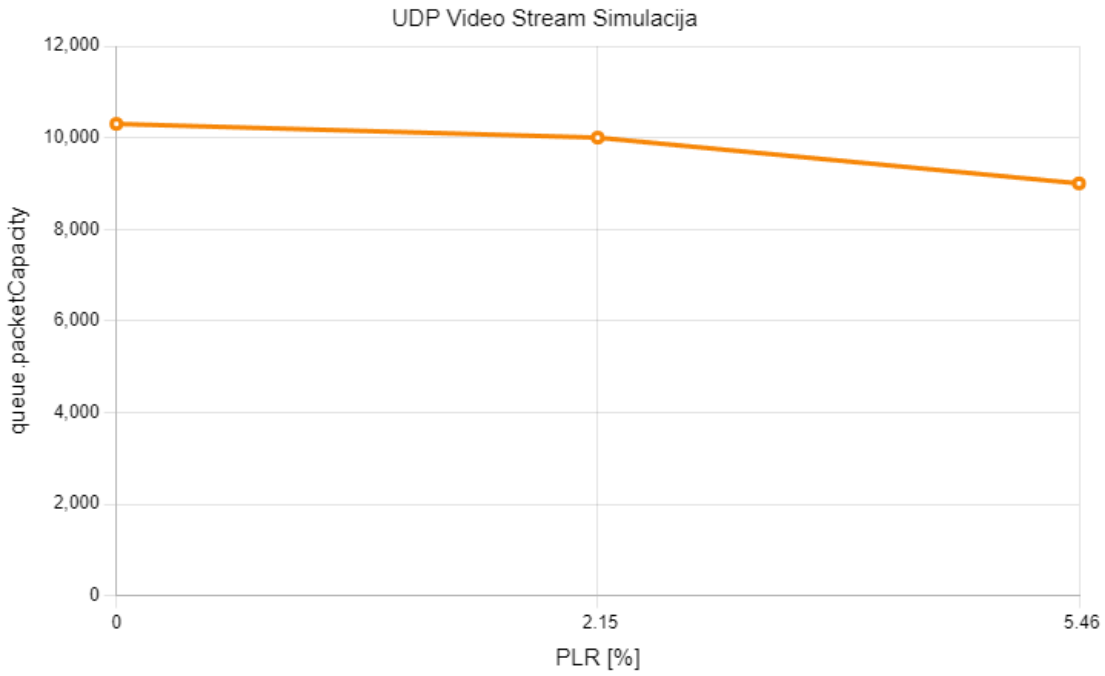


Slika 11.36 - broj izgubljenih paketa u redu čekanja na serveru - izvor: autor - OMNeT++

Povećanjem kapaciteta za 300 mogućih paketa, na 10 300 uvjetovalo je kvalitetnom prijenosu videosadržaja. Klijent A i B imali su besprijekoran prijenos podataka, te odlično korisničko iskustvo, dok su greške u paketima na lošijoj vezi između klijenta C i usmjerivača uvjetovali netočno primljenim paketima. Promet koji je prolazio kroz vezu izgubio je ukupno 73 paketa, što bi moglo dodatno utjecati na samu kvalitetu multimedijskog sadržaja koju je primio klijent C. Stopa izgubljenih paketa je 0,7%.



Slika 11.37 - broj izgubljenih paketa na vezi između klijenta C i usmjerivača - izvor: autor - OMNeT++

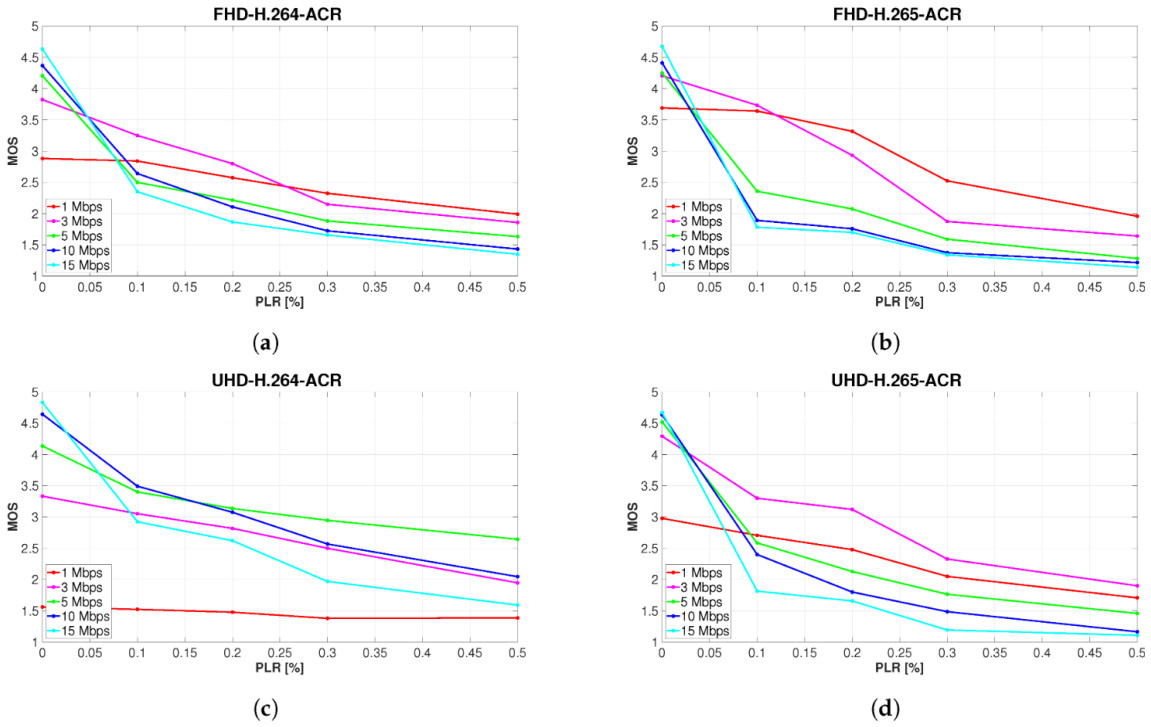


Slika 11.38 - prikaz povećanja stope izgubljenih paketa ovisno o kapacitetu reda čekanja - izvor: autor

Prema prikazu 11.38 može se zaključiti kako je sam kapacitet reda čekanja uvelike utjecao na stopu izgubljenih paketa. U posljednjoj simulaciji s najvećim kapacitetom, Tail Drop nije uzrokovao ni jedan izgubljeni paket, nego je samo klijent C imao stopu izgubljenih paketa od 0,7% zbog loše veze koja ga spaja s usmjerivačem.

