Sveučilište u Zagrebu
Prirodoslovno Matematički Fakultet
– Matematički odsjek

Luka Grubišić, Robert Manger

MREŽE RAČUNALA

skripta

Prvo izdanje

Zagreb, siječanj 2009 - listopad 2013.

Sadržaj

I. UVOD

1. Opis ovih skripta i kolegija o mrežama računala

Sadržaj Poglavlja 1

U ovom poglavlju govori se o ciljevima i svrsi kolegija „Mreže računala“ koji se predaje na PMF-Matematičkom odsjeku, te o vezi ove skripte s tim kolegijem.

Opći podaci o kolegiju

„Mreže računala“ predaju se na PMF – Matematičkom odsjeku Sveučilišta u Zagrebu kao izborni kolegij na trećoj godini preddiplomskog studija Matematika. Riječ je o jednosemestralnom kolegiju s 2 sata predavanja i 2 sata vježbi tjedno koji donosi 6 ECTS bodova. Od studenata se očekuje predznanje o programiranju u C-u, te o strukturama podataka i algoritmima.

Ciljevi i svrha kolegija

Ciljevi kolegija „Mreže računala“ su:

- upoznati studente s mrežama računala i mrežnim aplikacijama,
- omogućiti studentima da steknu vještinu pisanja programa koji komuniciraju preko mreže,
- omogućiti studentima da steknu vještinu oblikovanja web stranica i web sjedišta.

Svrha kolegija „Mreže računala“ mogla bi se ovako obrazložiti:

- kolegij je potreban zato što je znanje o mrežama računala sastavni dio obrazovanja svakog informatički obrazovanog stručnjaka;
- kolegij je potreban zato što su mreže računala danas svugdje prisutne i nezaobilazne, pa se od današnjeg stručnjaka očekuje da koristi i razvija aplikacije koje rade na mreži.

Savladavanjem gradiva kolegija „Mreže računala“ student se također osposobljava za praćenje niza naprednijih kolegija koji se predaju na PMF – Matematičkom odsjeku u okviru diplomskog studija Računarstvo i matematika. Ti napredniji kolegiji odnose se na razvoj web aplikacija, distribuirane procese, multimedijske sustave, kriptografiju i sigurnost mreža.

Opći podaci o skriptama

Ova skripta u potpunosti pokrivaju predavanja za kolegij „Mreže računala“. Svako poglavlje iz skripta odgovara jednom satu predavanja. Materijal je uglavnom oblikovan po ugledu na knjigu autora D. Komera koja je u našoj literaturi navedena pod brojem 1. Štoviše, u skripta su uključene i brojne slike iz Komerove knjige, dostupne na autorovim web stranicama s adresom http://www.netbook.cs.purdue.edu. U skripti se u manjoj mjeri koriste i preostali udžbenici iz literature. Tako je na primjer slika 12.5 preuzeta s adrese http://www-net.cs.umass.edu/cmpsci_591_453/schedule.htm, a potječe iz knjige autora J.F. Kurosea i K.W. Rossa koja je u literaturi navedena pod brojem 4. Svi spomenuti autori dozvolili su da

se njihove slike i drugi materijali slobodno koriste u nastavi, no oni i dalje zadržavaju autorska i sva druga prava.

Podjela gradiva na cjeline

Iz sadržaja ovih skripta vidljivo je da su predavanja podijeljena na četiri cjeline:

- uvod,
- povezivanje računala u mrežu,
- povezivanje raznorodnih mreža,
- korištenje mreža, mrežne aplikacije.

Svaka cjelina dalje se dijeli na više poglavlja. Nakon prve uvodne cjeline, druga cjelina obrađuje hardverska rješenja koja omogućuju slanje podataka od jednog računala do drugog te konkretne tehnologije koje omogućuju izgradnju lokalnih ili rasprostranjenih mreža računala. Treća cjelina posvećena je međusobnom povezivanju takvih raznorodnih lokalnih ili rasprostranjenih mreža u „mrežu svih mreža“ koju nazivamo Internet. Zadnja cjelina govori o raznim aspektima korištenja mreža i Interneta, te o pojedinim mrežnim aplikacijama.

Odnos skripta prema vježbama

Ova skripta ne pokrivaju vježbe za kolegij „Mreže računala“. Naime, na vježbama će se uz utvrđivanje gradiva s predavanja također obraditi i sljedeće dodatne teme:

- upoznavanje s građom lokalne mreže PMF-Matematičkog odsjeka,
- upoznavanje s građom Hrvatske akademske i istraživačke mreže CARNet,
- pisanje vlastitih programa koji komuniciraju preko mreže, korištenjem programskog jezika C i biblioteke Sockets API,
- rad s klasičnim aplikacijama na Internetu: ping, traceroute, telnet, FTP, e-mail, … ,
- detaljno proučavanje world wide web-a, jezika HTML i protokola HTTP,
- oblikovanje vlastitih web stranica, neposrednim pisanjem HTML koda odnosno korištenjem produktivnijih alata.

Pisani materijali vezani uz vježbe zasebno će se objavljivati na web stranicama PMF – Matematičkog odsjeka.

Sažetak Poglavlja 1

U današnjem umreženom svijetu svatko tko se bavi računalima i razvojem softvera mora poznavati mreže računala. Kolegij „Mreže računala“ daje osnovna znanja iz tog područja i služi kao priprema za naprednije kolegije s diplomskog studija. Ova skripta sadrže cjelokupni materijal s predavanja kolegija „Mreže računala“, no ne sadrže materijal s vježbi.

2. Osnovni pojmovi i činjenice o mrežama računala

Sadržaj Poglavlja 2

U ovom poglavlju objasnit ćemo osnovne pojmove kao što su mreža računala, protokol, mrežna aplikacija. Također, navest ćemo prednosti i mane umrežavanja. Objasnit ćemo razliku između tri vrste mreža: lokalnih, rasprostranjenih i interneta. Izložit ćemo kratku povijest umrežavanja, te naglasiti šire društvene posljedice nastanka i razvoja interneta.

Mreže i protokoli

Najprije objašnjavamo dva osnovna pojma koje susrećemo kad govorimo o umrežavanju.

- *Mreža računala* je skup samostalnih računala koja mogu međusobno komunicirati tako da razmjenjuju poruke preko nekog medija za prijenos podataka.
- *Protokol* je skup pravila koja definiraju format i značenje poruka putem kojih se odvija komunikacija dva računala ili dva programa. Ista riječ "protokol" može označavati i softver kojim se realizira određeni skup pravila za komunikaciju.

Razni oblici komunikacije između računala ili programa obično se ne uspijevaju realizirati jednim velikim protokolom. Umjesto toga, stvaraju se *porodice protokola* koji međusobno surađuju i organizirani su u "slojeve" (razine).

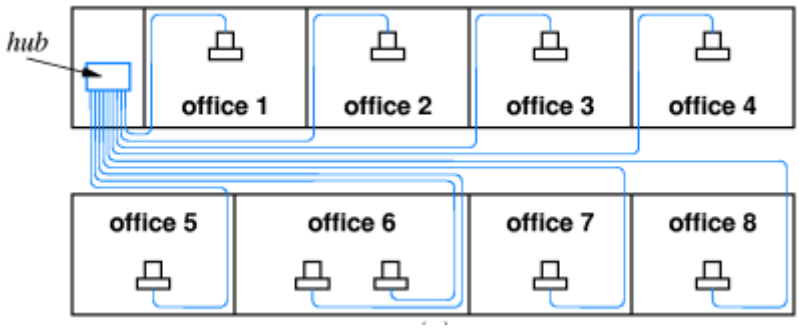
- Donji sloj (na primjer Ethernet protokol) neposredno radi s hardverom i podacima u sirovom obliku.
- Srednji slojevi (na primjer IP, TCP) pozivaju usluge nižih slojeva, te tako postaju neovisni o hardverskim detaljima.
- Gornji sloj (na primjer HTTP ili STMP) poziva usluge srednjih slojeva i bavi se porukama specifičnim za određenu aplikaciju.

Vrste mreža

S obzirom na načine povezivanja računala, razlikujemo tri vrste mreža.

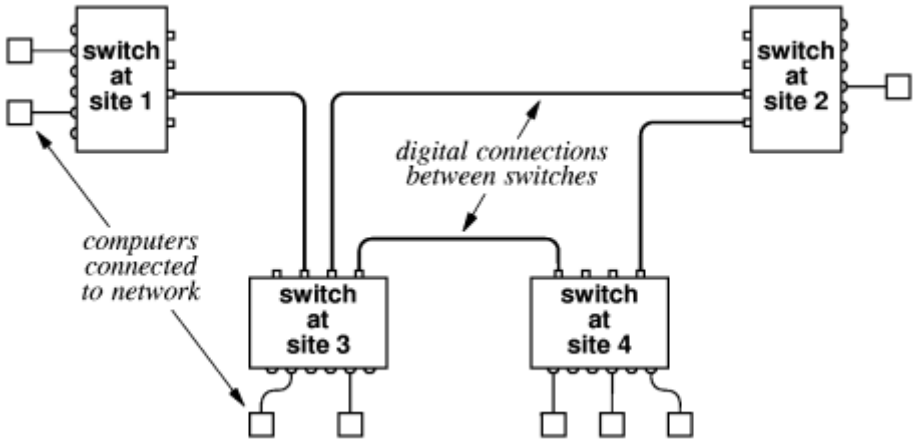
- *Lokalna mreža – LAN:* sastoji se od računala smještenih na relativno malom prostoru, na primjer u jednoj zgradi. Za povezivanje se koristi jedna određena tehnologija (na primjer Ethernet) s jednim određenim protokolom donjeg sloja. Odabrana tehnologija omogućuje veliku brzinu prijenosa podataka, ali je ograničena u pogledu maksimalne dozvoljene udaljenosti ili broja računala.
- *Rasprostranjena (globalna) mreža – WAN:* povezuje računala raspoređena na većim udaljenostima, na primjer u nekoliko gradova. Za povezivanje se obično još uvijek koristi jedinstvena tehnologija. Brzina prijenosa podataka bitno je manja nego kod LAN. Osim računala, uključeni su i posebni komunikacijski uređaji – *sklopke (switches)* koji služe za priključivanje računala, povezivanje udaljenih dijelova mreže i prijenos podataka.
- *internet:* skup raznorodnih mreža (LAN ili WAN) međusobno povezanih tako da djeluju kao jedinstvena mreža. Povezivanje se ostvaruje korištenjem posebnih komunikacijskih uređaja – *usmjernika (routera).* Svaki usmjernik istovremeno je čvor u dvije mreže, a njegova zadaća je da prebacuje podatke iz jedne mreže u drugu, konvertira ih iz jednog formata u drugi te ih usmjerava prema odredištu. Za transparentnu komunikaciju između raznorodnih mreža koje čine internet nužno je da sva računala i usmjernici primjenjuju iste protokole srednjih slojeva. Danas na svijetu postoji više-manje samo jedan ogromni *Internet* (s velikim "I"), koji povezuje više-manje sve mreže, i koji se zasniva na protokolima srednjeg sloja TCP/IP.

Na slici 2.1 vidi se primjer lokalne mreže – LAN-a. Računala su raspoređena u sobama iste zgrade (office 1 -8) i međusobno su povezana preko *koncentratora* (*hub*-a) koji omogućuje da poruka koju šalje jedno računalo bude vidljiva svim ostalim računalima.



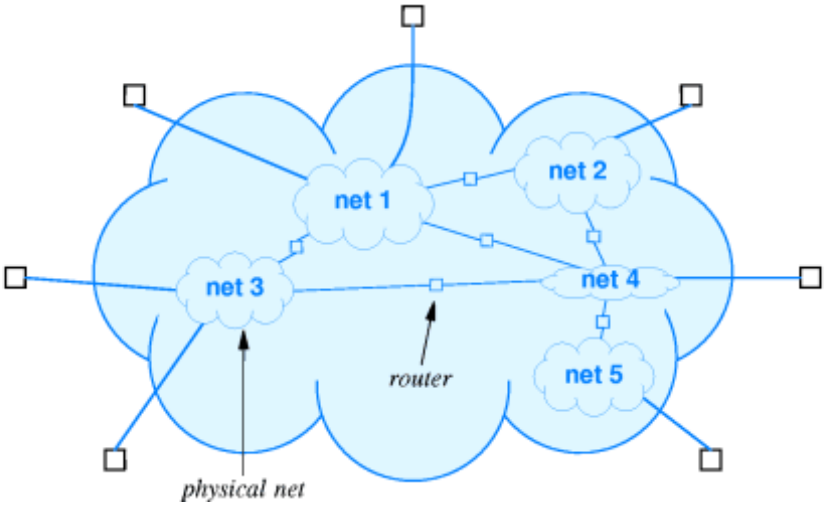
Slika 2.1: lokalna mreža – LAN.

Slika 2.2 ilustrira rasprostranjenu mrežu – WAN. Okosnicu mreže čine sklopke (switch at site 1 – 4) koje mogu biti prilično udaljene jedna od druge, na primjer svaka u drugoj zgradi ili čak drugom gradu. Sklopke su međusobno povezane telekomunikacijskim linijama (digital connections). Svako računalo (mali kvadratić) priključuje se na sklopku koja mora biti relativno blizu, na primjer u istoj zgradi. Komunikacija računala ostvaruje se posredstvom sklopki i slanjem podataka kroz telekomunikacijske linije.



Slika 2.2: rasprostranjena mreža – WAN.

Slika 2.3 predstavlja konceptualni prikaz interneta. Svaki mali „oblačić“ (net 1 – 4) označava jednu fizičku mrežu (LAN ili WAN). Svako od računala (kvadratići izvan velikog oblaka) zapravo je spojeno u jednu od tih fizičkih mreža. Usmjernici (routeri - mali kvadratići između malih oblaka) uspostavljanju komunikaciju između pojedinih fizičkih mreža. Na taj način, korisnik dobiva dojam da su sva računala spojena u jedinstvenu (virtualnu) mrežu, što je ilustrirano velikim oblakom.



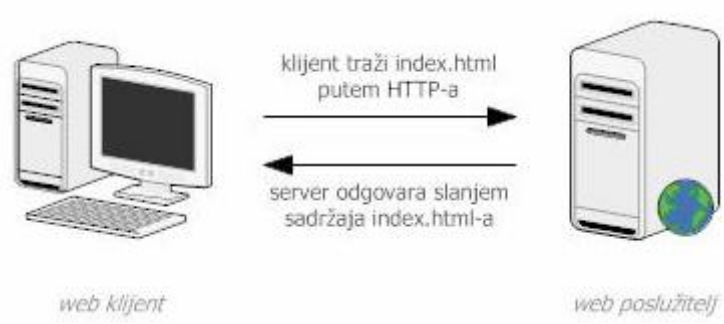
Slika 2.3: internet.

Korištenje mreža

Da bi mreže zaista mogli korisno upotrijebiti, potreban nam je aplikacijski softver koji radi na umreženim računalima. Govorimo o mrežnim aplikacijama. Točnije: *mrežna aplikacija* je skup programa koji rade na više umreženih računala, međusobno komuniciraju nekim protokolom gornjeg sloja te na taj način ostvaruju jednu određenu primjenu računala.

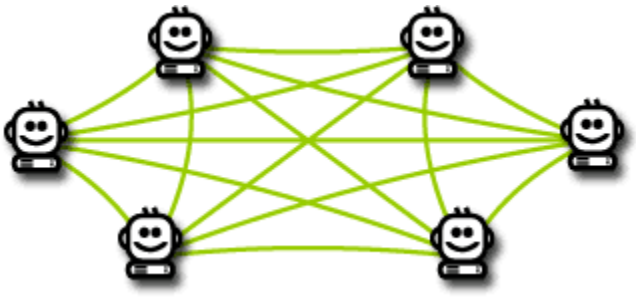
Unutar mrežne aplikacije obično se pojavljuju programi dva tipa:
- *klijenti* (traže usluge),
- *poslužitelji* (daju usluge).

Primjeri aplikacija oblika klijent-poslužitelj su world wide web odnosno e-mail, s protokolima HTTP odnosno SMTP. Slika 2.4 odnosi se konkretno na world wide web.



Slika 2.4: www kao primjer aplikacije oblika klijent-poslužitelj.

U novije vrijeme pojavljuju se i aplikacije oblika *peer-to-peer*, gdje nema jasne razlike između klijenata i poslužitelja, a poslužiteljski dio posla je distribuiran. Primjeri aplikacija oblika peer-to-peer su Napster odnosno BitTorrent, gdje svaki korisnik stavlja na raspolaganje svoje datoteke (pa obavlja funkciju poslužitelja), a traži druge datoteke (pa istovremeno radi i kao klijent). Ideja aplikacije oblika peer-to-peer vidljiva je na Slici 2.5.



Slika 2.5: aplikacija oblika peer-to-peer.

Prednosti i mane umrežavanja

Umrežavanje računala ima sljedeće prednosti.

- *Dijeljenje resursa.* Moguće je s jednog računala koristiti hardverske ili softverske resurse koji pripadaju drugom računalu, na primjer štampač, disk, datoteku, program.
- *Otvorenost.* Moguće je međusobno povezati hardver i softver različitih proizvođača, pod pretpostavkom da svi poštuju određene standarde.
- *Paralelni rad.* Usklađeni procesi koji se istovremeno odvijaju na više računala mogu obaviti više posla nego što bi bilo moguće u jednakom vremenu na jednom računalu.
- *Skalabilnost.* Performanse umreženog sustava mogu se u principu povećavati dodavanjem novih računala.
- *Robusnost* (fault tolerance). U slučaju kvara jednog računala u principu je moguće poslove preraspodijeliti na preostala računala, tako da sustav i dalje radi.
- *Transparentnost.* Korisniku se umreženi sustav može predočiti kao integrirana cjelina, dakle korisnik ne mora znati ni brinuti o tome gdje se fizički nalaze resursi koje on koristi.

Osim prednosti, umrežavanje također donosi i sljedeće mane.

- *Složenost.* Nužno je usvojiti velik broj tehnologija i standarda. Potreban je glomazan komunikacijski softver. Mrežne aplikacije teško je testirati jer paralelni rad može dovesti do suptilnih grešaka.
- *Smanjena sigurnost.* Podaci putuju mrežom pa ih je moguće "prisluškivati" ili čak mijenjati. Napadač se lažno može predstaviti kao dio sustava.
- *Otežano upravljanje.* Veći broj raznorodnih umreženih računala i komunikacijskih uređaja teže je držati pod kontrolom nego jedno računalo.
- *Nepredvidivost kakvoće usluge* (Quality of Service – QoS). Brzina odziva promatrane aplikacije ovisi o ukupnom opterećenju mreže a ne samo o toj aplikaciji.

Povijest umrežavanja i Interneta

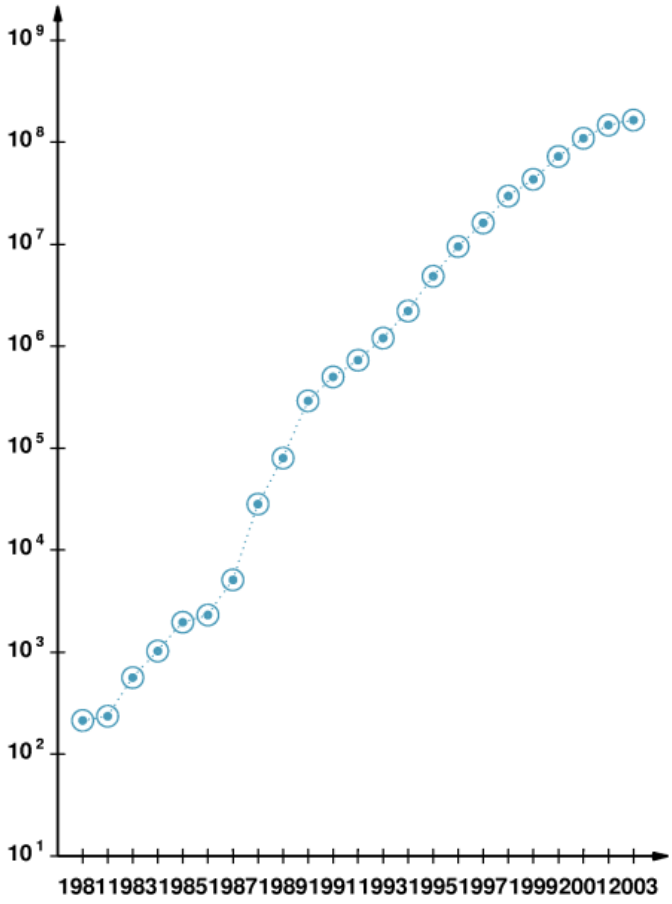
S umrežavanjem računala počelo se eksperimentirati još u 60-tim godinama 20. stoljeća. Prve mreže imale su „proprietary" karakter, dakle bile su stvorene unutar jedne organizacije s jasno određenom svrhom i koristile su jednu vrstu mrežne tehnologije. Do većeg povezivanja raznorodnih mreža i nastanka onog što danas zovemo Internet došlo je tek u 90-tim godinama.

Povijest umrežavanja tekla je otprilike ovako.

- *1961 – 1972.* Pojava "packet switchinga" i eksperimentalne mreže ARPANET (preteče današnjeg Interneta) s 15 čvorova.

- *1973 – 1980.* Razvoj drugih "proprietary" mreža. Pojava Ethernet-a. Oblikovanje ranih verzija internet protokola. Rast ARPANET-a na 200 čvorova.
- *1981 – 1990.* Širenje daljnjih akademskih mreža u SAD: BITNET, CSNET, NSFNET. Oblikovanje TCP/IP kombinacije protokola kakvu imamo danas. Pojava aplikacija s klijentima i poslužiteljima: telnet, FTP, e-mail.
- *1991 – 2000.* ARPANET prestaje postojati, a druge akademske mreže u SAD preuzimaju njegov protokol TCP/IP i međusobno se povezuju u Internet. Uključivanje akademskih mreža iz drugih zemalja u Internet, te njegovo širenje izvan akademske zajednice. Izum world wide web-a u institutu CERN.
- *2001 – 2010.* Daljnje širenje i komercijalizacija Interneta, jačanje kompanija poput Cisco, Yahoo, e-Bay, Google, Amazon. Pojava novih aplikacija poput VoIP, VideoIP, Napster, od kojih su neke tipa peer-to-peer. Širokopojasni pristup Internetu od kuće, bežični pristup preko mobitelske infrastrukture.
- *2011* – do danas. Širenje malih mobilnih uređaja poput pametnih telefona ili tableta koji su stalno spojeni na Internet. Uključivanje u Internet raznih kućanskih aparata kao što su televizori, fotoaparati, glazbene linije, uređaji za centralno grijanje, itd. Sveprisutnost Interneta, no sve manja zastupljenost konvencionalnih računala.

Rast Interneta u razdoblju od 1990. do 2003. godine lijepo je ilustriran grafom na Slici 2.6. Ovdje je riječ o logaritamskoj skali, što znači da je rast bio eksponencijalan, a broj spojenih računala povećavao se 10 puta svake 3-4 godine. Sličan trend rasta postoji i danas i to najviše zahvaljujući mobilnim uređajima i ostalim aparatima koje obično ne smatramo računalima.



Slika 2.6: rast Interneta.

Šire društvene posljedice postojanja Interneta

Makar je bio stvoren unutar akademske zajednice i bio motiviran sasvim određenim akademskim primjenama, Internet je ubrzo nadrastao ciljeve i vizije svojih stvoritelja. Danas je sasvim jasno da je Internet svojim postojanjem stvorio nesagledive društvene i kulturološke posljedice. Navodimo neke od njih.

- Mogućnost pristupa ogromnoj količini informacija pohranjenih diljem svijeta.
- Novi oblici komuniciranja: e-mail, diskusijske grupe, blog-ovi, tele-konferencije, mobilne komunikacije ...
- Veći stupanj automatizacije proizvodnih procesa.
- Mogućnost rada na daljinu i rada od kuće.
- Transformacija poslovnih i javnih djelatnosti: e-trgovanje, obrazovanje na daljinu, e-uprava, telemedicina, ...
- Novi oblici zabave: on-line igre, virtualni život.

Daljnji tijek ovih skripta

U daljnjim poglavljima ovih skripta detaljno će se razraditi pojmovi i činjenice spomenute u ovom poglavlju. Pritom će naglasak biti na konceptima, a zanemarivat će se tehnički detalji. Također, u daljnjim poglavljima utvrdit će se precizna i konzistentna terminologija. Redoslijed izlaganja slijedit će takozvani *bottom-up* pristup (odozdo prema gore), dakle:

- prvo ćemo govoriti o jednostavnijim vrstama mreža, zatim složenijim, sve dok se dođemo do Interneta;
- prvo ćemo objašnjavati donje slojeve protokola, zatim srednje i više, sve dok se dođemo do mrežnih aplikacija.

Sažetak Poglavlja 2

Prednosti umrežavanja daleko su veće od mana. Zato su danas gotovo sva računala uključena u neku lokalnu ili rasprostranjenu mrežu, a te lokalne ili rasprostranjene mreže međusobno su se povezale u globalnu „mrežu svih mreža“ Internet. Za takve složene oblike umrežavanja potreban je složeni komunikacijski softver, dakle protokoli organizirani u slojeve. Korištenje računala sve se rjeđe svodi na pokretanje lokalnih programa na tom računalu, a sve češće na pokretanje mrežnih aplikacija gdje programi rade na raznim računalima i međusobno razmjenjuju poruke kroz mrežu. Mrežne aplikacije obično se sastoje od dvije vrste programa: klijenata i poslužitelja. Razvoj i rast Interneta bio je vrlo intenzivan, te je doveo do širih i još uvijek nesagledivih društvenih i kulturoloških posljedica.

II. POVEZIVANJE RAČUNALA U MREŽU

3. Mediji za prijenos podataka

Sadržaj Poglavlja 3

U ovom poglavlju govorimo o medijima koji omogućuju prijenos podataka između računala. Za svaki od medija navodimo njegova svojstva, ističemo njegove prednosti i mane, te spominjemo gdje se oni najčešće koriste.

Klasifikacija medija za prijenos

Mediji za prijenos podataka dijele se u dvije skupine:
- *Žičani mediji.* Zahtijevaju da se računala povežu nekom vrstom žice.
- *Bežični mediji.* Računala nisu povezana nikakvim materijalom. Podaci se prenose kroz prostor nekom vrstom elektromagnetskih valova.

Žičani mediji mogu biti:
- *Bakrene žice*.
- *Optička vlakna*.

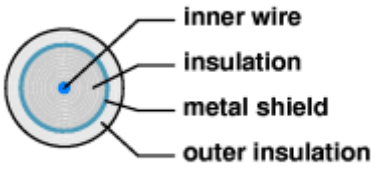
Bežični mediji su:
- *Radio valovi*.
- *Mikrovalovi.*
- *Infracrvene zrake.*
- *Laserske zrake.*

Svojstva bakrene žice

Prijenos podataka kroz bakrenu žicu ostvaruje se pomoću električne struje. Od raznih vodiča koristi se baš bakar zato jer je on dobar vodič električne struje, a još uvijek je relativno jeftin. Loša osobina bakrenih žica je pojava problema interferencije – dvije žice induciraju struju jedna u drugoj i tako proizvode smetnju. Konstrukcija pojedinih tipova žica nastoji smanjiti interferenciju. Dobro svojstvo bakrenih žica je da se one lagano se savijaju i spajaju.

Bakrene žice tradicionalno se primjenjuju za povezivanje računala u LAN. Pritom su u upotrebi tri tipa takvih žica:
- *neoklopljena parica* (unshielded twisted pair – UTP),
- *koaksijalni kabel* (coaxial cable – coax),
- *oklopljena parica* (shielded twisted pair) – kombinacija UTP i coax.

Slika 3.1: UTP kabel.



Slika 3.2: koaksijalni kabel.

UTP kabel sastoji se od dvije isprepletene bakrene žice, kao što je ilustrirano na Slici 3.1. Isprepletenost smanjuje interferenciju. Koaksijalni kabel je građen kao što se vidi na Slici 3.2, dakle riječ je o bakrenoj žici koja je pokrivena s dva sloja izolacije i oklopljena metalom. Oklop opet služi za smanjenje interferencije. Shielded twisted pair dobiva se tako da se dva UTP kabla građena kao na Slici 3.2 isprepletu na način prikazan na Slici 3.1.

Svojstva optičkih vlakana

Optička vlakna su tanke niti stakla u plastičnim ovojnicama. Njihov izgled vidi se na Slici 3.3. Podaci se kroz njih prenose pomoću svjetla određene boje kojeg proizvodi light emitting dioda (LED) ili laser. Mogu prenositi signal na puno veću udaljenost nego bakrena žica. Također, ona ostvaruju najveću moguću brzinu prijenosa. Daljnje dobro svojstvo im je da su otporna su na elektromagnetske smetnje. Mogu se donekle savijati, ali ne pod pravim kutom. Loša osobina im je da ih je teško spajati i popravljati u slučaju loma. Optička vlakna obično se primjenjuju se u WAN za povezivanje udaljenih lokacija, a koji put također i u LAN.

Slika 3.3: snop optičkih vlakana.

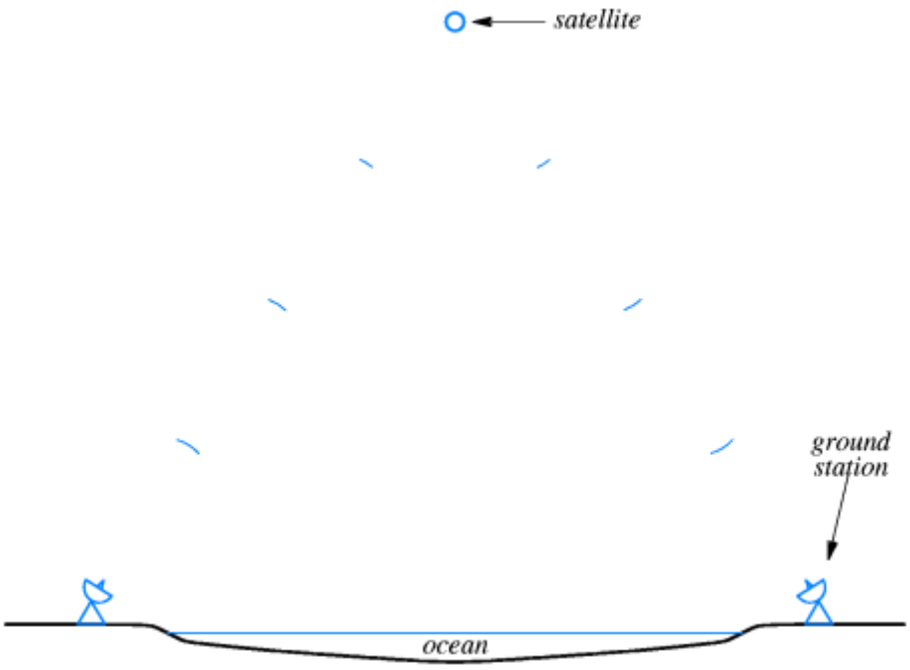
Svojstva radio valova

Riječ je o elektromagnetskim valovima iz frekventnog raspona koji se inače koristi za radio ili televiziju. Podaci se prenose preko valova određene frekvencije, slično kao radio program. Računala moraju imati antene za emitiranje i primanje valova. Domet ovisi o izabranoj frekvenciji valova. Primjenjuju se za bežične ("wireless") LAN-ove – vidi sliku 3.4, pogotovo za spajanje prijenosnika na mrežu. Također se primjenjuju za uspostavljanje interkontinentalnih veza između dijelova Interneta – tada su potrebni sateliti. Svrha satelita u interkontinentalnim vezama je pojačavanje radio signala i svladavanje zakrivljenosti zemlje, kao što se vidi na Slici 3.5.

Sateliti se dijele na dvije vrste.

- *Geostacionarni:* stoje u odnosu na Zemljinu površinu. Svi se "guraju" u istoj orbiti na 35785 kilometara iznad ekvatora, tako da je prostor za njih već potrošen.
- *Nisko-orbitni:* pomiču u odnosu na Zemljinu površinu. Mora ih biti nekoliko, a antene na Zemlji moraju se okretati.

Slika 3.4: instalacija za wireless mrežu.



Slika 3.5: interkontinentalna komunikacija preko satelita.

Svojstva mikrovalova

Riječ je o elektromagnetskim valovima iz frekventnog raspona iznad onog koji se koristi za radio ili televiziju. Podaci se opet prenose preko valova određene frekvencije, slično kao radio program. Za razliku od radio valova, mikrovalovi se mogu usmjeriti prema jednoj točki, čime se štedi energija i sprečava "prisluškivanje". Također, mikrovalovi mogu nositi više informacija nego radio valovi. Mana im je da ne mogu proći kroz neke vrste zapreka. Antene

se zato moraju postaviti tako da među njima postoji “optička vidljivost”. Primjena je u gradskim WAN-ovima, tamo gdje bi inače bilo skupo polaganje žica.

Svojstva infracrvenih zraka

Opet je riječ o elektromagnetskim valovima, no ovaj put iz infracrvenog (toplinskog) spektra, dakle ispod frekventnog raspona vidljive svjetlosti. Podaci se opet prenose preko valova određene frekvencije. Infracrvene zrake predstavljaju jeftino rješenje u odnosu na druge bežične medije jer ne zahtijevaju antene. No one imaju mali domet, svega nekoliko metara. Koriste se za bežično povezivanje uređaja unutar jedne sobe: na primjer prijenosnika, tipkovnice i miševa.

Svojstva laserskih zraka

Kod laserskih zraka podaci se pretvaraju u svjetlo, koji se umjesto optičkim vlaknima prenosi zrakom. Koristi se lasersko svjetlo, zato jer ono ima relativno veliki domet i može se usmjeriti prema jednoj točki. Primjena je ograničena zato jer laserske zrake ne mogu proći kroz vegetaciju, snijeg ili maglu. Prijemnici i predajnici moraju opet biti postavljeni tako da među njima postoji “optička vidljivost”.

Usporedba raznih vrsta medija

Žičani mediji općenito ostvaruju veće propusnosti, bolje se mogu zaštititi od “prisluškivanja”, nisu osjetljivi na atmosferske prilike. Bežični mediji općenito imaju manju cijenu uvođenja (osim onda kad trebamo satelite), nisu podložni oštećenjima medija, lakše ostvaruju difuziju (broadcast) iste poruke većem broju primatelja. Kod svih vrsta medija moguće su greške ili gubici pri prijenosu podataka. Za ožičenje LAN-a bakrene žice su jeftinije rješenje, a staklena vlakna pouzdanije i s većim dometom.

Sažetak Poglavlja 3

Mediji za prijenos podataka dijele se na žičane i bežične. Žičani mediji su bakrene žice i optička (staklena) vlakna. Od bežičnih medija naviše se koriste radiovalovi. Svaki od medija ima svoje prednosti i mane, te u skladu s njima on nalazi svoje najpogodnije mjesto primjene. Bakrene žice služe za ožičenje LAN-ova, optička vlakna za WAN-ove ili zahtjevnije LAN-ove, a radiovalovi za bežično spajanje prijenosnika na mrežu.

4. Slanje bitova kroz medije

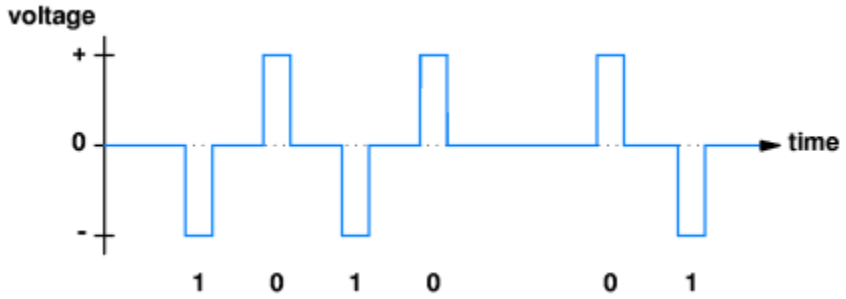
Sadržaj Poglavlja 4

Nakon što smo u prethodnom poglavlju proučili razne vrste medija za povezivanje računala, u ovom poglavlju bavimo se pitanjem kako ostvariti prijenos digitaliziranih podataka kroz te medije. Očito je da će se sami bitovi prije slanja morati pretvoriti u neki oblik kontinuiranog signala, te nakon primanja ponovo reproducirati iz tog primljenog signala. Uobičajeni načini pretvorbe su neposredno pretvaranje bitova u signal, odnosno moduliranje već zadanog kontinuiranog oscilirajućeg signala-nosača. U ovom poglavlju objasnit ćemo oba načina pretvorbe, no više pažnje ćemo posvetiti moduliranju. Također ćemo govoriti o

multipleksiranju, dakle istovremenom slanju većeg broja signala kroz isti medij. Spomenut ćemo i vrste hardverskih uređaja koji se koriste za moduliranje odnosno multipleksiranje.

Neposredno pretvaranje bitova u napon

Zamislimo da su pošiljatelj i primatelj povezani bakrenom žicom. Bavimo se pitanjem: kako niz bitova pretvoriti u električnu struju i poslati ga od pošiljatelja do primatelja kroz tu žicu? Najjednostavnija ideja koja nam odmah pada na pamet je: prikazati bit 1 jednim naponom (na primjer -15 V), a bit 0 drugim naponom (na primjer +15 V). Na taj način, niz bitova pretvara se u struju sa stepenastim naponskim dijagramom kao što se vidi na Slici 4.1.



Slika 4.1: neposredno pretvaranje zadanog niza bitova u signal s promjenjivim naponom.

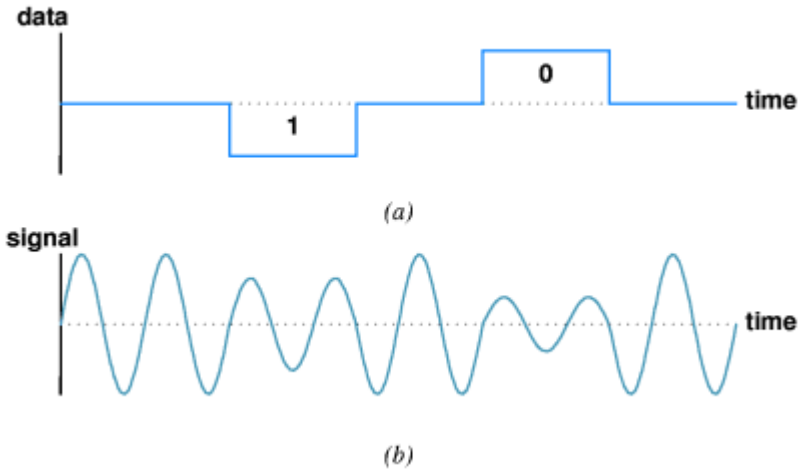
Spomenuta ideja koristi se unutar standarda RS-232 za povezivanje računala s tipkovnicom ili modemom ili tekstualnim terminalom. Sličan no nešto kompliciraniji način prijenosa postoji u Ethernet LAN-u. Ideja je nažalost primjenjiva samo za vrlo kratke udaljenosti. Naime, kod imalo većih udaljenosti zbog otpora u žici dolazi do pada jakosti struje i gubitka signala, pa moramo pribjeći složenijim tehnikama.

Moduliranje kontinuiranog oscilirajućeg signala

Iskustvo je pokazalo da se signal kroz bakrenu žicu može se pouzdano prenositi na znatno veću udaljenost ukoliko on ima oblik kontinuirane oscilirajuće funkcije, na primjer sinusoide s odabranom frekvencijom – vidi Sliku 4.2. Takav signal zove se *nosač* (carrier). Da bi poslao niz bitova, pošiljatelj na osnovu tih bitova lagano modificira nosač. Postupak modificiranja zove se *modulacija*. Primatelj otkriva "nepravilnosti" u nosaču i na taj način reproducira podatke.

Slika 4.2: signal - nosač u obliku sinusoide.

Na primjer, bit 1 može se prikazati tako da se amplituda nosača reducira na 2/3 svog iznosa, a bit 0 tako da se amplituda reducira na 1/3. Tada se na osnovu zadanog niza bitova od pravilnog signala sa Slike 5.2 dobiva modulirani signal prikazan na Slici 4.3.

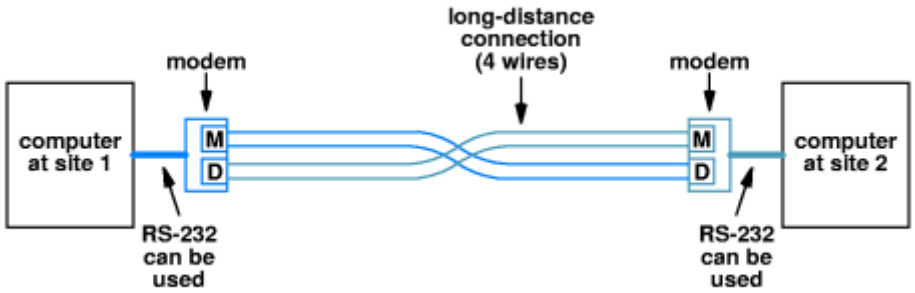


Slika 4.3: (a) niz bitova koje treba prenijeti, (b) modulirani nosač.

U primjeru sa Slike 4.3 koristila se modulacija amplitude (AM). No moguće je koristiti i modulaciju frekvencije (FM). Osim za bakrenu žicu, u osnovi ista ideja modulacije koristi se i za: optička vlakna, radio prijenos i mikrovalni prijenos. Jedina razlika je da nosač više nije električna struja nego svjetlo, odnosno radio val odnosno mikroval određene frekvencije.

Hardver za modulaciju i demodulaciju

Hardverski sklop koji prima niz bitova i na osnovi njega modulira nosač zove se *modulator*. Hardverski sklop koji prima modulirani nosač i na osnovi njega reproducira niz bitova zove se *demodulator*. Oba hardverska sklopa kombiniraju se u jednoj kutiji koja se zove *modem* (modulator i demodulator).



Slika 4.4: dva računala povezana pomoću para full-duplex modema i četiri žice.

Postoji više vrsta modema koji se razlikuju po načinu povezivanja i komuniciranja.

- *Full duplex modem:* omogućuje da dva računala povezana bakrenim žicama istovremeno razmjenjuju podatke u oba smjera – to je takozvana *full duplex* veza. Za to su ukupno potrebna 2 modulatora, 2 demodulatora i 4 žice – vidi Sliku 4.4.

- *Half duplex modem:* omogućuje da 2 žice naizmjenično služe za prijenos bitova u jednom odnosno drugom smjeru.
- *Dial-up modem:* služi za spajanje računala na mrežu preko telefonske linije, koristi nosač koji odgovara slušljivom tonu, simulira neke funkcije telefona, postiže propusnost od 54 Kbit/s.
- *Optički modem:* spaja se na optička vlakna, koristi kao nosač svjetlo određene "boje".
- *Radio modem:* koristi kao nosač radio val određene frekvencije, ugrađuje se u prijenosnike kao sučelje za bežični (wireless) LAN.

Multipleksiranje dijeljenjem frekvencija

Za sve promatrane medije vrijedi sljedeći važni princip: *dva ili više signala koji koriste nosače različitih frekvencija mogu se istovremeno prenositi kroz isti medij bez interferencije.* Riječ je o principu koji se koristi na primjer kod radio programa (moguće je birati razne stanice iz istog "etera") ili kod kabelske televizije (moguće je birati razne programe iz istog kabela).

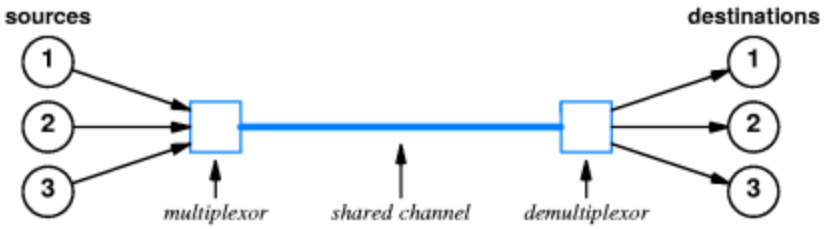
U kontekstu računalnih mreža princip daje metodu kojom više parova računala mogu istovremeno komunicirati kroz isti medij (na primjer kroz istu žicu). Takva metoda istovremenog komuniciranja kroz isti medij naziva se *multipleksiranje dijeljenjem frekvencija*. Njome se postiže veća ukupna propusnost medija, dakle prijenos većeg ukupnog broja bitova na sekundu.

Odabrane frekvencije moraju ipak biti dovoljno razdvojene da među njima ne bi dolazilo do interferencije. Mogućnosti multipleksiranja su dakle ograničene ukupnom *širinom pojasa frekvencija* (bandwidth) koje dotični medij dopušta. Tehnologija koja dopušta veći stupanj multipleksiranja zove se *širokopojasna* (broadband).

Hardver za multipleksiranje dijeljenjem frekvencija

Da bi se opisano multipleksiranje dijeljenjem frekvencija moglo automatski obavljati, potrebni su nam uređaji koji su nešto složeniji od običnog modulatora i demodulatora. Riječ je o sljedećim uređajima koji su ilustrirani na Slici 4.5.

- *Multipleksor* je hardverski sklop koji proizvodi nekoliko nosača različitih frekvencija, modulira svaki nosač s odgovarajućim nizom bitova, te spaja modulirane nosače u jedan signal.
- *Demultipleksor* je hardverski sklop koji prima signal, razlaže ga na modulirane nosače, te reproducira iz njih odgovarajuće nizove bitova.



Slika 4.5: istovremeno komuniciranje kroz isti medij pomoću multipleksora i demultipleksora.

Multipleksiranje dijeljenjem vremena

Alternativna metoda za "istovremeno" komuniciranje više parova pošiljatelja i primatelja kroz isti medij, koja donekle liči na multipleksiranje dijeljenjem frekvencija, zove se *multipleksiranje dijeljenjem vremena*. Kod te metode upotrebljava se samo jedan nosač s odabranom frekvencijom. Pošiljatelji naizmjenično koriste taj isti nosač, svaki u svojim zasebnim vremenskim intervalima.

Primijetimo da se dijeljenjem vremena ne povećava ukupna propusnost medija. Umjesto toga, polazna propusnost raspoređuje se na više parova pošiljatelja i primatelja. Što ima više parova, to svaki od njih trpi sve sporiju komunikaciju.

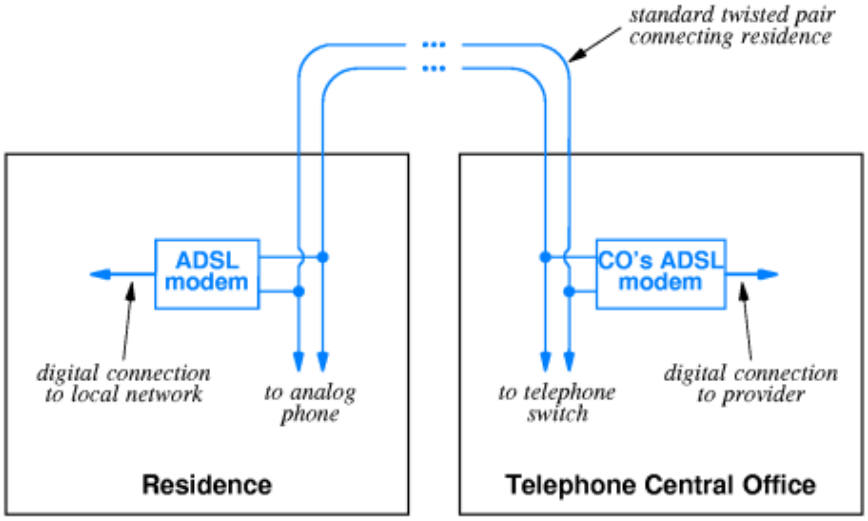
Postoje sljedeće varijante multipleksiranja dijeljenjem vremena.

- *Sinkrono multipleksiranje*. Multipleksor redom dodjeljuje jednako dugačke vremenske intervale prvom, drugom, trećem, …, zadnjem pošiljatelju, pa zatim opet ispočetka.
- *Statističko multipleksiranje*. Slično kao prethodna varijanta, no ako neki od pošiljatelja u danom trenutku nema podataka za slanje, multipleksor ga preskače.

