Računarske mreže

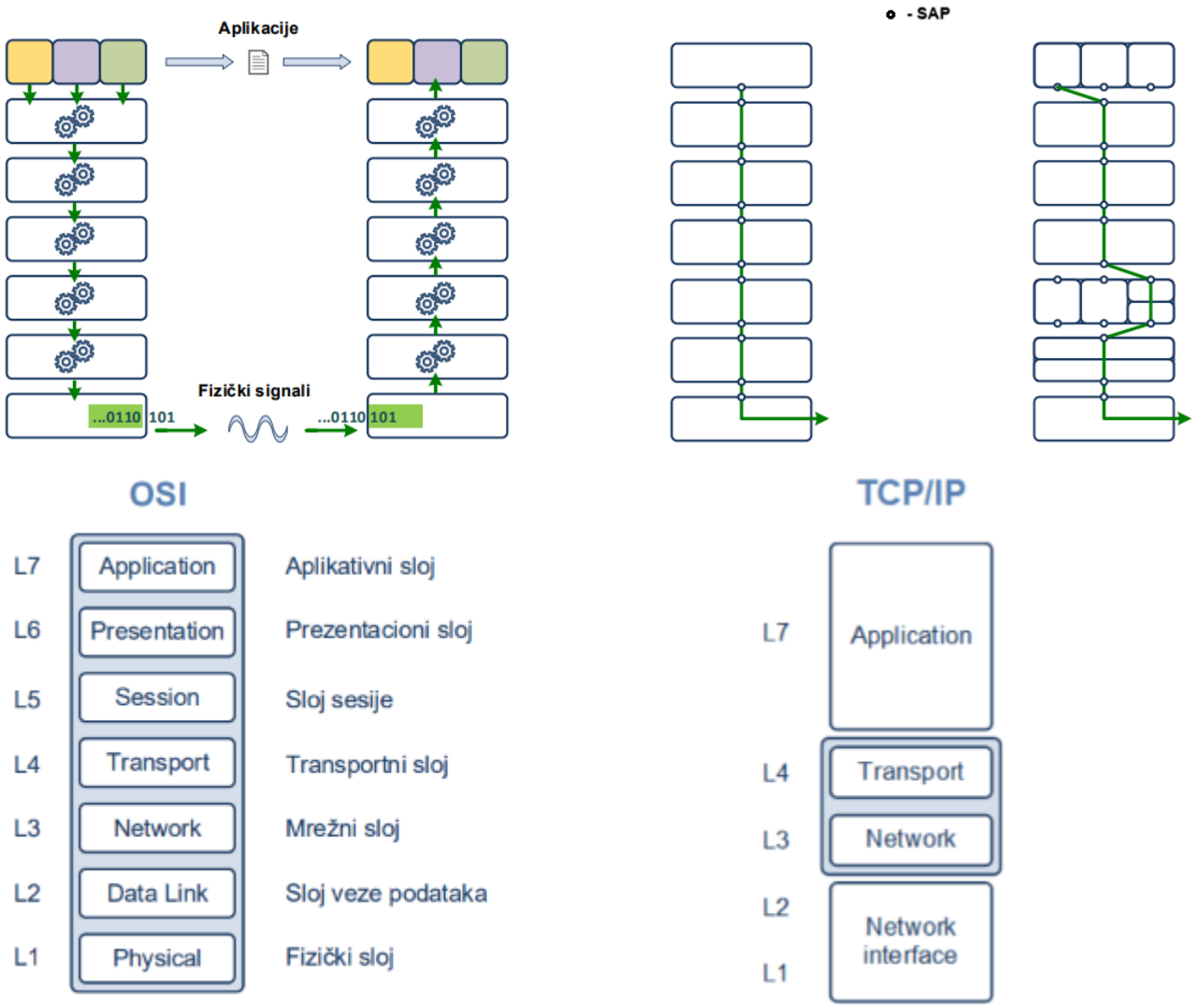
Uvod u računarske mreže

U starim računarskim centrima postojao je veliki **centralni računar** (**mainframe**) koji je bio povezan sa većim brojem terminala. Komunikacija je održavana modemskim vezama po običnim telefonskim kablovima. Bio je moguć prenos samo tekstualnog ulaza i izlaza, a brzina je bila veoma mala. Korišćeno je **svičovanje veze** (**circuit switching**) koje podrazumeva da se unapred rezerviše veza "s-kraja-na-kraj". Veza je mogla da se rezerviše trajno ili po potrebi. Ovakav pristup je neekonimičan jer su veze često zauzete i kada se ne koriste pa lako dolazi do zagušenja mreže. Još jedan problem predstavlja i neskalabilnost. U suštini, ovo nisu računarske mreže jer one podrazumevaju razmenu podataka između računara. Osnovne komponente računarskih mreža su:

- **komunikacioni uređaji** kao što su ruteri, svičevi, habovi, modemi, ripiteri, firewall
- **komunikacione veze** između tih uređaja koje se uspostavljaju putem različitih vrsta medijuma, različite brzine i sa različitim osobinama
- **funkcionalna logika** koja podrazumeva protokole, servise i konfiguracije mreže i predstavlja najbitniji deo računarskih mreža

Matematička osnova mreža postavljena je 1961. godine - **teorija redova čekanja** (**queueing theory**). Kasnije se uvodi **svičovanje paketa** (**packet switching**) koje podrazumeva podelu podataka na manje pakete koji se nezavisno prenose preko mreže, bez prethodno uspostavljene veze s-kraja-na-kraj. Dobijena je veća iskorišćenost i veća fleksibilnost i omogućeno deljenje komunikacionih resursa. Prva računarska mreža sa svičovanjem paketa je **ARPANET** napravljena 1969. godine kao projekat američkog ministarstva odbrane, a služila je za povezivanje univerziteta i istraživačkih organizacija u SAD. U okviru nje korišćen je prvi mrežni protokol - **NCP** (**Network Control Protocol**). Kasnije se razvijaju mreže i u drugim državama, kao što je Cyclades u Francuskoj.

International Network Working Group (INWG) zalagala se za svičovanje paketa i standardizaciju što je u to vreme bila radikalna ideja, suprotna interesima računarskih giganata. Vint Cerf i Robert Kahn objavljuju naučni rad u kome izlažu neke osnovne principe kao što su minimalizam, decentralizovana kontrola, best-effort koji podrazumeva da šaljemo paket i nadamo se najboljem, bez garancije da će zapravo stići. IBM uspostavlja svoju mrežu SNA, a prate ga i ostali proizvođači sa svojim mrežama. INWG podnosi tehnički predlog protokola međunarodnoj organizaciji za standarde u telekomunikacijama, ali on biva odbijen. Drugi zahtev podnose **Open System Interconnection** (**OSI**) unutar ISO 1977. godine sa ciljem da se donese međunarodni standard za računarske komunikacije. Osnovni principi su **otvorenost** koja podrazumeva komunikaciju nezavisnu od proizvođača uređaja i **modularnost** koja podrazumeva podelu kompleksnog problema u manje celine. Na taj način nastaje **OSI referentni model**. Na najvišem (logičkom) nivou odvija se prenos poruka između aplikacija, a na najnižem (fizičkom nivou) prenos bitova kao elektromagnetni signali. Između ova dva nivoa treba obezbediti da poruke nađu put do odredišnog uređaja i prepoznaju odredišnu aplikaciju na tom uređaju. Posao se deli na više **slojeva**, gde svaki sloj obavlja svoj deo posla i na usaglašeni način komunicira sa višim i nižim slojevima preko **interfejsa za komunikaciju** (**Service Access Point**). Jedan sloj može da ima više različitih implementacija (protokola) kao i više podslojeva. OSI model se sastoji od sedam slojeva. Komunikacija se vrši **porukama**. Svaka poruka sadrži **zaglavlje** (**header**) koje sadrži kontrolne podatke i **telo** (**body**) koje sadrži podatke koji se prenose.

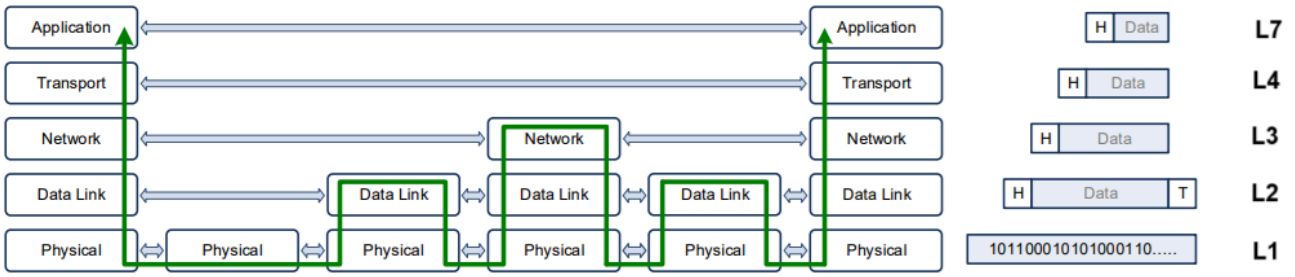


ARPANET je uporedo razvijao svoj protokol i 1980. objavljuje **TCP/IP** (**Transmission Control Protocol/Internet Protocol**) **model**. Ovde je fokus bio na unutrašnjim nivoima, tj. TCP na L4 nivou i IP na L3 nivou, dok su nivoi ispod i iznad spojeni u jedinstvene slojeve. Ministarstvo odbrane 1. januara 1983. donosi zvaničnu odluku o korišćenju TCP/IP protokola u okviru ARPANET mreže i ovaj datum smatra se rođenjem Interneta. U narednih par godina vlada ipak nalaže svojim ustanovama prelazak na OSI. Ispostaviće se da taj prelazak i nije toliko jednostavan, jer OSI iz različitih razloga nije do kraja završen. **Tim Berners-Lee** 1991. godine definiše **WorldWideWeb** koji omogućava jednostavnu razmenu tekstualnih poruka i slika, a veoma brzo se dozvoljava i komercijalni saobraćaj na Internetu. Standardizacija Internet protokola izvršena je putem **RFC (Request for Comments) dokumenata** koji su sadržali razne preporuke, tutorijale i slično.

U OSI protokolu **L7 aplikativni sloj** zadužen je za razmenu podataka između aplikacija. **L6 prezentacioni sloj** zadužen je za predstavljanje i konverziju podataka, odnosno bavi se formatiranjem, enkripcijom, kompresijom i slično. **L5 sloj sesije** zadužen je za uspostavljanje, kontrolu i raskidanje sesije između aplikacija. U TCP/IP modelu slojevi L6 i L5 integrisani su u aplikativni sloj. Slanje poruke funkcioniše na sledeći način:

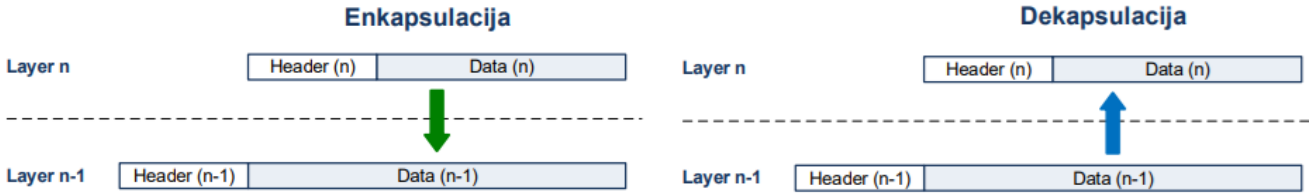
- **L7 aplikativni sloj**: aplikacije koriste mrežne usluge (npr. HTTP) za razmenu poruka i komunikacija se vrši s-kraja-na-kraj.
- **L4 transportni sloj**: prihvata poruku od aplikacije i nekim protokolom (npr. TCP) šalje poruku transportnom sloju druge aplikacije. Ovaj sloj ne zanima sadržaj poruke, već je šalje kao sirove podatke. Kažemo da se na ovom nivou razmenjuju **poruke** (**message**).
- **L3 mrežni sloj**: poruku od transportnog sloja zapakuje i šalje na mrežu koristeći određeni protokol (npr. IP). Paketi prolaze kroz mrežne čvorove i nalaze put do mrežnog sloja druge aplikacije. Kažemo da se na ovom nivou razmenjuju **paketi** (**datagram**). Razmenjuju se preko **rutera** s-kraja-na-kraj. Adrese u L3 zaglavlju se ne menjaju.

- **L2 sloj veze podataka**: radi u lokalu, na bliskim rastojanjima, postavlja svoje zaglavlje na poruku koju je dobio od mrežnog sloja, ali postavlja i **trailer** na kraj poruke koji se koristi za detekciju grešaka. Kažemo da se na ovom nivou razmenjuju **okviri** (**frame**). Razmenjuju se preko **svičeva** od-rutera-do-rutera. Na L2 segmentima između rutera adrese u L2 zaglavlju se ne menjaju, a menjaju se na prelasku između L2 segmenata.
- **L1 fizički sloj**: nule i jedinice pretvara u signale, definiše vrste kablova za prenos i slično.

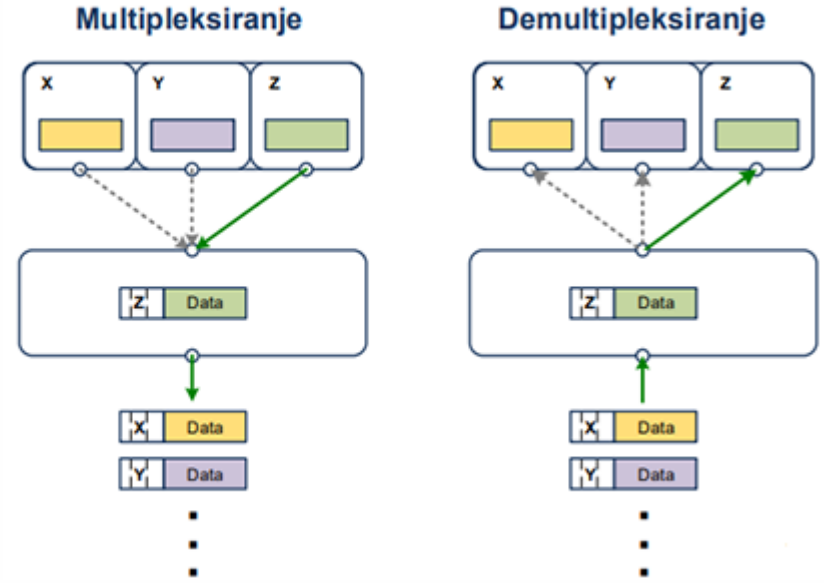


Komunikacija se vrši po vertikali nadole i nagore, kao i po horizontali kroz mrežu (L1, L2, a ponekad i L3 nivo).

Enkapsulacija vrši se pri prelasku sa višeg sloja na niži, pri čemu se dodaje zaglavlje na početku, a ponekad i potpis na kraju. Poruka višeg sloja se prenosi kao podatak u poruci nižeg sloja i ne gleda se njena struktura. **Dekapsulacija** vrši se pri prelasku sa nižeg sloja na viši sloj, pri čemu se odbacuje zaglavlje nižeg sloja i izdvajaju podaci višeg sloja.



Multipleksiranje vrši se pri prelasku sa višeg na niži sloj u slučaju kada se poruke različitih protokola višeg sloja obeležavaju u zaglavlju poruke nižeg sloja i na isti način prenose u niži sloj. **Demultipleksiranje** vrši se pri prelasku sa nižeg na viši sloj pri čemu se vrši prepoznavanje prokola višeg sloja na osnovu podataka iz zaglavlja.



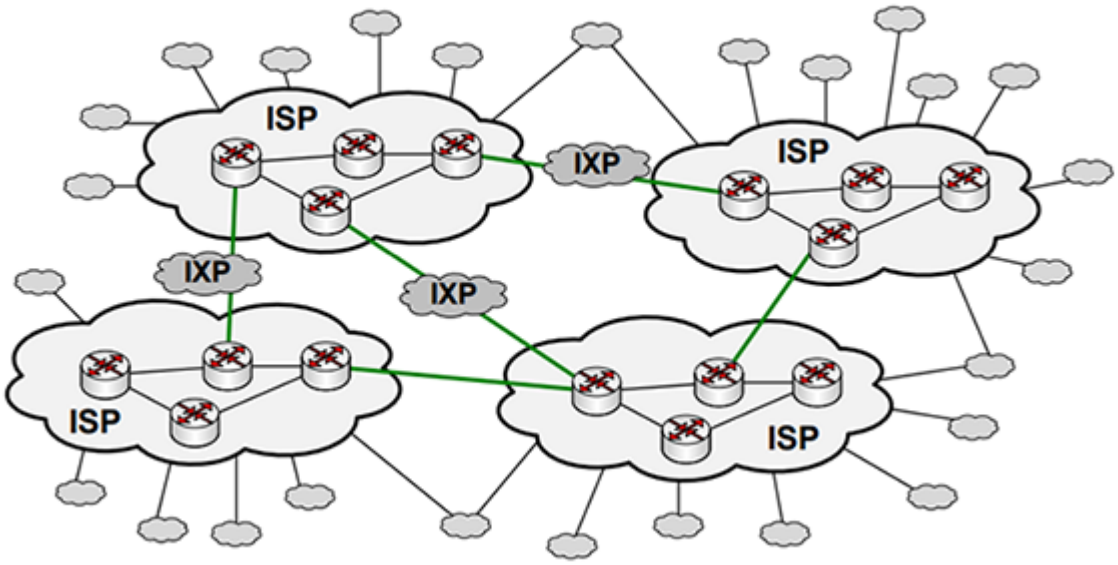
Klasična podela računarskih mreža:

- **Local Area Network - LAN**: uglavnom privatne instalacije unutar poslovnih mreža. Karakterišu ih velike brzine (100 Mbps - 10 Gbps). Koriste se L2 uređaji kao što su L2 svičevi. Moguće je povećati rastojanje i na preko 100km primenom optičkih kablova.
- **Wide Area Network - WAN**: instalacije telekomunikacionih provajdera na nivou gradova, regiona, država, kontinenata i slično, gde korisnici iznajmljuju telekomunikacione servise. Karakterišu ih velike brzine za provajdera, ali manje za korisnike. Pretežno se koriste L3 uređaji - ruteri i L3 svičevi.

Ova podela je zastarela, a umesto nje često se koristi **tehnološka podela mreža**:

- **L2 protokoli**: Ethernet, Wireless, MPLS. Koriste se L2 uređaji.
- **L3 protokoli**: IP, IPv6. Koriste se L3 uređaji.

Internet povezuje različite mreže u jednu jedinstvenu. Generalno ne pripada nikome, ali postoje **Internet Service Provider** (**ISP**) koji poseduju i kontrolišu delove interneta koje iznajmljuju korisnicima. Međusobno su povezani i komuniciraju putem **Internet Exchange Point** (**IXP**).



Jedan od najbitnijih parametara mreža je **protok podataka** (**throughput**, **speed**) koji predstavlja količinu podataka koja se može preneti u jedinici vremena. Koristi se i **kapacitet veze**, tj. **maksimalni protok** (**bandwidth**, **capacity**) koji predstavlja realni maksimalni protok u zavisnosti od fizičke veze. Jedinica mere je **bps - bit u sekundi**. Multiplikatori su dekadni: kbps (10^3), Mbps (10^6), Gbps (10^9), Tbps (10^{12}). Drugi bitan parametar je **kašnjenje paketa** (**delay**) koje predstavlja vreme prenosa celog paketa između dve tačke u mreži u jednom smeru. Jednak je zbiru sledeća četiri kašnjenja:

1. **vreme procesiranja u uređaju** d_n - odlučivanje o prosleđivanju paketa, u kom će pravcu ići i slično. Uglavnom manje od 1 ms.
2. **vreme čekanja u redu** d_q - redovi čekanja na izlaznim vezama sadrže i druge pakete pa je potrebno čekati. Iznosi $d_q = \frac{\sum L_i}{B}$, gde je L_i veličina i-tog paketa u bitima, a B kapacitet veze u bitima u sekundi.
3. **vreme sekvencijalnog izlaska paketa bit-po-bit** d_t - iznosi $d_t = \frac{L}{B}$, gde je L veličina paketa u bitima, a B kapacitet veze u bitima u sekundi.
4. **vreme propagacije** d_p - vreme potrebno da paket prođe kroz mrežu. Iznosi $d_p = \frac{D}{v}$, gde je D dužina veze, a v brzina prenosa signala u fizičkom medijumu koja najčešće iznosi oko $\frac{2}{3}c$.

Ostali parametri mreža koji se koriste su:

- **Round Trip Time** (**RTT**) - vreme prenosa paketa u oba smera, odnosno od trenutka slanja paketa do prijema odgovora.
- **Bit Error Rate** (**BER**) - u koliko prenetih bitova se javlja statistička greška. Uobičajeno reda 10^{-8} ili manje.
- **Packet loss** - gubitak paketa usled zagušenja u mreži. Izražava se u procentima.

Sloj veze podataka

L2 sloj, odnosno **sloj veze podataka** (**data-link**) ima zadatak da na neki način prenese poruke sa L3 nivoa na fizički L1 nivo. U opštem slučaju obavlja sledeće funkcije:

- **framing** - formiranje okvira

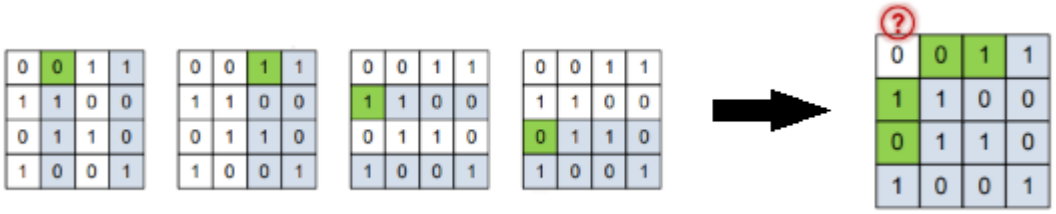
- **Media Access Control** (**MAC**) - kontrola pristup fizičkom medijumu
- **detekcija grešaka**
- korekcija grešaka - opciono
- pouzdanost (potvrda prijema) - opciono

Implementira se na nivou **mrežne kartice** (**Network Interface Card** - **NIC**) tako da imamo brzo procesiranje bez opterećenja CPU.

Algoritmi detekcije greške rade po sledećem principu: prilikom slanja poruke na kraj se dodaju **kontrolni bitovi**, a prilikom prijema se računaju kontrolni bitovi i porede sa onima koje smo prethodno primili. Neki algoritmi za detekciju greške su:

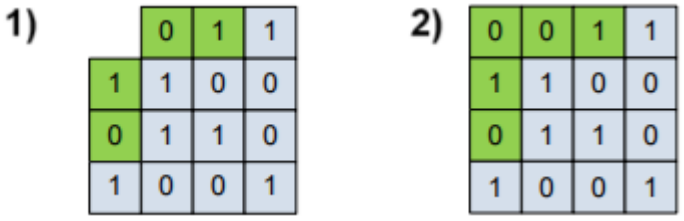
- **bitska parnost** (**parity bit**) - dodaje se jedan bit da bi se omogućilo da je broj jedinica u okviru uvek paran. Problem je što ako se promene dva bita greška neće biti detektovana.
- **Cyclic Redundancy Check** (**CRC**) - matematička operacija se primenjuje na niz bitova poruke D i dobija se novi niz bitova R koji se dodaje na poruku tako da se šalje okvir DR. Na primer, $R = ostatak([\frac{D \cdot 2^r}{G}])$, gde je G generator koji ima fiksnu vrednost, a r broj bitova u nizu R. Promenom jednog bita iz poruke D, niz bitova R se značajno menja tako da skoro da ne postoji šansa da promena dva bita dovede do anuliranja greške.

Za korekciju greške može se koristiti **dvodimenzionalna bitska parnost** gde se okvir deli na manje celine koje se posmatraju kao matrice određene dimenzije. Po svim redovima i kolonama vrši se bitska parnost, tj. imamo jednu dodatnu kolonu i jedan dodatni red kontrolnih bitova. Za blok od $n \times m$ bitova imamo $n + m - 1$ kontrolnih bitova što i nije baš efikasno. Ponovo može doći do nemogućnosti detekcije mesta greške ako se greška javila na bitovima u istom redu, istoj koloni, bitovima koji dele dijagonalu i slično. Drugi način osmislio je Richard Hamming tražeći način da izvrši detekciju i korekciju greške na bušenim karticama. Posmatrao je blok od 16 bita i proveravao parnosti u podgrupama od 8 bita.



Problem je u prvom bitu koji ostaje bez provere jer se ne sadrži ni u jednoj od podgrupa. Ovaj problem se može rešiti na dva načina:

1. Prvi bit se potpuno izostavlja tako da se koriste blokovi od 15 bita, od kojih 11 sadrži informacije, a 4 su kontrolna bita. Ovo se naziva **Hamingov kod (15, 11)**.
2. Prvi bit se koristi za proveru parnosti celog bloka što omogućava dodatnu detekciju ako imamo dve greške. Koriste se blokovi od 16 bita, od kojih 11 sadrži informacije, a 5 su kontrolna bita. Ovo se naziva **Hamingov kod (16, 11)**.



Hamingov kod se koristi i za detekciju i korekciju grešaka na L2 nivou, ali se koriste veći blokovi koji se dele u podgrupe na istom principu kao i blokovi od 16 bita.

Jedna od najbitnijih karakteristika mreža je **topologija**, odnosno na koji način je mreža povezana. Koriste se:

- **direktna veza** (**point-to-point**) - za dva direktno povezana učesnika.
- **prstenasta topologija** (**ring**) - paketi kruže u jednom ili oba smera prstena na koji je povezano više učesnika sa ravnopravnim pristupom medijumu.

- **bas topologija** (**bus**) - ravnopravni deljeni medijum gde se učesnici "bore" za pristup medijumu, što može dovesti do nastanka kolizija.
- **zvezdasta topologija** - centralni uređaj za međusobnu komunikaciju među korisnicima.
- **bežična topologija** (**wireless**) - logički može biti organizovana kao bas ili zvezda.

Kontrola pristup medijumu (**MAC**) je deo L2 sloja koji se oslanja na fizički (L1) nivo i treba da realizuje prenos bita kroz elektromagnetne signale. Medijum može biti **deljeni** (npr. bas) gde svi učesnici "vide" sve pakete, što nazivamo **broadcast**. Svaki uređaj ima svoju adresu, tako da paket prima samo onaj uređaj koji prepozna svoju adresu u tom paketu. Moguće je da dođe do kolizija kada više korisnika istovremeno šalje pakete. Protokoli pristupa deljenom medijumu definišu na koji način možemo izbeći ili rešiti ove kolizije, odnosno kako M učesnika na idealan način treba da deli medijum kapaciteta od B bps. Najvažniji zahtevi su da rešenje bude jednostavno i decentralizovano, odnosno da nema centralnog čvora koji bi kontrolisao redosled slanja, kao ni sinhronizacije. Takođe bi bilo idealno da kada samo jedan korisnik šalje pakete on na raspolaganju ima svih B bps, a kada pakete šalje M korisnika da svako ima po deo, odnosno $\frac{B}{M}$ bps. Moguća rešenja su:

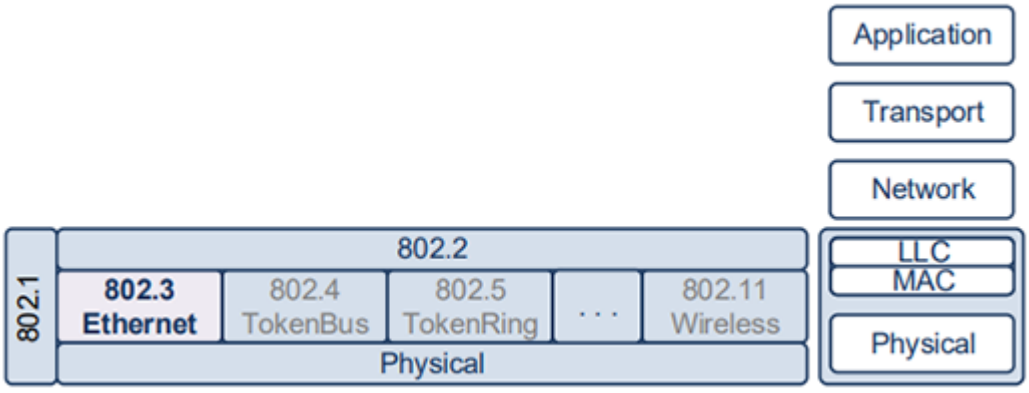
- **podela medijuma** (**channel partitioning**) - medijum (kanal) se deli na manje delove, bilo vremenski, frekvencijski ili prema nekim kodovima. Svaki učesnik ekskluzivno dobija svoj deo kanala. **Vremenska podela** (**Time Division Multiple Access** - **TDMA**) podrazumeva da svaki učesnik dobija svoj fiksni vremenski okvir (slot) za slanje paketa. Redosled slanja je unapred utvrđen. Ako nema paketa za slanje, slot i ceo kanal ostaje neiskorišćen. **Frekvencijska podela** (**Frequency Division Multiple Access** - **FDMA**) podrazumeva da svaki učesnik dobija svoju fiksnu frekvenciju za slanje. Učesnici pakete mogu da šalju istovremeno na ovaj način, ali opet nedodeljene frekvencije ostaju neiskorišćene. Ovaj pristup je efikasan za velika opterećenja.
- **pristup sa dodelom dozvole** (**taking turns**) - svaki učesnik čeka da mu se po nekom algoritmu dodeli dozvola za slanje paketa. Jedan algoritam dodele je **prozivanje** (**polling**) gde postoji **glavni učesnik** (**master**), dok su ostali **sporedni** (**slave**). Master proziva slave učesnike i dodeljuje im dozvolu za slanje. Drugi algoritam je **kruženje žetona** (**token passing**) gde po nekom prstenu kruži kontrolni paket, odnosno token. Svaki učesnik prihvata token kada mu se omogućava slanje paketa, a zatim ga prosleđuje dalje čime se određuje redosled slanja. Najpoznatija implementacija je IBM-ov TokenRing. Ovaj pristup je efikasan za različita opterećenja, ali je složen za implementaciju, ograničene je brzine i podržava samo prstenaste topologije (bilo fizičke ili logičke).
- **slučajan pristup** (**random access**) - svaki učesnik može da šalje pakete i može da koristi ceo propusni opseg. Ovaj pristup je efikazan za mala opterećenja. Najveći problem su kolizije koje je potrebno izbeći ili bar smanjiti verovatnoću njihovog nastanka. Ukoliko ipak dođe do kolizija potrebno ih je detektovati i oporaviti se od kolizije. Postoje različite vrste slučajnog pristupa:
  1. **Pure ALOHA** - svaki učesnik šalje okvir u bilo kom trenutku. Ako više učesnika šalje u isto vreme dolazi do kolizije. Tada je potrebno da ponovo pošalju pakete uz neko slučajno vreme čekanja. Nastoji se da se kolizija izbegne, ali se to ne garantuje. Iako je manje efikasan od Slotted ALOHA imao je primenu u implementaciji mreže koja je trebala da poveže zgrade univerziteta na Havajima tako da imamo jedan centralni računar koji šalje poruke ka perifernim putem jedne frekvencije u kom slučaju nema kolizija. Od perifernih ka centralnom se koristi druga frekvencija i tu su moguće kolizije ako više perifernih šalje u isto vreme. Ovim zahtevima je upravo odgovarala Pure ALOHA i nastaje ALOHAnet koja je bila prva bežična mreža povezana na ARPAnet mrežu.
  2. **Slotted ALOHA** - paketi se šalju samo u određenim intervalima (slotovima). Učesnici moraju da budu sinhronizovani, a ograničena je i veličina paketa. Ako više učesnika šalje u istom slotu dolazi do kolizije. Učesnik u koliziji ponovo šalje isti paket u nekom narednom slotu sa određenom verovatnoćom, tj. ne znači da će ga poslati u prvom narednom slotu. Prednost je što jedan učesnik može da koristi ceo propusni opseg, jednostavna implementacija i decentralizovanost. Nedostaci su trošenje slotova u slučaju kolizije, neiskorišćeni slotovi nakon kolizije, kao i sinhronizacija učesnika. Ima dva puta veću maksimalnu efikasnost od Pure ALOHA.