Iako proizvođači kao Cisco, preporučaju određene stope izgubljenih paketa, sami gubici paketa mogu različito utjecati na određeni promet ovisno o aplikaciji, vrsti video kompresije ili brzini prijenosa. Što znači da na pojedine aplikacije stopa izgubljenih paketa od samo 0,5% u kombinaciji s video kompresijom može znatno utjecati na kvalitetu multimedijskog sadržaja i u konačnici na samo korisničko iskustvo. Trenutno najpoznatiji oblik video kompresije je HEVC/H.265 (eng. High Efficiency Video Coding). H.265 nastao je kao unaprjeđenje prijašnjem H.264 obliku. H.265 može pružiti jednaku kvalitetu video sadržaja pri nižem bitrate-u, te podržava i više rezolucije, kao što su to 4K i 8K rezolucije.

Rezultate simulacija potrebno je usporediti s realnim istraživanjima i situacijama. U znanstvenom radu [16] izdanom 2023. godine istražuje se utjecaj stope izgubljenih paketa na kvalitetu komprimiranog video sadržaja visokih rezolucija. U prijenosu video sadržaja visoke rezolucije pri većim brzinama prijenosa autori iz istraživanja zaključuju kako već i mali postotak izgubljenih paketa može utjecati na subjektivni doživljaj korisnika. Istraživanje se provodilo u četiri slučaja, za „Full High Definition - FHD“ video sadržaj komprimiran H.264 i H.265 kodekom, te za „Ultra High Defintion – UHD“ video sadržaj komprimiran H.264 i H.265 kodekom s različitom stopom prijenosa podataka prikazanih na slici 11.39. MOS – „Mean Opinion Score“.

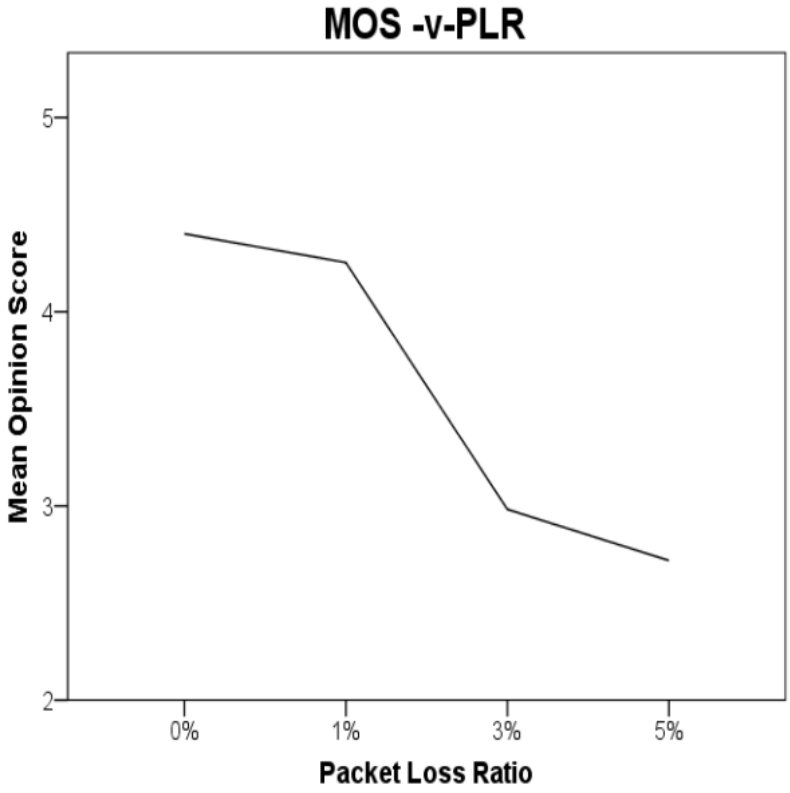


Slika 11.39 - razvoj subjektivno percipirane kvalitete video sadržaja s povećanjem stope izgubljenih paketa - izvor: „Impact of Packet Loss Rate on Quality of Compressed High Resolution Videos, Juraj Bienik, Miroslav Uhrina, Lukas Sevcik, Anna Holesova“

Prema rezultatima istraživanja, može se zamijetiti da vrlo mala stopa izgubljenih paketa počinje utjecati na subjektivni doživljaj video sadržaja visoke rezolucije, te da je potrebna veća brzina prijenosa kako bi korisničko iskustvo moglo biti zadovoljavajuće. U slučaju a) za UHD video sadržaj može se zamijetiti kako je spor prijenos podataka od 1Mbps uvjetovao veoma lošem subjektivnom doživljaju korisnika. U sva se četiri slučaja može zamijetiti kako su izgubljeni paketi znatno utjecali na korisničko iskustvo, te je stopa od 0,5% izgubljenih paketa uvjetovala ne zadovoljavajućem korisničkom iskustvu u svim slučajevima.