Složeniji oblici multipleksiranja

Osim za istovremenu komunikaciju više parova računala kroz isti medij, multipleksiranje dijeljenjem frekvencija može se upotrijebiti za povećanje propusnosti veze između jednog para računala. Tada se dijelovi istog niza bitova paralelno prenose preko više nosača, pa se time postiže prijenos većeg broja bitova u sekundi.

Primjer takvog multipleksiranja pojavljuje se unutar tehnologije *ADSL* (Asymmetric Digital Subscriber Line) – vidi Sliku 4.6. Riječ je o tehnologiji koja omogućuje širokopojasno spajanje korisnika od kuće na mrežu preko telefonske žice. Klasični ADSL modem stvara 286 nosača, od kojih 255 služi za prijenos od mreže prema korisniku, a 31 za prijenos u obratnom smjeru. Biraju se vrlo visoke frekvencije nosača koje ne interferiraju s glasovnim frekvencijama, tako da se telefon i dalje može koristiti preko iste žice. Klasični ADSL ima propusnost 6.4 Mbit/s prema korisniku, odnosno 640 Kbit/s u obratnom smjeru. Noviji ADSL2+ dostiže i tri do četiri puta veće vrijednosti, no pod uvjetom da su telefonske instalacije kvalitetne.



Slika 4.6: tehnologija ADSL za širokopojasno spajanje na mrežu preko telefonske linije.

Multipleksiranje dijeljenjem frekvencija može se kombinirati s multipleksiranjem dijeljenjem vremena. Dakle svaki nosač s određenom frekvencijom može se dijeljenjem vremena pretvoriti u više virtualnih komunikacijskih kanala. Dobiva se znatno više kanala nego što bi bilo moguće samo na osnovu frekvencija.

Primjer takvog kombiniranog multipleksiranja pojavljuje se unutar tehnologije za spajanje domova na Internet putem infrastrukture kabelske televizije. Za razliku od telefonske žice koja služi samo jednom korisniku, jedan kabel za kabelsku televiziju poslužuje par tisuća pretplatnika. Da bi sve te pretplatnike spojili na Internet, kroz isti kabel mora proći par tisuća komunikacijskih kanala. To je moguće samo ako se ista frekvencija nosača vremenski podijeli na skup pretplatnika. Zbog dijeljenja vremena, propusnost veze na Internet preko kabelske televizije je relativno mala, u svakom slučaju manja nego kod ADSL.

Sažetak Poglavlja 4

Budući da neposredna pretvorba bitova u signal funkcionira samo na kratkim udaljenostima, većina današnjih tehnika za prijenos digitaliziranih podataka kroz komunikacijski medij svodi se na moduliranje kontinuiranih oscilirajućih signala. Za takav prijenos potrebni su posebni hardverski uređaji – modemi. Da bi se omogućila istovremena komunikacija više parova računala kroz isti medij, ili veća propusnost veze između jednog para računala, koristi se multipleksiranje. Za multipleksiranje je potrebna složenija vrsta modema koji se zovu multipleksori.

5. Paketi, okviri, otkrivanje grešaka

Sadržaj Poglavlja 5

Nakon prošlog poglavlja, gdje smo govorili o tome kako se kroz komunikacijski medij šalju pojedini bitovi, u ovom poglavlju detaljnije objašnjavamo kako se kroz mrežu prenose cijele poruke, dakle nizovi bitova. Primjećujemo da se većina današnjih mreža zasniva na dijeljenju poruka na manje cjeline - pakete odnosno okvire, te na prospajanju paketa. Navodimo prednosti i mane korištenja paketa. Primjećujemo da kod prijenosa paketa može doći do grešaka, te obrađujemo tri mehanizma za otkrivanje takvih grešaka.

Pojam paketa

U većini računalnih mreža poruka se ne prenosi kao jedan kontinuirani niz bitova. Umjesto toga, svaka poruka dijeli se u male dijelove koji se zovu *paketi* i koji se šalju zasebno. Dakle pošiljatelj dijeli poruku u pakete, svaki paket putuje nezavisno kroz mrežu, a primatelj skuplja pakete pa ih ponovo sastavlja u poruku.

Zbog upotrebe paketa, mreže računala često se nazivaju *mreže s prospajanjem paketa* (packet switching networks). To je bitna razlika u odnosu na telefonske mreže, koje rade na drukčijem principu i nazivaju se *mreže s prospajanjem linija* (circuit switching networks).

Prednosti upotrebe paketa

Upotreba paketa donosi sljedeće važne prednosti u odnosu na kontinuirano slanje poruka.

- *Efikasnije i pravednije korištenje zajedničkih resursa*. Naime, kad bi se kroz zajednički resurs slale kontinuirane poruke, tada bi jedan par računala mogao zauzeti resurs, a drugi bi morali dugo čekati da dođu na red. Razbijanjem poruka u pakete postiže se vremensko dijeljenje zajedničkog resursa. Dakle računala naizmjenično šalju pakete kroz resurs, ni jedno računalo ne osjeća dugi zastoj. To je ilustrirano Slikom 5.1 gdje računalo A šalje poruku računalu D, a B istovremeno šalje poruku prema C.
- *Mogućnost da paketi paralelno putuju različitim putovima kroz mrežu.* Time se ubrzava prijenos podataka.
- *Lakše ispravljanje grešaka u prijenosu podataka.* Naime, ako se otkrije greška, tada treba ponovo prenijeti samo jedan paket, a ne cijelu poruku.

Slika 5.1: vremensko dijeljenje zajedničkog resursa.

Mane upotrebe paketa

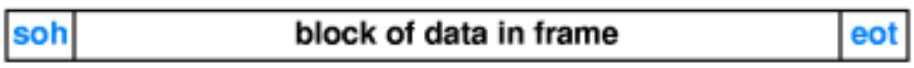
Upotreba paketa ima i sljedeće mane.

- Određeni slojevi protokola moraju se baviti dijeljenjem poruka u pakete, te kasnijim sortiranjem i ponovnim sastavljanjem paketa u poruke.
- Nije moguće garantirati propusnost veze između dva računala. Budući da veza nije ekskluzivno rezervirana za jednu poruku, prijenos podataka može se usporiti zbog dijeljenja vremena s drugim porukama.

Pojam okvira

Svaka mrežna tehnologija definira u detalje kako izgledaju paketi koji se mogu prenositi kroz tu vrstu mreže. Da bi razlikovali općenitu ideju paketnog prijenosa od njene konkretne realizacije, uvodimo pojam okvira. Dakle *okvir* (frame) je paket s precizno definiranim formatom koji se koristi unutar određenog tipa mreže.

Na primjer, neka mrežna tehnologija mogla bi koristiti okvire varijabilne duljine koji se sastoje od ASCII znakova. Pretpostavimo da posebni znakovi soh odnosno eot služe za označavanje početka odnosno kraja okvira. Tada okvir izgleda kao na slici 5.2, dakle sastoji se od stvarnih podataka koje treba prenijeti i od kontrolnih podataka.



Slika 5.2: primjer okvira koji koristi kontrolne znakova za početak i kraj.

Nadijevanje okteta

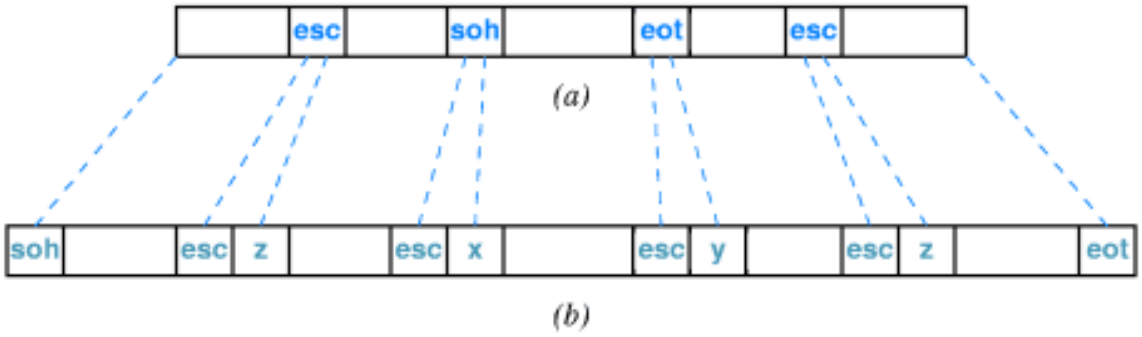
Prethodni format okvira s rezerviranim kontrolnim znakovima obično se ne može izravno primijeniti. Naime okteti (byte-ovi) jednaki kontrolnim znakovima mogu se slučajno pojaviti unutar podataka koji se prenose. Kad bi blok s podacima sadržavao znak eot, primatelj bi ga pogrešno protumačio kao kraj okvira. Slično, znak soh unutar podataka pogrešno bi se tumačio kao početak novog okvira.

Problem razlikovanja stvarnih podataka od kontrolne informacije može se riješiti tehnikom *nadijevanja okteta* (byte stuffing). U skladu s takvom tehnikom, podaci se lagano modificiraju prije slanja, te vraćaju u polazno stanje nakon slanja.

Character In Data	Characters Sent
soh	esc x
eot	esc y
esc	esc z

Slika 5.3: tablica za nadijevanje okteta.

Za naš primjer okvira s dva rezervirana znaka soh i eot, nadijevanje okteta zahtijeva da uvedemo i treći rezervirani znak, na primjer esc. Prije slanja, pošiljatelj prolazi kroz podatke i zamjenjuje pojavu bilo kojeg rezerviranog znaka s kombinacijom od dva znaka prema tablici sa Slike 5.3. Nakon ove zamjene, unutar dijela okvira s podacima više se ne pojavljuju ni soh ni eot, dakle podaci izgledaju kao što je prikazano na Slici 5.4. Primatelj zato može ispravno odrediti početak i kraj okvira i izdvojiti podatke. Da bi reproducirao originalne podatke, primatelj u dijelu okvira s podacima obavlja inverznu zamjenu znakova prema istoj tablici.



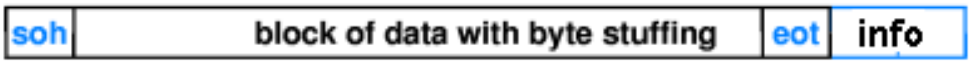
Slika 5.4: (a) podaci prije nadijevanja okteta; (b) isti podaci nakon nadijevanja okteta.

Otkrivanje grešaka u prijenosu

Mediji za prijenos podataka podložni su smetnjama. Kao rezultat smetnji, dešava se da podaci koji putuju mrežom budu izmijenjeni, oštećeni ili izgubljeni. Da bi se greške mogle otkloniti, računalne mreže koriste razne mehanizme za otkrivanje grešaka u prijenosu. Mehanizmi se svode na slanje neke dodatne informacije zajedno s podacima unutar istog okvira. Detaljnije:

- Pošiljatelj računa vrijednost dodatne informacije iz originalnih podataka i umeće je u okvir.
- Primatelj obavlja isto računanje na osnovi primljenih podataka.
- Ako se dvije izračunate vrijednosti razlikuju, očito je došlo do greške u prijenosu.

Ideja je ilustrirana Slikom 5.5. Svi takvi mehanizmi mogu otkriti neke vrste grešaka, no ne daju garanciju da greške nije bilo.



Slika 5.5: otkrivanje greške u prijenosu zasnovano na umetanju dodatne informacije.

Razmotrit ćemo tri mehanizma za otkrivanje grešaka:

- *bitovi za parnost* (parity bits),
- *kontrolni zbrojevi* (checksums),
- *cikličke provjere redundancije* (cyclic redundancy checks – CRC).

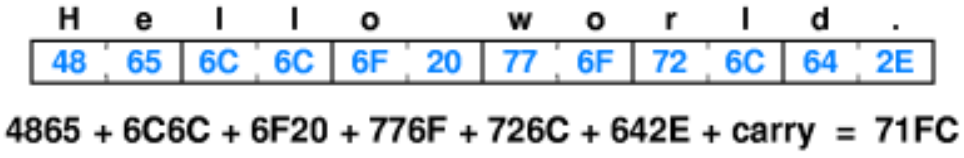
Bitovi za parnost

Riječ je o mehanizmu gdje se dodatna informacija dobiva proširivanjem svakog okteta iz originalnih podataka s još jednim bitom, tako da ukupan broj bitova-jedinica u proširenom oktetu bude paran (ili neparan). Primijetimo da se ista ideja koristila se u staroj 7-bitnoj verziji ASCII koda: budući da se 7-bitni znak zapravo pohranjivao u jednom oktetu, postojao je dodatni osmi bit za provjeru parnosti. Ovaj mehanizam otkriva promjenu jednog bita unutar okteta prilikom prijenosa, no ne otkriva promjenu dva bita.

Kontrolni zbroj

Kod ovog mehanizma podaci unutar okvira promatraju se kao niz cijelih binarnih brojeva određene duljine. Dodatna informacija dobiva se zbrajanjem tih cijelih brojeva i "normalizacijom" zbroja na neku određenu duljinu.

U sljedećem primjeru prikazanom na Slici 5.6, tekst se promatra kao niz 16-bitnih cijelih brojeva, tako da se ASCII kodovi od po dva susjedna znaka shvate kao jedan broj. Zbroj se normalizira na 16 bitova tako da se prijenos ponovo pribroji zbroju.



Slika 5.6: primjer računanja kontrolnog zbroja.

Kontrolni zbroj je pouzdaniji mehanizam od bitova za parnost. Ipak, neke greške i dalje ostaju neotkrivene. U primjeru sa Slike 5.7 promijenila su se 4 bita u podacima, a kontrolni zbroj je ostao isti.

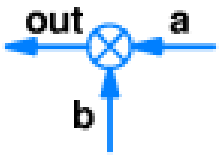
Data Item In Binary	Checksum Value	Data Item In Binary	Checksum Value
0001	1	0011	3
0010	2	0000	0
0011	3	0001	1
0001	1	0011	3
totals	7		7

Slika 5.7: primjer greške u prijenosu koja se ne može otkriti pomoću kontrolnog zbroja.

Cikličke provjere redundancije

Riječ je o mehanizmu gdje se dodatna informacija dobiva računanjem posebnog broja, takozvanog CRC. Računanje se implementira u hardveru kombiniranjem logičkih sklopova za ekskluzivno-ili te posmičnih registara.

Sklop za ekskluzivno-ili, prikazan na Slici 5.8, prima dva bita kao ulaz i daje jedan bit kao izlaz u skladu s tablicom. *Posmični registar* (shift register), prikazan na Slici 5.9, pohranjuje niz bitova. Izvršavanjem operacije posmaka (shifta) novi bit zdesna ulazi u registar, svi bitovi u registru pomiču se za jedno mjesto ulijevo, a bit koji je do tada bio na krajnjem lijevom mjestu se gubi. Registar kao izlaz daje vrijednost trenutnog bita na lijevom kraju.

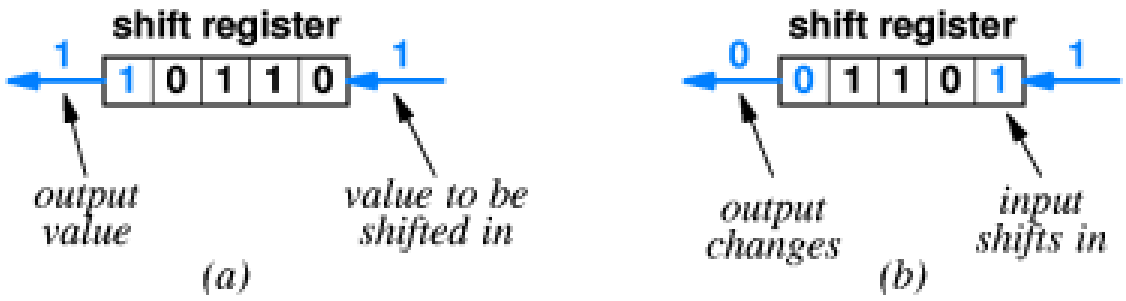


(a)

a	b	out
0	0	0
0	1	1
1	0	1
1	1	0

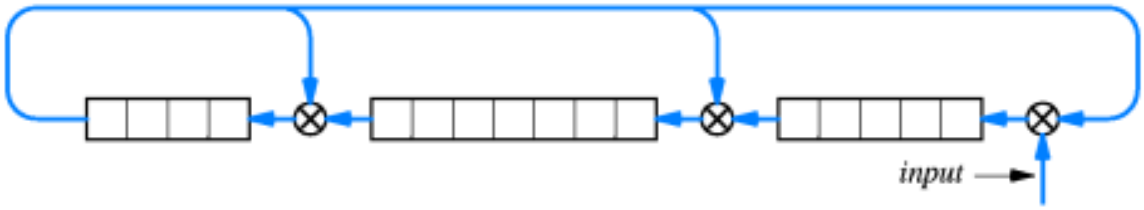
(b)

Slika 5.8: (a) shema sklopa za ekskluzivno ili, (b) tablica logičke operacije ekskluzivno ili.



Slika 5.9: posmični registar, (a) prije posmaka, (b) poslije posmaka.

Sljedeći sklop prikazan na Slici 5.10 računa CRC od 16 bitova. Svi registri najprije se postave na 0, a zatim se podaci iz okvira kao niz bitova uguravaju u sklop nizom operacija posmaka. Pritom svi registri simultano izvode svoje posmake. Nakon što je cijeli niz bitova uguran u sklop, registri sadrže traženi CRC. Matematička analiza pokazuje da CRC otkriva više grešaka od kontrolnog zbroja.



Slika5.10: primjer računanja CRC od 16 bitova pomoću odgovarajućeg sklopa.

Sažetak Poglavlja 5

Za razliku od telefonskih mreža, koje se zasnivaju na prospajanju linija, računalne mreže zasnivaju se na prospajanju paketa. Dakle, poruke se ne šalju u komadu, već se dijele na manje pakete koji neovisno jedan od drugoga putuju mrežom. Glavna prednost upotrebe paketa je efikasnije i pravednije korištenje mrežnih resursa. Svaka konkretna mrežna tehnologija koristi pakete s precizno definiranim formatom koji se zovu okviri. Prilikom prenašanja paketa primjenjuju se razni mehanizmi za otkrivanje grešaka u prijenosu, od kojih je najefikasnija takozvana ciklička provjera redundancije – CRC.

6. LAN tehnologije i struktura mreže

Sadržaj Poglavlja 6

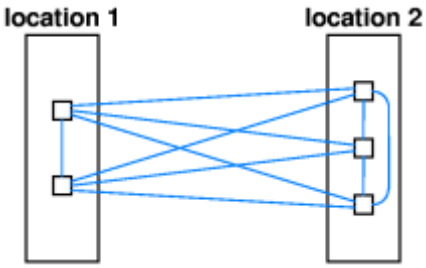
U prethodnim poglavljima uglavnom smo se bavili pitanjem kako uspostaviti vezu između dva računala. Sad nas zanima kako odjednom povezati veći broj računala u lokalnu mrežu - LAN. Najprije ćemo razmotriti mogućnost izravnog povezivanja svakog para računala. Zatim ćemo govoriti o ekonomičnijim LAN tehnologijama koje koriste neku vrstu zajedničkog komunikacijskog medija i time uspostavljaju određenu strukturu međusobne povezanosti računala. Kao tri najvažnije strukture povezanosti spomenut ćemo sabirnicu, prsten i zvijezdu. Za svaku strukturu navest ćemo konkretni primjer LAN tehnologije koja ju koristi.

Potreba za LAN-om

Pretpostavimo da u nekoj zgradi imamo više računala. Tada se prirodno javlja potreba za njihovim povezivanjem. Ta potreba zapravo je rezultat *principa lokalnosti reference* koji kaže: *svako računalo ima tendenciju da češće komunicira s računalima koja su mu fizički blizu, te s onima s kojima je već prije komuniciralo.*
Postavlja se pitanje kako na najbolji način povezati naša računala. Odabrana tehnologija mora osigurati *veliku brzinu* komuniciranja, treba u što većoj mjeri biti *skalabilna*, te razmjerno *jeftina*.

Izravna komunikacija

Najjednostavnija ideja kako povezati računala svodi se na uspostavljanje zasebne veze (žice) između svakog para računala. Ovakvo rješenje ima određenih prednosti, no gotovo se nikad ne primjenjuje u praksi jer je skupo i ne-skalabilno. Naime broj veza potrebnih za takvo povezivanje n računala je $n(n-1)/2$, dakle raste kao n^2. Kod imalo većeg broja računala broj kablova bi bio tako velik da bi imali problema s njihovim fizičkim polaganjem, kao što se to vidi na Slici 6.1.



Slika 6.1: velik broj kablova potreban za izravno povezivanje svih parova računala.

Zajednički komunikacijski mediji

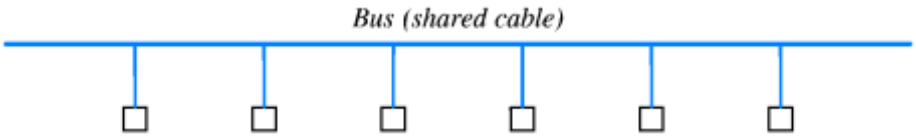
S obzirom da izravno povezivanje računala ne dolazi u obzir, u posljednjih 40-tak godina razvijale su se takozvane LAN tehnologije. Sve su one zasnovane na nekoj vrsti zajedničkog (dijeljenog) komunikacijskog medija. LAN tehnologije pokazale su se dovoljno brze, prilično jeftine, te u većoj ili manjoj mjeri skalabilne. Da bi računala mogla komunicirati preko zajedničkog medija, ona se moraju pokoravati određenim pravilima. Ta pravila osiguravaju da neće doći do kolizije u korištenju medija te da će svako računalo prije ili kasnije ostvariti svoje pravo na komuniciranje.

Struktura lokalne mreže

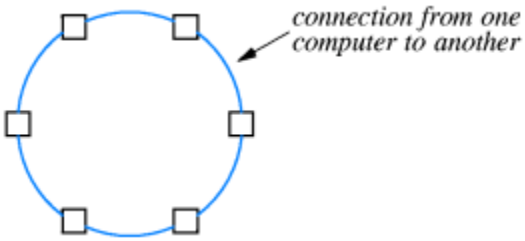
Svaka LAN tehnologija svojim zajedničkim komunikacijskim medijem uspostavlja određenu *strukturu međusobne povezanosti* dijelova opreme. Mnogi ljudi (ali ne matematičari!) tu strukturu nazivaju *topologija* mreže.

U dosadašnjim LAN tehnologijama pojavljivale su se tri različite strukture.

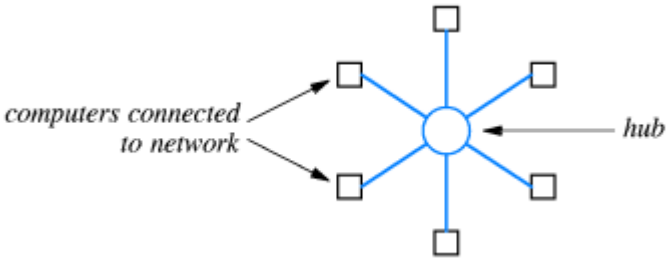
- *Sabirnica* (bus). Sva računala vežu se na jedan dugački kabel – sabirnicu. Poruka putuje tako da ju pošiljatelj pusti kao signal na sabirnicu. Druga računala mogu tada primiti taj signal. Ideja je ilustrirana Slikom 6.2.
- *Prsten.* Prvo računalo vezano je kablom za drugo, drugo za treće, ..., itd, …, zadnje ponovo za prvo. Dakle veze izgledaju kao na Slici 6.3. Poruke putuju u krug, dakle računala ih prosljeđuju u zadanom smjeru.
- *Zvijezda.* Svako računalo vezano je zasebnom vezom do zajedničkog elektroničkog uređaja koji se zove *koncentrator* (hub - glavčina kotača, središte) ili *sklopka* (switch). Situacija ne prikazane na Slici 6.4. Poruke putuju od pošiljatelja, preko koncentratora, do primatelja.



Slika 6.2: struktura lokalne mreže sa sabirnicom.



Slika 6.3: struktura lokalne mreže u obliku prstena.

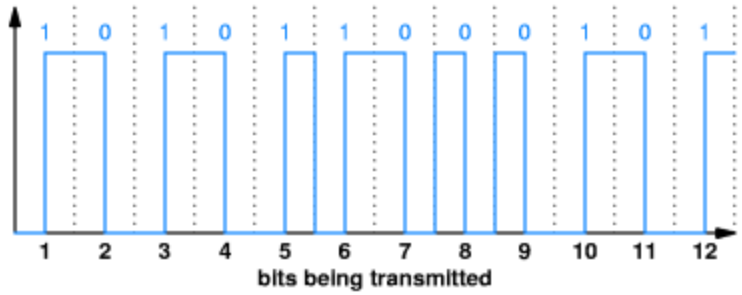


Slika 6.4: struktura lokalne mreže u obliku zvijezde.

LAN sa sabirnicom

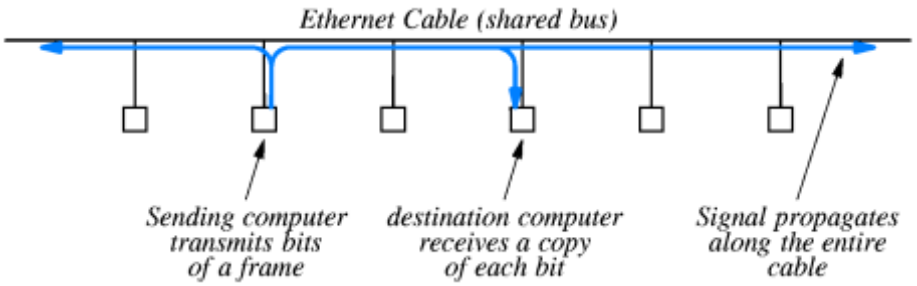
Najpoznatiji primjer LAN tehnologije sa sabirnicom je originalna verzija *Ethernet*-a. Riječ je o tehnologiji koja se razvija od ranih 1970-tih godina (Xerox, DEC, Intel, IEEE), doživjela je nekoliko generacija, te danas dominira tržištem. U originalnoj verziji postojala je sabirnica - koaksijalni kabel zvani *ether*. Taj kabel nije smio biti dulji od 500 m, a spojevi na njega morali su biti udaljeni barem 3 m.

Ethernet standard propisuje format okvira, te način slanja bitova kroz sabirnicu neposrednim pretvaranjem bitova u promjenu napona, po pravilu zvanom *Manchester Encoding* – vidi Sliku 6.5. Dok jedno računalo šalje podatke preko sabirnice, sva ostala čekaju. Pošiljatelj šalje okvir u obliku električnog signala koji se širi od pošiljatelja o oba smjera po kablu. Sva računala "vide" signal. Primatelj iz signala reproducira okvir. Slanje i primanje okvira prikazano je na Slici 6.6.



Slika 6.5: primjer pretvaranja bitova u promjenu napona po pravilu Manchester encoding.

Koordinacija računala koja žele u isto vrijeme slati svoje okvire preko sabirnice odvija se pomoću pravila *CSMA/CD* (Carrier Sense Multiple Access / Collision Detect). Računalo ispituje sabirnicu te započinje slanje okvira tek onda kad na sabirnici nema signala. Ako ipak dva računala počnu slati podatke u isto vrijeme, dolazi do kolizije koju oba pošiljatelja registriraju kao interferenciju na sabirnici. Nakon kolizije svako računalo čeka određeno vrijeme prije nego što pokuša ponovo slati podatke. Vrijeme čekanja bira se slučajno, a kod svake uzastopne kolizije udvostručuje se raspon iz kojeg se obavlja slučajni izbor.



Slika 6.6: komuniciranje računala u Ethernet LAN-u preko sabirnice.

Bežični LAN-ovi

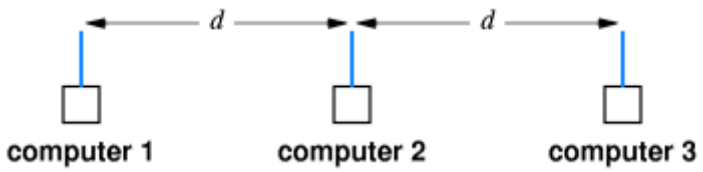
Danas postoje LAN tehnologije za povezivanje računala preko radio valova. Najpoznatiji primjer su tehnologije po standardu IEEE 802.11b ili 802.11g ili 802.11n, također poznate kao *Wi-Fi*. Bežični LAN konceptualno je sličan Ethernet-u. Umjesto sabirnice postoji zajednička radio frekvencija ~2.4 GHz. Koriste se slični okviri.

Opet je potrebna koordinacija računala koja istovremeno pokušavaju slati svoje okvire preko zajedničke frekvencije. Skup pravila zove se *CSMA/CA* (Carrier Sense Multiple Access / Collision Avoidance). Pravila CSMA/CA slična su no malo kompliciranija od CSMA/CD. Naime CSMA/CA mora riješiti dodatne komplikacije koje nastaju kad pošiljatelji nisu u stanju registrirati koliziju.

Na primjer, komplikacija nastaje na sljedećoj Slici 6.7, gdje su računala 1 i 3 previše udaljena da bi mogla međusobno razmjenjivati signale, ali oba još uvijek mogu komunicirati s računalom 2. Ako računala 1 i 3 istovremeno pošalju okvir računalu 2, ni 1 ni 3 neće

primijetiti koliziju. CSMA/CA zato predviđa male kontrolne poruke za najavu ili odobravanje komunikacije.

- Računala 1 i 3 najprije traže od računala 2 dozvolu za komuniciranje.
- Računalo 2 tada šalje dozvolu na primjer računalu 1.
- Ta dozvola vidljiva je i računalu 3, pa ono zna da mora čekati.

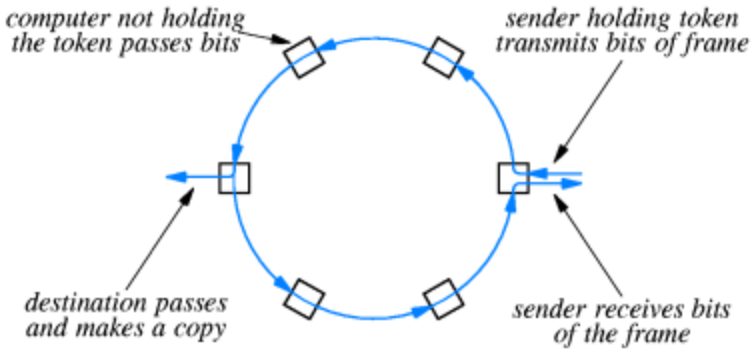


Slika 6.7: teškoće kod otkrivanja kolizije u bežičnom LAN-u.

LAN s prstenom

LAN tehnologije koje koriste povezivanje u obliku prstena bile su popularne u 1980-tim godinama. Najpoznatiji primjer je *IBM Token Ring*. Računala međusobno koordiniraju korištenje prstena služeći se posebnom kratkom porukom koja se zove *žeton* (token). U svakom trenutku u prstenu postoji samo jedan žeton.

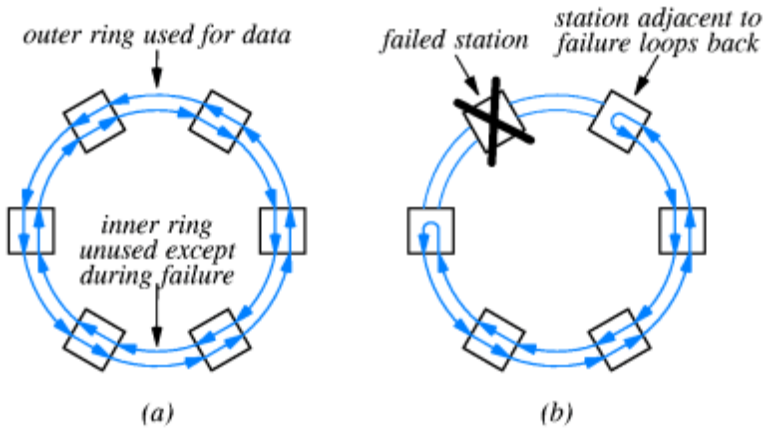
Da bi poslalo podatke, računalo prvo mora čekati da mu stigne žeton, zatim smije odaslati točno jedan okvir, te na kraju treba proslijediti žeton sljedećem računalu. Svi podaci putuju u istom smjeru. Jednom odaslani okvir putuje prstenom sve dok se ne vrati pošiljatelju. Ostala računala ga prosljeđuju, a primatelj ga usput kopira. Na kraju pošiljatelj može provjeriti da li je došlo do greške u prijenosu. Cijeli postupak ilustriran je Slikom 6.8.



Slika 6.8: tok podataka u LAN-u tipa „token ring”.

Da bi poslao sljedeći okvir, pošiljatelj mora čekati da žeton ponovo stigne do njega. U međuvremenu je svako od preostalih računala dobilo šansu za slanje jednog okvira. Računalo koje nema podataka za slanje dužno je odmah proslijediti žeton. Ako nitko ne šalje podatke, žeton kruži prstenom velikom brzinom.

Mana LAN-a s prstenom je da se komunikacija prekida čim jedno od računala ne radi. Postoji varijanta s dvostrukim prstenom, gdje se mreža re-konfigurira u slučaju kvara jednog računala. Način konfiguracije prikazan je na Slici 6.9.

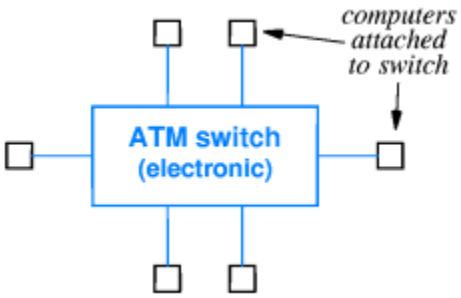


Slika 6.9: LAN s dvostrukim prstenom, (a) kad sva računala rade, (b) ako je jedno ne radi.

LAN u obliku zvijezde

Važan primjer LAN tehnologije koja koristi zvijezdu razvile su telefonske kompanije pod nazivom *ATM* (Asynchronous Transfer Mode). U središtu ATM mreže nalazi se jedan ili više elektroničkih uređaja koji se zovu *ATM sklopke* (ATM switch) – vidi Sliku 6.10.

Zbog brze dvosmjerne komunikacije, svako računalo izravno se spaja na ATM sklopku pomoću dvostrukog optičkog vlakna. Za razliku od sabirnice ili prstena, ATM sklopka ne distribuira podatke svim računalima, nego ih samo prebacuje od pošiljatelja do primatelja. U slučaju kvara jedne veze ili jednog računala ostatak ATM mreže radi dalje.



Slika 6.10: LAN u obliku zvijezde zasnovan na tehnologiji ATM.

U vrijeme svog nastanka početkom 1990-ih godina, ATM se isticao po visokoj propusnosti. Veza između računala i ATM sklopke osiguravala je propusnost od 155 Mbit/s ili više. Sredinom 1990-tih godina mislilo se da je ATM najperspektivnija LAN tehnologija koja će zavladati tržištem. Ipak, to se nije dogodilo zbog pojave gigabitne verzije Ethernet-a.

Sažetak Poglavlja 6

Povezivanje računala u LAN nije moguće postići zasebnim vezama između svakog para računala jer bi to bilo preskupo i ne-skalabilno. Zato sve upotrebljive LAN tehnologije koriste neki oblik zajedničkog komunikacijskog medija, te time uspostavljaju određeni oblik strukture međusobne povezanosti u LAN-u. Komuniciranje preko zajedničkog medija je jeftino, skalabilno i dovoljno brzo, no ono zahtijeva od sudionika da se pokoravaju određenim pravilima za sprečavanje konfliktnih radnji. Od tri poznata oblika strukture LAN-a, a to su sabirnica, prsten i zvijezda, danas se gotovo isključivo koristi zvijezda u čijem središtu se nalazi neka vrsta koncentratora ili sklopke.

7. Hardversko adresiranje i utvrđivanje tipova okvira u LAN-u

Sadržaj Poglavlja 7

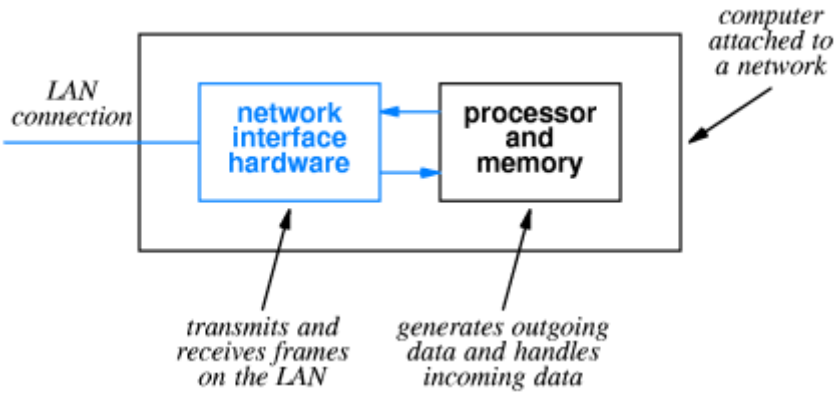
U prethodnim poglavljima govorili smo o LAN-ovima i okvirima. Sad ćemo detaljnije objasniti kako okviri u LAN-u pronalaze svoj put od pošiljatelja do primatelja. Najprije uvodimo pojam hardverske adrese računala, navodimo načine kako se te adrese mogu dodijeliti računalima, te spominjemo neke dodatne adrese koje služe za posebne vrste prijenosa podataka. Primjećujemo da svaki okvir uz ostale podatke mora sadržavati adresu pošiljatelja i primatelja - kao primjer za to specificiramo format okvira u Ethernet-u. Opisujemo ulogu LAN sučelja u slanju i primanju okvira. Usput se bavimo i nekim dodatnim pitanjima kao što je utvrđivanje sadržaja okvira, odnosno analiza performansi LAN-a.

Potreba za adresama

U većini LAN-ova paketi putuju kroz zajednički medij te su vidljivi svim spojenim računalima. Javlja se problem: kako ostvariti prijenos okvira od pošiljatelja *točno određenom* primatelju? Rješenje se zasniva na dodjeljivanju takozvanih *hardverskih (fizičkih) adresa* računalima. Svaki okvir uz ostale podatke mora sadržavati adresu pošiljatelja te adresu primatelja. Prilikom slanja okvira, pošiljatelj upisuje u okvir svoju vlastitu adresu te adresu računala kojem se okvir šalje. Računalo spojeno na LAN ispituje adrese unutar svakog okvira koji prolazi mrežom, prihvaća (kopira) one gdje se adresa primatelja poklapa s njegovom vlastitom adresom, te ignorira ostale.

Uloga LAN sučelja

Da središnja jedinica računala ne bi bila opterećena poslom stalnog praćenja prometa po mreži, u računalo se ugrađuje posebni hardverski sklop – *mrežno* ili *LAN sučelje* (mrežna kartica). LAN sučelje je snažan i samostalan uređaj koji radi bez pomoći procesora i memorije u računalu. Njegova zadaća je da se brine za sve detalje vezane uz slanje i primanje okvira, kao što je ilustrirano Slikom 7.1.



Slika 7.1: uloga LAN sučelja kod slanja i primanja okvira.

Prilikom slanja podataka, središnja jedinica računala šalje okvir svojem LAN sučelju i zahtijeva slanje. Nakon toga središnja jedinica može nastaviti s izvršavanjem aplikacijskog programa, a LAN sučelje čeka na pristup zajedničkom mediju i šalje okvir. Primanje podataka odvija se tako da LAN sučelje prati sve okvire koji putuju zajedničkim medijem, filtrira one s ispravnim CRC i odgovarajućom adresom primatelja, te ih prosljeđuje središnjoj jedinici. Dakle zahvaljujući LAN sučelju središnja jedinica je izolirana od većine aktivnosti na mreži, te ima posla samo s podacima koji se nje izravno tiču.

Dodjeljivanje adresa

Unutar jednog LAN-a svako računalo mora imati jedinstvenu adresu. Postoje tri sheme za dodjeljivanje adresa računalima.

- *Statičko dodjeljivanje.* Koristi se adresa koji je proizvođač LAN sučelja ugradio u svoj uređaj i koja je jedinstvena na cijelom svijetu.
- *Konfigurabilno dodjeljivanje.* Administrator mreže svakom računalu postavlja adresu koju je sam izabrao. Postavljanje adrese se obavlja pomoću sklopki na LAN sučelju ili upisivanjem u EPROM sučelja.
- *Dinamičko dodjeljivanje.* Računalo automatski bira adresu svaki puta kad se upali. Obično je riječ o biranju slučajnih brojeva, sve dok se ne pogodi slobodna adresa.

Osobina statičkog dodjeljivanja je da je adresa računala stalna, čak i onda kad ga selimo iz mreže u mrežu, sve dok mu ne promijenimo LAN sučelje. Također, uređaji raznih proizvođača mogu se odmah bez podešavanja adresa uključiti u istu mrežu. Svojstvo dinamičkog dodjeljivanja je da eliminira potrebu da proizvođači hardvera koordiniraju svoje adrese. Također, dinamičke adrese mogu biti znatno kraće od statičkih. Konfigurabilne adrese su kompromis između statičkih i dinamičkih. Slično kao statičke, one su relativno stalne. Slično kao dinamičke, one mogu biti kratke.

Difuzija (broadcasting)

Difuzija (broadcasting) je prijenos podataka gdje jedno računalo šalje iste podatke svim drugim računalima u mreži. U većini LAN tehnologija difuzija se može efikasno izvesti zato što podaci ionako putuju zajedničkim medijem i "vidljivi" su svim računalima. Uz postojeće adrese računala u LAN-u, uvodi se i dodatna rezervirana *difuzijska adresa* (broadcast address). LAN sučelje u svakom računalu prepravlja se tako da filtrira ne samo okvire čija adresa primatelja je jednaka adresi tog računala, nego i okvire čija adresa primatelja je

jednaka difuzijskoj adresi. Dakle ako okvir pošaljemo na difuzijsku adresu, svako računalo u mreži primit će kopiju tog okvira.

Difuzija u grupi (multicasting)

Difuzija u grupi (multicasting) je nešto između običnog prijenosa podataka i difuzije. Jedno računalo šalje iste podatke grupi "pretplaćenih" računala. U većini LAN tehnologija, difuzija u grupi može se efikasno izvesti na sličan način kao difuzija. Uvode se dodatne *adrese za difuziju u grupi* (multicast adresses). Svaka od tih adresa odgovara jednoj grupi računala. LAN sučelje računala koje je uključeno u grupu podešava se tako da osim vlastite i difuzijske adrese "prepoznaje" i dotičnu adresu za difuziju u grupi. Unos ili brisanje adrese za difuziju u grupi u LAN sučelju izvodi se dinamički, tako da aplikacijski program koji se izvršava na računalu pošalje odgovarajuću instrukciju sučelju.

Utvrđivanje sadržaja okvira

Iz samog sadržaja okvira teško je zaključiti koja vrsta podataka se nalazi u tom okviru. Na primjer, okviri koji nose e-mail poruke, tekstualne datoteke ili web stranice svi sadrže ASCII znakove. Da bi primatelj mogao odrediti vrstu nekog okvira, potrebna je dodatna informacija u samom okviru.

Postoje dvije metode za utvrđivanje sadržaja.

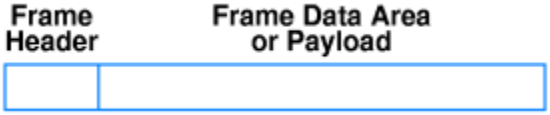
- *Eksplicitno navođenje tipa okvira*. Sama mrežna tehnologija predviđa da se u formatu okvira nalazi posebno polje za tip okvira. Također, sama tehnologija svojim standardima definira identifikatore za neke tipove okvira.
- *Implicitno navođenje tipa okvira.* Korištena mrežna tehnologija u svom formatu okvira ne predviđa polje za tip. Pošiljatelj i primatelj dogovaraju se da će razmjenjivati samo jednu vrstu sadržaja. Ili se dogovaraju da će polje za tip okvira sami uključiti na određeno mjesto u dio okvira koji je inače predviđen za podatke.

Obje metode imaju prednosti i mane. Eksplicitno navođenje je pouzdanije, no obuhvaća samo one tipove okvira koji su prepoznati i standardizirani na razini dotične mrežne tehnologije. Implicitno navođenje je fleksibilnije no lako može dovesti do nesporazuma.

Zaglavlje i korisni teret okvira

Vidjeli smo da osim stvarnih podataka okvir mora sadržavati i niz dodatnih informacija. Zbog toga je u stvarnim LAN tehnologijama format okvira kompliciraniji od onog iz Poglavlja 5. U većini tehnologija, okvir se u skladu sa Slikom 7.2 može podijeliti na:

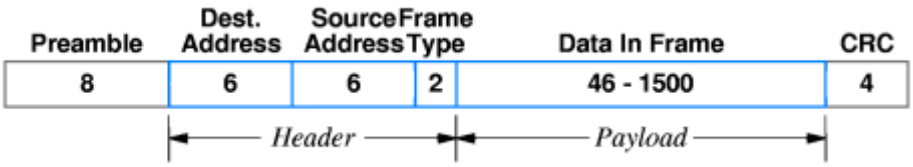
- *Zaglavlje*, koje sadrži dodatne informacije poput adresa, tipova i slično.
- *Korisni teret* (payload) ili područje za podatke, gdje se nalaze podaci koji se šalju.



Slika 7.2: općeniti format za okvir koji putuje LAN-om.

Primjer formata okvira

Kao konkretni primjer, navodimo format okvira koji se koristi u Ethernet-u. Taj format sastoji se od šest polja, kao što je prikazano na Slici 7.3.



Slika 7.3: format okvira koji se koristi u Ethernet-u.

Slijedi detaljni opis građe Ethernet-ovog okvira.

- Ethernet-ov okvir počinje 64-bitnim *predgovorom* (preamble) koji se sastoji od alternirajućih nula i jedinica i koji služi da bi se primateljev hardver mogao sinkronizirati s dolazećim signalom.
- Dalje slijede 48-bitne *adrese primatelja i pošiljatelja*. Ethernet koristi statičko dodjeljivanje adresa, naime koristi se činjenica da svako LAN sučelje ima jedinstvenu adresu koju je u njega ugradio proizvođač. Adresa 111…11 je rezervirana za difuziju, a druge adrese koje počinju s 1 služe za difuziju u grupi.
- Treće polje zaglavlja je 16-bitni Ethernet-ov *tip okvira*. Ethernet standard definira nekoliko stotina tipova okvira, od kojih je nekoliko navedeno u tablici na Slici 7.4. Uglavnom je riječ o tipovima koje koriste sustavi stvoreni u velikim kompanijama.
- Najveći dio Ethernet-ovog okvira zauzimaju *podaci*, dakle *korisni teret* (payload). Duljina nije fiksirana.
- Na kraju okvira nalazi se CRC izračunat onako kako smo objasnili u Poglavlju 5.

Analiziranje performansi LAN-a

Rekli smo da u većini LAN-ova paketi putuju kroz zajednički medij te su "vidljivi" svim računalima. Zbog toga je vrlo jednostavno napraviti *mrežni analizator* (network analyzer). Riječ je o uređaju koji prati događaje u mreži i računa statistike poput prosječnog broja okvira u sekundi, prosječne veličine okvira, broja kolizija u nekom vremenskom intervalu, i slično.

Da bi napravili mrežni analizator dovoljno nam je osobno računalo ili prijenosnik sa standardnim LAN sučeljem i odgovarajućim softverom. Da bi mogli pratiti sve okvire koji prolaze mrežom, LAN sučelje našeg uređaja moramo staviti u takozvani *promiskuitetni* način rada, dakle način gdje se preskače uobičajeno filtriranje okvira po adresi. Promiskuitetni način rada podržan je u svim standardnim komercijalno dobavljivim mrežnim karticama za osobna računala.