3. **CSMA** (**Carrier Sense Multiple Access**) - omogućava višestruki pristup deljenom medijumu. Pravila su jednostavna: svi uređaji "slušaju" aktivnost na medijumu. Ako je medijum zauzet čeka se sa slanjem paketa. Ako je medijum slobodan paket se šalje. Zbog ograničene brzine propagacije signala moguće je da kolizije nastanu u bliskim vremenskim trenucima. Zbog toga se koristi **CSMA/CD** (**CSMA/Collision Detection**), tj. višestruki pristup deljenom medijumu sa detekcijom kolizije. Jednostavno se implementira na žičanom medijumu jer tokom slanja uređaj istovremeno i prima signale i poredi da li su istovetni. Na bežičnom je komplikovano jer je jačina signala pri slanju daleko veća od jačine signala pri prijemu.

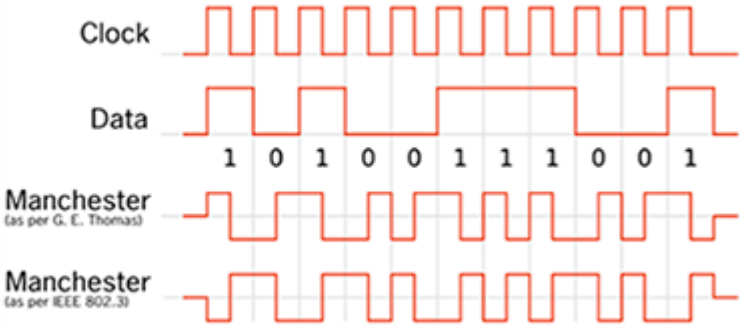
Bob Metcalfe početkom 70ih posećuje Havaje gde se upoznaje sa implementacijom ALOHAnet mreže. Poznavajući ARPAnet i ALOHAnet on primenjuje CSMA/CD principe na bakarni koaksijalni kabl što dovodi do nastanka **Etherneta** 1980. godine. Početna brzina je bila oko 2.94 Mbps, a danas iznosi i preko 1Gbps. Vremenom se Ethernet standardizovao i postao najkorišćeniji za LAN mreže.

Ethernet

Elementi L1 fizičkog sloja su prenosni medijum, konektori, elektromagnetne osobine signala, kodovanje i modulacija i slično. L2 sloj veze podataka zadužen je da poruke sa L3 nivoa prenese na fizički nivo. Sastoji se od dva podsloja: **Logical Link Control** (**LLC**) koji je zadužen za multipleksiranje i enkapsulaciju protokola višeg sloja, tj. sloja L3 i **Media Access Control** (**MAC**) koji je zadužen za pristup medijumu, adresiranje i kontrolu greške. **IEEE 802.1** predstavlja set različitih standarda slojeva L1 i L2. **IEEE 802.2** predstavlja standard za LLC, odnosno komunikaciju sa L3 mrežnim slojem. Postoje različiti protokoli za MAC podsloj kao što su **IEEE 802.5** (**TokenRing**), **IEEE 802.11** (**Wireless**) i **IEEE802.3** (**Ethernet**).

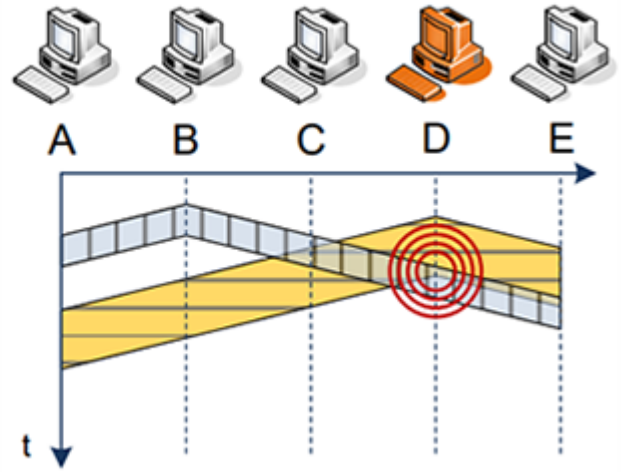


Ethernet I nastaje 1980. godine, a ubrzo 1982. nastaje i **Ethernet II**. Standardizovan je 1983. godine kao **IEEE 802.3** standard. Korišćen je koaksijalni kabl širokog prečnika koji je bio težak za instalaciju, pa je Ethernet bio nazivan **debeli** (**thick**) **Ethernet** ili **10BASE5**. Usled slabljenja signala uvodi se maksimalna dužina kabla do 500m, a korišćeni su i kablovi do uređaja (Media Attached Unit) do 50m. Tokom 1985. godine nastaje **tanki** (**thin**) **Ethernet** (**10BASE2**, **IEEE 802.3a**) koji koristi kabl manjeg promera. On je bio jeftiniji i jednostavniji za instalaciju, a maksimalna dužina segmenta bila je 185m. Na jednom segmentu bilo je povezano do 30 uređaja, putem BNC T konektora. Za kodovanje, tj. predstavljanje niza bitova odgovarajućim stanjem elektromagnetnog signala, koristio se **Mančester kod** gde 0 predstavlja promenu napona sa visokog na niski nivo, a 1 sa niskog na visoki.

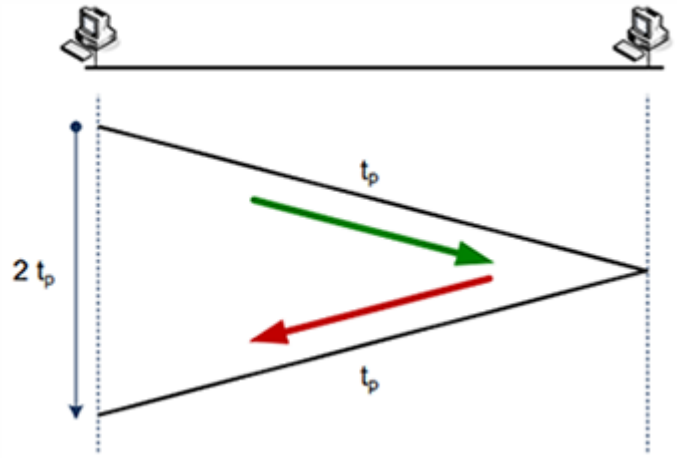


Ripiteri služe za povezivanje segmenata na L1 nivou. Funkcionišu tako što prime signale sa jednog segmenta,

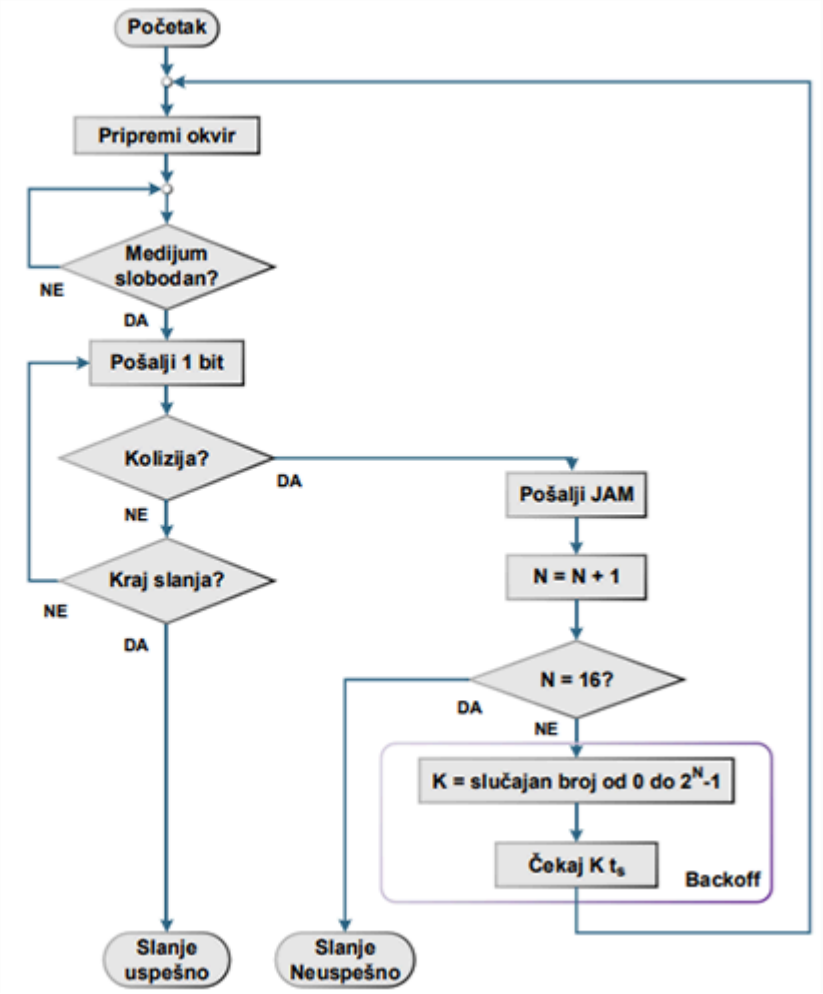
zatim prepoznaju bitove (nule i jedinice) i prosleđuju nove, regenerisane signale na sledeći segment. Na ovaj način se povećava maksimalno rastojanje mreže. Međutim, koristi sa CSMA/CD protokol pa moguće kolizije unose ograničenja. Kolizija nastaje kada se dva ili više signala "sudare" na medijumu. Signali tada međusobno interferiraju i čitavi okviri bivaju uništeni. Uređaji koji primaju okvir moraju da detektuju koliziju da bi zaključili da je okvir nevalidan i da bi ga odbacili. Takođe, uređaj koji šalje okvir mora da detektuje koliziju da bi znao da je okvir uništen i da ga treba ponovo poslati. Tokom slanja okvira, sa medijuma sa istovremeno i primaju okviri. Ako je prijemni okvir isti kao i poslati smatra se da nema kolizije. Uređaj može detektovati koliziju svog okvira samo dok se on još uvek šalje na mrežu. Na primer, uređaj D će u ovoj situaciji detektovati koliziju svog okvira, ali uređaj B neće detektovati koliziju svog okvira:



Ostali uređaji mogu da detektuju grešku putem CRC kontrole, ali i ne moraju. Zbog toga, kada dođe do kolizije, svaki uređaj čiji su okviri u koliziji emituje poseban signal - **JAM okvir**. Sastavljen je od 32 bita naizmeničnih nula i jedinica i namena mu je da izazove grešku u naponskim signalima i CRC kontroli, čak i kada su u pitanju male kolizije. Ostali uređaji detektuju koliziju preko JAM signala i odbacuju primljene okvire. Kao što je već pomenuto, uslov za detekciju kolizije je da okvir i dalje izlazi na mrežu. Najnepovoljniji slučaj je da kolizija nastaje na najudaljenijem uređaju u mreži, pa okvir prvo treba da dođe do njega, a zatim da se JAM signal vrati nazad za isto to vreme. Paket u tom slučaju mora čitavo vreme da izlazi na mrežu da bi se detektovala kolizija, tj. on mora da bude dovoljno velik. Iz tog razloga se uvode dva ograničenja - minimalna veličina paketa i maksimalno rastojanje u mreži.



Okvir izlazi na mrežu sekvencijalno, "bit-po-bit". **Bitsko vreme** (**bit-time**) je vreme izlaska jednog bita na mrežu. Ono iznosi $t_b = \frac{1}{B}$, tj. zavisi od protoka. Vreme izlaska okvira na mrežu zavisi i od protoka, ali i od veličine okvira i iznosi t_{out} = Lt_b = $\frac{L}{B}$. **Slot-time** je vreme potrebno da se okvir minimalne veličine L_{min} u celini pošalje na mrežu i iznosi t_s = $L_{min}t_b$ = $\frac{L_{min}}{B}$. Uslov za detekciju kolizije sada predstavlja $t_s > t_{d(max)}$, gde je $t_{d(max)}$ maksimalno kašnjenje u oba smera. Za Ethernet protok od 10Mbps dobija se da je L_{min} = 512b = 64B. Kada smo ograničili minimalnu veličinu paketa i maksimalno rastojanje u mreži, dolazimo da algoritma za slanje okvira koji se naziva **binarno eksponencijalni backoff algoritam**. Kada je medijum slobodan okvir se šalje sekvencijalno uz praćenje kolizija. Kada se detektuje kolizija šalje se JAM signal i ponovo se pokušava sa slanjem paketa. Maksimalan broj pokušaja je 16. Pri ponovnom slanju ne šaljemo odmah, već čekamo $K \cdot t_s$ sekundi, gde je K slučajan broj iz skupa {0, 1, 3, 7, 15, ..., 2^N - 1}, a t_s je slot-time.

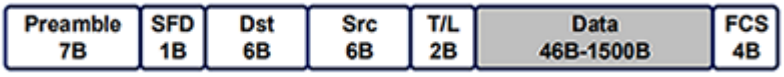


Pored kontrole greške, MAC podsloj (a samim tim i Ethernet protokol) zadužen je i za **adresiranje**. Postoje različite vrste adresa:

- **Unicat** adresa - jedinstveno adresira uređaj, upisana na mrežnu karticu uređaja.
- **Broadcast** adresa - adresira sve uređaje i sadrži sve jedinice.
- **Multicast** adresa - adresira pojedine uređaje određene namene.

Identifikacija uređaja na L2 sloju vrši se preko **MAC adresa**. One su fizički upisane u ROM na mrežnim karticama i globalno su jedinstvene, tj. svaka mrežna kartica ikad proizvedena ima različitu MAC adresu. Dužina MAC adrese je 6 bajtova. Prva 3 bajta služe za identifikaciju proizvođača (Organizational Unique Identifier), a druga 3 bajta za identifikaciju proizvedene kartice. Označavaju se heksa-dekadno, nekada četiri po četiri, a nekada dva po dva sa crticama ili dvotačkama između (npr. `3C:5F:9A:2D:B7:4E`).

Do objedinjavanja Ethernet II i IEEE 802.3 dolazi 1997. godine i nastaje **IEEE 802.3x**. Ethernet je obezbeđivao asinhrono slanje, tj. slanje u bilo kom trenutku, nezavisnost od brzine, adresiranje ko i kome šalje okvir, kontrolne podatke, kontrolu greške i same podatke, odnosno enkapsuliran paket L3 nivoa.



Ethernet okvir sastoji se od:

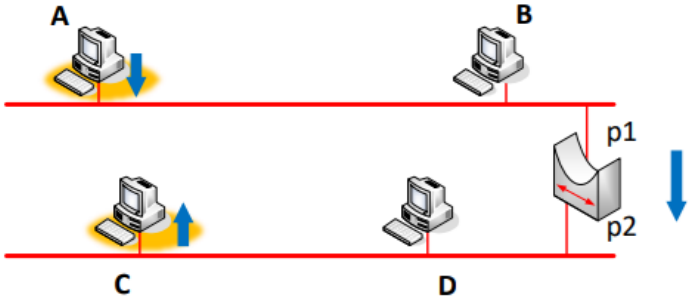
- **Preamble** - 7 bajtova naizmeničnih nula i jedinica (10101010...10) koji služe za sinhronizaciju. Potrebno je da se vreme prepoznavanja bita poklopi sa trenucima nailaska bita okvira.
- **Start of Frame Delimiter** (**SFD**) - 1 bajt koji ima vrednost 10101011. Poslednje dve jedinice ukazuju na početak "pravog" Ethernet okvira. Često se Preamble i SFD ne smatraju za sastavni deo Ethernet okvira, već ga samo najavljuju.
- **Destination** - MAC adresa uređaja kome je namenjen okvir. Svi uređaji primaju sve okvire i proveravaju da li se odredišna adresa poklapa sa njihovom adresom sa mrežne kartice. Ako adresa odgovara ili ako je u pitanju broadcast adresa okvir se preuzima, a inače se odbacuje.

- **Source** - MAC adresa uređaja koji je poslao okvir. Okviri se šalju nezavisno u oba smera, pa ako želimo poslati odgovor moramo znati od koga smo primili okvir.
- **Type/Length** - zaglavlje Type veličine 2B korišćeno je u okviru Ethernet II za identifikaciju protokola L3 sloja pri multipleksiranju i demultipleksiranju. Korišćene su standardizovane vrednosti za različite protokole. Zaglavlje Length korišćeno je u IEEE 802.3 standardu i predstavljalo je dužinu polja sa podacima, maksimalne dužine 1500B. Nakon objedinjavanja u IEEE 802.3x ova dva zaglavlja se spajaju u jedno. Ako je prosleđena vrednost do 1500 ona predstavlja dužinu poruke, a ako je prosleđena vrednost preko 1536 ona predstavlja tip.
- **Data** (**Payload**) - polje za podatke L3 nivoa. Veličina je bila ograničena na 1500B. Minimalna dužina zbog kolizije bila je 46B = 64B - 18B, gde je 18B dužina preostalog dela okvira bez preambule i SFD. Ako su podaci manji od 46B popunjavaju se dodatnim praznim bajtovima - **PAD**. Protokol L3 nivoa prepoznaje dužinu svojih podataka na osnovu T/L polja i odbacuje PAD podatke.
- **Frame Check Sequence** (**FCS**) - potpis na kraju okvira koji se koristi za CRC detekciju sa 4B. Bilo koja promena makar i jednog bita u okviru će promeniti FCS polje. Prilikom prijema okvira računa se CRC i upoređuje sa FCS poljem. Ako se detektuje greška paket se odbacuje, ali se ne obaveštava pošiljalac da je paket odbačen jer Ethernet ne garantuje upešnu isporuku okvira (best-effort strategija).

Između slanja okvira postoji minimalni razmak - **interframe gap**. Najčešće iznosi $96t_b$. Razlog je taj što okviri ne smeju da se spoje. Nakon slanja okvira, elektronika u mrežnoj kartici treba da pređe u stanje prijema za šta je potrebno određeno vreme. U suprotnom, mrežna kartica bi mogla da propusti početak narednog okvira.

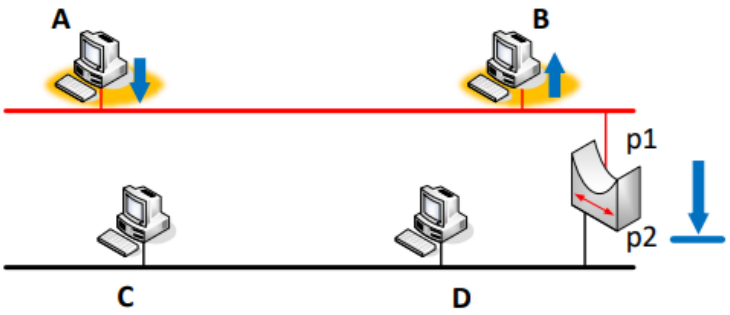
Ripiteri su nam omogućavali da produžimo veličinu mrežnog segmenta, ali sa povećanjem broja učesnika broj kolizija raste i performanse mreže značajno opadaju - problem skalabilnosti. Rešenje predstavlja **most** (**bridge**) koji LAN mrežu razdvaja na više **kolizionih domena** - skup uređaja povezanih na zajednički deljeni medijum gde svi poslati okviri dolaze do svih uređaja i sva komunikacija je podložna koliziji. **Portovi** mosta pripadaju različitim kolizionim domenima, a most blokira pakete ako je odredište na istoj strani kao i izvorište. Okviri se propuštaju samo kada je odredište na drugoj strani mosta, odnosno u drugom kolizionom domenu. Most radi po principu **transparentnog bridžinga** koji podrazumeva da hostovi ne znaju za postojanje mostova. Da bi to bilo moguće svaki bridž ima svoju **bridžing tabelu** koja sadrži MAC adrese svih hostova kao i identifikaciju porta na čijoj strani se nalazi taj host. Na osnovu bridžing tabele bridž zna koji se host nalazi na kom segmentu. Most obavlja sledeće procese:

1. **Forwarding** - prosleđivanje okvira kroz bridž kada se izvorišni i odredišni uređaji nalaze na različitim segmentima. Prima se okvir na jednom ulaznom portu, gleda se odredišna MAC adresa, traži se u bridžing tabeli i ako se nađe uslov je da nije uparena sa istim ulaznim portom što znači da se odredište nalazi na drugom segmentu.



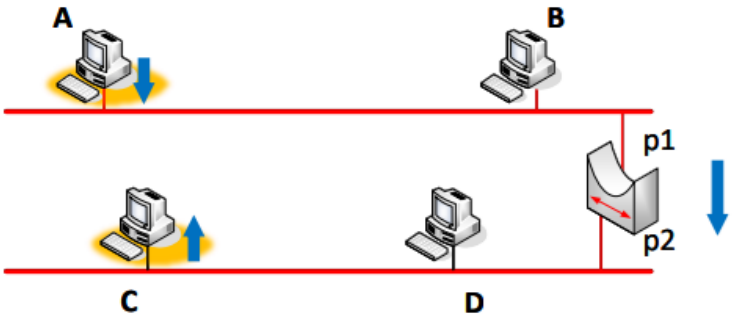
MAC adrese	Port
AAAA.0000.1111	p1
BBBB.2222.3333	p1
CCCC.4444.5555	p2
DDDD.6666.7777	p2

2. **Filtering** - blokiranje okvira kada se izvorišni i odredišni uređaji nalaze na istom segmentu, jer će se okvir primiti nezavisno od bridža.



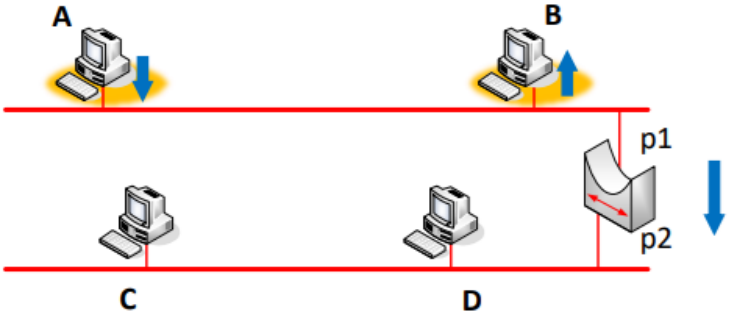
MAC adrese	Port
AAAA.0000.1111	p1
BBBB.2222.3333	p1
CCCC.4444.5555	p2
DDDD.6666.7777	p2

3. **Learning** - bridž saznaje na kom portu se nalaze MAC adrese u trenutku kada uređaji prvi put pošalju neki okvir, jer je na samom početku bridžing tabela prazna.



MAC adrese	Port
AAAA.0000.1111	p1

4. **Flooding** - propuštanje okvira kroz bridž kada se odredišna MAC adresa ne nalazi u bridžing tabeli. Bridž ne zna sa koje strane se nalazi odredište pa propušta paket iako to možda nije bilo potrebno.

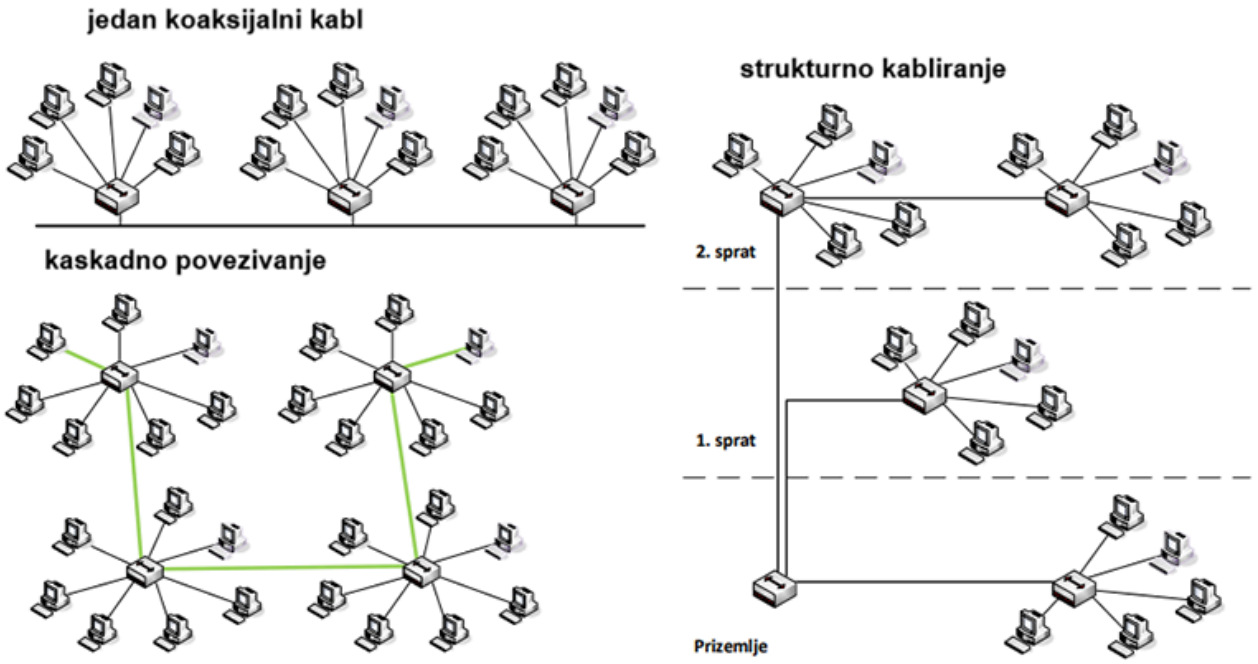


MAC adrese	Port
AAAA.0000.1111	p1

5. **Aging** - brisanje reda iz bridžing tabele ako sa određene MAC adrese nema saobraćaja neko vreme. Za svaki red postoji tajmer u tabeli koji se resetuje pri learning procesu. Ovaj proces pruža zaštitu od popunjavanja bridžing tabele, kao i zaštitu od netačnih podataka ukoliko se neki uređaj premešta sa jednog na drugi port bridža.

