

Autor: Mina Krivokuća

Datum: Februar 2021.

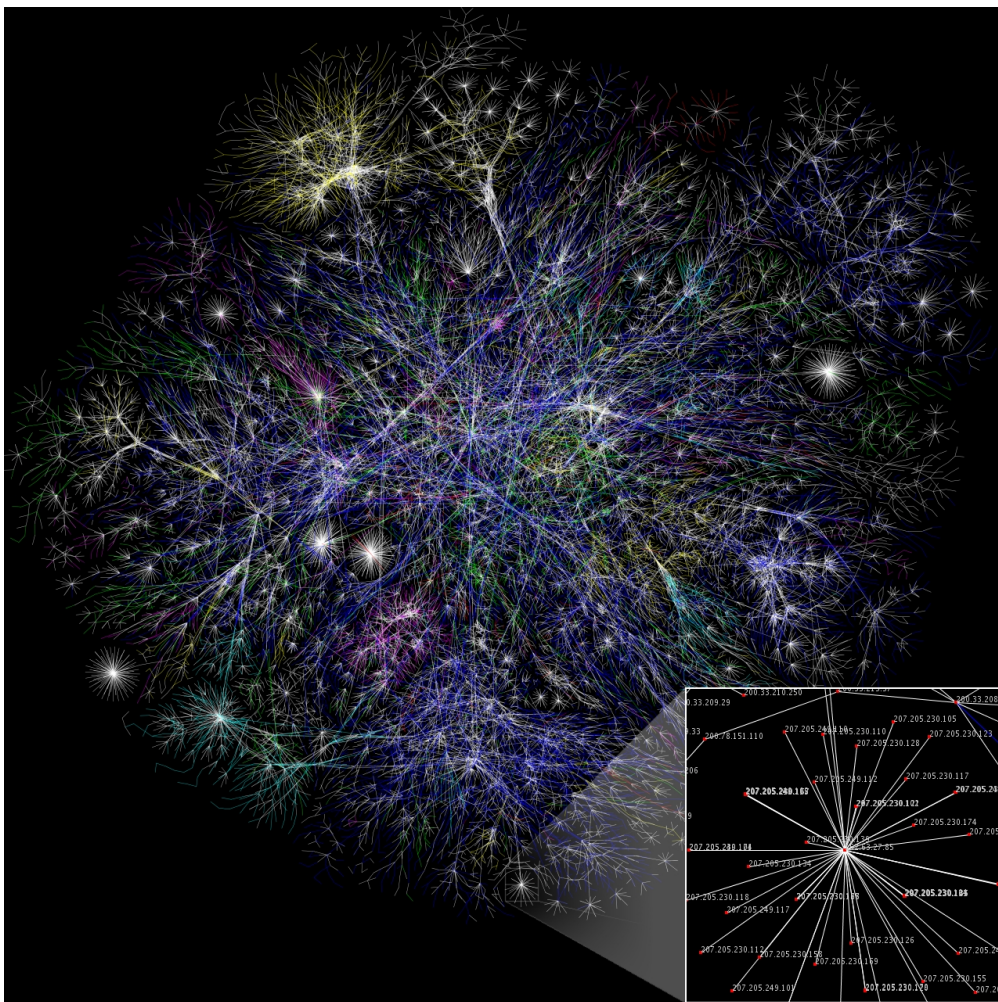
Skripta je pisana na osnovu snimaka predavanja prof. Aleksandra Kartelja i prezentacija.

- Počeci, širenje
- Uopšteno o računarskim mrežama
- Primeri upotreba mreža
- Komunikacija
- 1. Komponente mreže, tipovi veza, primeri mreža, mreže prema dimenziji, međumreže
 - Komponente mreže
 - Tipovi veze
 - Primeri mreža
 - Podela prema dimenziji
- Mreža-aplikacija interfejs

Počeci, širenje

Projekat Arpanet, 1970. - Amerika, američka vojna agencija Darpa. Projekat je imao za cilj da stvori pouzdanije sisteme komunikacije od fiksne telefonije koje bi koristili tokom nezgodnih situacija. Motivacija je bila hladni rat i strah od presečenih komunikacija - jer je fiksna telefonija bazirana na konceptu drveta i potpuno je hijerarhijska. Izbacivanjem nekog unutrašnjeg čvora ili grane drvo se preseca na dva dela. Takav sistem je bio relativno jednostavan za implementaciju i instalaciju, ali je predstavljao veliki problem u pouzdanosti i slabu tačku u situacijama gde je komunikacija i blagovremena sinhronizacija bila ključna.

