**Referentni modeli**

Ciljevi ove lekcije su upoznavanje sa:

* najznačajnijim referentnim modelima,
* funkcijama svakog od slojeva referentnih modela i
* protokolima koji čine osnovu TCP/IP skupa protokola.

Specifično predznanje

Slojevita organizacija mrežnih protokola objašnjena u pretnodnoj lekciji.

Osnovni koncepti

**Uokviravanje** - organizovanje binarnih podataka u grupe (tj. pakete bitova), koje se nazivaju **okviri**.

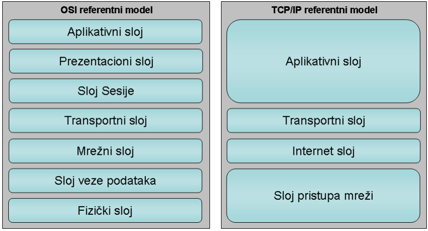
**Paket** - okvir podataka na **mrežnom** nivou.

**Segment** - okvir podataka na **transportnom** nivou.

**Port** - 16-to bitni celi broj koji identifikuje aplikacije koje međusobno komuniciraju.

-Lekcija

U prethodnoj lekciji razmatran je apstraktni model slojeva. Da bi se ostvarila konzistentnost u implementaciji različitih protokola potrebno je da postoji standardizovani referentni model. Referentni model ne daje specifikaciju kako treba implementirati protokole, već ima zadatak da omogući bolje razumevanje funkcija koje mrežni protokoli treba da implementiraju. Najpoznatiji referentni modeli su OSI i TCP/IP. Na slici 2.1 prikazana su uporedno ova dva modela.



**Slika 2.1.** Uporedni prikaz OSI i TCP/IP referentnih modela

OSI referentni model

*Open Systems Interconnection* (OSI) je model zasnovan na predlogu Međunarodne organizacije za standardizaciju (*International Standards Organization* - ISO), definisanom još početkom 80-tih godina prošlog veka, da se uvede međunarodni standard za komunikaciju otvorenih sistema. Pod pojmom „otvoreni sistem“ podrazumeva se sistem otvoren za komunikaciju sa drugim sistemima. Standard je revidiran 1995. godine i danas se najčešće koristi kao referentni model za razumevanje mrežnih protokola.

OSI model ima sedam slojeva, definisanih tako da:

-          svaki sloj predstavlja različiti nivo apstrakcije,

-          svaki sloj implementira dobro definisan skup funkcija,

-          funkcije sloja su izabrane tako da omoguće definisanje protokola koji treba da budu međunarodni standard,

-          granice slojeva treba da minimizuju protok informacija između slojeva, i to samo kroz dobro definisane interfejse i

-          broj slojeva mora biti dovoljan da se različite funkcije ne moraju smeštati u isti sloj, ali ne i prevelik, da model ne bio nezgrapan.

Razmotrimo funkcije svakog od OSI slojeva, polazeći od najnižeg sloja.

\*Fizički sloj

Osnovna funkcija fizičkog sloja je da omogući prenos digitalnih podataka (nula i jedinica) preko komunikacionog kanala. Ukratko, ako je poslata nula, da se na prijemnoj strani prepozna kao nula. A ako je poslata jedinica, da se prepozna kao jedinica. Na prvi pogled vrlo jednostavno, ali ovaj sloj mora obaviti mnogo vrlo teških zadataka. Fizički sloj sastoji se od:

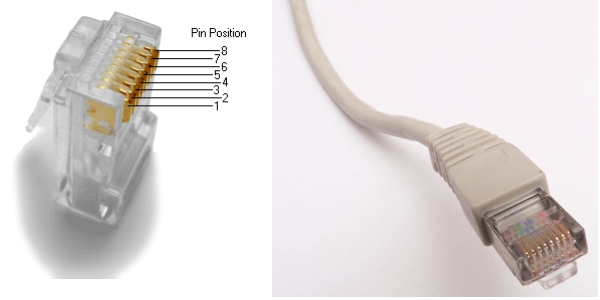
- mehaničke komponente,

- električne komponente,

- funkcionalne komponente i

- proceduralne komponente.

**Mehanička komponenta** definiše tipove kablova, konektora i raspored pinova. Na slici slici 2.2 prikazan je RJ-45 konektor. To je najčešći tip konektora koji se koristi danas kod „žičanih“ mreža. Ima 8 pinova (kontakata) koji služe za slanje i prijem podataka. Kod 10 i 100Mb *Ethernet*-a, koriste se pinovi: 1, 2, 3 i 6, dok se kod 1Gb *Ethernet*-a koristi svih osam. Detaljnije o *Ethernet* mrežama biće izložano u narednim poglavljima.