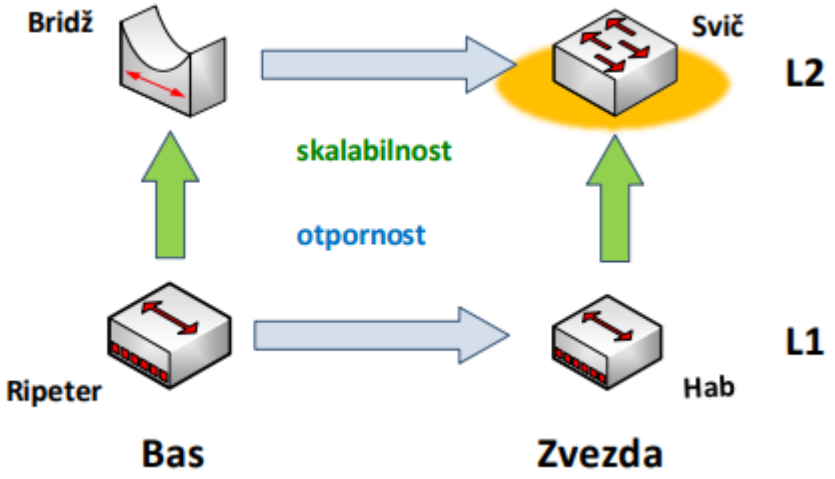
U početku su za Ethernet korišćeni koaksijalni kablovi. U slučaju greške na koaksijalnom kablu ili samo jednom konektoru dolazi do narušavanja elektro-magnetne karakteristike celog kabla, a samim tim i do prekida cele mreže. U slučaju korišćenja ripitera problem se javlja na svim segmentima koji su povezani ripiterom. Rešenje je pronađeno u zvezdastoj topologiji, odnosno korišćenju **habova** (**hub**) - višeportnih ripitera. Okvir primljen na jednom portu se regeneriše i reemituje na sve izlazne portove, a oštećenje na jednom kablu ili konektoru je u ovom slučaju lokalizovano. Početkom 90ih dva standarda definišu upotrebu habova:

1. **IEEE 802.3i** (**10BASE-T**) podrazumeva korišćenje **UTP** (**Unshielded Twisted Pairs**) **kablova** dužine do 100m. UTP kablovi sastoje se od 4 upredene neoklopljene parice, tj. 8 žica. Habovi su se u početku povezivali preko jednog koaksijalnog kabla. Zatim se koristilo **kaskadno povezivanje** gde su do 4 haba bila povezana direktno u nizu. Maksimalno rastojanje je bilo 5 · 100m. S' idejom da se iskoristi postojeća telefonska mreža uvodi se **strukturno kabliranje** gde se čvorište mreže nalazi u jednom centru (npr. prizemlje), a postoje vertikalne instalacije između spratova i horizontalne instalacije unutar spratova. Za veće brzine (100Mpbs, 1Gbps) koristili su se UTP kablovi kategorije 5e.
2. **IEEE 802.3j** (**10BASE-F**) podrazumeva korišćenje **optičkih kablova** dužine do 2km. To su kablovi sa većim brojem vlakana koji prenose optičke signale - fotone. Imaju veće brzine prenosa na većim rastojanjima i omogućavaju povezivanje odvojenih objekata. **Multimodna** (**MM**) **vlakna** su imala veća jezgra i korišćeni su laseri ili LED diode. Kablovi sa ovim vlaknima imali su veće rasipanje svetlosti i kraća rastojanja, ali su bili jeftiniji. **Singlmodna** (**SM**) **vlakna** su imala manja jezgra i korišćeni su laseri. Kablovi sa ovim vlaknima imali su manje rasipanje svetlosti i veća rastojanja, ali su bili skuplji.

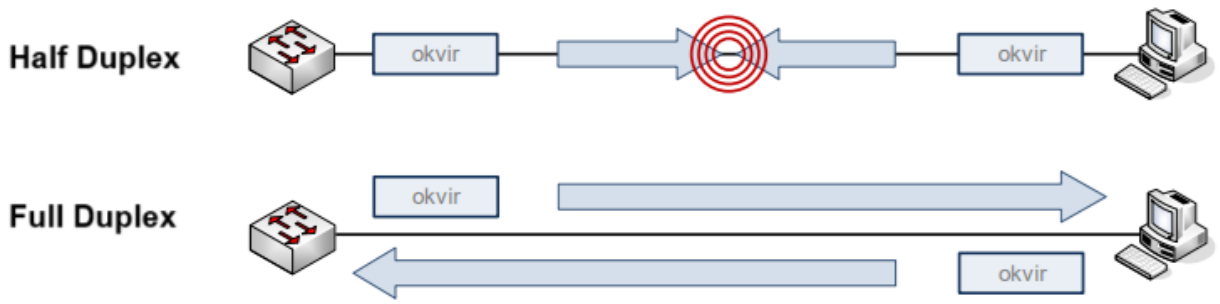


Standard **IEEE 802.3u** uvodi se 1995. godine i predstavlja **Fast Ethernet** koji dostiže brzinu od 100Mbps. Format okvira ostaje isti, a za kodovanje niza bitova koristi se **MLT-3 kod** koji podrazumeva 3 naponska nivoa (-1, 0, +1) koji ukuzuju na prisustvo ili odsustvo promene. Za kodovanje grupe bitova koristi se **4B5B kod** koji transformiše 4 bita u 5 bita. Koriste se habovi pa je vreme obrade okvira smanjeno, a smanjena je i maksimalna veličina mreže. Kao i kod 10Mbps minimalna veličina okvira je 64B. Standard **802.3ab** uvodi se 1999. godine i predstavlja **Gigabit Ethernet** koji dostiže brzinu od 1Gbps. Format okvira ostao je isti, koristi se signal sa 17 različitih naponskih nivoa i 8B10B kodovanje grupe bita koje je slično kao 4B5B. Minimalna veličina okvira je 512B.

Bridževi su rešavali problem skalabilnosti, ali ne i otpornosti, dok je kod habova priča suprotna. **Svič** možemo posmatrati kao pametan hab ili kao višeportni bridž koji postiže i skalabilnosti i otpornost.

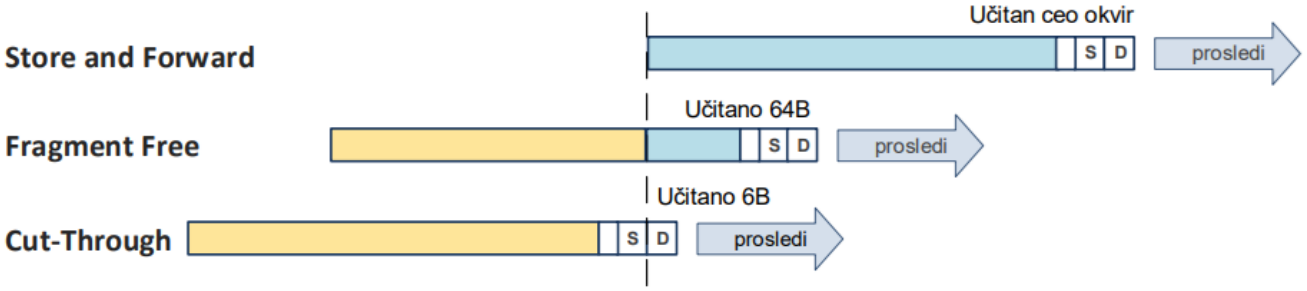


U početku su bili skupi, pa su zamenili habove samo u centralnim čvorištima LAN mreža. Svičevi su vremenom zamenili habove i cena im je opala, pa se danas habovi, bridževi i ripiteri praktično i ne koriste. Svaka veza sada postaje poseban kolizioni domen i ne postoje više ograničenja vezana za veličinu mreže i broj segmenata jer ne postoje kolizije. Termini bridž i bridžing se ponekad još uvek koriste za označavanje uređaja i način rada na L2 nivou. Svič je višeportni bridž pa funkcionalno radi isto što i on, odnosno obavlja procese forwarding, filtering, learning, flooding i aging. Svičevi omogućavaju istovremeni prenos različitih okvira između proizvoljnih parova ulaznih i izlaznih portova. **Half Duplex** podrazumeva da je segment između sviča i uređaja deljeni medijum gde su kolizije moguće. **Full Duplex** podrazumeva da UTP kablovi imaju razdvojene parice u oba smera - dve za slanje i dve za primanje. Na ovaj način je omogućen istovremeni rad u oba smera bez ikakvih kolizija, što znači da je efikasnost 200% veća u odnosu na Half Duplex. Full Duplex je podrazumevani način rada, a Half Duplex se koristi samo kada je svič povezan sa habom.

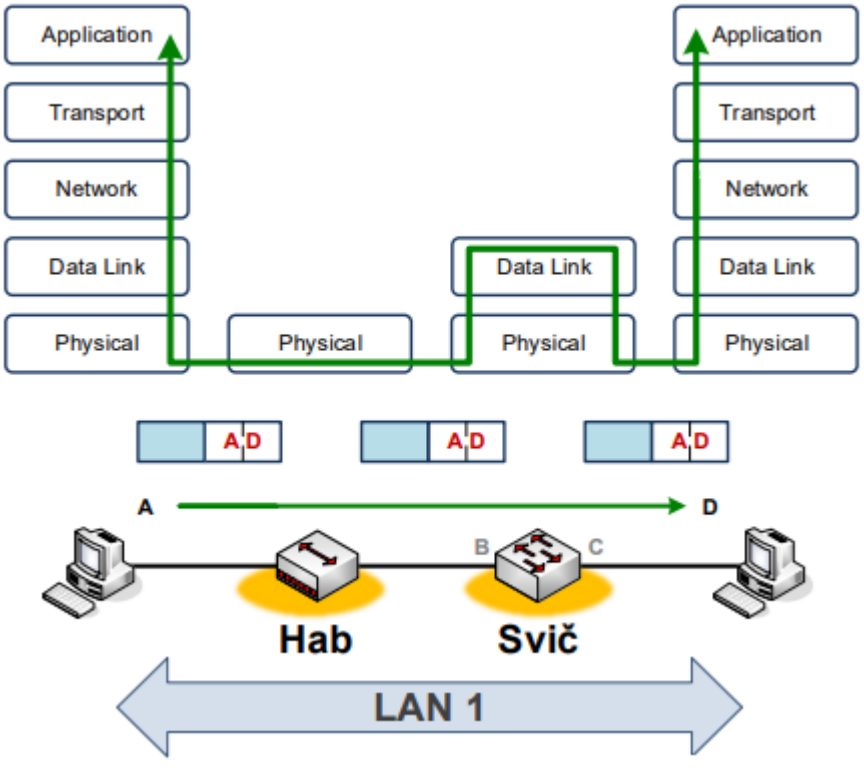


Postoje različite tehnike učitavanja i prosleđivanja okvira:

- **Store and Forward** - svič prima ceo okvir, proverava FCS i prosleđuje ga na izlazni port. Ovaj pristup ima najveće kašnjenje.
- **Fragment Free** - svič počinje prosleđivanje nakon primljenih prvih 64B. Koristi se ako nastane kolizija kada svič radi u half duplex modu.
- **Cut-Through** (**Fast Forward**) - svič počinje prosleđivanje čim primi odredišnu MAC adresu. Ovaj pristup ima najmanje kašnjenje.

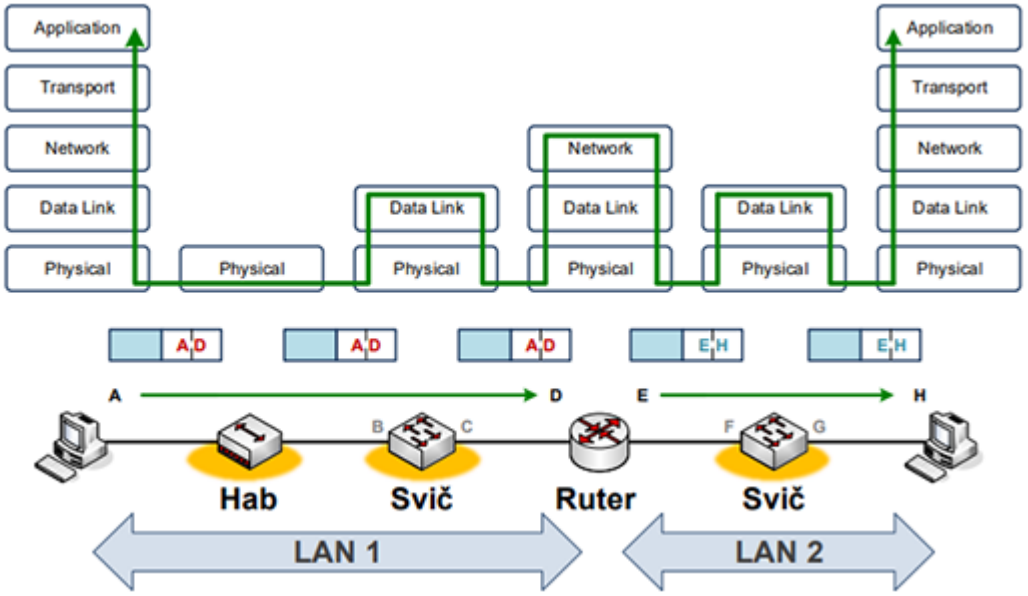


Svič ne mora imati sve portove iste brzine. Oznake za brzinu portova su 10/100/1000BASE-T. Svič i povezani uređaj automatski uspostavljaju vezu na najvećoj mogućoj brzini koju oba podržavaju što se naziva **auto-negotiation**. Prosleđivanje može biti **simetrično** kada su port sa kojeg dolazi okvir i port na koji odlazi okvir iste brzine. Ako su različitih brzina onda je u pitanju **asimetrično prosleđivanje**. Kada svič prosleđuje okvir sa porta više brzine na port niže brzine, on može koristiti Cut-Through tehniku. Ako prosleđuje okvir sa porta niže brzine na port više brzine, svič mora koristiti Store and Forward tehniku. Sviče vi su L2 uređaji - gledaju okvire na L2 nivou, ali ih ne menjaju. Portovi sviča imaju MAC adrese, ali se one ne koriste u zaglavlju okvira. To znači da je okvir nepromenljiv unutar LAN mreže.



Mrežni sloj

Ruteri su L3 uređaji koji povezuju LAN mreže. Portovi rutera imaju L2 nivo i MAC adrese koje se sada koriste u zaglavlju okvira. Ruteri okvire gledaju na L3 nivou, a adrese menjaju na L2 nivou. Komunikacija se između rutera vrši različitim L2 tehnologijama i uređajima, kao i različitim fizičkim medijumima.



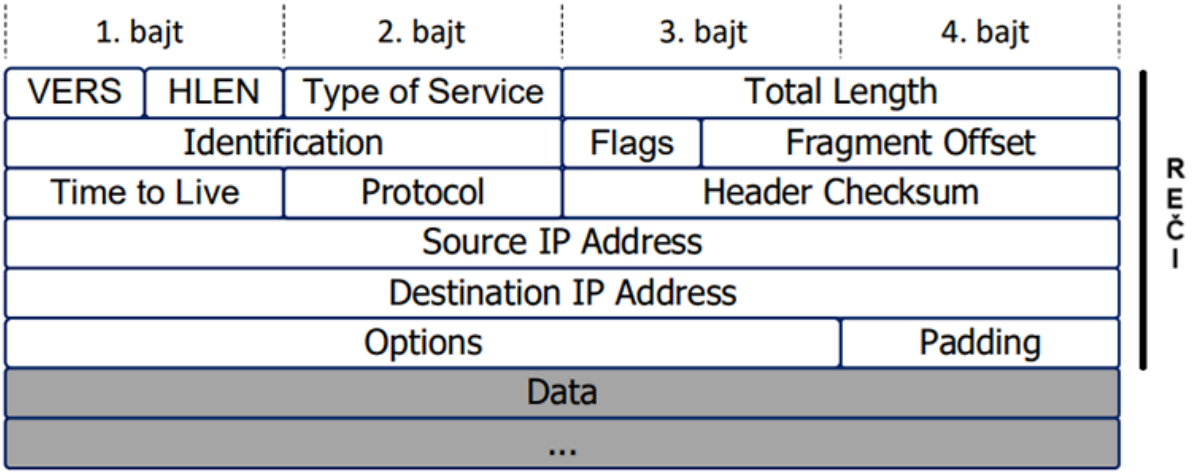
L3 sloj, odnosno **mrežni sloj** predstavlja proizvoljnu topologiju mreže povezanih rutera. **Adresiranje** je globalno i jedinstveno na celoj mreži. **Rutiranje** je prosleđivanje poruke od izvorišta do odredišta kroz mrežu rutera. Postoje različiti protokoli mrežnog sloja, ali se po pravilu koristi samo **Internet protokol** - **IP** definisan 1981. godine. Karakteristike IP protokola:

- **Connectionless** - nema uspostavljanja veze s-kraja-na-kraj. Pošiljalac ne zna da li je primalac povezan na mrežu, niti da li on uopšte postoji, da li je paket stigao do njega, da li je paket oštećen i slično. Takođe, primalac ne zna kada će paket da stigne.
- **Media Independent** - ne zavisi od fizičkog medija i protokola na prvom i drugom nivou, već se enkapsulira u pojedinačne protokole L2 sloja.
- **Best Effort** (**Unreliable**) - nema garancije da će paketi biti uspešno poslati, tj. neki paketi mogu da se izgube ili odbace od strane rutera.
- **dozvoljena proizvoljna tehnologija povezivanja** - kao posledica ovoga postoje redudantne veze i višestruke putanje što IP protokolu ne smeta. Na svakom ruteru vrši se **balansiranje saobraćaja (load balancing)** gde se različiti paketi prosleđuju različitim ruterima. Samim tim se ne garantuje redosled isporuke paketa i moguće je da paket koji je ranije poslat stigne kasnije. Dolazi i do **asimetričnog rutiranja**, tj. pojave da se biraju različite putanje u odlaznom i dolaznom smeru komunikacije između dva uređaja. Ovo nužno ne predstavlja problem, ali komplikuje stvari pri nastanku greške jer treba proveriti veći broj veza i rutera nego u slučaju **simetričnog rutiranja**.

IP paket enkapsulira poruke L4 nivoa (npr. TCP ili UDP protokol) i dodaje **IP zaglavlje** koje se sastoji od:

- **VERS** - verzija IP protokola. Osnovna verzija, koja je još uvek dominantna na Internetu, je **IPv4**, a postoji i novija verzija **IPv6**.
- **HLEN** - dužina zaglavlja označena u broju reči od 4B.
- **Total Length** - ukupna veličina IP paketa u bajtovima, uključujući zaglavlje.
- **Header Checksum** - služi za kontrolu grešaka (samo) u zaglavlju IP paketa.
- **Protocol** - služi za identifikaciju protokola L4 nivoa.

- **Type of Service** (**ToS**) - definisanje prioriteta paketa u odnosu na klasu saobraćaja kome pripadaju. Inicijalno je korišćen **IP Precendece** gde se sa prva 3 bita označava prioritet, a sa preostalih 5 način tretiranja. Modifikovan je u **DSCP** (DiffServ Code Point) gde se sa 6 bita označavaju klasa saobraćaja i prioritet paketa, a preostala 2 bita su neiskorišćena.
- **Options** - dodatne opcije za potrebe testiranja i budućeg unapređenja.
- **Padding** - proširivanje do pune reči od 4B ako za opcije nije potrebna cela reč.
- **Time to Live** (**TTL**) - ograničenje broja koraka (prolazaka kroz ruter) koje paket može da napravi. Zbog dozvoljenih petlji u fizičkoj topologiji paketi bi potencijalno beskonačno kružili u nekim slučajevima. Pri svakom prolasku kroz ruter TTL vrednost se smanjuje za 1, a ako dostigne 0 paket se odbacuje čime se sprečava beskonačno kruženje. Koristi se 1B pa je maksimalan broj koraka 256 što je sasvim dovoljno za Internet.

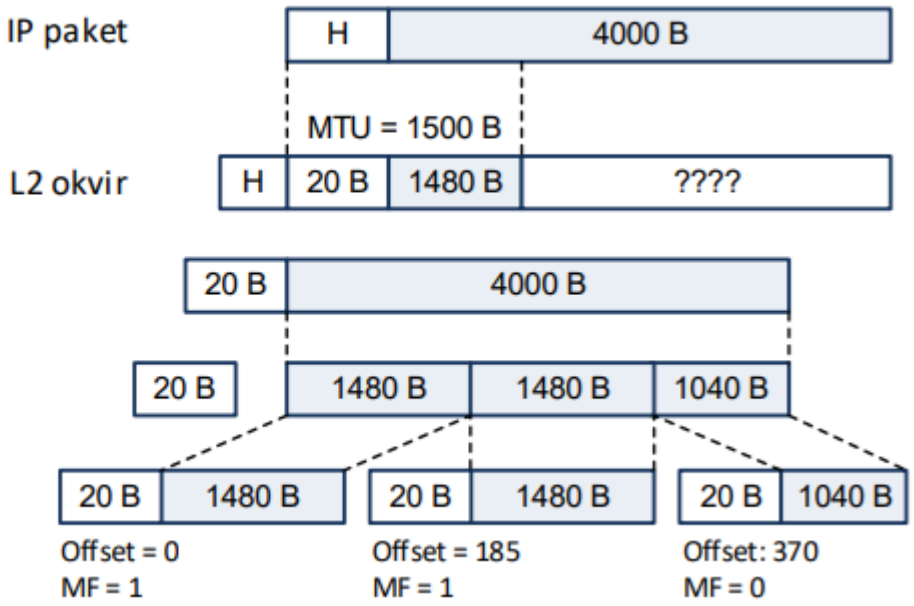


Okviri su na L2 nivou ograničeni sa **MTU** (**Maximum Transmission Unit**). Ako je IP paket veći od MTU i ne može da stane u okvir vrši se **fragmentacija** - podela jednog IP paketa na više manjih fragmenata. Sledeća polja IP zaglavlja služe za opisivanje fragmentacije:

- **Identification** - jedinstveni ID paketa od 2B. Svi fragmenti će imati isti ID, na osnovu čega znamo da pripadaju istom originalnom paketu.
- **Flags** - kontrolni bitovi. **Flag DF** (**Don't Fragment**) služi za zabranu fragmentacije. Jedinica znači da je fragmentacija zabranjena, a nula da nije. **Flag MF** (**More Fragment**) služi za označavanje da li postoji još fragmenata iza trenutnog. Jedinica znači da trenutni fragment nije poslednji, tj. da ima još fragmenata, a nula označava poslednji fragment.
- **Fragment Offset** - relativna pozicija podataka u odnosu na podatke iz originalnog IP paketa. Izražava se u jedinicama od 8 bajtova, što znači da dužine fragmenata moraju biti deljive sa 8.

Proces fragmentacije:

1. Ako je veličina IP paketa veća od MTU na L2 nivou, niz bajtova podataka deli se na više delova koji su zajedno sa IP zaglavljem manji od MTU. Dužina svakog dela mora biti umnožak od 8 bajtova, osim poslednjeg fragmenta koji sadrži ostatak.
2. Svaki fragment se enkapsulira u posebne IP pakete.
3. Fragment Offset se postavlja na relativni pomeraj od početka originalnog niza podataka. Za sve fragmente se MF flag setuje na 1, osim za poslednji. Preračunavaju se polja HLEN, Total Length i Header Checksum, a ostala polja se kopiraju iz originalnog zaglavlja.
4. Fragmentirani IP paketi mogu se ponovo fragmentirati.



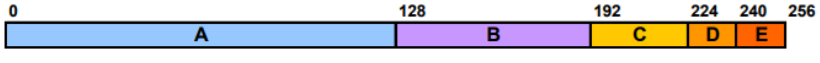
Reasembling je objedinjavanje podataka svih fragmenata u originalni niz podataka iz prvobitnog IP paketa. Sprovodi se na L3 nivou u odredištu. IP paket je fragmentisan ako mu je Fragment Offset različit od nule ili ako mu je MF flag različit od nule. Proces reasemblinga:

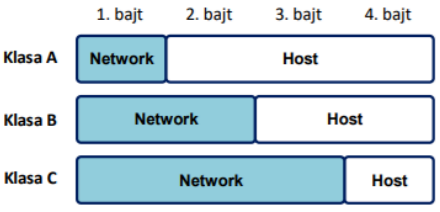
1. Alocira se bafer određene veličine i startuje se tajmer koji predstavlja maksimalno vreme čekanja da pristignu svi fragmenti. Vreme može biti predefinisano, a može se i prenositi u opcionim poljima IP zaglavlja.
2. Prikupljaju se svi IP fragmenti sa istim ID poljem i izdvajaju se pripadajući podaci.
3. Na osnovu polja Fragment Offset rekonstruiše se originalni niz podataka. Poslednji fragment se prepoznaje po resetovanom MF flagu.
4. U slučaju da ne stignu svi fragmenti tokom trajanja tajmera ili ako je oštećen bar jedan fragment podaci se odbacuju u celini.

IP adrese služe za identifikaciju uređaja. U zaglavlju IP paketa nalaze se izvorišna i odredišna IP adresa. IP adresa je dužine 4B pa je moguće adresirati 2^{32} ~ 4.3 milijarde uređaja. Za označavanje se koristi **"Dotted Decimal" notacija** - dekadni brojevi razdvojeni tačkom. LAN mreža na L2 nivou mapira se u IP mrežu na L3 nivou. IP adrese uređaja na istoj LAN mreži grupisane su u zajedničku mrežnu IP adresu. Odnosno, IP adresa se sastoji iz dva dela: **mrežna adresa (mrežni deo)** i **adresa uređaja u mreži (host deo)**.

Adresa mreže u host delu sadrži sve nule, a **broadcast adresa** u host delu sadrži sve jedinice. Adrese uređaja uzimaju proizvoljne preostale vrednosti u tom opsegu. Ako u host delu imamo n bita onda se može adresirati $2^n - 2$ uređaja. Jedna adresa služi i za adresiranje interfejsa rutera kojim je LAN mreža povezana na Internet. To može biti bilo koja adresa, ali najčešće ima vrednost prve adrese nakon adrese mreže. IP adresni prostor podeljen je u klase - **klasa A** za mrežni deo koristi 1B, **klasa B** koristi 2B, a **klasa C** 3B. Ove tri klase se dodeljuju korisnicima, dok je **klasa D** rezervisana za multikast, a **klasa E** za eksperimentalne upotrebe. **Classful adresiranje** podrazumeva da je mrežni deo svake IP adrese određen prema klasi kojoj ona pripada.

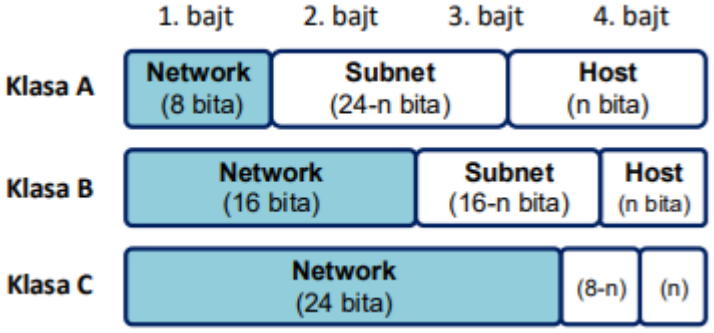
A	start	0	0	0	0	00000000	00000000	00000000	00000000
	end	127	255	255	255	01111111	11111111	11111111	11111111
B	start	128	0	0	0	10000000	00000000	00000000	00000000
	end	191	255	255	255	10111111	11111111	11111111	11111111
C	start	192	0	0	0	11000000	00000000	00000000	00000000
	end	223	255	255	255	11011111	11111111	11111111	11111111
D	start	224	0	0	0	11100000	00000000	00000000	00000000
	end	239	255	255	255	11101111	11111111	11111111	11111111
E	start	240	0	0	0	11110000	00000000	00000000	00000000
	end	255	255	255	255	11111111	11111111	11111111	11111111



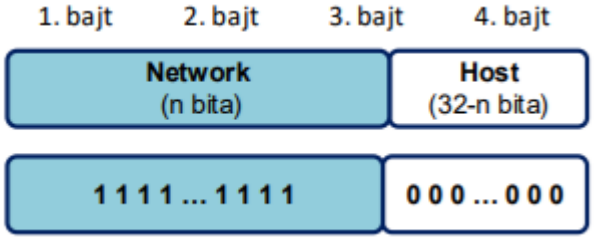


Dodeljivanje IP adresa ne može da vrši bilo ko, jer one moraju biti jedinstvene na Internetu. Prvobitno je to radila InterNIC organizacija, a nasledila ju je IANA koja dodeljuje delove IP adresnog prostora prema regionima (kontinentima) za koje su zaduženi regionalni internet registri. Pojedini mrežni opsezi su rezervisani za posebnu namenu. **Privatne adrese** su adrese za izolovano korišćenje nezavisno od Interneta i one ne smeju da budu vidljive na Internetu jer nisu jedinstvene. Postoje i druge rezervisane adrese, na primer **default ruta** obuhvata ceo opseg 0.0.0.0 - 0.255.255.255. U opsegu 127.0.0.0 - 127.255.255.255 postoji **loopback adresa** 127.0.0.1 preko koje referišemo na sopstveni uređaj.

Ako posmatramo klasu A u host delu postoji preko 16 miliona adresa, što deluje previše za samo jednu LAN mrežu. Uvodi se koncept **sabnetovanja** (**subnetting**) koji podrazumeva deljenje na manje podmreže. Za identifikaciju podmreže koristi se deo bitova host dela.



Naknadno se uvodi **Classless adresiranje** koje izjednačava pojam mreža i podmreža i potpuno zanemaruje klase. To znači da možemo uzeti proizvoljan broj bitova za mrežu, a preostale za host deo. Međutim, sada je potrebno označiti granicu između mrežnog i host dela. **Maska** predstavlja 4B sa vodećim jedinicama i ima ulogu da deli IP adresu na mrežni i host deo. Adresu mreže možemo dobiti ako primenimo bitsku AND operaciju na IP adresu i masku.



Može se zapisivati i u Dotted Decimal notaciji kao IP adrese (npr. 255.255.255.0) ili u **prefix notaciji** "/n" gde je n broj jedinica u masci (npr /24). Da bismo prebacili masku iz prefix notacije u dotted decimal notaciju prvo je potrebno podeliti masku u oktete, tj. bajtove, a zatim posmatrati poslednji bajt maske koji sadrži jedinice. Svi bajtovi pre njega će sadržati sve jedinice, tj. imaće vrednost 255. Poslednji bajt od interesa se može dobiti tako što vodeće jedinice pretvorimo u eksponente dvojke ili tako što od 256 oduzmemo eksponent dvojke gde se nalazi poslednja jedinica. Na primer, ako imamo masku /21 to znači da imamo 8 + 8 + 5 jedinica pa je treći bajt poslednji u kome se nalaze jedinice i oblika je 11111000. Odatle sledi da ima vrednost $2^7 + 2^6 + 2^5 + 2^4 + 2^3 = 248$ ili $256 - 2^3 = 248$. Dakle, maska je 255.255.248.0. Moguće je i korišćenje maski različitih dužina u jednoj mreži - **Variable Length Subnet Mask** (**VLSM**). One omogućavaju efikasnije korišćenje adresnog prostora, fleksibilniju preraspodelu adresa, skalabilan rast mreže i dodavanje novih adresa. Povećavanjem maske delimo adresni prostor na podmreže, ali možemo i da smanjujemo masku čime spajamo više susednih podmreža u veću podmrežu što se naziva **agregacija** (**supernetting**). Da bi se mreže agregirale, moraju da budu susedne i da se uklapaju u isti adresni blok koji se može opisati jedinstvenom adresom i maskom. Agregacija je potrebna jer na Internetu imamo eksponencijalni rast broja hostova pa bi bez agregacije ruteri odavno bili prezasićeni.