Iz svega ovog je vidljivo da podaci koji putuju LAN-om apriori nisu zaštićeni od neovlaštenog čitanja. Svaki korisnik s računalom spojenim na LAN u pravilu vrlo lako može čitati tuđe poruke. Da bi ipak zaštitili podatke, moramo se služiti suptilnijim metodama kao što je kriptiranje. O tome će biti riječi u Poglavlju 25.

Value	Meaning
0000-05DC	Reserved for use with IEEE LLC/SNAP
0800	Internet IP Version 4
0805	CCITT X.25
0900	Ungermann-Bass Corporation network debugger
0BAD	Banyan Systems Corporation VINES
1000-100F	Berkeley UNIX Trailer encapsulation
6004	Digital Equipment Corporation LAT
6559	Frame Relay
8005	Hewlett Packard Corporation network probe
8008	AT&T Corporation
8014	Silicon Graphics Corporation network games
8035	Internet Reverse ARP
8038	Digital Equipment Corporation LANBridge
805C	Stanford University V Kernel
809B	Apple Computer Corporation AppleTalk
80C4-80C5	Banyan Systems Corporation
80D5	IBM Corporation SNA
80FF-8103	Wellfleet Communications
8137-8138	Novell Corporation IPX
818D	Motorola Corporation
FFFF	Reserved

Slika 7.4: neki od standardnih tipova okvira u Ethernet-u.

Sažetak Poglavlja 7

Da bi okviri u LAN-u na ispravan način putovali od pošiljatelja do primatelja, nužno je da sva računala imaju svoje hardverske adrese, te da se u svakom okviru navodi adresa pošiljatelja i primatelja. To znači da format okvira u stvarnim LAN tehnologijama, osim samih podataka koji čine koristan teret, također mora predvidjeti i dodatne informacije. Dobar primjer za to je Ethernet-ov okvir, gdje se uz same podatke i CRC zaista nalaze i obje adrese, te informacija o tipu sadržaja okvira. Posao praćenja okvira koji u LAN-u prolaze zajedničkim medijem prilično je zahtjevan, tako da ga računala prepuštaju posebnim hardverskim sklopovima koji se zovu mrežna sučelja. Način slanja i primanja okvira kroz LAN omogućuje lagano analiziranje performansi LAN-a, no nosi u sebi određene sigurnosne rizike.

8. Ožičenje i fizička struktura LAN-a

Sadržaj Poglavlja 8

U ovom poglavlju detaljno se upoznajemo s hardverskim aspektima danas najpopularnije LAN tehnologije, a to je Ethernet. Najprije opisujemo mrežno sučelje za Ethernet koje se sastoji od mrežne kartice i (eventualno) transcievera. Dalje govorimo o ožičenju, i to posebno za svaku od tri dosadašnje generacije Ethernet-a: Thick Ethernet, Thin Ethernet, TP Ethernet.

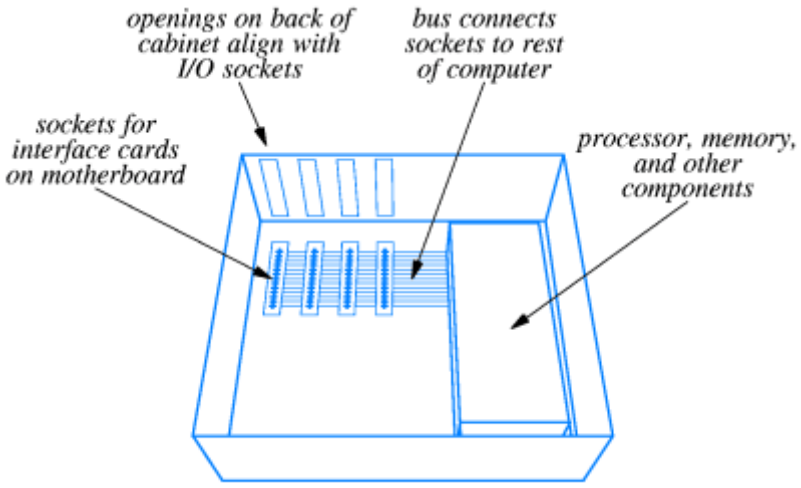
Na kraju dajemo usporedbu opisanih ožičenja, te služeći se pokazanim primjerima raspravljamo u razlici između fizičke i logičke strukture LAN-a.

Mrežne kartice i transcieveri

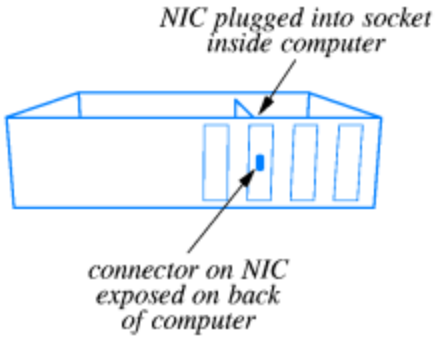
U prethodnom poglavlju objasnili smo da se računalo spaja na LAN pomoću posebnog hardverskog sklopa – mrežnog ili LAN sučelja. LAN sučelje preuzima poslove praćenja prometa na mreži, slanja i primanja okvira, te tako rasterećuje središnju jedinicu računala. Zahvaljujući takvoj raspodjeli poslova, današnji LAN-ovi rade na ogromnoj brzini, znatno brže nego što bi središnji procesori mogli slijediti. Danas je uobičajeno da LAN postiže propusnost od barem 1 Gbit/s.

LAN sučelje za Ethernet obično je realizirano kao *mrežna kartica* (network adapter card, network interface card – NIC) koja izgleda poput one na Slici 8.1. Mrežna kartica se utakne u utor (slot) na matičnoj ploči računala, kao što se vidi na Slici 8.2. Dio kartice koji viri na poleđini računala sadrži utičnicu (konektor) za kabel kojim će se računalo spojiti na mrežu – vidi Sliku 8.3. U nekim starijim varijantama Ethernet tehnologije postojao je i dodatni uređaj – *transciever*, koji se spajao između mrežnog medija i mrežne kartice. Posao mrežnog sučelja tada je bio podijeljen između transcievera i mrežne kartice, tako da je transciever obavljao analogni a mrežna kartica digitalni dio posla.

Slika 8.1: izgled mrežne kartice.



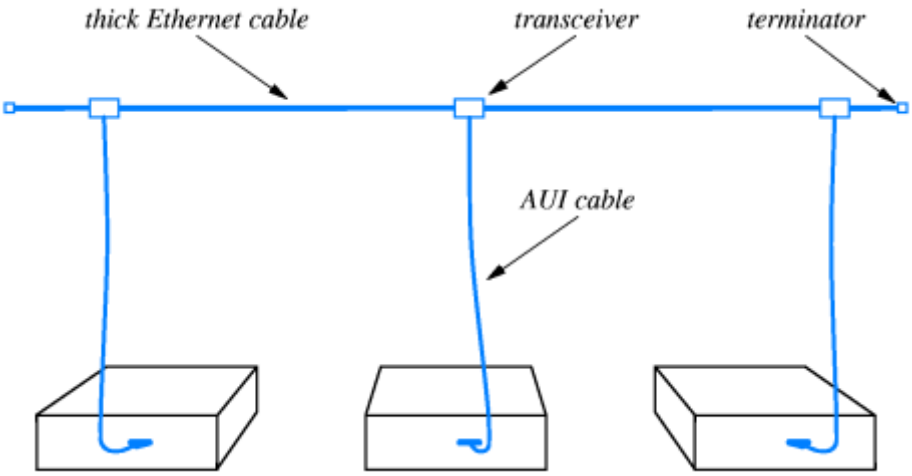
Slika 8.2: utori na matičnoj ploči računala za umetanje mrežne kartice.



Slika 8.3: stražnja strana računala s umetnutom mrežnom karticom.

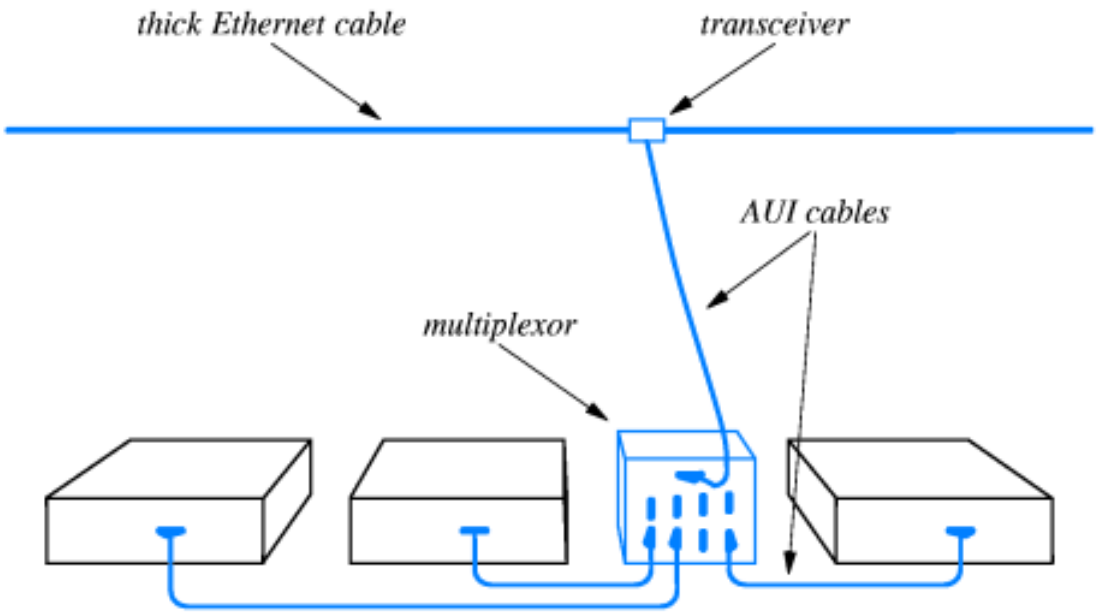
Ožičenje za Thick Ethernet

Sjetimo se da je osnovno svojstvo Ethernet tehnologije korištenje strukture LAN-a sa sabirnicom. Prva generacija Etherneta zvala se *Thick Ethernet* ili *Thicknet* ili *10Base5*. Mrežna sabirnica postojala je u fizičkom smislu i bila je realizirana kao debeli koaksijalni kabel koji se polagao daleko od računala. Mrežno sučelje svakog računala sastojalo se od mrežne kartice i transcievera. Pritom je transciever bio na mrežnom kabelu i spajao se s mrežnom karticom pomoću takozvanog AUI kabla. Shematski prikaz spajanja računala u Thick Ethernet vidi se na Slici 8.4.



Slika 8.4: tri računala povezana na Thick Ethernet.

U Thick Ethernet tehnologiji postojala su graničenja da duljina Ethernet kabla ne smije prijeći 500 metara, te da dva transcievera moraju biti udaljena najmanje 3 metra. Da bi se unatoč takvim ograničenjima povećao broj spojenih računala, kasnije je uveden *multipleksor* – uređaj koji omogućuje spajanje više računala na isti transciever – vidi Sliku 8.5.

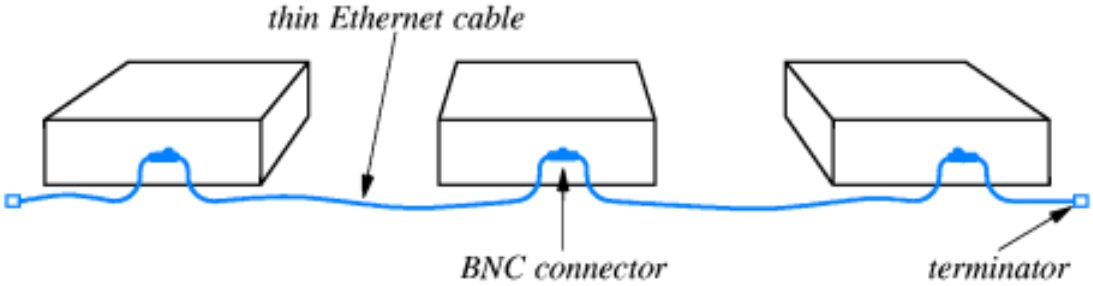


Slika 8.5: spajanje većeg broja računala na Thick Ethernet pomoću multipleksora.

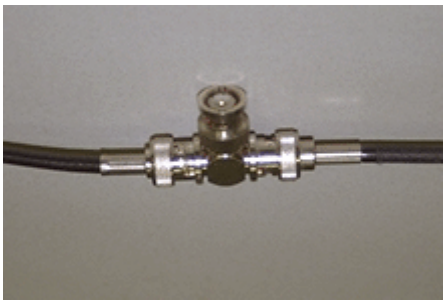
Ožičenje za Thin Ethernet

Druga generacija Etherneta zvala se *Thin Ethernet* ili *Thinnet* ili *10Base2*. Mrežna sabirnica i dalje je fizički postojala, no bila je realizirana kao tanki savitljivi koaksijalni kabel koji se polagao od računala do računala. Mrežno sučelje sastojalo se samo od mrežne kartice koja je preuzela i funkciju transcievera. Spoj računala na koaksijalni kabel ostvarivao se takozvanim BNC konektorom.

Slika 8.6 daje shematski prikaz spajanja računala u Thin Ethernet. Izgled BNC konektora odnosno tankog koaksijalnog kabela vidljiv je na Slici 8.7. Iduća Slika 8.8 bilježi stvarni prizor umrežavanja tri računala – opet se vidi tanki kabel koji se savija od računala do računala, te pripadni BNC konektori utaknuti u mrežne kartice računala.



Slika 8.6: spajanje računala na Thin Ethernet.



Slika 8.7: izgled tankog Ethernet kabla i BNC konektora.

Slika 8.8: tri računala povezana BNC konektorima na Thin Ethernet.

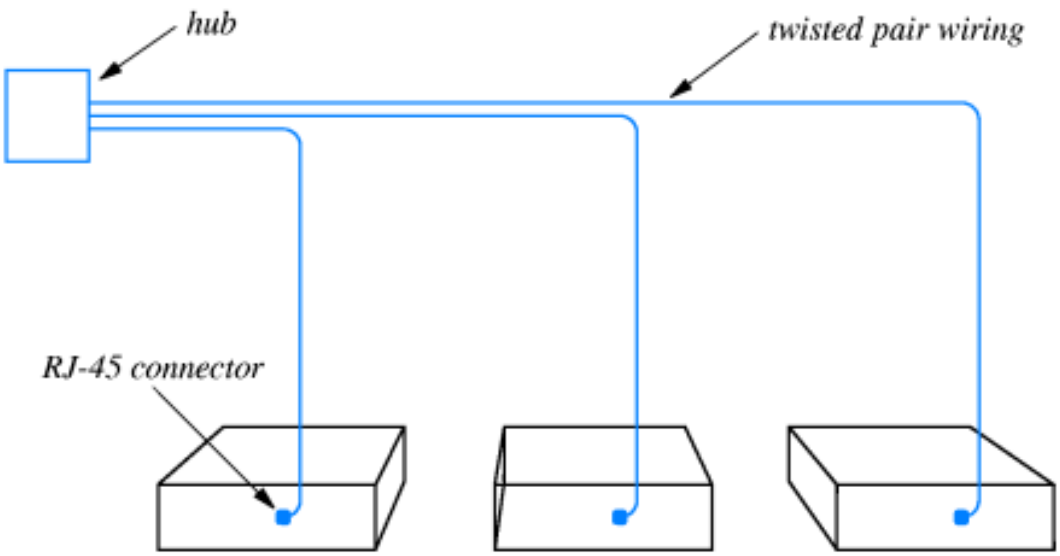
Ožičenje za TP Ethernet

Današnja generacija Etherneta zove se *Twisted Pair Ethernet* ili *TP Ethernet*. Funkciju sabirnice preuzima posebni elektronički uređaj – *koncentrator* (hub). Svako računalo vezano je zasebnom paricom (twisted pair žicom) na koncentrator. Mrežno sučelje izvedeno je kao mrežna kartica s RJ-45 konektorom (sličan kao za telefon). Elektronika u koncentratoru simulira ponašanje sabirnice, tako da cijeli sustav radi slično kao prethodna generacija Etherneta.

TP Ethernet vremenom je povećavao brzinu rada, tako da postoje tri verzije:

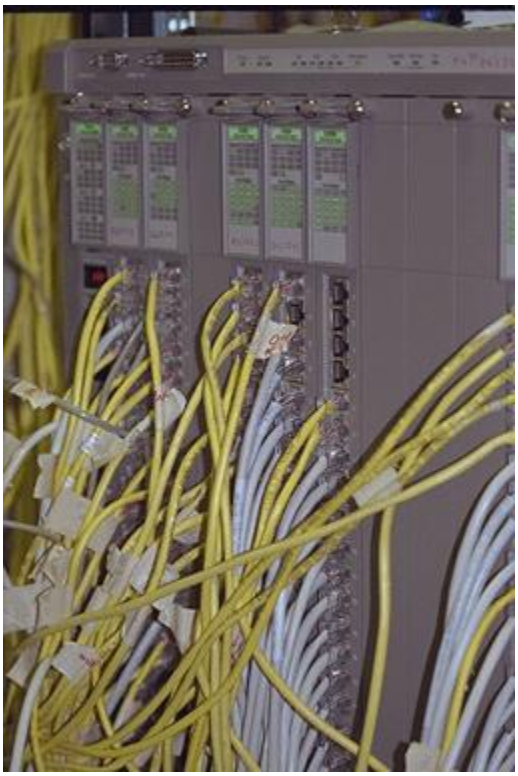
- *Obični TP Ethernet* – 10BaseT, propusnost 10 Mbit/s
- *Fast Ethernet* – 100BaseT, propusnost 100 Mbit/s
- *Gigabit Ethernet* – 1000BaseT, propusnost barem 1 Gbit/s, danas dostiže i 10 Gbit/s.

Slika 8.9 shematski prikazuje spajanje računala u TP Ethernet. Na Slikama 8.10 i 8.11 vidimo stvarni izgled RJ-45 konektora, TP kablova, poleđine računala s mrežnom karticom i koncentratora.



Slika 8.9: spajanje računala na TP Ethernet.

Slika 8.10: poleđina računala s mrežnom karticom i TP kablom za spoj na TP Ethernet.



Slika 8.11: Koncentrator (hub) za TP Ethernet sa uključenim TP kablovima.

Usporedba raznih ožičenja

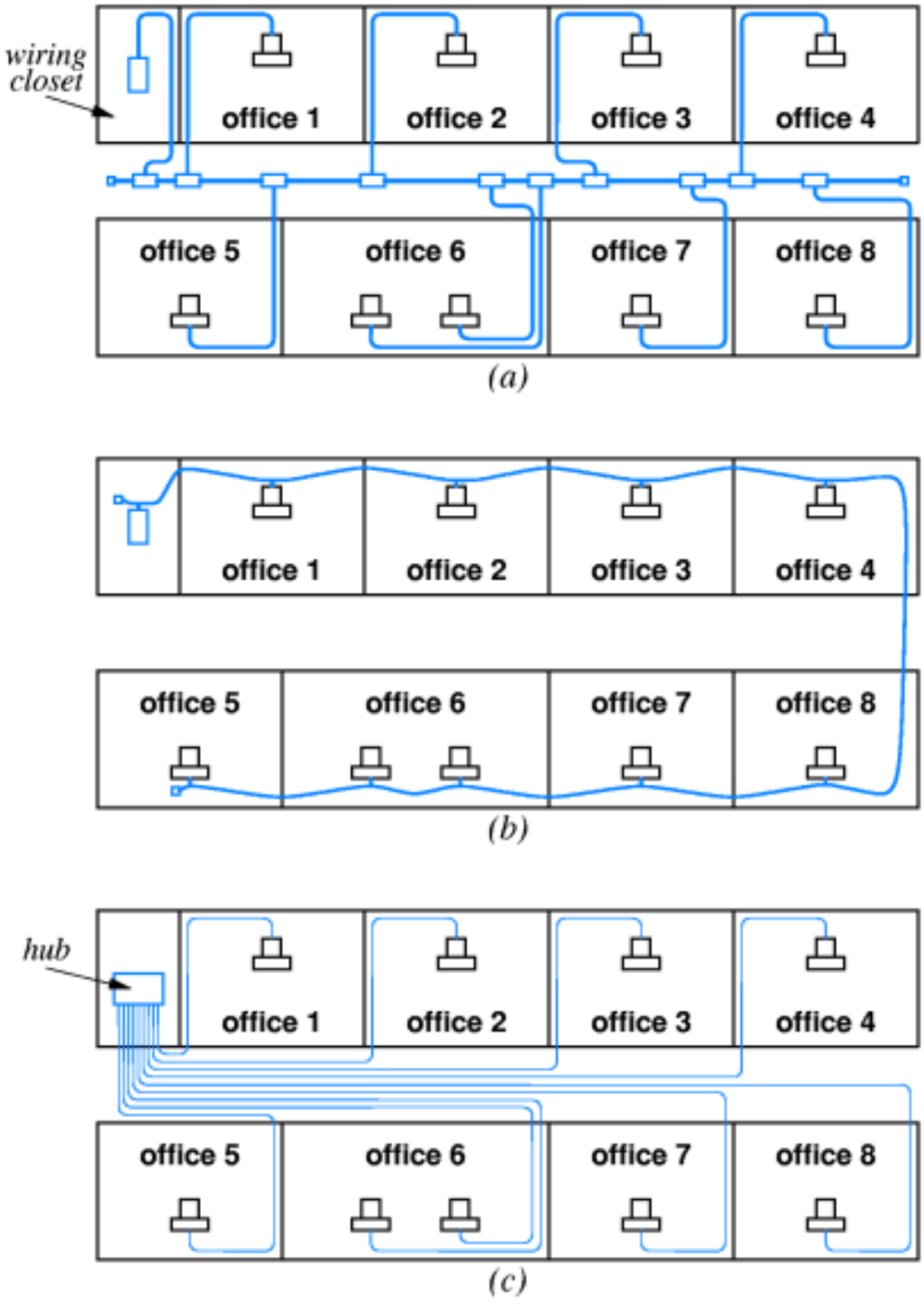
Ožičenje s debelim kablom i transcieverima dozvoljavalo je da se pojedino računalo ukloni, a da to ne poremeti mrežu. No, transcieveri su obično bili na nedostupnom mjestu gdje ih je bilo teško popravljati. Ožičenje s tankim kablom bilo je jeftinije od onoga s debelim kablom. No tanka mreža se lagano prekidala čim je netko otkopčao jedan BNC konektor. TP Ethernet je suvremeno rješenje koje je omogućilo velike brzine i propusnosti. Svaki stroj ima svoj TP kabel, tako da uklanjanje jednog stroja ne može prekinuti ostatak mreže. No, TP Ethernet zahtijeva polaganje velikog broja žica koje izlaze iz koncentratora.

Na Slici 8.12 su prikazan je primjer gdje je istih devet uredskih prostorija umreženo na tri različita načina.

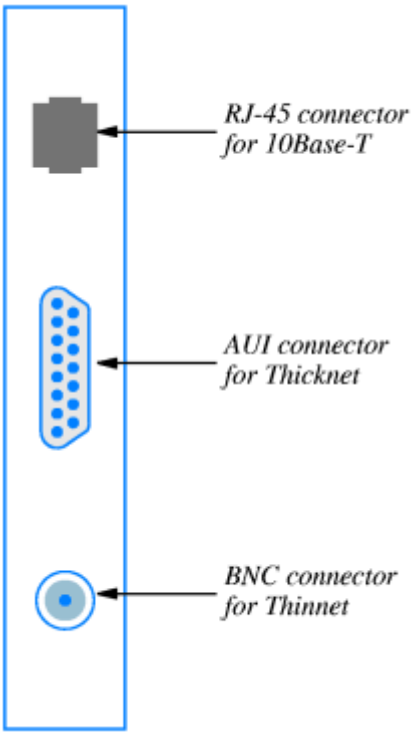
- Najprije pomoću Thick Ethernet-a,
- zatim pomoću Thin Ethernet-a,
- na kraju kao TP Ethernet.

"Wiring closet" je prostorija za smještaj zajedničkog koncentratora ili neke druge opreme za praćenje rada mreže.

Svaka vrsta ožičenja koristi drukčije mrežne kartice i drukčije konektore na tim mrežnim karticama. Postoje i kombinirane kartice s više vrsta konektora. Još nedavno, bile su raširene kartice s konektorima za Thin Ethernet i za TP Ethernet. Točan izgled konektora pojedinih vrsta vidljiv je na Slici 8.13.



Slika 8.12: ista računala spojena preko (a) Thick, (b) Thin, (c) TP Ethernet-a.



Slika 8.13: mrežna kartica za Ethernet s konektorima za tri različite vrste ožičenja.

Fizička struktura LAN-a

Do sada smo govorili da je Ethernet LAN tehnologija sa strukturom sabirnice. Ipak, vidjeli smo da suvremeni TP Ethernet koristi spajanje računala preko koncentratora. Kako je to moguće? Da li to znači da se u TP Ethernet-u prešlo na strukturu zvijezde?

Odgovor na ovo pitanje zahtijeva da uočimo razliku između logičke i fizičke strukture LAN-a. Zadana LAN tehnologija može koristiti razne načine ožičenja. Sama tehnologija određuje logičku strukturu LAN-a, a način ožičenja određuje fizičku strukturu. Moguće je da se fizička struktura razlikuje od logičke.

TP Ethernet ima fizičku strukturu zvijezde, no on zadržava logičku strukturu sabirnice. Naime, koncentrator unatoč svom zvjezdastom načinu povezivanja u potpunosti simulira ponašanje sabirnice. Na primjer, kad neko računalo pošalje okvir, tada koncentrator pušta odgovarajući signal po svim TP kablovima, tako da svako računalo "vidi" taj signal onako kako bi ga vidjelo na sabirnici. I dalje su moguće kolizije. Primjenjuju se ista CSMA/CD pravila za pristup sabirnici i postupanje u slučaju kolizije. Zbog ovakvih svojstava, TP Ethernet popularno se naziva i "sabirnica u obliku zvijezde" (star-shaped bus) ili "sabirnica u kutiji" (bus in a box).

Sažetak Poglavlja 8

Ethernet je trenutno prevladavajuća tehnologija za povezivanje računala u LAN. Od svojih početaka u 1970-tim godinama pa do danas, ta tehnologija prošla je kroz tri generacije.

Sadašnja generacija naziva se TP Ethernet, i ona se od 1990-tih godina do danas usavršavala u pogledu propusnosti, tako da razlikujemo njene tri verzije: obični TP Ethernet, brzi Ethernet i gigabitni Ethernet.

Osnova Ethernet tehnologije je struktura LAN-a sa sabirnicom. Dok je u prve dvije generacije Ethernet-a sabirnica zaista fizički postojala u obliku debelog odnosno tankog koaksijalnog kabla, u TP Ethernet-u sabirnica je postala virtualna, dakle ona se samo simulira preko koncentratora. Na taj način, kod današnjeg TP Ethernet-a pojavila se razlika između logičke i fizičke strukture LAN-a: dok je logička struktura i dalje zasnovana na sabirnici, fizička struktura je u obliku zvijezde.

9. WAN tehnologije i usmjeravanje

Sadržaj Poglavlja 9

Za razliku od prethodnih poglavlja gdje smo se bavili LAN-ovima, sad se počinjemo baviti rasprostranjenim mrežama – WAN-ovima. Govorimo o paketnim sklopkama, uređajima koji omogućuju povezivanje udaljenih dijelova WAN-a preko komunikacijskih linija. Objašnjavamo način na koji se računalima u WAN-u pridjeljuju fizičke adrese. Uočavamo problem usmjeravanja paketa u WAN-u, dakle problem da sklopke trebaju u ispravnom smjeru prosljeđivati pakete jedna drugoj. Opisujemo tablice usmjeravanja koje se pohranjuju u sklopkama da bi omogućile izbor „sljedećeg skoka“ za paket kod usmjeravanja. Na kraju spominjemo nekoliko konkretnih tehnologija koje su se koristile za izgradnju WAN-ova.

Potreba za WAN tehnologijama

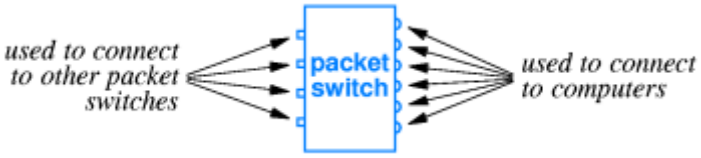
Ukoliko velik broj međusobno udaljenih računala želimo povezati u WAN, potrebne su nam drukčije tehnologije od onih koje smo koristili za LAN. Očekujemo da će brzina međusobnog komuniciranja računala spojenih u WAN biti manja od one u LAN-u. Ipak, od WAN tehnologije tražimo da osigura *skalabilnost*, dakle mogućnost dodavanja novih računala i novih udaljenih lokacija.

Sve WAN tehnologije zasnivaju se na:
- *Vezama* koje omogućuju digitalnu komunikaciju na veliku udaljenost (optička vlakna, sateliti),
- *Paketnim sklopkama* koje omogućuju usmjeravanje paketa od jedne do druge lokacije.

Svojstva paketne sklopke

Paketna sklopka (packet switch) je uređaj koji ima dvije vrste ulazno/izlaznih priključaka (port-ova). Prva vrsta priključaka radi na velikoj brzini i služi za priključivanje veza prema drugim sklopkama. Druga vrsta priključaka radi na manjoj brzini i služi za priključivanje računala.

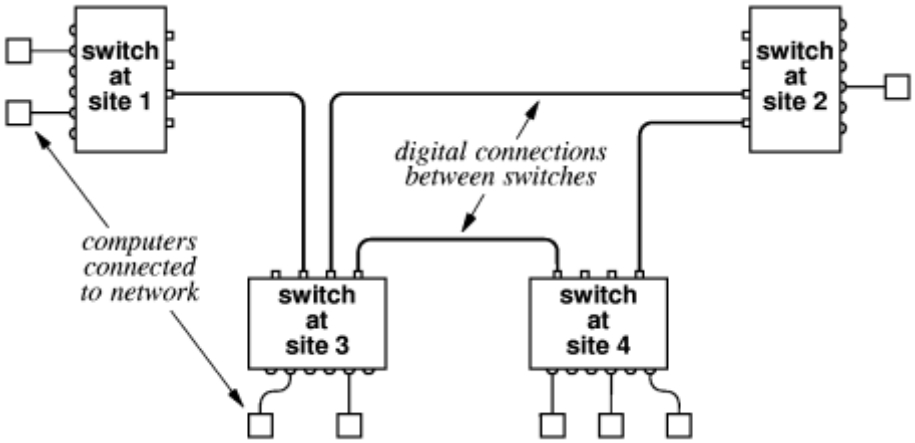
Osnovna zadaća sklopke je prebacivanje cijelih paketa s jednog priključka na drugi. Dakle paket koji je stigao s jednog računala ili jedne telekomunikacijske veze može se usmjeriti prema drugom računalu ili drugoj vezi. Shematski prikaz paketne sklopke vidi se na Slici 9.1.



Slika 9.1: paketna sklopka i njeni priključci.

Oblikovanje WAN-a

Da bi oblikovali WAN, najprije na svaku fizičku lokaciju postavimo bar jednu paketnu sklopku. Zatim svako od računala priključimo na najbližu sklopku. Na kraju uspostavimo veze između sklopki. Slika 9.2 prikazuje jedan od mogućih načina da se izgradi WAN od četiri sklopke i osam računala.



Slika 9.2: mali WAN dobiven povezivanjem paketnih sklopki na četiri lokacije.

WAN ne mora biti simetričan. Veze između pojedinih sklopki te kapaciteti veza biraju prema očekivanom prometu. Veze moraju osigurati povezanost mreže, dakle mora postojati put između svakog para računala. Dobro je da veze osiguraju određenu redundanciju, dakle više različitih putova između istih računala. To jer korisno u slučaju kvara pojedinih veza ili sklopki. Moguće je dodavati i "unutrašnje" sklopke, koje nemaju priključenih računala i služe samo za prijenos i usmjeravanje podataka.

Spremanje i prosljeđivanje

Paketna sklopka zapravo je jedna vrsta specijaliziranog računala. Osim ulazno/izlaznih jedinica, ona ima memoriju i procesor. Svoju zadaću sklopka obavlja tako da pristigle pakete privremeno pohranjuje u memoriju i obrađuje pomoću procesora.

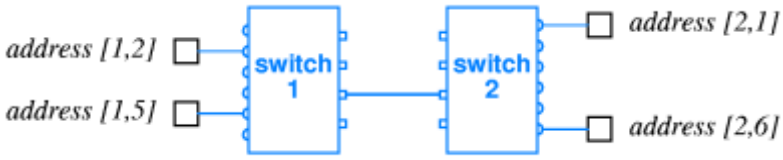
Pohranjeni paketi organiziraju se u red (queue). Novo-pristigli paket stavlja se na začelje reda. Procesor skida paket sa čela reda, gleda njegov sadržaj, te odlučuje kamo će ga dalje proslijediti. Korištenje memorije omogućuje sklopki da izađe na kraj s iznenadnim velikim

prometom paketa. Ipak, veličina memorije je ograničena, tako da može doći do *zagušenja* (congestion) i gubitka podataka.

Fizičko adresiranje u WAN-u

Svaka WAN tehnologija definira format okvira za slanje ili primanje podataka. Svakom računalu spojem u WAN pridružena je fizička adresa. Prilikom slanja okvira, pošiljatelj mora u okvir uključiti adresu primatelja.

Većina WAN-ova koristi dvoslojnu hijerarhijsku shemu adresiranja. Adresa se dijeli na dva dijela: prvi dio identificira paketnu sklopku, a drugi dio određuje računalo spojeno na tu sklopku. Ideja ne ilustrirana Slikom 9.3.

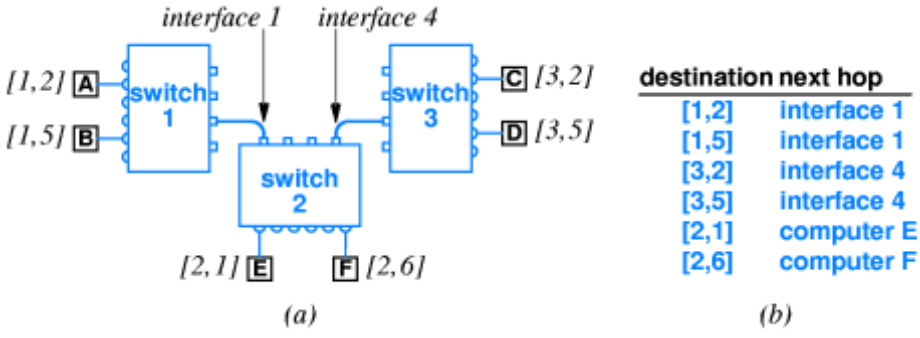


Slika 9.3: primjer hijerarhijskih adresa u WAN-u.

Izbor sljedećeg skoka

Za svaki pristigli paket, sklopka mora odlučiti koji putem će ga dalje proslijediti. Da bi donijela odluku, sklopka gleda adresu primatelja u paketu. Ako je paket namijenjen računalu koje je izravno spojeno na sklopku, tada sklopka prosljeđuje paket tom računalu. Ako je paket namijenjen računalu koje je spojeno na drugu sklopku, tada se paket mora proslijediti po komunikacijskoj vezi koje vodi prema toj drugoj sklopki.

Sklopke ne pohranjuju cjelovitu informaciju o tome kako doseći svako moguće odredište. Umjesto toga, postoji samo informacija o *sljedećem skoku* (next hop) kojeg paket mora napraviti da bi se približio odredištu. Informacije potrebne za izbor sljedećeg skoka mogu se organizirati kao tablica. Na Slici 9.4 vidi se mreža s tri sklopke i tablica unutar sklopke 2.



Slika 9.4: (a) mreža s tri sklopke; (b) tablica sa sljedećim skokovima za sklopku 2.

Hijerarhijske adrese i usmjeravanje

Opisana tablica s informacijama o sljedećem skoku obično se zove *tablica usmjeravanja* (routing table). Prosljeđivanje paketa izborom sljedećeg skoka zove se *usmjeravanje* (routing).

Tablica usmjeravanja može se znatno pojednostaviti ukoliko se koriste dvodijelne hijerarhijske adrese. Naime, sljedeći skok uglavnom je određen prvim dijelom adrese. Pojednostavnjena verzija tablice sa Slike 9.4 tada izgleda kao što je prikazano na Slici 9.5.

Destination	Next Hop
(1, anything)	Interface 1
(3, anything)	Interface 4
(2, anything)	local computer

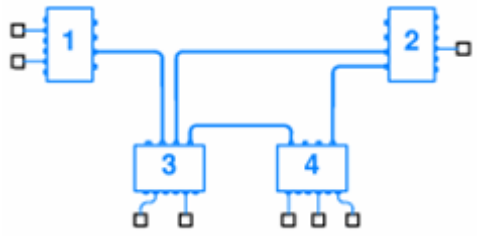
Slika 9.5: skraćena verzija tablice usmjeravanja sa Slike 9.4 (b).

Usmjeravanje u WAN-u

Da bi WAN ispravno radio, sve paketne sklopke moraju u sebi imati pohranjene tablice usmjeravanja, te se moraju baviti prosljeđivanjem paketa. Štoviše, mora se garantirati sljedeće.

- *Univerzalno usmjeravanje.* Svaka tablica određuje sljedeći skok za svako moguće odredište.
- *Optimalni putovi.* U svakoj tablici vrijednost sljedećeg skoka za zadano odredište odgovara početku optimalnog puta prema tom odredištu.

Slika 9.6 prikazuje WAN sa 4 paketne sklopke i ispravne tablice usmjeravanja za svaku sklopku. Brojevi u redcima tablice odnose se na sklopke. Uređeni parovi brojeva su veze između sklopki koje se koriste za sljedeći skok. Tablice na Slici 9.6 zaista osiguravaju univerzalno usmjeravanje. Putovi su optimalni jer koriste najmanji broj skokova.



destin-ation	next hop
1	-
2	(1,3)
3	(1,3)
4	(1,3)

node 1

destin-ation	next hop
1	(2,3)
2	-
3	(2,3)
4	(2,4)

node 2

destin-ation	next hop
1	(3,1)
2	(3,2)
3	-
4	(3,4)

node 3

destin-ation	next hop
1	(4,3)
2	(4,2)
3	(4,3)
4	-

node 4

Slika 9.6: WAN sa 4 sklopke i pripadne tablice usmjeravanja za svaku sklopku.

Korištenje default putova

Unatoč hijerarhijskom adresiranju, tablica usmjeravanja može i dalje sadržavati mnogo redaka s istim sljedećim skokom. Da bi se tablica usmjeravanja još više smanjila, uvodi se *default put*. Kod pretraživanja tablice, najprije se traži redak koji se eksplicitno odnosi na traženo odredište. Ako se takav redak ne nađe, koristi se default. Nakon uvođenja default puta, tablice usmjeravanja sa prethodne Slike 9.6 izgledaju kao na sljedećoj Slici 9.7. U svakoj tablici, redak koji odgovara default putu prepoznajemo po tome što mu je u stupcu „destination" upisana zvjezdica.

Slika 9.7: skraćene verzije tablica sa prethodne slike koje koriste default putove.

Primjeri WAN tehnologija

Većinu WAN tehnologija razvile su telefonske (telekom) kompanije ili organizacije koje se bave standardizacijom telefonskog prometa. Veze između udaljenih paketnih sklopki zapravo su iznajmljene digitalne telefonske linije. Korisnik WAN-a plaća telefonskoj kompaniji najam tih linija. Svaka WAN tehnologija zahtijeva posebnu vrstu paketnih sklopki koje međusobno komuniciraju svojim protokolom i razmjenjuju svoje okvire.

Navodimo nekoliko poznatih primjera WAN tehnologija, od kojih su zadnja dva aktualni i danas.

- *X.25.* Standard kojeg je razvila organizacija ITU. U ranim 80-tim godinama često se koristio za povezivanje ASCII terminala s udaljenim višekorisničkim računalom.
- *Frame Relay.* Prvenstveno namijenjen za povezivanje udaljenih segmenata LAN-a. Radi na brzinama do 100 Mbit/s. Koristi "connection oriented" paradigmu za komuniciranje.
- *Switched Multi-megabit Data Service (SDMS).* Radi na većim brzinama od frame relay-a i zasnovan je na "conectionless" paradigmi komuniciranja.
- *Asynchronous Transfer Mode (ATM).* Tehnologija koja osim za LAN-ove može služiti i za WAN-ove, te osim za prijenos podataka također i za digitalizirani telefonski promet. WAN zasnovan na ATM-u sastoji se od više udaljenih i povezanih ATM sklopki.
- *Multi-Protocol Label Switching (MPLS).* Suvremena tehnologija donekle u srodstvu s ATM, no čvrsto integrirana s internetom. Riječ je o protokolu koji omogućava suvremenim internetskim usmjernicima (routers) da izravno razmjenjuju internetske pakete preko zasebnih telekomunikacijskih veza. Pritom umjernici osim svoje osnovne uloge preuzimaju i ulogu paketnih sklopki. Izravne veze između usmjernika služe kao zamjena za klasični WAN. Ova tehnologija implementirana je u današnjim Cisco usmjernicima.

Sažetak Poglavlja 9

WAN tehnologije omogućuju umrežavanje velikog broja međusobno udaljenih računala. Tipični WAN sastoji se od računala, paketnih sklopki i komunikacijskih linija. Svako računalo priključeno je na najbližu sklopku. Zadaća sklopke je da prosljeđuje pakete od računala do računala, odnosno od računala preko komunikacijske linije do druge sklopke ili obratno. Prosljeđivanje se obavlja u skladu s tablicom usmjeravanja, koja je pohranjena unutar sklopke da bi odredila idući skok (računalo ili drugu sklopku) za paket. Tablice usmjeravanja u svim sklopkama zajedno trebaju osigurati ispravno rješavanje problema usmjeravanja, dakle da svaki paket optimalnim putom stigne od pošiljatelja do primatelja.

10. Algoritmi za usmjeravanje

Sadržaj Poglavlja 10

U prethodnom poglavlju upoznali smo se s paketnim sklopkama, njihovim tablicama usmjeravanja, te s problemom usmjeravanja u WAN-u. Sada obrađujemo algoritme koji na automatski način rješavaju problem usmjeravanja, tako da generiraju ispravne tablice usmjeravanja i upisuju ih u sklopke. Najprije primjećujemo da ti algoritmi polaze od prikaza WAN-a pomoću grafa, te od formulacije problema usmjeravanja kao problema pronalaženja optimalnih putova u grafu. Zatim obrađujemo Dijkstrin algoritam za najkraće putove koji omogućuje statičko usmjeravanje. Na kraju opisujemo algoritam zasnovan na vektorima udaljenosti koji služi za dinamičko usmjeravanje.

Općenito o problemu usmjeravanja

Vidjeli smo da svaka paketna sklopka u WAN-u mora imati upisanu tablicu usmjeravanja. *Problem usmjeravanja* sastoji se od računanja i upisivanja tablice usmjeravanja u svakoj od sklopki, tako da se osigura univerzalno usmjeravanje i optimalnost putova.

Problem se može riješiti na dva načina:
- *Ručno.* Administrator mreže upisuje retke u svaku od tablica. To je izvedivo samo za vrlo male mreže.
- *Automatski.* Tablice se računaju korištenjem softvera koji je instaliran u svakoj sklopki. Taj softver radi u skladu s određenim *algoritmom za usmjeravanje.*

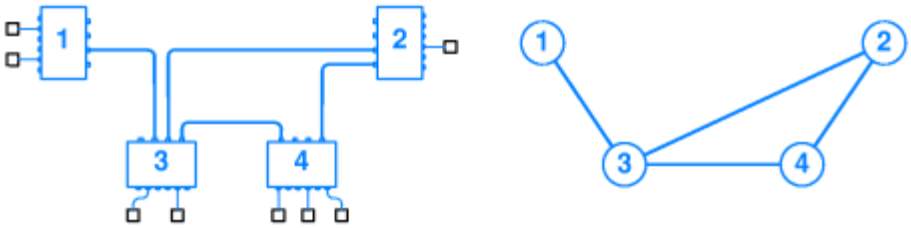
Automatsko usmjeravanje dalje se dijeli na:
- *Statičko usmjeravanje.* Softver generira tablicu u trenutku pokretanja sklopke. Tablica se dalje ne mijenja.
- *Dinamičko usmjeravanje.* Softver u trenutku pokretanja sklopke stvara polaznu verziju tablice. Softver kasnije mijenja tablicu ako se uvjeti u mreži promijene.

Statičko usmjeravanje je jednostavnije i manje opterećuje mrežu. Dinamičko usmjeravanje je fleksibilnije pa se danas češće koristi.

Prikaz WAN-a pomoću grafa

Svi algoritmi za usmjeravanje zasnivaju se na prikazu strukture WAN-a pomoću *grafa*. U takvom prikazu *čvorovi* grafa predstavljaju paketne sklopke, a *bridovi* predstavljaju izravne

veze (komunikacijske linije) između sklopki. Graf ističe bitna svojstva povezanosti WAN-a, a zanemaruje nebitne detalje kao što su fizički smještaj sklopki i priključena računala. Na slici 10.1 vidi se primjer WAN-a sa četiri sklopke i četiri komunikacijske linije, te odgovarajući graf sa četiri čvora i četiri brida.



Slika 10.1: primjer WAN-a i odgovarajućeg grafa.

Da bi problem usmjeravanja u WAN-u mogli formulirati kao problem iz teorije grafova, svakom bridu grafa moramo još pridružiti njegovu *duljinu* (težinu, cijenu). Duljina puta u grafu računa se kao zbroj duljina pripadnih bridova.

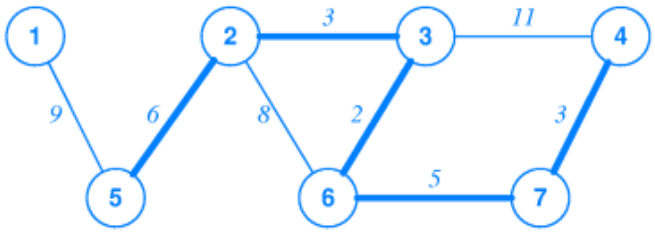
Duljine mogu imati razne interpretacije.

- Ako svim bridovima zadamo duljinu 1, tada je duljina puta jednaka broju skokova od polazišta do odredišta.
- Ako duljina brida odgovara brzini prijenosa duž odgovarajuće komunikacijske linije, tada duljina puta izražava ukupno vrijeme potrebno za prijenos podatka od polazišta do odredišta.
- Duljina brida može biti i cijena koju plaćamo za najam komunikacijske linije po prenesenom Mbit-u. Duljina puta tada odgovara ukupnoj cijeni prijenosa jednog Mbit od polazišta do odredišta.

Grafovska formulacija problema

Problem usmjeravanja u WAN-u zapravo se svodi na pronalaženje najkraćih putova između svih parova čvorova u pripadnom grafu. Na grafu sa Slike 10.2, najkraći put između čvorova 4 i 5 označen je podebljanim bridovima i njegova duljina je 19. Slično bi mogli odrediti najkraći put za sve parove čvorova.

Najkraći put je optimalan po onom kriteriju kojeg smo odabrali kad smo zadavali duljine bridova. Na primjer, ako su duljine bridova jedinične cijene najma komunikacijskih linija, tada najkraći put zapravo određuje najjeftiniji način prijenosa podataka. Ako su sve duljine bridova postavljene na 1, tada je najkraći put ustvari put s najmanjim brojem skokova.

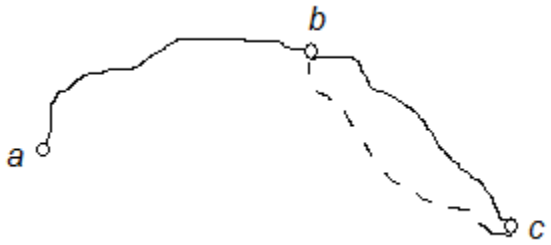


Slika 10.2: primjer grafa čijim bridovima su zadane duljine.