**Slika 2.2.** RJ-45 konektor

**Električna komponenta** definiše karakteristike elektromagnetnih ili svetlosnih signala, tip linijskog koda, slabljenje signala, maksimalni domet, itd.

**Funkcionalna komponenta** određuje značenje pojedinih pinova i signala. Signali se najčešće dele na:

- signale podataka (*data*)

o predajne (*transmit*)

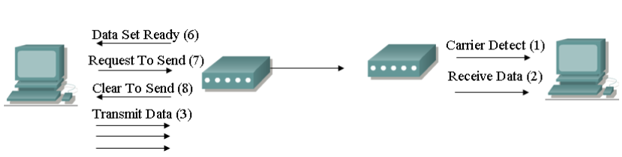
o prijemne (*receive*)

- upravljačke signale

- sinhronizacione signale i

- uzemljenje.

**Proceduralna komponenta** određuje redosled signala kojima se definišu određene operacije. Na slici 2.3 prikazan je redosled aktiviranja pojedinih pinova na 9-to pinskom serijskom konektoru prilikom komunikacije računara i eksternog serijskog modema.



**Slika 2.3.** Asinhroni prenos podataka između dva računara korišćenjem direktne veze dva modema

Kada je modem spreman za slanje podataka, on aktivira pin 6 (signal *Data Set Ready*), javljajući računaru da je uključen. Kada računar ima podatke za slanje aktivira pin 7 (signal *Request To Send*), ali ne počinje sa slanjem dok ne dobije potvrdu od modema da je spreman da primi podatke. tek nakon aktiviranja pina 8 (signal *Clear To Send*), računar počinje da šalje podatke preko pina 3 (*Transmit Data*).

Na prijemnoj strani modem obaveštava računar da je detektovao prenos aktiviranjem pina 1 (signal *Carrier Detect*), a podaci se mogu preuzeti preko pina 2 (*Receive Data*).

Fizički sloj omogućuje direktnu komunikaciju dva računara. Ukoliko je potrebno povezati više računara neophodno je uvesti i sloj veze podataka.

\*Sloj veze podataka

Sloj veze podataka omogućuje formiranje lokalnih mreža sastavljenih od više od dva računara. Osnovne funkcije ovog sloja su:

- upravljanje pristupom medijumu,

- uokviravanje,

- adresiranje,

- kontrola grešaka i

- kontrola toka.

U mrežama u kojima postoji više od dva računara koji dele zajednički medijum za prenos podataka neophodan je protokol koji će odrediti kada ko ima pravo da šalje svoje podatke. Postoje različita rešenja za ovaj problem. Jedno rešenje, koje se inače koristi kod *Ethernet* mreža, jeste da svaka stanica osluškuje komunikacioni kanal i kada utvrdi na nema emisije korisnog signala u njemu šalje svoje podatke. Tokom prenosa nastavlja da osluškuje, kako bi utvrdila da li se ono što šalje poklapa sa onim što je na kanalu. Ako postoji odstupanje, znači da još neko šalje svoje podatke i stanica se povlači i prestaje sa slanjem dok se kanal ne oslobodi. Drugo rešenje, koje se koristilo kod *TokenRing* mreža, zasniva se na postojanju specijalnog okvira koji cirkularno prolazi kroz mrežu. Kada takav okvir-žeton stigne do neke stanice, ona ima pravo da šalje. Nakon završetka slanja samo jednog okvira, stanica prosleđuje žeton susednoj stanici. Da bi ovakav algoritam mogao da funkcioniše potrebno je da stanice budu povezane u prsten.

**Uokviravanje** podrazumeva organizovanje binarnih podataka u grupe (tj. pakete bitova), koje se nazivaju **okviri**. Okviri omogućuju da se sa sigurnošću može utvrditi kada se u komunikacionom kanalu nalaze validni podaci, a kada ne postoji prenos. Na početku okvira se obično nalazi sekvenca podataka koja služi za sinhronizaciju prijemnika i predajnika, u slučaju asinhronog prenosa. Iza te sekvence nalaze se adrese pošiljaoca i primaoca, kao i druge kontrolne informacije. Uvođenje okvira omogućuje proširivanje osnovne poruke različitim upravljačkim i kontrolnim informacijama. Na kraju okvira obično se nalazi kontrolna suma koja omogućuje otkrivanje grešaka. Uokviravanje zavisi od načina pristupa medijumu i funkcija je na kojoj se baziraju sve ostale funkcije sloja veze podataka.

