TCP/IP KOMPIUTERINIAI TINKLAI

Saulius

Vilnius

ĮVADAS

Kiekvienas amžius turi vieną ar keletą dominuojančių technologijų. Viena iš svarbiausių 20-ojo amžiaus technologijų - informacijos rinkimas, apdorojimas, platinimas - glaudžiai susijusi su ryšių bei kompiuterinės technikos neprecedentiniu vystymusi. Ryšių ir kompiuterių apjungimas iš esmės pakeitė kompiuterinių sistemų organizavimo principus. Kompiuterinių centrų funkcijas dažnai atlieka daug atskirų, tačiau tarpusavyje sujungtų kompiuterių.

Kompiuterinis tinklas - tai tarpusavyje sujungtų autonominių kompiuterių rinkinys. Du kompiuteriai yra sujungti į tinklą, jeigu jie gali keistis informacija. Griežto pavaldumo sistemos, taip pat paskirstytos kompiuterinės sistemos nėra kompiuteriniai tinklai.

KOMPIUTERINIŲ TINKLŲ TAIKYMAI

Tinklai kompanijoms

- Bendras resursų naudojimas. Toli vienas nuo kito esantys kompanijos kompiuteriai apjungti į tinklą. Informacija nuo vartotojo gali būti labai toli. Ją gali naudoti daug darbuotojų tuo pačiu metu.
- Didelis patikimumas. Duomenų failai turi po keletą kopijų skirtinguose kompiuteriuose. Kai vienas sugenda, galima naudotis kitais, neprarandant duomenų.
- Pinigų taupymas. Vienas didelis kompiuteris (mainframe) yra brangesnis už daug mažų, sujungtų į tinklą ir atliekančių tą patį darbą. Duomenys saugomi viename arba keliuose bendro naudojimo failų serveriuose. Vartotojai - klientai. Kliento - serverio modelis, užklausa - atsakymas.
- Galimybė nesunkiai plėsti tinklą, pajungiant naujus vartotojus ir naujus serverius pagal poreikius.
- Galinga toli esančių darbuotojų bendravimo priemonė, atliekant bendrą darbą. Kartais tai yra svarbiau už anksčiau minėtus privalumus.

KOMPIUTERINIŲ TINKLŲ TAIKYMAI

Tinklai žmonėms

Pradedant nuo 1990 m. kompiuteriniai tinklai pradėjo teikti paslaugas privatiems asmenims į namus. Pagrindinės priežastys:

- Priėjimas prie nutolusios informacijos. Finansinės institucijos. Parduotuvės. Vaizdi informacija apie prekes. Elektroniniai laikraščiai. Mokslo žurnalai, bibliotekos. Pasaulio voratinklis (World Wide Web), kur galima rasti informacijos beveik apie viską. Visa tai - asmens ir nutolusios duomenų bazės sąveika.
- Asmenų bendravimas. Tai 21-mojo amžiaus atsakas į 19-tojo telefoną. Elektroninis paštas - naudojasi milijonai žmonių. Galima pasiųsti beveik viską : tekstus, vaizdus, garsus, failus. Trūksta tik kvapo.

  Toli vienas nuo kito esančių žmonių virtualūs susirinkimai - videokonferencijos. Nuotolinis mokymas, medicininė pagalba. Pasaulinės naujienų grupės.
- Interaktyvios pramogos. Video pagal pareikalavimą. Bet kokį bet kur padarytą filmą galima gauti tuoj pat. Interaktyvūs filmai (pasirenkant kokią nori siužeto kryptį). Interaktyvios televizijos laidos. Žaidimai. Trimatė virtuali realybė.

TINKLŲ APARATINĖS PRIEMONĖS

Du svarbūs požymiai:

- Perdavimo technologija
- Dydis

Pagal perdavimo technologiją tinklai skirstomi į du tipus:

- Transliacinio tipo tinklai (Broadcast networks).
- Taškas - taškas tipo tinklai (Point-to-point networks).