Ideja novog sistema nije bila previše deterministička - grafovi komunikacije nisu izgledali kao da su planirani, već su podsećali na slučajne grafove (princip: dodaj granu tamo gde misliš da treba, pravile su se prečice tamo gde se osećala potreba). Ideja je prvo prihvaćena od strane univerziteta u Americi, naučnicima se dopala ideja stalne međusobne komunikacije. Najpoznatiji softver koji je korišćen za pristup udaljenom čvoru je Teolnet (?). Najveća motivacija za širenje je bila laka dostupnost udaljenih baza znanja i informacija. Kao druga "aplikacija" pojavljuje se mejl.



Na slici iznad je površna ilustracija izgleda Internet mreže 2005. godine. Ono što je jasno je da izgleda poprilično haotično i puna redundantnosti - što za nas znači da je razbijanje mreže na više delova presecanjem komunikacija značajno teže. Čvorovi koji deluju izolovaniji i manje povezani sa ostalima nisu nužno loše isplanirani - već su pre čvorovi koji pripadaju nekim vladinim organizacijama, vojsci, itd. Ovakvi čvorovi često imaju sisteme nezavisne komunikacije i za njih je bitno da budu i fizički udaljeni od ostalih, pored korišćenja drugačijih softvera i sistema enkripcije i zaštite.

Uopšteno o računarskim mrežama

Vrlo složena i raznorodna računarska disciplina - zahteva ekspertizu različitih struka: fizičara, elektrotehničara, tehnologa, matematičara, hemičara, programera... Teško ih je učiti na monolitni način.