Transportni sloj

L4 sloj, odnosno **transportni sloj** ima ulogu da obezbedi komunikaciju za različite aplikacije između krajnjih uređaja. Podaci sa aplikativnog sloja se enkapsuliraju i prenose, a zatim na drugom kraju dekapsuliraju.

Moguće su dve vrste prenosa: **Byte-stream** podrazumeva segmentaciju i prenos niza bajtova, a **Message-stream** prenos celokupnih poruka. Multipleksiranje podrazumeva da se na strani pošiljaoca obeleži koja aplikacija je poslala poruku, a demultiplksiranje da se na osnovu oznake prepozna ciljna aplikacija. Multipleksiranje i demultipleksiranje vršilo se i na nižim nivoima. Na L2 nivou polje Type u zaglavlju identifikovalo je protokol L3 nivoa, a na L3 nivou polje Protocol u zaglavlju identifikovalo je protokol L4 nivoa. Na L4 nivou, identifikacija aplikacija na jednom uređaju naziva se **port**. Aplikacije mogu biti **serverske** i **klijentske** koje iniciraju komunikaciju sa serverskim aplikacijama. Za identifikaciju porta koriste se 2B pa port može imati vrednosti od 0 do 65535. U zavisnosti od namene portovi se dele na:

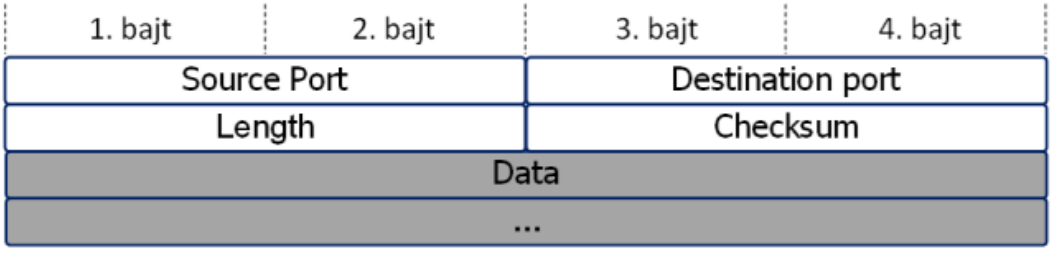
- **Well-known Ports** - samo za dobro poznate serverske aplikacije (npr. FTP, HTTP), opseg 0-1023
- **Registered Ports** - za klijentske i serverske aplikacije, opseg 1024-49151
- **Private (Dynamic) Ports** - samo za klijentske aplikacije, opseg 49152-65535

Soket (Socket) predstavlja jednoznačnu identifikaciju aplikacije na mreži. Sadrži IP adresu uređaja, broj porta aplikacije i identifikaciju transportnog protokola. Komunikacija između klijentskih i serverskih soketa je dvosmerna: korisnik šalje zahtev prema serveru, a server šalje odgovor prema klijentu. Serverske aplikacije su dostupne za pristup od strane proizvoljnih korisnika i soket ima unapred poznatu IP adresu, port i protokol. Klijentske aplikacije iniciraju komunikaciju sa serverskim aplikacijama i imaju proizvoljnu IP adresu i dinamički dodeljen port i protokol.

Pored osnovnih funkcija L4 sloj može vršiti i dodatne funkcije koje su za većinu aplikacija korisne, ali usporavaju prenos podataka:

- **uspostavljanje veze** - uspostavljanje i održavanje sesije.
- **pouzdan prenos** - garancija prenosa svih aplikativnih podataka i ponovno slanje izgubljenih ili oštećenih segmenata.
- **održavanje redosleda segmenata** - segmenti po različitim putevima mogu stići u promenjenom redosledu, ali prijemna strana rekonstruiše originalni redosled.
- **kontrola toka** - dinamičko povećavanje i smanjivanje protoka podataka u zavisnosti od mogućnosti i trenutnog opterećenja mreže.

Dve osnovne vrste transportnog protokola su **UDP** i **TCP**. **User Datagram Protocol** obezbeđuje samo osnovne funkcije. Kažemo da je connectionless message-stream protokol jer ne uspostavlja vezu, a prenosi celokupne poruke (datagrame) koje se dobijaju od aplikacije. Svaki paket se prenosi nezavisno. Nepouzdan je, ali jednostavan i brz.



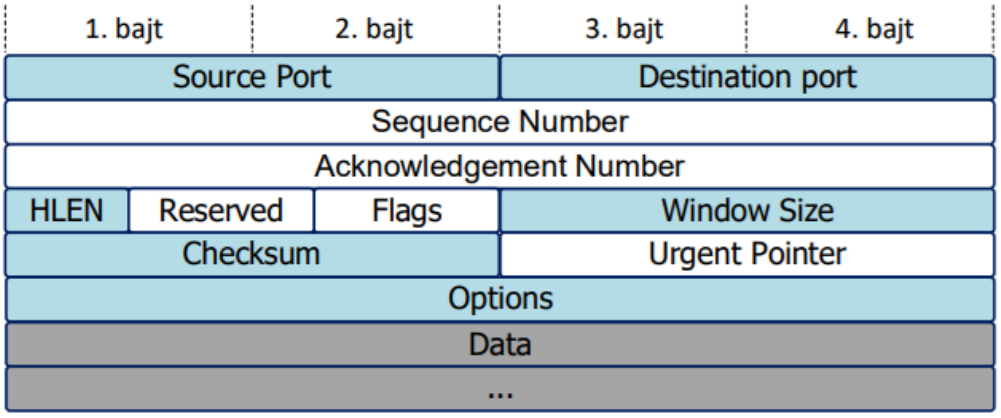
UDP zaglavlje sastoji se iz:

- **Source Port** i **Destination Port** - po 2B za izvorišni i odredišni port.
- **Length** - dužina podataka uključujući i zaglavlje.
- **Checksum** - 2B za proveru greške, računa se kao prvi komplement sume UDP zaglavlja, podataka i **pseudo-hedera**. Pseudo-heder čini izvorišnu i odredišnu IP adresu, identifikaciju UDP protokola i dužinu UDP paketa. Ovde se donekle krši princip razdvajanja slojeva jer koristimo informacije L3 nivoa.

UDP se primenjuje kod jednostavnih aplikacija sa periodičnom komunikacijom gde nije bitna pouzdanost. To su real-time aplikacije kao što je IPTV, Zoom, IP telefonije, video konferencije i slično. Za njih je bitan kontinuitet pristizanja poruka, čak i ako se neka poruka izgubi. Potrebno je da **kašnjenje (delay)** bude malo, kao i da **varijacija kašnjenja (jitter)** bude mala. UDP se koristi i kada je potreban broadcast ili multicast. Složenije funkcije

UDP prepušta aplikaciji - samostalno ili korišćenjem podslojeva. **Real-time Transport Protocol (RTP)** je podsloj L7 aplikativnog sloja koji vrši serijalizaciju, baferovanje, kontrolu džitera i slično.

Transmission Control Protocol pored osnovnih funkcija obavlja i sve dodatne funkcije pa je sporiji od UDP-a. Kažemo da je connection-oriented byte-stream protokol jer vrši uspostavljanje veze sa odredištem, a prenosi segmentisane nizove bajtova. Vrši pouzdan prenos podataka, nezavisno u oba smera. S obzirom da se komunikacija uspostavlja s-kraja-na-kraj ne podržava broadcast ili multicast. Koristi Full-Duplex arhitekturu kako bi ostvario nezavisnu komunikaciju u oba smera. Čak i kada se aplikativni podaci prenose samo u jednom smeru, u drugom smeru se prenose kontrolni podaci. Byte-stream prenos podrazumeva **segmentaciju (segmentation)** koja se vrši na strani pošiljaoca. Aplikativni sloj predaje niz bajtova, podeljen u proizvoljne delove, a onda TCP vrši podelu većih ili spajanje manjih u segmente. Na drugom kraju vrši se **objedinjavanje (reassembling)**, odnosno rekonstrukcija originalnog niza bajtova aplikativnih podataka.

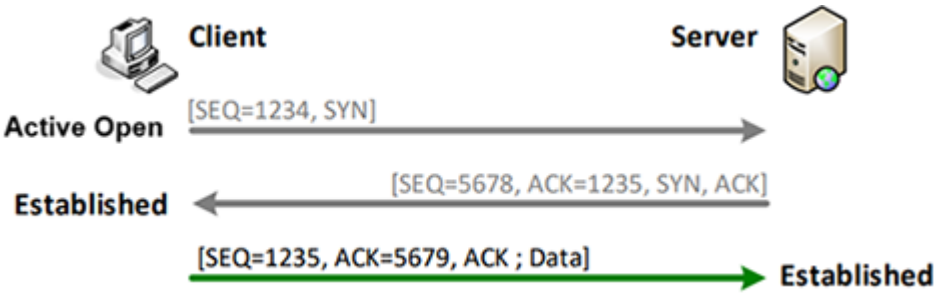


TCP zaglavlje sastoji se iz:

- **Source Port**, **Destination Port**, **Checksum** - kao kod UDP.
- **HLEN (Header Length)** - 4 bita koji predstavljaju dužinu zaglavlja u rečima (4B).
- **Window Size** - ukupan broj bajtova koji se mogu poslati pre nego što se čeka na potvrdu.
- **Options** - dodatne opcije, varijabilne dužine.
- **Sequence Number (SEQ)** - obezbeđuje identifikaciju segmenata. Inicijalna vrednost prvog segmenta je slučajno izabran broj u fazi uspostavljanja veze, a svaki sledeći segment ima vrednost za jednu veću.
- **Acknowledgement Number (ACK)** - koristi se za potvrdu prijema kontinualnog niza bajtova. Predstavlja SEQ sledećeg segmenta koji se očekuje za prijem, odnosno ima značenje "ovo je pozicija sledećeg segmenta koji se očekuje za prijem, a svi prethodni su uspešno primljeni".
- **Flags** - kontrolni flagovi koji opisuju značenje paketa i drugih polja u zaglavlju.
  - **SYN (Synchronization)** - inicijalizacija SEQ vrednosti, tj. u pitanju je prvi segment.
  - **ACK (Acknowledgement)** - polje Acknowledgement Number je validno.
  - **FIN (Finish)** - poslednji segment, završetak konekcije u jednom smeru.
  - **RST (Reset)** - resetovanje konekcije.
  - **PSH (Push)** - zahteva se momentalna predaja segmenta aplikaciji na prijemu, bez baferovanja i čekanja ostalih segmenata.

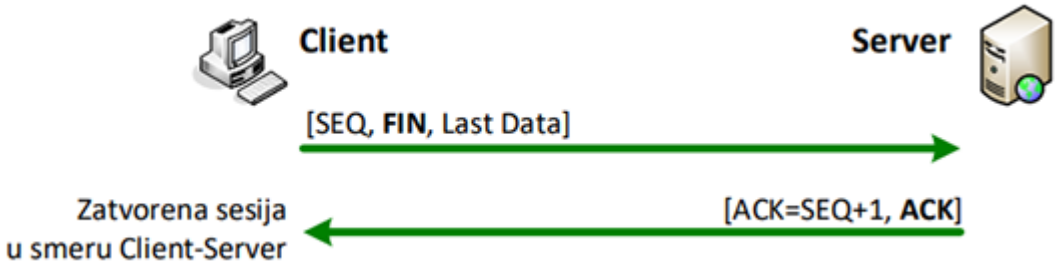
TCP je Full-Duplex pa održava dve odvojene komunikacione sesije, po jednu u oba smera. To podrazumeva dva nezavisna para SEQ i ACK. **Uspostavljanje sesije u oba smera** vrši se u 3 koraka (**Three-way handshake**):

- **Active Open** stanje - klijent šalje zahtev za uspostavljanje sesije serveru. Šalje se inicijalna vrednost SEQ pa se postavlja SYN flag.
- **Established** stanje - server šalje potvrdu otvaranje sesije klijentu. ACK vrednost je za jedan veća od dobijene SEQ vrednosti i postavljen je ACK flag. Šalje se i zahtev za otvaranje sesije u drugom smeru (od servera ka klijentu) pa se šalje inicijalna vrednost SEQ i postavlja SYN flag.
- **Established** stanje - klijent šalje potvrdu otvaranja sesije serveru. ACK vrednost je za jedan veća od dobijene SEQ vrednosti od servera i postavljen je ACK flag. Klijent sada šalje i podatke, tj. uspostavljena je sesija u oba smera.



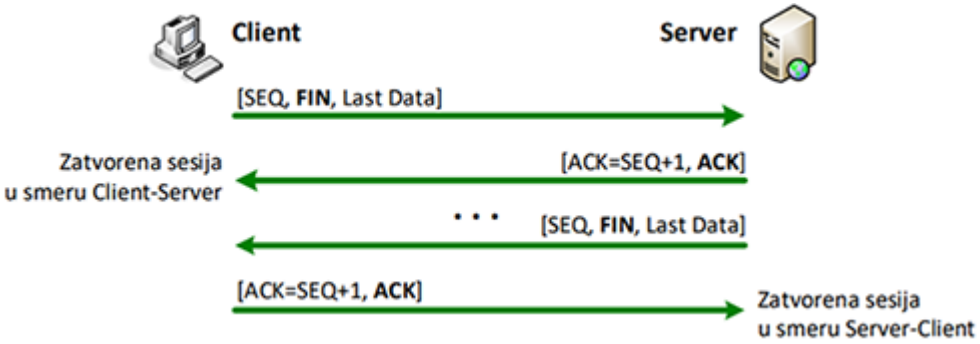
Raskidanje sesije u jednom smeru vrši se u 2 koraka (**Two-way handshake**):

- Kada jedna strana nema više podataka za slanje šalje se FIN flag.
- Druga strana potvrđuje prijem poslednjeg paketa slanjem ACK za taj paket i sesija u tom smeru je zatvorena.

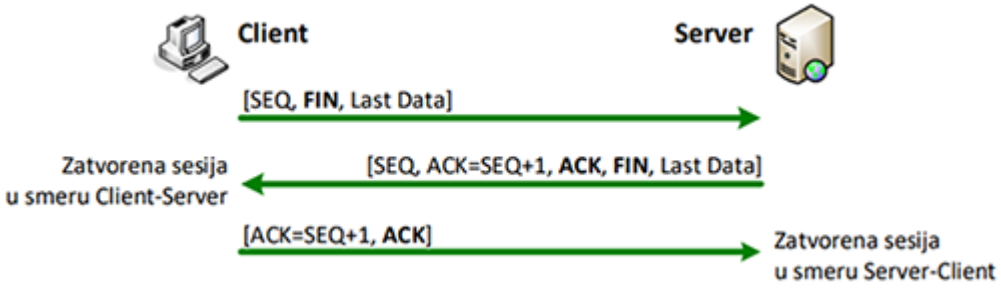


Raskidanje sesije u oba smera vrši se na dva načina:

- 4 koraka (**2x Two-way handshake**):
  - Raskidanje sesije u jednom smeru (2 koraka). Ako druga strana ima još podataka za slanje, oni se šalju u drugom smeru.
  - Kada više nema podataka za slanje u drugom smeru šalje se FIN flag.
  - Potvrđuje se prijem poslednjeg paketa slanjem ACK i sesija se raskida i u drugom smeru.



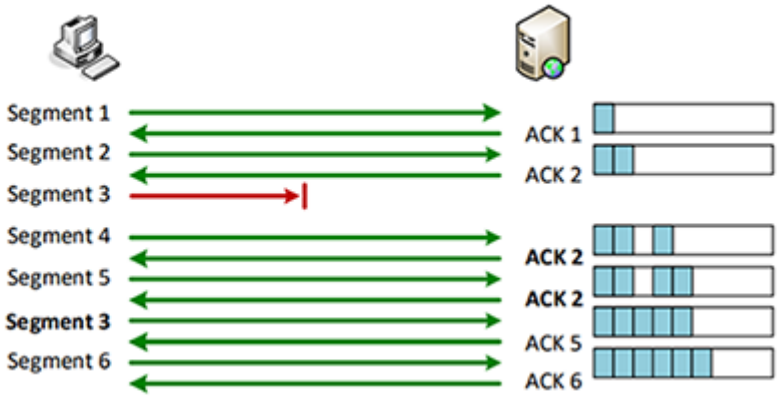
- 3 koraka:
  - Kada jedna strana nema više podataka za slanje šalje se FIN flag.
  - Druga strana potvrđuje prijem poslednjeg segmenta slanjem ACK. Ako druga strana nema podataka za slanje ili ih može poslati u poslednjem paketu šalje se FIN flag.
  - Potvrđuje se prijem poslednjeg paketa slanjem ACK.



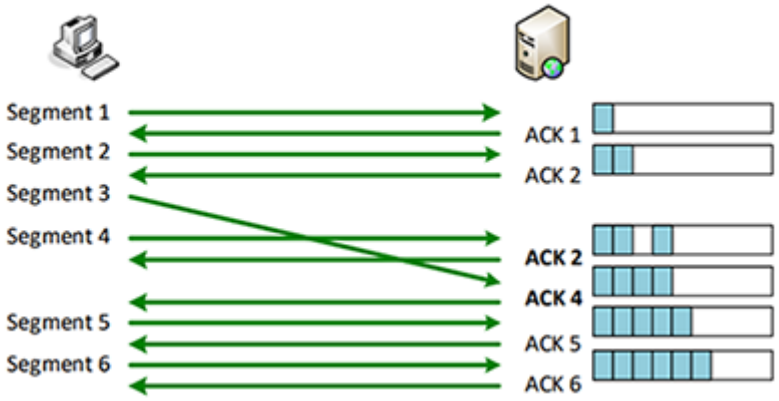
Pouzdan prenos svakog segmenta podrazumeva potvrdu primljenih podataka putem SEQ i ACK vrednosti. Vrši se nezavisno u oba smera, što znači da se prenose različiti SEQ u oba smera, a ACK predstavlja potvrdu prenosa u drugom smeru. Važi $SEQ_c + DATA_c = ACK_s$ i obrnuto $SEQ_s + DATA_s = ACK_c$. Ako se izgubi paket koji prenosi

podatke ili paket koji prenosi potvrdu čeka se određeno vreme - **timeout**. Vreme čekanja treba da bude veće od RTT (Round Trip Time), ali RTT varira tokom vremena pa ga treba proceniti. Jedan način je da se RTT procenjuje dinamički tako što se uzima težinski prosek prethodne procene RTT_{old} i vrednost RTT za poslednji poslati i potvrđeni segment RTT_{new}, odnosno RTT = a · RTT_{old} + (1 - a) · RTT_{new}, a $\in [0, 1]$. Za timeout se onda uzima vrednost b · RTT, pri čemu je b faktor uvećanja koji najčešće ima vrednost 2. Ukoliko istekne timeout period segment se ponovo šalje, tj. vrši se retransmisija.

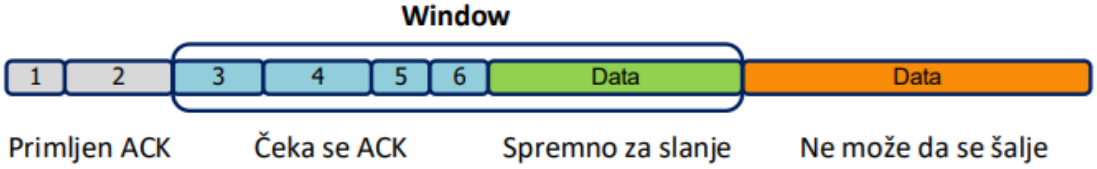
Oporavak od greške vrši se na osnovu SEQ i ACK vrednosti. ACK vrednost šalje se samo kada pristigne neki segment i uvek se odnosi na poslednji segment u kontinuitetu. Ako na primer 3. segment ne bude uspešno primljen uvek će se vraćati ACK vrednost 2 sve dok se on ne primi uspešno. Višestruka ACK vrednost označava da se i dalje očekuje segment 3, iako uspešno pristižu segmenti 4 i 5.



Rekonstrukcija redosleda segmenata vrši se na prijemnoj strani i to na osnovu SEQ vrednosti primljenih segmenata. Ako podaci stižu različitim putanjama može doći do promene redosleda prijema, odnosno pojave rupa. Rekonstrukcija je uspešna tek kada se popune sve te rupe.

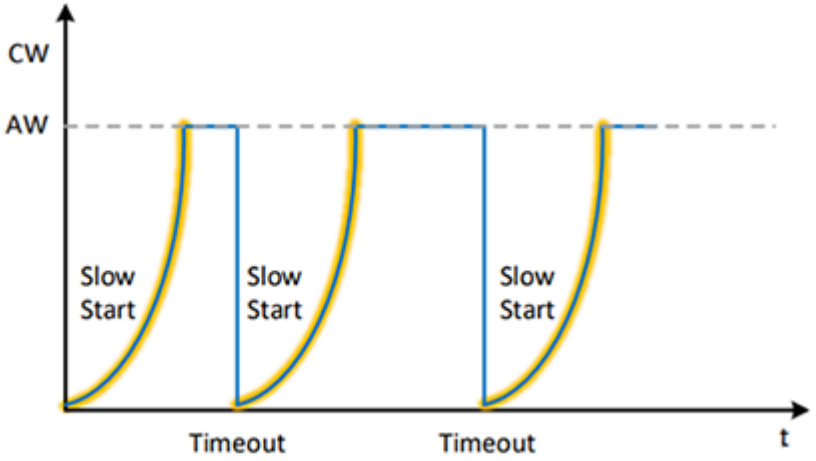


Kontrola toka radi na principu mehanizma prozora (**Window**) i sprovodi se na strani koja šalje podatke, nezavisno u oba smera. Pre prozora nalaze se podaci koji su već poslati i za koje je primljen ACK, tj. potvrda o prijemu. U prozoru nalaze se poslati podaci za koje se čeka ACK, kao i podaci koji su spremni za slanje. Posle prozora nalaze se podaci koji se ne mogu poslati dok se ne dobije potvrda za podatke iz prozora.

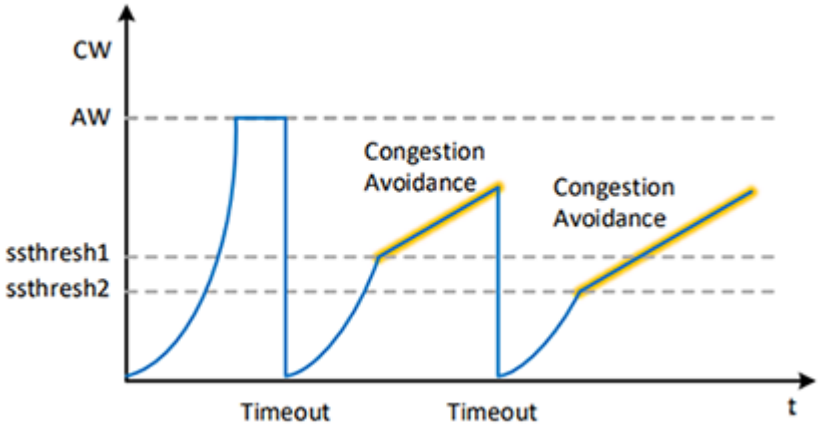


Prozor se pomera kada se dobije ACK za segment na početku prozora - **Sliding Window**. Kada se prozor popuni obustavlja se sa slanjem novih segmenata i čeka se potvrda segmenata sa početka prozora. Potvrđeni segmenti se oslobađaju, prozor se pomera i oslobađa se prostor u prozoru za slanje novih segmenata na kraju prozora. Kontrola toka se uspostavlja veličinom prozora. Manji prozor znači sporije slanje, a veći prozor brže slanje. Veličina prozora uspostavlja se dinamički - **Dynamic Window**. Inicijalno se obe strane dogovore o veličini prozora. U slučaju opterećenja prijemne strane ili gubitka paketa od pošiljaoca može da se zahteva smanjenje veličine prozora čime se usporava slanje. Ako nema grešaka, veličina prozora može postepeno da se povećava. Prilagođavanjem prozora trenutnom opterećnju vrši se **kontrola zagušenja**. Pri kontroli zagušenja koriste se sledeći algoritmi:

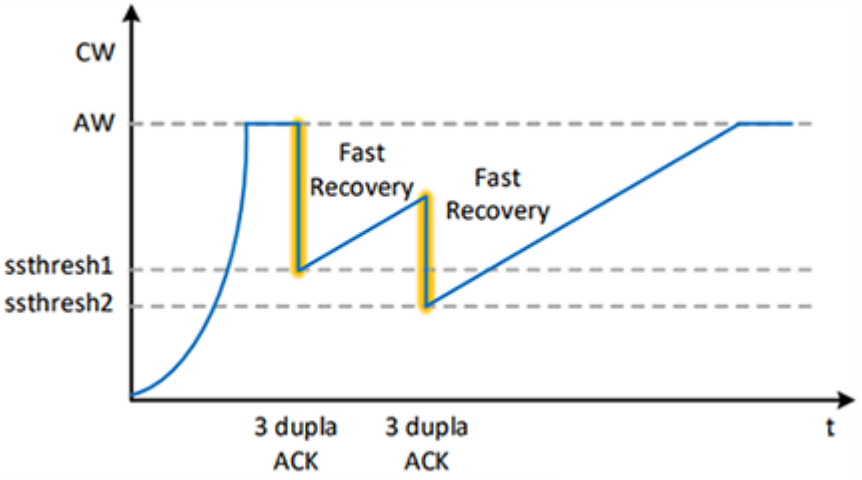
- **Slow Start**: Primalac postavlja inicijalnu vrednost prozora koja se naziva **Advertised Window (AW)** čime kontorliše brzinu prijema segmenta. Da bi se izbeglo zagušenje, pošiljalac postepeno povećava stvarnu veličinu prozora koja se naziva **Congestion Window (CW)**. Inicijalna vrednost CW je 1 segment, a povećava se dva puta svakim prijemom ACK vrednosti. Ako se potvrđuje svaki segment CW se eksponencijalno povećava do maksimalne vrednosti AW. Cilj je krenuti oprezno sa malom brzinom prenosa, a onda dostići maksimalnu brzinu ako mreža to može da podnese. Ako nastane gubitak segmenta i timeout počinje novi Slow Start.



- **Congestion Avoidance**: Prozor se povećava linearno, a ne eksponencijalno. U Slow Start fazi veličina prozora raste eksponencijalno do **Slow Start Threshold Size (ssthresh)**, a potom nastavlja da raste linearno. Ako ne stigne ACK za neki segment sstresh se smanjuje za polovinu poslednje vrednosti.

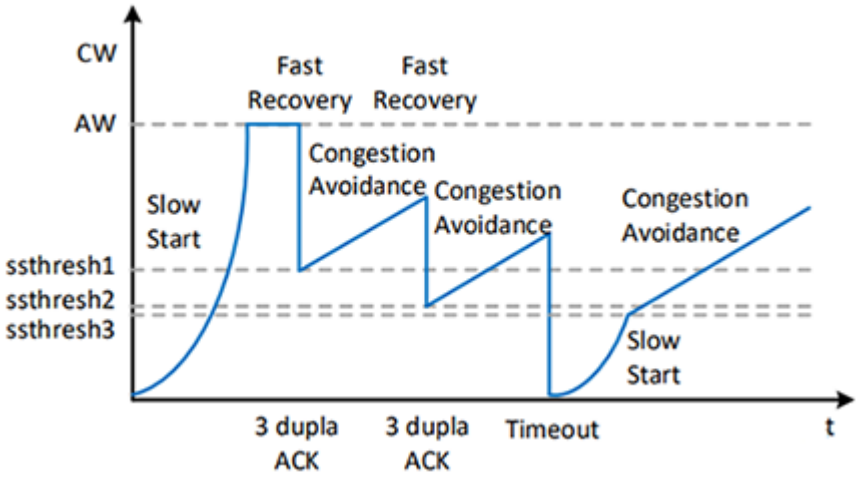


- **Fast Retransmit**: Ponovno slanje segmenta pre isteka timeout intervala. Ako imamo 1 ili 2 dupla ACK smatramo da segment nije izgubljen, već da je došlo do promene redosleda. Ako dobijemo 3 dupla ACK smatramo da je paket izgubljen i šaljemo ga ponovo, ne čekajući timeout da istekne.
- **Fast Recovery**: Primenjuje se nakon Fast Retransmit faze. Pošto je paket ponovo poslat tokom Fast Retransmit faze smatramo da nema potrebe za drastičnim usporenjem slanja podataka, pa se prozor smanjuje na vrednost sshtresh (polovina od CW) umesto da se kreće od početka (veličina od 1 segmenta). Nakon Fast Recovery faze ide se direktno u Congestion Avoidance, bez Slow Start faze.



Dakle, kontrola zagušenja vrši se na sledeći način:

- Slow Start - početak od najmanje veličine prozora uz eksponencijalni rast. Takođe se radi ako je došlo do gubitka segmenta uz timeout.
- Congestion Avoidance - linearni rast, kada se dostigne polovina prethodne veličine prozora.
- Fast Retransmit - ponovno slanje segmenta nakon 3 dupla ACK.
- Fast Recovery - smanjenje veličine prozora na polovinu i direktan ulazak u Congestion Avoidance.