Bilo koji algoritam za usmjeravanje zapravo je jedna vrsta algoritma za traženje najkraćeg puta u grafu. Takav algoritam u svakoj sklopki rješava takozvanu *single-source* varijantu problema najkraćih putova. Dakle, u svakoj sklopki traži se najkraći put od dotičnog čvora kao polazišta do bilo kojeg drugog čvora kao odredišta. Zatim se za svaki takav najkraći put u tablicu usmjeravanja upisuje sljedeći skok koji odgovara početnom bridu puta – to je dovoljno da se kasnije reproducira cijeli najkraći put.

Bellmanov princip optimalnosti

Da bi reproducirali bilo koji najkraći put, rekli smo da je dovoljno je pamtiti samo njegov početni brid. Razlog zašto je to moguće naziva se *Bellmanov princip optimalnosti*. Princip kaže da ako najkraći put od čvora *a* do čvora *c* prolazi čvorom *b*, tada dio tog puta između *b* i *c* ujedno predstavlja najkraći put između *b* i *c* – vidi Sliku 10.3.



Slika 10.3: Bellmanov princip optimalnosti

Reproduciranje najkraćeg puta od neke polazne sklopke do neke odredišne sklopke obavlja se tako da najprije iz polazišta napravimo sljedeći skok prema odredištu, zatim iz sklopke u kojoj se nađemo opet sljedeći skok prema odredištu, … i tako dalje dok ne stignemo u odredište. Bellmanov princip garantira da će dobiveni put sastavljen od niza sljedećih skokova zaista biti najkraći put.

Statičko usmjeravanje pomoću Dijkstrinog algoritma

Prvi algoritam za usmjeravanje koji ćemo razmatrati je verzija Dijkstrinog algoritma za najkraće putove u grafu. Riječ je o poznatom i relativno brzom postupku optimizacije koji se i inače koristi u teoriji grafova i operacijskim istraživanjima.

Da bi mogli primijeniti Dijkstrin algoritam, graf koji prikazuje građu WAN-a mora biti poznat unaprijed, te se ne može mijenjati. Također, duljine bridova moraju biti nenegativne jer u protivnom algoritam neće korektno raditi. Algoritam radi tako da svaka paketna sklopka pohranjuje u svojoj memoriji opis cijelog grafa, te rješava neovisno o drugim sklopkama svoj single-source problem najkraćih putova. To znači da će se u svrhu usmjeravanja isti Dijkstrin postupak istovremeno pokrenuti mnogo puta, onoliko puta koliko ima sklopki.

Pseudo-kod Dijkstrinog algoritma

Detalji postupka kojeg izvodi pojedina sklopka vidljivi su iz sljedećeg pseudo-koda. U tom pseudokodu *S* označava skup odredišta za koje još nije određena najkraća udaljenost. Dokaz korektnosti nećemo izvoditi, no on se može naći u udžbenicima o diskretnoj matematici i teoriji grafova.

Ulaz:

Graf s nenegativnim duljinama bridova i istaknutim čvorom – polazištem.

Izlaz:

Polje *D* s najkraćim udaljenostima,
D[*v*] je najkraća udaljenost od polazišta do čvora *v*.
Tablica *R* sa sljedećim skokovima.
R[*v*] je sljedeći skok od polazišta prema *v*.

Postupak:

```
inicijaliziraj skup S tako da sadrži sve čvorove osim polazišta;
inicijaliziraj D tako da je D[v] duljina brida od polazišta do v
        ako takav brid postoji, odnosno ∞ inače;
inicijaliziraj R tako da je R[v] jednako v ako postoji brid od polazišta do v,
        odnosno prazno inače;
dok ( S nije prazan) {
        izaberi u iz S takav da je D[u] minimalno;
        ako (D[u] je jednak ∞) {
                greška: graf nije povezan; prekini rad;
        }
        izbaci u iz S;
        za svaki v takav da postoji brid (u,v) {
                ako (v je još uvijek u S) {
                        c = D[u] + duljina brida (u,v);
                        ako (c < D[v]) {
                                R[v] = R[u];
                                D[v] = c;
                        }
                }
        }
}
```

Dinamičko usmjeravanje pomoću vektora udaljenosti

Drugi algoritam za usmjeravanje koji ćemo razmatrati zasnovan je na takozvanom vektoru udaljenosti. Njegova velika prednost u odnosu na Dijkstrin algoritam je u tome što graf koji prikazuje građu WAN-a nigdje ne mora biti eksplicitno pohranjen. Umjesto toga, pojedina paketna sklopka treba znati samo koji su njezini susjedi, te kolike su duljine bridova između nje i njezinih susjeda.

Kod primjene algoritma zasnovanog na vektoru udaljenosti, svaki redak tablice usmjeravanja unutar sklopke sadrži tri polja: odredište, sljedeći skok, udaljenost odredišta duž puta koji odgovara sljedećem skoku (otuda naziv *vektor udaljenosti*). Sklopke povremeno šalju podatke iz svoje tablice po mreži susjednim sklopkama. Riječ je o *porukama usmjeravanja* (routing messages) koje sadrže parove odredišta i udaljenosti. Kad god pojedina sklopka primi od svog susjeda poruku navedenog tipa, ona analizira tu poruku i mijenja svoju vlastitu tablicu ukoliko susjed ima kraći put do nekog odredišta.

Nakon dovoljnog broja iteracija, informacije o građi grafa implicitno će se proširiti cijelom mrežom. Svaka sklopka sagradit će ispravnu tablicu. Ako se naknadno pojave nove veze u

mreži, tablice će se i dalje popravljati, dakle sklopke će uvažiti pojavu novih boljih (kraćih) putova. Ako se neke veze u mreži naknadno prekinu, potrebno je resetirati tablice i početi cijeli postupak iznova. Nakon dovoljnog broja iteracija sklopke će pronaći alternativne putove koji zaobilaze prekinute veze.

Pseudo-kod algoritma zasnovanog na vektorima udaljenosti

Detalji algoritma zasnovanog na vektoru udaljenosti vidljivi su iz sljedećeg pseudo-koda. Taj pseudo-kod odnosi se na rad jedne sklopke (čvora). Korektnost cijelog postupka je prilično očigledna iz Bellmanovog principa te iz činjenice da algoritam nikad ne prestaje raditi tako da se informacije o svakom najkraćem putu prije ili kasnije moraju proširiti do svih sklopki.

```
Ulaz:
        Identifikator lokalnog čvora.
        Lokalna tablica usmjeravanja.
        Duljine svih bridova prema susjednim čvorovima.
        Niz ulaznih poruka za usmjeravanje.
Izlaz:
        Niz ažuriranih verzija lokalne tablice usmjeravanja.

Postupak:
        inicijaliziraj tablicu usmjeravanja tako da sadrži samo jedan redak:
                odredište jednako lokalnom čvoru, sljedeći skok prazan, udaljenost 0;
        ponavljaj zauvijek {
                čekaj da stigne iduća poruka za usmjeravanje;
                        neka je pošiljatelj te poruke susjedni čvor N;
                za svaki redak iz poruke {
                        neka je V odredište u tom retku, a D udaljenost;
                        C = D + duljina brida kojim je poruka stigla;
                        pregledaj i ažuriraj lokalnu tablicu usmjeravanja:
                        ako (u tablici ne postoji put do V) {
                                dodaj redak s odredištem V, sljedećim
                                skokom N i udaljenošću C;
                        } inače ako (postoji put i sljedeći skok je N) {
                                promijeni udaljenost za taj put u C;
                        } inače ako (postoji put s udaljenošću > C) {
                                promijeni sljedeći skok u N, udaljenost u C;
                        }
                }
        }
```

Sažetak Poglavlja 10

WAN se može matematički opisati kao graf čijim bridovima su pridružene duljine. Svi algoritmi za usmjeravanje u WAN-u zapravo su algoritmi za traženje najkraćih putova u grafu. Dijkstrin algoritam pronalazi najkraće putove na najefikasniji način, no on je primjenjiv samo u statičkom slučaju, dakle onda kad je cjelokupna struktura WAN-a unaprijed poznata svim sklopkama i ne može se mijenjati. Algoritam zasnovan na vektorima udaljenosti omogućuje dinamičko usmjeravanje u situaciji kad se WAN mijenja i kad svaka sklopka poznaje samo lokalni dio njegove strukture.

11. Mjerenje performansi mreže

Sadržaj Poglavlja 11

U ovom poglavlju govorimo o performansama mreže. Opisujemo četiri mjere za performanse, a to su: kašnjenje, propusnost, umnožak kašnjenja i propusnosti, te varijacija kašnjenja. Raspravljamo o značenju tih mjera, te o njihovim međusobnim odnosima.

Općenito o performansama

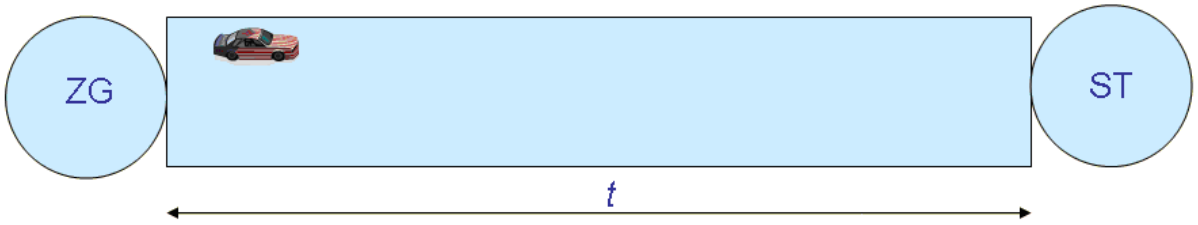
Ljudi često neformalno govore o brzim i sporim mrežama. No mi ćemo izbjegavati pojam „brzine“ jer je on suviše subjektivan i nema jasnu definiciju. Umjesto toga, kod opisa performansi mreže služit ćemo se precizno definiranim i mjerljivim veličinama.

Dvije osnovne veličine koje ćemo proučavati su:
- *Kašnjenje* (delay, latency) – mjeri se u vremenskim jedinicama.
- *Propusnost* (throughput) – mjeri se u bitovima po vremenskoj jedinici.

Osim toga promatrat ćemo i dvije dodatne veličine, koje su izvedene iz osnovnih no također daju važnu informaciju o performansama. To su:
- *Umnožak kašnjenja i propusnosti* – mjeri se u bitovima.
- *Varijacija kašnjenja* (jitter) – mjeri se u vremenskim jedinicama.

Kašnjenje

Kašnjenje se definira kao vrijeme koje je potrebno jednom bitu da prijeđe put kroz mrežu od jednog do drugog računala. Mjeri se u vremenskim jedinicama, obično u mili-sekundama.
Ovisi o izabranom paru računala, te varira čak i za isti par računala. Zato se obično izražava kao prosječno odnosno maksimalno kašnjenje. Ako se poslužimo analogijom s automobilima i cestom, tada kašnjenje odgovara vremenu koje jedan automobil provede na putu od jednog grada do drugog – vidi Sliku 11.1.



Slika 11.1: ilustracija kašnjenja u mreži pomoću analogije s automobilima i cestom.

Detaljnija analiza pokazuje da se kašnjenje sastoji od više dijelova:
- *Kašnjenje zbog prolaska* (propagation delay) – vrijeme potrebno signalu da prođe kroz medij.
- *Kašnjenje zbog prospajanja* (switching delay) – vrijeme potrebno da paketna sklopka prihvati cijeli paket te izabere sljedeći skok.

- *Kašnjenje zbog čekanja na pristup* (access delay) – vrijeme koje računalo u LAN-u mora čekati da bi dobilo pristup do zajedničkog medija.
- *Kašnjenje zbog čekanja u redu* (queuing delay) – vrijeme koje paket provede čekajući u memoriji paketne sklopke.

Tipične vrijednosti za kašnjenje zbog prolaska su od 1 ms (u slučaju LAN-a) do nekoliko stotina ms (u slučaju satelitskih veza). Kašnjenje zbog prospajanja obično ima jako male vrijednosti, tako da je to najmanje značajan dio ukupnog kašnjenja. Vrijednosti kašnjenja zbog čekanja na pristup ili zbog čekanja u redu ovise o opterećenosti mreže, te obično predstavljaju najznačajniji dio ukupnog kašnjenja.

Propusnost

Propusnost se definira kao količina podataka koja se u jedinici vremena može slati kroz mrežu od jednog računala prema drugom. Mjeri se u bitovima po vremenskoj jedinici, obično u Mbit/s ili Gbit/s. Po analogiji s automobilima i cestom, propusnost odgovara broju automobila koji mogu ući na cestu u jedinici vremena – vidi Sliku 11.2.

Slika 11.2: ilustracija propusnosti mreže pomoću analogije s automobilima i cestom.

Ljudi često baš propusnost nazivaju “brzina”, no to je pogrešno jer je propusnost ustvari mjera za *kapacitet*, a ne za brzinu mreže. Tipične vrijednosti za propusnost u današnjim mrežama kreću se od 50-tak Kbit/s kod dial-up veza, preko nekoliko Mbit/s u WAN-ovima ili kod ADSL veza, sve do 10 Gbit/s u LAN-ovima.

Pojam propusnosti komunikacijske linije prilično je srodan pojmu širine pojasa (bandwidth) te linije. No, tu ipak postoji razlika. Naime:
- Propusnost mjeri stvarnu količinu podataka koji se mogu slati u jedinici vremena.
- Širina pojasa daje teorijsku gornju ogradu za propusnost koju postavlja sam fizički medij.

Odnos između kašnjenja i propusnosti

U teoriji, kašnjenje i propusnost su dvije nezavisne veličine. U praksi, te veličine ipak djeluju jedna na drugu. Razlog za međusobnu ovisnost lako je razumjeti pomoću analogije s automobilima i cestom. Ako je cesta zakrčena prometom, tada svi moraju sporije voziti pa se vrijeme putovanja produljuje.

Ukoliko u WAN ulazi velika količina podataka, tada paketne sklopke nisu u stanju odmah obraditi velik broj paketa, pa se povećava kašnjenje zbog čekanja u redu. Slično, ukoliko kroz LAN krene velika količina podataka, tada se povećava kašnjenje zbog čekanja na pristup zajedničkom mediju. Pojava povećanog kašnjenja zbog velikog prometa u mreži zove se

zagušenje (congestion). U slučaju zagušenja, odgovarajući protokol trebao bi smanjiti intenzitet ubacivanja novih podataka u mrežu.

Iskustvo je pokazalo da vrijedi sljedeća približna formula koja povezuje kašnjenje i propusnost.

- Neka je D_0 kašnjenje u situaciji kad u mreži nema prometa.
- Neka je U vrijednost između 0 i 1 koja kaže koliki dio ukupne propusnosti se trenutno koristi.
- Tada se stvarno kašnjenje D dobiva kao:
  $D = D_0 / (1 - U)$.

Znači, ako je mreža neopterećena, stvarno kašnjenje je D_0. Ako mreža radi na 50% svoje propusnosti, stvarno kašnjenje se udvostručuje. Kad se promet približi kapacitetu mreže, kašnjenje teži prema beskonačnosti.

Umnožak kašnjenja i propusnosti

Umnožak kašnjenja i propusnosti mjeri se u bitovima. Ta izvedena veličina ima zanimljivu interpretaciju: ona daje količinu podataka koja odjednom može biti prisutna u mreži. Po analogiji s automobilima i cestama, umnožak kašnjenja i propusnosti odgovara maksimalnom broju automobila koji se u jednom trenutku mogu zateći na cesti – vidi Sliku 11.3.

Slika 11.3: ilustracija umnoška kašnjenja i propusnosti po analogiji s automobilima i cestom.

U slučaju zasebne komunikacijske linije između dva računala, umnožak kašnjenja i propusnosti daje količinu podataka koju pošiljatelj treba proizvesti i poslati prije nego što primatelj dobije prvi bit. U slučaju mreže s paketnim sklopkama, umnožak postavlja zahtjev na ukupni kapacitet memorija unutar sklopki. Ako sklopke nemaju toliko memorije, tada lako može doći do gubitka podataka jer se oni neće imati gdje pohraniti. Situacija kad sklopka gubi pakete zato jer ih nema gdje pohraniti zove se *kolaps uslijed zagušenja* (congestion collapse).

Varijacija kašnjenja

Varijacija kašnjenja je broj koji kaže koliko kašnjenje može biti veće ili manje od svoje prosječne vrijednosti. Mjeri se u vremenskim jedinicama, na primjer u mili-sekundama. Ova veličina je važan pokazatelj ukoliko pokrećemo multimedijske aplikacije, na primjer reprodukciju video zapisa preko mreže. Za takve aplikacije zapravo nam je potrebna mreža sa što manjom varijacijom kašnjenja (zero-jitter network).

Kašnjenje u mreži svakako će uzrokovati da se video zapis kod primatelja reproducira s vremenskim pomakom u odnosu na pošiljatelja. No varijacija u kašnjenju poremetit će i takvu vremenski pomaknutu reprodukciju. Naime:

- Iznenadno smanjenje kašnjenja uzrokovat će da se dio video zapisa reproducira neprirodno brzo.
- Iznenadno povećanje kašnjenja vidjet će se kao usporenje ili zastajkivanje video reprodukcije.

Kod mreže s velikom varijacijom kašnjenja, reprodukcija video zapisa može se ostvariti jedino tako da se dijelovi video zapisa spremaju u privremeni spremnik (buffer) na strani primatelja, te da se reprodukcija odvija iz tog spremnika s vremenskim pomakom koji je još veći od kašnjenja.

Sažetak Poglavlja 11

Osnovne mjere za performanse mreže su: kašnjenje i propusnost. Kašnjenje se mjeri u mili-sekundama, a propusnost u mega-bitovima ili giga-bitovima po sekundi. Poželjno je da kašnjenje bude što manje a propusnost što veća. Daljnje mjere za performanse koje se izvode iz osnovnih su: umnožak kašnjenja i propusnosti, te varijacija kašnjenja. Makar su kašnjenje i propusnost u teorijskom smislu nezavisne veličine, u praksi one ipak utječu jedna na drugu, na primjer smanjenje propusnosti uzrokovat će povećano kašnjenje.

III. POVEZIVANJE RAZNORODNIH MREŽA

12. Temeljne postavke i arhitekture interneta

Sadržaj Poglavlja 12

U ovom poglavlju raspravljamo o tehnologijama povezivanja raznorodnih mreža. Budući da niti jedna mrežna tehnologija nije optimalna za sve potrebe, kombiniranjem tehnologija želimo dobiti prilagodljivije (robusno) rješenje. Glavni problem koji se javlja je problem pružanja jedinstvene usluge pri komunikaciji među čvorovima u mrežama koje koriste različite tehnologije povezivanja. Spajanje raznorodnih *fizičkih mreža* u jedinstvenu *logičku mrežu* je koncept koji se naziva *internetworking*.

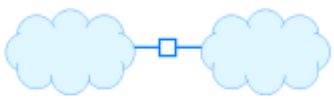
Pojam jedinstvene usluge

Budući da nije moguće spojiti dvije tehnološki nekompatibilne mreže jednostavnim spajanjem žica, postavlja se pitanje: je li moguće pružiti jedinstvenu uslugu bez uvođenja jedinstvenog tehnološkog standarda za sve fizičke mreže?

Hardversko rješenje zvano *premošćivanje* (bridgeing) ne rješava problem povezivanja raznorodnih mreža u potpunosti. Naime, različite mreže mogu koristiti različite tipove adresiranja i formate paketa i okvira. Problem se rješava kombiniranjem softverskog i hardverskog rješenja. Paradigma, koja se zove *internetworking,* koristi dodatne uređaje koji se zovu *usmjernici* (router-i). Usmjernik je istovremeno čvor u više mreža; on može prebacivati podatke iz jedne mreže u drugu, te konvertirati podatke iz jednog formata u drugi. Usmjernik se sastoji od procesora, memorije i posebnog I/O sučelja za svaku od mreža u kojima je čvor. Osnovna razlika između usmjernika i paketne sklopke je to što prilikom usmjeravanja mreže tretiraju usmjernik kao bilo koji drugi čvor.

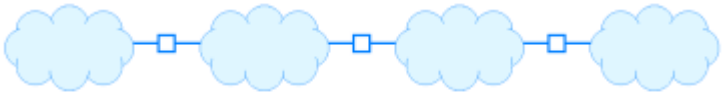
Spajanje fizičkih mreža

Spajanje fizičkih mreža postiže se tako da jedan usmjernik uključimo u više fizičkih mreža. Na Slici 12.1 vidimo dvije mreže spojene preko jednog usmjernika. Podaci iz jedne mreže mogu se preko usmjernika prebaciti u drugu mrežu, i obratno.



Slika 12.1: dvije mreže spojene usmjernikom.

Budući da usmjernik mora usmjeriti svaki paket, njegov procesor nije dovoljno snažan da bi održavao promet između proizvoljnog broja mreža. Zato za spajanje većeg broja mreža obično trebamo i više usmjernika. Na Slici 12.2 vide se četiri mreže povezane preko tri usmjernika.



Slika 12.2: internet koje je nastao spajanjem četiriju fizičkih mreža pomoću tri usmjernika.

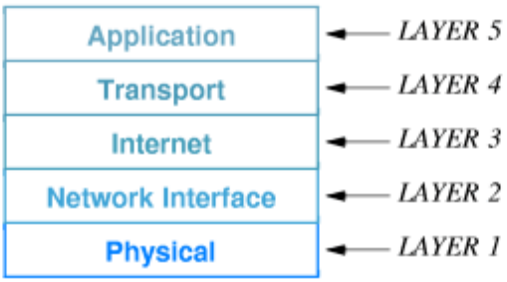
Uvođenje većeg broja usmjernika često stvara i redundantnost putova među čvorovima, Takva redundantnost povećava pouzdanost i omogućava upravljanje konstantnim protokom informacija. Komunikacijski protokol nadgleda i usmjerava promet pri preopterećenju usmjernika.

Protokoli za internetworking - TCP/IP

Zadatak koji usmjernik obavlja je složen budući da svaki okvir iz jedne mreže mora biti preusmjeren u drugu mrežu i upućen do krajnjeg čvora. Zato i protokol koji upravlja radom usmjernika mora biti složen. Prvi model za složeni mrežni protokol je bio 7-slojni *OSI model* (Open Systems Interconnection Basic Reference Model). OSI model nikada nije bio implementiran do kraja. Njegova mana je da on ne sadrži sloj za internet. Ipak, u razgovoru se mrežni profesionalci često referenciraju na OSI model kao primjer apstraktnog modela višeslojnog protokola. Zato je OSI model izvrsno ishodište za početak učenja o arhitekturi mrežnih protokola. Više informacija može se naći na http://en.wikipedia.org/wiki/OSI_model.

Umjesto daljnjeg razmatranja OSI modela opisat ćemo 5-slojni *TCP/IP stog protokola* za kojeg možemo reći da je komunikacijski model koji je omogućio globalni Internet. Ovaj stog protokola, shematski prikazan na slici 12.3, omogućuje komunikaciju među programima koje se izvršavaju na računalima u fizički različitim mrežama. Osnovni princip dizajna se može svesti na sljedeća pravila:

- *pošiljatelj*: dijeli podatke u segmente (pakete), i šalje ih mrežnom sloju;
- *primatelj*: slaže segmente i prosljeđuje ih aplikacijskom sloju.



Slika 12.3: pet slojeva TCP/IP modela

Konceptualno, paket bi na čvoru *pošiljatelj* trebao proći cijeli stog protokola odozgo prema dolje, dok bi na čvoru *primatelj* trebalo proći taj isti stog u suprotnom putu. Detaljni popis protokola koji sudjeluju u TCP/IP stogu vidi se na Slici 12.4 koja je preuzeta i prevedena iz wikipedije s adrese http://en.wikipedia.org/wiki/TCP/IP .

5. Aplikacija	DNS, TFTP, TLS/SSL, FTP, Gopher, HTTP, IMAP, IRC, NNTP, POP3, SIP, SMTP, SNMP, SSH, TELNET, ECHO, BitTorrent, RTP, PNRP, rlogin, ENRP
	Neki protokoli za usmjeravanje mogu biti dio aplikacijskog ali i Internet sloja (BGP border gateway protokol).
4. Transport	TCP, UDP, DCCP, SCTP, IL, RUDP
3. Internet	Protokoli za usmjeravanje koji se odvijaju u IP sloju se smatraju dijelom Internet sloja (OSPF).
	IP (IPv4, IPv6)
	ARP and RARP su protokoli koji se odvijaju ispod IP sloja ali ipak iznad Mrežnog sučelja.
2. Mrežno sučelje	Ethernet, Wi-Fi, token ring, PPP, SLIP, FDDI, ATM, Frame Relay, SMDS

Slika 12.4: protokoli u TCP/IP stogu, razvrstani po slojevima.

Slojevito projektiranje protokola

Osnovni komunikacijski hardver ima mehanizme koji znaju prenijeti bitove podataka od jednog čvora do drugog. *Protokol* je apstrakcija koja definira skup pravila po kojoj čvorovi u mreži mogu izmjenjivati poruke bez da direktno stupaju u interakciju s hardverom. Komunikacijski problem se ne rješava monolitnim modelom već se organizira kao *stog slojeva* (layering model). To olakšava analizu, projektiranje i izvođenje softvera. Također, povećava prilagodljivost i robusnost rješenja. Svaki sloj protokola rješava dio komunikacijskog problema. U tu svrhu svaki sloj na početnom čvoru dodaje informacije zaglavlju izlaznog okvira. Na krajnjem čvoru se ti podaci uklanjaju.

Na primjer, *mrežno sučelje* na čvoru 1 dodaje jedinični komplement okviru kojeg obrađuje, dok ga *mrežno sučelje* na čvoru 2 koristi za kontrolu pristiglog okvira i nakon toga ga uklanja.

Neki slojevi protokola rade više nego li je puka detekcija greške. Kao primjer, spomenimo *algoritam retransmisije* (positive acknowledgement and retransmission), prikazan na Slici 19.2. Unutar tog algoritma, paketi pri slanju dobivaju slijedni broj. Čvor primatelj šalje potvrdu da je primio paket, te slaže pakete po slijednim brojevima. Izgubljeni paketi ponovno se šalju.

Usmjernici, čvorovi i slojevi protokola

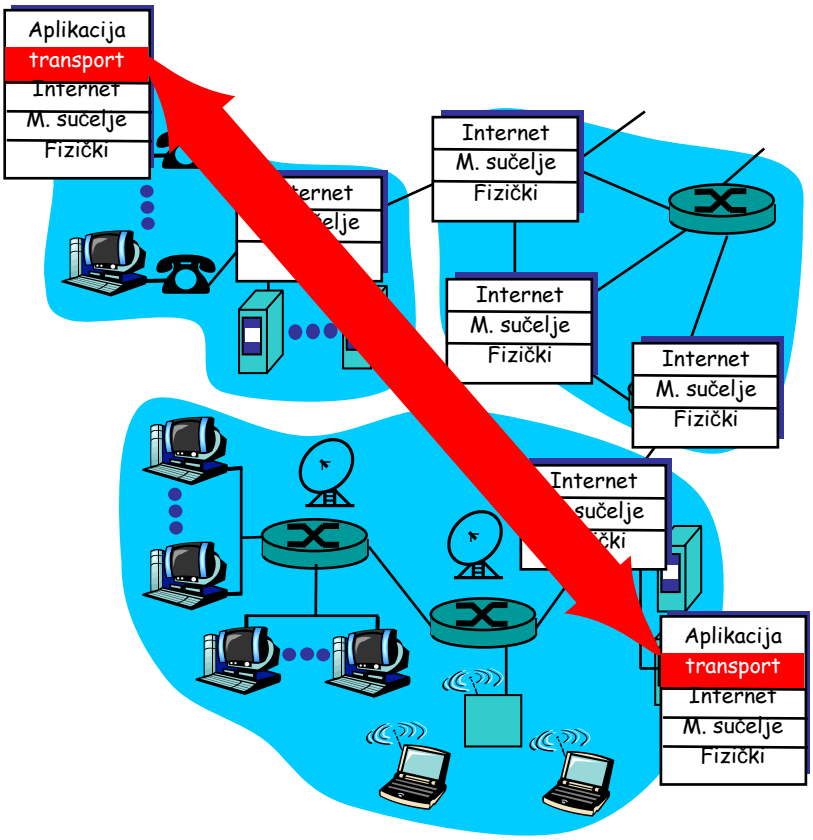
Razlikujemo *čvorove domaćine* (host-computers – računala koja izvode neku aplikaciju), te *čvorove usmjernike* (routers) koji povezuju raznorodne mreže. Usmjernik je računalo sa dva mrežna sučelja koje prebacuje i prevodi podatke između dvije raznolike mreže. Svi čvorovi koriste slojeve TCP/IP protokola, no pojedini čvor ne mora koristiti sve slojeve protokola. Na primjer, usmjernik ne treba protokole sloja 5 (aplikacijski sloj).

Stvara se dojam *virtualne mreže* budući da gornji slojevi protokola skrivaju fizičke detalje mreža. Posebno se skrivaju detalji:

- fizičkih veza i fizičkih adresa,
- fizičkih formata okvira i prikaza podataka.

Umjesto toga u virtualnoj mreži imamo:

- virtualne IP adrese,
- virtualne pakete ili datagrame.



Slika 12.5: logički spoj čvor–čvor.

Virtualna mreža

Pojam apstraktnog paketa datagrama, te apstraktne IP adrese omogućava izmjenu poruka između čvorova u raznorodnim mrežama. Time smo dobili apstraktnu ili virtualnu mrežu u kojoj su krajnji čvorovi aplikacijski programi, kao što je ilustrirano na Slici 12.5. To indicira da virtualna mreža skriva i interakciju komunikacijskog protokola s operacijskim sustavom. Preciznije, *internet socket* je krajnja točka komunikacije među procesima koji se izvršavaju na

računalima koja komuniciraju putem mreže bazirane na internet protokolu. Glavni parametri internet socketa su:

- protokol,
- lokalna IP adresa,
- lokalni port,
- IP adresa udaljenog računala,
- port udaljenog računala.

Najčešće vrste socketa su:

- Datagram socket koji koristi UDP protokol (više u poglavlju 18).
- Stream socket koji koristi TCP protokol (više u poglavlju 19).
- Raw socket (raw IP socket) koji se tipično koristi na usmjernicima i u protokolima za dojavu greške ICMP (vidi poglavlje 14).

Takav model omogućuje skalabilno i robusno rješenje problema pružanja jedinstvene usluge u stalno rastućoj mreži.

Sažetak Poglavlja 12

U ovom poglavlju smo prikazali arhitekturu raznorodne mreže koja koristi IP protokol za rješavanje osnovnog komunikacijskog zadatka. Uveli smo pojmove virtualnog paketa, virtualne adrese i virtualne mreže. Dali smo i reference na buduća poglavlja u kojima se detaljnije razrađuju pojedini koncepti.

13. Adrese za Internet protokol – IP

Sadržaj Poglavlja 13

U ovom poglavlju diskutiramo format virtualnih ili IP adresa. Opisujemo ograničenja tog formata, te prikazujemo neka moguća rješenja za ublažavanje ograničenja. U ovom poglavlju opisujemo IP protokol verzija četiri. Verzija šest koja se sada uvodi bit će opisana u zadnjem poglavlju ovih skripata.

Adrese za virtualnu mrežu

Virtualnu mrežu možemo realizirati isključivo ukoliko svi čvorovi koriste jedinstven sistem adresiranja. Sistem adresiranja u virtualnoj mreži mora biti neovisan o fizičkim adresama. Rezultat je da dva aplikacijska programa ili dva korisnika izmjenjuju poruke bez znanja fizičkih adresa. Informaciju o fizičkim adresama trebaju samo niži slojevi protokola. IP adresiranje stvara dojam velike homogene mreže s jedinstvenom uslugom. Adresiranje u stogu protokola TCP/IP je određeno i*nternet protokolom* - IP. Svaki čvor u mreži ima 32-bitni broj koji se naziva *internet protocol address* ili skraćeno *IP adresa*. Svaki paket koji se šalje kroz virtualnu mrežu u zaglavlju ima IP adresu polaznog i dolaznog čvora. Sva komunikacija se odvija jedino korištenjem IP adresa.

Format virtualne IP adrese

Internet adresa je 32-bitni broj koji se dijeli prefiks i sufiks. *Prefiks* ili *adresa mreže* identificira fizičku mrežu u kojoj se čvor nalazi. *Sufiks* ili *adresa čvora* označava pojedinačni

čvor u mreži. Nikoje dvije fizičke mreže ne mogu imati istu mrežnu adresu i nikoja dva čvora u fizičkoj mreži ne mogu imati isti sufiks. Svaki čvor ima jedinstvenu IP adresu koja je uređeni par (prefiks, sufiks). Ovakva hijerarhijska struktura Internet adresa olakšava usmjeravanje kao i administriranje Interneta. Naime, administriranje mrežnih adresa je globalno, a sufiksa lokalno.

Pojavljuje se sljedeće konceptualno pitanje: kako podijeliti 32 bita na prefiks i sufiks? Pitanje je otežano sljedećim suprotstavljenim činjenicama:

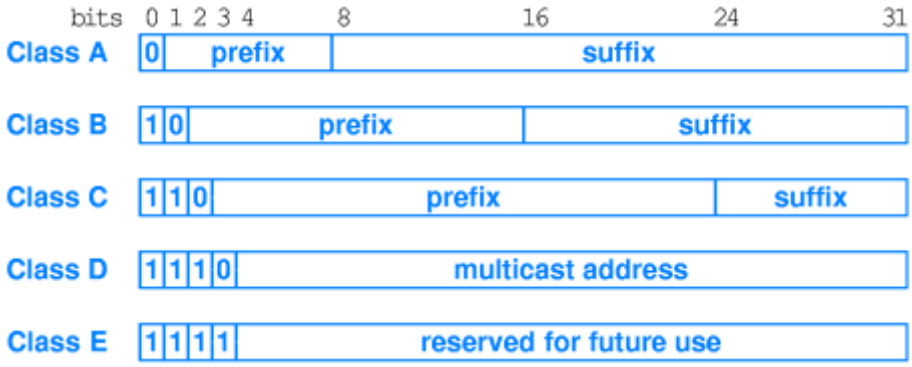
- preveliki prefiks ograničava veličinu fizičkih mreža,
- preveliki sufiks ograničava njihov broj.

Znači li to da treba izgraditi poseban Internet za velike, a poseban za male korisnike?

Kompromis se postiže tako da se uvode klase IP adresa. Prva 4 bita adrese određuju klasu IP adrese. Postoji 5 klasa, tri primarne i dvije sekundarne. Klase se označavaju slovima A,B,C,D,E. Pritom vrijedi sljedeće.

- Klase A, B, C su primarne klase i koriste se za adresiranje čvorova.
- Klasa D se koristi za difuziju u grupi (multicasting). Skup čvorova dijeli multicast adresu, svaki dobiva kopiju svakog paketa poslanog na multicast adresu.
- Klasa E je rezervirana za buduću upotrebu.

Na Slici 13.1 prikazane su klase IP adresa kao i početak njihovih binarnih zapisa.



Slika 13.1: klase internet adresa.

Decimalna notacija

Na prethodnom primjeru smo prikazali početak binarnih zapisa IP adresa. Internet adrese, koji su 32-bitni brojevi češće se zapisuju u ljudima čitljivijem formatu tako da se pojedini okteti zapišu decimalno i odijele točkom. Decimalna 0 se pojavljuje kada su svi bitovi u oktetu 0, dok je 255 najveća vrijednost i označava 8 jedinica. Tablica na Slici 13.2 daje primjere nekih adresa u binarnoj i decimalnoj notaciji.

32-bit Binary Number	Equivalent Dotted Decimal
10000001 00110100 00000110 00000000	129.52.6.0
11000000 00000101 00110000 00000011	192.5.48.3
00001010 00000010 00000000 00100101	10.2.0.37
10000000 00001010 00000010 00000011	128.10.2.3
10000000 10000000 11111111 00000000	128.128.255.0

Slika 13.2: decimalna notacija.

Ovakav način zapisa IP adresa ima više mana. Decimalna notacija ne omogućava lako očitavanje tipa mreže. Tipovi se prepoznaju po rasponima vrijednosti prvog okteta, prema tabeli na Slici 13.3.

Class	Range of Values
A	0 through 127
B	128 through 191
C	192 through 223
D	224 through 239
E	240 through 255

Slika 13.3: rasponi prvih okteta u klasama IP adresa.

Jedino lako možemo prepoznati kada adresa nije klase A, budući da prvi bit u IP adresi klase A mora biti 0. Nadalje, klase IP adresa nisu jednakih veličina, kao što se vidi na Slici 13.4.

Address Class	Bits In Prefix	Maximum Number of Networks	Bits In Suffix	Maximum Number Of Hosts Per Network
A	7	128	24	16777216
B	14	16384	16	65536
C	21	2097152	8	256

Slike 13.4: Broj različitih IP adresa po klasama.

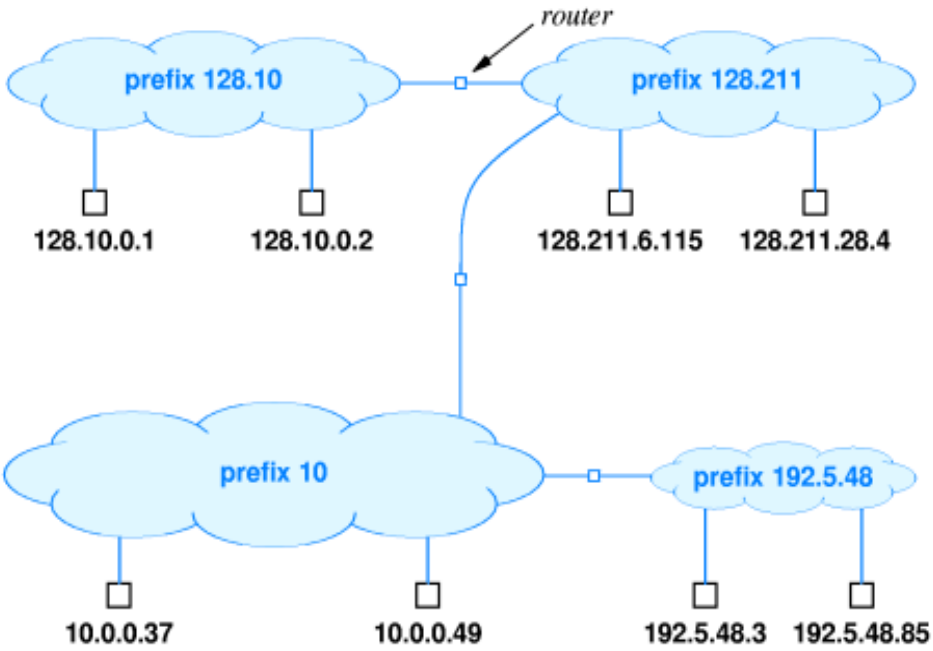
Besklasno adresiranje i CIDR notacija

Želimo definirati standard adresiranja koji neće patiti od ograničenja koje klase IP adresa postavljaju. Osnovni princip je da adresiranje unutar Interneta mora biti jedinstveno, a usmjeravanje olakšano hijerarhijskom strukturom IP adresa.

Adresa mreže se dobiva u koordinaciji s centralnom međunarodnom agencijom za upravljanje mrežom (International Assigned Number Authority). Sufiks adrese se dobiva od lokalnog administratora u kompaniji koja pruža uslugu pristupa Internetu, poznatijom kao ISP (Internet Service Provider).

Promotrimo primjer klasnog adresiranja sa Slike 13.5. Projektant dodjeljuje prefikse mrežama na osnovu pretpostavke o mogućem broju čvorova. Najčešće su adrese tipa B i C! Sufiksi se mogu proizvoljno dodjeljivati. Primijetimo da u jednoj mreži klase B ima od 0 do 65536 čvorova dok u mreži klase C ima od 0 do 256 čvorova. Za mrežu od 300 čvorova moramo potrošiti jednu adresu kase B, kao i za mrežu od 65535 čvorova. Takvo adresiranje bi bitno ograničavalo rast Interneta budući da bi mnogo adresa ostalo neiskorišteno.

Ovaj problem rješavamo uvođenjem novih apstrakcija. Uvodi se pojam *podmrežnog adresiranja* (subnet adressing) i pojam *besklasnog adresiranja* (classless addressing). Ideja je da se dijeljenje 32-bitne adrese na prefiks i sufiks može se realizirati na proizvoljnom mjestu. Na primjer, zamislimo mrežu s 9 čvorova. Klasa C ima mjesta za 256 sufiksa. ISP može za ovakve mreže koristiti prefikse s 28 bitova (što mu ostavlja 14 sufiksa).



Slika 13.5: heterogena mreža od 4 pod-mreže. Primjer klasnog adresiranja.

Kod besklasnog adresiranja, IP adresa A se dijeli na sufiks i prefiks pomoću adresne maske M. Adresna maska M je dodatni 32-bitni broj, koji počinje nizom uzastopnih jedinica a završava nizom uzastopnih nula. Jedinice u M označavaju mjesta u A koja pripadaju prefiksu, a nule u M označavaju mjesta u A koja čine sufiks. Besklasna IP adresa je sada uređeni par (A,M). Dakle, da bi zapamtili besklasnu adresu, moramo pamtiti ukupno 64 bita umjesto dosadašnjih 32. Prefiks mreže se računa po sljedećoj formuli, gdje znak & označava operaciju „logičko i“ po bitovima.

$$PR = (A \,\&\, M).$$

Besklasne adrese zapisuju se u ljudima čitljivijem formatu pomoću takozvane *CIDR notacije* (Classless Inter Domain Routing) – riječ je o proširenju standardne decimalne notacije gdje se s desne strane dodaje kosa crta i duljina prefiksa izražena kao decimalni broj. Uređeni par IP adrese i adresne maske se zapisuje u CIDR notaciji kao na primjer 128.10.2.3/16. Za naš primjer maska se sastoji od 16 jedinica i 16 nula, u decimalnoj notaciji ta maska bi bila

255.255.0.0

pa bi za zadanu IP adresu dala prefiks

128.10.0.0.

CIDR notacija je uvedena od strane IETF (Internet Engineering Task Force) 1993 godine zbog toga što je klasnim adresiranjem ubrzo bio blokiran adresni prostor. Opis standarda možete naći na adresi http://www.faqs.org/rfcs/rfc1519.html . Mi ćemo problem zauzimanja adresnog prostora ilustrirati nastavkom našeg model primjera.

Promotrimo ponovno primjer mreže sa 9 čvorova. Adresa 128.10.2.3 je primjer adrese klase B. U klasnoj notaciji ISP može ovu adresu dodijeliti jednom korisniku koji u svojoj mreži

može imati 65536 čvorova. Korištenjem CIDR notacije ISP može za veliku mušteriju dodijeliti ekvivalentnu CIDR adresu 128.10.2.3/16. Alternativno, ISP može dodijeliti dva prefiksa 128.10.2.16/28 i 128.10.2.32/28 za manje mušterije s maksimalno 14 čvorova.

Primijetimo, za masku /28 ili decimalno 255.255.255.240, korisnik ima 4 bita za dodjeljivanje lokalnih adresa. Decimalno je to 128.10.2.16 do 128.10.2.31, što na prvi pogled daje maksimum od 16 sufiksa. No, stvarni broj sufiksa je samo 14, jer su adrese 128.10.2.16 i 128.10.2.31 rezervirane.

Naime:
- *Adresa mreže* 128.10.2.16/28 je 128.10.2.16 i ona se ne smije koristiti kao adresa čvora.
- Adresa 128.10.2.31/28 (odgovara sufiksu u kojem se binarno pojavljuju sve jedinice) je *adresa za difuzijska adresa* (broadcast adress).

Paket koji izvana dođe na difuzijsku adresu mreže bit će dostavljen svim čvorovima u toj mreži. Slično se ponaša i paket koji je odaslan na multicast adresu. Pored ovih adresa, rezervirana je i takozvana l*oopback* adresa koja služi programerima za testiranje softvera koji koristi TCP/IP protokol. Loopback adrese imaju mrežni prefiks 127/8. Omiljena loopback adresa je 127.0.0.1. Adresa 255.255.255.255 je takozvana *limited broadcast* adresa koristi se za lokalno slanje paketa kroz cijelu lokalnu mrežu (prilikom starta mreže). Rezervirane IP adrese sažeto su prikazane u tabeli na Slici 13.6.

Prefix	Suffix	Type Of Address	Purpose
all-0s	all-0s	this computer	used during bootstrap
network	all-0s	network	identifies a network
network	all-1s	directed broadcast	broadcast on specified net
all-1s	all-1s	limited broadcast	broadcast on local net
127	any	loopback	testing

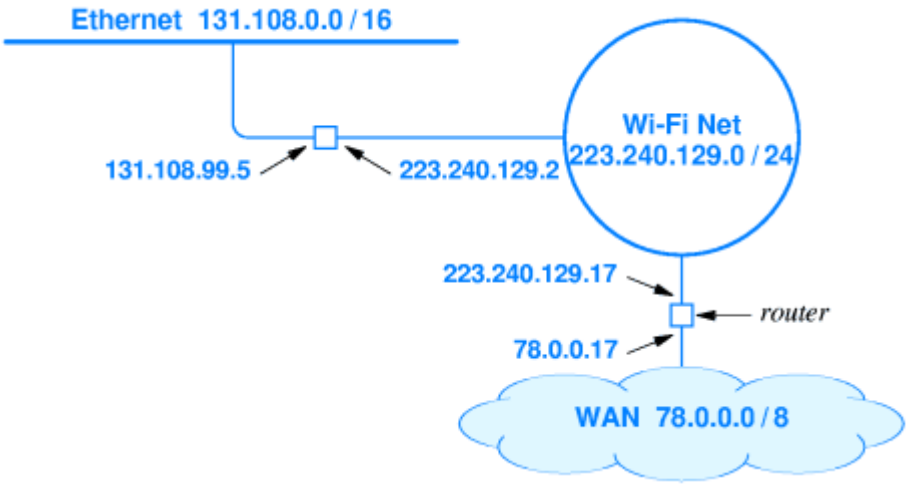
Slika 13. 6: rezervirane IP adrese.

Napomenimo još sljedeći detalj o usmjeravanju korištenjem maski pod-mreža. Maska pod-mreže se koristi samo za izlazne pakete. Paketi s udaljenih lokacija idu direktno adresi klase B ili CIDR adresi čvora primatelja.

Usmjernici i IP adresiranje

Prisjetimo se definicije razlike između usmjernika i paketne sklopke. Usmjernik je čvor u dvije mreže, a protokoli za usmjeravanje ga tretiraju kao i bilo koji drugi čvor u mreži. Zbog toga svaki usmjernik po TCP/IP protokolu ima svoju IP adresu. Štoviše, svaki usmjernik ima barem dvije pridružene IP adrese, budući da je:
- usmjernik čvor u više fizičkih mreža,
- svaka IP adresa ima prefiks koji označava fizičku mrežu.

Naglasimo da IP adresa ne označava računalo (host), nego spoj između računala i mreže. Računala, kao i usmjernici, mogu imati vezu s nekoliko mreža. Princip multi-homed host povećava robustnost i performanse mreže. Na Slici 13.7 vidi se primjer koji pokazuje IP adrese pridružene dvama usmjernicima koji povezuju tri mreže.



Slika 13.7: usmjernici i njima pridružene IP adrese.

Sažetak poglavlja 13

U ovom poglavlju smo prikazali strukturu IP adresa. Opisan je princip klasnog i besklasnog adresiranja te su diskutirana ograničenja svakog od modela. Prikazali smo adresnu shemu iz verzije 4 IP protokola (Internet Protocol version 4 - IPv4) . U IPv4 se koriste 32 bitne adrese i 32 bitne mrežne maske. Zbog eksponencijalnog rasta broja čvorova u Internetu ovaj princip adresiranja nije više dovoljan za pokrivanje čitavog adresnog prostora. Privremena rješenja manjka adresa uključuju tehnike kao network address translation (NAT) koje omogućavaju stvaranje privatnih mreža s rasponom IP adresa koje se ne prosljeđuju izvan dosega privatne mreže (više u Poglavlju 26, te na http://en.wikipedia.org/wiki/Network_address_translation). Pravo rješenje problema manjka adresnog prostora je dano uvođenjem 128 bitnih adresa u novom protokolu Internet Protocol version 6 (IPv6). Detalji, kao i problematika oko interakcije između IPv4 i IPv6 opisani su u Poglavlju 26.