- **Fizički sloj** - prenos signala, medijumi (komunikacija oprema, kablovi, hardver), teorija signala, bežični i nebežični prenos, optički kablovi, komunikacioni sateliti, sistem mobilne telefonije, kablovske televizije. Fizički sloj **se bavi omogućavanjem prenosa bitova između dve povezane tačke.**
- **Sloj veze podataka** - nadograđuje se na fizički sloj. Svaki naredni sloj koristi usluge prethodnog, i nijedan sloj ne mora da razmišlja o problemima koje je rešio prethodni. Sloj veze podataka koristi sistem za prenos bitova između tačaka koji je obezbedio fizički sloj (point to point protokol), a onda dodaje tom sistemu neke aspekte koji bi mu povećali pouzdanost i efikasnost. Konkretno, grupisaće bitove na određen način jer nema toliko smisla raditi nad pojedinačnim bitovima, pa se nad tim grupama bitova softverski vrši detekcija i korekcija grešaka. Dakle, podizanje pouzdanosti prenosa softverskim putem. Možemo zaključiti da je realan scenario u praksi da je fizički sloj loše implementiran (žice nisu dobro izolovane od šuma npr.), ali da se softverskim algoritmima to može ispraviti. Osim što povećava pouzdanost point to point komunikacije, sloj veze podataka povećava i njenu efikasnost, npr. postoje situacije u kojima jedna strana šalje drugoj veće količine podataka za koje druga strana nema odgovarajući softver da prihvati, što se rešava na nivou sloja veze podataka. Sloj veze podataka **isporučuje narednim slojevima pouzdan i efikasan sistem za prenos podataka između dve povezane tačke.**
 - **Podsloj za upravljanje pristupom medijumima** - u slučaju nepostojanja point to point protokola, već je n korisnika povezano u neku topologiju mreže, tj. nije jasno ko šta kome šalje i kada šalje (moguće je da je medijum zauzet u trenutku kada se poruka šalje). Ovaj podsloj je tehnološki prevaziđen (zastareo) i danas postoje drugi načini za rešavanje ovakvih problema. U ovom podsloju je bitna upotreba MAC adrese (hardverski identifikator uređaja u lokalnoj mreži).
- **Mrežni sloj** - koristi usluge sloja veze. Od ovog sloja pa nadalje radi se samo softverski. Dosta je apstraktniji od prethodnih slojeva i manje je opterećen elektrotehničkim detaljima. Na mrežnom sloju su ruteri. Glavna uloga mrežnog sloja (tj. rutera generalno) je da odredi optimalnu putanju u mreži kojom će se kretati paketi. Ovakav algoritam ne sme da bude centralizovan jer ukoliko bi postojali ruteri koji bi vršili usluge rutiranja za druge, ubrzo bi baš oni postali usko grlo (eng. *bottleneck*) jer bi većina zahteva stizalo upravo do njih, i odlazilo od njih. Ovakvi ruteri su prirodno distribuirani geografski i nalaze se relativno blizu svojih korisnika. Postoje ruteri i udaljeniji od krajnjeg korisnika, a oni služe uglavnom za povezivanje sa većim brojem sledećih rutera. Jedan ruter može da zna trenutno koja je najbolja putanja za paket koji mu je pristigao, ali ne može da naređuje drugim ruterima kako će dalje da ga rutiraju. Potrebno je implementirati odluke rutera tako da se odluka trenutnog poklapa sa odlukom koju bi doneo sledeći ruter u putanji ("razmišljaj globalno, deluj lokalno"). Mrežni sloj **se generalno bavi algoritmima rutiranja, upravljanja zagušenjem i povezivanjem različitih mreža.**
- **Transportni sloj** - najveći sloj, najveći napor uložen u ovaj sloj. Transportni sloj omogućava da krajnje tačke nisu sami računari, već uvodimo novi logički nivo - procese. Cilj transportnog sloja je da **apstrahuje komunikaciju između dva procesa na takav način da mi kao korisnici ne razlikujemo komunikaciju između dva procesa na istom računaru, i dva procesa na različitom računaru (ako se izuzme kašnjenje).** TCP/UDP protokoli.
- **Aplikativni sloj** - koristi usluge transportnog sloja. Protokoli višeg nivoa. Potreba da se stvari koje se koriste na isti način definišu i automatizuju (mejl npr.).

Primeri upotreba mreža

- Poslovna - razmena datoteka, deljeni resursi (npr. štampači), sredstvo komunikacije (mejl, video konferencije), poslovne transakcije elektronskim putem, poslovanje sa potrošačima preko interneta...
- Kućna - pristupanje udaljenim informacijama (filmovi, muzika, vesti, audio, video), razmena poruka, elektronska kupovina...
- Mobilna - pozivi, SMS, igrice, mape, pristup informacijama...

Komunikacija

Analogna komunikacija - u kablovima za prenos nije ugrađeno prepoznavanje brojeva (bajtova). Zvuk se transformiše u promene napona, one prolaze kroz žicu i na drugoj strani postoji dekodirer koji rekonstruiše zvuk na osnovu promene napona.

S druge strane **digitalna komunikacija** je zasnovana na međufornatu i postoji više konverzija. Prvo analizator zvuka prevodi zvuk u neke brojeve (analogno-digitalna konverzija), brojeve šalje mo kroz mrežu i ti brojevi se čuvaju preko nekih flip flopova (digitalno-analogna konverzija), ponovo se dekodira u brojeve na drugoj strani (analogno-digitalna konverzija), pa nakon toga dekodirer ponovo rekonstruiše zvuk na osnovu dobijenih brojeva (digitalno-analogna konverzija). Zaključak - konverzija se dešava više puta u oba smera.

Može biti između dva čoveka, između čoveka i računara i između dva računara.