Postoje različiti alati za proveru TCP konekcija:

- **netstat** - prikazuje otvorene TCP konekcije na lokalnom uređaju.
- **nmap** - prikazuje otvorene TCP portove na udaljenom uređaju čija se IP adresa prosleđuje kao argument.

Rezime:

TCP	UDP
Connection oriented	**Connectionless**
Prenos kontinualnog niza bajtova	**Prenos nezavisnih poruka**
Pouzdan prenos	**Nema garancije prenosa**
Kontrola toka	**Nema kontrole toka**
Zaglavlje 20B	**Zaglavlje 8B**
Sporiji i složeniji	**Brz i jednostavan**
Primenjuje se kada je bitna pouzdanost (veb, mejl, FTP, baze podataka, poslovne aplikacije...)	**Primenjuje se kada nije bitna pouzdanost (jednostavne aplikacije, periodična komunikacija) i kada je bitna brzina (real-time saobraćaj kao kod IPTV, IP telefonije, video konferencija...)**

Specifičnosti veb saobraćaja i potreba za enkripcijom podataka (što ni TCP ni UDP ne pružaju) dovode do razvoja **Transport Layer Security (TLS) protokola**. Aplikacije ga koriste za šifrovanje saobraćaja. On na transportnom sloju koristi TCP. Google objedinjuje funkcionalnosti TCP i TLS u **QUIC** protokol za sigurnu veb komunikaciju. QUIC na transportnom nivou koristi UDP protokol. QUIC omogućava brzo uspostavljanje veza, efikasniji i fleksibilniji mehanizam kontrole toka kao i mogućnost zadržavanja QUIC veze čak i prilikom promene IP adrese klijenta (npr. pri prelasku sa WLAN na 5G).

Aplikativni sloj

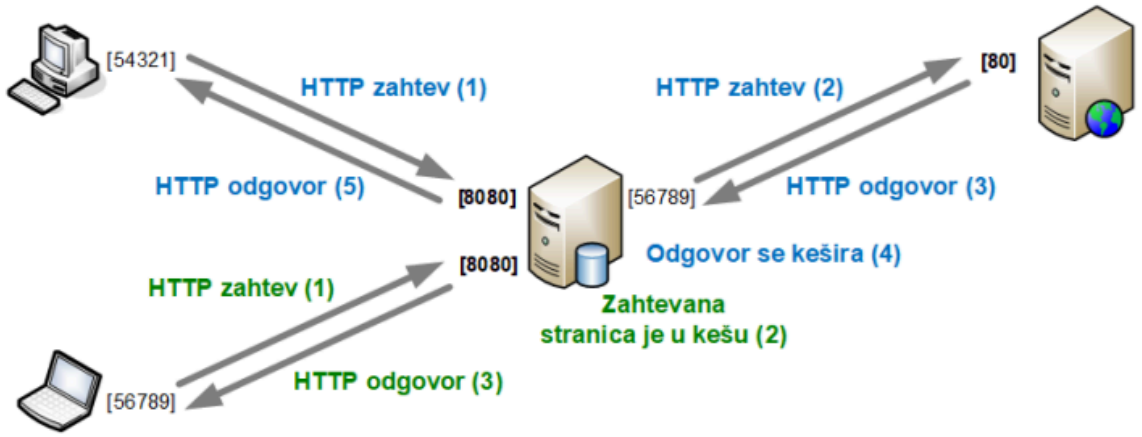
Komunikacija na internetu zasniva se na **klijent-server arhitekturi**. Server pokreće serverske aplikacije koje su otvorene za pristup od strane proizvoljnih korisnika. Imaju unapred poznatu IP adresu i poznat TCP ili UDP port. Klijenti (korisnici) putem klijentskih aplikacija iniciraju komunikaciju sa serverskim aplikacijama. Njihov soket karakteriše proizvoljna IP adresa i dinamički dodeljen TCP ili UDP port. Komunikacija između klijentskih i serverskih soketa je dvosmerna - zahtev od klijenta prema serveru i odgovor servera prema klijentu. Najpoznatiji internet servis je **web servis (World Wide Web - WWW)** koji je kreirao Tim Berners-Lee 1989. godine. On vrši prenos tekstualnih poruka, sa posebnim tagovima i ugnježdenom strukturom koristeći **HTTP (HyperText Transfer Protocol) protokol**. Tekst se formatira koristeći **HTML (HyperText Markup Language)**. Moguć je i prenos slika, audio i video zapisa tako što se binarni podaci referenciraju i prenose kao posebni objekti. HTTP protokol ima TCP port 80. Kasnije je uveden **HTTPS (Secure HTTP) protokol** koji omogućava šifrovanje i zaštićen prenos podataka. On ima TCP port 443. HTTP ima dva moda za održavanje konekcija:

- **Non-Persistent** - za svaki upit se uspostavlja TCP veza koja se po završetku raskida.
- **Persistent** - uspostavlja se jedna TCP veza koja se koristi za više upita tokom nekog perioda (timeout). Ovaj pristup je efikasniji jer se ne otvara svaki put nova veza.

Takođe, ima i dva moda za praćenje konekcija:

- **Stateless** - ne pamti stanje aktivnosti klijenta.
- **Stateful** - pamti stanje aktivnosti klijenta korišćenjem **kolačića (cookies)**.

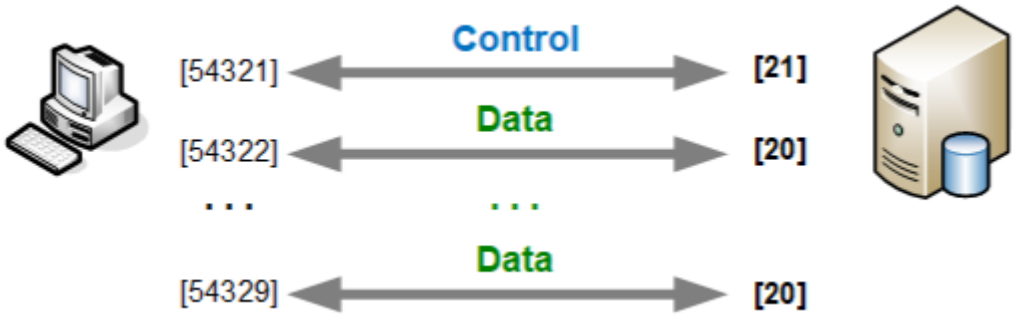
Proxy servis (Web Cache) predstavlja posredni server za HTTP protokol. Kada klijent zahteva komunikaciju sa HTTP-om zahtev se šalje preko proxy servera. Proxy server kešira stranice koje klijenti zahtevaju. Ako se ponovo traže iste stranice, čak i od strane drugog korisnika, proxy vraća keširane podatke umesto da ih opet zahteva od HTTP-a. Zbog ovoga proxy ima brži odziv nego direktan poziv HTTP-a. Povećana je i privatnost korisnika jer je njihov "identitet" sakriven za spoljne servere, sve što oni vide je proxy. Na ovaj način se postiže i filtriranje saobraćaja (**blacklist**) kao i dozvola pristupa određenom saobraćaju (**whitelist**). Podešavanje proxy servera vrši se u podešavanjima pretraživača. Najčešće ima port 8080.



Pre pojave veba korišćen je **FTP (File Transfer Protocol) protokol** za prenos datoteka. On koristi dve TCP konekcije:

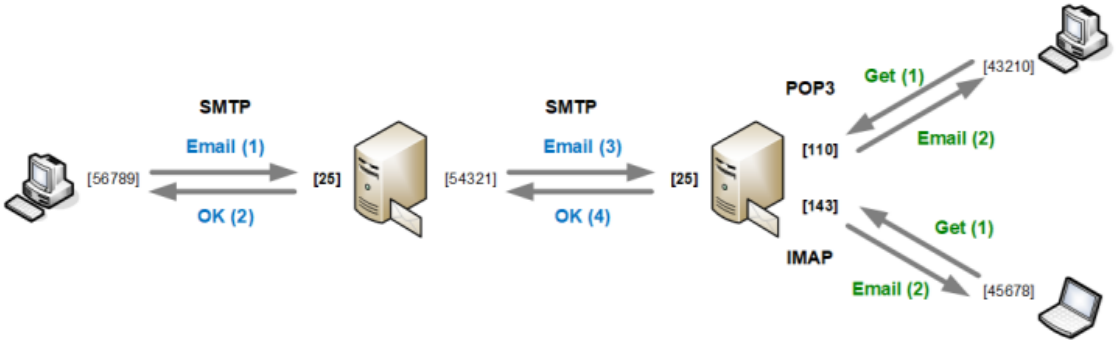
- **Kontrolna konekcija** - TCP port 21, služi za zadavanje tekstualnih komandi (npr. put, get, quit). Konekcija je otvorena dok je korisnik ne zatvori komandom quit. Konekcija je Stateful, odnosno pamti stanje aktivnosti korisnika.
- **Konekcija za podatke** - TCP port 20, služi za prenos jedne datoteke. Prenos se vrši u oba smera, od klijenta do servera (put) i od servera do klijenta (get).

Danas se prenos datoteka obično realizuje preko veb servisa i HTTP protokola. Postoji i Trivial FTP (TFTP) koji koristi UDP port 69 za prenos jednostavnijih datoteka.



Email servis služi za slanje elektronske pošte. Pošta se šalje od pošiljaoca do matičnog servera gde pošiljalac ima otvorenu elektronsku poštu, kao i od servera pošiljaoca do servera primaoca. Za slanje se koristi **SMTP (Simple Mail Transfer Protocol) protokol** koji ima TCP port 25. Postoji i bezbedna verzija protokola sa enkripcijom - SSL/TLS encrypted SMTP. Da bi primalac preuzeo poštu on pristupa matičnom serveru gde ima otvorenu elektronsku poštu i preuzima pristiglu poštu. Za preuzimanje pošte koriste se protokoli:

- **POP3 (Post Office Protocol, v3)** - TCP port 110
- **IMAP (Internet Message Access Protocol)** - TCP port 143

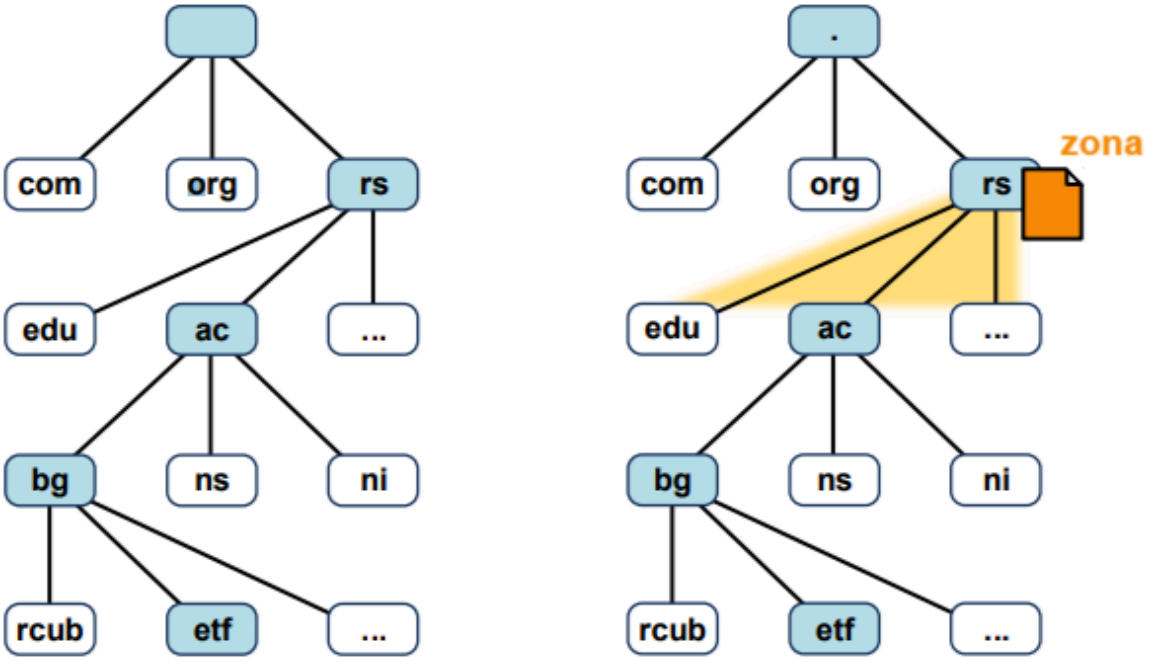


Pored pomenutih aplikacija, korisne su i aplikacije koje pružaju **udaljeni pristup uređajima**. Neke od njih su:

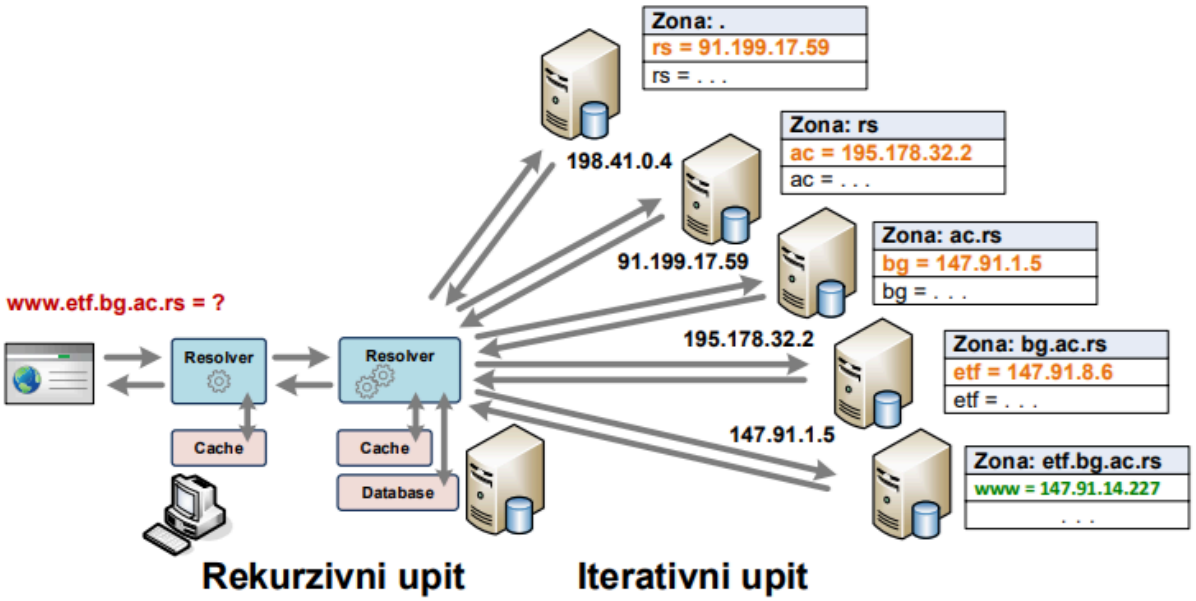
- **Telnet** - TCP port 23. Omogućava udaljeni pristup tekstualnoj konzoli (Command Line Interface - CLI).
- **SSH (Secure Shell)** - TCP port 22. Omogućava šifrovani udaljeni pristup tekstualnoj konzoli i uglavnom se koristi na Linux sistemima.
- **RDP (Remote Desktop Protocol)** - TCP ili UDP port 3389. Omogućava udaljeni pristup grafičkoj konzoli na Windows sistemima.

Domain Name System (DNS) je servis za pretvaranje simboličkih naziva u IP adrese i obrnuto. IP adrese su zgodne za mašinsko korišćenje, ali ne i za korisnike pa se uvode simbolička imena. Inicijalno su nazivi svih računara bili definisani u jednoj datoteci koja se zvala HOSTS.txt, a korisnici bi putem mreže periodično preuzimali ovu datoteku i koristili je. Ovaj pristup je bio koristan samo dok je na mreži postojao mali broj računara, ali nije skalabilan. DNS omogućava distribuirano definisanje i translaciju IP adresa i imena. Hijerarhija imena je u obliku stabla. Koren stabla je **root domen** koji se označava praznim stringom. Čvorovi stabla predstavljaju simboličke nazive samih računara. **Apsolutni naziv domena** predstavlja putanju od čvorova do korena stabla, gde su nazivi čvorova razdvojeni tačkom. Prazan string i poslednja tačka se obično izostavljaju, na primer umesto "matf.bg.ac.rs. " piše se "matf.bg.ac.rs". **Relativni naziv domena** predstavlja sam naziv nekog poddomena. Na primer, "matf" je poddomen domena "bg.ac.rs", a "bg" je poddomen domena "ac.rs". Svaki poddomen je maksimalne dužine od 63 karaktera, a maksimalna dužina punog imena je 255 karaktera. U imenima poddomena mogu se koristiti slova, brojevi, donja crta ("_") i crtica ("-") i ne pravi se razlika između velikih i malih slova. **Top Level Domains (TLD)** su poddomeni root domena. Inicijalno su postojali com, edu, gov, net, org, mil i slično, a danas postoje razni TLD od kojih su najzastupljeniji upravo com, org i net. Kasnije su uvedeni i **County Code TLD** koji pripadaju različitim državama, npr. yu, rs, eu i slično. Logička struktura je fizički organizovana na distribuirani način, tako što je celo stablo podeljeno u zone. **Zona** je deo stabla koji sadrži informacije o pripadajućim domenima. Obično je jedna zona jedan čvor (domen) i administrativno pripada jednoj celini kao što su firme, univerziteti, države i slično. Čuvaju se kao tekstualne datoteke na **DNS serverima**.

Primarni DNS server nekog domena je DNS server na kome je definisana zona za taj domen. Sadrži sve podatke o domenu, kao i definicije poddomena. **Sekundarni DNS server** je DNS server koji periodično preuzima zonu od primarnog DNS servera (**transfer zone**). Preporuka je da postoji bar jedan sekundardni DNS server za svaki domen kao rezervna kopija. Kažemo da su primarni i sekundardni DNS serveri **autoritativni DNS serveri** jer sadrže celokupne zone za određeni domen i ravnopravno učestvuju u razrešavanju imena. Na taj način se rasterećuje rad primarnog servera i postiže veća pouzdanost. **Delegacija zona** podrazumeva da zona domena definiše nazive poddomena, dok su ostali poddaci o poddomenima definisani u odvojenim zonama. Topologija domena je tehnički potpuno nezavisna od topologije fizičkog povezivanja u mreži, odnosno uređaji iz jednog domena mogu da pripadaju različitim mrežama i obrnuto, uređaji u jednoj fizičkoj mreži mogu da pripadaju različitim domenima.



DNS serveri razrešavaju upite klijenata. Svaki uređaj ima lokalno podešene DNS servere kojima šalju upite. **DNS Solver** je deo DNS softvera koji razrešava imena, odnosno uparuje nazive sa IP adresama. IP adresa se prvo traži na strani klijenta u lokalnom kešu. Ako tu ne postoji podatak šalje se upit lokalno podešenom DNS serveru. Ako ni na serveru nema podataka u lokalnom kešu ili bazi zona šalje se upit drugim DNS serverima koji su autoritativni za odgovarajući domen.



Razlikujemo dve vrste upita:

- **rekurzivni upit** - DNS server u potpunosti vraća konačan odgovor ili grešku. Generalno je namenjen za upite klijenata prema serveru.

- **iterativni upit** - DNS server daje delimičan odgovor. Na primer, želimo da pristupimo na "`www.etf.bg.ac.rs`". Naš lokalni DNS server vrati odgovor da ne može da razreši koja je IP adresa u pitanju, ali može da referiše na druge servere u hijerarhiji koji mogu da reše upit. Prvo će referisati na root domen koji će mu vratiti IP adresu DNS servera za "rs" domen. Domen "rs" će vratiti odgovor za svoj poddomen "ac". Poddomen "ac" onda traži od autoritativnog servera za "bg" da pronađe IP adresu tog poddomena. On dalje nalazi adresu za "etf" koji onda konačno pronalazi IP adresu za ime "`www.etf.bg.ac.rs`".

Zone se definišu kao tekstualni fajlovi. Osnovna jedinica podataka u tim fajlovima su **Resource Record** koji sadrže informacije o imenima, adresama i neke druge parametre. Osnovna sintaksa RR podatka je `Name TimeToLive Class Type Value`:

- **Name** - naziv podatka (domen ili host adresa)
- **Time To Live (TTL)** - vreme validnosti podatka u kešu
- **Class** - za Internet oznaka "IN"
- **Type** - tip podatka
- **Value** - vrednost pridružena podatku, npr. adresa

Tipovi RR podataka su:

- **SOA (Start of Authority)** - definiše se na početku svake zone. Sadrži naziv primarnog DNS servera i email adresu DNS admina.
- **Serial** - definiše serijski broj zona fajla. Preporučeni format je "yyyymmddnn" jer ovaj broj mora da se inkrementira pri svakoj promeni i na ovaj način sekundarni serveri znaju kada je nastala nova verzija.
- **Refresh** - posle koliko sekundi sekundarni DNS server proverava primarni da vidi da li ima promena.
- **Retry** - ako je prethodna provera neuspela, posle koliko sekundi se ponavlja.
- **Expire** - koliko dugo u sekundama DNS čuva zone učitane od primarnog DNS.
- **Minimum TTL** - koliko dugo se rekordi iz zone čuvaju u lokalnom kešu drugih DNS servera.
- **NS** - definiše autoritativne DNS servere za tekuću zonu ili poddomene.
- **MX** - definiše email server za tekuću zonu ili poddomene.
- **A** - definiše adresu veb servera za navedeno ime ili predefinisani server za tekuću zonu. Ako se za poddomen navodi DNS server preko imena, obavezno mora da bude definisana i IP adresa tog DNS servera. Kažemo da se uvodi **glue record** - IP adresa DNS servera poddomena, definisana u zoni domena.
- **CNAME (Cannonical Name)** - uvođenje alternativnih naziva za već definisane nazive. Prednost je što se može definisati više naziva za jednu adresu što je česta potreba. Nedostatak je što se razrešavanje upita za alijas vrši u dva koraka - prvo se vraća originalni naziv, a potom adresa za originalni naziv.
- **PRT** - koristi se za mapiranje IP adresa u nazive. Kreiran je domen "in-addr.arpa" u kojem su sve IP adrese u inverznom dotted-decimal formatu. Na primer, IP adresa 147.91.1.7 se mapira u "7.1.91.147.in-addr.arp", jer su kod IP adresa vodeći bajtovi dominantniji.

Domeni se mogu zakupiti preko **DNS provajdera** kao što je GoDaddy. Korisni **DNS alati**:

- **nslookup** i **ping** - pružaju informacije o domenu čije je ime prosleđeno kao argument. Koriste se na Windowsu.
- **dig** - na Linuxu pruža informacije o domenu čije je ime prosleđeno kao argument. Ako se ne da argument vraća informacije o 13 root domena.
- **NetVizura** - grafički prikaz povezanosti domena i detaljne informacije o domenima.

Primer definicije zone: Definišemo primarni DNS za `bg.ac.rs`, pri čemu se istovremeno definišu i osnovni parametri u zagradi. Ispod definišemo autoritativne servere (NS), mejl (MX) i adresu veb servera (A). Zatim uvodimo novi poddomen `etf` sa autoritativnim serverom `ns.etf.bg.ac.rs.`. Pošto smo uveli DNS server koji počinje sa `ns.etf` moramo da definišemo njegovu IP adresu (glue record). Za `proxy-web` postavljamo adresu veb servera, a zatim mu dajemo alias `www`.

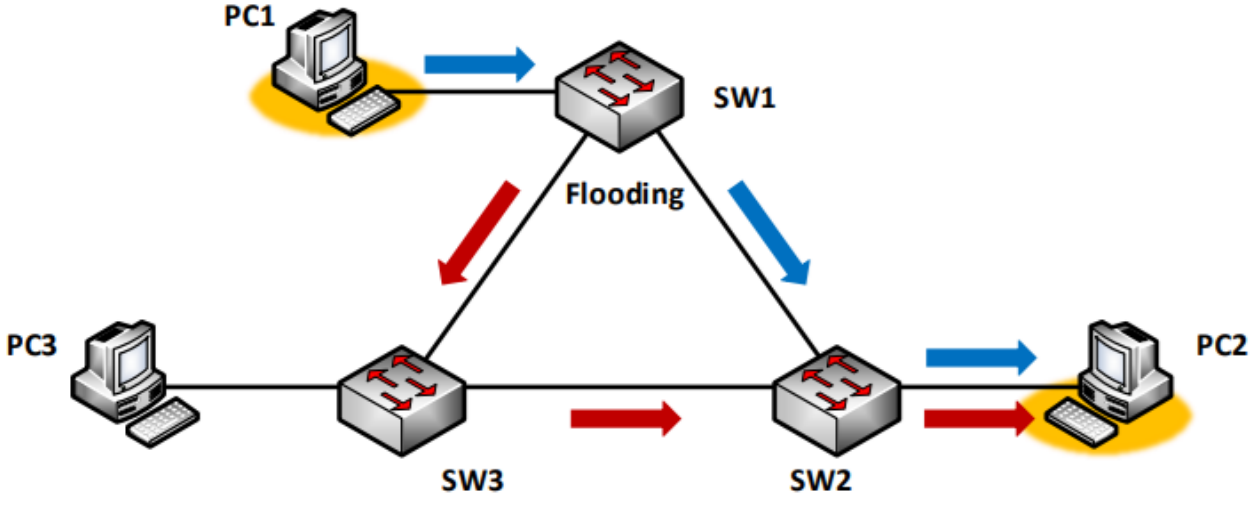
```
bg.ac.rs  IN    SOA NS1.NIC.RS. HOSTMASTER.BG.AC.RS. (
                    2019042000 ; serial
                    10800 ; refresh
                    3600 ; retry
                    2419200 ; expire
                    86400 ; minimum TTL
                    )
        NS 147.91.1.5
        NS 147.91.1.7
        MX 147.91.79.3
        A 147.91.79.3

etf       IN    NS ns.etf.bg.ac.rs.
ns.etf    IN    A 147.91.8.6

proxy-web IN    A 147.91.14.227
www       IN    CNAME proxy-web
```

KOLOKVIJUM

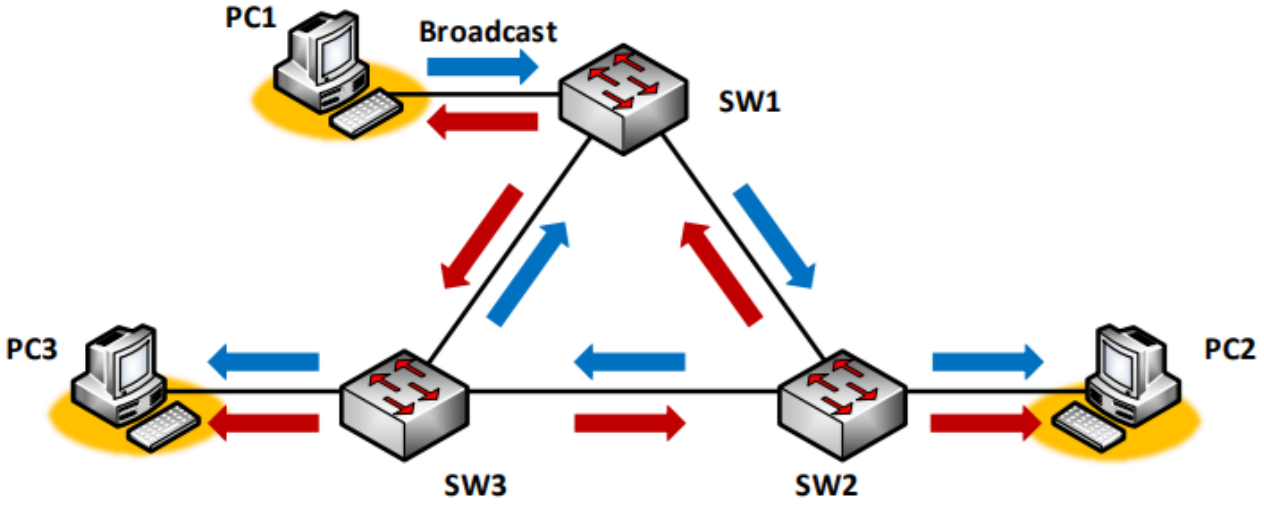
Ethernet tehnologije i protokoli

Ethernet, pored činjenice da je najkorišćeniji i dosta efikasan protokol, ima i određena ograničenja. Problem skalabilnosti rešavao se bridževima i svičevima. Problem otpornosti bas topologije rešavao se uvođenjem zvezdaste topologije. Problem otpornosti zvezdaste topologije (npr. prekid veze ili sviča) može se rešiti redudantnom topologijom, tako što se uvede još jedan svič. Ovde nastaje problem jer original Ethernet ne predviđa i ne dozvoljava petlje. Ripiteri i habovi samo reemituju okvire na izlazne portove i u slučaju petlji paketi će beskonačno da se vrte. Bridževi i svičevi su pametniji, ali i dalje ne prepoznaju petlje. Problemi koji mogu nastati:

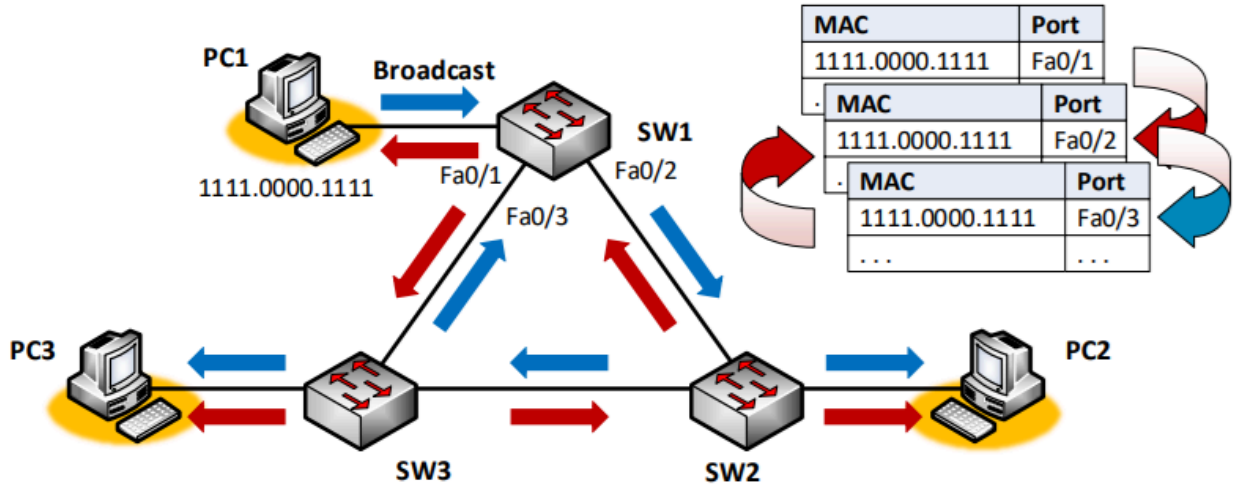
- **dupliranje pristiglih okvira** - prilikom flooding-a paketi do destinacije mogu doći na dva načina, odnosno dolaze dva ista paketa umesto jednog.