Transliaciniai. Turi vieną ryšio kanalą, kurį bendrai naudoja visi kompiuteriai tinkle. Kompiuteriai siunčia trumpus pranešimus, vadinamus paketais (packets). Siunčia vienas, gauna visi. Adreso laukas pakete nurodo, kam paketas adresuotas. Gavęs paketą, kompiuteris patikrina adreso lauką ir apdoroja paketą, jei jis yra skirtas minėtam kompiuteriui. Yra galimybė paketą adresuoti visiems adresatams, nurodant specialų kodą adreso lauke. Kai kurios sistemos palaiko perdavimą tik tam tikram kompiuterių poaibiui.

Taškas - taškas. Sudaryti iš daugelio sujungimų tarp individualių kompiuterių porų. Paketai nueina adresatui per keletą tarpinių stočių. Kadangi paketui yra skirtingo ilgio keliai, tai reikalingi maršrutizacijos algoritmai, kurie yra labai svarbūs taškas - taškas tinkluose. Dažniausiai maži, geografiškai lokalizuoti tinklai yra transliaciniai, o dideli - taškas - taškas tipo.

TINKLŲ APARATINĖS PRIEMONĖS

Tinklų klasifikacija pagal dydį

Atstumas tarp procesorių	Procesoriai išdėstyti bendrame(oje)	Pavyzdys
0,1 m	plokštėje	Duomenų srautų mašinos (Data flow machines)
1 m	sistemoje	Multikompiuteriai
10 m	kambaryje	Vietiniai tinklai (VT) (Local Area Networks, LAN)
100 m	pastate	
1 km	kvartale	
10 km	mieste	Miesto tinklai (Metropolitan Area Networks, MAN)
100 km	valstybėje	Globalieji tinklai (Wide Area Networks, WAN)
1000 km	kontinente	
10000 km	planetoje	Internet

Duomenų srautų mašinos - daugiaprocesoriniai lygiagretaus veikimo kompiuteriai su daugeliu funkcinių vienetų, dirbantys pagal vieną programą.

Multikompiuteriai - sistemos, besikeičiančios pranešimais per trumpas, labai greitas magistrales.

Tikrieji tinklai - tai kompiuteriai, bendraujantys per ilgesnius kabelius. LAN, MAN, WAN.

Keleto tinklų sujungimas yra vadinamas tarptinkliniu ryšiu (Internetwork). Pasaulinis Internetas - gerai žinomas tarptinklinių ryšių pavyzdys.

Atstumas labai svarbus klasifikacinis požymis, kadangi, priklausomai nuo jo, yra naudojama skirtinga technika tinklams realizuoti.

TINKLŲ APARATINĖS PRIEMONĖS

Vietiniai tinklai

Dažniausiai įrengiami viename pastate. Dydis - iki kelių kilometrų. Plačiai naudojami asmeninių kompiuterių ir darbo stočių sujungimui kompanijų, įstaigų kontorose bendram resursų naudojimui ir apsikeitimui informacija. Skiriasi nuo kitų tinklų dydžiu, perdavimo technologija, topologija.

Vietiniai tinklai riboto dydžio. Perdavimo laikas yra ribotas ir žinomas. Tai leidžia panaudoti tam tikrus metodus, kurie negalimi kitomis sąlygomis. Dažnai naudojamas vienas kabelis, prie kurio prijungti visi kompiuteriai.

Duomenų perdavimo sparta 10 - 100 Mb/s, mažas vėlinimas (dešimtys μs), nedaug klaidų. Tuo pačiu metu duomenis gali siųsti tik vienas kompiuteris. Reikalingas tam tikras konfliktus sprendžiantis mechanizmas, kai tuo pačiu metu duomenis nori siųsti keletas kompiuterių. Mechanizmas centralizuotas arba paskirstytas. Ethernet su magistraline topologija - tinklo su decentralizuotu valdymu pavyzdys.

Kitas transliacinio tinklo topologijos pavyzdys - žiedas. IBM token ring - populiarus žiedo pavidalo vietinis tinklas. Sparta 4 ir 16 Mb/s.

Transliaciniai tinklai toliau skirstomi į statinius ir dinaminius, priklausomai nuo to, kaip skiriamas kanalas. Statiniai, kai kiekvienam kompiuteriui yra skiriamas tam tikras laiko intervalas, kada jis gali siųsti (slot). Neproduktyviai panaudojamas kanalas, kai ne visos stotys turi ką siųsti. Dėl to dažniau naudojamos dinaminės sistemos, kai kanalas skiriamas dinamiškai pagal pareikalavimą. Yra centralizuoti ir decentralizuoti kanalo skyrimo metodai. Pirmuoju atveju yra vienas arbitražo įrenginys, kuris nusprendžia, kas perduos sekantis. Tai galima daryti priimant užklausas ir sprendžiant pagal tam tikrą algoritmą. Decentralizuotu atveju kiekviena stotis turi nuspręsti pati - siųsti ar ne. Yra įvairūs algoritmai kaip šiuo atveju išvengti chaoso.