**Adresiranje** omogućuje da se odredi kom odredišnom računaru je poruka namenjena. Da nema adresiranja mogle bi da postoje samo direktne veze između dva računara. U okvir koji se šalje dodaju se dve adrese: adresa pošiljaoca i adresa primaoca. Adresa primaoca služi da samo jedan računar u lokalnoj mreži prepozna okvir kao nešto što je namenjeno njemu, a adresa pošiljaoca da primalac zna kome da vrati odgovor. Adrese na nivou veze podataka nazivaju se još i fizičke ili MAC adrese. Fizičkim se nazivaju zato što su najčešće definisane u ROM memoriji mrežnih kartica i ne mogu se menjati. MAC je skraćenica engleskog termina *Media Access Control* (kontrola pristupa medijumu). Adresa na nivou veze podataka zavise od načina pristupa medijumu. Zbog toga se ove adrese često zovu i MAC adrese.

**Kontrola grešaka** omogućuje otkrivanje grešaka koje se javljaju tokom prenosa podataka kroz komunikacioni kanal. Razlozi za nastajanje grešaka su različiti: preslušavanje signala iz susednih provodnika, indukcija pod dejstvom spoljašnjeg elektromagnetnog zračenja, slabljenje u provodniku, greška u odmeravanju i sl. Kontrola grešaka najčešće se implementira dodavanjem kontrolne sume, koja se izračunava na predajnoj strani na osnovu podataka koji se šalju. Na prijemnoj strani se ponovo računa kontrolna suma, na isti način kao i na predajnoj strani, i ukoliko se poklope, smatra se da je podatak dobro primljen.

**Kontrola toka** rešava probleme koji nastaju kada pošiljalac brže šalje podatke nego što je primalac u stanju da ih primi. Da bi se sprečilo „pretrpavanje“ primaoca i gubitak podataka, primalac vraća informaciju pošiljaocu šta je primio, a šta nije, i kojom brzinom je u stanju da prima podatke. Kod današnjih protokola kontrola toka najčešće nije implementirana na sloju veze podataka, već na transportnom nivou.

Vrlo često se sloj veze podataka deli na dva podsloja:

- sloj za kontrolu pristupa medijumu i

- sloj za kontrolu logičke veze.

Ovakva podela nastala je još daleke 1980. godine, kada je postojalo više protokola nivoa podataka koje je trebalo objediniti u međunarodni standard. Sve specifičnosti svakog od protokola preseljene su u sloj za kontrolu pristupa medijumu (MAC – *Media Access Control*). To je niži podsloj i odgovoran je za implementaciju odgovarajuće tehnologije. Veza sa višim slojevima, upravljanje tokom i sve ono što je moglo da se standardizuje bez obzira na tehnologiju nižeg nivoa objedinjeno je nivoom logičke veze (LLC – *Logical Link Control*).

\*Mrežni sloj

Sve dok se računari nalaze u okvirima jedne lokalne mreže, sloj veze podataka je dovoljan za njihovu komunikaciju. Međutim, ukoliko želimo da komuniciramo sa računarom koji nije u našoj lokalnoj mreži potrebna je i mrežni sloj.

Osnovni zadaci mrežnog sloja su:

-          da omogući jedinstveno i hijerarhijsko adresiranje svih računara i

-          nalaženje optimalnog puta do odredišta

Za razliku od adresa sloja veze podataka, koje su ugrađene u mrežne kartice i zavise samo od identifikatora proizvođača i serijskog broja konkretne kartice, adrese mrežnog sloja moraju imati odgovarajuću strukturu. Ta struktura treba da omogući da svaki računar na čitavom svetu ima jedinstvenu adresu i da se analiziranjem te adrese može lako locirati. Razmotrimo hijerarhijsko adresiranje na primeru adrese stanovanja. Ako šaljemo pismo nekome u inostranstvu, celokupna adresa trebalo bi da se sastoji od:

-          naziva države,

-          naziva grada (i poštanskog broja),

-          naziva ulice,

-          kućnog broja,

-          broja stana (u slučaju zgrade) i

-          imena i prezimena.

Za razliku od prethodnog primera, mrežna adresa nije tekstualna, već numerička i zavisi od protokola mrežnog nivoa koji se koristi. Kod IPv4 (*Internet Protocol* ver. 4) to je 32-bitni broj. Početni bitovi adrese definišu nadmrežu, srednji podmrežu, a najniži definišu računar. Detaljnije o adresiranju IP protokola biće reči u narednim poglavljima.

Hijerarhijska struktura adrese omogućuje rutiranje paketa. **Paket** je osnovna jedinica podataka na mrežnom nivou. Pod pojmom ruteranje podrazumeva se prosleđivanje paketa određenom trasom (rutom). Uređaji koji treba da obezbede optimalnu rutu nazivaju se **ruteri**. Oni održavaju tablicu adresa odredišta i na osnovu nje prosleđuju pakete koji pristižu odgovarajućim susednim ruterima, sve dok ne stignu do odredišta. Osnovni zadatak rutera je da pronađu optimalnu trasu, kako bi paketi stigli što brže.