- **broadcast storm** - prilikom broadcasta paketi beskonačno kruže (u oba smera) i prijem je višestruk.



- **nestabilnost bridžing tabela** - neprestano kruženje u oba pravca dovodi do pogrešnih vrednosti u bridžing tabelama pri čemu se stalno menja kako stižu novi paketi.



Rešenje je u uklanjanju petlji, ali tako da se zadrži potpuna povezanost. **Spanning-Tree Protocol (STP)** je protokol koji uklanja petlje u Ethernet mrežama. Parametri protokola:

- **Identifikacija sviča/bridža (Bridge ID)** - sastoji se od dva polja: Bridge Priority (2B) preko kojeg se može forsirati značaj sviča i MAC (6B) koji predstavlja MAC adresu sviča.
- **Cena porta (Port Cost)** - celobrojna vrednost obrnuto proporcionalna brzini prenosa na portu (inicijalno $\frac{1000\ Mbps}{bandwidth}$). Vrednost je ista za uparene portove na ptp vezama i deljenim segmentima.
- **Cena putanje (Path Cost)** - suma cena portova na putu od izvora do odredišta. Računa se samo jednom po vezi. Određuje metriku putanja - najbolja putanja ima najmanju cenu.

Svičevi međusobno komunicirau putem **BPDU (Bridge Protocol Data Unit) poruka** koje prenose STP informacije između susednih svičeva. BPDU poruke se enkapsuliraju unutar Ethernet okvira, tako da STP predstavlja protokol L3 nivoa. Izvorišna adresa u tom paketu je port sviča koji šalje okvir, a odredišna adresa je fiksna multikast MAC adresa. Samo STP svičevi prihvataju ove pakete, a ostali uređaji ih odbacuju. Postoje tri vrste BPDU poruka: **Configuration BPDU**, **TNC (Topology Change Notification)** i **TCA (Topology Change Acknowledgment)**. Između ostalog, sadrže ID root sviča, kao i ID sviča koji je pošiljalac i Path Cost od root sviča do tog sviča. STP protokol odvija se u četiri faze:

1. **Izbor root sviča** - bira se svič sa najmanjim ID-em. Prvobitno svaki svič imenuje sam sebe za root svič. Nakon toga šalju Configuration BPDU poruku sa svojim ID-em u polju za root ID. Kada neki svič dobije poruku sa root ID-em koji je manji od njegovog počinje da šalje poruke sa tim novim ID-em, a inače nastavlja da šalje svoj ID. Na kraju su svi svičevi usaglašeni i u polju root ID imaju informaciju o stvarnom root sviču (onom koji među njima ima najmanji ID).

2. **Izbor root portova (RP)** - RP je port na sviču od koga vodi putanja sa najnižom cenom do root sviča. Samo jedan port na sviču može biti RP. Nakon izbora root sviča, on nastavlja da šalje Configuration BPDU poruke, a ostali svičevi ih samo primaju i sada posmatraju polje Path Cost. Pri prijemu poruke na Path Cost dodaju Port Cost i reemituju BPDU sa novom vrednošću Path Cost susednim svičevima. Ako do nekog sviča može da se dođe iz više putanja, bira se ona putanja koja ima najnižu cenu i odgovarajući port se proglašava za root port. Ako najmanju vrednost putanje ima više portova, bira se onaj koji je dobio Configuration BPDU poruku od sviča sa manjim ID-em. Ako postoje dve paralelne veze sa svičem i obe imaju istu vrednost Path Cost bira se onaj sa manjim internim rednim brojem. Root svič nema RP. RP uvek pripada STP stablu.
3. **Izbor designated portova (DP)** - DP je port na segmentu od koga vodi putanja sa najnižom cenom do root sviča. Samo jedan port na segmentu može biti DP. DP na segmentu ostvaruje najbolju putanju prema root sviču. Svič ne zna da li je na njega direktno povezan drugi svič ili se nalazi hab sa krajnjim uređajima i drugim svičevima. Ako više svičeva ima istu vrednost Path Cost bira se port čiji svič ima manji ID. Svi portovi root sviča su DP. DP može, a i ne mora da pripada STP stablu.
4. **Blokiranje preostalih portova** - RP i DP se stavljaju u **Forwarding stanje**, tj. prosleđuju okvire, a ostali portovi se stavljaju u **Blocking stanje**. Svičevi ne prosleđuju okvire kroz blokirane portove, a u ulaznom smeru primaju samo BPDU poruke, dok ostale okvire odbacuju.

U stacionarnom stanju root svič emituje Configuration BPDU na svake 2 sekunde - **Hello tajmer**. Ostali svičevi na svoje RP primaju Configuration BPDU poruke, a na svoje DP reemituju BPDU poruku pri čemu menjaju ID pošiljaoca i Path Cost. Usled promena u topologiji (prekid ili dodavanje veza) menja se stanje STP i dolazi do **STP konvergencije**. Osnovni problem je na koji način sprečiti privremene petlje tokom konvergencije. Prelazak iz Forwarding u Blocking stanje je momentalan, ali prelazak u suprotnom smeru mora ići postepeno kako ne bi došlo do nastajanja petlji tokom konvergencije. Pored Hello tajmera, za konvergenciju su bitni i **Max Age tajmer** koji iznosi $10 \cdot$ Hello tajmer i predstavlja vreme čekanja do pokretanja novog procesa uspostavljanja STP topologije u slučaju da svič više ne prima BPDU poruke, kao i **Forward Delay tajmer** koji obično iznosi 15 sekundi i predstavlja dodatno vreme čekanja kako bi se sve informacije propagirale u sve delove mreže. Tajmeri moraju biti usaglašeni u celoj mreži. Pri prelasku iz Blocking stanja u Forwarding stanje se onda uvode dva nova stanja:

1. **Listening stanje** - traje kao Forward Delay tajmer. Podaci se i dalje ne prenose, ali počinju da se šalju BPDU paketi iz prethodno blokiranog porta. Takođe se računaju STP parametri - cena, RP, DP.
2. **Learning stanje** - traje kao Forward Delay tajmer. U ovom stanju se prihvataju i okviri sa podacima, ali se oni ne prosleđuju dalje. Svič počinje da uči MAC adrese i formira bridžing tabelu.

Primeri promena topologije:

- **prekid na vezama krajnjih uređaja (access links)** - svičevi sve portove tretiraju na isti način i štite od petlji, pa i prekid veze sa nekim računarom dovodi do promene u topologiji. Svič koji detektuje promenu šalje TCN BPDU poruku na RP, koja se prenosi do root sviča. Root svič prima tu poruku i šalje Configuration BPDU sa TCN flagom. Svi svičevi detektuju TCN flag i preračunavaju stanje portova. Ujedno smanjuju Aging vreme u bridžing tabelama na 15 sekundi. Ovde nema konvergencije jer su stanja između svičeva nepromenjena.
- **prekid na direktnim vezama između svičeva** - svičevi kod kojih se desio prekid detektuju promenu i svič čiji RP nije u prekidu šalje TCN BPDU poruku koja se prenosi do root sviča. Root svič prima tu poruku i šalje Configuration BPDU sa TCN flagom. Svi svičevi detektuju TCN flag i preračunavaju stanje portova. Ujedno smanjuju Aging vreme u bridžing tabelama na 15 sekundi. Svič čiji je RP u prekidu detektuje bolji put i prelazi u Listening i Learning stanje, nakon čega se vraća u Forwarding stanje. Konvergencija traje 15 + 15 = 30 sekundi.
- **prekid na indirektnim vezama između svičeva** - svičevi mogu biti povezani i preko L1 uređaja (npr. hab) i u tom slučaju prekid u vezi ne može da se detektuje na portovima svičeva. Po isteku Max Age tajmera svič čiji je RP u prekidu nalazi bolji put i prelazi u Listening i Learning stanje, nakon čega se vraća u Forwarding stanje. Konvergencija traje 20 + 15 + 15 = 50 sekundi.
- **dodavanje nove veze** - nakon što je nova veza dodata neki od svičeva će možda detektovati bolju putanju. U tom slučaju se stari port stavlja u blokirano stanje, a novi port prelazi u Listening i Learning stanje, nakon čega prelazi u Forwarding stanje. Konvergencija traje 15 + 15 = 30 sekundi.

EtherChannel predstavlja više paralelnih veza između dva sviča, pri čemu se i dalje zadržava jedna logička veza. Na ovaj način se povećava kapacitet i otpornost na otkaz veze. Da bi se pokrenula STP konvergencija morale bi da otkažu sve veze. STP se inicijalno sprovodi na svim portovima, čak i na onima na koje su povezani drugi uređaji a ne svičevi. Zbog toga uređaj može da čeka i do 30 sekundi dok se priključi na mrežu. **PortFast** je opcija za port koja znači da se neće generisati TCN BPDU poruke i da taj port može odmah pređi u Forwarding stanje. Ako se na taj port ipak poveže svič, prilikom STP konvergencije će biti preskočena Listening i Learning stanja što dovodi do mogućnosti nastajanja petlji.

Jedan od načina na koji se može eksploatisati STP protokol je ako bi napadač povezao laptop na dva sviča. Laptop ima funkciju sviča pa bi postavljanjem prioriteta mogao da postane root svič i u tom slučaju bi mogao da prisluškuje saobraćaj. Za zaštitu se mogu koristiti:

- **BPDU Guard** - na pristupnim (access) portovima za uređaje se ne očekuju BPDU okviri. Ako neki takav port primi BPDU okvir na njega se postavlja zabrana. Aktivira se tek kada port prestane da prima BPDU okvire. Primenjuje se u paru sa PortFast tehnikom.
- **Root Guard** - na portu se dozvoljava prijem BPDU okvira, ali se zabranjuje prijem BPDU okvira sa boljom Bridge ID vrednošću. Na ovaj način se sprečava da se na tom portu javi bolji kandidat za root svič.

Osnovni problemi STP protokola su spora konvergencija, nedostatak load balancing-a, kao i često neoptimalne putanje saobraćaja ako se ne postavi pravi prioritet. **Rapid Spanning-Tree Protocol (RSTP)** je unapređena verzija STP protokola koja ubrzava konvergenciju. Uvode se različiti tipovi veza: **edge type** za pristupne veze sa uređajima i **link type** za veze sa drugim svičevima. Link type veze mogu biti **Point-to-Point** ako su direktne ili **Shared** ako su preko drugog uređaja (haba). Port cost se sada računa kao $\frac{20\ Tbps}{bandwidth}$ kako bi se uračunale i veće brzine. RP i DP se biraju na isti način kao kod STP, a uvode se i nove vrste portova:

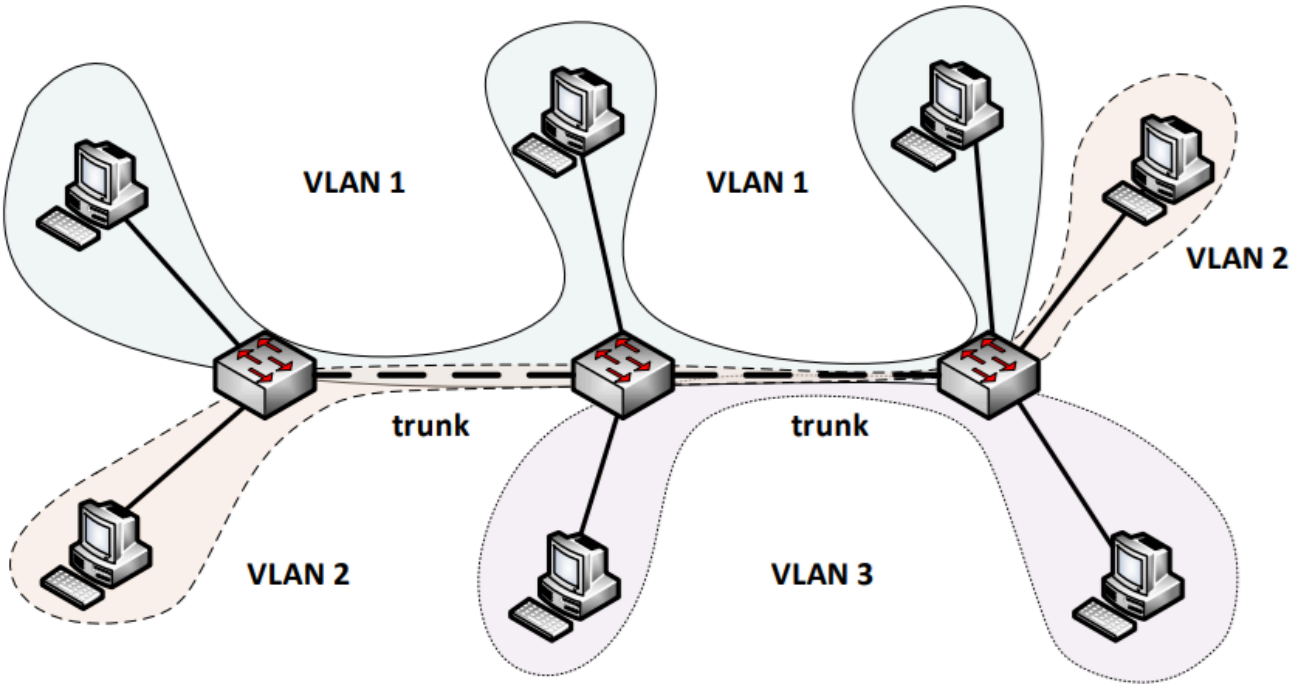
- **Alternate port** - blokiran port koji je najbolji posle RP i služi kao njegova zamena u slučaju da RP prestane da dobija BPDU sa najmanjim Path Cost.
- **Backup port** - blokiran port koji je najbolji posle DP i služi kao njegova zamena u slučaju kada je svič povezan sa više veza na hab.

RTSP se različito ponaša u zavisnosti od vrste veze. Na edge type vezama koristi PortFast mehanizam. Na link type shared vezama se ponaša isto kao STP. Kod link type ptp veza dolazi do unapređenja. Max Age tajmer sada koristi vrednost $3 \cdot$ Hello tajmer, a svičevi razmenjuju nove vrste BPDU poruka:

- **Proposal** - BPDU sa setovanim Proposal flagom.
- **Agreement** - BPDU kojom se potvrđuje grana u stablu. Šalje je RP.

Listening stanje se ne koristi, a Learning stanje je kratkotrajno, pa ceo proces obično traje manje od sekunde. RSTP funkcioniše tako što se prilikom promene u topologiji šalje BPDU sa Proposal flagom. Ako svič detektuje bolju vezu odgovara sa BPDU porukom sa setovanim Acknowledgment flagom. Postupak se propagira kroz mrežu na isti način.

LAN mreže sa svičevima omogućavaju jedinstvenu mrežu na L2 nivou. Grupe korisnika su najčešće povezane na jedan svič u okviru sprata, a spratovi su povezani na jedan glavni svič. S obzirom da je cela mreža na L2 nivou postoji jedinstven broadcast domen, pa paketi iz jedne grupe korisnika mogu dospeti do druge grupe koja ne treba da ima pristup. Jedno moguće rešenje je da svičevi budu povezani na ruter, umesto na glavni svič. Problem u ovom slučaju nastaje jer bi onda grupe korisnika morale biti fizički grupisane (npr. svi na istom spratu) što često nije moguće. **Virtual Local Area Network (VLAN)** logički deli fizičku LAN mrežu na nezavisne logičke LAN mreže koje funkcionalno postaju kao fizičke LAN mreže. Konfigurisanje VLAN mreža se vrši na svičevima, softverski. Odgovarajući port sviča se postavlja u odgovarajući LAN. Na osnovu nekog parametra paketa saobraćaj se onda svrstava u određeni VLAN. Dobija se veća fleksibilnost, skalabilnost i sigurnost po cenu malo veće složenosti i više administrativnog rada. Problem nastaje kada su VLAN mreže rasprostranjene na više svičeva. Klasično rešenje bi podrazumevalo da za svaki VLAN imamo posebne veze što je skupo i neskalabilno. Umesto toga, koristi se zajednička veza za sve VLAN mreže - **trunk link**.



Da bi se na trank vezama razlikovali VLAN-ovi u Ethernet okvire se dodaje zaglavlje od 4B koje identifikuje tu VLAN mrežu. Ovaj postupak naziva se **VLAN Frame Tagging**. Na izlasku sa trank veza skida se VLAN tag. Portovi sviča mogu biti:

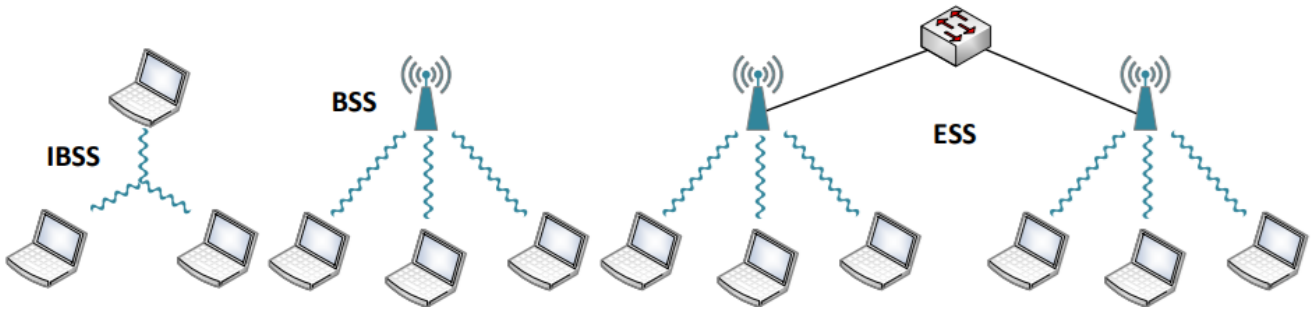
- **access port** - na vezama koje prenose samo jedan VLAN.
- **trunk port** - na trank vezama koje prenose više VLAN-ova.

Vrste portova moraju da se poklope na obe strane jedne veze. Povezivanje različitih LAN-ova (pa i VLAN-ova) mora se obaviti preko rutera, tj. L3 nivoa. Ako paket putuje između dva uređaja iz iste VLAN mreže on ide preko sviča, a ako putuje između dve VLAN mreže onda mora da ide preko rutera. Kad je u pitanju STP protokol, moguć je klasičan pristup gde postoji zajedničko STP stablo za sve VLAN-ove. Blokirani portovi se ne koriste ni za jedan VLAN, što je neoptimalno. Noviji pristup podrazumeva da se pravi STP stablo za svaki VLAN, što se naziva **Per-VLAN STP (PVST)**. Na ovaj način se dobija optimalnije iskorišćenje fizičkih linkova.

Bežične LAN mreže

Bežične LAN mreže (Wireless LAN, WLAN) koriste deljeni medijum, odnosno jednu frekvenciju i rade u half-duplex modu, tj. samo jedan uređaj može da šalje okvire u jednom trenutku. Tokom slanja podataka, prijem podataka je isključen pa se kolizija ne može detektovati kao kod Etherneta. Takođe, zbog slabljenja signala ne može se garantovati da će svi uređaji da detektuju koliziju (**"hidden station"**). Umesto detektovanja, vrši se izbegavanje kolizije - **Carries-Sense Multiple Access/Collision Avoidance (CSMA/CA)**. Zahteva se slanje potvrde za uspešan prijem svakog okvira. WLAN mreže mogu biti:

- **Independent Basic Service Set (IBSS)** - svi učesnici su ravnopravni.
- **Basic Service Set (BSS)** - postoji centralni uređaj koji se naziva **Access Point (AP)** i sva komunikacija se obavlja preko njega.
- **Extended Service Set (ESS)** - više AP povezanih preko sviča.



IBSS predstavlja ad-hoc režim, a BSS i ESS infrastrukturne režime. U oba slučaja "Service Set" označava grupu povezanih uređaja, odnosno WLAN mrežu. **WLAN ćelija** predstavlja oblast dometa signala jednog AP. Refleksija od objekata u okruženju stvara izobličavanje signala. Problem se rešava tako što se na AP postavljaju dve antene razmaknute za polovinu talasne dužine. **Service Set Identifier (SSID)** predstavlja naziv WLAN mreže. **Beacon okvir** je okvir kojim se AP periodično oglašava. Sadrži SSID i MAC adresu AP. Mreža na ovaj način postaje vidljiva za ostale uređaje u dometu. Povezivanje se odvija u tri faze:

1. **Razmena parametara** - vrši se usaglašavanje podržanih standarda, frekvencije, brzine prenosa i slično. AP šalje **Probe Request**, a uređaj odgovara sa **Probe Response**.
2. **Autentifikacija** - uređaj može da se autentifikuje i kroz više koraka. AP šalje **Authentication Request**, a uređaj odgovara sa **Authentication Response**.
3. **Učlanjivanje** - AP šalje **Association Request**, a uređaj odgovara sa **Association Response**.

Sva komunikacija u WLAN mreži odvija se preko AP. Uređaji mogu da detektuju sve okvire, ali prihvataju samo okvire od AP. AP je i rešenje za "hidden station" problem jer omogućava komunikaciju uređaja koji su međusobno van dometa. Komunikacija sa ostalim uređajima u LAN mreži vrši se tako što se AP povezuje na svič. Moguća je i integracija sa VLAN mrežom tako što se SSID mapira u VLAN. Takođe je moguće i da više SSID-a na jednom AP uređaju bude mapirano u odvojene VLAN-ove koristeći trank sa svičem. AP zahteva napajanje, a obično je lociran pri plafonu gde je nezgodno dovesti napajanje. Kao rešenje mogu se koristiti posebni svičevi koji prenose napajanje od 12V preko UTP kablova, pa se AP može napajati preko sviča putem Ethernet priključka - **Power Over Ethernet (PoE)**.

WLAN kanal predstavlja frekvencijski domen fiksne širine, najčešće 20MHz. Služi za prenos jednog signala. Susedni kanali su razdvojeni za 5MHz. AP podržava više kanala, ali radi samo na jednom. Da bi se obezbedilo pokrivanje većeg prostora, koristi se više AP uređaja čije ćelije moraju da se preklapaju da bi se obezbedio kontinuitet WLAN mreže. Problem nastaje ako preklopljene ćelije rade na susednim kanalima jer može doći do preklapanja frekvencijskog domena i interferencije signala i grešaka. Da bi se problem izbegao susedni AP čije se ćelije preklapaju moraju da koriste različite, međusobno udaljene kanale, odvojene za najmanje 5 kanala. U slučaju da signal sa jednog AP oslabi, automatski se prelazi na AP sa jačim signalom. Signal se traži skeniranjem kanala. **Pasivno skeniranje** znači da uređaj čeka da AP oglasi beacon okvir koji sadrži kanal na kome taj AP radi. **Aktivno skeniranje** podrazumeva da uređaj šalje upit za raspoložive kanale - **Request Probe paket**.

Ctrl 2B	Dur 2B	Adr1 6B	Adr2 6B	Adr3 6B	Seq 2B	Adr4 6B	Data 0-2312B	FCS 4B

Format **WLAN okvira** razlikuje se od Ethernet okvira i sadrži sledeća polja:

- **Frame Control (Ctrl)** - koristi se za različite flagove.
- **Frame Check Sequence (FCS)** - za kontrolu greške (CRC).
- koriste se 4 polja za adrese: **Source Address** za izvorišni uređaj, **Transmitter Address** za izvorišni uređaj ili AP, **Receiver Address** za odredišni uređaj ili AP i **Destination Address** za odredišni uređaj. Korišćenje ovih polja zavisi od konkretnog slučaja koji se određuju sa dva flaga - **To DS (Distribution System)** i **From DS**.
- **Duration** - koristi se za procenu vremena zauzeća medijuma. To vreme se naziva **Network Allocation Vector (NAV)** i zavisi od veličine okvira i brzine prenosa okvira. Uređaji fizički osluškuju medijum i gledaju da li je slobodan iil zauzet. Na osnovu NAV vremena znaju koliko će medijum biti zauzet i ne moraju stalno da proveravaju čime se smanjuje aktivnost uređaja i potrošnja baterije.

Izbegavanje kolizije (CSMA/CA) može se vršiti na dva načina:

- **Centralizovana koordinacija (Point Coordination Function - PCF)** - jedan centralni uređaj (AP) proziva ostale uređaje i daje im dozvolu za slanje. Može da se koristi samo kod infrastrukturnih mreža.
- **Distribuirana koordinacija (Distributed Coordination Function - DCF)** - svi uređaji, uključujući i AP, su jednaki i nadmeću se za zauzimanje medijuma. Može da se koristi i kod infrastruktunih i kod ad-hoc mreža. Ovaj pristup se najčešće koristi. Uređaji osluškuju medijum i ako je on slobodan čekaju fiksni vremenski interval **DIFS (Distributed Inter Frame Space)** nakon čega šalju okvir. DIFS obično traje $50\mu s$. Ako je medijum zauzet prvo se čeka DIFS, a potom se čeka i slučajan vremenski interval koji se naziva **back-off**. On iznosi $R \cdot ST$, gde je R slučajan broj iz intervala od 0 do **CW (Contention Window)**, a ST **slot-time** koji iznosi $20\mu s$. Sa brojem neuspešnih pokušaja eksponencijalno se povećava CW. Ako medijum postane zauzet tokom čekanja back-off vremena ono se pauzira. Kada se medijum opet oslobodi čeka se DIFS, a onda preostalo back-off vreme nastavlja da teče. Izbegavanje kolizije se sprovodi, ali se ne garantuje. Svaki okvir koji se uspešno primi bez kolizije mora da se potvrdi **Positive Acknowledgment (ACK) okvirom**. Ako nastane kolizija, uređaj koji je poslao okvir neće primiti potvrdu, pa se okvir ponovo šalje.

Postoje dva režima prenosa okvira od uređaja A do uređaja B:

- **Prenos u dva koraka (Two-Way Handshake)**:
  1. $(A \rightarrow B)$ okvir sa podacima
  2. $(B \rightarrow A)$ potvrda (ACK)
- **Prenos u četiri koraka (Four-Way Handshake)**:
  1. $(A \rightarrow B)$ zahtev za prenos (**RTS - Request To Send**)
  2. $(B \rightarrow A)$ odobravanje prenosa (**CTS - Clear To Send**)
  3. $(A \rightarrow B)$ okvir sa podacima
  4. $(B \rightarrow A)$ potvrda (ACK)

U oba slučaja, između svaka dva koraka čeka se **SIFS (Short Inter Frame Space)**. SIFS predstavlja vreme čekanja da pristigne ACK okvir i iznosi $10\mu s$. U oba slučaja, NAV vreme se rezerviše za celokupno trajanje prenosa. Prenos u četiri koraka traje duže, ali je bezbedniji i koristi se kod slanja velikih okvira.

Poseban problem WLAN mreža je sigurnost jer svi članovi mreže mogu da čitaju pakete deljenog medijuma i bez fizičkog pristupa objektu. Rešenja su:

- **WEP (Wired Equivalent Privacy)** - uvodi se šifrovanje paketa statičkim simetričnim ključem. Problem je što je svaki korisnik morao ručno da podešava ključ i što je bio nedovoljne dužine pa je lako mogao da se provali brute force napadom. Sami proizvođači su imali specifična rešenja, kao što su sakrivanje SSID naziva i filtriranje korisnika po MAC adresama.
- **WPA (Wi-Fi Protected Access)** - Wi-Fi je udruženje proivođača wireless opreme koje je uvelo nove WLAN standarde. Ključevi su sada bili dinamički, uvedena je autentifikacija korisnika, pristupni ključ i priključivanje korisnika preko username-a i šifre.
- **WPA2 (Wi-Fi Protected Acces 2)** - unapređen je algoritam šifrovanja i povećana dužina ključa. Preporuka je za korišćenje u modernim WLAN mrežama.