14. Pretvaranje IP-adrese u hardversku – ARP

Sadržaj Poglavlja 14

IP adrese su virtualne budući da su realizirane softverski. Koriste ih viši slojevi protokola. Hardverski sloj ne razumije virtualne IP adrese. Okviri koji nemaju korektnu fizičku adresu ne mogu biti preneseni kroz fizičku mrežu. Zbog toga se javlja potreba za prevođenjem virtualne u fizičku adresu. U ovom poglavlju opisujemo protokol koji se koristi za prevođenje fizičkih adresa u IP adrese i obrnuto.

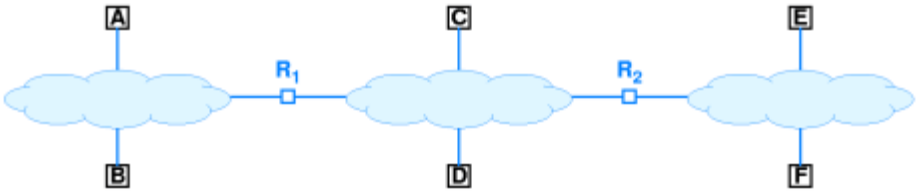
Adresiranje i slanje paketa

Zamislimo da dva programa žele razmijeniti podatke kroz mrežu. Softver iz komunikacijskog protokola generira paket koji sadrži adresu pošiljatelja i primatelja. Softver na svakom računalu (host) ili usmjerniku (router) koristi IP adresu za računanje sljedećeg skoka. Slanje

kroz fizičku mrežu zahtjeva određivanje fizičke adrese sljedećeg skoka. Svako računalo može odrediti samo fizičke adrese čvorova (host ili router) mreže u kojoj se nalazi.

Tehnike pretvaranja adresa

Preslikavanje između virtualne IP adrese i fizičke adrese se zove *pretvaranje adresa* (address resolution). Računalo (host) ili usmjernik (router) koriste prevođenje adresa samo kada šalju pakete unutar iste fizičke mreže. Adresa iz daleke fizičke mreže se nikada ne prevodi. Na primjer, ako na Slici 14.1 računalo A šalje datagram računalu F, tada usmjernik R_1 ne prevodi IP adresu od F, nego cijeli datagram prosljeđuje kroz srednju mrežu prema usmjerniku R_2.



Slika 14.1: Heterogene mreže povezane usmjernicima.

Postoje sljedeće tri osnovne tehnike prevođenja adresa:

- pretvaranje adrese *korištenjem tablice* (table lookup),
- pretvaranje adrese *direktnim računanjem* (closed-form computation),
- pretvaranje adresa *izmjenom poruka* (message exchange).

U nastavku ćemo detaljnije opisati svaku od tih tehnika.

Pretvaranje adresa korištenjem tablice

Tablica se sastoji od niza parova (P,H) virtualne IP adrese P i fizičke (hardverske) adrese H. Svaka fizička mreža ima svoju tablicu. Na primjer, Slika 14.2 prikazuje tablicu za prefiks 197.15.3.0/24. Pretraživanje velikih tablica može biti algoritamski zahtjevno.

IP Address	Hardware Address
197.15.3.2	0A:07:4B:12:82:36
197.15.3.3	0A:9C:28:71:32:8D
197.15.3.4	0A:11:C3:68:01:99
197.15.3.5	0A:74:59:32:CC:1F
197.15.3.6	0A:04:BC:00:03:28
197.15.3.7	0A:77:81:0E:52:FA

Slika 14.2: Pretvaranje IP adresa korištenjem tablica.

Pretvaranje adresa izračunavanjem

Koriste se brzo izračunljive (takozvane zatvorene) formule u Booleov-oj algebri. Ovakvo pretvaranje adresa je pogodno za prevođenje adresa koje nisu statičke (configurable addressing scheme). Naime, postoje mrežna sučelja čije se fizičke adrese biraju prilikom prve instalacije.

Na primjer, za mrežu s prefiksom 220.123.5.0/24 pogodna zatvorena formula glasi

$$H = P \,\&\, 0.0.0.255\ .$$

Ovdje je & opet operator „logičko i“ po bitovima. Fizičke adrese tada biramo u rasponu od 1 do 254.

Pretvaranje adresa izmjenom poruka

Primijetimo da računanje adresa opterećuje računala. Kao alternativu pretraživanju tablica ili izračunavanju fizičkih adresa, posao prevođenja virtualnih adresa se može distribuirati.

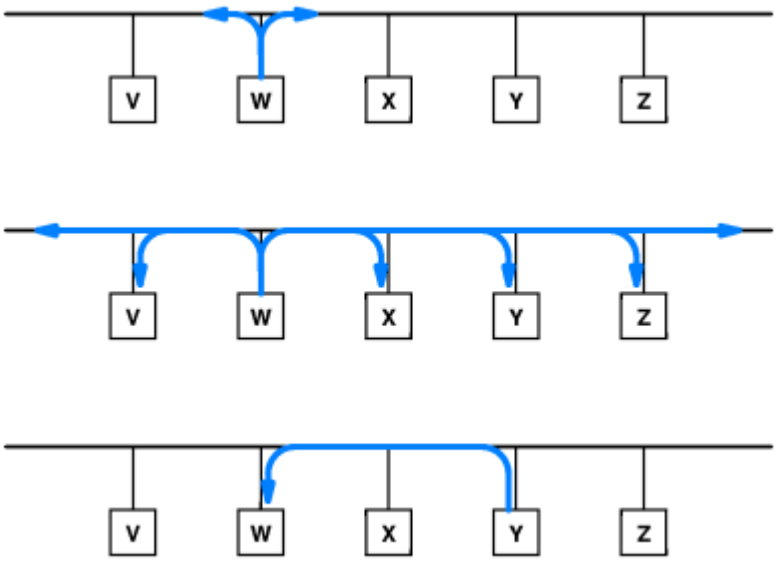
Kod pretvaranja adresa izmjenom poruka, računala dolaze do fizičke adrese

- tako da postave upit *poslužitelju za prevođenje* (resolution server).
- ili tako da svako računalo u mreži vraća odgovor na poruku kojom se traži njegova fizička adresa.

Ovakav princip prevođenja adresa je moguć samo u mrežama koje imaju *topologiju pogodnu za difuziju* (broadcast topology) kao što je Ethernet.

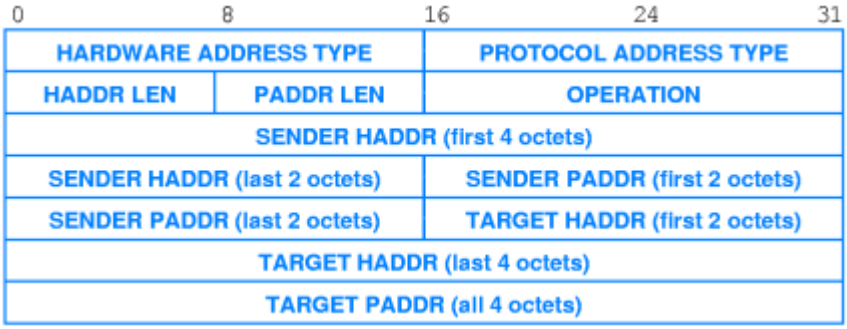
Protokol ARP

TCP/IP stog protokola može koristiti sva tri navedena tipa prevođenja virtualnih adresa. Tablice se najčešće koriste za prevođenje adresa u WAN-u. Prevođenje izračunavanjem se koristi za mreže koje podržavaju konfiguriranje fizičkih adresa. Prevođenje izmjenom poruka se koristi u LAN-ovima. Dio stoga koji rješava ove probleme se zove *ARP* (Adress Resolution Protocol). ARP standard specificira slanje ARP poruka kroz mrežu, a ARP zahtjev se šalje kao difuzija (broadcast). Odgovor se šalje u okviru koji je namjenjen samo čvoru koji je poslao difuzijsku poruku. Postupak je ilustriran Slikom 14.3.



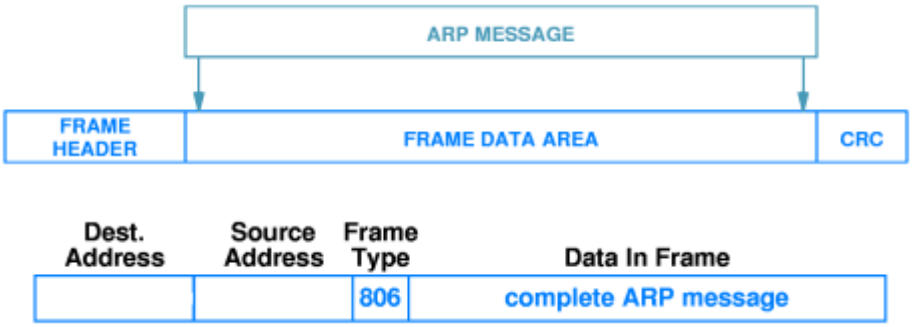
Slika 14.3: Slanje poruka u skladu s ARP protokolom.

ARP protokol se najčešće koristi za prevođenje 32-bitnih IP adresa u 48-bitne Ehternet adrese. Standard određuje samo opći oblik ARP poruke. Posebno, određeno je da se cijela ARP poruka transportira unutar fizičkog okvira kao njen korisni teret. Dakle riječ je o *enkapsulaciji* poruka. Format zaglavlja ARP poruke vidi se na Slici 14.4.



Slika 14.4: format poruka u ARP protokolu.

Koncept enkapsulacije predviđa da se u zaglavlju okvira specificira tip poruke koja je u okviru. Razlikovanje među različitim ARP porukama moguće je tek analizom ARP poruke. Mehanizam enkapsulacije, te korištenje tipa poruke u okviru ilustriran je Slikom 14.5.



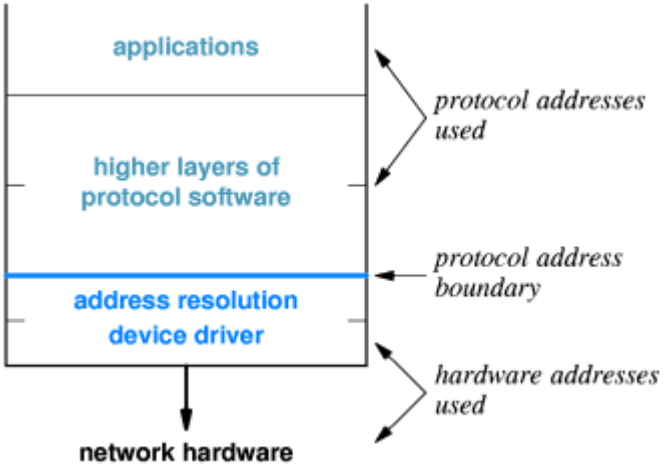
Slika 14.5: okvir, te zaglavlje okvira koji nosi ARP poruku.

ARP protokol sadrži različite algoritme za optimizaciju komunikacije. Ti algoritmi zasnivaju se na tome da računalo pohranjuje tablicu nedavno prevedenih adresa, te koristi tu tablicu da bi se izbjegla nepotrebna komunikacija. Osnovni princip je da:

- računala najčešće komuniciraju dvosmjerno,
- računala ne mogu memorirati proizvoljno mnogo prevedenih virtualnih adresa.

Iz svake ARP poruke s prevedenom adresom primatelj prvo ekstrahira pretvorenu adresu i ažurira tablicu prevedenih adresa. Tablica statistički smanjuje broj potrebnih ARP poruka.

U hijerarhiji stoga protokola ARP protokol odjeljuje više i niže slojeve protokola. ARP softver sakriva detalje fizičke mreže tako što omogućava višim slojevima protokola da komuniciraju korištenjem samo IP adresa – vidi Sliku 14.6.



Slika 14.6: mjesto ARP-a u stogu protokola, na granici između viših i nižih slojeva.

Sažetak Poglavlja 14

U ovom poglavlju smo opisali osnovni ARP protokol kojim se iz IP adrese dobiva fizička adresa u lokalnoj mreži. Najvažniji primjer realizacije ARP protokola koji smo prikazali odnosi se na mreže koje imaju difuzijsku topologiju kao Ethernet. Korištenjem ARP-a, 32-bitne IP adrese računala u Ethernetu prevode se u odgovarajuće 48-bitne hardverske adrese mrežnih sučelja tih računala.

15. IP datagrami i njihovo prosljeđivanje

Sadržaj Poglavlja 15

Do sada smo opisali:
- arhitekturu interneta,
- adresiranje u internetu,
- protokole za prevođenje adresa u internetu.

U ovom predavanju opisujemo osnovnu komunikacijsku uslugu koju pruža internet. Objasnit će se format paketa koji se šalju mrežom, te način na koji usmjernici (router-i) obrađuju i prosljeđuju takve pakete.

Bezspojna i spojna usluga

Tehnički cilj prilikom povezivanja raznorodnih mreža (internetworking) je omogućiti programu koji se izvršava na jednoj radnoj stanici slanje podataka programu koji se izvršava na drugoj radnoj stanici. Ta usluga se pruža neovisno o fizičkim svojstvima mreža u kojima se radne stanice nalaze. Razlikujemo *spojnu* (connection-oriented) i *bezspojnu* uslugu (conectionless). TCP/IP stog protokola nudi oba tipa usluge.

Bezspojna usluga je poopćenje principa izmjene paketa (packet-switching). Osnovne značajke su: svaki paket putuje neovisno i svaki paket sadrži informacije koje identificiraju primatelja i pošiljatelja. Paket putuje od usmjernika do usmjernika, dok ne dođe do onog usmjernika koji ga može proslijediti primatelju. Budući da usmjernici spajaju heterogene mreže, nije moguće koristiti fizički format paketa. Zbog toga se koristi univerzalni virtualni tip paketa.

IP Datagrami

TCP/IP protokol koristi naziv *IP datagram* kao oznaku za pojam internet paketa. Ima isti općeniti format kao i fizički okvir, dakle:

- sastoji se od zaglavlja i korisnog tereta;
- zaglavlje sadrži IP adrese koje su potrebne za usmjeravanje;
- količina podataka koje paket može prenositi nije fiksirana.

Veličinu datagrama određuje pošiljatelj. U standardu IP (version 4) datagram može imati između jednog i 64 K okteta podataka.

Datagrami prolaze kroz Internet slijedeći put od izvora do konačne destinacije putujući od usmjernika do usmjernika. Adresa sljedećeg skoka se bira korištenjem tablica usmjeravanja. Na Slici 15.1 (a) vidi se Internet koji se sastoji od četiri mreže i tri usmjernika. Slika 15.1 (b) prikazuuje tablicu usmjeravanja za usmjernik R_2 sa Slike 15.1 (a).

(a)

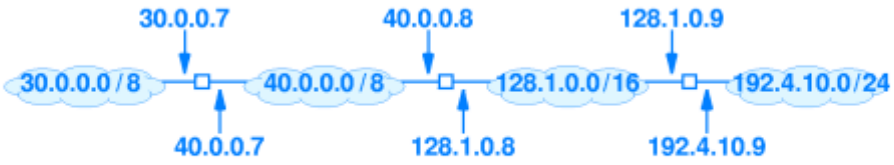
Destination	Next Hop
net 1	R_1
net 2	deliver direct
net 3	deliver direct
net 4	R_3

(b)

Slika 15.1: usmjeravanje IP datagrama.

IP adrese i tablice usmjeravanja

IP tablica usmjeravanja u stvarnosti je nešto kompleksnija nego na prethodnoj slici. Naime ona sadrži konkretne IP adrese i adresne maske. Svaki usmjernik može imati različite sufikse u različitim mrežama - IP ne zahtjeva uniformnost u ovakvom adresiranju. Slika 15.2 predstavlja detaljniju verziju slike 15.1 – vide se IP adresa svake mreže u CIDR notaciji, te IP adresa svakog usmjernika u svakoj od mreža. Također se vidi detaljni zapis tablice usmjeravanja za srednji usmjernik.



(a)

Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	deliver direct
128.1.0.0	255.255.0.0	deliver direct
192.4.10.0	255.255.255.0	128.1.0.9

(b)

Slika 15.2: IP adrese i tablice usmjeravanja.

Proces izbora sljedećeg skoka korištenjem tablica usmjeravanja se naziva *usmjeravanje* (routing ili forwarding) danog datagrama. Maska je broj koji se koristi za ekstrahiranje mrežnog dijela IP adrese. Binarni zapis omogućuje efikasno usmjeravanje. Osnovni korak usmjeravanja prema IP adresi D mogao bi se realizirati ovako:

```
If ((Mask[i] & D) == Destination[i] ) forward to NextHop[i] ;
```

Ovdje Mask[i], Destination[i], te NextHop[i] označavaju podatke zapisane u i-tom retku tablice usmjeravanja – vidi Sliku 15.2 (b). Znak & je opet operator „logičko i“ po bitovima.

Kakav je odnos između ciljne adrese i adrese sljedećeg skoka? Polje DESTINATION IP ADDRESS u zaglavlju datagrama uvijek sadrži IP adresu krajnjeg odredišta! Kada usmjernik prosljeđuje datagram, IP adresa sljedećeg skoka se nikada ne pojavljuje u zaglavlju. Svi putovi se računaju korištenjem IP adresa. Nakon određivanja IP adrese sljedećeg skoka, fizička adresa sljedećeg usmjernika se određuje korištenjem ARP protokola.

Best-effort delivery

Osim što je u IP protokolu specificiran format internet datagrama, određen je i princip na osnovu kojeg se rješava problem usmjeravanja datagrama. Pojam *najboljeg pokušaja* (best effort) opisuje smisao optimizacije protokola usmjeravanja.

Naglasimo da unatoč tome što je IP protokol zasnovana na principu najboljeg pokušaja, on ne osigurava rješenje za:

- duplikaciju datagrama,
- kašnjenje ili krivi redoslijed pristizanja datagrama,
- oštećenje datagrama,
- gubitak datagrama.

Ove greške rješavaju viši slojevi protokola.

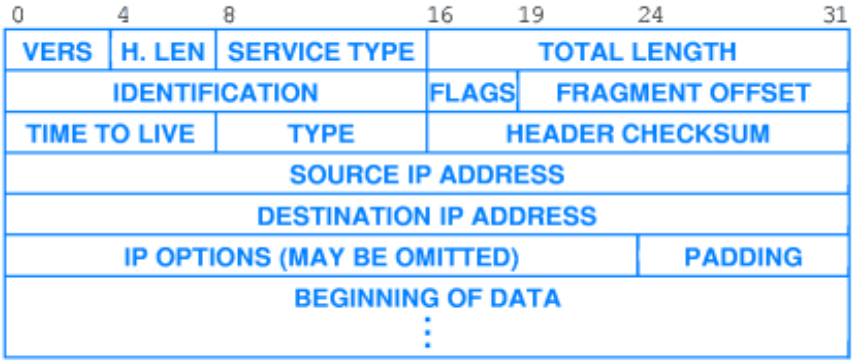
Pogledajmo sljedeći opis *best effort* mrežne usluge, koji je preveden s web stranice http://en.wikipedia.org/wiki/Best_effort_delivery .

U mreži koja pruža uslugu tipa *best effort communication* svi korisnici dobivaju najbolju moguću uslugu. To znači da oni dobivaju nespecificirani i promjenjivi *bit rate* i *delivery time*, koji ovise o trenutnom opterećenju mreže. Budući da mreža ne pruža usluge ispravljanja pogrešaka u slanju datagrama ili rješavanja problema gubitka paketa, te za te usluge ne mora prealocirati resurse, rezultirajući sustav funkcionira efikasnije uz bitno jeftiniju realizaciju čvorova. Primjer best-effort usluge je standardna poštanska služba. Konkretno, pošta dostavlja pisma korištenjem best-effort pristupa. Isporuka bilo kojeg danog pisma nije unaprijed određena i nikakvi poštanski resursi nisu unaprijed alocirani u poštanskom uredu. Poštar će pokušati isporučiti poruku na najbolji mogući način, best-effort, ali isporuka pisama može biti odgođena, na primjer u slučaju kada izrazito puno pošiljaka bude predano istovremeno u poštanskom uredu. Nadalje, pošiljatelj neće biti informiran o uspješnosti isporuke. S druge strane, ukoliko pošiljatelj plati dodatnu uslugu povratnice dobit će pouzdaniju uslugu – poruku o primitku koju će pismonoša tražiti od primatelja kao potvrdu uspješne isporuke.

Format zaglavlja IP datagrama

Format zaglavlja prikazan je Slikom 15.3. Svako polje u zaglavlju je fiksne veličine.

- Datagram počinje s poljem od 4-bita koje opisuje verziju protokola. Trenutna verzija je 4.
- Sljedeće 4-bitno polje opisuje koliko 32-bitnih "vrijednosti" ima u zaglavlju.
- Sljedeći oktet opisuje tip usmjeravanja koji se bira. Postoje mogućnosti biranja rute:
  - s minimalnim kašnjenjem,
  - ili s maksimalnom propusnošću.
- Sljedeća dva okteta sadrže polje koje opisuje ukupnu duljinu datagrama (zaglavlje i korisni teret).
- Značenje daljnjih nekoliko polja vezano je uz fragmentaciju i bit će opisano u idućem poglavlju.
- Polje TIME TO LIVE sadrži broj između 1 i 255, koji određuje koliko skokova s usmjernika na usmjernik datagram smije napraviti na svom putu prije nego što bude obrisan. Time se sprečava eventualno vječno kruženje datagrama po neispravnom kružnom putu.
- HEADER CHECKSUM je polje koje sadrži komplement do 1 sume svih 16-bitnih "vrijednosti" iz zaglavlja.
- SOURCE IP ADDRESS i DESTINATION IP ADRESS su pune IP adrese pošiljatelja odnosno primatelja datagrama.
- Nakon nekih opcionalnih dijelova, iza zaglavlja datagrama slijede podaci.



Slika 15.3: Format zaglavlja IP datagrama

Sažetak Poglavlja 15

IP datagram je osnovna jedinica za slanje podataka u Internetu. Po formatu je sličan fizičkom okviru, ali u zaglavlju ima samo IP adrese. IP softver koristi tablice usmjeravanja za određivanje IP adrese sljedećeg skoka. Veličina tablice je proporcionalna broju mreža. IP adresa sljedećeg skoka se nikada ne zapisuje u datagram.

16. IP enkapsulacija, fragmentacija i sastavljanje

Sadržaj Poglavlja 16

U ovom izlaganju opisujemo detalje prijenosa IP datagrama. Opisat će se kako radna stanica ili usmjernik šalje datagram kroz fizičku mrežu. Posebno ćemo promotriti slanje velikih datagrama kroz heterogeni niz mreža. Prije nego li nastavimo izlaganje navedimo konvenciju o terminologiji koju smo preuzeli s http://hr.wikipedia.org/wiki/Host .

Host je bilo koji uređaj povezan u računalnu mrežu (najčešće Internet) a koji može korištenjem standardnih protokola ostvariti komunikaciju s drugim sličnim uređajima (hostovima). Host je engleski izraz koji ima brojna značenja, ali u kontekstu uređaja spojenog na računalnu mrežu još nema odgovarajući prijevod, pa se koristi izvorni oblik izraza.

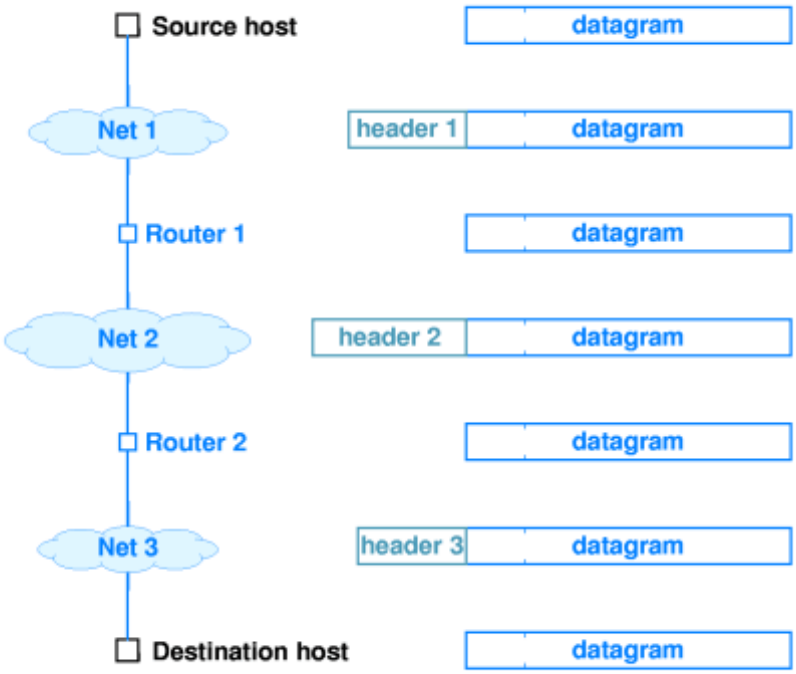
Host je u tom kontekstu najčešće konkretno osobno računalo, ali može biti i poslužitelj, usmjernik, odnosno bilo koji uređaj koji ima mogućnost komunikacije. U kontekstu internet protokola (IP), host može biti bilo koji uređaj koji ima dodijeljenu IP adresu.

Slanje datagrama i okviri

Po IP protokolu usmjernik ili host prvo biraju IP adresu sljedećeg skoka. Nakon toga slijedi prevođenje IP adrese u fizičku adresu. U Internetu koji se sastoji od heterogenih mreža format fizičkog okvira se može razlikovati od mreže do mreže.

Kako prenijeti datagram između različitih mreža? Koristi se tehnika *enkapsulacije*, slična tehnici koja je opisana u kontekstu protokola za razlučivanje adresa. Cijeli datagram se nalazi u polju namjenjenom prenošenju korisnih podataka. IP datagram se identificira u odgovarajućem polju za tip okvira u zaglavlju fizičkog okvira. Time ga primatelj razlikuje od ARP datagrama na primjer. Ideja enkapsulacije ilustrirana je Slikom 16.1.

Slika 16.1: Enkapsulacija datagrama.



Slika 16.2: Prijenos datagrama kroz internet.

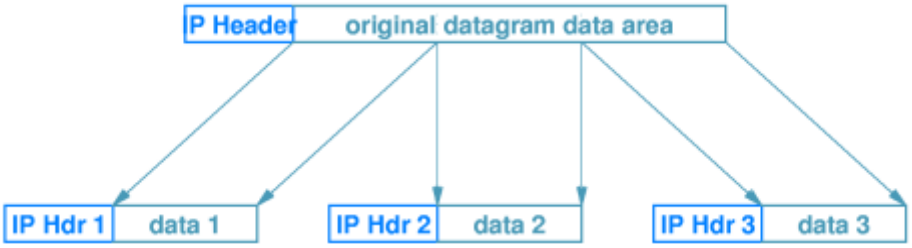
Enkapsulacija se primjenjuje samo na jednu transmisiju u danom trenutku. Primatelj na adresi sljedećeg skoka ekstrahira datagram iz fizičkog okvira. Ako datagram treba putovati dalje kroz još jednu mrežu, tada se stvara novi okvir. Na putu od polazišta do odredišta obavlja se niz enkapsulacija i de-enkapsulacija, kao što se vidi na Slici 16.2. Dakle, zaglavlja okvira se ne akumuliraju! Svaka mreža može koristiti drugu mrežnu tehnologiju, pa se zato formati okvira mogu razlikovati.

Podjela datagrama na fragmente

Svaka mrežna tehnologija određuje maksimalnu količinu podataka *maximum transmission unit* (MTU) koje je kroz tu mrežu moguće prenijeti unutar jednog okvira. Okviri koji su veći od MTU-a se ne prenose. Za prenošenje većih datagrama se koristi tehnika koja se zove *fragmentacija*. Ako je datagram veći od MTU, usmjernik ga dijeli u nekoliko manjih dijelova koji se zovu *fragmenti*. Na primjer, u mreži sa Slike 16.3, datagram veličine 1500 byte koji putuje od hosta H_1 do hosta H_2 morat će se pri prolasku kroz drugu mrežu podijeliti u dva fragmenta, budući da MTU u toj mreži iznosi samo 1000 byte.

Slika 16.3: potreba za fragmentacijom.

Fragment ima isti format kao i ostali datagrami. Bit informacije u polju FLAGS zaglavlja označava je li datagram fragment ili cjelovit datagram. Svaki fragment počinje kopijom zaglavlja originalnog datagrama. Odgovarajuća polja u kopiji zaglavlja se modificiraju da omoguće ponovno sastavljanje. Svaki fragment nosi samo jedan dio podataka iz originalnog datagrama. Postupak fragmentacije detaljnije je ilustriran Slikom 16.4.



Slika 16.4: podjela velikog datagrama u tri manja fragmenta.

Sastavljanje i identificiranje datagrama

Proces sastavljanja kopije originalnog datagrama iz fragmenata se naziva *sastavljanje* (reassembly). Budući da svaki fragment počinje s kopijom zaglavlja originalnog datagrama, svi fragmenti imaju istu adresu odredišta kao i originalni datagram. Datagram koji nosi zadnji fragment ima poseban bit u zaglavlju koji označava da je cijeli datagram stigao. IP protokol određuje da host-primatelj obavlja sastavljanje fragmenata.

Slika 16.5: Primjer za fragmentiranje i sastavljanje.

Čvorovi mreže na putu između pošiljatelja i primatelja prosljeđuju fragmente kao da je riječ o cjelovitim datagramima. Na primjer, u mreži sa slike 16.5, datagram od H_1 do H_2 veličine 1500 byte prolazi u komadu kroz prvu mrežu. Zatim ga usmjernik R_1 dijeli na dva fragmenta. Usmjernik R_2 ne sastavlja fragmente već ih svakog posebno šalje do H_2. Na kraju H_2 skuplja i sastavlja fragmente.

IP protokol ne garantira uspješno dostavljanje datagrama. Fragmenti datagrama mogu stizati u proizvoljnom redoslijedu. Kako IP softver uspijeva sastaviti fragmente koji stignu krivim redom? U zaglavlju IP datagrama postoji polje IDENTIFICATION koje jedinstveno određuje datagram. Svaki fragment istog datagrama sadrži kopiju istog identifikacijskog broja. Dodatno polje FRAGMENT OFFSET određuje redoslijed fragmenata.

Postoji mogućnost gubitka cijelih enkapsuliranih datagrama ili pojedinih fragmenata. Iako ponekad datagram ne može biti sastavljen, treba čuvati pristigle fragmente. Postoji mogućnost da drugi fragmenti samo kasne. IP određuje maksimalnu vrijeme čekanja fragmenata. Ukoliko svi fragmenti stignu prije isteka tog vremena, IP obavlja sastavljanje, u protivnom se svi pristigli fragmenti brišu. Rezultat sklapanja je po IP-u je sve ili ništa.

Fragmentiranje fragmenta

Što napraviti ako fragment dođe do mreže koja ima manji MTU nego li je veličina tog fragmenta? Protokol fragmentiranja je tako projektiran da omogućuje fragmentiranje fragmenata. IP ne razlikuje originalne datagrame od fragmenata ili podfragmenata. IP omogućava primatelju sklapanje svih fragmenata u originalni datagram bez prethodnog spajanja podfragmenata.

Sažetak Poglavlja 16

IP datagram se enkapsulira u fizički okvir i tako šalje kroz mrežu. Pošiljatelj stavlja cijeli datagram u polje fizičkog okvira koje je predviđeno za korisni teret. Pored toga pošiljatelj mora prevesti adresu sljedećeg skoka i unijeti je u odgovarajuće polje zaglavlja fizičkog okvira. Enkapsulacija se odnosi samo na trenutnu fizičku transmisiju. Mreže koje imaju različite MTU uzrokuju fragmentaciju datagrama. Primatelj obavlja sklapanje fragmenata.

17. Mehanizam dojave grešaka – ICMP

Sadržaj Poglavlja 17

IP definira princip *traženja najboljeg rješenja* (best effort delivery). Pritom IP ne garantira komunikaciju bez grešaka - to je problem koji rješavaju viši slojevi u stogu protokola. Ipak, tijekom svog rada IP otkriva i dojavljuje greške. U ovom predavanju opisujemo mehanizam dojave grešaka koji je ugrađen u IP. Isti mehanizam dojave greška pokazao se korisnim i za skupljanje dodatnih informacija o mreži.

Princip traženja najboljeg rješenja

IP definira mrežnu uslugu best-effort delivery. Kao što je već rečeno, prilikom komunikacije po principu best-effort delivery je moguće da datagrami budu duplicirani, izgubljeni, kasne ili stignu u slučajnom poretku.

Postavlja se pitanje: Je li za best-effort komunikaciju nužno imati mehanizam dojave grešaka? Primjer detekcije greške u prenošenju IP datagrama gdje nije potrebna dojava greške je neslaganje kontrolnog zbroja. U takvom slučaju se cijeli datagram briše, bez slanja ikakve poruke o grešci. Naime, budući da se ne zna se je li IP adresa pošiljatelja korumpirana ne bi bilo optimalno opterećivati mrežu možda uzaludnim dojavama o grešci.

Problemi koji su manje rizični od greške u transmisiji se ipak dojavljuju. U TCP/IP stogu protokola to rješava kolekcija protokola *ICMP* (Internet Control Message Protocol**).** Svaka standardna implementacija IP protokola mora sadržavati ICMP protokole. ICMP koristi IP protokol za slanje poruka o grešci. Osim dojave o grešci, ICMP šalje i druge informacije.

ICMP poruke

Popis svih ICMP poruka može se naći na slici 17.1. Postoje dvije vrste poruka, to su:

- poruke o grešci,
- informativne poruke.

Sada ćemo navesti neke standardne poruke o grešci koje navodi ICMP.

- *Source Quench*: Usmjernik šalje ovu poruku kada u međuspremniku nema mjesta. Pošiljatelj mora reagirati smanjivanjem brzine generiranja novih datagrama.
- *Time Exceded*: Generira se kada je usmjernik spustio polje TIME TO LIVE u datagramu na nulu ili kada host pri ponovnom sklapanju fragmentirane poruke prekorači REASSEMBLY TIMER
- *Destination Unreachable*: Šalje se kad usmjernik ustanovi da se datagram ne može isporučiti na svoje odredište. Iskazuje se razlika između nedostupnog hosta i nedostupne mreže.
- *Redirect*: Ukoliko usmjernik utvrdi da bi datagram trebao biti poslan po drugoj ruti, šalje ovu poruku. Može zahtijevati promjenu za host ili za mrežu.

Type	Name
0	Echo Reply
1	Unassigned
2	Unassigned
3	Destination Unreachable
4	Source Quench
5	Redirect
6	Alternate Host Address
7	Unassigned
8	Echo
9	Router Advertisement
10	Router Selection
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply
19	Reserved (for Security)
20-29	Reserved (for Robustness Experiment)
30	Traceroute
31	Datagram Conversion Error
32	Mobile Host Redirect
33	IPv6 Where-Are-You
34	IPv6 I-Am-Here
35	Mobile Registration Request
36	Mobile Registration Reply
37-255	Reserved

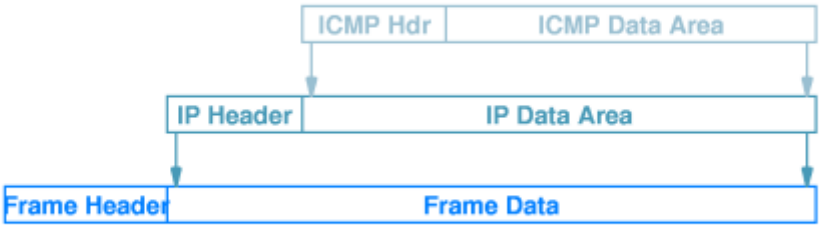
Slika 17.1: Internet Control Message Protocol ICMP.

Dalje slijede primjeri informativnih ICMP poruka.

- *Echo Request/Reply*: Poruka *echo request* se može slati ICMP software-u na bilo kojem čvoru. ICMP software bi trebao na primitak takve poruke reagirati slanjem *echo reply* poruke. Odgovor sadrži iste podatke kao i zahtjev.
- *Address Mask Request/Reply*: Host difuzijom šalje upit o adresnoj masci. Usmjernik koji primi poruku šalje korektni 32-bitni broj koji sadrži adresnu masku za tu mrežu.

Slanje ICMP poruka

ICMP koristi IP za slanje poruka. Koristi se dvostruka enkapsulacija, kao što se vidi na Slici 17.2. Dakle sama ICMP poruka pohranjuje se kao korisni teret u IP datagram, a taj IP datagram sprema se kao korisni teret u okvir fizičke mreže. Datagrami s ICMP porukama nemaju posebni prioritet. Ako pri slanju i usmjeravanju same ICMP poruke dođe do greške, tada se ne šalje nikakva nova poruka o greški.



Slika 17.2: enkapsulacija kod prijenosa ICMP poruka.

Korištenje ICMP u ping i traceroute

Dva poznata alata za ispitivanje povezanosti mreže, ping i traceroute, zasnivaju se na dosjetljivom korištenju ICMP poruka. Opisat ćemo kako rade ti alati, te koje ICMP poruke oni koriste.

Ping koristi *echo request* da bi testirao dostupnost nekog host-a. Ping šalje *echo request* paket najviše dva puta. Ukoliko nema odgovora niti na ponovno poslani paket ili ako stigne poruka *destination unreachable*, ping deklarira da ne postoji put do udaljenog stroja. ICMP software po protokolu uvijek mora odgovoriti na *echo request* upit. Doduše, neki sistem inženjeri blokiraju ove odgovore iz sigurnosnih razloga, pa u tom slučaju ping ne radi kako bi trebao.

Traceroute koristi TIME TO LIVE polje u zaglavlju IP datagrama za ispitivanje puta između dva stroja. Generiraju se datagrami s TTL vrijednostima postavljenim na 1,2, ... i tako dalje. ICMP poruke *time exceeded* se koriste za određivanje liste usmjernika između početnog i krajnjeg čvora. ICMP poruka putuje u IP datagramu, pa je moguće iz IP zaglavlja odrediti IP adresu usmjernika koji je poslao ICMP poruku.

Traceroute mora riješiti problem retransmisije, duplikacije ili gubitka IP paketa. Vrijeme retransmisije je parametar koji korisnik sam određuje. Problem puteva koji se dinamički mijenjaju nije lako riješiti. Traceroute je najkorisniji u mrežama sa stabilnim putevima.

Opisani postupak slanja IP datagrama s promjenjivim vrijednostima za TIME TO LIVE omogućuje da dobijemo odgovor od svih usmjernika koji su između polazišta i odredišta. No na taj način nećemo dobiti odgovor od samog odredišta, budući da ono nije dužno dalje prosljeđivati datagram. Da bi dobio odgovor i od konačnog odredišta, traceroute koristi jednu od sljedeće dvije metode:

- šalje ICMP *echo request* poruku,
- šalje UDP datagram nepostojećoj aplikaciji.

Microsoft-ova verzija traceroute-a, s imenom tracert, koristi prvi pristup. Tako kod svake retransmisije tracert prima ili ICMP *time exceeded* poruku ili ICMP *echo reply* od krajnjeg računala. Originalni UNIX-ov traceroute koristi UDP poruku (User Datagram Protocol) upućenu nepostojećem programu. Tako traceroute prima ICMP *time exceeded* ili ICMP *destination unreachable* od krajnjeg računala.

Primijetimo da pozivi tracert i traceroute na istom računalu mogu generirati različite odgovore. Naime:

- *echo reply* se šalje preko IP adrese sučelja preko kojega je stigao *echo request*.
- Poruka o grešci pri slanju UDP poruke može ići i preko sučelja s drugom IP adresom (ukoliko ih ima na host-u).

ICMP i računanje MTU-a za put

Fragmentacija datagrama je metoda kojom se rješava problem slanja velikih datagrama kroz hetoregene mreže. No usmjernici troše svoje procesorsko vrijeme na fragmentaciju. Taj gubitak vremena moguće je izbjeći ukoliko se odredi najmanji MTU na putu u mreži, te se od aplikacije traži da šalje još manje datagrame. Postavlja se pitanje: kako odrediti najmanji MTU za put?

Postupak određivanja najmanjeg MTU za put opet se može realizirati dosjetljivom primjenom ICMP poruka. Naime, u IP zaglavlju unutar polja FLAGS postoji zastavica čijim postavljanjem se osigurava da fragmentacija datagrama nije dozvoljena. Također, postoji ICMP poruka koja prenosi informaciju da je fragmentacija pokušana, ali nije bila dozvoljena.

IP software može odrediti MTU puta šaljući niz datagrama. Svaki datagram u zaglavlju ima postavljenu zastavicu koja sprečava fragmentaciju. Veličina datagrama varira tako da se odredi maksimalna veličina koja neće generirati ICMP poruku o neuspješnoj fragmentaciji.

Sažetak Poglavlja 17

Iako IP koristi semantiku najbolje usluge, u protokolu postoji mehanizam detekcije i dojave grešaka. Pored kontrolne sume za kontrolu zaglavlja, IP koristi niz protokola koji se zovu ICMP (Internet Control Message Protocol) za dojavu grešaka kao i za slanje informacija o mreži. ICMP protokol se može koristiti za testiranje Interneta, npr. ping, traceroute, itd.

18. Jednostavni transportni protokol – UDP

Sadržaj Poglavlja 18

U prethodnom predavanju smo prikazali bezspojnu paradigmu za slanje poruka o grešci u IP komunikaciji. U ovom poglavlju prikazat ćemo bezspojnu komunikacijsku paradigmu koja omogućava proizvoljnim *aplikacijskim programima* izmjenu poruka kroz heterogenu mrežu. Ako se osnovna komunikacijska usluga pruža po principu *best effort delivery* onda je nužno definirati protokol kojim, uz dodatnu cijenu, rješavaju neke neželjene posljedice takvog načina komunikacije. U ovom poglavlju opisujemo jedno rješenje koje omugućava bezspojnu komunikaciju između aplikacijskih programa metodom prosljeđivanja paketa u heterogenoj mreži. Protokol je formalno definiran u dokumentu RFC 768 kojega možete naći na adresi http://tools.ietf.org/html/rfc768 .

End-To-End transportni protokol

Protokol koji omogućava izmjenu podataka između dva programa nazivamo *End-To-End* transportni protokol. IP ne pruža takvu uslugu budući da IP zaglavlje identificira računala kao krajnje točke u komunikaciji (a ne programe). U TCP/IP stogu protokola End-To-End transportni protokoli se nalaze u sloju 4. Operativni sustav host računala pruža dodatnu uslugu prosljeđivanja poruke do krajnjeg odredišta, to jest do odgovarajućeg programa.

UDP protokol

U TCP/IP stogu protokola *User Datagram Protokol* - UDP pruža uslugu bezspojne End-To-End komunikacije. UDP komunikacija se može karakterizirati kao:

- *End-To-End*: UDP protokol može identificirati ciljni program na danom računalu.
- *Bezspojna*: Sučelje koje UDP pruža aplikacijama implementira paradigmu bezspojne komunikacije.
- *Orijentirana na slanje poruka*: Osnovna komunikacijska jedinica u UDP protokolu se naziva poruka.
- *Best-effort delivery*: UDP pruža uslugu koja koristi best-effort semantiku IP protokola.
- *Bez ograničenja na broj interakcija*: UDP dopušta programu slanje poruka prema neograničenom broju drugih programa, te primanje poruka od neograničenog broja drugih programa ili komunikaciju s točno jednim programom.
- *Neovisnost o operativnom sustavu*: UDP pruža mehanizam za identifikaciju ciljnog programa. Mehanizam je neovisan o operativnom sustavu host računala.

Bezspojna usluga

Bezspojna usluga znači da aplikacijski program koji koristi UDP protokol ne mora prije slanja podataka uspostaviti komunikacijski kanal s ciljnim programom. UDP ne koristi kontrolne poruke. Posebno:

- UDP dopušta proizvoljno dugo čekanje između slanja dviju poruka.
- Ukoliko dva programa prestanu komunicirati, nikakve daljnje poruke o tome se ne šalju.

Osnovna značajka UDP protokola je efikasnost.

Sučelje za prosljeđivanje poruka

UDP pruža aplikacijskim programima sučelje za slanje poruka. UDP ne dijeli poruke u pakete i ne sastavlja poruke po primitku. Svaka poruka koju neki program pošalje direktno se prenosi kao zasebni IP datagram kroz Internet do krajnjeg odredišta. Posljedice su sljedeće.

- Programer se može osloniti na protokol za kontrolu transporta poruke (to je pozitivno).
- Poruke se ne fragmentiraju, pa je veličina IP datagrama gornja ograda za veličinu UDP poruke (to je negativno).

Rezultat korištenja opisanog sučelja je da UDP poruke mogu uzrokovati neefikasno korištenje fizičke mreže. Ukoliko aplikacijski program šalje male UDP poruke omjer korisnog tereta i zaglavlja će biti loš. Protokol takvo korištenje mreže ne ograničava. Ukoliko program šalje jako velike poruke, UDP protokol ne osigurava da datagrami koji se šalju neće biti veći od MTU-a mreže. Anomalija UDP protokola je što slanje (pre)velikih poruka usporava komunikaciju. Ponekad UDP šalje poruke koje IP protokol mora fragmentirati već na polaznom računalu.

Semantika UDP komunikacije

UDP koristi IP protokol za svu komunikaciju. Aplikacijskim programima pruža istu (naslijeđenu) semantiku best-effort komunikacije koju ima i IP protokol. Zbog toga je nasljeđen i problem gubitka poruke, dupliciranja poruke, te gubitak redosljeda slanja poruka. UDP ne identificira i ne rješava probleme koji nastaju zbog ovih svojstava IP protokola. UDP je pogodan za aplikacije koje šalju audio ili video podatke. Nije pogodan za aplikacije gdje je redoslijed događaja bitan.

Unicast, difuzija u grupi, difuzija

UDP dopušta sljedeće komunikacijske paradigme

- 1 prema 1
- 1 prema mnogo
- mnogo prema 1
- mnogo prema mnogo

Komunikacijski model mnogo-prema-mnogo znači da grupa aplikacijskih programa može grupno izmjenjivati poruke.

Komunikacijski model 1-prema-mnogo bi se mogao riješiti slanjem pojedinačnog paketa svakom primatelju. UDP optimizira ovakvu komunikaciju korištenjem IP adresa za difuziju ili difuziju u grupi. Na primjer, lokalna difuzija može biti riješena slanjem na adresu 255.255.255.255. Takav način komunikacije je posebno pogodan za korištenje na Ethernet mrežama.

Interakcija s operativnim sustavom

UDP ne može koristiti isti mehanizam koji koristi i operativni sustav za identifikaciju pojedinačnih aplikacijskih programa. Različita računala koriste različite mehanizme: *process number, job number, task identifier* ... UDP definira apstraktni niz identifikatora za procese zvan *protocol port numbers* ili neformalno port. Protocol port numbers su neovisni o operativnom sustavu. Na računalima na kojima je implementiran UDP, operativni sustav pruža uslugu povezivanja procesa i port-ova.

UDP standard među ostalim određuje da pojedini portovi pripadaju određenim aplikacijama. Na primjer:

- port 7 pripada aplikaciji echo,
- port 37 pripada aplikaciji timeserver.

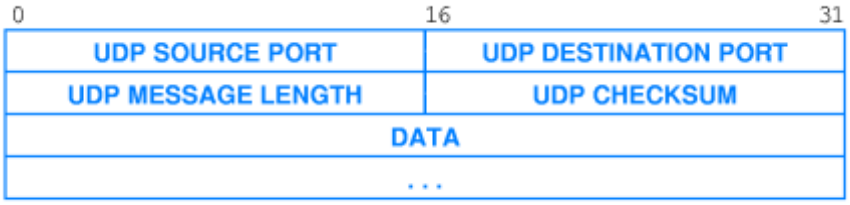
Sva računala na kojima se izvršava UDP prepoznaju standardne port-ove neovisno o operacijskom sustavu. Ako dođe poruka na port 7, operativni sustav mora proslijediti poruku onom programu kojim se izvršava echo servis.