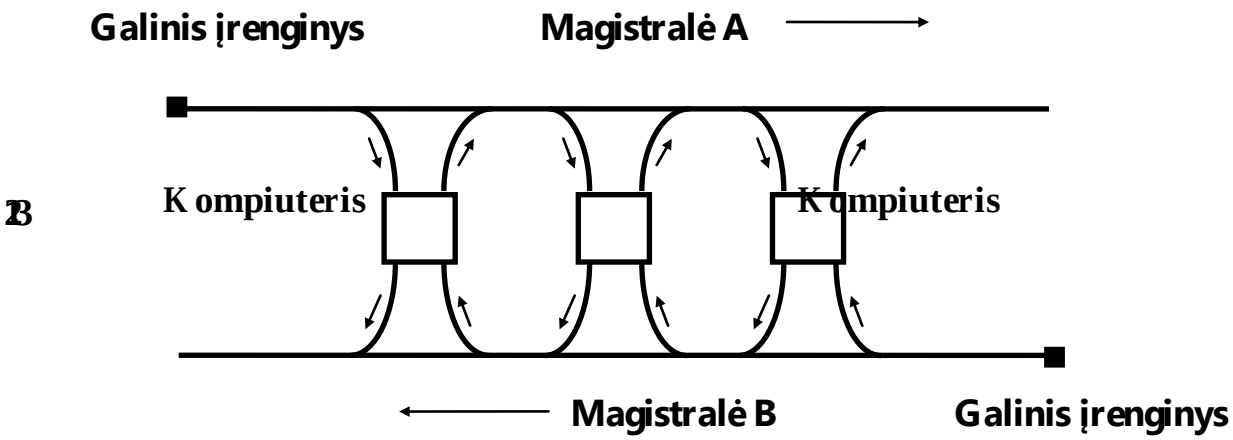
TINKLŲ APARATINĖS PRIEMONĖS

Miesto tinklai (MAN)

Tai didesnė vietinio tinklo versija. Miesto tinklas turi vieną arba du kabelius ir neturi komutatorių. Sukurtas specialus standartas - DQDB (Distributed Queue Dual Bus - paskirstytos eilės dviguba magistralė). Du vienkrypčiai kabeliai, prie kurių prijungti visi kompiuteriai. Kiekviena magistralė (kabelis) turi galinį įrenginį, inicijuojantį siuntimą.

Galiniai įrenginiai generuoja pastovų 53 baitų ilgio ląstelių (cells) srautą. Ląstelės keliauja į magistralės galą ir ten dingsta. Duomenų laukas 44 baitų. Kai ląstelė užimta, tą rodo *Busy* bitas. Bitas *Request* prilyginamas 1, kai stotis nusiunčia užklausą galimam duomenų išsiuntimui. Sudaroma eilė siųsti duomenis pagal pasiruošimo siųsti laiką. Išsiunčiama FIFO(First In First Out) tvarka. Eilei sudaryti kiekvienoje stotyje yra du skaitliukai: *RC* (Request Counter) ir *CD*. *RC* skaičiuoja, kiek yra išsiųsta užklausų iš visų stočių esančių prieš duotąją stotį, iki to momento, kai stotyje atsiranda celė siuntimui. Tada *RC* kopijuojamas į *CD* ir prilyginamas nuliui. Toliau *RC* skaičiuoja, kiek atėjo užklausų po celės paruošimo siuntimui. Pvz.: *CD* = 3, *RC* = 2. Tris tuščias ląsteles stotis praleidžia, po to siunčia ir po siuntimo dar 2 ląstelės lieka rezervuotos stotims, išsiuntusioms užklausas. Prieš siunčiant paruoštą ląstelę, stotis turi išsiųsti užklausą priešinga kryptimi. Tą užklausą mato visos stotys pakeliui ir padidina savo *RC*.