Mrežni sloj „daje sve od sebe“ da paket stigne do odredišta, ali nema načina da to proveri. Drugi veliki problem koji ovaj sloj ne može da reši je kako odvojiti pakete različitih aplikacija. Jedino što mrežni sloj poznaje je adresa, a ona nije dovoljna da bi se razlikovali svi programi koji mogu da se izvršavaju na jednom računaru. A tek da ne govorimo o višestrukim instancama jednog programa. Zato su potrebne usluge viših slojeva.

\*Transportni sloj

Zadaci transportnog sloja su:

- da omogući većem broju aplikacija da komuniciraju preko zajedničkog mrežnog sloja,

- da segmentira (podeli) podatke na manje celine,

- da obezbedi kontrolu toka podataka.

Već je pomenuto da mrežna adresa na jedinstveni način određuje računar u čitavom svetu. Računari koji nude neke usluge nazivaju se **serveri**, a aplikacije koje se izvršavaju na njima **servisi**. Često se i za te aplikacije koristi termin **server**, ali ćemo mi u daljem tekstu serverom smatrati računar, a servisom – program. Navođenjem adrese servera, klijent (računar kome je potrebna neka usluga servera) može pristupiti serverskom računaru. Međutim, potreban je način da se odredi koju uslugu želi. Mrežna adresa nije dovoljna za to. Takođe, kada server vraća odgovor klijentu, potrebno je da zna kojoj aplikaciji su traženi podaci potrebni. Da bi sve ovo bilo moguće, transportni sloj uvodi **portove**. **Port** je 16-to bitni celi broj koji identifikuje aplikacije koje međusobno komuniciraju. Uvođenje transportnog sloja omogućuje da na jednom računaru istovremeno možemo krstariti Internetom, čitati elektronsku poštu, komunicirati preko nekog programa za instant-komunikaciju (*Instant Messenger*) i gledati program na nekom *on-line* kanalu, a da se paketi svih ovih aplikacija ne pomešaju.

Da bi prilagodio podatke za slanje preko mrežnog sloja, transportni sloj ih deli u manje celine. Te celine nazivaju se **segmenti**, a sam proces deljenja – **segmentacija**. Svakom segmentu dodaje se odgovarajući identifikator (broj koji na jedinstveni način određuje dati segment), kako bi na prijemnoj strani bilo moguće uklapanje segmenata u celinu. Proces segmentacije se može porediti sa deljenjem teksta dokumenta na rečenice. Svaka rečenica se zasebno šalje ka odredištu. Ali da bi primalac imao koristi od poslatog teksta, rečenice na prijemnoj strani moraju biti u istom redosledu kao na predajnoj strani. Ako dođe do tumbanja rečenica, poslati tekst može sasvim promeniti smisao. Transportni sloj zadužen je za deljenje poruke u segmente, ali i za sklapanje segmenata u celinu.

Prilikom transporta, neki segmenti mogu biti oštećeni ili izgubljeni. Zato, ovaj sloj obezbeđuje i kontrolu toka. Brzina slanja prilagođava se brzini kojom prijemnik može da prihvata i obrađuje poruke. A ako dođe do oštećenja ili gubitaka segmenata, vrši se retransmisija (ponovno slanje) takvih segmenata. Pri slanju svakog segmenta aktivira se tajmer (brojač). Ako pre istek tajmera stigne potvrda od prijemnika da je segment primljen, znači da je sve prošlo kako treba i tajmer se zaustavlja. Međutim, može se desiti da potvrda ne stigne u datom vremenskom intervalu. Čim istekne tajmer, odgovarajući segment se ponovo šalje (i restartuje se tajmer).

Iskorišćenost komunikacionog kanala, i brzina komunikacije, je vrlo mala ako se čeka da stigne potvrda za svaki segment pre nego što se pošalje sledeći. Ovakva kontrola toka naziva se „stani i čekaj“. Da bi se ubrzao prenos, koristi se tehnika poznata kao „klizni prozor“. „Klizni prozor“ omogućuje da se pošalje više segmenata pre nego što neki od njih bude potvrđen. Na primer, ako je veličina prozora 10, to znači da pošiljalac može u nizu da pošalje 10 segmenata ne čekajući ništa. Tek nakon slanja 10-tog segmenta, pošiljalac se blokira i čeka da stigne potvrda za prvi poslati segment. Čim stigne potvrda za prvi, nastavlja sa slanjem 11-tog. Kada stigne potvrda za drugi, šalje 12-ti segment, itd. Moguće je istovremena potvrda i više segmenata, samo jednom potvrdnom porukom. Dakle dozvoljeno je maksimalno 10 segmenata da čeka na potvrdu. Ukoliko se veličina prozora izabere tako da slanje svih segmenata iz prozora traje duže od vremena da jedan segment stigne do odredišta i vrati se potvrda, tok poruka je neprekidan. Tada kažemo da imamo puno iskorišćenje komunikacionog kanala.