Usporedba istraživanja sa simulacijama dokazuje da bi prijenos video sadržaja visoke rezolucije rezultirao vrlo lošim subjektivnim doživljajem korisnika zbog malih brzina prijenosa i visokih stopa izgubljenih paketa. U trećoj simulaciji u prijenosu paketa od servera do korisnika A i B nije bilo izgubljenih paketa, no zbog male brzine prijenosa od samo 64Kbps sadržaj bi previše kasnio i znatno utjecao na korisničko iskustvo. Prijenos video sadržaja od servera do korisnika C, zbog loše veze izgubio je 0,7% paketa, što bi prema istraživanjima uvjetovalo „ne gledljivim“ sadržajem iako je video komprimiran. Konfiguracijom veza, te omogućavanjem veće brzine prijenosa od najmanje 3Mbps osiguralo bi kvalitetniji subjektivni doživljaj treće simulacije UDP Video Stream prometa, red čekanja bi se manje punio što bi omogućilo nesmetani prijenos prometa.

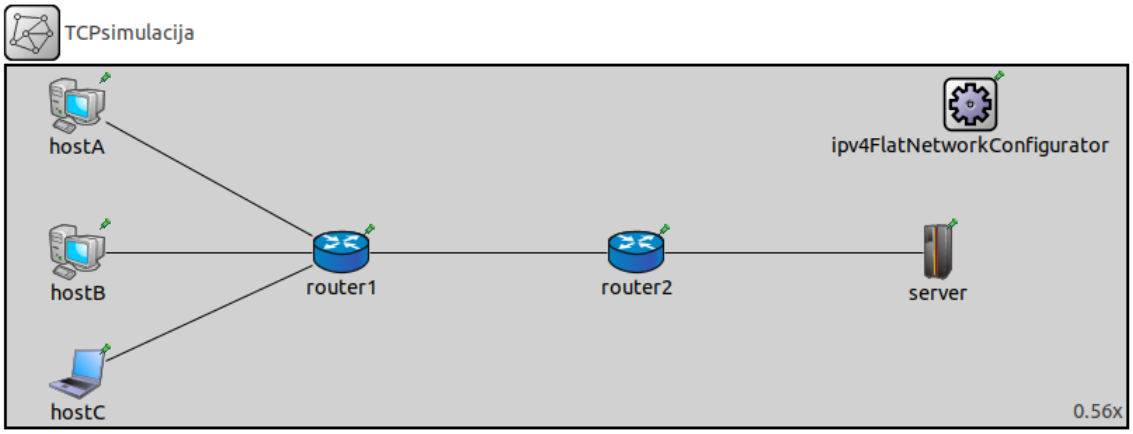
U istraživanju [20], autori istražuju utjecaj stope izgubljenih paketa na video sadržaj nižih rezolucija, komprimiran HEVC/H.265 kodekom. Prema istraživanjima zaključuje se da je utjecaj stope izgubljenih paketa na subjektivni doživljaj nešto slabiji pri video sadržaju nižih rezolucija. Korisnici uočavaju zamjetljive razlike pri stopi izgubljenih paketa bliže 1%, no opisuju sadržaj kao „gledljiv“ video sadržaj. Znatne razlike u prijenosu koje ometaju korisnike događaju se na stopi izgubljenih paketa od 3%, te korisnici tada opisuju video sadržaj kao „ne gledljiv“. Slika 11.40 prikazuje utjecaj povećane stope izgubljenih paketa na subjektivni doživljaj krajnjeg korisnika.



Slika 11.40 - prosjek subjektivnog doživljaja za svaku kategoriju gubitaka paketa, uz prikaz prosječnog smanjenja subjektivnog doživljaja između 0% i 5% PLR - izvor: „The Impact of Network Impairment on Quality of Experience (QoE) in H.265/HEVC video streaming“ in IEEE Transactions on Consumer Electronics, J Nightingale, Q. Wang, C Grecos, S.Goma

Usporedba rezultata istraživanja, s rezultatima simulacija pokazuje da bi prijenos video sadržaja nižih rezolucija bio moguć, s manjim poteškoćama. U prvoj simulaciji vrlo visoka stopa izgubljenih paketa od 5,46% onemogućuje kvalitetan subjektivan doživljaj, te je sama kvaliteta video sadržaja vrlo loša. Povećanjem kapaciteta reda čekanja u drugoj simulaciji smanjila se stopa izgubljenih paketa na 2,15% što bi narušilo subjektivan doživljaj korisnika pri prijenosu, no video sadržaj vrlo niskih rezolucija bi mogao biti podnošljiv. U trećoj simulaciji s većim kapacitetom reda čekanja omogućen je nesmetan promet između servera i klijenata A i B bez stope izgubljenih paketa, no prema klijentu C pojavljuje se gubitak od 0,7%. U slučaju treće simulacije korisničko iskustvo svih troje korisnika bilo bi zadovoljavajuće prema istraživanjima navedenog znanstvenog rada [20].

11.2.3. Simulacija gubitka paketa u prijenosu Web prometa putem TCP prijenosnog protokola



Slika 11.41 - vizualni prikaz simulirane TCP mreže - izvor: autor - OMNeT++

U simulaciji prijenosa Web prometa putem TCP prijenosnog protokola utjecat će se na red čekanja prvog usmjerivača kako bi se simulirali gubici paketa. Prvi usmjerivač povezuje troje klijenata koji će putem „http“ protokola prenijeti Web podatke na server u simuliranoj mreži prikazanoj na slici 11.41. Podaci moraju proći dva usmjerivača kako bi stigli do servera ili obratno. TCP simulacija također koristi „ipv4FlatNetworkConfigurator“ modul koji omogućuje dinamičku dodjelu IPv4 adresa uređajima na mreži. Korištene veze između modula tipa su „Datarate Channel“ i konfigurirane su s određenom brzinom prijenosa, te nemaju mogućnost greške u paketima. Najveći promet događat će se na prvom usmjerivaču, te mu je konfiguriran Tail Drop red čekanja.

Klijenti koriste TCP aplikaciju dodanu u njihove module kako bi stvorili Web promet koji uobičajeno putuje preko TCP ulaza 80 za „http“ ili 443 za „https“. Server koristi TCP aplikaciju koja će primati promet dobiven od klijenata kroz usmjerivače. Klijenti šalju podatke različitih veličina na jedan ulaz. Nakon pokretanja simulacije rezultati će se analizirati kroz ANF datoteku analize alata OMNeT++.