TINKLŲ APARATINĖS PRIEMONĖS



1 pav. Miesto tipo tinklo DQDB architektūra. IEEE 802.6 standartas.

TINKLŲ APARATINĖS PRIEMONĖS

Globalieji tinklai (WAN)

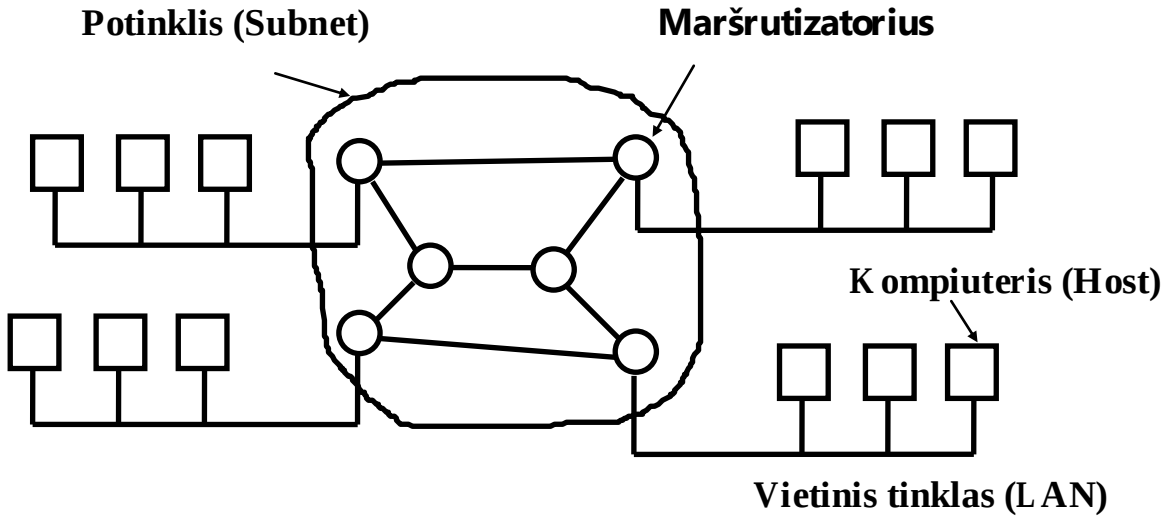
WAN užima didelę teritoriją (šalis, kontinentas). Į jį įeina kompiuteriai (hosts), kur vykdomos vartotojų programos, ir komunikacijų potinklis (communication subnet), kurio pagalba kompiuteriai sujungti į tinklą (2 pav.) Potinklis perneša pranešimus nuo vieno kompiuterio prie kito. Atskiriant grynai komunikacinius aspektus nuo taikomųjų, tinklų kūrimas supaprastėja.

Daugumoje WAN potinkliai susideda iš dviejų skirtingų komponenčių - perdavimo linijų ir komutacinių elementų. Linijos (circuits, channels, trunks) perduoda bitus iš vieno kompiuterio į kitą. Komutaciniai elementai yra specializuoti kompiuteriai, naudojami sujungti keletą linijų. Jie vadinami: paketų komutacijos mazgai, tarpinės sistemos, maršrutizatoriai (routers). Potinklis (subnet) dabar turi ir kitą reikšmę, susijusią su adresavimu tinkle.