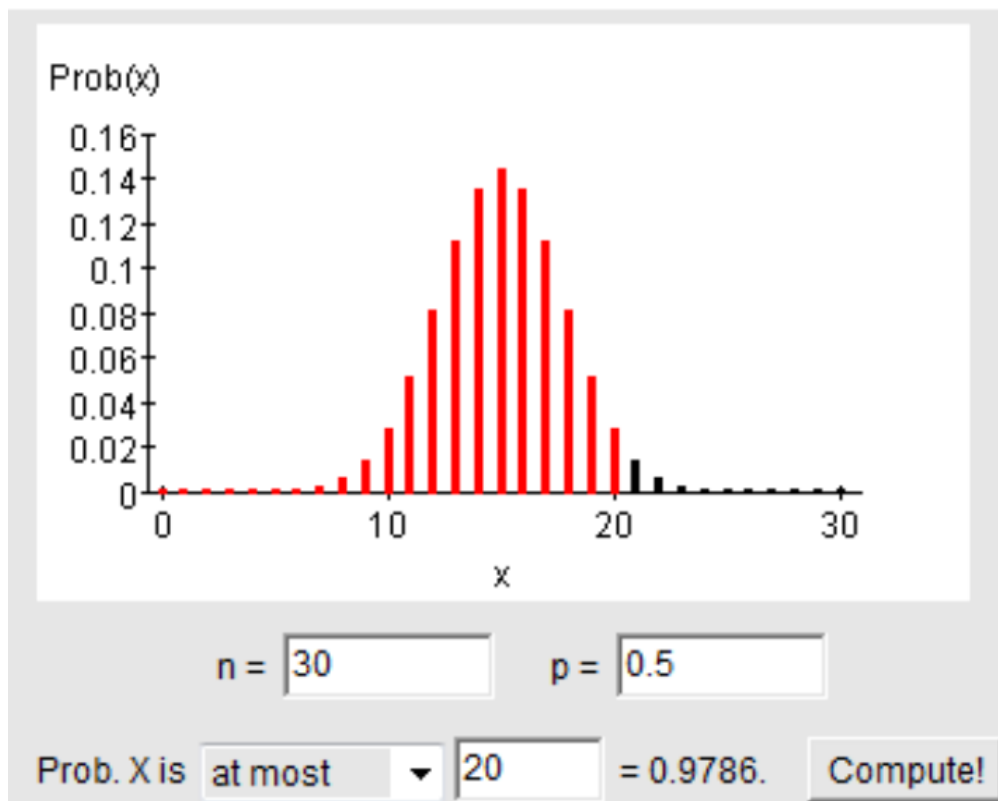
- **VoIP** - komunikacija između dva čoveka. Pozivi preko interneta. Zastarelo, danas najčešće podržava i video osim glasa. Vid digitalne komunikacije. Razlika između mobilne telefonije i VoIP-a je što informacije putuju različitim kanalima - VoIP koristi internet, a mobilna telefonija ima svoje kanale. Potrebno je obezbediti da paketi u VoIP-u što manje kasne, kao i da stignu ispravnim redosledom. Na velikim udaljenostima (npr. Srbija-Australija) se oseća kašnjenje i do par stotina milisekundi.
- **Deljenje resursa** - ekonomičan i logičan koncept, nikome ne treba sve vreme sve što poseduje. Ovo može da se odnosi i na deljenje štampača u okviru firme, a i na deljenje računarskih resursa (memorije, procesorske snage, diska, protoka...). Ne koriste svi sve resurse u isto vreme - što znači da bismo sa mnogo manjom procesorskom snagom mogli da opskrbimo više korisnika nego što bi bilo potrebno snage kada bi svako imao svoju zasebnu procesorsku jedinicu. Postoji šansa da se resursi preopterećete ako ih svi korisnici okupiraju u isto vreme, ali to se statistički retko dešava. Dakle, ideja je da se obezbede isti resursi koje bi PC obezbedio istom broju korisnika, ali sa mnogo manje iskorišćene električne struje. Sistemi koji isporučuju te resurse su klad sistemi, i njihova potrošnja električne struje je ogromna, ali je u svakom slučaju manja od potencijalne sume potrošene električne struje od strane svakog korisnika ponaosob u scenariju gde svako koristi samo svoje resurse. Nekada je postojala ideja o centralizovanim računarima u budućnosti gde bi svako kod kuće imao samo terminal, a koristio resurse udaljenog računara, ali nije zaživela.