\*Sloj sesije

Sloj sesije ima zadatak da upravlja „dijalogom“ između učesnika u komunikaciji. Ovaj sloj učestvuje u formiranju, upravljanju i raskidanju sesije. Prema smeru toka podataka, sve komunikacije možemo podeliti na:

- Simpleks (eng. *simplex*),

- poludupleks (eng. *half-duplex*) i

- dupleks (eng. *full-duplex*).

**Simpleks** je komunikacija samo u jednom smeru. To je, na primer, emitovanje signala „u etar“. Pošailjalac samo šalje, a primalac samo prima podatke. Ovde zapravo i nema sesije, jer ne može da dođe do sukoba oko toga ko šta treba da radi u komunikaciji. Uloge su unapred dodeljenje i nepromenljive.

**Poludupleks** komunikacija omogućuje prenos podataka u oba smera, ali ne istovremeno. Najbolji primer je komunikacija preko radio stanice. Samo dok drži dugme za „predaju“ jedna strana može da govori. Ali u tom periodu ne može da čuje drugu stranu. Nakon toga uloge se menjaju. Kod računarskih mreža, sloj sesije treba da upravlja procesom dodeljivanja „uloge“ u komunikaciji, i pravom da se šalju podaci.

**Dupleks** komunikacija je potpuno dvosmerna. U istom trenutku obe strane u komunikaciji mogu istovremeno i da šalju i da primaju podatke. Primer iz svakodnevnog života koji ilustruje ovaj vid komunikacije je razgovor preko telefona. Oba sagovornika mogu istovremeno da pričaju i čuju drugu stranu.

Sloj sesije ima zadatak da pre početka protoka korisnih informacija razmeni određene upravljačke informacije između strana u komunikaciji, da alocira prostor za smeštanje podataka. U toku prenosa podataka, definiše kontrolne tačke u kojima pamti dokle se stiglo sa komunikacijom. A po završetku komunikacije, raskida komunikaciju i „čisti“ sve ono što je prethodno alocirao.

\*Prezentacioni sloj

Zadaci prezentacionog sloja su:

- da standardizuje format zapisa podataka,

- da kompresuje podatke i

- da šifruje podatke.

Prezentacioni sloj ima zadatak da standardizuje zapis podataka kako bi bio razumljiv obema stranama u komunikaciji. Prosto je neverovatno koliko različitih tipova računara postoji na Internetu. Svako od njih ima svoj način za interno predstavljanje podataka, zvuka, slika i video materijala. Jedni tekstualne podatke kodiraju ASCII kodom, drugi EBCDIC kodom. Jedni predstavljaju podatke tako da se prvo smeštaju bajtovi veće težine kod višebajtnih podataka. Kod drugih je redosled obrnut. O varijetetu zvučnih i video zapisa je teško i govoriti u ovako kratkom pregledu. Ali, i pored svega, svi ovi računari komuniciraju preko mreže. I to vrlo uspešno. Zasluga je protokola transportnog nivoa, koji prevode podatke u standardni format. Na primer, standardni formati za predstavljanje statičkih slika su: PNG, JPEG, GIF, itd.

Drugi, vrlo važni, zadatak ovog sloja jeste da omogući efikasniju komunikaciju. Obzirom da komunikacija najčešće podrazumeva prenos velike količine informacija, efikasnost u mnogome zavisi od mogućnosti da se ti podaci kompresuju (smanji njihova veličina).

Nekim aplikacijama je bitno i da podaci koji se šalju na mrežu budu zaštićeni od „neželjenih posmatrača“. Zato ovaj sloj nudi i mogućnost šifrovanja podataka. Pod pojmom **šifrovanje** podrazumeva se zamena originalnih kodnih sekvenci drugim kodnim sekvencama. Na prijemnoj strani, prezentacioni sloj mora jednoznačno dešifrovati podatke. Postoje različiti metodi šifrovanja, ali najčešće su zasnovani na metodi „javnog i tajnog ključa“.

\*Aplikacioni sloj

Aplikacioni sloj nudi standardne „servise“ krajnjim korisnicima. To je najviši sloj u OSI modelu i on direktno komunicira sa korisničkim aplikacijama i predstavlja njihov interfejs ka mreži. Zadatak ovog sloja je da ponudi standardne usluge (servise). Na primer, ako je potrebno prenositi datoteke preko računarske mreže, nije neophodno razvijati poseban program za to. Odgovarajući protokol aplikacionog nivoa već brine o tome. Potrebno je samo napraviti korisnički interfejs ka njemu i proslediti datoteku.