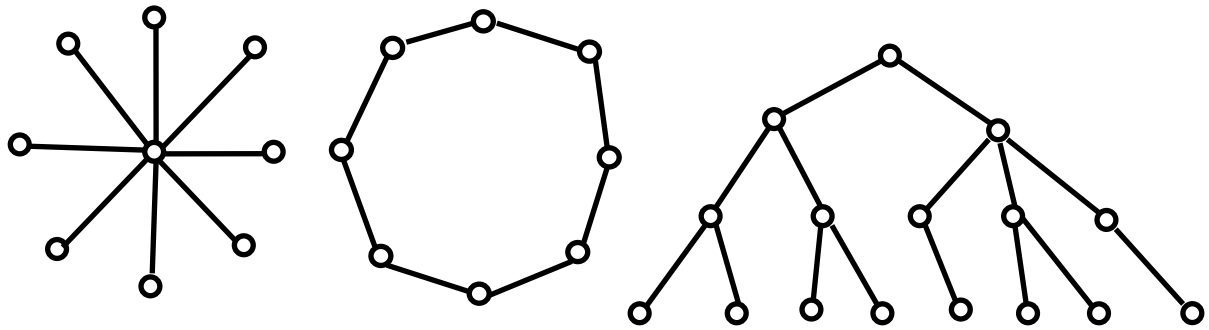
Kai paketas yra persiunčiamas iš vieno maršrutizatoriaus į kitą, jis gali praeiti per daugelį tarpinių maršrutizatorių. Kiekviename iš jų paketas priimamas ir saugomas, kol atsilaisvina reikalingas išėjimas. Potinklis, naudojantis šį principą, yra vadinamas taškas - taškas (point to point), išsaugoti ir persiųsti (store and forward) arba potinklis su paketų komutacija (packet switched subnet). Beveik visi WAN'ai (išskyrus satelitinius) turi tokius potinklius. Kai paketai yra maži ir fiksuoto dydžio, jie dažnai vadinami ląstelėmis. Yra įvairios taškas - taškas potinklių organizavimo topologijos (3 pav.). Vietiniai tinklai paprastai turi simetrinę topologiją. Globalieji tinklai dažniausiai yra nereguliarios topologijos.

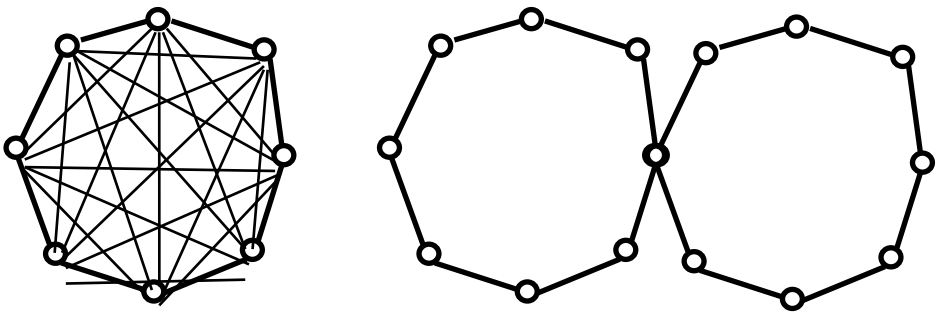
Dar yra palydoviniai tinklai ir antžeminės radijo sistemos. Kiekvienas maršrutizatorius turi anteną, per kurią priima ir perduoda. Visi arba tik kai kurie maršrutizatoriai gali priimti signalą iš palydovo.

TINKLŲ APARATINĖS PRIEMONĖS



2 pav. Ryšiai tarp kompiuterių (hosts) ir komunikacinio potinklio.





3 pav. Keletas galimų taškas - taškas potinklio topologijų.

TINKLŲ PROGRAMINĖ ĮRANGA

Protokolų hierarchija

Daugelis tinklų yra suorganizuoti kaip lygių arba sluoksnių rinkiniai, sudėti vienas ant kito. Kiekvieno lygio paskirtis - suteikti tam tikrą paslaugą aukštesniam lygiui. Lygis n viename kompiuteryje bendrauja su n lygiu kitame. Taisyklės ir susitarimai, naudojami šiam bendravimui, yra vadinami n lygio protokolu.

Fiziškai duomenys yra perduodami iš aukštesnio lygio į žemesnį, pasiekia fizinį lygį ir per fizinę aplinką perduodami į kitą kompiuterį. Ten pakyla iki atitinkamo lygio. Vienoduose lygiuose esantiems procesams susidaro įspūdis, kad jie bendrauja "horizontaliai", naudodami savo lygio protokolą.