Za realizaciju 1-1 komunikacije po UDP protokolu aplikacijski program se veže na lokalni port i specificira IP adresu udaljenog računala. UDP tada prosljeđuje samo one poruke koje stižu sa specificirane IP adrese na dani port. Za realizaciju paradigme mnogi-1 po UDP protokolu aplikacijski program određuje lokalni port i informira UDP da pošiljatelj može biti proizvoljan. UDP tada prosljeđuje tom programu sve poruke koje dođu na dani port. UDP dopušta samo jednom programu da se u danom trenutku veže na dani lokalni port.

Format UDP datagrama

UDP poruka se naziva *user datagram.* Sastoji se od dva dijela

- zaglavlja koje određuje polazni i krajnji aplikacijski program,
- korisnog tereta, gdje se nalaze podaci koji se šalju.

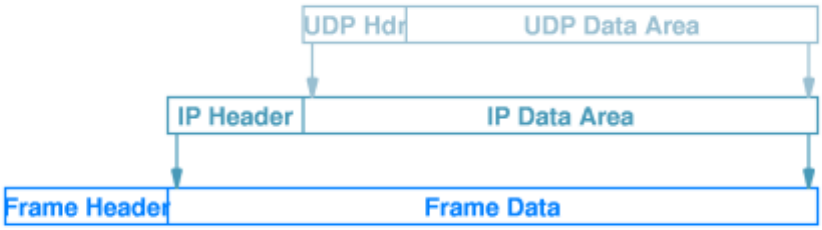


Slika 18.1: Format UDP datagrama.

UDP pseudo-zaglavlje sadrži polje od 16 bitova koje se zove UDP CHECKSUM. Korištenje je opcionalno. UDP pretpostavlja da se IP adrese pošiljatelja i primatelja nalaza u IP datagramu. Zbog toga u pseudo zaglavlju stoje samo brojevi port-ova. Ova optimizacija smanjuje količinu dodatnih podataka koji se prenose, ali povećava mogućnost greške u transmisiji. Na primjer, UDP ne bi mogao detektirati slučaj u kojem je IP proslijedio poruku na krivu adresu. Da bi se ipak spriječila takva greška, UDP-ova kontrolna suma pokriva i IP adresu pošiljatelja.

Enkapsulacija UDP poruka

UDP koristi IP za transport. Enkapsulacija se provodi slično kao i kod ICMP protokola i ilustrirana je Slikom 18.2. Dakle, UDP datagram enkapsulira se u IP datagram, a taj IP datagram dalje se ekapsulira u okvir fizičke mreže.



Slika 18.2: UDP enkapsulacija.

Sažetak Poglavlja 18

UDP protokol pruža aplikacijskim programima takozvanu End-To-End uslugu slanja poruka. UDP pruža istu semantiku best-efort komunikacije kao i IP protokol. UDP poruke se enkapsuliraju u IP datagrame. UDP zaglavlje ne sadrži IP adrese pošiljatelja i primatelja. Pored grešaka koje su moguće unutar IP protokola, moguće je da poruke koje imaju kontrolnu sumu postavljenu na 0 stignu na krivo računalo. Korištenjem protokolskih brojeva portova za identifikaciju aplikacijskih programa, UDP postaje neovisan o operativnom sustavu host-a.

19. Složeniji transportni protokol – TCP

Sadržaj Poglavlja 19

Osnovno pitanje kojim se bavimo u ovom poglavlju glasi: je li moguće realizirati pouzdanu komunikaciju korištenjem IP datagrama? Pokazat će se da TCP protokol rješava problem gubitka i kašnjenja paketa bez stvaranja pretjeranog „dodatnog" opterećenja usmjernika i fizičkih mreža.

Pouzdani transportni protokol

Jedna od osnovnih pretpostavki pri razvoju računalnih aplikacija je pouzdanost. Operativni sustav garantira pouzdanost I/O operacija. Posebno, on garantira da podaci neće biti izgubljeni ili duplicirani. Traži se transportni protokol koji će garantirati isti tip komunikacijske usluge koju osigurava standardni operativni sistem. To u prvom redu znači sljedeće:

- podaci moraju stizati u poretku u kojem su poslani,
- ne smije biti duplikacije ili gubitka podataka.

Osobine TCP protokola

Transmission Control Protocol - TCP je najpopularniji općeniti transportni protokol u Internetu. Njegove osnovne značajke su:

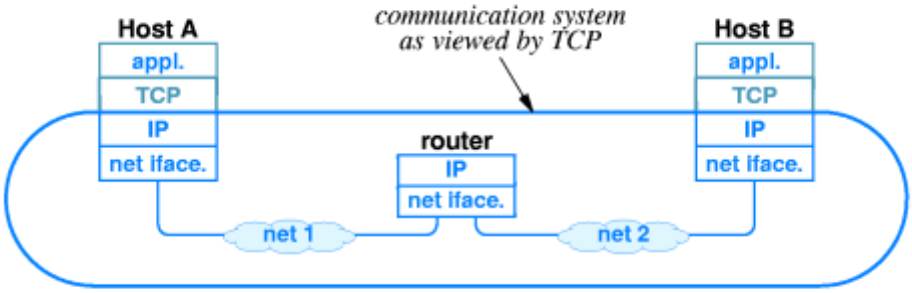
- *Spojna usluga*: aplikacijski program mora prvo zatražiti vezu, a tek onda slijedi prijenos podataka.
- *Point-To-Point*: ili čvor-čvor komunikacija, znači da svaka TCP veza ima točno dva kraja.

- *Pouzdanost*: protokol osigurava da će podaci doći u redoslijedu u kojem su poslani i da neće biti gubitka ili duplikacije podataka.
- *Puni dupleks*: Oba aplikacijska programa mogu slati podatke u svakom trenutku. Omogućuje i pretpostavlja optimizaciju korištenjem komunikacije u oba smjera.
- *Stream Interface*: Sučelje koje TCP pruža aplikacijskim programima omogućuje slanje kontinuiranih nizova okteta kroz čvor-čvor vezu. TCP ne definira pojam zapisa koji ima fiksnu veličinu. Primatelj ne mora trošiti pristigle podatke u komadima kako ih je pošiljatelj poslao, već ih opet čita kao kontinuirani niz okteta.
- *Pouzdano otvaranje veze*: Pri stvaranju čvor-čvor veze, oba čvora moraju pristati na komunikaciju. Paketi koji kasne iz prethodnih veza među tim čvorovima neće interferirati s novom vezom.
- *Pouzdano zatvaranje veze*: TCP osigurava da će svi poslani podaci biti isporučeni prije nego li se veza raskine.

End-To-End komunikacija i datagrami

TCP je End-To-End protokol jer omogućava direktnu logičku vezu između *aplikacijskih programa*. Veza je virtualna budući da je realizirana u softveru. Niti fizički hardver niti IP protokol ne pružaju nikakvu podršku spojnoj komunikaciji. No ipak, TCP softver pruža aplikacijskim programima dojam spojne veze u skladu sa Slikom 19.1. To znači sljedeće.

- TCP poruke se enkapsuliraju u IP datagrame.
- TCP tretira IP kao metodu prenosa paketa.
- TCP software je nužan samo na krajnjim čvorovima. Ostatak Interneta je sistem koji prenosi poruke bez da ih interpretira ili mijenja njihov sadržaj.



Slika 19.1: Komunikacija preko prividne direktne komunikacijske linije.

Problem pouzdanosti

TCP koristi niz tehnika kojima nastoji osigurati pouzdanu komunikaciju. Od tih tehnika najvažnije su: *adaptivna retransmisija, algoritam prozora, algoritam "trostrukog rukovanja"* i *algoritam za kontrolu zagušenja*.

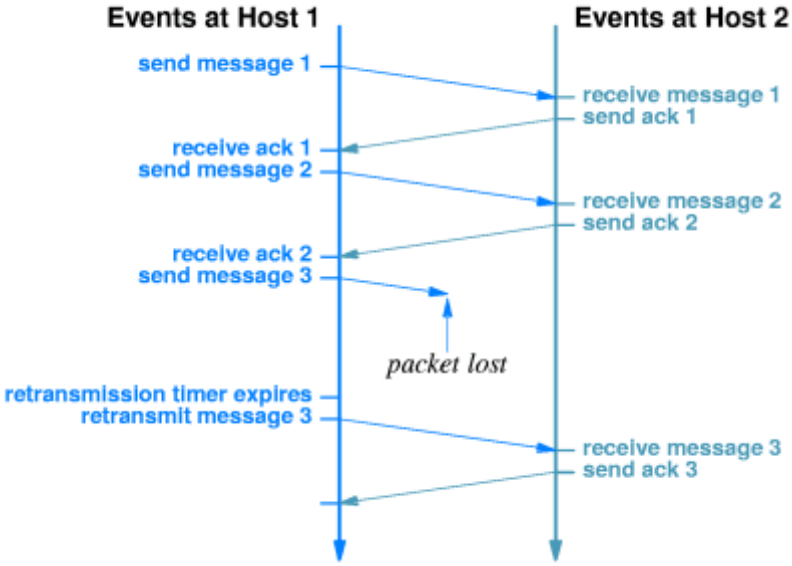
O svakoj od gore navedenih tehnika reći ćemo nešto više u idućim odjeljcima ovog poglavlja. Za sada napomenimo da pouzdanost veze najviše ugrožavaju:

- nepouzdanost IP protokola,
- ponovno pokretanje (reboot) računala.

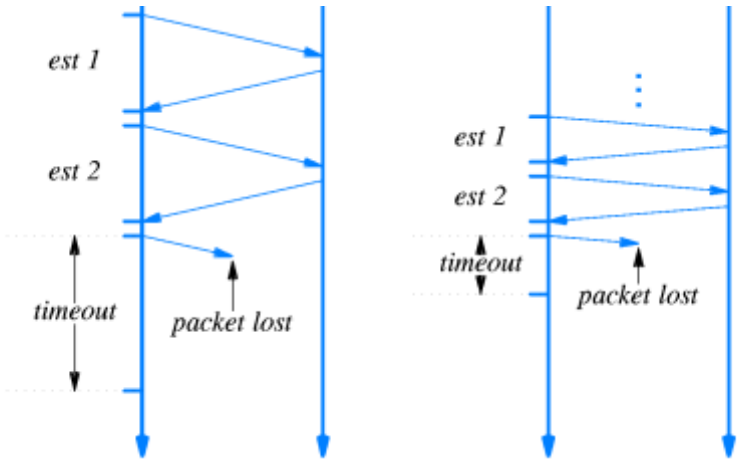
U vezi s nepouzdanošću IP protokola: zamislimo da dva računala otvore vezu, komuniciraju i nakon toga zatvore vezu i otvore novu. Kako tretirati pakete koji kasne iz prethodne veze, a nastali su retransmisijom? U vezi s reboot-om računala: zamislimo da dva računala stvore vezu i nakon toga jedno od njih izvrši reboot. Kako riješiti problem što računalo koje je izvršilo reboot ne zna ništa o vezi, a računalo koje nije izvršilo reboot još je uvijek vezu drži valjanom? Kako odbacivati pakete koji su nastali prije reboot-a?

Gubitak paketa i retransmisija

Prije TCP protokola, za ponovno slanje izgubljenog datagrama (retransmisiju) korišteni su algoritmi s fiksnim vremenima retransmisije (time-out). Osnovna ideja algoritma ilustrirana je Slikom 19.2. Za svaku primljenu poruku, primatelj šalje pošiljatelju potvrdu (acknowledgement - ack) da je primio tu poruku. Ako potvrda ne stigne nakon nekog određenog vremena, pošiljatelj zaključuje da je poruka izgubljena, te je šalje ponovo. No pokazalo se da se rješenje s fiksno određenim vremenom transmisije loše skalira u eksponencijalno rastućem Internetu.



Slika 19.2: primjer algoritma retransmisije.



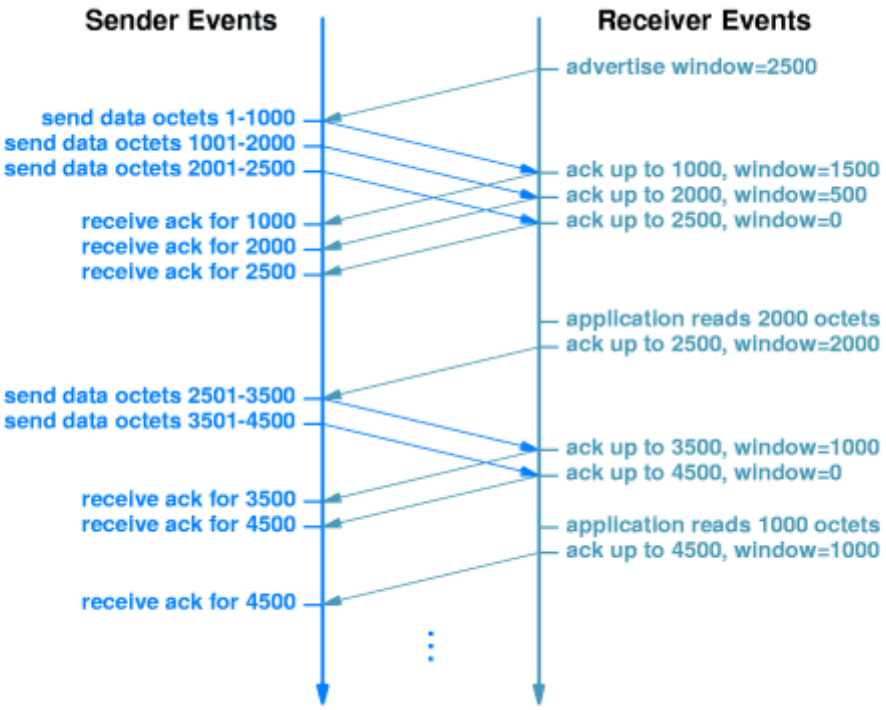
Slika 19.3: određivanje vremena retransmisije.

TCP koristi nešto sofisticiraniju varijantu algoritma retransmisije koja se zove *adaptivna retransmisija*. TCP procjenjuje takozvani *round-trip-delay* za svaku otvorenu vezu. To se postiže mjereći vrijeme od slanja podatka do primanje potvrde. Round-trip-delay se procjenjuje korištenjem odgovarajućih težinskih statističkih funkcija - time se rješava problem naglih oscilacija kašnjenja (bursts). Prilikom retransmisije, vrijednost timeout-a se postavlja da bude malo dulja od prosječnog round-trip delay-a. Znači, u vezama koje imaju različiti round-trip delay, time-out se isto razlikuje - to je ilustrirano Slikom 19.3.

Međuspremnici, kontrola toka i prozori

Algoritam prozora rješava problem kontrole toka podataka. Definirajmo prvo pojam prozora. TCP na strani primatelja prima podatke tako da iz sprema u međuspremnik. Aplikacijski program na strani primatelja uzima podatke iz međuspremnika, te tako ponovno oslobađa prostor u njemu. Veličina neiskorištenog dijela međuspremnika u danom trenutku naziva se *prozor*.

Zajedno s potvrdom primitka poruke, pošiljatelj primatelju šalje trenutnu veličinu prozora. Potvrda se računa relativno prema nizu podataka (stream-u) koji se šalje. Ako pošiljatelj šalje podatke brže nego što ih je primatelj u stanju obraditi, veličina prozora će pasti na nulu. Kad pošiljatelj primi poruku o prozoru veličine nula, on mora prestati slati podatke sve dok mu primatelj ponovo ne dojavi da prozor ima pozitivnu veličinu. Primjer rada algoritma prozora vidi se na Slici 19.4.

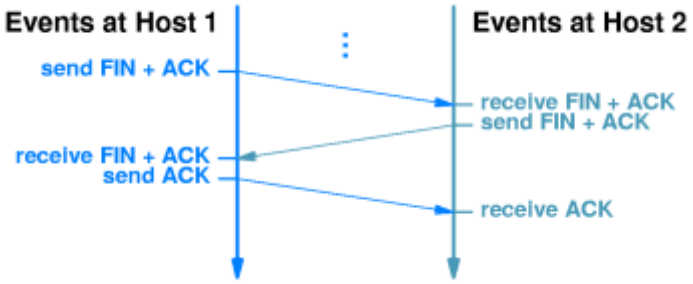


Slika 19.4: algoritam prozora.

Algoritam "trostrukog rukovanja"

Algoritam *trostrukog rukovanja* rješava problem pouzdanog otvaranja i zatvaranja veze u TCP protokolu. TCP poruke koje se koriste za otvaranje komunikacije se nazivaju *SYN*

segmenti, a za zatvaranje se koriste *FIN segmenti*. Veza se identificira slučajnim brojem koji se generira pri uspostavljanju veze. Na Slici 19.5 prikazano je trostruko rukovanje prilikom zatvaranja veze.



Slika 19.5: algoritam trostrukog rukovanja.

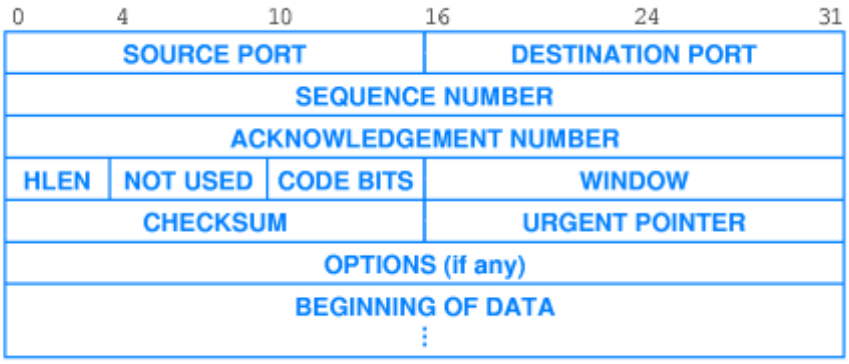
Kontrola zagušenja

U modernim mrežama kašnjenje ili gubitak podataka najčešće je uzrokovano zagušenjem (congestion), a ne hardverskom greškom. Protokoli koji koriste algoritam retransmisije mogu dodatno pogoršati problem zagušenja time što ubacuju u mrežu nove kopije paketa u trenutku kad ih usmjernici nemaju gdje pohraniti.

TCP komunikacija se temelji na među-spremnicima (prozorima). TCP pokušava kontrolirati zagušenje tako da dodatno smanji veličinu prozora onda kad je došlo do kašnjenja (dakle najvjerojatnije do zagušenja). Pri retransmisiji nakon gubitka podataka, TCP dijeli podatke u vrlo male pakete, čiju veličinu eksponencijalno povećava dok ne dosegne polovinu stvarnog prozora. Zatim TCP usporava dinamiku slanja i linearno povećava veličinu paketa dok ne dosegne stvarnu veličinu prozora. Ako se u međuvremenu opet dogodi zagušenje, algoritam počinje od početka.

Format TCP zaglavlja

TCP poruke imaju format koji se sastoji od zaglavlja i dijela s podacima. Format zaglavlja prikazan je na Slici 19.6. Jedan te isti oblik koristi se za sve poruke (podaci, potvrda, FIN i SYN). Format TCP zaglavlja smišljen je tako da omogući prije spomenutu duplex komunikaciju. To znači da TCP može koristiti jedan datagram za slanje više poruka istovremeno, na primjer za potvrdu prijema, objavu prozora i slanje izlaznih podataka.



Slika 19.6: format TCP zaglavlja.

U nastavku slijedi detaljniji opis pojedinih polja u zaglavlju TCP poruke.

- Polja ACKNOWLEDGMENT NUMBER i WINDOW se odnose na dolazeći stream. ACKNOWLEDGMENT NUMBER sadrži SEQUENCE NUMBER sljedećeg paketa, a WINDOW daje informaciju o slobodnom dijelu međuspremnika za podatke koji polaze iz čvora kojem se šalje potvrda.
- Polje SEQUENCE NUMBER se uvijek odnosi na izlazeći stream i pokazuje na prvi oktet koji se nalazi u segmentu.
- CHECKSUM sadrži kontrolnu sumu za TCP zaglavlje i podatke.

Sažetak Poglavlja 19

TCP je najvažniji transportni protokol u TCP/IP stogu. On pruža aplikacijskim programima End-To-End spojnu komunikaciju koja je:

- pouzdana,
- omogućava kontrolu zagušenja,
- full-duplex,
- orijentirana na slanje kontinuiranih nizova podataka (streams).

TCP protokol koristi IP protokol za komunikaciju, a sve poruke šalje koristeći isti format datagrama.

20. Usmjeravanje u internetu

Sadržaj Poglavlja 20

Do sada smo objasnili kako se datagrami šalju kroz internet korištenjem usmjernika. Postavlja se pitanje: kako se same tablice usmjeravanja stvaraju i obnavljaju? Ili drugim riječima, kako se informacije o tablicama usmjeravanja šalju kroz mrežu? Odgovor na to pitanje daju protokoli usmjeravanja. U ovom poglavlju će se opisati način na koji rade osnovni protokoli usmjeravanja.

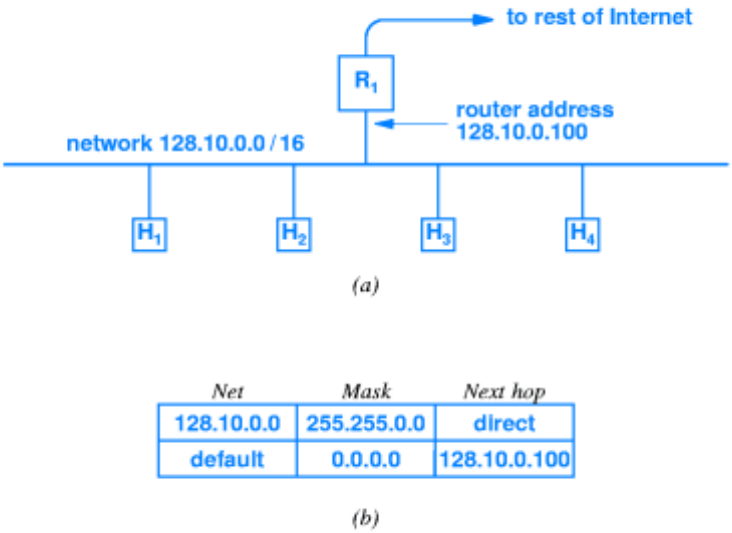
Statičko i dinamičko usmjeravanje

Usmjeravanje u IP protokolu se dijeli u dvije grube kategorije *statičko* i *dinamičko* usmjeravanje. Statičke tablice usmjeravanja se učitavaju kod pokretanja operativnog sustava i ne mjenjaju se ukoliko se ne dogodi greška. Algoritam dinamičkog usmjeravanja počinje u trenutku pokretanja operativnog sustava na isti način kao i algoritam statičkog usmjeravanja. Nakon toga posebni *route propagation software* prilagođava lokalne tablice usmjeravanja u ovisnosti o informacijama koje dolaze s drugih čvorova.

Statičko usmjeravanje

Kod statičkog usmjeravanja nije potreban dodatni softver za usmjeravanje. Takvo usmjeravanje ne troši procesorsko vrijeme i ne opterećuje mrežu. No ono je relativno nefleksibilno prema promjenama topologije mreže. Većina PC-a u Internetu koriste statičke tablice usmjeravanja. Naime, PC je obično dio neke lokalne mreže koja je spojena s

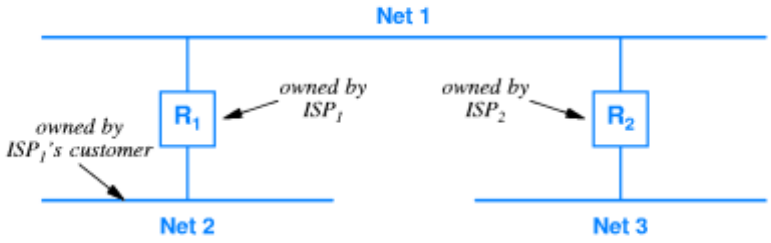
Internetom preko jednog usmjernika. U skladu sa Slikom 20.1, tablica usmjeravanja unutar PC-a ima samo dva retka, koji kažu da se primatelju s IP adresom iz lokalne mreže podaci šalju direktno, a u slučaju svih ostalih adresa podaci se prosljeđuju usmjerniku.



Slika 20.1: Statičko usmjeravanje.

Dinamičko usmjeravanje

Algoritam usmjeravanja kao na Slici 20.1, koji koristi statičke tablice usmjeravanja i default rute, uglavnom nije dovoljan za korištenje na većini usmjernika. Kada se povežu mreže koje pripadaju dvama ISP-iovima nužno je izmjenjivanje informacija o usmjeravanju između dvije mreže. Inače na Slici 20.2 usmjernik u mreži 2 ne bi mogao imati informacije o putovima u mreži 3.



Slika 20.2: potreba za razmjenom informacija o usmjeravanju.

Dinamičko usmjeravanje odvija se tako da na svakom usmjerniku postoji softver koji skuplja informacije o putovima prema čvorovima u drugim mrežama. Usmjernici informiraju "susjedne" usmjernike o putovima prema čvorovima koje oni mogu doseći. Na osnovu ovih informacija se ažuriraju lokalne tablice usmjeravanja.

Opisani postupak je sličan dinamičkom usmjeravanju za WAN-ove koje smo obradili u Poglavlju 10. Postavlja se pitanje je li moguće paradigmu iz WAN-ova proširiti na cijeli Internet? Pokazuje se da to ipak nije moguće zbog veličine Interneta, te da je potrebno naći nešto drukčije (kompromisno) rješenje koje se zasniva na konceptu autonomnog sustava.

Koncept autonomnog sustava (AS)

Kada bi svaki usmjernik imao u lokalnoj tablici usmjeravanja put do svakog čvora u Internetu, protok informacija za održavanje takvih tablica usmjeravanja bi zagušio fizičku mrežu. Stoga je nužno da se volumen prometa routing poruka kontrolira uvođenjem hijerarhije protokola usmjeravanja. Usmjernici i mreže u Internetu su podjeljeni u grupe. Svi usmjernici u jednoj grupi izmjenjuju tablice usmjeravanja. No samo neki izabrani usmjernici iz dane grupe izmjenjuju routing informacije s usmjernikom iz druge grupe.

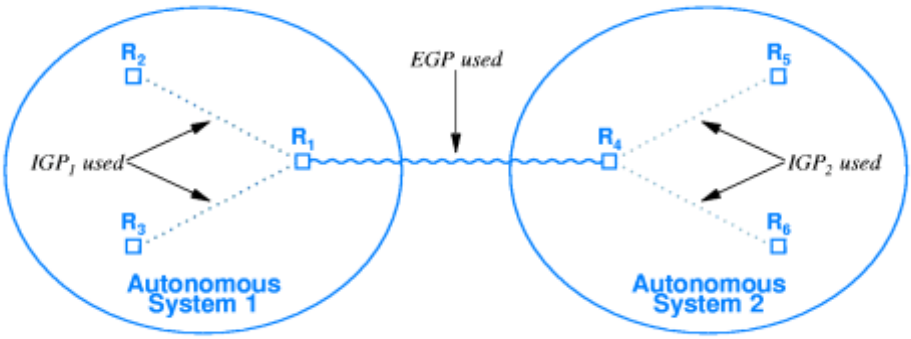
Autonomni sustav je termin koji se koristi za opisivanje grupe usmjernika koji se nalaze pod kontrolom jednog "administratora" i koji izmjenjuju informacije o usmjeravanju. Informacije o rutama se optimiziraju prije nego se proslijede drugim autonomnim grupama. Arhitektura protokola je dovoljno fleksibilna da podnosi veliku lepezu različitih konfiguracija. Jedna organizacija može imati jedan ili više autonomnih sustava. Izbor veličine autonomnog sustava se donosi na osnovu ekonomskih, tehnoloških ili administrativnih razloga.

Protokoli za usmjeravanje u Internetu

Svi protokoli za usmjeravanje u Internetu se mogu podjeliti u dvije kategorije.

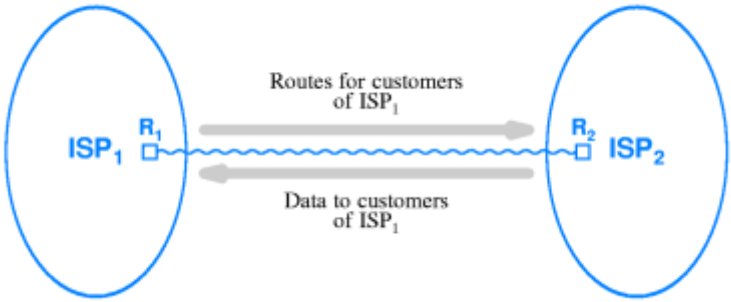
- *Interior Gateway Protocols (IGP).* Koriste ih usmjernici unutar jedne autonomne grupe. Postoji nekoliko IGP-a i svaki autonomni sustav može izabrati svoj (jedan) IGP.
- *Exterior Gateway Protocol (EGP).* Usmjernik u autonomnoj grupi koristi EGP za izmjenjivanje informacija o rutama sa usmjernikom u drugoj autonomnoj grupi. EGP-i su fleksibilniji i uzrokuju manje opterećenje fizičke mreže.

EGP protokoli omogućuju administratorima implementiranje *policy constraints,* dakle kontrolu nad informacijama koje se šalju u drugi autonomni sustav. IGP koristi *routing metriku* za nalaženje optimalnog puta. EGP nalazi put do svake destinacije. Ne može naći optimalni put jer ne može uspoređivati *routing* metrike u različitim IGP-ima. Arhitektura usmjeravanja u Internetu, te uloga IGP odnosno EGP vidljiva je na Slici 20.3.



Slika 20.3: autonomni sustavi, uloga IGP odnosno EGP.

Širenje informacija o putovima kroz Internet omogućuje korištenje tih putova. Podaci na neko odredište stižu tek nakon što se objavi put do tog odredišta. Dakle, promet podataka prema zadanom odredištu teče točno u suprotnom smjeru od poruka za usmjeravanje – ideja je ilustrirana Slikom 20.3.



Slika 20.3: putovi i tok podataka

Border Gateway Protocol (BGP)

Border Gateway Protocol verzija 4 (BGP-4) je najpopularniji Exterior-Gateway-Protokol koji se koriste u Internetu. ISP-ovi koriste BGP-4 za izmjenu informacija o putovima s drugim ISP-ovima. Pouzdano usmjeravanje datagrama je moguće samo ako su globalne informacije o putevima konzistentne. Na primjer, organizacija RIPE (Resaux IP Europeens) u tu svrhu održava registar informacija u putevima i o ISP-ovima koji sadržavaju dane čvorove.
Osnovne značajke BGP protokola su sljedeće.

- *Usmjeravanje među AS*. Put kroz Internet opisuje se na višoj razini, kao niz autonomnih sustava kroz koje treba proći.
- *Policy-constraints*. BGP dopušta i pošiljatelju i primatelju ograničavanje skupa putova koji se proslijeđuju.
- *Mogućnost "tranzitnog" usmjeravanja.* BGP klasificira svaki AS kao *tranzitni sustav* ukoliko taj AS dopušta prosljeđivanje informacije susjednim AS-ovima, odnosno kao *sustav umetak* (stub) ukoliko ne on ne prosljeđuje podatke o putovima.
- *Pouzdan transport.* BGP koristi TCP protokol za komunikaciju usmjernika u jednom AS s usmjernikom u drugom AS.

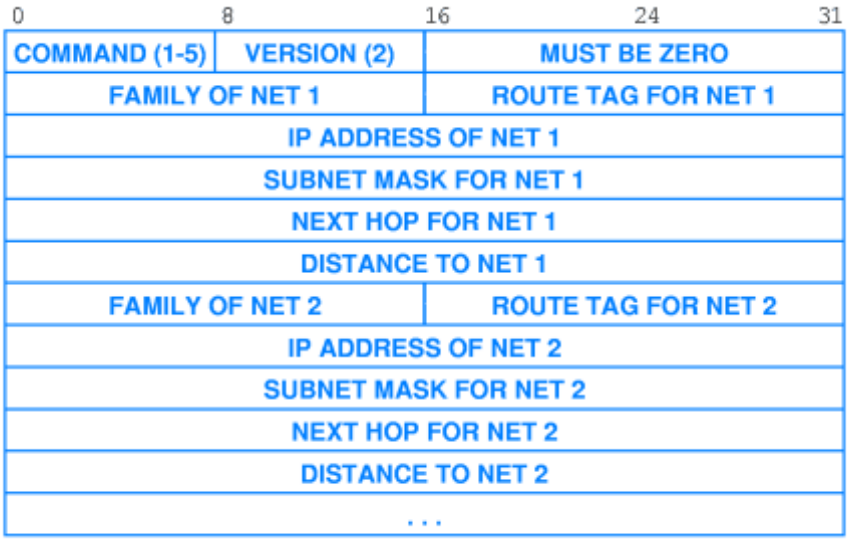
Routing Information Protocol (RIP)

Routing Information Protocol (RIP) jedan od dva najpopularnija Interior-Gateway-Protokola. Implementiran na većini UNIX sistema kao program routed.

Osnovne značajke RIP protokola su sljedeće.

- *Usmjeravanje unutar AS-a.* Prosljeđuje informacije između usmjernika unutar istog AS.
- *Metrika Hop-Count* (origin-one-counting). Duljina puta mjeri se kao broj skokova između različitih mreža unutar istog AS, počevši od 1.
- *Nepouzdani transport*. Koristi UDP protokol za sve poruke.
- *Koristi difuziju ili difuziju u grupi*. Projektiran za korištenje s Ethernet tehnologijom.
- *Optimizirano korištenje standardnog puta* (default route propagation). Dovoljno je konfigurirati jedan usmjernik da ima default put prema ISP-u. RIP tada prosljeđuje informaciju o default putu svim ostalim usmjernicima u AS-u. Svaki datagram koji se šalje izvan AS-a će odmah biti prosljeđen ISP-u.

- *Koristi Distance-Vector algoritam.* Riječ je o otprilike istom algoritmu kao što je bio opisan u Poglavlju 10. Pored informacije o svim čvorovima koji se mogu dosegnuti šalju se i informacije o duljini puta.
- *Pasivna verzija za host-ove.* Samo usmjernik može slati informacije u putovima. Host-ovi mogu pasivno ažurirati svoje tablice putova.



Slika 20.4: format RIP paketa.

Na slici 20.4 prikazan je format za poruke koje se koriste u RIP. Jedna RIP poruka sastoji se od liste odredišta i pripadnih udaljenosti. Kao podrška za CIDR, uz IP adresu svakog odredišta navodi se i pripadna adresna maska.

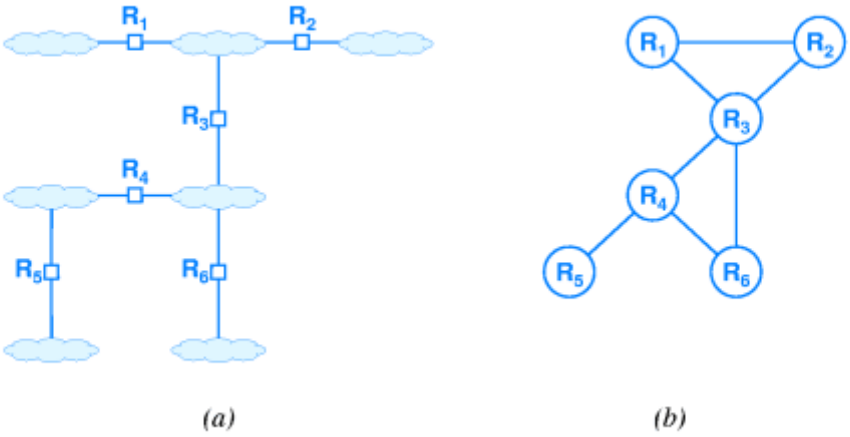
Open Shortest Path First Protocol (OSPF)

Distance-Vector algoritam kojeg koristi RIP ne nalazi uvijek put između dva čvora. Primjer za to je takozvani Count-to-Infinity problem. Također, RIP zahtijeva glomazne poruke i nije dovoljno skalabilan. Zato se kasnije pojavio i drugi IGP protokol: *Open Shortest Path First Protocol* (OSPF), koji je stabilan i u velikim i u malim AS-ovima.

OSPF koristi *link-status* algoritam za ažuriranje tablica usmjeravanja. Postupak izgleda otprilike ovako.

- Svaki usmjernik unutar AS mora periodički izmjenjivati informacije sa susjednim usmjernicima i zatim poslati na difuziju svoj link-status (stanje veza prema drugima).
- Svaki usmjernik na osnovu takvih difuzija samostalno rekonstruira graf cijelog AS, te koristi Dijkstrin algoritam (najkraći putovi među čvorovima – vidi Poglavlje 10) za računanje nove tablice usmjeravanja.

Za *Link-state* usmjeravanje AS se apstrahira kao graf u čijem svakom čvoru je usmjernik. Bridovi su putovi između usmjernika (ustvari mreže). Način apstrahiranja vidljiv je na Slici 20.5. Lijevi dio slike predstavlja AS koji se sastoji od više fizičkih mreža i usmjernika. Desni dio slike prikazuje odgovarajući graf. OSPF dopušta definiranje hijerarhije unutar AS-a. Zbog toga se OSPF koncept može bolje prenijeti na konfiguraciju u kojoj se nalazi veliki broj usmjernika nego ijedan drugi IGP.



Slika 20.5: idealizirani prikaz skupa mreža povezanih usmjernicima.

Usmjeravanje poruka za difuziju u grupi

Do sada smo opisivali usmjeravanje prema određenoj adresi (unicast usmjeravanje). Osnovna značajka unicast usmjeravanja statičnost adresa čvorova. IP grupe za difuziju (multicast grupe) stvaraju se dinamički. Dakle aplikacijski program se može pridružiti grupi u proizvoljnom trenutku i napustiti je u svakom trenutku. IP grupe za difuziju su anonimne. Takva grupa određuje samo skup primatelja. Svaki program može poslati datagram grupi. Difuzija u grupi se koristi za teleconferencing (male grupe), webcasting (tipično velike grupe), i tako dalje.

Problem usmjeravanja poruka za difuziju u grupi prilično je težak. Predloženi su neki protokoli, na primjer *Distance Vector Multicast Routing Protocol*, ili *M-OSPF*, no oni ne rješavaju problem na zadovoljavajući način.

Internet Group Multicast Protocol (IGMP) predstavlja djelomično rješenje, i on se koristi za komunikaciju između host-a i usmjernika. Po tom protokolu je host, a ne program član grupe. Host informira najbliži usmjernik da pristupa određenoj grupi onda kad neki njegov program pristupi toj grupi. Slično, host informira usmjernik da napušta grupu kada zadnji program na host-u napusti grupu.

Sažetak Poglavlja 20

Hostovi i usmjernici koriste IP tablice usmjeravanja. Hostovi često koriste statičko usmjeravanje, dok usmjernici uglavnom koriste dinamičko usmjeravanje. Internet se dijeli u skup AS-ova. Unutar AS-a se koristi IGP protokol, a između AS-a EGP. Unicast usmjeravanje je dobro riješeno: RIP i OSPF koriste se kao IGP, a BGP kao EGP. Unatoč većem broju predloženih protokola za usmjeravanje poruka za difuziju u grupi (multicast usmjeravanje), za sada na Internetu ne postoji opće prihvaćena tehnologija za tu svrhu.

IV. KORIŠTENJE MREŽA, MREŽNE APLIKACIJE

21. Interakcija klijenata i poslužitelja, osnovne aplikacije u Internetu

Sadržaj Poglavlja 21

U dosadašnjim predavanjima je opisano kako funkcionira hardver i protokoli koji su nužni za pouzdanu komunikaciju u Internetu. U ovom predavanju ćemo opisati osnovnu softversku arhitekturu koje se koriste u razvoju mrežnih aplikacija. Ta osnovna arhitektura zasniva se na interakciji klijenata i poslužitelja, te ona dopušta brojne varijacije. Na primjerima DNS, E-maila, FTP, NFS i WWW pokazat ćemo neke od mogućih programerskih rješenja koji su zasnovani na ovakvom modelu.

Aplikacijski softver i Internet

Mrežne aplikacije mogu se promatrati kao primjeri distribuiranog računarstva, tj. modela računarstva u kojem se osnovni problem veće složenosti rješava skupom računala ili procesa koje rade jednostavnije zadatke i međusobno izmjenjuju informacije. Sa strane operativnog sustava osnovna podrška je dana u obliku principa *višedretvenosti* aplikacija (multitreading). Ideja višedretvenog programiranja jest u tome da se program sastoji od više jedinica koje se samostalno mogu izvoditi. Programer ne mora brinuti o redoslijedu njihova izvođenja, već to obavlja sam operacijski sustav. Komunikacija među dretvama je jednostavna i brža u odnosu na komunikaciju među procesima. Ipak, osnovni principi razvoja aplikacija koje koriste neki dijeljeni resurs – na primjer isti komunikacijski kanal – je na apstraktnom nivou isti.

Internet daje mrežnim aplikacijama osnovni komunikacijski okvir. Internet protokoli ne određuju tip usluga koje se nude u mreži, niti mogu inicirati ili prihvatiti komunikaciju. To je funkcionalnost koju pružaju mrežne aplikacije. Mrežne aplikacije određuju format u kojem će se informacije prikazivati, te daju mehanizme za izbor i pristup informacijama. Za ostvarivanje komunikacije u Internetu moraju sudjelovati dva programa (procesa) kao dva kraja komunikacijskog kanala.

Arhitektura klijent-poslužitelj

Model distribuiranog računarstva u kojem jedan program - poslužitelj (server) pasivno čeka drugi program - klijenta koji inicira komunikaciju naziva se model klijent-poslužitelj (client-server). Informacije mogu teći u oba smjera, ali komunikaciju uvijek inicira klijent. Pojam poslužitelj se odnosi na program koji je dio modela klijent-poslužitelj, a ne na računalo. Samo računalo na kojem se poslužiteljski program izvršava naziva se poslužiteljsko računalo (serversko računalo). Unatoč brojnim varijacijama, interakcija klijenta i poslužitelja najčešće ima sljedeće značajke.

Značajke klijenta.

- Proizvoljni program postaje privremeno klijent u trenutku iniciranja komunikacije.
- Pokreće ga korisnik, a njegovo izvršavanje je vezeno za trajanje sesije.
- Aktivno otvara kontakt s poslužiteljem.

- Može pristupiti različitim poslužiteljima, ali aktivno komunicirati samo s jednim poslužiteljem.
- Ne treba mu posebni hardver za izvršavanje.

Značajke poslužitelja.

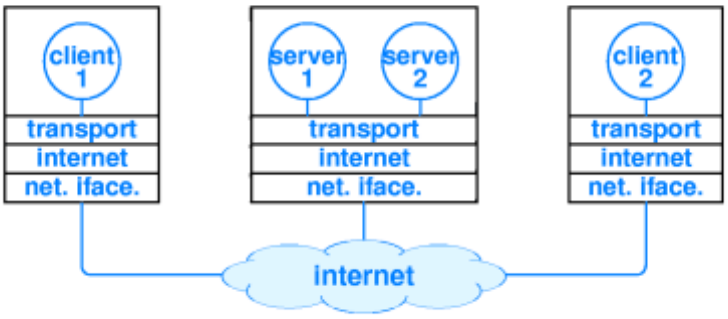
- To je posebni program koja je stvoren za pružanje jedne usluge. Pri tome on može komunicirati s više klijenata istovremeno.
- Automatski se pokreće pri pokretanju operativnog sustava i izvršavanje mu nije vezano uz korisničku sesiju.
- Pasivno čeka na kontakt.
- Prima kontakt od različitih klijenata, ali im daje samo jednu uslugu.
- Potreban mu je jaki hardver i poseban operativni sustav.

Da bi klijent i poslužitelj mogli komunicirati, potreban im je komunikacijski softver. Budući da govorimo o aplikacijama na Internetu, taj komunikacijski softver zapravo je TCP/IP stog protokola. Komunikacija se odvija preko transportnog sloja, pa sve to izgleda kao na slici 21.1.

Slika 21.1: komunikacija klijenta i poslužitelja pomoću TCP/IP stoga protokola.

Poslužitelji i poslužiteljska računala

Poslužiteljska računala mogu istovremeno pružati više usluga. Za svaku uslugu je potreban poseban poslužitelj. Komunikacijski protokol pruža mehanizam koji klijentima omogućava identifikaciju poslužitelja. Ideja je ilustrirana Slikom 21.2.

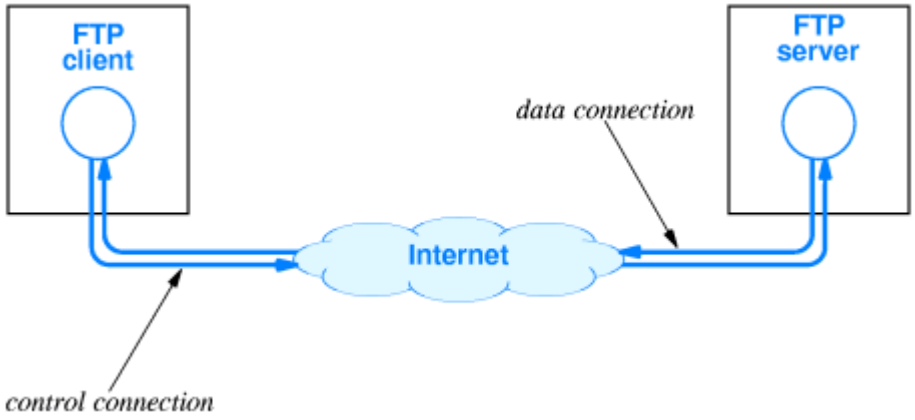


Slika 21.2: više poslužitelja na istom poslužiteljskom računalu.

Višedretvenost je nužna za interakciju klijent-poslužitelj u kojoj više klijenata istovremeno pristupa poslužitelju. Glavna dretva poslužitelja pasivno čeka na sljedećeg klijenta i tada poslužitelj stvara novu dretvu koja obrađuje zahtjeve tog klijenta, a glavna dretva se vraća pasivnom čekanju. Transportni protokoli pridružuju identifikacijski broj svakom poslužitelju i svakom klijentu. Kombinirajući ove identifikacijske brojeve transportni protokol određuje kojoj kopiji poslužitelja pripada koji klijent.

Prvi primjer aplikacije tipa klijent-poslužitelj: FTP

File Transfer Protocol (FTP) predstavlja jedan od najstarijih primjera aplikacije s klijentom i poslužiteljem. Ta aplikacija služi za prijenos datoteka s jednog računala na drugo. Na primjer, klijent traži datoteku koja se nalazi na poslužiteljevom računalu. Poslužitelj šalje kopiju. Klijent i poslužitelj uspostavljaju „kontrolnu“ TCP vezu i komuniciraju pomoću aplikacijskog protokola FTP. Za samo slanje datoteke koristi se druga, takozvana “podatkovna” TCP veza. Cijela arhitektura prikazana je na Slici 21.3.



Slika 21.3: prijenos datoteke pomoću FTP.

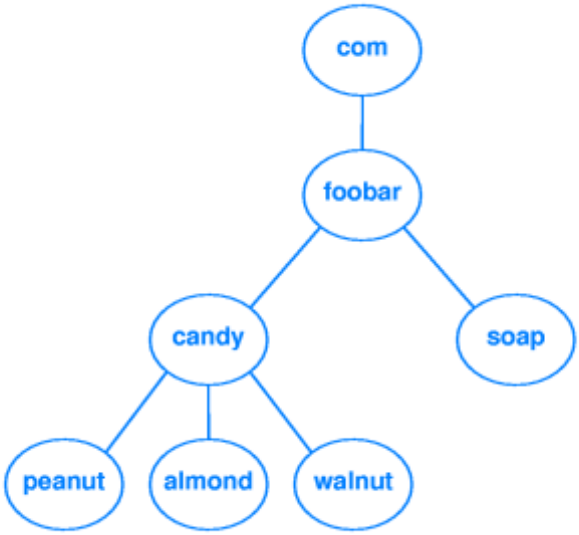
Drugi primjer aplikacije tipa klijent-poslužitelj: DNS

Paradigma klijent-poslužitelj pruža mogućnost da klijenti postanu poslužitelji i obrnuto. To je samo jedan od oblika kompleksnije interakcije klijenata i poslužitelja. Kao primjer, promotrit ćemo aplikaciju *Domain Name System* (DNS).