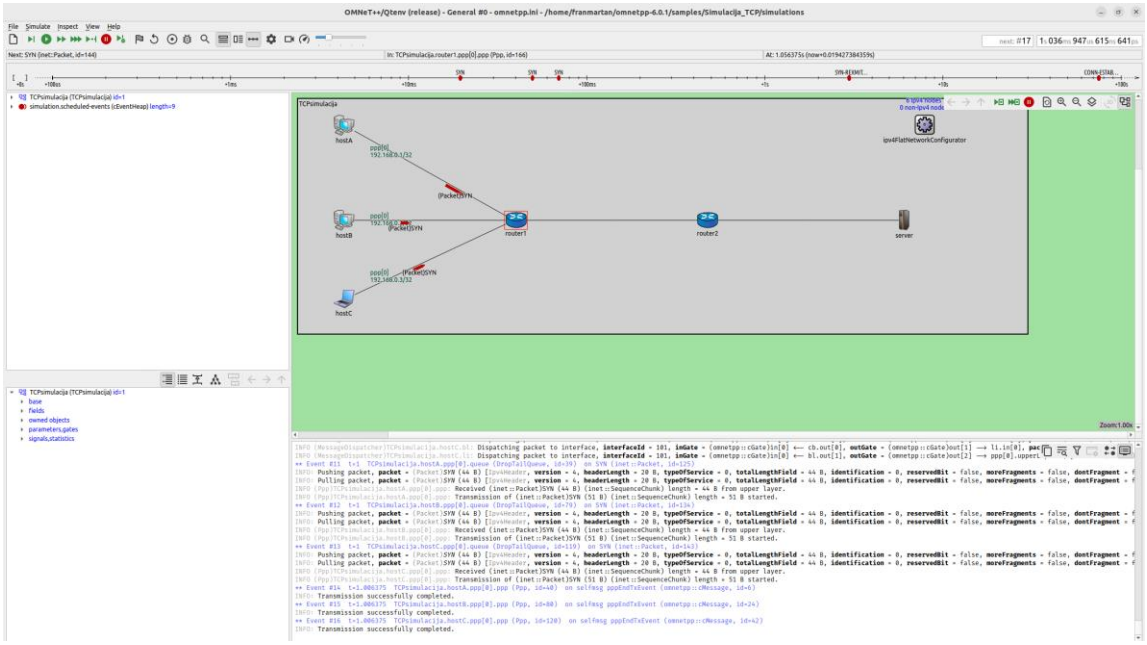
```
network TCPsimulacija
{
    @display("bgb=1413.4088,454.1625");
    submodules:
        hostA: StandardHost {
            @display("p=89.45625,50.921253");
        }
        hostB: StandardHost {
            @display("p=89.45625,235.33876");
        }
        hostC: StandardHost {
            @display("p=89.45625,388.1025;i=device/laptop");
        }
        router1: Router {
            @display("p=422.50876,235.33876");
        }
        router2: Router {
            @display("p=794.09625,235.33876");
        }
        server: StandardHost {
            @display("p=1175.3175,235.33876;i=device/mainframe");
        }
        ipv4FlatNetworkConfigurator: Ipv4FlatNetworkConfigurator {
            @display("p=1215.0775,49.594997");
        }
    connections:
        hostA.pppg++ <--> DatarateChannel {  delay = 50ms; datarate = 64kbps; } <--> router1.pppg++;
        hostB.pppg++ <--> DatarateChannel {  delay = 100ms; datarate = 64kbps; } <--> router1.pppg++;
        hostC.pppg++ <--> DatarateChannel {  delay = 80ms; datarate = 64kbps; } <--> router1.pppg++;
        router1.pppg++ <--> DatarateChannel {  delay = 90ms; datarate = 64kbps; } <--> router2.pppg++;
        router2.pppg++ <--> DatarateChannel {  delay = 90ms; datarate = 64kbps; } <--> server.pppg++;
}
```

Slika 11.42 - kod NED datoteke TCP simulacije - izvor: autor - OMNeT++

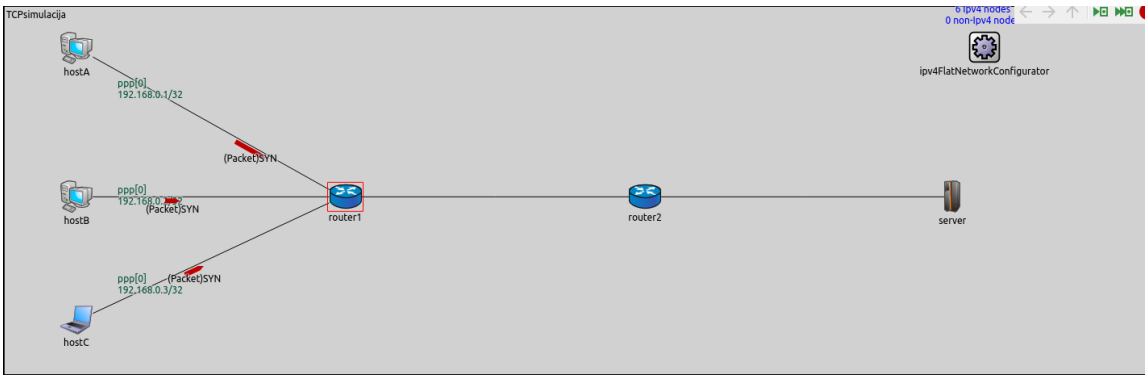
```
[General]
network = TCPsimulacija

*.hostA.numApps = 1
*.hostA.app[*].typename = "TcpSessionApp"
*.hostA.app[*].connectAddress = "server"
*.hostA.app[*].connectPort = 80
*.hostA.app[*].dataTransferMode = "object"
*.hostA.app[*].sendBytes = 9MiB
*.hostA.app[*].localPort = 80

*.host*.*.displayAddresses = true
*.host*.hasTcp = true
*.server.hasTcp = true

*.hostB.numApps = 1
*.hostB.app[*].typename = "TcpSessionApp"
*.hostB.app[*].connectAddress = "server"
*.hostB.app[*].connectPort = 80
*.hostB.app[*].dataTransferMode = "object"
*.hostB.app[*].sendBytes = 13MiB
*.hostB.app[*].localPort = 80

*.hostC.numApps = 1
*.hostC.app[*].typename = "TcpSessionApp"
*.hostC.app[*].connectAddress = "server"
*.hostC.app[*].connectPort = 80
*.hostC.app[*].dataTransferMode = "object"
*.hostC.app[*].sendBytes = 7MiB
*.hostC.app[*].localPort = 80


*.server.numApps = 1
*.server.app[*].typename = "TcpSinkApp"
*.server.app[*].localAddress = ""
*.server.app[*].localPort = 80
*.server.app[*].packetReceived.result-recording-modes = all

*.router1.ppp[*].queue.typename = "DropTailQueue"
*.router1.ppp[*].queue.packetCapacity = 10

**.ppp[*].packetDropped.bin-recording = true

**.channel.throughput.result-recording-modes = all
**.channel.packets.result-recording-modes = all
**.channel.packetBytes.result-recording-modes = all
```