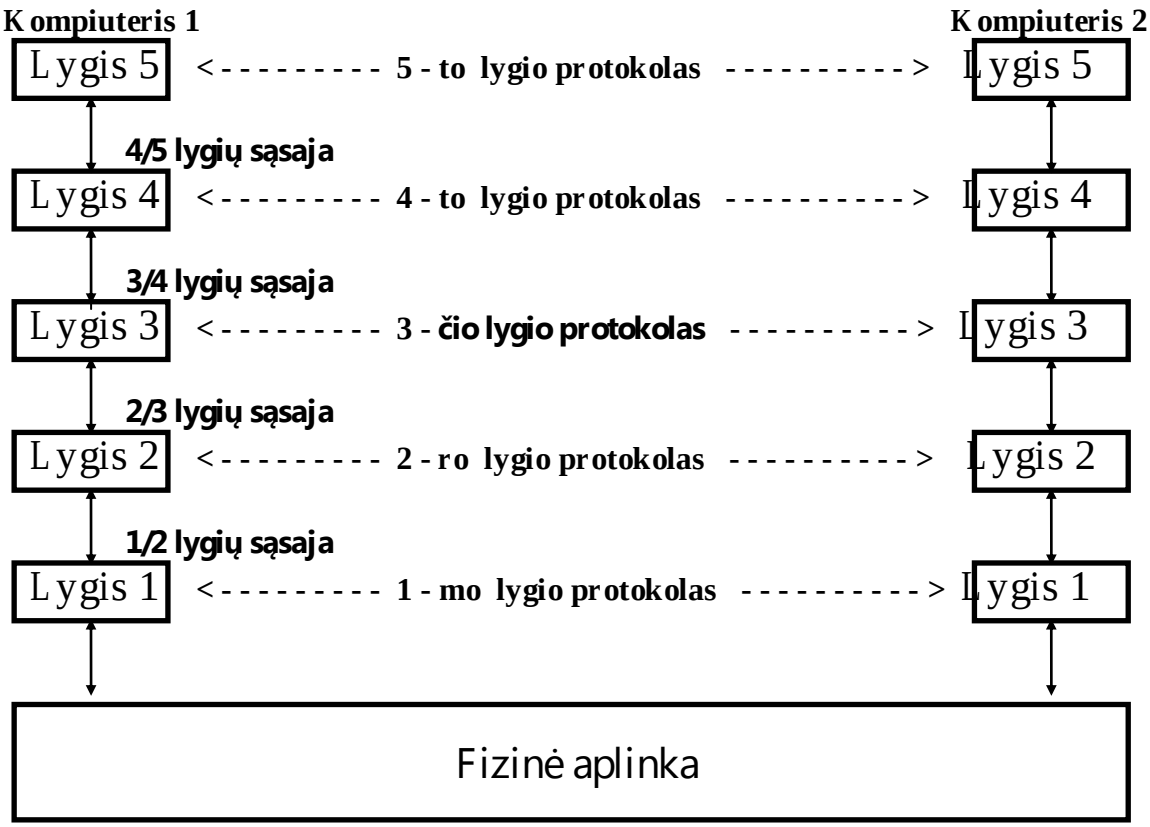
Tarp kiekvienos lygių poros yra sąsaja (interfeisas), nusakantis, kokiu būdu perduoti informaciją iš vieno į kitą. Kai sąsaja aiškiai apibrėžta, lengviau vieną lygio realizaciją pakeisti kita.

Lygių ir protokolų rinkinys vadinamas tinklo architektūra. Architektūros specifikacija turi turėti pakankamai informacijos, kad būtų galima parašyti programas arba padaryti aparatūrą kiekvienam lygiui. Nei realizacijos detalės, nei sąsajų specifikacijos nėra architektūros dalis. Protokolų sąrašas, naudojamas sistemoje, vienas protokolas vienam lygiui, yra vadinamas protokolų steku.