Statističko multipleksiranje - deljenje mrežnog protoka među korisnicima na osnovu statistike zahteva. Potrebno je da znamo informacije o raspodeli upotrebnih navika tog resursa krajnjih korisnika. Primer su internet provajderi i korisnici u ISP mreži. Ako mreža ima protok od 100 Mbps, a svaki korisnik je aktivan samo 50% vremena, da li je zaista ova mreža može da podrži samo 20 korisnika, ili je to znatno veći broj?

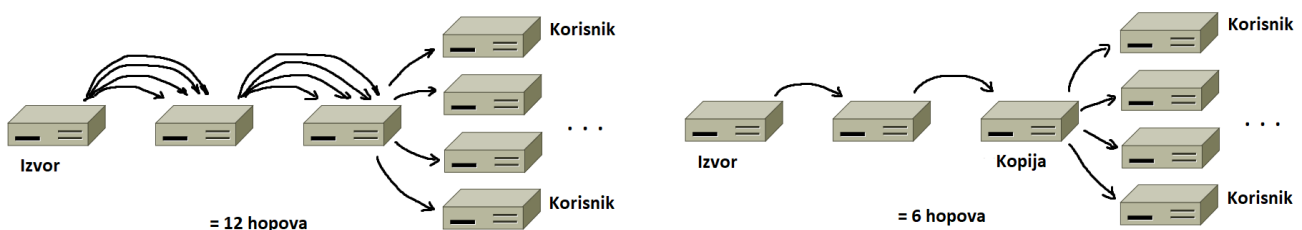
Pojednostavljena varijanta - imamo informacije o verovatnoći da kompletan protok bude iskorišćen. Binomnom raspodelom se može izračunati da čak i za 30 nezavisnih korisnika,

šanse da će biti potrebno više od 100Mbps su i dalje jako male (2%). Dobit u ovom slučaju je 1.5x.

Binomial Calculator

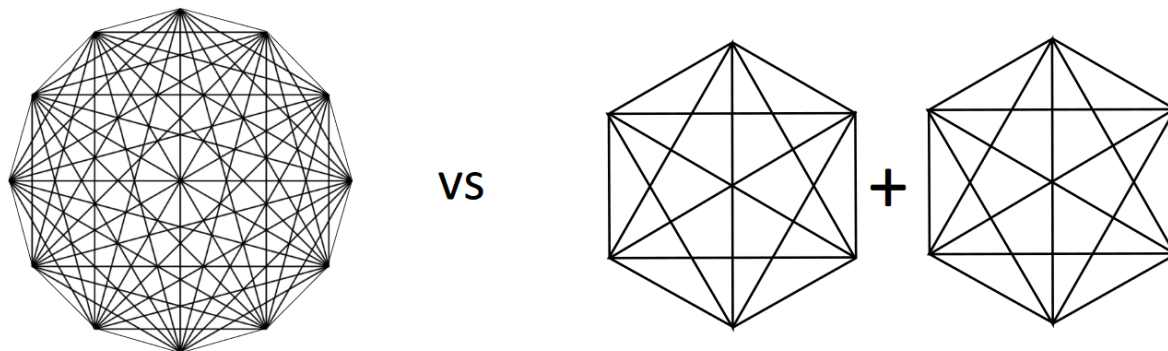


- **Dostavljanje sadržaja (content delivery)** - potreba da dostavimo neki nepromenljiv sadržaj većem broju korisnika. Trošak se uglavnom meri nekom funkcijom koja direktno zavisi od broja skokova sa čvora na čvor u mreži. Ako je svakoj grani dodeljena težina 1, za skokove kažemo da su hopovi. Ukoliko bi sadržaj putovao od izvora do svakog korisnika ponaosob, broj hopova je značajno veliki. Dakle, ideja je da se naprave kopije sadržaja u nekim čvorovima kako bi se smanjila dužina putanje kojom sadržaj putuje. Kako to izgleda:



- **Komunikacija između računara** - elektronsko poslovanje, rezervacija karata, itd. Omogućava automatsku obradu informacija nad nezavisnim sistemima.
- **Povezivanje računara sa uređajima** - razvoji senzora, očitavanje informacija sa njih. Područje primene u konstantnom razvoju (IoT - Internet of Things, pametni uređaji), zahteva jake mrežne resurse. Npr. frižider koji čita vlažnost i temperaturu, smart satovi.