Slika 11.43 - kod omnetpp.ini datoteke TCP simulacije - izvor: autor - OMNeT++

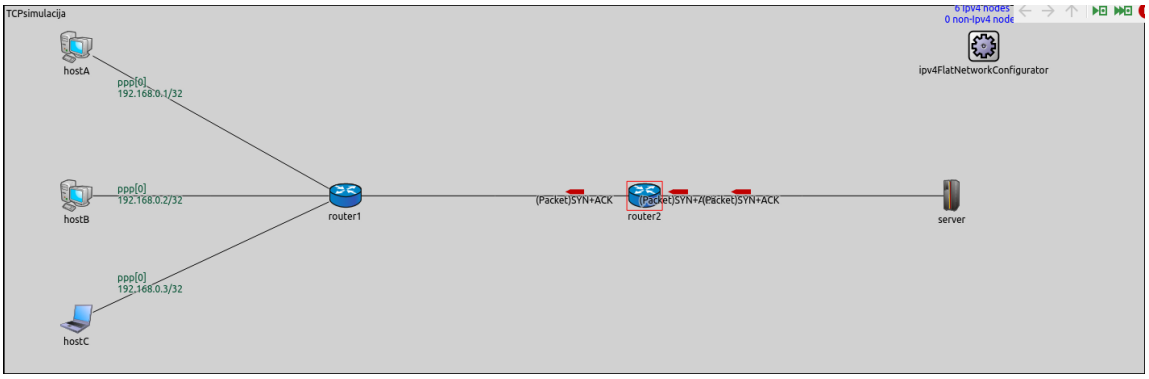
Klijenti A,B i C koriste „TcpSessionApp“ aplikacije koje generiraju pojedine TCP sesije na Webu. Klijenti se povezuju na TCP ulaz 80 koji se koristi za „http“ (HyperText Transfer Protocol) protokol. Parametar „sendBytes“ određuje količinu podataka koju će poslati svaki klijent. U stvorenoj TCP simulaciji, prvi će klijent slati 9Mb podataka, drugi 13Mb i treći klijent 7Mb. Na modulu servera nalazi se aplikacija „TcpSinkApp“ koja će primati sve TCP pakete na ulazu 80. Na sam red čekanja prvog usmjerivača primijenjen je parametar typename = „DropTailQueue“ i vrlo mali kapacitet reda čekanja od deset paketa. Slike 11.44 – 11.48 prikazuju događaje u TCP simulaciji.



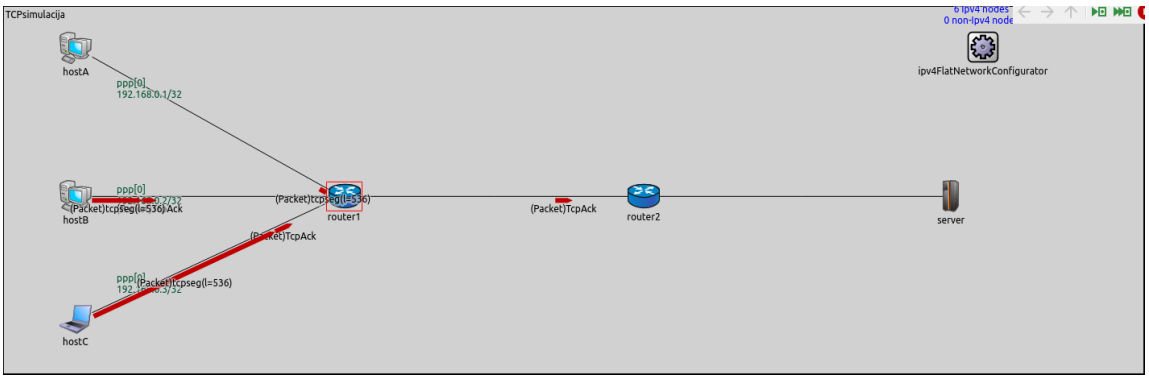
Slika 11.44 - povezivanje TCP protokola, slanje SYN poruke prema serveru - izvor: autor - OMNeT++