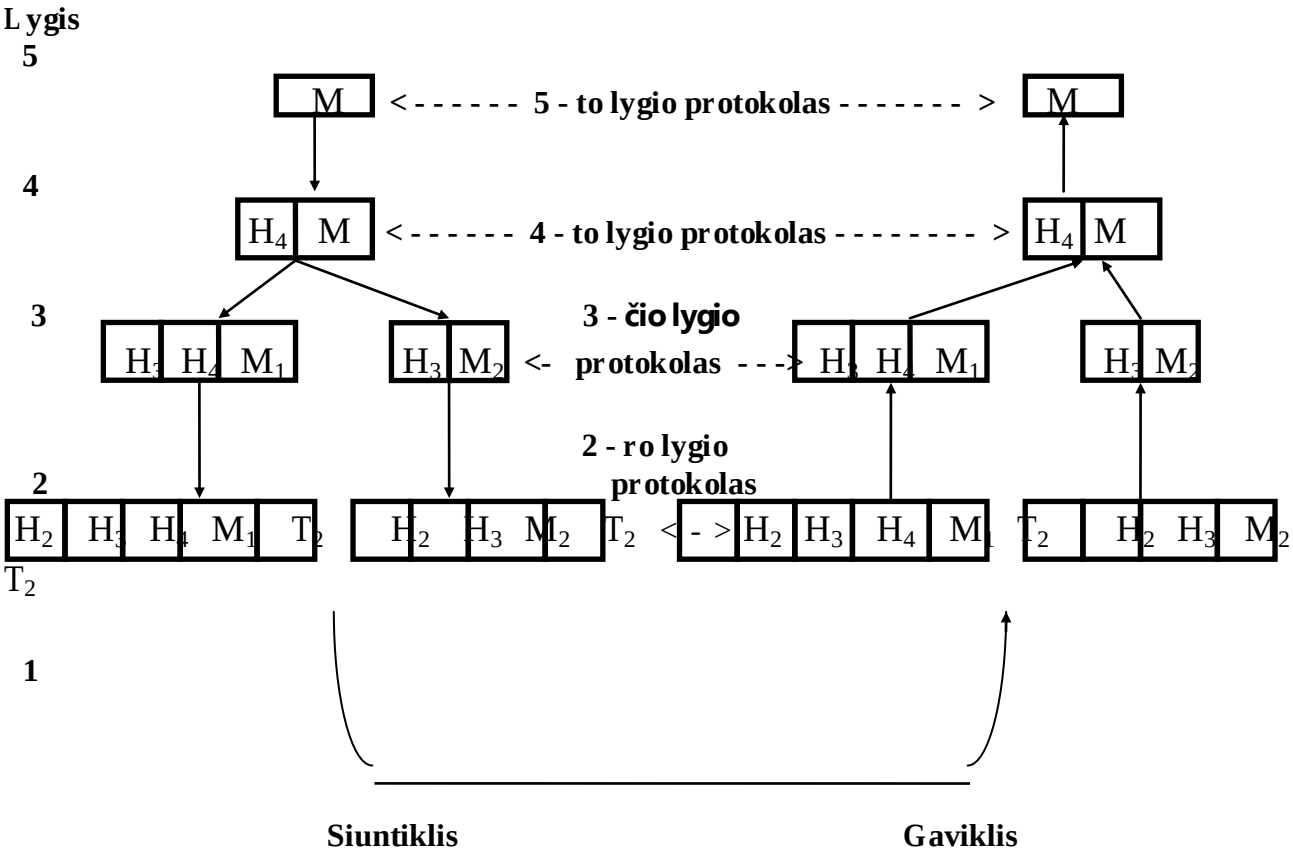
Pranešimą M suformuoja taikomasis procesas 5-tame lygyje ir perduoda į 4-tąjį išsiuntimui. Čia uždedama antraštė H_4 prieš M ir viskas perduodama į 3-čiąjį lygį. Antraštėje yra valdanti informacija, tokia kaip eilės numeris. Kitame kompiuteryje 4-tasis lygis pagal numerius galės pranešimus perduoti eilės tvarka, jei to neužtikrina žemesnieji lygiai. Pranešimų antraštės įvairiuose lygiuose gali turėti informaciją apie dydį, laiką ir t.t.

Daugelyje tinklų pranešimų dydis neribojamas 4-tame lygyje. Tačiau beveik visur ribojamas 3-čiame lygyje . Dėl to trečias lygis pranešimus gali sukarpyti į dalis, suteikiant kiekvienai daliai antraštę. 2-asis lygis duoda ne tik antraštę, bet ir pabaigą ir viską perduoda į fizinį lygį. Priimančioje pusėje viskas vyksta atvirkščia tvarka. Svarbus yra ryšys tarp virtualaus ir realaus perdavimo, taip pat skirtumas tarp sąsajos ir protokolo. Lygiaverčiai procesai

(peers) 4-tajame lygyje galvoja, kad jų bendravimas "horizontalus", naudojantis 4-tojo lygio protokolą. Lygiaverčių procesų abstrakcija yra labai svarbi visų tinklų kūrimui. Ją panaudojant galima didelį sunkiai valdomą viso tinklo kūrimo uždavinį sudalyti į keletą mažesnių ir paprastesnių, skirtų kiekvienam lygiui.