WAN tehnologije

Wide Area Network (WAN) su mreže na većem prostoru kao što su gradovi, regioni i države. U pitanju su mreže telekomunikacionih provajdera, a korisnici iznajmljuju servise određenog kapaciteta. Pretežno se koriste L3 uređaji - ruteri i L3 svičevi. Mogu biti:

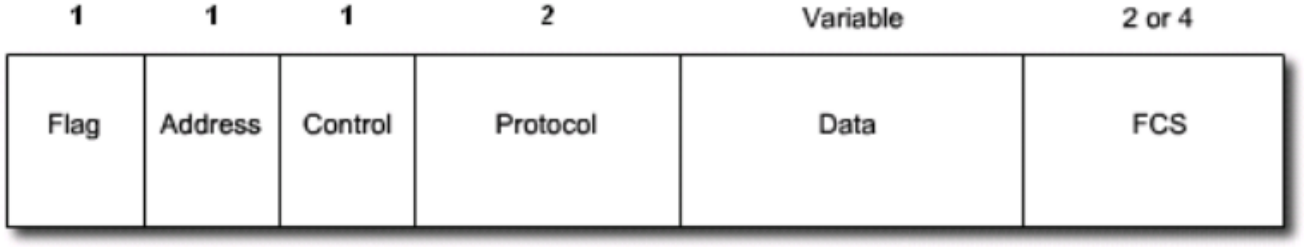
- **Circuit Switched** - povezivanje se vrši preko iznajmljenih telefonskih linija. Postoji fiksna fizička veza između dve lokacije. Brzine prenosa su male, instalacije skupe i ceo pristup je nefleksibilan i neskalabilan.
- **Packet Switched** - povezivanje se vrši preko iznajmljenih logičkih veza. Vrši se svičovanje paketa. Korisnik se povezuje na najbližu tačku mreže, a provajder omogućava logičku vezu. Povezivanje korisnika i rad mreže provajdera su fleksibilni.

WAN veze na L1 nivou mogu biti **analogne** ili **digitalne**. Digitalni signali se modulišu i pretvaraju u analogne, i obrnuto, koristeći **modem**. Modemi mogu biti:

- **voiceband (uskopojasni)** - povezivali su se na telefonske linije. Rastojanje je bilo neograničeno, ali brzine male. Nule i jedinice su se pretvarale u zvuk.
- **broadband (širokopojasni)** - koriste modulaciju na višim frekvencijama. Primer su **DSL (Digital Subscriber Line) modemi**. Rastojanje je limitirano, zbog visokih frekvencija. Postoje i SHDSL (Symmetric DSL), ADSL (Asymmetric DSL) i VDSL (Very High Bitrate DSL). **Kablovski modemi** su omogućavali prenos preko koaksijalnog kabla kablovskog operatera.

Dakle, digitalni signal se prenosio do modema, a zatim se analogni signal prenosio do centralnog modema u gradu, odakle se signal opet pretvara u digitalni i ide dalje kroz mrežu. Postojali su i uređaji koji su omogućavali da digitalni signal dolazi direktno do korisnika, ali bili su jako skupi. To su bili **Channel Service Unit (CSU)** i **Data Service Unit (DSU)**. Obično su integrisani u jedan eksterni uređaj ili u karticu na ruteru. Digitalni uređaji na obe strane moraju da usaglase brzinu slanja i primanja podataka, kako bi mogli da komuniciraju. **Klok (clock)** je takt kojim se šalju signali. U komunikaciji, **master** definiše klok, a **slave** se prilagođava dobijenom taktu. U digitalnim servisima, provajder definiše takt, a CSU/DSU se prilagođavaju. Postoje i bežične veze koje prenose signale na velikim udaljenostima putem antena - **WiMax**.

PPP (Point-to-Point Protocol) je L2 protokol za prenos preko sinhrone i asinhrone serijske veze. **Sinhrona** serijska veza podrazumeva prenos digitalnih signala sa unapred usklađenim taktom, a **asinhrona** bez.



PPP okvir sadrži sledeća polja:

- **Flag** - označava početak i kraj svakog okvira i ima vrednost "01111110". U ostatku okvira ne sme da se dozvoli šest uzastopnih jedinica. Pošiljalac nakon svakih pet uzastopnih jedinica veštački dodaje nulu. Prijemnik nakon svakih primljenih pet uzastopnih jedinica ignoriše nulu ako je primi, a ako primi šestu jedinicu zna da se radi o Flag polju. Flag polje ujedno označava i kraj prethodnog i početak narednog okvira.
- **Address** - ne koristi se. Fiksna vrednost "11111111".
- **Control** - fiksna vrednost "00000011".
- **Protocol** - identifikacija protokola L3 nivoa.
- **Data** - enkapsulirani podaci višeg nivoa.
- **FCS** - provera greške

PPP sadrži dva podsloja - LCP i NCP. **Link Control Protocol (LCP)** - vrši uspostavljanje, održavanje i raskidanje veze, pregovaranje između učesnika, postavljanje kontrolnih opcija, usklađivanje različitih limita u veličinama okvira između dve strane, detekciju grešaka u konfiguraciji i na linku. LCP šalje tri vrste okvira:

- **Link-establishment frames** - okviri za uspostavljanje veze. Configure-Request inicijalizuje vezu, Configure-Ack prihvata vezu, a Configure-Nak ili Configure-Reject odbija vezu.
- **Link-maintenance frames** - okviri za održavanje veze. Šalje okvire Code-Reject i Protocol-Reject kad se ne prepoznaje LCP kod ili protokol, kao i okvire za testiranje veze kao što su Echo-Request, Echo-Reply, Discard-Request.

- **Link-termination frames** - okviri za raskidanje veze. Terminate-Request zahteva, a Terminate-Ack prihvata raskidanje veze.

LCP okvir sadrži sledeća polja:

- **Code** - tip LCP okvira.
- **Identifier** - za uparivanje request i reply okvira.
- **Length** - ukupna dužina LCP okvira.
- **Data** - podaci viših podslojeva i slojeva.

Opcione funkcije LCP sloja su i:

- **autentifikacija učesnika** - međusobna provera identiteta. **Password Authentication Protocol (PAP)** obavlja se u 2 koraka: klijent šalje lozinku u čistom tekstu, na početku uspostavljanja sesije, a druga strana prihvata ili odbija vezu. Ovaj pristup je nesiguran jer se prisluškivanjem linije lako može saznati lozinka. **Challenge Authentication Protocol (CHAP)** odvija se u 3 koraka:
  - **Challenge** - server šalje challenge, koji predstavlja kombinaciju vremena i slučajnih podataka.
  - **Response** - klijent dodaje lozinku i vraća vrednost izračunatu unapred poznatim algoritmom.
  - **Accept/Reject** - server računa istu vrednost na osnovu lokalnih podataka i poredi je sa dobijenom.
- **kompresija podataka** - koristi različite algoritme. **Predictor** predviđa sekvencu karaktera u nizu podataka. Koristi tabelu čestih sekvenci - **rečnik kompresije**. Prepoznati nizovi se zamenjuju sa indeksima u rečniku. **STAC** u ulaznom nizu podataka traži sekvence koje se ponavljaju. Pronađene sekvence se zamenjuju indeksima koji su kraći od originalnog niza karaktera. Problem u kompresiji nastaje kada se kompresuju već kompresovani podaci. Ponovna kompresije prouzrokuje duže podatke, umesto da ih skrati.
- **multi-link povezivanje** - mehanizam kojim se više fizičkih serijskih veza spajaju u jednu logičku. Svaki datagram se deli na delove fiksne veličine koji se naizmenično šalju preko svakog od serijskih linkova. Ukupan kapacitet veze se sumira.
- **detekcija grešaka** - koristi **LQM (Link Quality Monitoring)**. Na obe strane povremeno šalje poruke sa brojem ispravno primljenih paketa i bajtova, a na drugoj strani se ovi podaci porede sa ukupno poslatim, na osnovu čega se računa procenat grešaka u prenosu. U slučaju da je procenat grešaka veći od konfigurisanog, LCP prekida vezu. Ovaj pristup ima smisla samo ako postoje rezervne veze.
- **call-back podrška** - klijent inicira poziv, zahteva povratni poziv i prekida vezu, a server inicira novi poziv prema klijentu. Na ovaj način se povećava sigurnost, jer server može da odbije neželjenog klijenta.

Network Control Protocol (NCP) - predstavlja interfejs prema L3 nivou i vrši pregovaranje o konfiguraciji mrežnog sloja. Kada je NCP sesija uspostavljena, paketi između mrežnih slojeva mogu da se razmenjuju. Primer je **IPCP** koji razmenjuje IP adrese krajeva linka, IP adrese DNS servera i tako dalje.

Principi rutiranja i pomoćni protokoli

Ruteri su uređaji trećeg nivoa OSI modela koji povezuju različite IP mreže. Na osnovu **odredišne IP adrese** iz zaglavlja paketa, ruter određuje **izlazni interfejs** preko kog šalje paket ka sledećem koraku, poznatom kao **next-hop** - IP adresa susednog rutera na zajedničkom linku. Ruteri imaju svoje **interfejse** sa IP i MAC adresama, kao i simboličkim i numeričkim oznakama (npr. GigabitEthernet0/1). Rutiranje je **hop-by-hop** proces, gde svaki ruter samostalno odlučuje o sledećem koraku, na osnovu svoje **ruting tabele**. Ruting tabela sadrži parove (**rute**): IP mreža - next-hop adresa. Prilikom obrade paketa, ruter traži kojoj mreži pripada odredišna adresa. Ako postoji više mreža kojima pripada odredište, koristi **najspecifičniju rutu (longest prefix match)** - bira se najmanja mreža, tj. ona sa najdužim prefiksom (maskom). Postoje dva osnovna pristupa:

- **Destination-based rutiranje** - standardno, na osnovu odredišta.
- **Source-based rutiranje** - ređe, zasnovano na izvorišnoj adresi, koristi se u specijalnim slučajevima.

Kada host želi da komunicira van svoje mreže, koristi **default gateway** - unapred definisani izlaz ka drugim mrežama. Ako u ruting tabeli ne postoji konkretna ruta, koristi se **default ruta (0.0.0.0/0)** koja pokriva „sve ostalo“. Rute se dele na:

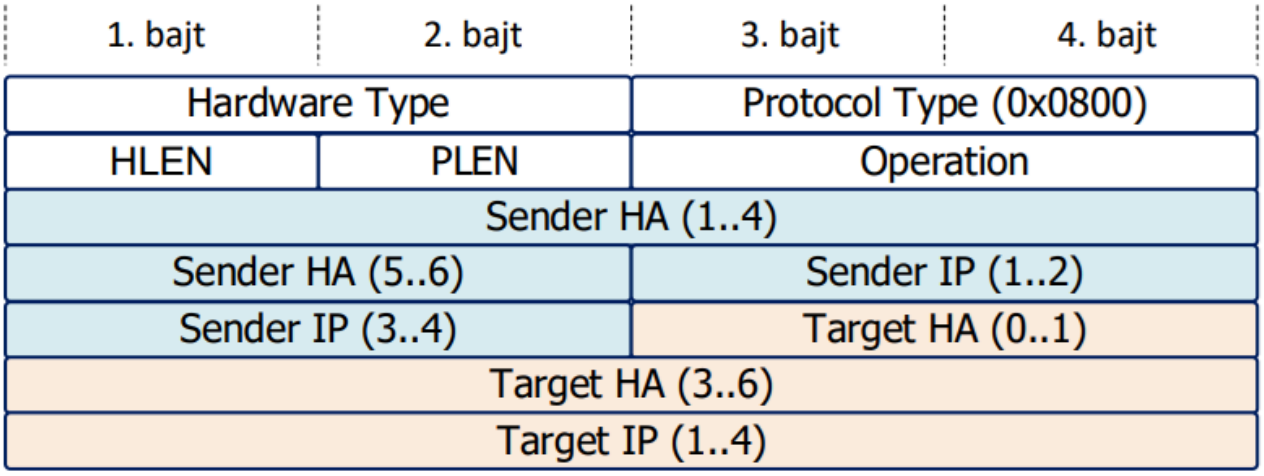
- **Statičke rute** - manuelno konfigurisane, jednostavne, ali neprilagodljive. Dodavanje samo jedne mreže zahteva rekonfiguraciju svih ruting tabela.
- **Dinamičke rute** - ruteri međusobno razmenjuju informacije koristeći **ruting protokole**, što omogućava skalabilnost i adaptivnost mreže.

U mrežama se koristi i **brodkast**. U tom slučaju odredišna adresa sadrži sve jedinice (npr. 10.20.50.255 u mreži 10.20.50.0/24). Brodkast IP paketi se enkapsuliraju u **L2 brodkast pakete**, koji takođe sadrže sve jedinice (FF:FF:FF:FF:FF:FF). **Brodkast domen** je jedan L2 segment mapiran u IP mrežu. Svi uređaji u istom **brodkast domenu** primaju brodkast poruke.

Ako se i izvorišni i odredišni uređaj nalaze u istoj mreži, paket se šalje direktno, tj. komunikacija se vrši u okviru iste LAN mreže. Da bi se L3 paket enkapuslirao u Ethernet okvir na L2 nivou, potrebno je da znamo izvorišne i odredišne IP adrese i MAC adrese. Izvorišna IP adresa je poznata iz lokalne konfiguracije, a izvorišna MAC adresa je poznata sa mrežne kartice. Odredišna IP adresa je poznata preko aplikacije koja zahteva komunikaciju, ali odredišna MAC adresa nije poznata. **ARP (Address Resolution Protocol)** je L3 protokol koji služi za povezivanje IP i MAC adresa u lokalnim mrežama. Kada uređaj zna IP adresu odredišta ali ne zna njegovu MAC adresu, šalje:

1. **ARP Request** - brodkast poruka: „Ko ima ovu IP adresu?“ Šalje se na **brodkast MAC adresu** (FF:FF:FF:FF:FF:FF). Paket sadrži IP i MAC adresu pošiljaoca i IP adresu uređaja za koji se traži MAC adresa. Ako niko ne odgovori, ARP javlja grešku.
2. **ARP Reply** - odgovor sa odgovarajućom MAC adresom od odredišnog uređaja. Šalje se na unikast MAC adresu uređaja koji je poslao ARP Request. Pošiljalac ažurira svoj keš (ARP tabelu) kada prima ARP Reply.

ARP tabela je keš koji privremeno čuva sve otkrivene MAC adrese, tj. sadrži parove IP adresa i MAC adresa. Red u tabeli se briše nakon nekog vremena (npr. Windows 2 minuta, povećava se na 10 ako se koristi opet u toku prva 2 minuta). ARP tabele mogu da se izlistaju ili obrišu (komande `show arp` i `clear arp` na ruterima). **ARP paket**:



Portovi rutera imaju MAC adrese koje se koriste u zaglavljima L2 okvira. Prilikom rutiranja paketu se u ruteru ne menjaju IP adrese, ali se menjaju MAC adrese.

ICMP (Internet Control Message Protocol) koristi se za kontrolne i dijagnostičke poruke u IP mreži. Enkapsulira se u IP paket. Tehnički je protokol transportnog nivoa, ali ne sprovodi funkcije tog nivoa. Postoje dve osnovne grupe:

- **Error message** - poruke o greškama.
- **Query message** - poruke uputa i odgovora na upite.

Vrste ICMP poruka:

- **Destination Unreachable** - kada paket ne može da se prenese, on se odbacuje, a izvorišni uređaj se obaveštava ovom porukom. Sadrži prvih 100B originalnog paketa, da bi uređaj prepoznao koji paket je odbačen. Podvrste:
  1. **Can’t Fragment** - IP paket je veći od MTU vrednosti u narednom L2 segmentu, a setovan je "Don't fragment" flag. Ruter obaveštava pošiljaoca.
  2. **Network Unreachable** - u ruting tabeli ne postoji mreža kojoj odgovara odredišna adresa. Ruter obaveštava pošiljaoca.
  3. **Host Unreachable** - IP adresa odredišta se ne nalazi u ARP kešu, a niko se ne odaziva na ARP Request. Ruter obaveštava pošiljaoca.
  4. **Protocol Unreachable** - paket je stigao do IP nivoa odredišnog uređaja, ali ne postoji protokol na L4 nivou koji je naznačen u zaglavlju IP paketa. Odredišni uređaj obaveštava pošiljaoca.
  5. **Port Unreachable** - paket je stigao do L4 nivoa odredišnog uređaja, ali ne postoji aplikacija koja je naznačena u zaglavlju poruke L4 nivoa. Odredišni uređaj obaveštava pošiljaoca.
- **Redirect** - kada je više rutera povezano na LAN mrežu. Default gateway obaveštava uređaj da za odredište postoji bolja specifičnija ruta.
- **Time Exceeded** - kada TTL vreme dostigne 0. Obično se javlja kad postoji petlja u rutiranju.
- **Echo Request, Echo Reply** - proverava dostupnost na IP nivou (**ping komanda**). Šalje se Echo Request na proizvoljnu IP adresu i čeka se Echo Reply poruka od tog uređaja.

Komande za dijagnostiku:

- **ping** - proverava dostupnost i kvalitet veze. Prikazuje se broj poslatih i primljenih paketa, procenat izgubljenih paketa, vreme između slanja i prijema paketa (Round Trip Time).
- **traceroute/tracert** - otkriva niz rutera do odredišta koristeći poruke sa TTL = 1, 2, 3 i tako dalje, a ruteri onda odgovaraju sa TTL.
- **route print / show ip route / route** - prikazuje ruting tabele na Windows, Cisco i Linux uređajima, redom.
- **ipconfig** - daje pregled mrežnih interfejsa i konfiguracija na Windows sistemima.

Protokoli rutiranja (Distance Vector)

Protokoli rutiranja ne obavljaju direktno rutiranje poruka, već omogućavaju ruterima da nauče kako da to rade - tj. uspostavljaju ruting tabele. Osnovni principi rada ruting protokola uključuju:

- Oglašavanje mreža koje su poznate ruteru kroz tzv. **routing update**.
- Prikupljanje informacija od drugih rutera.
- Izbor najbolje rute pomoću neke metrike, kada postoji više ruta do neke mreže.
- Ažuriranje ruta u slučaju promene topologije.

Protokoli rutiranja mogu biti:

1. **Interni** - unutar jednog autonomnog sistema. **Autonomni sistem (AS)** je jedinstveni administrativni domen računarske mreže (npr. akademska mreža). **Network Operation Center (NOC)** je centar za upravljanje mrežom. U okviru AS imamo usaglašeno rutiranje, kao i usaglašene konfiguracije rutera i pažljiv dizajn i upravljanje jedinstvenim adresnim prostorom. Interni protokoli mogu biti:
   - **Distance Vector**: **RIP**, IGRP
   - **Link State**: **OSPF**, IS-IS
   - **Hibridni**: EIGRP
2. **Eksterni** - između različitih autonomnih sistema (npr. Internet). Primer je BGP.

Cilj svih ruting protokola je:

- **Potpunost** - svi ruteri znaju sve mreže u ruting domenu.
- **Konzistentnost** - bez ruting petlji.
- **Optimalnost** - najbolja ruta prema metrici.
- **Adaptivnost** - automatsko prilagođavanje promenama topologije.

Classful protokoli ne prenose maske u rutama. Maske su podržane, ali su iste dužine u svim IP mrežama. Svi ruteri implicitno znaju maske na osnovu konfiguracije svojih interfejsa. Na vezi sa drugim ruting domenom sprovodi se **autosumarizacija** - automatska agregacija svih IP mreža. Sprovodi se na nivou mrežnog dela klase A, B i C, nezavisno od maske koja se koristi. **Classless protokoli** sadrže maske u rutama koje se razmenjuju između rutera. Dužina maske je promenljiva (VLSM). Agregiranje IP mreža je fleksibilno - prenošenje agregirane rute u drugi ruting domen. Dosta dugo se koriste isključivo classles ruting protokoli.

Metrika se koristi za izbor najbolje rute kada postoji više ruta do određene mreže. Posmatra se na nivou cele putanje do mreže. Moguće metrike su:

- **Hop count** - broj rutera do mreže.
- **Bandwidth (propusni opseg)** - izvedeno iz brzine veza.
- **Delay** - kašnjenje veza.
- **Load** - opterećenje veze.
- **Reliability** - pouzdanost veze.
- **Cost** - proizvoljna cena koja se definiše na neki način.

Kada postoji više ruta sa istom metrikom do jedne mreže, omogućava se **load balancing**, tj. **balansiranje saobraćaja** - paketi se šalju preko više veza, čime se povećava iskorišćenost ukupnog propusnog opsega. Kada se više ruting protokola koristi istovremeno, može doći do konflikta među rutama. **Administrativna distanca** predstavlja fiksne vrednosti za različite protokole i određuje prioritet rute u slučaju poređenja sa rutom drugog protokola. Niža vrednost znači veći prioritet.

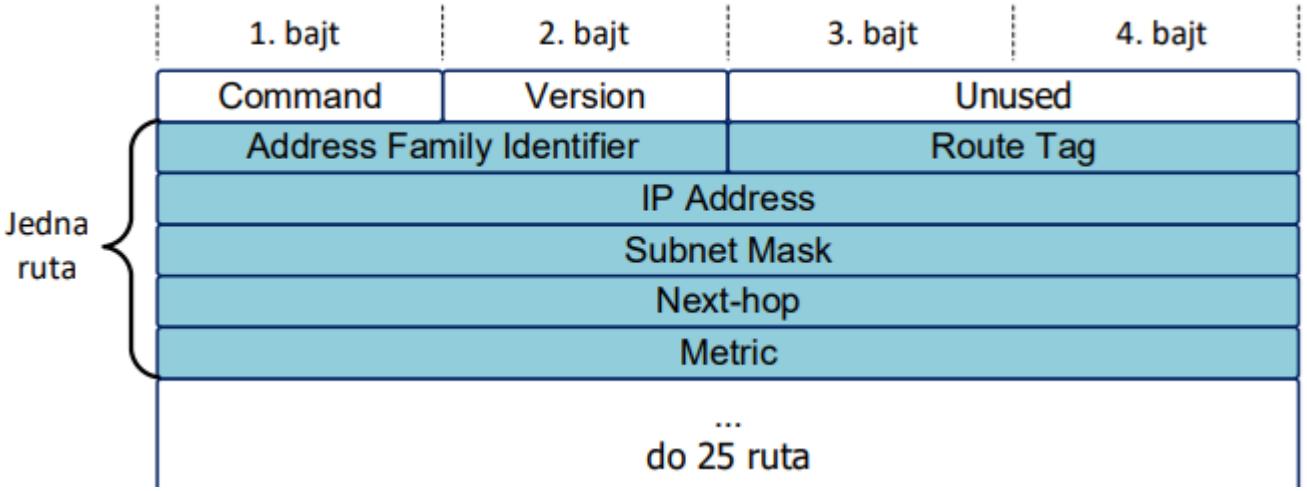
Kod **Distance Vector protokola**, ruteri znaju samo svoje susede i sa njima razmenjuju informacije (**ruting update**). Ne znaju celu topologiju i ostale rutere, već samo metriku i next-hop do određene mreže. Funkcionišu po sledećem principu - samo se najbolja ruta bira i upisuje u ruting tabelu, a u slučaju više najboljih ruta sve se upisuju. Kao rezultat, dobijamo popunjenu ruting tabelu za svaku IP mrežu u ruting domenu, ali i dalje se ne zna topologija i drugi detalji. Ruteri periodično oglašavaju rute iz svoje ruting tabele, čak i kada nema promena. Oglašavanje rute u jednom smeru, utiče na rutiranje ka oglašenoj mreži iz suprotnog smera. **Konvergencija** je proces uspostavljanja stabilnog i konzistentnog stanja na svim ruterima u mreži. Stabilno stanje znači da se ruting tabele više ne menjaju sa novim ruting update-ima, a konzistentno stanje da su sve rute ispravne. Konvergencija zavisi od brzine propagacije ruting update-a od rutera do rutera, kao i brzine računanja ruta i uspostavljanja ruting tabela. **Ruting petlje** nastaju usled neusaglašenih tabela tokom konvergencije. Kada nastane prekid ka nekoj mreži, ruter ga detektuje. Neki drugi ruter je imao informaciju o vezi ka toj mreži sa nekom metrikom. Prvi ruter sad prihvata tu rutu i povećava metriku za jedan, iako ta veza više ne postoji. Metrike se pri svakom ruting update-u povećavaju za jedan između ova dva rutera i tako do beskonačnosti. Ovaj problem naziva se **Count-to-Infinity**. Na nivou IP protokola, koristi se TTL za zaštitu od petlji. Na nivou ruting protokola koriste se naredne tehnike:

- **Route Poisoning** - oglašavanje da je mreža postala nedostupna. Mreža se oglašava sa beskonačnom metrikom i ruta se briše iz ruting tabele. Ostali ruteri upisuju tu rutu u ruting tabelu i imaju informaciju da je nevalidna. Oglašavanje se dešava na sledećem ruting update-u.
- **Triggered Update** - promene se oglašavaju odmah, ne čekajući sledeći ruting update. Oglašava se samo jedna ruta, a ne cela ruting tabela.
- **Split Horizon** - ruta se nikad ne oglašava interfejsom preko kojeg je primljena.
- **Poison Reverse** - nevalidna ruta se ipak oglašava kao nevažeća. Split Horizon se suspenduje samo za ovaj slučaj, a za oglašavanje se koristi Triggered Update.
- **Holddown Timer** - čeka se određeno vreme da bi se informacija o promeni propagirala do svih rutera. Kada ruter dobije Route Poisoning Triggered Update startuje se Holddown Timer i tokom tog vremena se ignorišu sve nove rute.

RIPv1 (Routing Information Protocol, version 1) je ruting protokol koji radi na aplikativnom nivou. RIP poruke se prenose unutar UDP paketa na L4 nivou. Protokol je Classful, a za metriku koristi hop-count sa maksimalnom vrednošću 16. Komunikacija se vrši u dva koraka, na svakih 30 sekundi:

1. **RIP Request** - navodi se mrežna adresa za koju se traže rute ili 0.0.0.0 za sve rute. Šalje se na brodkast adresu 255.255.255.255.
2. **RIP Response** - odgovor na upit. Do 25 ruta može da se dobije u jednoj poruci. Šalje se na unikast adresu rutera koji je poslao upit.

RIPv2 je ruting protokol koji je kompatibilan sa RIPv1, ali postoje razlike. RIPv2 je Classless pa ima podršku za VLSM. RIP Request se šalje na multikast adresu 224.0.0.9 koju slušaju svi RIPv2 ruteri. RIP Response se šalje na unikast adresu rutera koji je poslao upit, kao kod RIPv1. Sadrži i međusobnu **autentifikaciju** susednih uređaja, čime se povećava sigurnost. Može se koristiti zajednički ključ, a može i niz ključeva (**key chain**). Key chain sadrži više ključeva koji se identifikuju preko indeksa. Ruteri periodično menjaju indeks ključa koji se trenutno koristi. Ključevi se razmenjuju korišćenjem heš funkcija.



Format **RIP paketa**:

- **Command**: 1 za Request, 2 za Response.
- **Version**: 1 za RIPv1, 2 za RIPv2.
- **Address Family Identifier**: 2 za IP adrese.
- **IP Address** - adresa na koju se odnose rute ili 0.0.0.0 za sve.
- **Subnet Mask** - koristi se samo za RIPv2.
- **Next-hop** - koristi se samo za RIPv2.
- **Metric** - broj koraka, od 1 do 16.

Prednosti Distance Vector ruting protokola:

- Jednostavna implementacija, konfigurisanje i održavanje
- Nisu zahtevni u pogledu performansi
- Malo zauzeće linka za manje mreže

Nedostaci Distance Vector ruting protokola:

- Neadekvatna metrika
- Spora konvergencija, posebno za veće mreže
- Slaba skalabilnost
- Podložni ruting petljama

Protokoli rutiranja (Link State)

"Link" predstavlja interfejs rutera, a **"link-state"** su informacije o tim interfejsima, kao što su IP adresa i maska mreže, IP adresa interfejsa, ID rutera, tip interfejsa, cena linka, susedni ruteri na linku i slično. Kod **Link State protokola**, ruteri imaju punu informaciju o mrežnoj topologiji i razmenjuju informacije samo pri promenama. Razmenjuje se veća količina informacija - **LSA (Link-State Advertisements)**. Svaki ruter kreira sopstveni **LSA paket** koji opisuje stanje njegovih interfejsa i informacija o susedima. LSA se razmenjuju sa direktnim susedima, a zatim prosleđuju dalje - **flooding**. Tako svi ruteri dobijaju LSA svih ostalih rutere. Na osnovu njih svaki ruter formira sopstvenu bazu podataka **LSDB (Link-State Database)**. LSDB predstavlja skup svih LSA koje ruter prikupi. Sadrži informacije o kompletnoj mreži. Baza podataka je identična na svim ruterima jer se svi LSA razmenjuju. Na osnovu LSDB baze, svaki ruter kreira graf sa čvorovima (ruteri) i granama (linkovi). Svaka grana ima pridruženu cenu. Na osnovu te topologije, primenom **Dijkstrinog algoritma (SPF - Shortest Path First**), kreira se ruting tabela. Složenost je $O(n \log n)$. Rezultat SPF algoritma je ruting tabela u kojoj se za svaku mrežu nalazi next-hop i ukupna cena. Ako postoji više putanja sa istom cenom, koristi se load balancing. Svaki ruter ima svoju ruting tabelu.

Zbog mogućnosti preopterećenja, ruteri i mreže se grupišu u **oblasti (Area)**, koje imaju dva nivoa hijerarhije:

- **Centralna oblast (Backbone Area, Transit Area)** - označava se sa Area 0.
- **Periferne oblasti** - označavaju se sa Area n. Sve periferne oblasti se povezuju isključivo na centralnu oblast.

Pad linka u jednoj oblasti izaziva flooding samo unutar nje. Tako se postiže veća skalabilnost. Osobine Link-State protokola:

- Brza konvergencija - flooding samo po uključenju i pri promeni topologije.
- Zahtevaju više memorije zbog više informacija, više CPU vremena zbog SPF-a i više propusnog opsega zbog flooding-a.
- U stabilnom stanju, opterećenje je minimalno jer se prenose samo poruke za održavanje susedstva (**keepalive poruke**).