Slika 11.45 - SYN poruka klijenata - izvor: autor - OMNeT++

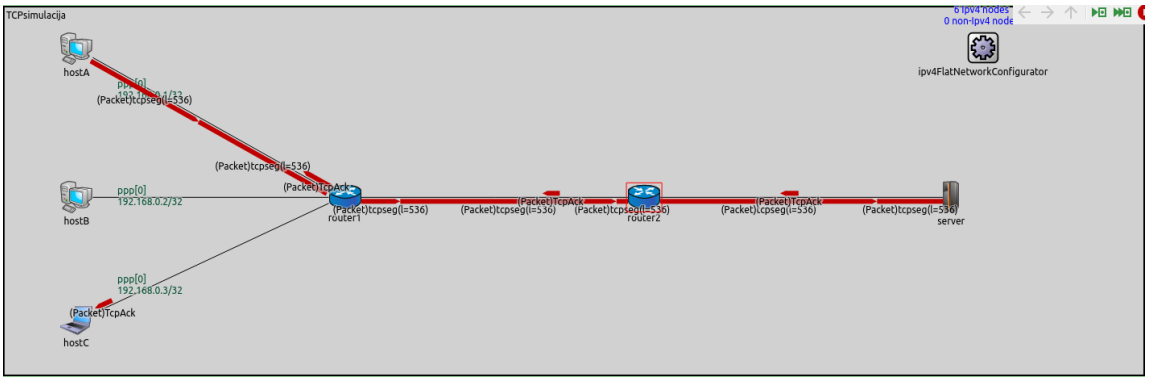


Slika 11.46 - SYN/ACK poruka servera - izvor: autor - OMNeT++



Slika 11.47 - ACK poruka klijenata - izvor: autor - OMNeT++

Kako bi TCP protokol mogao započeti prijenos podataka, potrebno je utvrditi uspostavu veze pomoću „Three-Way Handshake-a“. Three-Way Handshake sastoji se od tri poruke (SYN, SYN/ACK, ACK) kojima potvrđuje uspostavu veze i dozvoljen prijenos podataka. Klijenti prvi šalju SYN (eng. Synchronize) prema serveru, koji kada primi poruku odgovara sa SYN/ACK (Synchronize and Acknowledge) i potvrđuje primitak SYN poruke. Zadnji korak je da klijenti potvrde uspostavu veze putem ACK (eng. Acknowledge) poruke. Nakon završetka Three-Way Handshake procesa klijenti počinju slati Web podatke prema server modulu i njegovoj TcpSink aplikaciji. Podaci prolaze kroz dva usmjerivača brzinom od 64kbps kako bi stigli do servera. Kašnjenje na vezama uvjetuje koji će podaci stići prije od ostalih. Simulacija nema određeno vrijeme kako bi se prikupilo čim više podataka potrebnih za precizniju analizu stope izgubljenih paketa.



Slika 11.48 - prijenos web prometa i ACK odgovor - izvor: autor - OMNeT++

U trajanju simulacije može se zamijetiti da na svaki primljeni paket, server ponovo odgovara ACK porukom kako bi klijenti znali da je server sigurno primio podatke. Ukoliko ne stigne ACK odgovor, za razliku od UDP protokola, TCP server može ponovo zatražiti pakete koji nisu stigli ili su odbačeni. TCP šaljući SYN i ACK poruke koristi nešto više resursa mreže nego UDP.

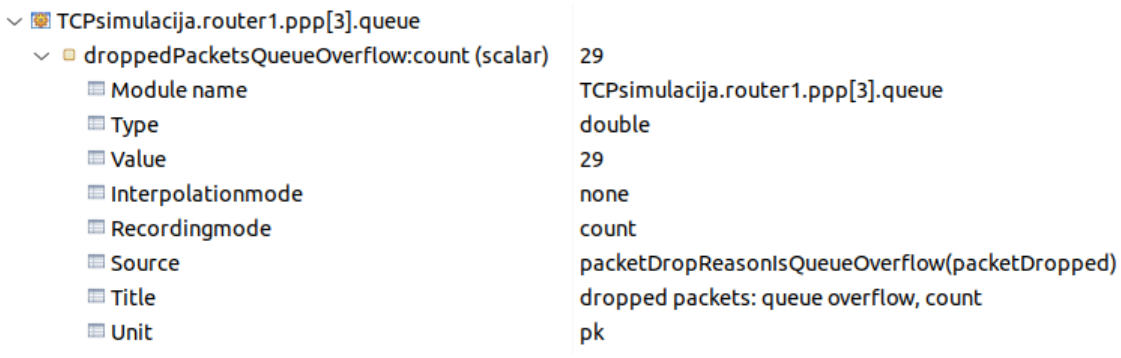
```
INFO: Sending (inet::Packet)TcpAck (40 B) (inet::SequenceChunk) length = 47 B to upper layer.
INFO (MessageDispatcher)TCPsimulacija.router1.li: Dispatching packet to protocol, protocol = ipv4(38), servicePrimitive = 2, inGate = (omnetpp::cGate)in[5] <-- ppp[3].upperLayerOut, outGate =
INFO (MessageDispatcher)TCPsimulacija.router1.bl: Dispatching packet to protocol, protocol = ipv4(38), servicePrimitive = 2, inGate = (omnetpp::cGate)in[1] <-- li.out[0], outGate = (omnetpp::
INFO (MessageDispatcher)TCPsimulacija.router1.cb: Dispatching packet to protocol, protocol = ipv4(38), servicePrimitive = 2, inGate = (omnetpp::cGate)in[0] <-- bl.out[0], outGate = (omnetpp::
INFO (MessageDispatcher)TCPsimulacija.router1.nl: Dispatching packet to protocol, protocol = ipv4(38), servicePrimitive = 2, inGate = (omnetpp::cGate)in[1] <-- cb.out[1], outGate = (omnetpp::
INFO (MessageDispatcher)TCPsimulacija.router1.ipv4.lp: Dispatching packet to protocol, protocol = ipv4(38), servicePrimitive = 2, inGate = (omnetpp::cGate)in[2] <-- <parent>.ifIn, (ned.IdealC
** Event #656  t=3.8285  TCPsimulacija.router1.ipv4.ip (Ipv4, id=202)  on TcpAck (inet::Packet, id=498)
INFO: Received (inet::Packet)TcpAck (40 B) (inet::SequenceChunk) length = 47 B from network.
DETAIL: Received datagram `' with dest=192.168.0.1
INFO: Routing (inet::Packet)TcpAck (40 B) (inet::SequenceChunk) length = 40 B with destination = 192.168.0.1, output interface = ppp0, next hop address = <unspec>
INFO: Sending (inet::Packet)TcpAck (40 B) (inet::SequenceChunk) length = 40 B to output interface = ppp0.
INFO (MessageDispatcher)TCPsimulacija.router1.ipv4.lp: Dispatching packet to interface, interfaceId = 101, inGate = (omnetpp::cGate)in[1] <-- ip.queueOut, outGate = (omnetpp::cGate)out[2] -->
INFO (MessageDispatcher)TCPsimulacija.router1.nl: Dispatching packet to interface, interfaceId = 101, inGate = (omnetpp::cGate)in[0] <-- ipv4.ifOut, outGate = (omnetpp::cGate)out[1] --> cb.in
INFO (MessageDispatcher)TCPsimulacija.router1.cb: Dispatching packet to interface, interfaceId = 101, inGate = (omnetpp::cGate)in[1] <-- nl.out[1], outGate = (omnetpp::cGate)out[0] --> bl.in[
INFO (MessageDispatcher)TCPsimulacija.router1.bl: Dispatching packet to interface, interfaceId = 101, inGate = (omnetpp::cGate)in[0] <-- cb.out[0], outGate = (omnetpp::cGate)out[1] --> li.in[
INFO (MessageDispatcher)TCPsimulacija.router1.li: Dispatching packet to interface, interfaceId = 101, inGate = (omnetpp::cGate)in[0] <-- bl.out[1], outGate = (omnetpp::cGate)out[2] --> ppp[0]
** Event #657  t=3.8285  TCPsimulacija.router1.ppp[0].queue (DropTailQueue, id=165)  on TcpAck (inet::Packet, id=498)
INFO: Pushing packet, packet = (Packet)TcpAck (40 B) [Ipv4Header, version = 4, headerLength = 20 B, typeOfService = 0, totalLengthField = 40 B, identification = 16, reservedBit = false, moreF
INFO: Pulling packet, packet = (Packet)TcpAck (40 B) [Ipv4Header, version = 4, headerLength = 20 B, typeOfService = 0, totalLengthField = 40 B, identification = 16, reservedBit = false, moreF
INFO (Ppp)TCPsimulacija.router1.ppp[0].ppp: Received (inet::Packet)TcpAck (40 B) (inet::SequenceChunk) length = 40 B from upper layer.
INFO (Ppp)TCPsimulacija.router1.ppp[0].ppp: Transmission of (inet::Packet)TcpAck (47 B) (inet::SequenceChunk) length = 47 B started.
** Event #658  t=3.834375  TCPsimulacija.router1.ppp[0].ppp (Ppp, id=166)  on selfmsg pppEndTxEvent (omnetpp::cMessage, id=60)
INFO: Transmission successfully completed.
```