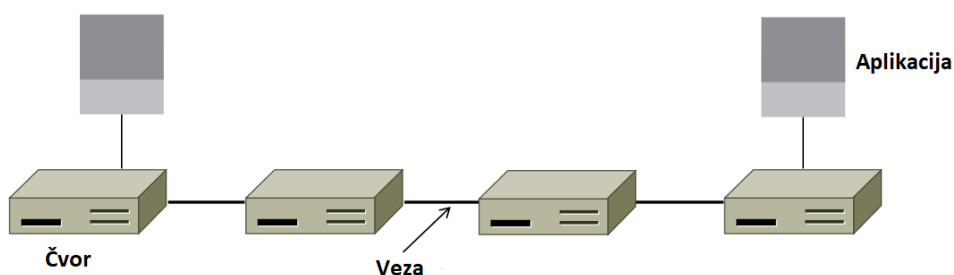
Vrednost povezivanja - Metkalfov (čovek koji je osmislio Ethernet) zakon, 1980. Vrednost mreže sa N čvorova je proporcionalna vrednosti N^2 . Velika mreža je vrednija nego više malih sa istim ukupnim brojem čvorova, to jest, veze su te koje nose informativnost, a ne čvorovi.



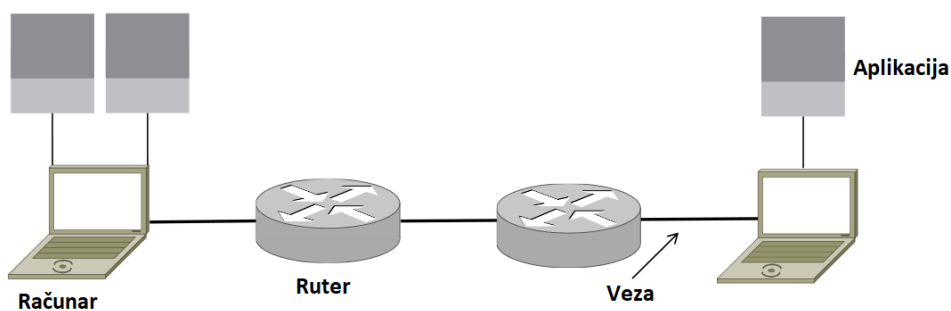
1. Komponente mreže, tipovi veza, primeri mreža, mreže prema dimenziji, međumreže

Komponente mreže

Mreže se sastoje iz čvorova, veza i aplikacija.



Listovi (krajnji računari) mogu biti krajnji korisnici ili serverski računari, a unutrašnji čvorovi ruteri. Mi se bavimo ruterima (i sličnim uređajima koji su unutrašnji čvorovi, pomenućemo kasnije).



1. **Aplikacija**, korisnik - koristi mrežu (Skype, iTunes, Amazon)
2. **Završni čvor**, računar, serverski računar, izvor, uređaj - podržava aplikaciju (laptop, mobilni telefon, dekstop računar)
3. **Ruter** ili usmerivač, unutrašnji čvor - prosleđuje poruke između čvorova (pristupna tačka, kablovski/DSL modem)
4. **Veza** ili kanal - spaja čvorove (žičane, bežične)

Tipovi veze

Podela po smeru toka informacija:

- Simpleks veza - jednosmerna, nema povratnih informacija. Primer: televizor.
- Polu-dupleks - moguća komunikacija u oba smera, ali ne istovremeno. Primer: voki-toki, radio stanice. Koristi se jedan frekvencioni opseg za slanje i primanje, što znači da se istovremenim korišćenjem signali mogu izmešati i nastao bi neki signal koji nije preterano koristan.
- Puni dupleks - moguća komunikacija u oba smera istovremeno. Primer: mobilna telefonija (istovremeno pričanje) ili mogućnost downloada/uploada istovremeno. Koristi se frekvenciona podela.