-Enkapsulacija podataka

U realnom životu neprestano se srećemo sa enkapsulacijom (pakovanjem). Na primer, kada kupujemo sok i ne razmišljamo u kakvim je prenosnim sredstvima bio, dok ga mi nismo kupili. Najpre se flaša puni sokom. Logično je da sok, obzirom da je tečnost, ne može biti dostavljen krajnjem korisniku ako nije sipan u neku flašu, bocu ili tetrapak. Ova ambalaža sprečava da se sok razlije. Nakon toga, više flaša pakuje se u veću plastičnu ili kartonsku ambalažu (najčešće kutiju) da bi se olakšalo nošenje više flaša odjednom. Kutije se, zatim, smeštaju u kamion koji ih odvozi na veće rastojanje. Ako je potrebno odneti flaše na par stotina metara, onda kamion nije potreban. Ali za veće rastojanja, potreban nam je i ovaj vid „enkapsulacije“. Kada se sok odvozi u neku daleku zemlju na drugom kontinentu, kamion se mora ukrcati na brod. Još jedan sloj enkapsulacije!

Dakle, u nekom trenutku, dok prelazi more ili okean, sok se nalazi u flaši, koja je zapakovana u kutiju, koja se nalazi u kamionu, koji je na brodu. Četiri nivoa enkapsulacije, svaki za prenos preko odgovarajućeg „medijuma“  i na odgovarajuće rastojanje.

Kada brod stigne do obale, kamion nastavlja put kopnom da stovarišta. U stovarištu se kutije skidaju iz kamiona, a jedan deo direktno prenosi do prodavnice. U prodavnici se flaše vade iz kutija i postavljaju na rafove. Ovaj postupak „otpakivanja“ naziva se **dekapsulacija**.

Prilikom prolaska kroz slojeve OSI modela dešava se slična pojava. Podaci se inicijalno nalaze u nekom baferu u memoriji. Aplikacije direktno manipulišu tim podacima, upisuju ih i čitaju. Količina podataka može biti jako velika, i neophodno je podeliti ih u manje celine pre slanja.

**Transportni** **sloj** je prvi OSI sloj koji vrši enkapsulaciju podataka. Deli ih u celine koje se nazivaju **segmenti**. Svakom segmentu dodaje zaglavlje u koje upisuje: port pošiljaoca, port primaoca, redni broj segmenta koji se šalje (kako bi se segmenti sklopili u pravilan redosled na prijemnoj strani), redni broj segmenta koji se potvrđuje (ako se podaci primaju), veličinu prozora za slanje poruka, itd.

**Mrežni sloj** preuzima segmente i pakuje ih u **pakete**. Paketima se dodaje novo zaglavlje koje sadrži: mrežnu adresu izvora, mrežnu adresu odredišta, i neke druge kontrolne podatke. Ovako „upakovani“ podaci šalju se nižem sloju.

**Sloj veze podataka** preuzima paket i pakuje ga u **okvir**. Okvir osim paketa mrežnog nivoa sadrži i: odredišnu MAC adresu, izvornu MAC adresu, identifikator protokola višeg nivoa čiji su podaci upakovani u okvir ili dužinu okvira i kontrolnu sumu. Okvir se prosleđuje fizičkom sloju.

**Fizički sloj** šalje bit po bit okvira na komunikacioni medijum.

Na prijemnoj strani, **fizički sloj** preuzima bitove i pakuje ih i **okvire**, koje zatim prosleđuje **sloju veze podataka**. **Sloj veze podataka** proverava kontrolnu sumu. Ako utvrdi da je ona dobra, tj. da je primljen ispravan okvir, proverava se odredišna adresa, da bi se videlo da li je okvir namenjen baš toj stanici. Ako je odredišna adresa adresa te stanice, iz okvira se „izvlači“ mrežni **paket** i prosleđuje mrežnom protokolu čiji je identifikator naveden u okviru.

**Mrežni sloj** preuzima paket i proverava da li je ispravan i da li se u njemu navodi mrežna adresa baš te stanice kao odredišna adresa. Ako je tako, „izvlači“ se **segment** iz **paketa** i prosleđuje odgovarajućem protokolu višeg sloja.

**Transportni sloj** preuzima segment i proverava da li je to segment sa rednim brojem koji očekuje. Ako jeste, vraća pošiljaocu poruku da je primio sve kako treba. Ova poruka naziva se **potvrda** (*acknowlege*). Ukoliko nije po redu, smešta se u odgovarajući bafer i čeka se da stignu segmenti sa manjim rednim brojevima, kako bi se sklopila poruka. Da li će protokol transportnog sloja poslati poruku pošiljaocu da mu je stigao segment preko reda i da mu neki segmenti nedostaju, zavisi od implementacije samog protokola.