Slika 11.49 - događaji u simulaciji - izvor: autor - OMNeT++

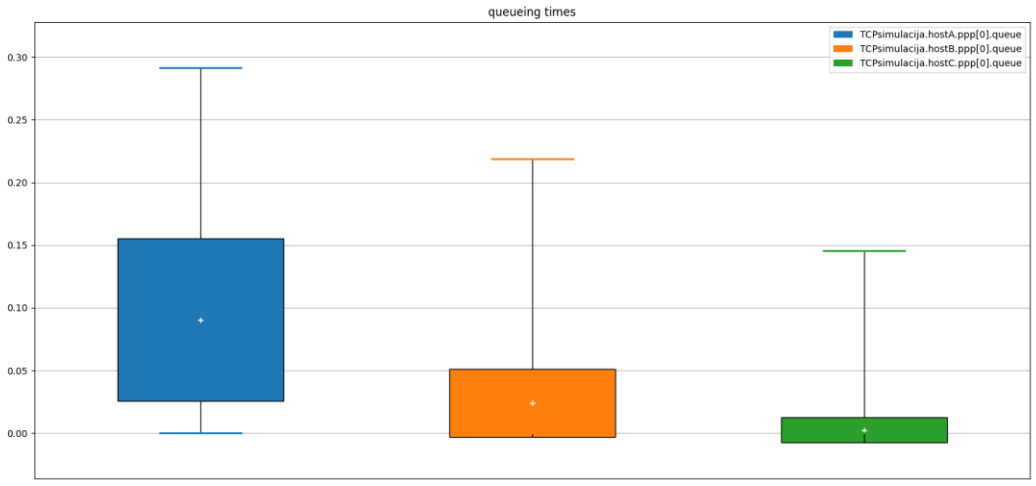
Ispod prikaza simulacije među događajima na slici 11.49 može se uočiti paket koji je izgubljen zbog Tail Drop parametra na redu čekanja. Uočljive su i ACK poruke pristigle od servera kako bi klijenti znali da su podaci primljeni.

Nakon završetka trajanja simulacije, pomoću .sca i .vec datoteka stvorit će se datoteka analize, te će se moći provjeriti točne brojke izgubljenih paketa u prijenosu preko mreže.

Klijenti su zajedno u simulaciji poslali nešto više od 2500 paketa (2527) koji su prolazili kroz usmjerivače do servera. Povećani promet na prvom usmjerivaču uvjetovao je odbacivanjem manje količine izgubljenih paketa. Iznos paketa koji su se izgubili na prvom usmjerivaču je 29 kao što je prikazano na slici 11.50. Stopa izgubljenih paketa u TCP simulaciji iznosila bi 1,14%, te se nalazi u preporučljivim iznosima za nesmetani web promet. Preporuka za nesmetani web promet je stopa izgubljenih paketa od maksimalno 1% - 2%. Stope iznad 2% uvelike narušavaju kvalitetu web sadržaja i produžuju vrijeme učitavanja web stranica. Stopa od 1,14% uvjetovala je nesmetanom prijenosu web prometa bez naročitih grešaka.