Podela po mobilnoj fleksibilnosti:

- Bežične - na ovom predmetu mislimo na mreže u kojima ne postoji tačno određen smer (vektor) u kom se šalju poruke (za razliku od lasera npr. koji je usmeren), već se one emituju. Prihvataju je svi čvorovi u opsegu. Korisnik može da se kreće i da i dalje ne izgubi vezu, što ne bi bilo moguće da je veza usmerena.
- Žične

Primeri mreža

- WiFi - bežična tehnologija, lokalna mreža
- Ethernet - žična tehnologija, lokalna mreža
- ISP mreže - mreža provajdera/usluge, nešto su veće, obuhvataju veće regije
- Kablovska/DSL
- Mobilna telefonija (2G, 3G, 4G, 5G)
- Bluetooth

Podela prema dimenziji

Tip	Dimenzija	Primer
PAN (Personal Area Network)	neposredna blizina	Bluetooth
LAN (Local Area Network)	kancelarija, sprat, zgrada	WiFi, Ethernet
MAN (Metropolitan Area Network)	grad	kablovska, DSL
WAN (Wide Area Network)	država	veliki ISP, npr. Telekom, SBB
Internet (mreža svih mreža)	planeta	internet

Međumreža ili internet, se dobija povezivanjem više različitih mreža. Međumreža je bilo koja heterogena mreža, tj. mreža koja se sastoji iz podmreža koje imaju jasnu razliku u tehnologiji izrade, topologiji i drugim aspektima. Hibrid različitih mreža.

Internet (velikim slovom) je internet koji svi koristimo. Međumreža.

Mreža-aplikacija interfejs

Utvrđeno dobri načini organizovanja mreža. Apstraktni sloj, model upotrebe mreže iz aspekta transportnog sloja. Softverske apstrakcije na mreži.

Klijent-server koncept upotrebe (model)

Zasniva se na tome da imamo računare koji nude neke usluge, i računare koji zahtevaju neke usluge. Serveri su obično hardverski bolje i naprednije mašine kako bi mogli da usluže veći broj klijenata (zahteva) istovremeno. Najveći deo posla se izvršava na serveru, dok se na klijentu izvršava samo prikaz (grafički deo). Dakle, aplikacije implementirane na klijentu nemaju nešto specijalnu/naprednu logiku, dok se na serveru dešavaju algoritmi koji zahtevaju jaku procesorsku moć.