Kada su segmenti sklopljeni po redosledu slanja, protokoli viših slojeva mogu pročitati podatke i vršiti dalju obradu.

-TCP/IP

Za razliku od ISO modela TCP/IP model je mnogo bliži konkretnoj implementaciji mrežnih protokola. Zasniva se na hijerarhiji protokola koji čine osnovu globalne svetske mreže – Interneta.

TCP/IP model, kao i čitav Internet, nastao je iz ARPAnet projekta. To je projekat započet 60-tih godina prošlog veka u cilju razvoja kompjuterske komunikacione mreže. Razvoj ovog projekta vodilo je specijalno telo – DARPA (*Defense Advanced Research Projects Agency*), u sastavu Ministarstva odbrane Sjedinjenih Američkih Država (*United Stated Departmet of Defense*). Internetom i danas kruže priče da je ARPAnet bio projekat razvoja mreže koja bi „preživela“ nuklearni rat. Ova priče nikada nisu potvrđene, ali je robusnost ove mreže bila izuzetno velika. Gubitak čak i velikog dela mreže nije sprečavao preostale čvorove da komuniciraju. Ta žilavost počivala je na tehnologiji „mreža sa skretnim paketima“, koja je i danas „srž“ Interneta.

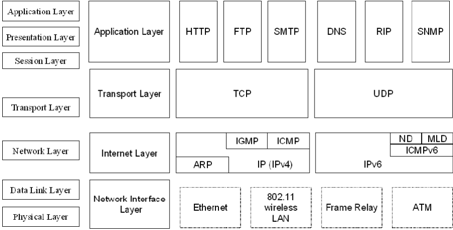
TCP/IP referentni model sastoji se od četiri sloja:

- sloja pristupa mreži,

- internet sloja,

- transportnog sloja i

- aplikacionog sloja.



**Slika 2.4.** Najvažniji protokoli TCP/IP skupa protokola i njihov raspored po slojevima

\*Sloj pristupa mreži

Sloj pristupa mreži objedinjuje fizički i sloj veze podataka OSI modela. Ova dva sloja objedinjena su iz praktičnih razloga. Naime, oni zavise od konkretne tehnologije koja je primenjena u implementaciji mreže. Zavise od tipa mrežne kartice i softvera za upravljanje tom karticom (eng. *device driver*). Svi viši slojevi mogu biti potpuno nezavisno implementirani, ali dva najniža sloja, tačnije, fizički i niži podsloj sloja veze podataka (MAC) to ne mogu. Najrašireniji protokoli ovog sloja su: *Ethernet*, 802.11, *Frame Relay* i ATM.

*Ethernet* protokol, definisan IEEE 802.3 standardom, je najrasprostranjeniji standard za žičane lokalne mreže. Javlja se u više oblika, i na fižičkom nivou koristi upredene bakarne kablove ili optička vlakna.

Za bežične lokalne mreže, danas se najčešće koriste standardi iz grupe IEEE 802.11. Kao medijum za prenos podataka koriste vazduh, tj. ne zahtevaju kablove. Brzine prenosa podataka su značajno manje u odnosu na *Ethernet* LAN, ali se mnogo lakše „postavljaju“, a omogućuju u pristup mobilnim klijentima.

U oblasti WAN mreža koriste se *Frame Relay* i ATM protokoli.

\*Internet sloj

Internet sloj je po funkciji identičan mrežnom sloju OSI modela. Osnovu ovog sloja čini *Internet Protocol* – IP.

Internet protokol, kao što mu i samo ime kaže, predstavlja osnovni protokol za prenos informacija putem Internet-a. Osnov je za formiranje mreža sa **skretnim paketima**, i omogućuje hijerarhijsko adresiranje svih računara na svetu. „Daje sve od sebe“ da paket stigne do odredišta, ali ne garantuje sigurnu isporuku, dolazak paketa po redu, niti izbegavanje dupliranja paketa. Za sve to zadužen je transportni sloj.

Prilikom prenosa podataka mogu da se dogode različite greške na putu do odredišta. Sam IP protokol ne omogućuje signalizaciju tih grešaka, ali postoji specijalan protokol koji se naziva ICMP (*Internet Contol Message Protocol*) čije je zadatak upravo to. Kada pokušamo pristup nekom računaru i dobijemo odgovor da nije dostupan, zasluga za to obaveštenje pripada upravo ICMP-u. On se oslanja na IP za prenos svojih informacija i obično se implementira kao integralni deo IP-a.