U toj aplikaciji, DNS poslužitelj prevodi simbolička imena računala u IP adrese. Baza podataka koja sadrži veze IP adresa i simboličkih imena se ne nalazi na jednom računalu, već je distribuirana između mnogo DNS poslužitelja. Struktura simboličkih imena računala je strogo hijerarhijska, s najvažnijim dijelom imena na krajnjem desnom kraju. Na primjer:

wallnut.candy.foobar.com

Navedeno ime pripada na primjer hijerarhiji koja je prikazana na Slici 21.4. Ta hijerarhija mogla bi odgovarati nekoj korporaciji, a njeni dijelovi podružnicama. Čvorovi u hijerarhiji na najnižoj razini obično odgovaraju konkretnim računalima unutar podružnica ili odjela.



Slika 21.4: hijerarhija simboličkih imena.

Dijelovi hijerarhije zovu se *domene*. Domene na vrhu hijerarhije zovu se *vršne* (top-level) domene, i one su pod kontrolom ustanove koja se zove *Internet Corporation for Assigned Names* (ICANN). Domene odmah ispod vršne kontrolira ustanova koju je ovlastio ICANN. Najvažnije vršne domene popisane su u tablici na Slici 21.5.

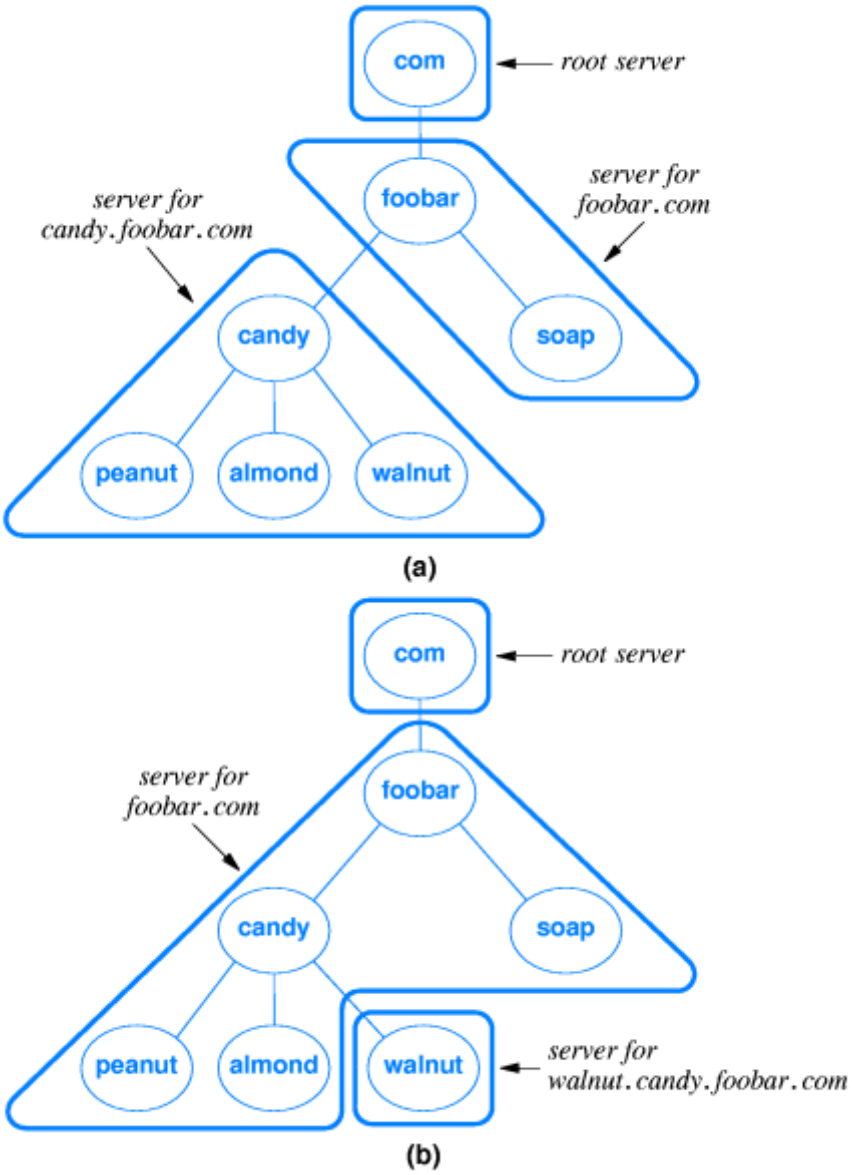
Domain Name	*Assigned To*
com	*Commercial organization*
edu	*Educational institution*
gov	*Government organization*
mil	*Military group*
net	*Major network support center*
org	*Organization other than those above*
arpa	*Temporary ARPA domain (still used)*
int	*International organization*
country code	*A country*

Slika 21.5: vršne domene.

DNS poslužitelji su također organizirani hijerarhijski. Hijerarhija poslužitelja prati hijerarhiju imena domena. Slika 21.6 prikazuje dva načina da se ista hijerarhija imena podijeli na tri poslužitelja. Pritom jedan DNS poslužitelj mora biti odgovoran za sva imena s danim sufiksom.

Prevođenje simboličkog imena u IP adresu se naziva *name resolution*. Realizirano je UNIX programom gethostbyname. Da bi preveo neko ime, program postavlja pitanje svom lokalnom DNS poslužitelju. Svi DNS poslužitelji znaju kako se povezati s *vršnim poslužiteljem* (root serverom) i kako se povezati s poslužiteljima koji su odgovorni za pod-domene koje su niže u

hijerarhiji. Kad DNS poslužitelj dobije zahtjev za prevođenjem simboličkog imena, on ga potraži u lokalnoj bazi podataka. U slučaju da ga ne može pronaći, poslužitelj postaje klijent vršnog poslužitelja ili poslužitelja koji je niže u hijerahiji DNS-a.

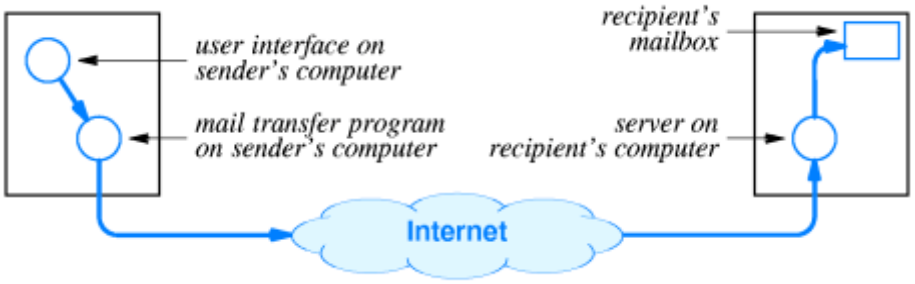


Slika 21.6: dva načina podjele iste hijerarhije imena domena na tri DNS poslužitelja.

Treći primjer aplikacije tipa klijent-poslužitelj: e-mail

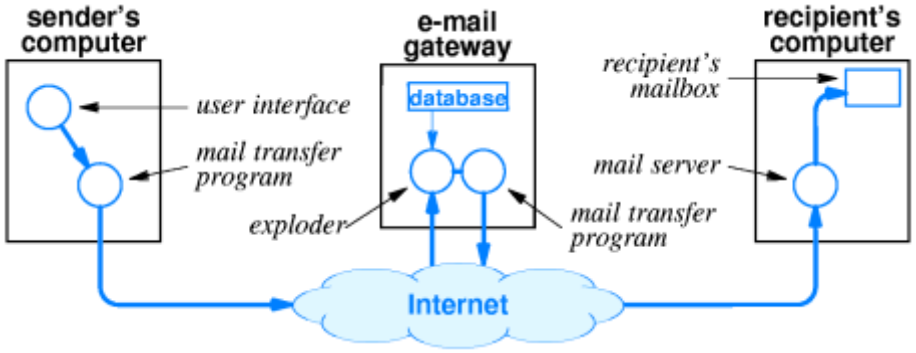
Elektronička pošta (e-mail) je usluga koja je nastala kao replika tradicionalnog sistema uredske pošte. Riječ je o prenošenju tekstualnih poruka. Elektronski poštanski sandučić se identificira s dva parametra: ime sandučića, ime računala. Dakle adresa sandučića je oblika mailbox@computername. Komunikacije između e-mail poslužitelja i e-mail klijenta se odvija preko *SMTP* protokola (Simple Mail Transfer Protocol). Internet e-mail može prenositi samo tekst. Za prenošenje binarnih podataka (slike, glazba,...) kasnije je uveden *Mutipurpose*

Internet Mail Exstensions format (MIME). Taj format omogućava primatelju i pošiljatelju biranje standarda za kodiranje (encoding) binarnih podataka. Arhitektura e-mail-a u najjednostavnijem obliku vidljiva je na Slici 21.7.

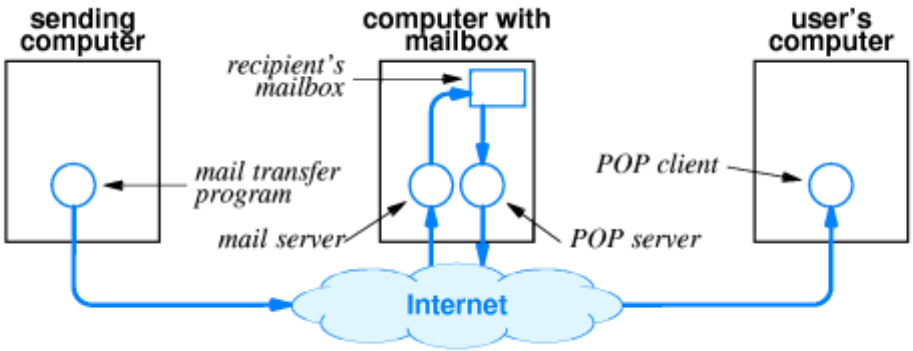


Slika 21.7: osnovna arhitektura e-mail usluge.

Programi osim procesiranja elektronske pošte mogu i samostalno slati elektronsku poštu. Primjer su *mailing-liste* i programi *mail-exploder*-i koji prosljeđuju poruku na listu e-mail adresa koje se nalaze u bazi podataka. Sam mail-exploder odnosno pripadna lista e-mail adresa identificira se zasebnom e-mail adresom. Način rada s mailing listom prikazan je na Slici 21.8. Mail-exploder obično radi na posebnom računalu koje se zove *e-mail gateway*.



Slika 21.8: rad s mailing-listom.



Slika 21.9: proširena arhitektura e-mail-a, korištenje POP protokola.

U današnje vrijeme, korisnici rade s elektroničkom poštom sa svog osobnog računala. To je dodatno zakompliciralo arhitekturu e-maila. Uveden je *Post Office Protokol* (POP), koji omogućava da mailbox i e-mail klijent ne moraju biti na istom računalu. Takva proširena arhitektura e-maila vidi se na Slici 21.9.

Četvrti primjer aplikacije tipa klijent-poslužitelj: WWW

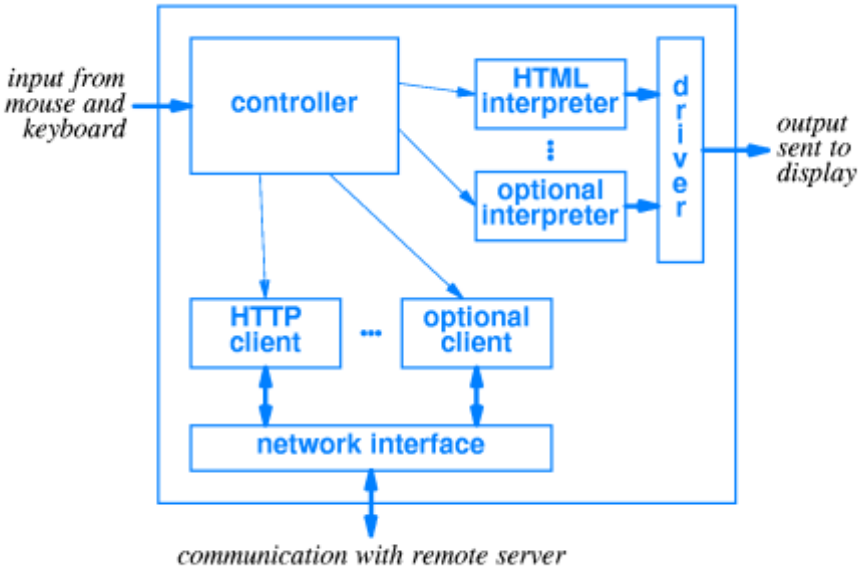
World wide web (WWW ili web) je veliki repozitorij dokumenata. Web dokumenti se zapisuju i spremaju u jeziku koji se zove *HyperText Markup Language* (HTML). *Web preglednik* (browser) je interaktivni aplikacijski program koja služi za pristup web dokumentima i za njihov pregled. Pritom zapis dokumenta u HTML-u samo ugrubo specificira smjernice za prikazivanje dokumenta, te dopušta pregledniku određivanje detalja. Svaki web dokument identificira se svojom *Universal Resource Location* (URL) adresom, koja se zapisuje u formatu

protokol://ime_računala/ime_dokumenta

Pregledavanje weba odvija se u potpunosti u skladu sa paradigmom klijent-poslužitelj. Naime, web preglednik je klijent koji traži od odgovarajućeg web poslužitelja da mu pošalje web dokument. Taj web poslužitelj je program koji radi na onom računalu gdje je pohranjen web dokument. Za komunikaciju web klijenta i web poslužitelja koristi se TCP veza i posebni aplikacijski protokol koji se zove *HyperText Transfer Protocol* (HTTP). Ukoliko web stranica sadrži mutimedijske objekte i tekst, za svaki objekt se otvara nova TCP/IP veza.

HTTP protokol je relativno jednostavan. Sastoji se od svega nekoliko komandi.

- GET: zahtjeva određeni podatak od poslužitelja.
- HEAD: traži informaciju o statusu dokumenta
- PUT: šalje podatke poslužitelju, koje poslužitelj koristi da bi zamjenio određeni dokument.
- POST: šalje podatke poslužitelju, koje poslužitelj dodaje danom dokumentu.



Slika 21.10: komponente web preglednika.

Osim HTTP postoje i drugi protokoli za komunikaciju web klijenata i poslužitelja, na primjer HTTPS koji se koristi za kriptiranu komunikaciju. Također, osim HTML postoje i drugi jezici za zapisivanje dokumenata - takozvani *markup* jezici - na primjer XML, VXML. Sve su to razlozi zašto današnji web preglednici imaju prilično složenu građu. Slika 21.10 prikazuje glavne komponente web preglednika.

Opis WWW iz prethodnih odjeljaka zapravo se odnosio na polaznu i osnovnu verziju te tehnologije. U posljednjih 15-tak godina WWW se intenzivno razvijao te su se u njega uklopile brojne dodatne tehnologije. Zahvaljujući takvom razvoju, današnji web dokumenti ne moraju isključivo biti tekstovi u HTML-u ili nekom sličnom markup jeziku, već također mogu poprimiti složenije oblike. Točnije, današnji web dokumenti dijele se na sljedeća tri tipa.

- *Statički*. To su klasični web dokumenti zapisani na primjer u HTML-u, koji se nalaze u jednoj datoteci i prilikom svakog pozivanja izgledaju isto.
- *Dinamički*. Web poslužitelj stvara dinamički web dokument prilikom svakog poziva. Dokument se realizira izvršavanjem nekog dodatnog programa na strani poslužitelja čiji ispis poslužitelj šalje klijentu. Koriste se tehnologije poput Common Gateway Interface (CGI), ASP, JSP,PHP, ...
- *Aktivni*. Sastoji se od programa kojeg web poslužitelj pri svakom pozivu šalje pregledniku za lokalno izvršavanje (na strani preglednika). Koriste se tehnologije poput Java, JavaScript, …

Za one koji žele znati više

Tehnologije vezane uz WWW dalje će se proučavati u kolegiju "Računarski praktikum 2" na diplomskom studiju Računarstvo i matematika. Ciljevi tog kolegija bit će:

- Povezivanje znanja o programiranju, bazama podataka i mrežama računala.
- Ovladavanje tehnologijama koje omogućuju razvoj web aplikacija.

Obradit će se alati i tehnologije koje služe za realizaciju dinamičkih odnosno aktivnih web dokumenata, na primjer:

- CGI,
- PHP,
- JavaScript,
- Java.

Sažetak Poglavlja 21

U ovom poglavlju smo opisali standardne mrežne aplikacije kao prototipove različitih distribuiranih programskih arhitektura koje se mogu realizirati korištenjem standardnih mrežnih komunikacijskih protokola. Distribuirano računarstvo je oblik paralelnog programiranja koji se bavi hardverskim i softverskim sustavima gdje se više istovremenih programa ili procesa odvijaju u manje ili više strogom režimu upravljanja. U distribuiranom računarstvu se aplikacija dijeli u više programa koji se istovremeno izvršavaju na skupu računala koji mogu komunicirati preko neke mrežne tehnologije. Efikasni distribuirani programi moraju uspješno rješavati problem kašnjenja i grešaka u mrežnoj komunikaciji. U središtu pažnje bila nam je arhitektura mrežnih aplikacija zasnovana na klijentima i poslužiteljima. Neki daljnji primjeri distribuiranih programskih arhitektura su: klijent 3-tier architecture, N-tier architecture, loose coupling, tight coupling, peer-to-peer, distribuirani objektni modeli i slično. Više o problemima komunikacije među procesima možete naći sljedećem poglavlju i na http://en.wikipedia.org/wiki/Distributed_computing.

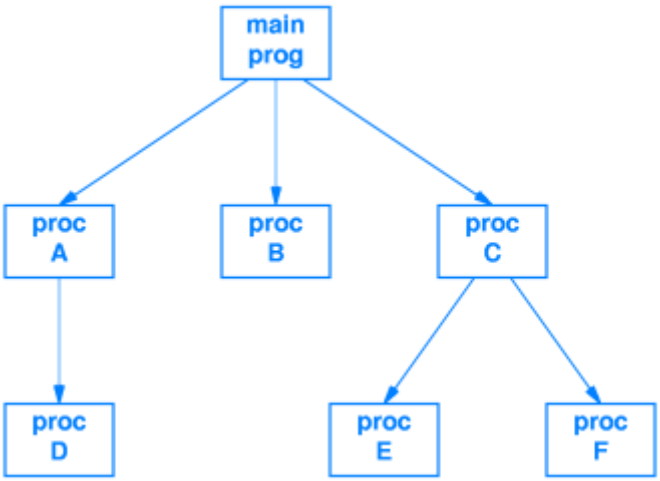
22. Povezivanje udaljenih procedura – RPC i middleware

Sadržaj Poglavlja 22

U ovom poglavlju bavimo se metodama koje služe programerima za razvoj mrežnih aplikacija. Govorimo o paradigmi poziva udaljenih procedura – RPC. Također opisujemo suvremenu inačicu iste paradigme koja je prilagođena objektnom programiranju i zove se pokretanje udaljenih metoda – RMI. Nabrajamo poznate primjere takozvanog middleware-a, dakle primjere softverskih alata za realizaciju RPC odnosno RMI.

Potreba za paradigmom RPC

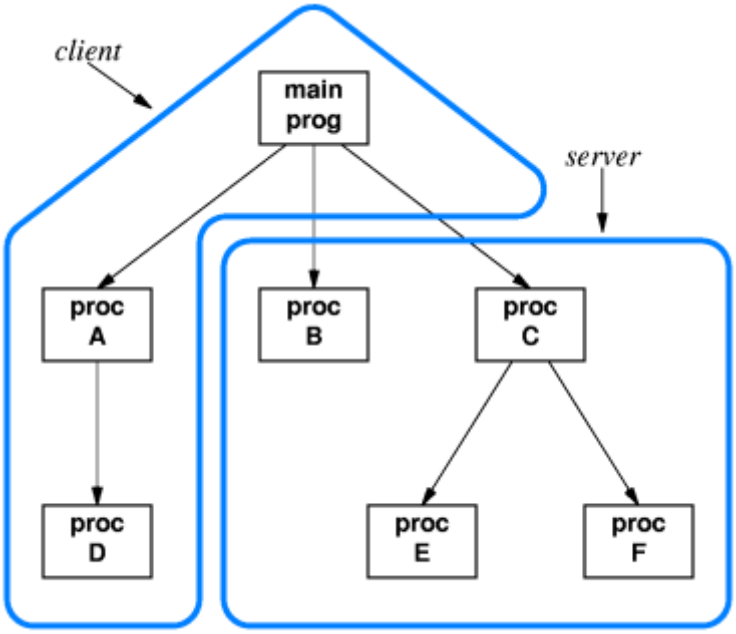
Većina programera naučila je pisati konvencionalne programe koji rade na jednom računalu i sastoje se od glavnog programa i procedura (potprograma). Građa takvog konvencionalnog programa prikazana je na Slici 22.1. S druge strane, programi predviđeni za rad na mreži bitno su kompliciraniji budući da osim uobičajenih elemenata također moraju uključiti i komunikaciju s drugim programima.



Slika 22.1: građa konvencionalnog programa koji radi na jednom računalu.

Da bi se većini programera olakšao razvoj mrežnih aplikacija, u 1980-tim godinama nastala je paradigma *poziva udaljenih procedura* (Remote Procedure Call – RPC). Osnovna ideja RPC paradigme je sakrivanje eksplicitne mrežne komunikacije korištenjem uobičajenog mehanizma pozivanja procedura i prosljeđivanja parametara.

RPC tehnologija dozvoljava da se dijelovi konvencionalnog programa naknadno rasporede na dva ili više računala. Programer se zato ne mora puno opterećivati mrežom i komunikacijskim protokolima. Umjesto toga, on se može koncentrirati na sam problem kojeg njegov softver treba riješiti.



Slika 22.2: program sa prethodne slike podijeljen na klijentski i poslužiteljski dio.

Razvoj aplikacija pomoću RPC

Postupak razvoja aplikacije u skladu s paradigmom RPC teče otprilike ovako.

- Programer najprije razvija konvencionalni program koji radi na jednom računalu.
- Nakon toga, programer dijeli program u dva dijela, na primjer onako kao što je prikazano na Slici 22.2. Dio s glavnim programom postat će klijent. Preostali dio postat će poslužitelj. Podjela uzima u obzir eventualne globalne podatke, dakle oni se smještaju u onaj dio koji ih koristi.
- Dalje programer stvara specifikaciju koja opisuje način podjele programa, odnosno sučelje udaljenih procedura. Specifikacija se piše u *jeziku za definiranje sučelja* (Interface Definition Language – IDL).
- Odgovarajući RPC alat čita IDL specifikaciju i automatski generira programski kod koji omogućuje dijeljenje programa i prebacivanje toka kontrole s jednog računala na drugo.
- Programer na kraju prevodi i povezuje vlastite programske jedinice s dijelovima koje je automatski generirao RPC alat, te tako dobiva klijentski i poslužiteljski program.

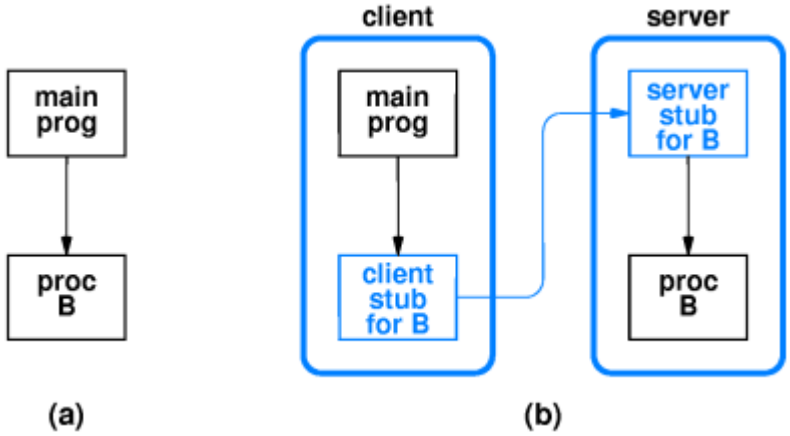
Vidimo da je programeru za razvoj aplikacije na opisani način potreban posebni softverski alat koji podržava proces razvoja u skladu s RPC. Danas postoje brojni komercijalni RPC alati. Oni se jednim imenom nazivaju *middleware* budući da stoje "između" aplikacije i mrežnog softvera.

Komunikacijski umetci

Vidimo da u programu razvijenoj po paradigmi RPC tok kontrole skače iz jednog dijela programa u drugi. Naravno, tok kontrole ne može zaista preskočiti iz programa na jednom računalu u proceduru na drugom računalu. Taj privid zapravo se postiže interakcijom između

klijenta i poslužitelja. Preciznije, riječ je o komunikaciji između dijelova softvera koje je automatski generirao RPC alat, i koji se zovu *komunikacijski umetci* (communication stubs).

Promotrimo na Slici 22.3 (a) najjednostavniji slučaj glavnog programa koji poziva jednu proceduru B. Tu proceduru želimo prebaciti na drugo računalo. Nakon ugradnje komunikacijskih umetaka, klijent i poslužitelj izgledaju kao na Slici 22.3 (b).



Slika 22.3: (a) konvencionalni poziv procedure; (b) realizacija istog poziva pomoću umetaka.

Rad komunikacijskih umetaka sa Slike 22.3 odvija se na sljedeći način.

- Klijentov umetak s jedne strane komunicira s poslužiteljem, a s druge strane simulira proceduru B (ima isto ime i parametre). Glavni program "misli" da i dalje poziva proceduru B.
- Poslužiteljev umetak s jedne strane komunicira s klijentom, a s druge strane simulira glavni program. Procedura B "misli" da nju i dalje poziva glavni program.

Vanjski prikaz podataka

Osim prijenosa kontrole, pozivanje procedure uključuje i prijenos podataka.

- Najprije se iz glavnog programa parametri prenašaju u potprogram.
- Zatim se iz potprograma izračunati rezultati prenašaju u glavni program.

Kod konvencionalnog poziva procedure unutar jednog računala podaci se mogu slobodno prenašati u internom (binarnom) formatu, dakle onako kako su zaista prikazani u memoriji računala. No kod poziva udaljene procedure stvari se kompliciraju. Naime, podaci se moraju prebacivati preko mreže iz jednog računala u drugo. Ako bi se oni i dalje prenašali u internom formatu, to bi moglo stvoriti probleme jer razna računala koriste različite interne formate. Na primjer, neka računala spremaju najznačajniji oktet 32-bitnog cijelog broja na najnižu adresu, a neka na najvišu.

Većina RPC alata rješava spomenuti problem uvođenjem "vanjskog" (usklađenog) prikaza standardnih tipova podataka. Primjenjuje se sljedeći postupak konvertiranja.

- Prije slanja, pošiljateljev komunikacijski umetak konvertira podatak iz pošiljateljevog internog prikaza u vanjski prikaz.

- Nakon primanja, primateljev komunikacijski umetak konvertira podatak iz vanjskog prikaza u primateljev interni prikaz.

Stariji alati za RPC

Alati za RPC razvijaju se od 1980-tih godina do danas. Sljedeći popis sadrži nekoliko alata koji su odigrali važnu ulogu u razvoju same paradigme, makar se danas više ne koriste.

- *SUN RPC.* Službeno ime mu je Open Network Computing Remote Procedure Call – ONC RPC. Razvila ga je kompanija SUN Microsystems u 1980-tim godinama. To je prvi RPC alat koji je stekao širu popularnost. Sadrži svoj IDL, protokol za komunikaciju i standard za vanjski prikaz podataka.
- *DCE/RPC.* Razvila ga je organizacija Open Software Foundation u ranim 1990-tim godinama, kao dio sveobuhvatne okoline Distributed Computing Environment – DCE. Definira svoj IDL. Dozvoljava klijentu pristup do više poslužitelja.
- *MS-RPC.* Razvio ga je Microsoft u 1990-tim na temelju DCE/RPC. Makar je po koncepciji sličan, u mnogim detaljima se razlikuje od DCE/RPC.

Objektno programiranje i distribuirani objekti

U 1990-tim godinama proširili su se objektno-orijentirani programski jezici poput C++, Java, C# i drugih, koji su omogućili razvoj objektnih programa. Za razliku od tradicionalnog programa koji se sastoji od procedura i globalnih podataka, objektni program građen je kao skup objekata. Svaki objekt je zasebna programska cjelina koja se sastoji od vlastitih podataka, te vlastitih operacija - takozvanih metoda - koje djeluju nad tim podacima. Umjesto poziva procedura, u objektnom programu se kao osnovni kontrolni mehanizam pojavljuje pokretanje metoda (method invocation).

Kao posljedica promjena u programskim jezicima, u kasnim 1990-tim godinama došlo je do promjena u RPC paradigmi. Umjesto o pozivu udaljenih procedura, danas se govori o pokretanju metoda u udaljenim objektima. Zbog toga se suvremena inačica RPC obično naziva *udaljeno pokretanje metoda* (remote method invocation – RMI). Arhitektura klijent-poslužitelj zamjenjuje se nešto općenitijom *arhitekturom distribuiranih objekata*, gdje pojedini objekt može igrati ulogu i klijenta i poslužitelja. Odgovarajući objektno-orijentirani middleware trebao bi omogućiti transparentno raspoređivanje objekata koji čine program po računalima, te transparentnu međusobnu interakciju takvih distribuiranih objekata.

Objektno-orijentirani middleware

Danas postoji velik alata za razvoj distribuiranih objektnih aplikacija u skladu s paradigmom RMI. Nabrojat ćemo nekoliko najpoznatijih i najznačajnijih primjera.

- *CORBA.* Kratica znači: Common Object Request Broker Architecture. Riječ je o skupu standarda koje je definirao konzorcij Object Management Group – OMG – krajem 20. stoljeća. Postoje implementacije za UNIX, Linux i MS Windows. Slijedi spomenute ideje o distribuiranim objektima, s time da se komunikacijski umetci stvaraju dinamički tijekom rada programa.
- *DCOM.* Kratica znači: Distributed Component Object Model. To je standard razvijen i implementiran od Microsofta 1998. godine, te integriran u operacijske sustave MS Windows. Model distribuiranog računanja manje je općenit od CORBA-inog te je ograničen na Microsoftove platforme.

- *.NET Remoting.* Noviji Microsoftov produkt, objavljen 2002. godine. Predstavlja dio cjelovite .NET arhitekture. Predviđen je kao zamjena za stariji DCOM. Opet je riječ o "proprietary" tehnologiji koja je ograničena na Microsoftove platforme.
- *Java RMI.* Razvila ga je kompanija SUN Microsystems početkom 21. stoljeća. Kratica znači: Java Remote Method Invocation. Služi kao proširenje programskog jezika Java kojim se postiže pokretanje udaljenih metoda.

Za one koji žele znati više

Distribuirani objekti vrlo su zanimljiva i aktualna tema koja će se znatno detaljnije obraditi u kolegiju "Distribuirani procesi" na diplomskom studiju Računarstvo i matematika. U istom kolegiju govorit će se općenitije o distribuiranim procesima i algoritmima, o klasičnim problemima koji nastaju zbog distribuiranog načina rada, te načinima rješavanja tih problema. Obradit će se bar jedan objektno-orijentirani middleware poput CORBA, Java RMI, DCOM, ili .NET Remoting.

Sažetak Poglavlja 22

Većina programera naučila je pisati konvencionalne programe koji rade na jednom računalu. Da bi se takvim programerima olakšao razvoj mrežnih aplikacija, nastala je paradigma RPC. Osnovna ideja RPC je skrivanje eksplicitne mrežne komunikacije podržavanjem predodžbe o mogućnosti pozivanja „udaljenih" procedura. Suvremena objektno-orijentirana inačica RPC zove se RMI i ona podržava predodžbu o mogućnosti pozivanja „udaljenih" metoda. Primjenom klasične paradigme RPC moguće je stvoriti aplikacije tipa klijent-poslužitelj, dok RMI omogućuje aplikacije s nešto općenitijom arhitekturom distribuiranih objekata. Da bi razvio svoju mrežnu aplikaciju u skladu s RPC odnosno RMI, programer mora koristiti odgovarajući alat – takozvani middleware. Najznačajniji primjeri objektno-orijentiranog middleware-a su: CORBA, MS .NET Remoting i SUN Java RMI.

23. Multimedija na Internetu

Sadržaj Poglavlja 23

U ovom poglavlju govorimo o multimedijskim mrežnim aplikacijama. Raspravljamo o podobnosti Interneta za multimediju. Budući da većina multimedijskih aplikacija koristi sažimanje podataka, usput spominjemo i algoritme za sažimanje. Kao primjer zahtjevnijeg oblika multimedije na Internetu, detaljno proučavamo IP telefoniju.

Osobine multimedijskih mrežnih aplikacija

Multimedijske mrežne aplikacije zasnivaju se na prenošenju "multimedijskih" sadržaja:
- digitaliziranih slika,
- animacija,
- zvučnih zapisa,
- video zapisa.

U većoj ili manjoj mjeri te aplikacije su interaktivne. Također, one mogu zahtijevati izvođenje u realnom vremenu. Postavljaju relativno velike zahtjeve u pogledu performansi mreže. Naime:

- sve zahtijevaju veliku propusnost;
- neke od njih traže malo kašnjenje ili malu varijaciju kašnjenja.

Vrste multimedijskih mrežnih aplikacija

Postoje sljedeće vrste multimedijskih mrežnih aplikacija, sa sljedećim zahtjevima na performanse.

- *Prenošenje slika*, na primjer pregledavanje web albuma. Zahtijeva se zadovoljavajuća propusnost, može se tolerirati kašnjenje.
- *Prenošenje zvučnog ili video zapisa*, na primjer slušanje radio programa ili video-on-demand. Zahtijeva se nešto veća propusnost, može se donekle tolerirati kašnjenje, no traži se mala varijacija kašnjenja.
- *Interaktivna razmjena zvuka ili slike*, na primjer telefoniranje preko Interneta (voice over IP), ili tele-konferencija. Zahtijeva se velika propusnost i zanemarivo kašnjenje.

Podobnost Interneta za multimediju

Internet zapravo *nije podoban* za izvođenje multimedijskih aplikacija, jer on ne pruža nikakvu garanciju o kakvoći usluge. Na primjer, nema garancije da će se telefonski razgovor preko Interneta uspješno odvijati unatoč kašnjenjima. Potrebni kapaciteti mreže ne mogu se rezervirati.

Unatoč nepogodnostima, multimedija na Internetu je ipak postala moguća zbog dva razloga.

- Koristi se mrežna infrastruktura s pretjerano velikim kapacitetima.
- Razvijeni su složeni transportni i aplikacijski protokoli koji donekle kompenziraju povremena zagušenja i varijacije kašnjenja.

Sažimanje podataka

Da bi se smanjili zahtjevi za prostorom na disku i za propusnošću mreže, multimedijski sadržaji se redovito pohranjuju i prenašaju u sažetom (komprimiranom) obliku. U slučaju prijenosa podataka, postupak koju uključuje sažimanje izgleda ovako.

- Prije slanja podataka, algoritam za sažimanje (kompresiju) pretvara originalne podatke u sažeti oblik.
- Nakon primanja podataka, inverzni algoritam za dekompresiju vraća sažete podatke natrag u njihov originalni oblik.

Da bi sažimanje bilo učinkovito, u originalnim podacima mora postojati neki oblik redundancije ili pravilnosti. Taj uvjet je uvijek ispunjen kod multimedije, jer na primjer tipična fotografija sadrži mnoštvo sličnih piksela, dvije uzastopne sličice u nizu koji čini video zapis neznatno se razlikuju, i tako dalje.

Postoje dvije vrste algoritama za sažimanje:

- *Algoritmi bez gubitaka* (lossless). Podaci nakon dekompresije identični su originalnim podacima.
- *Algoritmi s gubitkom* (lossy). Podaci nakon dekompresije neznatno se razlikuju od originalnih podataka.

Primjer algoritma bez gubitaka je *LZW* (Lempel-Ziv-Welch) koji se koristi u GIF formatu za slike. Primjer algoritma s gubitkom je *JPEG* za sažimanje fotografija, odnosno *MPEG* za video, odnosno *MP3* za audio.

IP telefonija – Voice over IP (VoIP)

Riječ je o zahtjevnijem primjeru multimedijske mrežne aplikacije, koji omogućuje obavljanje telefonskih razgovora preko Interneta. Prijenos zvuka od jednog do drugog sugovornika ostvaruje se sljedećim nizom koraka.

- Analogni audio signal najprije se na polazištu digitalizira.
- Zatim se taj digitalizirani signal dijeli u pakete koji se dalje šalju kroz IP mrežu prema odredištu.
- Na odredištu se paketi ponovo sastavljaju u digitalizirani signal.
- Zatim se taj signal vraća u analogni oblik da bi se mogao reproducirati.

Za jedan telefonski razgovor potrebna su dva ovakva paralelna transfera podataka u suprotnim smjerovima.

Makar je osnovna ideja IP telefonije jednostavna, mnogi „detalji“ kompliciraju njenu realizaciju. Navodimo neke od tih detalja.

- Ne smije biti kašnjenja sa slanjem i prenošenjem podataka jer bi to stvorilo “zastajkivanje” u razgovoru.
- Osim prijenosa glasa, sustav mora obavljati i uspostavu telefonskog razgovora.
- Kod početka poziva, pozvana stranka mora prihvatiti poziv i odgovoriti na njega.
- Kad razgovor završi, obje strane moraju se složiti o načinu kako da zaključe komunikaciju.
- Mora se osigurati interoperabilnost s tradicionalnim telefonskim mrežama.

Standardi za IP telefoniju

Da bi se hardver i softver različitih proizvođača mogao uključiti u zajednički IP telefonski sustav, nužno je da svi proizvođači poštuju zajedničke standarde. Postoje dvije grupe koje su stvorile takve standarde:

- *International Telecommunications Union* (ITU). Međunarodna organizacija pod okriljem UN koja postavlja standarde za obične telefone.
- *Internet Engineering Task Force* (IETF). Međunarodna udruga koja kontrolira TCP/IP standarde.

<table>
<tr><th>Layer</th><th>Call Process.</th><th>User Audio or Video</th><th>User Data</th><th>Support</th><th>Routing</th><th>Signal Transport</th></tr>
<tr><td>5</td><td>H.323
Megaco
MGCP
SIP</td><td>RTP</td><td>T.120</td><td>RTCP
RTSP
NTP
SDP</td><td>ENUM
TRIP</td><td>SIGTRAN†</td></tr>
<tr><td>4</td><td>TCP
UDP</td><td>UDP</td><td>TCP</td><td colspan="2">TCP
UDP</td><td>SCTP</td></tr>
<tr><td>3</td><td colspan="6">IP, RSVP, and IGMP</td></tr>
</table>

Slika 23.1: skup protokola za IP telefoniju.

Skup svih protokola koje su obje grupe predložile prikazan je u tablici na Slici 23.1. Protokoli su razvrstani s obzirom na svrhu kojoj služe i s obzirom na sloj u TCP/IP referentnom modelu u kojem djeluju. Prijedlozi dviju grupa nisu sasvim usklađeni, no ipak se slažu u dijelu koji se tiče kodiranja, prijenosa i reprodukcije audio signala. Razlike u prijedlozima dviju grupa najviše se osjećaju u dijelu koji se tiče takozvane signalizacije, dakle sustava i protokola za uspostavljanje poziva i upravljanje pozivom.

Kodiranje, prijenos, reprodukcija

Prema zajedničkom prijedlogu ITU i IETF, kodiranje i prijenos audio-signala odvija se na sljedeći način:

- Audio se kodira pomoću poznatog standarda *Pulse Code Modulation* (PCM) ili sličnog.
- Kodirani audio prenosi se kroz mrežu pomoću posebnog *Real-Time Transport Protocol*-a (RTP).

Ime protokola RTP zapravo je neispravno, jer nije riječ o transportnom nego o aplikacijskom protokolu koji radi na 5. sloju. Za transport se koristi UDP, budući da dodatni posao kojeg radi TCP nema smisla u slučaju telefonskog razgovora.

Svaka RTP poruka sadrži redni broj i podatak o realnom vremenu. RTP na strani primatelja koristi te dvije vrijednosti da bi eliminirao duplikate, poredao poruke u ispravan redoslijed i odredio vremena reproduciranja. Ako se paket izgubi, nedostajući dio audio signala reproducira se kao tišina, a reprodukcija ide dalje. Reprodukcija ne može čekati da se paket ponovo pošalje.

Sustavi i protokoli za signalizaciju

Najsloženiji dio telefonije je uspostavljanje poziva i upravljanje pozivom. U telefonskoj terminologiji to se zove *signalizacija*. Signalizacija uključuje:

- određivanje lokacije pozvane stranke na osnovu telefonskog broja;
- traženje puta kroz mrežu;
- započinjanje, prosljeđivanje, zaključivanje poziva, … .

Mehanizam koji se u tradicionalnim telefonskim sustavima koristi za ove svrhe zove se *Signaling System 7* (SS7).

Protokoli za signalizaciju u IP telefoniji su:

- *Session Initiation Protocol* (SIP) – prijedlog IETF.
- Skup protokola *H.323* – prijedlog ITU.
- *Megaco* i *MGCP* – zajednički prijedlog IETF/ITU.

Temeljni IP telefonski sustav

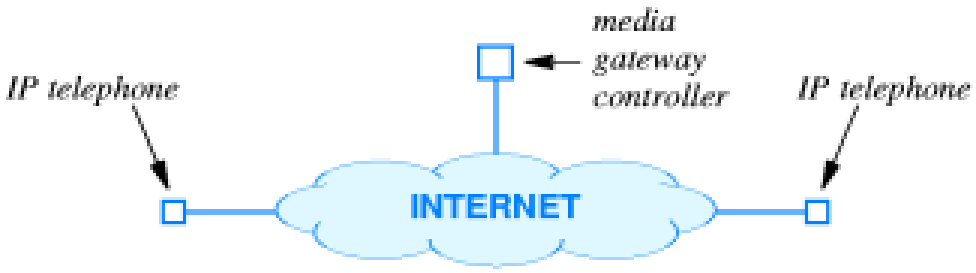
Najjednostavniji VoIP sustav sastoji se od sljedećih komponenti povezanih preko IP interneta.

- *IP telefoni* (po jedan za svakog korisnika).
- *Media Gateway Controller* (bar jedan u mreži).

Građa takvog sustava prikazana je na Slici 23.2.

IP telefon je uređaj za vođenje telefonskog razgovora, koji se umjesto na običnu telefonsku mrežu spaja na IP mrežu. IP telefon može biti zasebni uređaj sličan tradicionalnom telefonu,

ili to može biti osobno računalo s odgovarajućim softverom, zvučnicima i mikrofonom. Budući da se telefonski razgovor sastoji od slanja signala u dva smjera, IP telefon istovremeno djeluje kao RTP pošiljatelj i RTP primatelj.



Slika 23.2: organizacija najjednostavnijeg sustava za IP telefoniju.

Media Gateway Controller je uređaj (poslužitelj) koji obavlja sveobuhvatnu kontrolu i koordinaciju nad IP telefonima. Na primjer Media Gateway Controller omogućuje pozivatelju da na osnovu telefonskog broja locira pozvanu stranku, dakle odredi njenu IP adresu.

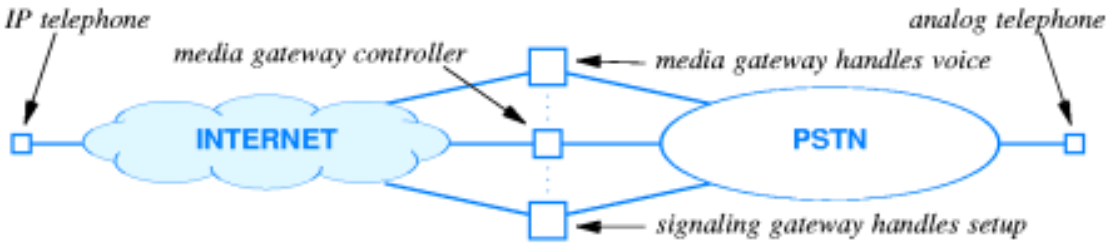
Interoperabilnost s drugim telefonskim sustavima

Sve tradicionalne telefonske mreže (fiksne i mobilne) povezane su u veliki sustav koji se zove *Public Switched Telephone Network* (PSTN). Korisnik koji ima IP telefon htio bi razgovarati s korisnikom koji ima tradicionalni telefon i obratno.

Da bi sustav IP telefonije mogao surađivati s PSTN, protokoli Megaco i MGCP predviđaju dvije dodatne komponente:

- *Media Gateway,*
- *Signaling Gateway.*

Cijela arhitektura koja povezuje IP telefonski sustav s tradicionalnim PTSN prikazana je na Slici 23.3.



Slika 23.3: povezivanje sustava IP telelefonije s tradicionalnim telefonskim sustavom.

Media Gateway obavlja prevođenje audio signala iz formata IP mreže u format PSTN. Signaling Gateway obavlja prevođenje operacija za signalizaciju. Na primjer, zahtjev za uspostavljanje poziva u skladu sa SIP prevodi se u ekvivalentni zahtjev u skladu sa SS7.
Media Gateway Controler koordinira rad Media i Signaling Gateway-a.

Za one koji žele znati više

Multimedija na Internetu detaljnije će se proučavati u kolegiju "Multimedijski sustavi" na diplomskom studiju Računarstvo i matematika. Kolegij će uključiti sljedeće teme: uvod u multimediju, stvaranje multimedijskih aplikacija, formati za prikaz multimedijskih podataka, sažimanje multimedijskih podataka,multimedijske baze podataka, multimedija i Internet. Analizirat će se matematički aspekti algoritama za sažimanje kao što LZV, JPEG, MPEG, MP3. Opširnije će se obraditi najnovije multimedijske aplikacije – VoIP i druge.

Sažetak Poglavlja 23

Internet zapravo nije podoban za multimediju, budući da on ne pruža nikakve garancije o kakvoći usluge, te se njegovi kapaciteti ne mogu rezervirati. Ipak, u posljednje vrijeme pojavile su se brojne multimedijske aplikacije na Internetu. Unatoč nepovoljnim uvjetima, te aplikacije prilično dobro funkcioniraju zahvaljujući sve bržoj i propusnijoj mrežnoj infrastrukturi, te sve boljim transportnim i aplikacijskim protokolima. Veliki zahtjevi multimedije za propusnošću mreže uspješno se smanjuju primjenom metoda za sažimanje podataka.

Dobar primjer složene multimedijske aplikacije koja unatoč svim nepogodnostima uspješno funkcionira na Internetu stječući sve veću popularnost je IP telefonija – voice-over-IP. Složenost te aplikacije dodatno se povećava uvođenjem zahtjeva za interoperabilnošću s tradicionalnim telefonskim sustavima.

24. Upravljanje mrežama – SNMP

Sadržaj Poglavlja 24

U ovom poglavlju bavimo se upravljanjem mrežama. Najprije obrazlažemo potrebu za upravljanjem, te opisujemo posao mrežnog administratora. Zatim govorimo o softveru za upravljanje mrežama, te o odgovarajućem protokolu SNMP. Primjećujemo da u skladu sa SNMP svaki mrežni uređaj ili protokol mora imati definiranu svoju bazu upravljačkih informacija – MIB. Opisujemo građu i način imenovanja objekata pohranjenih u MIB.

Potreba za upravljanjem mrežama

U radu mreža i interneta pojavljuju se problemi. Hardver se kvari, kapaciteti pojedinih veza postaju premali, dolazi do zagušenja i gubitka podataka. Mrežni hardver i softver sadrže mehanizme koji automatski otkrivaju greške i ponovo šalju pakete. Ipak, probleme treba otkrivati i rješavati na vrijeme jer u protivnom dolazi do prevelike degradacije performansi mreže.