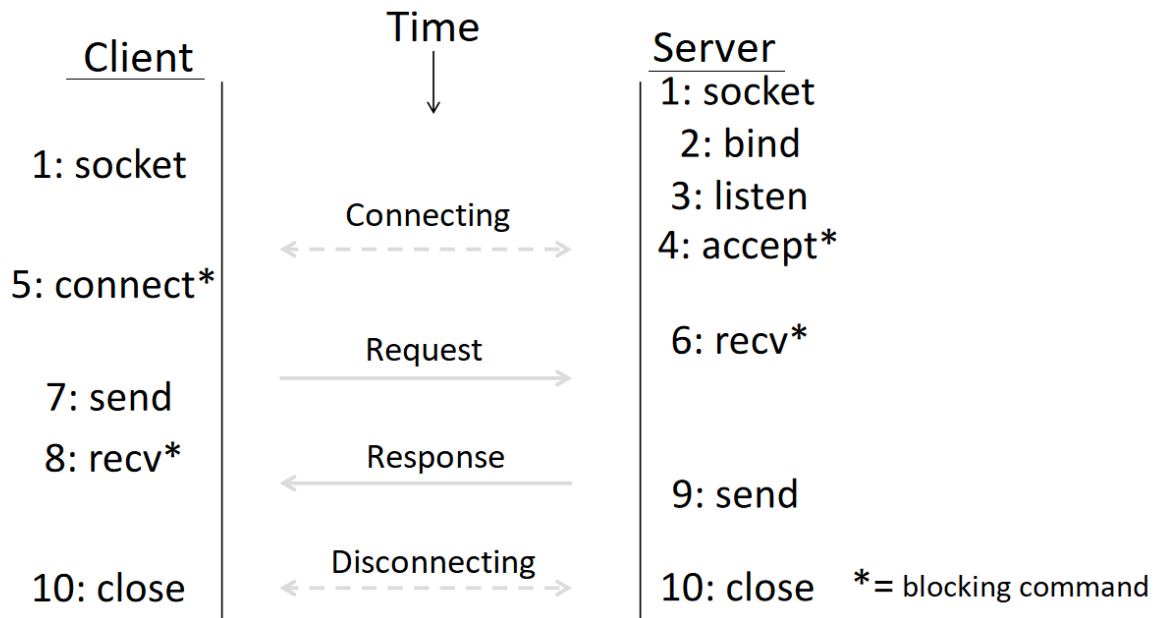
Da bi klijent-server model funkcionisao, potrebno je da klijent-server sa implementacijom ne ide do nivoa bitova, priča direktno sa fizičkim slojem ili karticom, već će pričati sa slojem ispod - transportnim slojem, i koristi njegove usluge, tj. sokete. Socket API je programski interfejs - objekat - koji ima dupleks, tj. tokove podataka (input i output stream). To je mrežni API koji se koristi za pisanje svih internet aplikacija i deo je svih poznatijih operativnih sistema. Soket postoji sa obe strane konekcije, i sa klijentske i sa serverske strane, i oba imaju odlazni i dolazni tok podataka. Socket API omogućava da se ostvari konekcija sa suprotnom stranom ne razmišljajući o aspektima ispod - bitovima, korekciji i detekciji grešaka, brzini slanja i primanja, itd. Soketi su vrlo univerzalan način razmišljanja koji se može posvetiti sa načinom komuniciranja između dva procesa. Dakle, ako bi dva procesa na istom računaru ostvarila komunikaciju, to bi izgledalo isto kao i korišćenje Socket API-ja za povezivanje na udaljeni računar. Output stream iz servera je zakačen za input stream klijenta, i obrnuto. Omogućena je pouzdana komunikacija u oba smera.

Za strimove su zakačene različite metode, nesimetrične jer se prirodno klijent i server ne ponašaju isto. Server se ponaša pasivno, a klijent aktivno. Klijent inicira komunikaciju, a server nikad samoinicijativno ne šalje zahtev. Ideja je da se server kreira, podesi i počne da radi, i čeka da se nešto desi, dok klijent čim se kreira treba da se zakači negde.

Operacija	Značenje	Gde?
SOCKET	Kreira komunikacionu tačku	oba
BIND	Pridružuje soketu lokalnu adresu	server
LISTEN	Označava da je soket spreman da prihvata konekcije	server
ACCEPT	Pasivno prihvata i uspostavlja dolaznu konekciju	server
CONNECT	Aktivno pokušava da ostvari konekciju	klijent
SEND	Šalje podatke preko uspostavljene konekcije	oba
RECEIVE	Prihvata podatke preko uspostavljene konekcije	oba
CLOSE	Zatvara konekciju	oba

Upotreba soketa, tok komunikacije

Bitno je da server bude aktivan pre klijenta, tj. da bude aktivan u trenutku kada klijent želi na njega da se zakači. Klijent inicira komunikaciju, tj. šalje se zahtev od strane klijenta, i zatim server šalje odgovor koji klijentu stiže. Veza se prekida. Ovaj proces se često dešava iterativno.



Blokirajuće komande čekaju na dešavanje i ne koriste računarske resurse. Kada stigne zahtev za konekciju na server, accept se odblokirava, prihvata konekciju i kao povratnu vrednost ima deskriptor konekcije u kom se nalaze informacije o klijentu koji želi da se zakači od kojih su najbitniji adresa i port. Nakon toga, server odlučuje da li prihvata tog klijenta i odbacuje zahtev za konekciju. Serveri se najčešće implementiraju višenitno kako bi prihvatao više klijenata istovremeno.

Aplikacije komuniciraju bez realne predstave šta je unutar mreže. **Tracert** komanda na GNU/Linux sistemima ispituje uzastopne