Treći, takođe vrlo važan, protokol internet sloja je ARP (*Address Resolution Protocol*). Njegov zadatak je da prevede IP adresu u MAC adresu. IP adresa je „logička“ adresa i ona zavisi od položaja računara u globalnoj mreži. MAC adresa je „fizička“ adresa i zavisi od proizvođača i serijskog broja mrežne kartice. Prevođenje jedne u drugu adresu od suštinskog je značaja za uspešno funkcionisanje računarskih mreža (bar kada je u pitanju IPv4), ali se ne može algoritamski rešiti, jer adrese nikako međusobno ne zavise. Jedini način za prevođenje jeste formiranje tabele koja definiše preslikavanje. Za održavanje te tabele zadužen je ARP.

\*Transportni sloj

Transportni sloj je identičan istoimenom sloju OSI modela, sa jednom malom razlikom. U TCP/IP modelu, transportni sloj ne mora da obezbedi uspostavljanje veze, niti kontrolu toka. Tipični protokoli ovog nivoa su *Transmission Control Protocol* – TCP, i *User Datagram Protocol* – UDP.

TCP je tipičan protokol transportnog nivoa. Omogućuje uspostavljanje veze, pouzdan prenos, numeraciju poruka, kontrolu toka, itd. On se koristi kad god je potreban pouzdan prenos podataka. Upravo zbog toga se mnogi protokoli aplikacionog nivoa oslanjaju na njega, kao na primer: HTTP (Web), SMTP (elektronska pošta), POP3 (elektronska pošta), FTP (prenos datoteka), itd.

UDP nema ni jednu od karakteristika transportnog nivoa OSI modela, osim multipleksiranje/demulpleksiranja pristupa različitih protokola aplikativnog nivoa korišćenjem portova. Podatke organizuje u segmete koji se nazivaju **datagram**-i i šalje ih ka odredištu, bez da ih numeriše ili čeka potvrdu na njih. Datagrami mogu biti oštećeni, izgubljeni, ili mogu doći do odredišta, ali da nema dovoljno mesta za njihovo smeštanje. U svakom od tih slučajeva datagram se odbacuje, bez da se izvor obavesti o tome.

Postavlja se pitanje: čemu nepouzdan protokol, ako već postoji TCP koji ima sve potrebne karakteristike pouzdanog protokola transportnog nivoa? Odgovor je jednostavan. TCP je prilično složen protokol koji može nepotrebno da optereti strane u komunikaciji. To nije veliko opterećenje za savremene računare, ali pošiljalac može biti i „pametni“ senzor, koji vrši neko merenje i rezultate tog merenja šalje ka centralnoj stanici. Implementacija TCP protokola u tom slučaju jeste nepotrebno opterećenje. Štaviše, merenja se vrše periodično. Ako neki segment i ne stigne do odredišta, to ne mora biti problem, jer će sledeće merenje dati novu validnu vrednost. U tom slučaju retransmisija može čak predstavljati i neželjeno ponašanje senzora.

Čak i kada pošiljalac i/ili primalac nisu jednostavne konstrukcije i ograničenih mogućnosti, nekada postoji potreba za implementacijom protokola koji je sasvim drugačiji od TCP-a. Tada je UDP idealan da pruži osnovnu nad kojom se nadgrađuje dati protokol.

\*Aplikacioni sloj

Aplikacioni sloj TCP/IP modela objedinjuje sloj sesije, prezentacioni i aplikacioni sloj OSI modela. Na ovom sloju postoji mnoštvo protokola od kojih ćemo pomenuti samo neke:

- HTTP (*Hypertext Transfer Protocol*) – omogućuje pristup dokumentima preko Web-a,

- FTP (*File Transfer Protocol*) – omogućuje prenos datoteka,

- SMTP (*Simple* *Mail Transfer Protocol* ) – omogućuje slanje elektronske pošte,

- POP3 (*Post Office Protocol v3*) – omogućuje preuzimanje elektronske pošte sa servera,

- DNS (*Domain Name System*) – omogućuje prevođenje simboličkih imena računara u IP adrese,

- Telnet (*Telecommunication network*) – omogućuje pristup udaljenim računarima i izvršavanje komandi na njima.

Zadatak

Upoznati se sa enkapsulacijom podataka korišćenjem **WireShark** protokol analizatora.

* Pokrenuti **WireShark** i odabrati interfejs na kome postoji saobraćaj.
* Otvoriti Web pretraživač i uneti adresu **Google**-a - www.googl.com.
* Izabrati jedan od HTTP-paketa i pogledati kako izgleda enkapsulacija podataka.
* Utvrditi koji sve protokoli učestvuju u enkapsulaciji i kako izgledaju njihova zaglavlja.