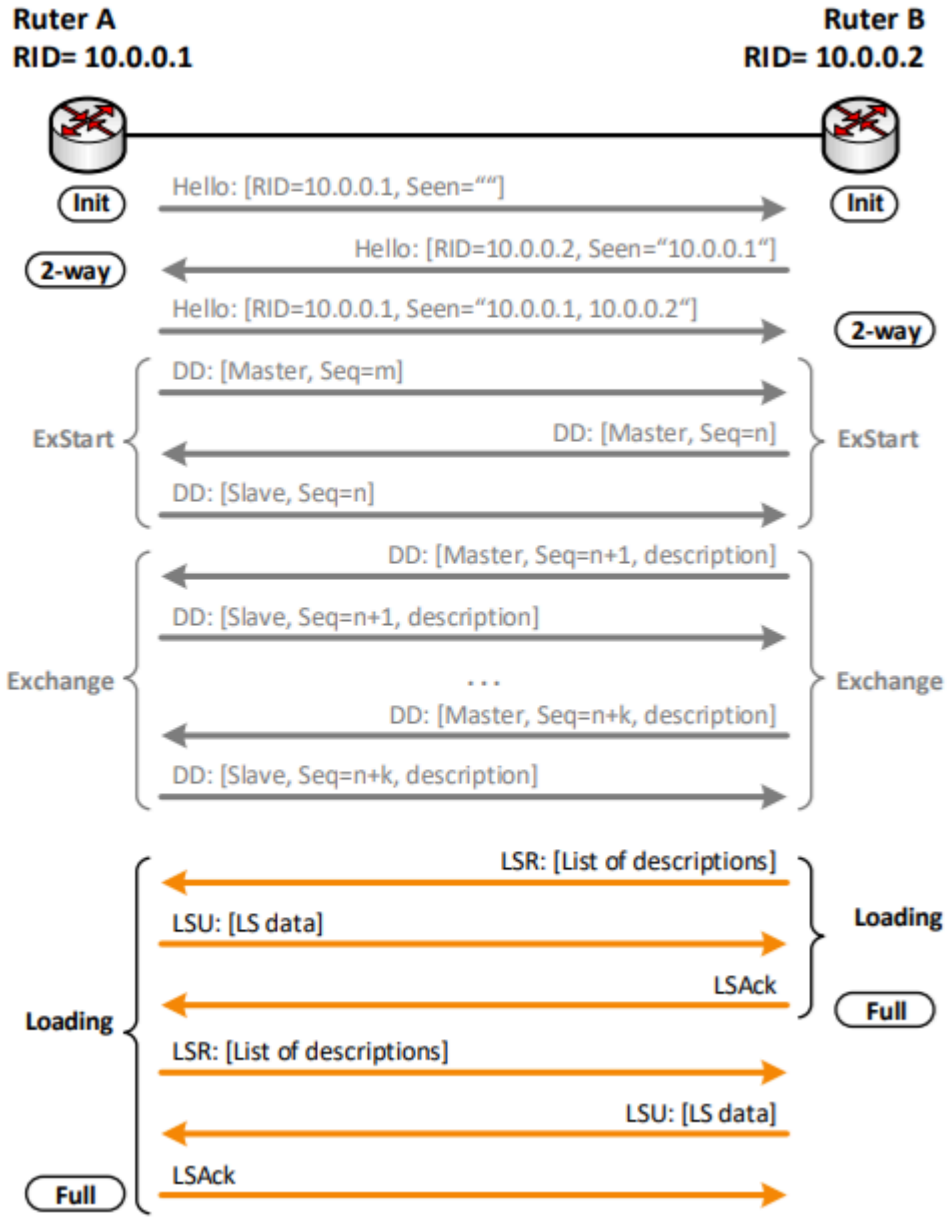
Postoje dve osnovne vrste Link-State protokola - **OSPF (Open Shortest Path First)** i IS-IS (Intermediate System to Intermediate System). **OSPF poruke** se enkapsuliraju u IP pakete, a za slanje se koristi multikast odredišna adresa. **Hello protokol** je protokol za uspostavljanje susedstva. Ruteri postaju susedi ako imaju iste sledeće parametre:

- mrežna adresa
- **Hello interval** - period oglašavanja Hello poruka. Uglavnom 10s.
- **Dead interval** - period nakon kog se prekida susedstvo. Uglavnom $4 \cdot$ hello interval.
- **Area ID** - broj oblasti kojoj pripadaju.
- Autentifikacija - ako se koristi.
- Ostali parametri - npr. stub area flag.

RID (Router ID) je jedinstveni identifikator rutera na nivou OSPF protokola. Može biti najveća IP adresa fizičkog interfejsa ili **loopback interfejsa**, ako postoji. Loopback interfejs je logički interfejs koji sadrži proizvoljnu IP adresu i masku (maska može biti čak i 32). On je uvek aktivan i zgodan je za pingovanje, logovanje i pristup ruteru generalno. Da bi se uspostavilo susedstvo putem Hello protokola prolazi se kroz sledeća stanja:

- **Down** - početno stanje.
- **Init** - nakon podizanja interfejsa. Ruter je spreman za slanje Hello poruka.
- **2-way** - uspostavljeno susedstvo uz uslove da su usaglašeni svi obavezni parametri i da se prepoznaje RID u "Seen" polju koje sadrži sve do tada otkrivane susedne rutere na tom segmentu.

Na dalje, ruteri mogu da ostanu u 2-way stanju. To su **Neighbour**, odnosno fizički susedi. Drugi slučaj je da ruteri nastave razmenu informaciju, kada se radi o OSPF susedima - **Adjecency**. **ExStart** proces predstavlja dogovor ko je Master, a ko Slave. Master je ruter koji ima veći RID. Koristi se Seq broj za razmenu vrednosti i postizanje dogovora. **Exchange** proces služi za razmenu LSA podataka iz LSDB tabela. Master počinje komunikaciju tako što šalje **Database Description (DD) pakete**. Šalje se opis podataka, bez potpunih informacija. U slučaju da se neki paket izgubi, šalje se ponovo. **Loading** je proces kojim se razmenjuju svi nedostajući podaci. Master šalje **Link-State Request (LSR) paket** gde navodi listu deskriptora nedostajućih podataka. Slave odgovara sa jednim ili više **Link-State Update (LSU) paketa** koji sadrže nedostajuće podatke. Koristi se i **Link-State Acknowledgement (LSAck) paket** za potvrdu prijema. Kada su LSDB tabele sinhronizovane šalje se **Full paket**. Ceo proces uspostavljanja susedstva:



Ethernet je multiaccess mreža i ne preporučuje se svako-sa-svakim susedstvo jer broj veza raste sa kvadratom broja rutera. Umesto toga, bira se:

- **DR (Designated Router)** - centralni ruter pri uspostavljanju susedstva.
- **BDR (Backup DR)** - rezervni centralni ruter.
- **DROthers** - ostali ruteri.

Samo DR i BDR uspostavljaju direktno susedstvo sa ostalim ruterima. Izbor za DR i BDR se vrši na osnovu prioriteta, a zatim prema RID-u. Prioritet je vrednost između 0 i 255 koja se dodeljuje interfejsu rutera. Veća vrednost označava veći prioritet, a vrednost 0 da ruter ne učestvuje u izboru. Prioritet se upisuje u Hello poruke. Dakle, ruter sa najvećim prioritetom postaje DR, a ruter sa sledećim najvećim prioritetom postaje BDR. Ako su prioriteti isti, gleda se najveći RID. Novododati ruteri neće izazvati promenu DR i BDR, bez obzira na njihove

prioritete ili IP adrese. DR i BDR će izgubiti funkcije jedino kada ruter prestane da radi, interfejs rutera prestane da radi ili OSPF na ruteru prestane da radi. Prenos SLA paketa se vrši na sledeći način:

- DROthers ruteri šalju LSA na multikast adresu 224.0.0.6 (ALLDRouters).
- DR i BDR slušaju saobraćaj na toj adresi i primaju LSA.
- DR prosleđuje LSA paket na multikast adresu 224.0.0.5 (ALLSPFRouters).
- Svi OSPF ruteri slušaju saobraćaj na toj adresi i primaju LSA.

Cena veze izvedena je iz propusnog opsega veze i iznosi $\frac{10^8}{bandwidth}$. Propusni opseg se definiše na interfejsu rutera i služi za određivanje cene veze, ne utiče na stvarnu brzinu veze. Manja cena ima veći prioritet. Cena putanje se dobija sabiranjem cena svih linkova. Za precizno rutiranje, treba definisati stvarnu brzinu ili konkretnu cenu interfejsa.

Vrste rutera prema mestu i ulozi u oblasti:

- **ABR (Area Border Router)** - granični ruter između oblasti.
- **ASBR (Autonomous System Boundary Router)** - granični ruter između OSPF domena i drugog ruting domena.
- **Internal Router** - unutar jedne oblasti.
- **Backbone Router** - unutar centralne oblasti.

Prema načinu oglašavanja unutar oblasti i prenošenju između oblasti, LSA mogu biti:

- **Router LSA (tip 1)** - generišu ih svi ruteri, daju informacije o svim interfejsima rutera i propagiraju se unutar jedne oblasti.
- **Network LSA (tip 2)** - generiše ih DR za oglašavanje Ethernet mreže prema ostalim ruterima unutar jedne oblasti.
- **Summary LSA (tip 3 i 4)** - tip 3 LSA su LSA u koje se pretvaraju tip 1 i tip 2 LSA kada dođu na ABR. Predstavljaju informacije o lokalnim linkovima i mrežama, koje ABR iz jedne oblasti prenosi kroz Area 0 i preko drugih ABR u druge oblasti. Na ovaj način se SPF algoritam ne preračunava u drugim oblastima. Tip 4 LSA su LSA koje oglašava ASBR ruter za svoje interfejse.
- **External LSA (tip 5)** - informacije o mrežama van OSPF domena, koje generiše ASBR. Postoje dve vrste:
  1. **O E1** - na metriku iz drugog ruting domena dodaje se OSPF metrika.
  2. **O E2** - na metriku iz drugog ruting domena ne dodaje se OSPF metrika.

Prema vrsti LSA paketa koji u njih ulaze, oblasti mogu biti:

- **Standard Area** - podržava sve vrste LSA. Backbone Area je uvek Standard Area.
- **Stub Area** - periferna oblast koja ne prima External LSA. Na svim ruterima u ovoj oblasti postavlja se Stub flag. ABR automatski generiše default rutu i ubacuje je u oblast, za saobraćaj prema odredištima van OSPF domena.
- **Totally Stubby Area** - periferna oblast koja ne prima ni External LSA ni Summary LSA. Na svim ruterima u ovoj oblasti postavlja se Stub flag. ABR automatski generiše default rutu i ubacuje je u oblast, za saobraćaj prema odredištima van OSPF domena, što su u ovom slučaju i druge oblasti, a i drugi ruting domeni.

Virtuelni linkovi omogućavaju logičku vezu sa Area 0, ako ne postoji fizička. Povezuju periferne oblasti sa backbone-om ili dva dela Area 0. **Redistribucija ruta** omogućava povezivanje različitih ruting protokola, kada je potrebno obezbediti IP konektivnost mreža iz različitih ruting domena. Jedan ruting domen učitava rute iz drugog i nastavlja da ih distribuira kao svoje rute. **Connected rute** se automatski uključuju ako su obuhvaćene konfiguracijom ruting protokola, a sve ostale rute zahtevaju manuelno konfigurisanje redistribucije iz jednog u drugi ruting domen.

Ostali protokoli i mrežne tehnologije

Svaki uređaj u IP mreži mora imati dodeljenu IP adresu, masku i default gateway. Postoje dva osnovna načina dodele IP adresa:

- **Statičko dodeljivanje** - IP adresa se ručno unosi na svakom uređaju. Povećava mogućnost grešaka (npr. dupliranje adresa, pogrešna maska). Nefleksibilno rešenje jer svaka promena zahteva ručne izmene na svakom uređaju. Nije pogodno za privremene korisnike (npr. WiFi, VPN).
- **Dinamičko dodeljivanje** - IP adresa se automatski dodeljuje iz definisanog opsega. Konfiguriše se centralno - na serveru. Manja je mogućnost grešaka, fleksibilnije rešenje. Podržava korisnike koji se privremeno povezuju.

RARP (Reverse Address Resolution Protocol) je prvi protokol za dodelu IP adresa. Protokol je L3 nivoa. RARP funkcioniše obrnuto od ARP-a, tj. umesto da se MAC adresa traži na osnovu IP adrese, koristi se MAC adresa da bi se pronašla odgovarajuća IP adresa. Namenjen je uređajima bez lokalne memorije (npr. diskless stanice). **RARP server** sadrži mapiranje MAC ⇔ IP. **RARP paket** ima isti format kao ARP paket. Polje **Operation** određuje ARP/RARP funkcije - 1 za ARP Request, 2 za ARP Reply, 3 za RARP Request i 4 za RARP Reply. Uređaj po uključivanju šalje **RARP Request** poruku koja sadrži MAC adresu pošiljaoca. Paket se šalje na MAC brodkast adresu. Svi uređaji primaju paket, ali samo RARP server odgovara sa **RARP Reply** porukom koja sadrži IP adresu. Nedostaci RARP protokola su činjenica da ne dodeljuje masku i gateway i da funkcioniše samo unutar lokalnog segmenta.

BOOTP (Bootstrap Protocol) je unapređenje RARP-a. Takođe koristi MAC adresu za određivanje IP adrese, ali dodatno može slati masku, gateway i DNS server. Protokol je aplikativnog sloja, a na L4 nivou koristi UDP port 68. **BOOTP server** sadrži ručno mapiranje MAC ⇔ IP. Uređaj po uključivanju šalje **BOOT-REQUEST** poruku koja sadrži MAC adresu pošiljaoca. Ona se enkapsulira u UDP poruku, a ona u IP poruku koja se šalje na brodkast IP adresu, a ona u Ethernet okvir koji se šalje na brodkast MAC adresu. BOOTP server preuzima poruku, dok je ostali odbacuju. On kreira **BOOT-REPLY** poruku koja sadrži IP adresu, masku, default gateway, IP adresu DNS servera, IP adresu TFTP servera i tako dalje. Pošiljalac prihvata podatke. Nedostatak je statičko dodeljivanje, odnosno potrebno je unapred poznavati MAC adrese, manuelna konfiguracija za sve korisnike, mapiranje "1-na-1". Ne mogu se podržati privremeni korisnici.

DHCP (Dynamic Host Configuration Protocol) je naprednija verzija BOOTP-a. Omogućava dinamičko dodeljivanje adresa na ograničeno vreme (**leasing**), kao i dodelu preko 20 dodatnih parametara pored IP adrese. Može postojati više **DHCP servera**. Proces rada:

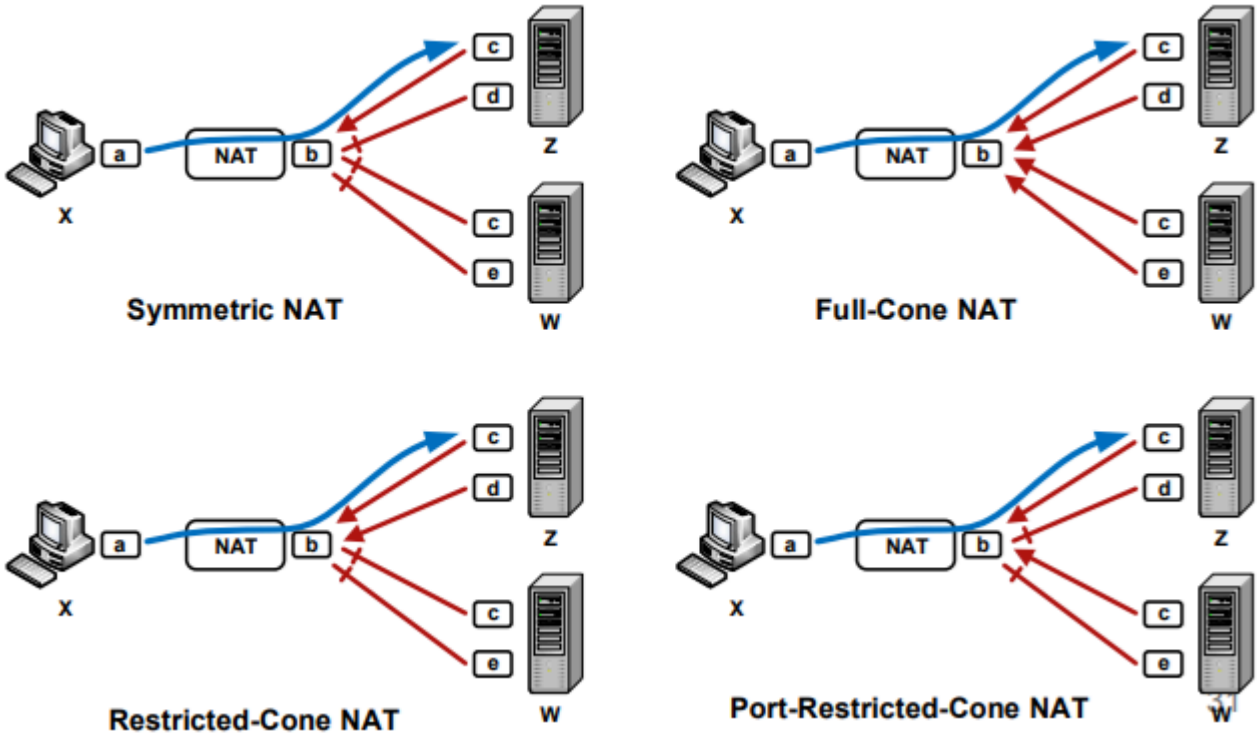
- **DHCP-DISCOVER** poruka - uređaj pri uključivanju šalje poruku koja sadrži njegovu MAC adresu. Svi DHCP serveri preuzimaju poruku, dok je ostali odbacuju.
- **DHCP-OFFER** poruka - svaki server nezavisno nudi IP adresu i ostale parametre.
- **DHCP-REQUEST** - klijent prihvata jednu od ponuda.
- **DHCP-ACK** - potvrda generisana samo od strane DHCP servera čija je ponuda prihvaćena. Uređaj počinje da koristi IP adresu i ona se označava zauzetom na neko vreme.

NAT (Network Address Translation) pretvara IP adrese iz jednog skupa adresa u drugi. Koristi se za omogućavanje korišćenjenja privatnih IP adresa koje se ne pojavljuju na Internetu, čime se štedi potrošnja javnih IP adresa. Na graničnom ruteru se prevode lokalne adrese u javne i obrnuto. Terminologija:

- **Inside Local Address** - lokalna (privatna) IP adresa uređaja.
- **Inside Global Address** - javna IP adresa uređaja koju dodeljuje provajder.
- **Outside Global Address** - IP adresa uređaja na spoljašnjoj mreži.

Statički NAT koristi fiksno mapiranje "1-na-1". **NAT tabela** sadrži unapred definisana pravila mapiranja - par lokalne i globalne adrese. Kada se komunikacija inicira iz unutrašnje mreže ka spoljašnjoj, lokalne adrese se

pretvaraju u globalne i obrnuto. Prednost je mogućnost inicijalizacije komunikacije iz spoljne mreže ka unutrašnjoj, a nedostatak što se ne postiže puna ušteda adresa. **Dinamički NAT** definiše skup (pool) javnih adresa, a pri komunikaciji iz unutrašnje mreže uzima se slobodna adresa. NAT tabela se dinamički popunjava. Posle završetka sesije, javna adresa se oslobađa i može biti dodeljena drugoj unutrašnjoj adresi, čime se postiže ušteda adresa. Za svaki red u tabeli uvodi se tajmer (**timeout**) i red se briše nakon njegovog isteka. Nedostatak je što se komunikacija može inicirati samo iz unutrašnje ka spoljašnjoj mreži. **Overload NAT** omogućava da više lokalnih adresa istovremeno koristi manji broj globalnih adresa, tako što se koristi i TCP i UDP port (**PAT - Port Address Translation**). Klijent mora da bude u unutrašnjoj mreži. Klijentski port se slučajno bira na strani klijenta, pa može i da se promeni prilikom NAT-a. **Port Forwarding** omogućava pristup serverima u unutrašnjoj mreži iz spoljašnje mreže. Spoljašnji zahtev za globalnu IP adresu i serverski port će se mapirati u lokalnu IP adresu servera, a serverski port će ostati nepromenjen. Nedostatak je to što samo jedna lokalna IP adresa može da bude uparena sa serverskim portom, odnosno samo jedan server za svaki servis (port). Da bi dva uređaja iz različitih mreža komunicirali direktno preko NAT-a najpre se registruju na javno dostupnom serveru, a server im prosleđje "NAT-ovane" adrese i portove nakon čega oni nastavljaju direktno da komuniciraju. TCP protokol je Connection Oriented, tj. pre komunikacije potrebno je uspostaviti konekciju. UDP protokol ne uspostavlja konekciju, pa svako može da pristupi na adresu i port. Postoje 4 pristupa pri određivanju ko sme da koristi globalnu adresu i port za otvorene NAT konekcije:



NAT se koristi i za ICMP pakete, koji ne koriste UDP/TCP portove. ICMP poruke upita sadrže ID polje koje se koristi za NAT mapiranje. ICMP poruke o grešci ne sadrže ID polje pa se originalni IP paket prenosi u telu ICMP poruke o grešci. U tom slučaju je potrebno promeniti lokalne adrese i portove i u originalnom paketu. Postoje i aplikacije koje prenose informacije o IP adresama u svojim podacima pa NAT uređaj mora da gleda i menja aplikativne podatke što se naziva **Application Level Gateway (ALG)**. Prednosti korišćenja NAT-a:

- Veća sloboda u dodeljivanju i korišćenju privatnih IP adresa (npr. za privatne korporacijske mreže).
- Povećana sigurnost kod dinamičkog NAT-a jer je privatni deo mreže izolovan.
- Manja potrošnja javnih IP adresa.
- Ne mora da se vrši promena adresa u privatnoj mreži prilikom promene provajdera.

Mane korišćenja NAT-a:

- Složenija konfiguracija i administracija.
- Komplikovanije procesiranje na ruterima i povećano kašnjenje saobraćaja.
- Otežano praćenje događaja poput virusa, DoS i hakerskih napada.
- Može da predstavlja problem za pojedine aplikacije koje se na aplikativnom nivou oslanjaju na IP adrese.

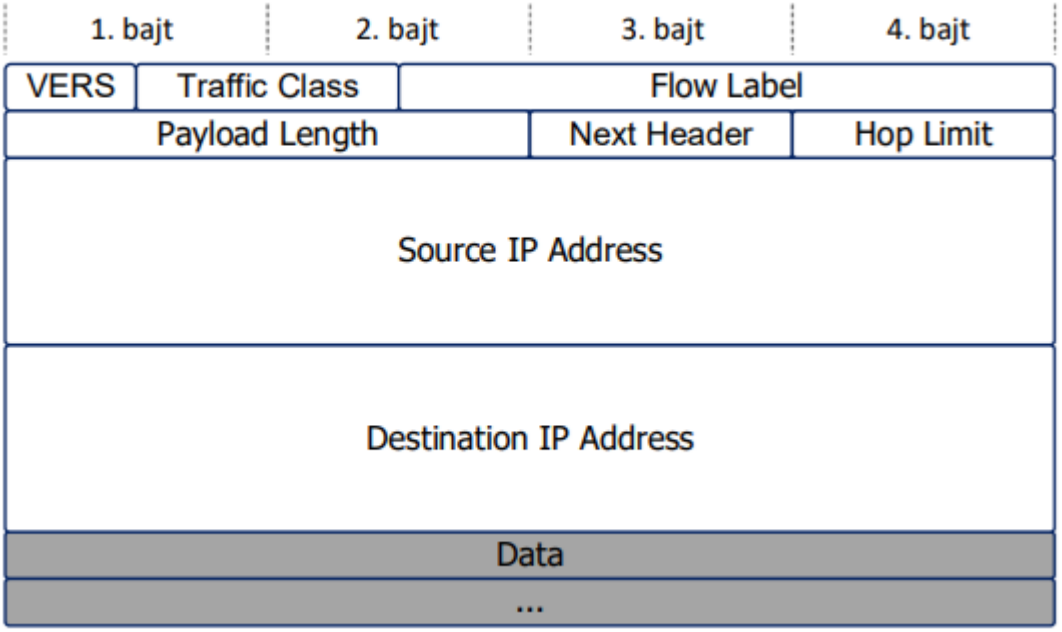
ACL (Access Control Lists) omogućava filtriranje saobraćaja na ruterima. Vrši inspekciju zaglavlja na L3 i L4 nivou. Paket može dobiti **dozvolu (permit)** u kom slučaju se propušta ili **zabranu (deny)** u kom slučaju se šalje na Null interfejs. Paket se proverava redom kroz listu uslova i pravila, sve dok se ne pronađe prvo pravilo koje odgovara. Pravilo se tada izvršava i završava se prolazak kroz listu. Ako nijedno pravilo ne odgovara paket se odbacuje - **implicite deny**. Koristi se i na ulazu u interfejs i na izlazu iz interfejsa.

Ipv6

IPv4 je standardna verzija IP protokola. Njen problem je nedostatak adresnog prostora zbog eksponencijalnog rasta Interneta i povezanih uređaja. Nove IP adrese se mogu kupiti samo od "preprodavaca". Privatne adrese i NAT donekle rešavaju problem, ali ipak su potrebne nove adrese. Takođe je potrebna i veća bezbednost podataka na IP nivou i ostvarivanje kvaliteta servisa (QoS - Quality of Service). **Ipv6** je novija verzija IP protokola sa sledećim karakteristikama:

- Veći adresni prostor.
- Efikasnije rutiranje - zaglavlje je jednostavnije što omogućava efikasniju obradu paketa. Takođe postoji manji broj eksternih ruta na Internetu - hijerarhijska struktura mrežnih adresa omogućava efikasnije agregiranje.
- Podrška za automatsku konfiguraciju računara.
- Podrška za bezbednost podataka sa IPSec implementacijom.
- Poboljšana podrška za mobilne uređaje.
- Ugrađena podrška za alokaciju resursa i QoS.
- Povećan broj multikast adresa.

Format **IPv6 zaglavlja**:



U odnosu na IPv4, izbačena su mnoga polja:

- **Internet Header Length** - IPv4 je sadržao opcije koje čine veličinu zaglavlja promenljivom. IPv6 zaglavlje je fiksne dužine jer su opcije izdvojene u posebna zaglavlja.
- **Header Checksum** - provera integriteta se sprovodi na L2 nivou, pa ovo polje nije potrebno.
- **Options** - uveden je novi mehanizam izdvajanja opcija u dodatna zaglavlja.

- **Polja za fragmentaciju (Identification, Flags, Fragment Offset)** - fragmentacija je ograničena i sprovodi se na izvorištu, a ne u ruterima. IPv6 garantuje MTU od najmanje 1280B. Izvorište koristi ili garantovani MTU ili radi **Path MTU Discovery** - šalje pakete određene veličine i prati da li je od rutera dobio **ICMPv6 poruku "Packet Too Big"**.

IPv6 zaglavlje sadrži i nova polja:

- **Traffic Class** - isto kao ToS polje kod IPv4. Izvorište može da generiše pakete koji pripadaju različitim klasama saobraćaja, sa različitim prioritetima.
- **Flow Label** - jedinstveno označava svaki flow (tok) - komunikaciju između aplikacija izvorišta i odredišta. Samo prvi paket se rutira, a Flow Label uparen sa izlaznim portom se kešira. Naredni paketi istog toka ne zahtevaju rutiranje.
- **Payload Length** - dužina podataka u bajtovima.
- **Hop Limit** - isto kao TTL kod IPv4.
- **Next Header** - identifikuje zaglavlje višeg nivoa, isto kao Protocol kod IPv4, ili zaglavlje sa IPv6 opcijama. IPv6 opcije se tretiraju isto kao zaglavlja L4 nivoa.

Ruting opcija je IPv6 opcija (paket) koja utiče na put ostalih paketa. Izvorište definiše sekvencu rutera, tj. **međutačaka (checkpoints)**, pri čemu je poslednja adresa odredište. Odredišna adresa IPv6 zaglavlja je adresa naredne međutačke. Usputni ruteri prosleđuju pakete prema navedenim međutačkama. Zaglavlje ruting opcije sadrži datu sekvencu i **brojač (Segment Left)** koji predstavlja koliko je još međutačaka preostalo. Kada ruter prepozna sebe kao odredište, a postoji ruting zaglavlje, on radi sledeće:

- N = Segment Left
- Adresa odredišta se menja sa adresom na N-toj poziciji od kraja sekvence
- Segment Left = $N - 1$

IPv6 adresa je dužine 16B, odnosno 4 puta veća od IPv4 arese. Zapisuje se u heksadekadnom obliku. Maska se koristi u prefiks notaciji ("/n"). **Skraćeni zapis** dobija se na sledeći način:

1. Izbaciti vodeće nule u grupama od 4 cifre (":000x" $\rightarrow$ ":x", ":00xy" $\rightarrow$ ":xy", ":0xyz" $\rightarrow$ ":xyz").
2. Izbaciti samo jedan niz grupa sa nulama ("0:0:0:" $\rightarrow$ "::").

Na primer: `2001:417b:0000:0000:002c:0000:0000:01af/64`

1. `2001:417b:0:0:2c:0:0:1af/64`
2. `2001:417b::2c:0:0:1af/64` ili `2001:417b:0:0:fff::1af/64`

Vrste IPv6 adresa:

- **Unicast** - jedinstvena adresa koja identifikuje interfejs.
- **Multicast** - identifikuje više interfejsa različitih uređaja prema nekoj zajedničkoj nameni. Paket poslat na multicast adresu biće prosleđen na sve pripadajuće interfejse.
- **Anycast** - adresa koja identifikuje više interfejsa različitih uređaja. Paket poslat na anycast adresu biće prosleđen samo jednom interfejsu.