Slika 11.50 - broj odbačenih paketa u redu čekanja na prvom usmjerivaču - izvor: autor - OMNeT++



Slika 11.51 - vrijeme zadržavanja podataka u redovima čekanja - izvor: autor - OMNeT++

12. Zaključak

U diplomskom radu pobliže se istražuju ključne komponente komunikacijskih mreža, kao što su ključni mrežni protokoli, mrežne tehnologije i uređaji, te kakav je utjecaj gubitka paketa na sam rad mreže. Diplomski rad opisuje arhitekturu OMNeT++ simulacijskog alata i INET programskog okvira i proces analize utjecaja gubitka paketa kako bi se dobio uvid u dinamiku mrežne komunikacije.

Ključni prijenosni protokoli TCP i UDP predstavljaju temeljne elemente komunikacijskih mreža sa svojim prednostima i ograničenjima. TCP (eng. Transmission Control Protocol) prijenosni protokol usmjeren je na pouzdanu i redoslijednu isporuku podataka, dok UDP (eng. User Datagram Protocol) nudi bržu i jednostavniju komunikaciju bez potvrde o primitku. Razumijevanje razlike u prijenosnim protokolima omogućuje jednostavniju prilagodbu odabira potrebnog protokola za realizaciju specifičnih zahtjeva mrežnih opcija kako bi se postigle najoptimalnije performanse i korisničko iskustvo. Aplikacije ovisno o svojim potrebama biraju koje će protokole koristiti.

OMNeT++ simulacijski alat, posjeduje modularnu strukturu alata koji pružaju platformu za stvaranje simulacija i modela svih vrsta komunikacijskih mreža. OMNeT++ omogućuje stvaranje i praćenje različitih scenarija, evaluaciju rezultata i performansi, te optimizaciju samog dizajna mreže. Simulacije omogućuju kraće vrijeme i resurse potrebne za implementaciju stvarne produkcijske mreže. Uz OMNeT++ simulacijski alat moguće je instalirati INET programski okvir koji omogućuje korisnicima pristup širokom spektru gotovih modula, protokola, aplikacija ili komponenata mreže. INET na taj način uvelike ubrzava proces modeliranja mreža, te omogućuje bržu efikasniju i bržu analizu performansi. INET programski okvir podržava veliki broj vrsta komunikacijskih mreža kao što su žičane, bežične, mobilne i senzorske (npr. kamere).

Uz OMNeT++ i INET alate omogućena je analiza stope izgubljenih paketa dviju različitih mreža i nekoliko događaja. Analiza diplomskog rada naglasila je ključnu ulogu stope izgubljenih paketa u performansama žičanih komunikacijskih mreža. Previsoka stopa izgubljenih paketa može značajno utjecati na kvalitetu same usluge, te može uzrokovati

degradaciju kvalitete sadržaja i poteškoće u isporuci podataka preko mreže. Upravljanje i praćenje stope izgubljenih paketa neophodno je kako bi se osigurao stabilan i optimalan prijenos podataka.

Literatura

[1] Kumar S, Dalal S, Dixit V., “The OSI model: Overview on the seven layers of computer networks”, International Journal of Computer Science and Information Technology Research, vol. 2, no. 3, 2014, str. 461-466.

[2] Nath, P. B.; Uddin, M. M., Tcp-ip model in data communication and networking. American Journal of Engineering Research, 2015, 4.10: 102-107.

[3] Wendell, Odom, CCNA 200-301 Official Cert Guide Library, Cisco Press, 2019.

[4] Ali, Amer Nizar Abu. Comparison study between IPV4 & IPV6, International Journal of Computer Science Issues (IJCSI), 2012, 9.3: 314.

[5] Cotton, Michelle; Vegoda, Leo, Special Use IPv4 Addresses, Internet Engineering Task Force (IETF), 2010.

[6] Barnes, David; Sakandar, Basir. Cisco LAN Switching Fundamentals. Cisco Press, 2004.

[7] RFC 2328 – OSPF Version 2, April 1998.

[8] Albrightson, R., Garcia-Luna-Aceves, J. i Boyle, J. "EIGRP--A Fast Routing Protocol based on Distance Vectors", UC Santa Cruz, 1994.

[9] RFC 7868 – Cisco’s Enhanced Interior Gateway Routing Protocol (EIGRP), May 2016.

[10] RFC 2131 – Dynamic Host Configuration Protocol, March 1997.

[11] RFC 793 – Transmission Control Protocol, September 1981.

[12] RFC 768 – User Datagram Protocol, August 1980.

[13] RFC 3550 – RTP: A Transport Protocol for Real-Time Applications, July 2003.

[14] RFC 2326 – Real Time Streaming Protocol, April 1998.

[15] Held, Gilbert. Working with cisco access lists. International Journal of Network Management, 1999, 9.3: 151-154.

[16] Bienik, Juraj, M. Uhrina, L. Sevcik i A. Holesova. "Impact of Packet Loss Rate on Quality of Compressed High Resolution Videos" Sensors 23, no. 5: 2744, 2023, doi: 10.3390/s23052744.

[17] Kim, Hyun Jong; Choi, Seong Gon. A study on a QoS/QoE correlation model for QoE evaluation on IPTV service. In: 2010 The 12th International Conference on Advanced Communication Technology (ICACT). IEEE, 2010. p. 1377-1382.

[18] H. E. Egilmez, S. T. Dane, K. T. Bagci and A. M. Tekalp, "OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks," Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit i Conference, Hollywood, CA, USA, 2012, str. 1-8.

[19] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: enabling innovation in campus networks,” SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, 2008, str. 69–74.

[20] J. Nightingale, Q. Wang, C. Grecos i S. Goma, "The impact of network impairment on quality of experience (QoE) in H.265/HEVC video streaming," in IEEE Transactions on Consumer Electronics, vol. 60, 2023.

[21] https://omnetpp.org/, dostupno 01.09.2023.

[22] https://inet.omnetpp.org/, dostupno 01.09.2023.

[23] https://github.com/, dostupno 01.09.2023.

Popis slika

Popis tablica