Osoba zadužena za upravljanje mrežom naziva se *mrežni administrator* (network manager). Posao administratora je otkrivanje i rješavanje problema funkcioniranja mreže, te otklanjanje situacija u kojima bi opet moglo doći do istih problema. Upravljanje mrežom može biti težak posao zbog dva razloga:

- Interneti su obično heterogeni jer sadrže hardverske i softverske komponente od različitih proizvođača.

- Neki dijelovi interneta obično su fizički udaljeni tako da ih se mora posredno nadzirati "na daljinu".

Softver za upravljanje mrežama

Da bi mrežni administrator mogao efikasno obavljati svoj posao, potreban mu je odgovarajući *softver za upravljanje mrežama*. Glavne funkcije takvog softvera su sljedeće.

- Softver dozvoljava administratoru da "na daljinu" ispituje uređaje poput računala, usmjernika, sklopki, pisača, te da odredi njihovo stanje ili dobije statistiku o dijelovima mreže na koje su oni spojeni.
- Softver dozvoljava administratoru da "na daljinu" upravlja takvim uređajima, na primjer da mijenja njihove tablice usmjeravanja ili da konfigurira njihova mrežna sučelja.

Upravljanje mrežom je u internetu implementirano na najvišem, dakle aplikacijskom, sloju protokola. Kad administrator treba stupiti u interakciju s određenim hardverskim uređajem, on pokreće odgovarajući aplikacijski program koji se ponaša kao klijent. Na samom hardverskom uređaju radi drugi aplikacijski program koji se ponaša kao poslužitelj. Klijent i poslužitelj koriste uobičajene transportne protokole kao što su TCP ili UDP.

Da bi se naglasila razlika između aplikacija za "obične" korisnike i onih za mrežne administratore, kod sustava za upravljanje mrežama izbjegavaju se termini "klijent" i poslužitelj". Aplikacijski program na administratorovom računalu naziva se *manager*, a aplikacijski program na mrežnom računalu zove se *agent*.

Korištenje obične mrežne infrastrukture za upravljanje tom istom mrežom može izgledati čudno. Naime, greške u mreži koja je predmet upravljanja mogu spriječiti administratora da obavlja svoj posao. Korištenje obične mrežne infrastrukture u praksi ipak radi dobro iz sljedećih razloga.

- Kad hardverska greška spriječi komunikaciju s jednim uređajem, administrator može pokušati komunicirati sa susjednim uređajima, te metodom pokušaja i pogreške locirati problem.
- Kad dođe do zastoja u mrežnom prometu, administrator to odmah primijeti jer se zastoj vidi na njegovim paketima.

Neki administratori instaliraju i posebni hardver da bi neovisno o mreži mogli administrirati važne uređaje – na primjer dial-up modem spojen u usmjernik.

Protokol za upravljanje – SNMP

Standardni protokol za upravljanje internetom zove se *Simple Network Management Protocol* (SNMP). Trenutna verzija je SNMPv3. SNMP definira način kako manager komunicira s agentom. Dakle, SNMP definira format i značenje managerovih zahtjeva odnosno agentovih odgovora.

SNMP koristi *paradigmu dohvaćanja i spremanja* (fetch and store paradigm). Osnovne operacije su:

- fetch za dohvaćanje vrijednosti nekog virtualnog objekta unutar nekog uređaja,
- store za spremanje vrijednosti u objekt unutar uređaja.

Objekt koji može biti dohvaćen ili spremljen ima jedinstveno ime. Naredba fetch ili store sadrži ime objekta.

Nadgledanje udaljenog uređaja postiže se dohvatom vrijednosti. Definiraju se objekti koji opisuju status uređaja. Definiraju se imena tih objekata. Da bi saznao status uređaja, administrator naredbom fetch dohvaća vrijednost odgovarajućeg objekta. Na primjer, u uređaju može biti definiran brojač okvira odbačenih zbog greške u prijenosu. Sam uređaj je napravljen tako da povećava brojač kad god se otkrije greška u prijenosu okvira. Administrator tada može pomoću SNMP dohvatiti vrijednost brojača i vidjeti da li je broj odbačenih okvira neuobičajeno velik.

Upravljanje udaljenim uređajem postiže se kao „nusprodukt“ (side-effect) spremanja vrijednosti. Definiraju se objekti koji odgovaraju pojedinim operacijama kao što su resetiranje brojača, pražnjenje među-spremnika (buffera), ponovno pokretanje uređaja (reboot) i slično. Definiraju se imena tih objekata. Da bi izvršio operaciju, administrator naredbom store "sprema" odgovarajuću vrijednost u objekt. Na primjer, u uređaju se definira apstraktni objekt koji odgovara ponovnom pokretanju uređaja. Ako administrator pomoću SNMP u taj objekt spremi vrijednost 0, agent unutar uređaja interpretirat će taj zahtjev tako da pozove proceduru ponovnog pokretanja.

Baza upravljačkih informacija – MIB

Svaki objekt do kojeg SNMP ima pristup mora biti definiran, te mora imati jedinstveno ime. Manager i agent moraju se usuglasiti u pogledu imena objekta te značenja odgovarajućih fetch i store operacija.

Skup svih objekata unutar uređaja kojima SNMP može pristupiti zove se *Baza upravljačkih informacija* (Management Information Base - MIB). SNMP zapravo ne definira MIB. Umjesto toga, SNMP standard samo definira format poruke i način kako se poruke kodiraju. Definicije MIB varijabli te značenje odgovarajućih fetch i store operacija predmet su posebnih standarda.

Imena objekata u MIB

Za imena objekata u MIB koristi se općenita hijerarhijska shema ASN.1 s dugačkim prefiksima. Osigurano je da će imena biti jedinstvena. Na primjer, brojač IP datagrama koje je uređaj primio zove se:

iso.org.dod.internet.mgmt.mib.ip.ipInReceives

Kad se ime objekta prikaže unutar SNMP poruke, svaki dio imena pretvara se u određeni cijeli broj. Na primjer, spomenuto ime brojača IP datagrama unutar SNMP poruke izgleda ovako:

1.3.6.1.2.1.4.3

Budući da SNMP ne definira unaprijed skup MIB objekata, dizajn je vrlo fleksibilan. Kad god se pojavi potreba, lako se mogu se definirati i standardizirati novi objekti.

- Na primjer, kad se pojavi novi protokol, grupa ljudi koja je stvorila protokol također definira MIB objekte za kontrolu protokolovog softvera.
- Ili kad se pojavi novi hardverski uređaj, grupa ljudi koja je razvila uređaj definira MIB objekte za nadgledanje i upravljanje uređajem.

Do danas je stvoreno mnogo takvih skupova objekata, na primjer oni koje odgovaraju protokolima UDP, TCP, IP, ARP, oni koji odgovaraju Ethernet tehnologiji, objekti za pojedine tipove usmjernika, sklopki, pisača, itd.

MIB varijable i tablice

MIB objekti najčešće su varijable koje imaju jednostavni tip poput cijelog broja. No oni mogu odgovarati i složenijim strukturama kao što su tablice. Na primjer, cijela IP tablica usmjeravanja unutar nekog usmjernika specificira se MIB objektom s imenom:

iso.org.dod.internet.mgmt.mib.ip.ipRoutingTable

Da bi se moglo pristupiti pojedinim osnovnim podacima unutar tablice, definiraju se imena tih podataka kao proširenja imena tablice. Na primjer, osnovni podatak koji određuje idući skok prema nekom odredištu naziva se

iso.org.dod.internet.mgmt.mib.ip.ipRoutingTable.ipRouteEntry.ipRouteNextHop

Primijetimo da u istoj tablici usmjeravanja postoji cijeli skup vrijednosti istog osnovnog podatka. Svaka vrijednost odgovara jednom od odredišta. Smatra se da je skup vrijednosti istog osnovnog podatka građen kao polje indeksirano s IP adresom odredišta. Da bi imenovali točno određenu vrijednost, na ime osnovnog podatka dalje lijepimo konkretnu vrijednost indeksa, dakle IP adresu odredišta. Na primjer, sljedeći skok za određeno odredište je:

iso.org.dod.internet.mgmt.mib.ip.ipRoutingTable.ipRouteEntry.ipRouteNextHop.*d*

Ovdje je *d* vrijednost 32-bitnog cijelog broja koji odgovara IP adresi tog odredišta. Kad se ovo ime unutar SNMP poruke pretvori u cjelobrojnu reprezentaciju, dobivamo:

1.3.6.1.2.1.4.21.1.7.*d*

Vidimo da ASN.1 nema pravog mehanizma za indeksiranje. Ipak, indeks pojedine vrijednosti može se proslijediti tako da ga se prilijepi uz ime objekta. Kad agent naiđe na ime koje odgovara osnovnom podatku u tablici, on izdvaja informaciju o indeksu da bi izabrao ispravni element iz skupa vrijednosti tog osnovnog podatka.

Sažetak Poglavlja 24

Da bi mreža što bolje funkcionirala, njome je potrebno upravljati. Osoba zadužena za upravljanje mrežom zove se mrežni administrator. Da bi efikasnije radio, mrežni administrator koristi posebni softver za upravljanje mrežama. Taj softver građen je u skladu s arhitekturom klijent-poslužitelj, s time da se klijent obično naziva manager, a poslužitelj agent. Komunikacija managera i agenta odvija se preko standardnog protokola SNMP. Osnovne operacije unutar SNMP su dohvaćanje odnosno spremanje vrijednosti nekog virtualnog objekta unutar nekog mrežnog uređaja ili softvera. Skup svih objekata kojima SNMP može pristupiti definira se bazom upravljačkih podataka - MIB. Za imena objekata u MIB koristi se standardizirana hijerarhijska shema ASN.1. Sami objekti mogu biti građeni kao jednostavne varijable ili kao složenije tablice.

25. Sigurnost u mrežama

Sadržaj Poglavlja 25

U ovom poglavlju govorimo o sigurnosti u mrežama, posebno u Internetu. Raspravljamo o pojmu sigurne mreže, te o sigurnosnoj politici. Uočavamo četiri važna aspekta sigurnosti: integritet, dostupnost, povjerljivost, te autentičnost podataka. Objašnjavamo kako se dostupnost može čuvati pomoću lozinki, a integritet, povjerljivost i autentičnost pomoću raznih oblika kriptiranja. Osim kriptiranja, spominjemo i druge metode koje se koriste za zaštitu na Internetu, a to su vatrozidovi i virtualne privatne mreže. Ne kraju nabrajamo neke konkretne softverske tehnologije za sigurnost koje se kao gotova rješenja ugrađuju u pojedine mreže ili Internet aplikacije.

Sigurna mreža i sigurnosna politika

Što je to *sigurna mreža*? Postoje razne definicije, na primjer:

- To je mreža koja ne dopušta osobama izvana da pristupe računalima unutar naše organizacije.
- To je mreža koja sprečava osobama izvana da mijenjaju informacije na web stranicama naše organizacije.
- To je mreža koja osigurava povjerljivost komuniciranja, na primjer da e-mail poruku ne može čitati nitko osim pošiljatelja i primatelja.

Budući da nema jednoznačne definicije sigurne mreže, svaka organizacija mora definirati svoju *sigurnosnu politiku* (*što* treba dozvoliti, što treba spriječiti). Kod definiranja sigurnosne politike potrebno je naći kompromis između sigurnosti, jednostavnosti i cijene korištenja mreže. Treba odlučiti koji aspekti sigurnosti su za dotičnu organizaciju najvažniji, a koji se eventualno mogu zanemariti.

Aspekti sigurnosti

Postoje razni aspekti sigurnosti. Nabrojat ćemo nekoliko najvažnijih.

- *Integritet podataka*. Da li primatelj zaista dobiva podatke koje je poslao pošiljatelj, ili ih je netko putem promijenio?
- *Dostupnost podataka*. Da li ovlašteni korisnici mogu doći do podataka, ili ih netko u tome ometa?
- *Povjerljivost podataka*. Da li su podaci koji putuju mrežom zaštićeni od neovlaštenog čitanja?
- *Autentičnost podataka*. Da li podaci koje je dobio primatelj zaista potječu od navedenog pošiljatelja?

U sljedećim odlomcima navest ćemo po jednu metodu za osiguranje svakog od ovih aspekata sigurnosti.

Čuvanje integriteta pomoću kriptiranja

Tehnike za zaštitu podataka od slučajnog oštećenja (na primjer kontrolni zbrojevi) ne osiguravaju integritet. Naime, ako napadač namjerno mijenja podatke koji prolaze mrežom, on će također promijeniti i kontrolni zbroj.

Stvarna zaštita od zlonamjernog mijenjanja podataka zasniva se na *kriptografskoj hash funkciji* i tajnom ključu koji je poznat samo pošiljatelju i primatelju. Postupak je sljedeći.

- Pošiljatelj na osnovu sadržaja poruke i ključa računa vrijednost H hash funkcije i šalje je uz poruku.
- Primatelj ponavlja isti račun i provjerava da li je dobio istu vrijednost H.

Napadač koji pokušava promijeniti poruku ne može na ispravan način promijeniti i H jer ne zna ključ.

Čuvanje dostupnosti pomoću lozinki

Dostupnost podataka osigurava se tako da se neovlaštenim korisnicima spriječi nepotrebno zauzimanje računalnih ili mrežnih resursa. Jedan način zaustavljanja neovlaštenih korisnika je uvođenje lozinki za pristup resursima. Ako uvedemo lozinke, onda moramo paziti da se one ne šalju po mreži u nezaštićenom obliku, pogotovo ako je riječ o bežičnoj mreži. Na primjer, ako se korisnik prijavljuje za rad na drugom računalu pomoću Telnet, tada svatko tko prisluškuje promet na mreži može doznati njegovu lozinku. Danas postoje protokoli koji prenose lozinke u kriptiranom obliku, a primjer ssh umjesto Telnet.

Čuvanje povjerljivosti pomoću kriptiranja

Zaštita od neovlaštenog čitanja podataka koji putuju mrežom postiže se kriptiranjem. Neke od tehnologija za kriptiranje zasnivaju se na tajnom ključu kojeg znaju samo pošiljatelj i primatelj. Postupak izgleda ovako.

- Pošiljatelj koristi ključ da bi stvorio kriptiranu poruku koja putuje mrežom.
- Primatelj koristi isti ključ da bi dekriptirao primljenu poruku.

Napadač koji prisluškuje komunikaciju ne zna ključ pa ne može izvući nikakvu informaciju iz kriptirane poruke.

Neka je K ključ, M poruka, a E kriptirana poruka. Cijeli postupak može se interpretirati kao primjena dviju funkcija encrypt( ) i decrypt( ):

$$E = \text{encrypt}(K, M),$$
$$M = \text{decrypt}(K, E).$$

Druga funkcija je inverz prve funkcije:

$$M = \text{decrypt}(K, \text{encrypt}(K, M)).$$

Snaga opisane zaštite zasniva se na matematičkim svojstvima funkcije za kriptiranje. Pogađanje M na osnovu E bez znanja K predstavlja zadatak koji je suviše složen u računskom smislu.

Kriptiranje javnim ključem

Novije tehnologije za kriptiranje zasnivaju se na tome da se svakom korisniku pridruže dva ključa. Prvi ključ korisnik čuva kao svoju tajnu, a drugog objavljuje zajedno sa svojim imenom i prezimenom.

Javni i tajni ključevi opet omogućuju povjerljivu komunikaciju. Naime:

- Bilo tko može pomoću javnog ključa public_u1 određenog korisnika u1 kriptirati svoju poruku, te ju poslati korisniku u1.
- Jedino korisnik u1 može pomoću svog tajnog ključa private_u1 dekriptirati poruku i saznati njen sadržaj.

Dakle sve zajedno izgleda ovako:

$$M = \text{decrypt}(\text{private\_u1}, \text{encrypt}(\text{public\_u1}, M)) .$$

Prednost korištenja javnog ključa je u tome što nema potrebe da pošiljatelj i primatelj razmjenjuju tajni ključ preko nesigurnog komunikacijskog kanala. Objavljivanje javnog ključa ne predstavlja sigurnosni rizik zahvaljujući matematičkim svojstvima korištenih funkcija. Naime, pronalaženje tajnog ključa na osnovu poznatog javnog ključa predstavlja zadatak koji je suviše složen u računskom smislu.

Osiguranje autentičnosti pomoću digitalnog potpisa

Kriptiranje s dva ključa može se koristiti i u obratnom smjeru. Poruka kriptirana pomoću tajnog ključa može se dekriptirati pomoću pripadnog javnog ključa:

$$M = \text{decrypt}(\text{public\_u1}, \text{encrypt}(\text{private\_u1}, M)) .$$

Ovaj mehanizam služi za autentikaciju poruke i naziva se *digitalni potpis*. Da bi "potpisao" poruku, pošiljatelj je kriptira pomoću svog tajnog ključa. Primatelj dekriptira poruku pomoću pošiljateljevog javnog ključa. Primatelj je siguran da poruka zaista potječe od dotičnog pošiljatelja. Naime jedino taj pošiljatelj zna tajni ključ koji odgovara upotrebljenom javnom ključu.

Istovremeno osiguranje autentičnosti i povjerljivosti

Da bi se osigurala i autentičnost i povjerljivost, postupak kriptiranja potrebno je provesti dvaput. Poruka se najprije "potpisuje" kriptiranjem pomoću pošiljateljevog tajnog ključa private_u1. Takva kriptirana poruka se ponovo kriptira pomoću primateljevog javnog ključa public_u2:

$$X = \text{encrypt}(\text{public\_u2}, \text{encrypt}(\text{private\_u1}, M)) .$$

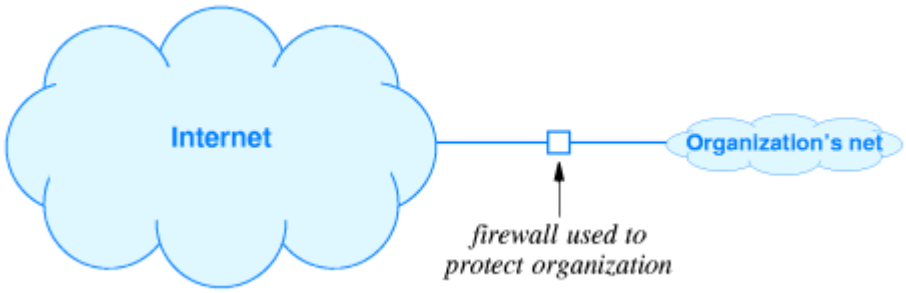
Primatelj najprije dekriptira poruku pomoću svog tajnog ključa private_u2, a zatim je još jednom dekriptira pomoću pošiljateljevog javnog ključa public_u1:

$$M = \text{decrypt}(\text{public\_u1}, \text{decrypt}(\text{private\_u2}, X)) .$$

Ako nakon ovog postupka primatelj dobije smislenu poruku, tada je sigurno da je ta poruka autentična i povjerljiva. Naime, jedino primatelj je mogao pročitati poruku jer samo on zna odgovarajući tajni ključ private_u2 potreban za uklanjanje kriptiranja s public_u2. Također, jedino pošiljatelj je mogao poslati poruku jer samo on zna tajni ključ private_u1 potreban za kriptiranje koje je uklonjivo s public_u1.

Korištenje vatrozida

Za još veći stupanj zaštite neke organizacije spojene na Internet koristi se dodatni mehanizam koji se zove *Internetski vatrozid* (Internet firewall). Vatrozid je poseban uređaj (računalo), koje se smješta između unutarnje mreže organizacije i vanjskog interneta, i koje štiti unutarnju mrežu od prometa izvana. Ideja je ilustrirana slikom 25.1.



Slika 25.1: vatrozid kao zaštita unutarnje mreže od neželjene interakcije s Internetom.

Da bi vatrozid obavljao funkciju, potrebno je da:
- Sav ulazni promet prolazi kroz vatrozid.
- Sav izlazni promet prolazi kroz vatrozid.
- Vatrozid implementira sigurnosnu politiku i odbija promet koji krši tu politiku.
- Sam vatrozid je otporan na sigurnosne napade.

Osnovna zadaća koju obavlja vatrozid je *filtriranje paketa*. Administrator konfigurira vatrozid tako da on propušta samo pakete upućene na određene IP adrese i određene TCP portove. Na primjer, može se postići da vanjski subjekti mogu pristupati samo nekim (osiguranim) računalima unutar organizacije, te da pritom smiju komunicirati samo preko određenih portova (servisa).

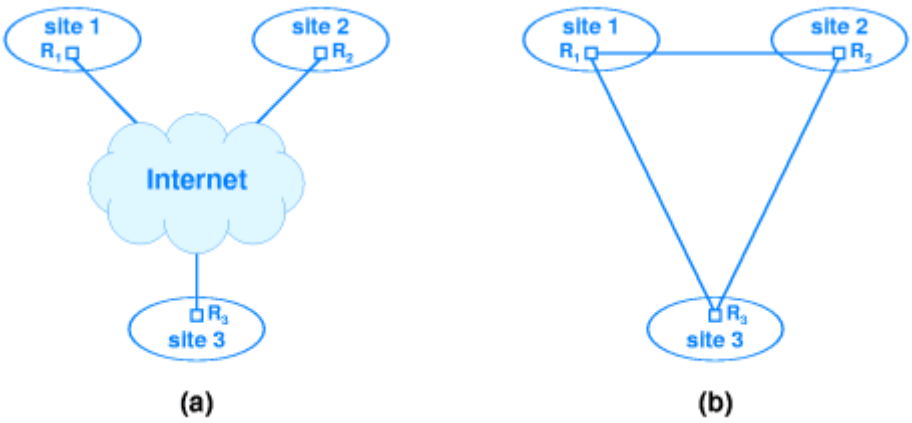
Druga zadaća koju obavlja vatrozid je pokretanje posebnih aplikacijskih programa koji se zovu *appication-layer gateways* ili *proxies*. Na primjer, može se postići da zaposlenici unutar organizacije mogu dovlačiti datoteke s Interneta jedino posredstvom FTP proxy-ja na vatrozidu. Taj proxy najprije kontrolira da li je zaposlenikov zahtjev dozvoljen u smislu sigurnosne politike, zatim on dovlači datoteku s vanjskog Interneta i provjerava da u njoj nema virusa, na kraju on šalje datoteku zaposleniku.

Virtualne privatne mreže

Zamislimo neku organizaciju koja je raspoređena na više geografskih lokacija. Da bi ta organizacija povezala svoje lokacije u jedan privatni internet (takozvani *intranet*), ona može koristiti.
- *Privatne (iznajmljene) veze* koje izravno povezuju usmjernike na dotičnim lokacijama,
- *Javne Internet veze* kojima se usmjernik na svakoj lokaciji preko lokalnog ISP-a veže na globalni Internet.

Drugo rješenje je znatno jeftinije no predstavlja sigurnosni rizik jer promet između lokacija prolazi drugim mrežama i podložan je "prisluškivanju".

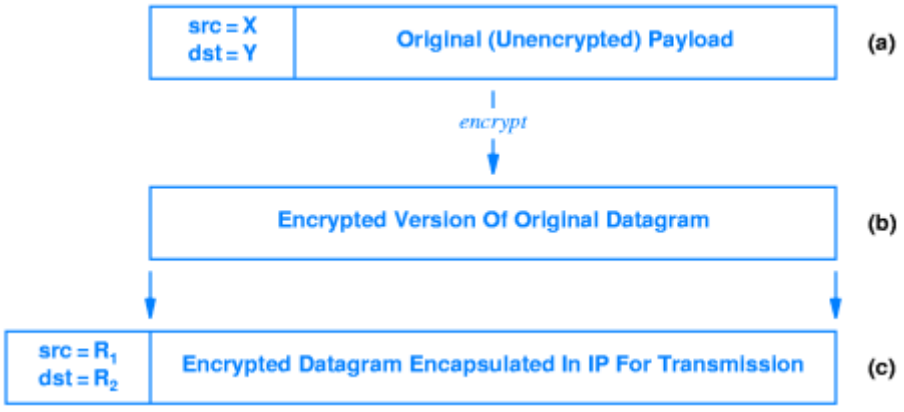
Kompromisno rješenje naziva se *virtualna privatna mreža* (Virtual Private Network – VPN). Podaci između lokacija šalju se javnim Internetom. No VPN softver u usmjernicima na lokacijama osigurava da ti usmjernici komuniciraju isključivo jedan s drugim, u skladu sa Slikom 25.2 (a). Dobiva se iluzija privatne mreže, kao što je prikazano na Slici 25.2 (b). VPN softver također obavlja *kriptiranje* i *tuneliranje*. Datagram između lokacija putuje u kriptiranom obliku. Sakriven je ne samo sadržaj nego i adrese pošiljatelja i primatelja.



Slika 25.2: (a) Internetske veze između usmjernika; (b) prividna logička struktura VPN-a.

Ako se datagram od pošiljatelja X na prvoj lokaciji preko usmjernika R1 i R2 šalje primatelju Y na drugoj lokaciji, kriptiranje i tuneliranje izgleda kao što je prikazano na Slici 25.3. Dakle:

- R1 kriptira cijeli datagram koji je krenuo od X, te ga umeće kao korisni teret u novi datagram s pošiljateljem R1 i primateljem R2.
- R2 dekriptira korisni teret iz novog datagrama i tako dobiva polazni datagram kojeg prosljeđuje Y.



Slika 25.3: slanje datagrama kroz VPN; (a) polazni datagram; (b) njegova kriptirana verzija; (c) enkapsulacija kriptiranog datagrama u novi datagram koji putuje Internetom.

Tehnologije za sigurnost

Metode za sigurnost poput kriptiranja prilično su složene tako da ih većina administratora mreža i razvijača aplikacija nije u stanju sama implementirati. Zato su se vremenom razvile neke standardne tehnologije, koje se kao gotova rješenja mogu uključiti u rad pojedinih mreža ili ugraditi u druge aplikacije. Navodimo nekoliko takvih standardnih tehnologija za sigurnost koje se intenzivno koriste na Internetu.

- *Intrusion Detection System* (IDS). Sustav koji prati sve pakete koji stižu u lokalnu mrežu i upozorava administratora ako se pojavila neka sumnjiva radnja, kao na primjer sistematsko ispitivanje TCP portova u potrazi za aktivnim poslužiteljem (TCP port scanning) ili uspostavljanje beskorisnih TCP veza u svrhu namjernog zagušenja poslužitelja (SYN flood).
- *Pretty Good Privacy* (PGP). Kriptografski sustav koji se može uključiti u razne aplikacije u svrhu kriptiranja podataka prije slanja na mrežu. Razvijen na MIT, popularan u akademskoj zajednici.
- *Secure Shell* (ssh). Aplikacijski protokol sličan Telnet-u, s time da se svi podaci između klijenta i poslužitelja prenose u kriptiranom obliku. Koristi se unutar programa Putty za sigurno prijavljivanje na udaljeno računalo.
- *Secure Socket Layer* (SSL). Softver koji se umeće između aplikacijskog programa i Socket API i koji kriptira podatke prije slanja kroz Internet. Koristi se na web stranicama koje uključuju financijske transakcije.
- *Remote Authentication Dial-In User Service* (RADIUS). Protokol koji omogućuje centraliziranu autentikaciju, autorizaciju i obračunavanje usluga za grupu korisnika. Popularno rješenje za ISP-ove koji imaju dial-up korisnike, te za VPN-ove koji dozvoljavaju zaposlenicima da se spajaju na zaštićenu mrežu od kuće.
- *Wi-Fi Protected Access* (WPA). Dio standarda za Wi-FI bežični LAN. Služi se kriptiranjem, omogućuje povjerljivost komuniciranja i autentičnost korisnika koji se spajaju na LAN.

Za one koji žele znati više

Problemi obrađeni u ovom poglavlju detaljnije će se proučavati u kolegiju "Kriptografija i sigurnost mreža" na diplomskom studiju Računarstvo i matematika. Kolegij će uključiti sljedeće teme:

- klasična kriptografija,
- moderni simetrični blokovski kriptosustavi,
- kriptosustavi s javnim ključem,
- testovi prostosti i metode faktorizacije,
- sigurnost mreža.

Definirat će se i detaljno analizirati neke od najpoznatijih metoda za kriptiranje: DES (tajni ključ), RSA (javni i tajni ključ). Objasnit će se matematički razlozi zašto su navedene metode smatraju sigurnima. Razlozi leže u teoriji brojeva, te imaju veze s računskom složenošću rastavljanja velikih brojeva na proste faktore.

Sažetak Poglavlja 25

Internet je sam po sebi nesigurna mreža. Zato svaka organizacija spojena na Internet mora definirati svoju sigurnosnu politiku, te ostvariti one aspekte sigurnosti koji su njoj najvažniji. Najčešće je riječ o sljedećim aspektima: integritet, dostupnost, povjerljivost, te autentičnost podataka.

Većina spomenutih vidova sigurnosti može se postići kriptiranjem podataka. Još veći stupanj zaštite organizacije od neželjenog utjecaja s Interneta ostvaruje se pomoću vatrozidova. Povezivanje udaljenih dijelova organizacije na ekonomičan i siguran način postiže se pomoću virtualnih privatnih mreža.

Administratori mreža te razvijači Internet aplikacija ne moraju sami implementirati složene sigurnosne postupke poput kriptiranja. Umjesto toga, njima stoje na raspolaganju brojne standardne tehnologije za sigurnost, koje se kao gotova rješenja mogu uključiti u pojedine mreže odnosno aplikacije.

26. Budućnost korištenja Interneta

Sadržaj Poglavlja 26

U ovom poglavlju govorimo o predvidivom razvoju Interneta u sljedećih nekoliko godina. Kao najvažniji problem sadašnjeg Interneta kojeg treba riješiti u bliskoj budućnosti ističemo nedostatak slobodnih IP adresa. Opisujemo kratkoročno rješenje tog problema pomoću tehnologije NAT, odnosno dugoročno rješenje koje će se postići skorim prelaskom na novu generaciju interneta IPv6. Analiziramo općenita svojstva IPv6 kao što su format datagrama, te oblik i način zapisivanja IP adresa. Diskutiramo o postupku prelaska sa sadašnje generacije interneta na novu.

Predviđanje budućnosti

Znanstvenici su se oduvijek bavili predviđanjima budućeg razvoja tehnologije. No takve vizije obično su se pokazale netočne ili nepotpune, pogotovo ako su se odnosile na dalju budućnost. Evo nekih primjera takvih krivih predviđanja.

- Pioniri računarstva vjerovali su u 1940-tim godinama da će Amerika u predstojećim desetljećima trebati svega 4-5 snažnih računala!
- U 1960-tim godinama većina znanstvenika smatrala je da će oko 2000-te godine ljudi stati svojom nogom na Mars, izgraditi stalno naselje na Mjesecu, stvoriti ogromno super-računalo s osobinama umjetne inteligencije.

Dizajneri ARPANet-a (preteče Interneta) u 1970-tim godinama zamišljali su svoju mrežu isključivo kao akademsku infrastrukturu za FTP i Telnet. Prije 30-tak godina, ni najveći vizionari nisu imali dovoljno mašte da prepoznaju važnost umrežavanja, da predvide nastanak globalne mreže Internet, te da sagledaju društvene i kulturološke posljedice koje će iz toga proizaći. Zato je uzalud je očekivati da mi danas možemo predvidjeti kao će Internet izgledati za 20-30 godina. Pouzdane prognoze mogu se odnositi samo na najbližu budućnost – do 5 godina unaprijed.

Vidljivi trendovi razvoja

Vrlo je vjerojatno da će se u sljedećih nekoliko godina nastaviti trendovi razvoja Interneta koje uočavamo danas. Dakle:

- Broj čvorova spojenih u globalnu mrežu i dalje će eksponencijalno rasti, tako da se udeseterostruči svake 3-4 godine.
- Povećavat će se udio bežičnih veza u odnosu na fiksne.
- Umjesto klasičnih računala, u mreži će prevladavati mali uređaji poput mobitela i PDA.
- Internet će postati "svugdje prisutan": u perilici rublja, u automobilu, televizoru, foto-aparatu, …
- Razvit će se *grid*-tehnologija, koja će omogućiti transparentno korištenje neograničenih računalnih i podatkovnih resursa onda kad ih trebamo.

Problem s IP-adresama

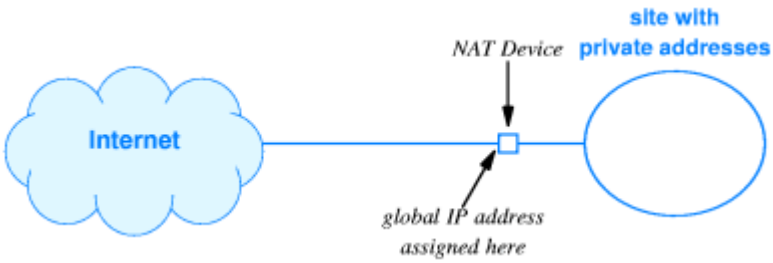
Postoji ozbiljna zapreka koja ograničava daljnji rast Interneta. Riječ je o sadašnjem formatu IP-adrese. Makar je s 32 bita u principu moguće zadati oko 4 milijarde IP-adresa, stvarni broj je daleko manji zato što pojedini dijelovi adrese služe za klasifikaciju.

Zaliha slobodnih IP-adresa stalno se smanjuje. Očekivalo se da će ona do danas već nestati. Razlog zašto danas još uvijek imamo slobodnih IP-adresa je u tome što se primjenjuju kratkoročna rješenja problema – dijeljenje jedne adrese na veći broj računala. Paralelno se radi i na dugoročnom rješenju – prelasku na nove 128-bitne IP-adrese.

Kratkoročno rješenje problema – NAT

Kratica *NAT* znači *prevođenje mrežnih adresa* (Network Addres Translation). Riječ je o dosjetljivoj tehnologiji koja omogućuje da jedan veliki segment Interneta (jedna kompanija, korisnici jednog ISP) troši samo jednu IP-adresu.

Svako računalo unutar segmenta dobiva *privatnu* IP-adresu, koja se inače ne koristi u pravom Internetu, no možda je koriste drugi slični segmenti. Segment unutar sebe funkcionira kao izolirani Internet. Računala unutar segmenta međusobno komuniciraju pomoću svojih privatnih adresa.



Slika 26.1: korištenje tehnologije NAT, postavljanje uređaja za NAT.

Segment je povezan s pravim Internetom preko jedne komunikacijske linije na kojoj je smješten posebni *uređaj za NAT* (NAT device) – vidi Sliku 26.1. Kad klijent uputi poruku poslužitelju u pravom Internetu, uređaj za NAT prerađuje tu poruku tako da privatnu IP adresu pošiljatelja zamijeni s *globalnom* IP-adresom koja je pridružena cijelom segmentu. Poslužitelju izvan segmenta cijeli segment izgleda kao jedno računalo. Kad poslužitelj pošalje odgovor, uređaj za NAT prerađuje odgovor tako da se globalna IP-adresa primatelja zamijeni s IP-adresom klijenta koji je prethodno slao poruku tom poslužitelju.

Postupak je nešto složeniji ukoliko više klijenata iz segmenta istovremeno šalju svoje poruke istom poslužitelju izvan segmenta. Da bi ispravno usmjerio poslužiteljeve odgovore, uređaj za NAT tada mora pratiti i po potrebi mijenjati TCP-portove klijenata. Preciznije, uređaj stvara tablicu prevođenja IP-adresa i TCP-portova, kao u primjeru na Slici 26.2.

Direction	Fields	Old Value	New Value
out	IP SRC:TCP SRC	10.0.0.1:30000	128.10.19.20:40001
out	IP SRC:TCP SRC	10.0.0.2:30000	128.10.19.20:40002
in	IP DEST:TCP DEST	128.10.19.20:40001	10.0.0.1:30000
in	IP DEST:TCP DEST	128.10.19.20:40002	10.0.0.2:30000

Slika 26.2: primjer tablice prevođenja IP adresa i TCP portova.

Dugoročno rješenje problema – IPv6

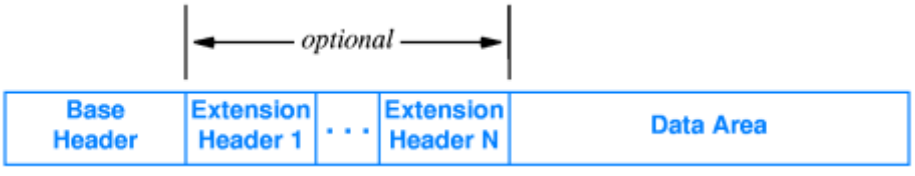
IPv6 je nova generacija interneta, koju je definirala organizacija IETF, i koja bi uskoro trebala zamijeniti sadašnju verziju IPv4. Osim što rješava problem s IP-adresama, IPv6 donosi i razna druga poboljšanja.

Najvažnije novosti u IPv6 su:

- *Produljenje IP adrese* sa sadašnjih 32 bita na 128 bitova.
- *Hijerarhijska građa IP adresa*, uvođenje više razina hijerarhije.
- *Novi oblici adresiranja*, koji na primjer omogućuju slanje istih podataka grupi računala.
- *Novi format datagrama*. Uvođenje glavnog zaglavlja i većeg broja neobaveznih dodatnih zaglavlja. Drukčija polja u zaglavljima.
- *Podrška za multimediju*. Uvodi se mehanizam koji omogućuje bolje osiguranje kakvoće mrežnih usluga za aplikacije koje to trebaju.
- *Fleksibilnost i nadogradivost*. Postoji mehanizam koji će omogućiti da se u budućnosti dodaju nove informacije u datagram, u skladu s nekim novim za sada nepoznatim potrebama.

Format datagrama u IPv6

Datagram sadrži osnovno zaglavlje, zatim 0 ili više dodatnih zaglavlja, te na kraju podatke. Zaglavlja imaju različite duljine. Opći oblik vidi se na Slici 26.3.



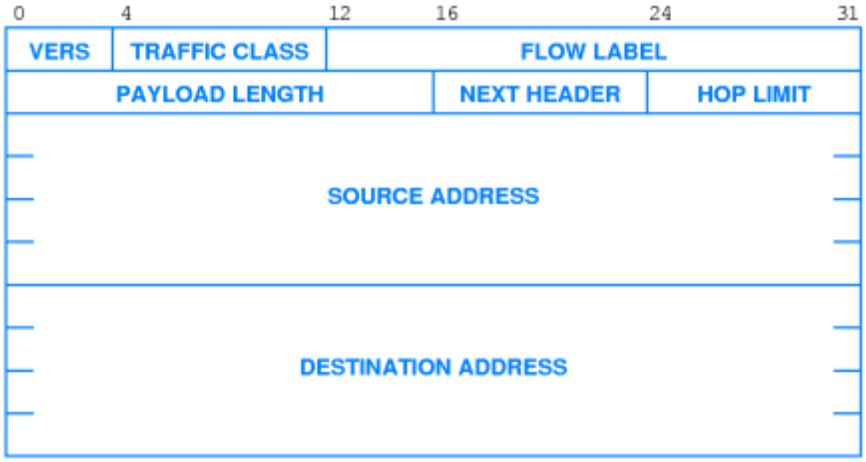
Slika 26.3: opći oblik IPv6 datagrama.

Osnovno zaglavlje ima fiksnu duljinu 40 byte i građeno je kao što se vidi na Slici 26.4. Najveći dio prostora zauzimaju:

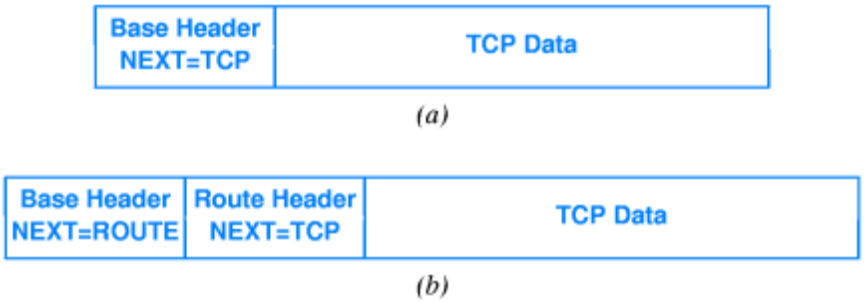
- adresa pošiljatelja SOURCE ADDRESS, i
- adresa primatelja DESTINATION ADDRESS.

Ostatak čine šest polja.

- VERS označava verziju IP protokola, dakle 6.
- PAYLOAD LENGTH je duljina dijela s podacima.
- TRAFFIC CLASS određuje traženu kvalitetu usluge. Za multimediju tražit ćemo bolju klasu.
- HOP LIMIT je broj skokova koje datagram smije napraviti prije nego što se odbaci.
- FLOW LABEL je također namijenjen za aplikacije koje traže garantirane performanse. Kad mreža pronađe put koji zadovoljava zahtjeve iz TRAFFIC CLASS, ona vraća identifikator. Pošiljatelj stavlja taj identifikator u datagramov FLOW LABEL. Usmjernici koriste vrijednost FLOW LABEL da bi usmjeravali datagram točno po odabranom putu.
- NEXT HEADER određuje vrstu informacije koja slijedi iza osnovnog zaglavlja. Ako datagram sadrži dodatno zaglavlje, NEXT HEADER sadrži tip prvog dodatnog zaglavlja. Ako nema dodatnog zaglavlja, NEXT HEADER navodi vrstu podataka u dijelu za podatke.



Slika 26.4: format osnovnog zaglavlja u IPv6.



Slika 26.5: IPv6 datagram koji sadrži (a) samo osnovno zaglavlje i podatke; (b) osnovno zaglavlje, jedno dodatno zaglavlje i podatke.

Polje NEXT HEADER nalazi se također i u dodatnim zaglavljima. Čitajući redom to polje u svakom od zaglavlja saznajemo što slijedi iza tog zaglavlja, sve dok ne dođemo do dijela s podacima – vidi Sliku 26.5. Neka od dodatnih zaglavlja imaju fiksnu a neka varijabilnu duljinu. Zaglavlja s varijabilnim duljinama sadrže podatak o svojoj vlastitoj duljini.

IP adrese u IPv6

Slično kao IPv4, i IPv6 pridružuje posebnu IP-adresu svakoj vezi između računala i fizičke mreže. To znači da računalo s više mrežnih sučelja (usmjernik) ima više adresa. Za razliku od IPv4 gdje se adresa sastojala od dva dijela, adresa u IPv6 može se sastojati od više dijelova, što omogućuje uspostavljanje hijerarhije na više razina. Možemo zamišljati da najviša razina odgovara nekom ISP-u, druga razina nekoj kompaniji, treća razina nekoj lokaciji, i tako dalje. Slično kao u CIDR, duljine dijelova adrese u IPv6 nisu fiksirane, već se mogu mijenjati jedna na račun druge.

Svaka adresa pripada jednom od sljedećih tipova.

- *Unicast*: adresa odgovara jednom računalu. Datagram se šalje tom jednom računalu.
- *Multicast*: adresa odgovara skupu računala. Članstvo u skupu može se mijenjati u svakom trenutku. Kopija datagrama šalje se svakom članu skupa.
- *Anycast*: adresa odgovara klasteru računala koja imaju zajednički prefiks u adresi. Datagram se isporučuje jednom (najbližem) članu klastera. To je korisno kad računala u klasteru zajednički obavljaju isti servis.

Pisanje 128-bitnih IP-adresa u *dotted decimal* notaciji postaje prilično nezgrapno, na primjer:

105.220.136.100.255.255.255.255.0.0.18.128.140.10.255.255

Da bi zapis adrese bio kompaktniji i čitljiviji, dizajneri IPv6 predlažu colon hexadecimal notaciju. U skladu s tim pravilom, svaka grupa od 16 bitova napiše se kao 4 heksadekadske znamenke. Između grupa umeću se dvotočke. Za prethodni primjer to izgleda ovako:

69DC:8864:FFFF:FFFF:0:1280:8C0A:FFFF

Daljnju optimizaciju zapisa IP-adrese omogućuje pravilo *zero compression*. Po tom pravilu, niz uzastopnih nula može se zamijeniti s dvije dvotočke. To se naravno smije napraviti samo na jednom mjestu u adresi jer se inače adresa više ne bi mogla jednoznačno reproducirati. Na primjer, adresa u colon hexadecimal notaciji:

FF0C:0:0:0:0:B1

piše se pomoću zero compression kao:

FF0C: :B1

Zero compression je učinkovito pravilo zato jer se očekuje da će mnoge IPv6 adrese sadržavati dugačke nizove nula. Na primjer, sve dosadašnje 32-bitne IPv4 adrese preslikat će se u nove IPv6 adrese tako da im se na početak doda 96 nula.

Prelazak na IPv6

Internet je ogroman i decentraliziran. Ne postoji način da se organizira istovremeni prelazak svih njegovih dijelova s IPv4 na IPv6. Zato se IPv6 mora uvesti postepeno, tako da računala i usmjernici koji razumiju samo IPv4 mogu nastaviti raditi što je dulje moguće. U tranzicijskom periodu treba omogućiti međusobno komuniciranje čvorova zasnovanih na IPv4. Također, treba omogućiti da čvorovi s IPv6 razgovaraju međusobno čak i ako dio infrastrukture između njih još uvijek podržava samo IPv4.

Za ostvarenje takve tranzicije predlažu se dva mehanizma.

- *Rad s dvostrukim stogom* (dual stack operation).
- *Tuneliranje* (tunneling).

Ideja dvostrukih stogova je u tome da čvorovi sposobni za IPv6 pokreću dva stoga protokola, dakle stogove za IPv4 i za IPv6. Na osnovu polja VERS unutar datagrama donosi se odluka koji stog treba obraditi taj datagram.

Tuneliranje je slanje IPv6 datagrama kroz dio mreže koji razumije samo IPv4. Slanje se odvija na sljedeći način

- IPv6 datagram na ulazu u "tunel" ulaže se kao korisni teret u IPv4 datagram.
- Taj IPv4 datagram šalje se kroz IPv4 mrežu prema IPv4 adresi izlaza iz tunela.
- Na izlazu iz tunela se iz IPv4 datagrama ponovo reproducira polazni IPv6 datagram.

Krajevi tunela moraju biti usmjernici ili računala koja su u stanju obrađivati i IPv4 i IPv6.

Sažetak Poglavlja 26

Iskustva iz prošlosti govore da nije zahvalno predviđati budućnost tehnološkog napretka. Ipak, s velikom sigurnošću možemo tvrditi da će se u sljedećih nekoliko godina nastaviti sadašnji trendovi razvoja Interneta, dakle eksponencijalni rast broja čvorova u mreži, povećanje udjela bežičnih veza u odnosu na fiksne, sve veći broj malih mobilnih uređaja umjesto klasičnih računala, i tako dalje. Najvažnija promjena koja se mora desiti u bliskoj budućnosti bit će prelazak na novu generaciju interneta IPv6, čime će se pojaviti sasvim novi format IP adresa i datagrama. Uvođenjem IPv6 dugoročno i trajno će se riješiti problem nedostatka IP adresa, dakle problem koji se ovog trenutka samo kratkoročno ublažava korištenjem tehnologije NAT. Zbog veličine i decentraliziranosti Interneta, prelazak na IPv6 neće biti lagan, već će se odvijati postepeno korištenjem mehanizma rada s dvostrukim stogom protokola, te mehanizma tuneliranja IPv6 datagrama kroz staru mrežu.

LITERATURA

1. Comer D.E. *Computer Networks and Internets with Internet Applications. Fifth Edition.* Pearson - Prentice Hall, 2009.
2. Peterson L.L., Davie B.S. *Computer Networks: A Systems Approach. Fifth Edition.* Morgan Kaufmann – Elsevier, 2011.
3. Stallings W. *Data and Computer Communications. Ninth Edition.* Pearson – Prentice Hall, 2010.
4. Kurose J.F., Ross K.W., *Computer Networking: A Top-Down Approach. Sixth Edition.* Pearson – Addison Wesley, 2012.
5. Tanenbaum A.S., Wetherall D.J., *Computer Networks. Fifth Edition.* Prentice Hall, 2010.
6. Stevens R.W., Fenner B., Rudoff A.M. *UNIX Network Programming, Vol. 1: The Sockets Networking API, Third Edition.* Addison Wesley, 2003.