4 pav. Lygiai, protokolai, sąsajos



5 pav. Informacijos perdavimo 5-ame lygyje pavyzdys

TINKLŲ PROGRAMINĖ ĮRANGA

Lygių kūrimo ypatumai

Kiekvienam lygiui reikalingas siuntiklio ir gaviklio identifikavimo mechanizmas. Taip pat reikalingas adresavimas, nurodantis, kokiam procesui kompiuteryje yra skirtas pranešimas.

Svarbios yra duomenų perdavimo taisyklės. Gali būti :

1. Vienkryptis perdavimas (simplex communication).
2. Dvikryptis nevienalaikis perdavimas (half-duplex communication).
3. Dvikryptis vienalaikis perdavimas (full-duplex communication).

Klaidų kontrolė. Abi pusės turi naudoti tą patį klaidų radimo ar taisymo būdą. Gaviklis turi galėti pranešti, ar duomenys atėjo be klaidų ar su klaidomis.

Pranešimų eiliškumas. Pranešimai gali būti sunumeruoti. Kaip elgtis su ne laiku atėjusiais pranešimais?

Siuntiklio ir gaviklio veikimo spartos suderinimas. Vienas iš būdų - informuoti siuntiklį apie susidariusią situaciją gaviklyje, kad siuntiklis galėtų pakeisti duomenų išsiuntimo spartą.

Pranešimų ilgis. Dažnai ilgus pranešimus reikia išskaidyti, persiųsti ir vėl surinkti kitoje pusėje. Kartais yra efektyviau keletą pranešimų apjungti į vieną ir gaviklyje vėl išskaidyti.

Maršruto parinkimas, kai perdavimo kelių yra daug.

TINKLŲ PROGRAMINĖ ĮRANGA

Paslaugų primityvai

Paslauga formaliai nusakoma primityvų (operacijų) rinkiniu, kuriuo gali pasinaudoti vartotojas ar procesas. Primityvai skirstomi į keturias klases:

1. Request (Reikalavimas, paklausimas). Prašoma paslaugos atlikti tam tikrą darbą.
2. Indication (Indikacija). Procesas informuojamas apie įvykį.
3. Response (Atsakymas, reakcija). Noras atsakyti, reaguoti į įvykį.
4. Confirm (Patvirtinimas). Atsakymas į ankstesnį reikalavimą (Request)

Pvz.: Susijungimas ir atsijungimas. CONNECT.

Pasiunčiamas paketas CONNECT.request iš susijungimo iniciatoriaus. Gaviklis gauna CONNECT.indication, pranešantį, kad procesas iš kažkur reikalauja susijungimo. Gaviklis su CONNECT.response pasako, ar jis nori užmegzti ryšį ar ne. Siuntiklis sužino, kas atsitiko (priimtas pasiūlymas ar ne), per CONNECT.confirm primityvą.

Primityvai gali turėti parametrus. CONENCT.request gali būti nurodyta, prie kokios mašinos norima jungtis, koks reikalingos paslaugos tipas, maksimalus pranešimo ilgis. CONNECT.indication gali turėti kviečiančiojo identifikacinę informaciją, reikalingos paslaugos tipą, siūlomą maksimalų pranešimo ilgį. Jei gaviklis nesutinka su tokiu pranešimo ilgiu, tai alternatyvusis pasiūlymas gali būti atsakymo primityve, kuris gaunamas per CONNECT.confirm. Derybų detalės yra protokolo dalis. Pvz.: gali būti pasirinktas mažiausias pranešimo ilgis iš pasiūlytųjų.

Paslaugos gali būti patvirtintosios arba nepatvirtintosios. Pirmuoju atveju naudojami visi keturi primityvai. Antruoju atveju yra tik Request ir Indication. CONNECT yra visada patvirtinama paslauga. Duomenų perdavimas gali būti kaip patvirtintasis taip ir ne, priklausomai nuo siunčiančiosios pusės pareikalavimo. Tinkluose naudojami abu šie tipai.

BAZINIAI MODELIAI

OSI bazinis modelis

Sukurtas pagal ISO (International Standards Organization) pasiūlymą kaip pirmas žingsnis, einant į protokolų, naudojamų įvairiuose lygiuose, tarptautinį standartizavimą.

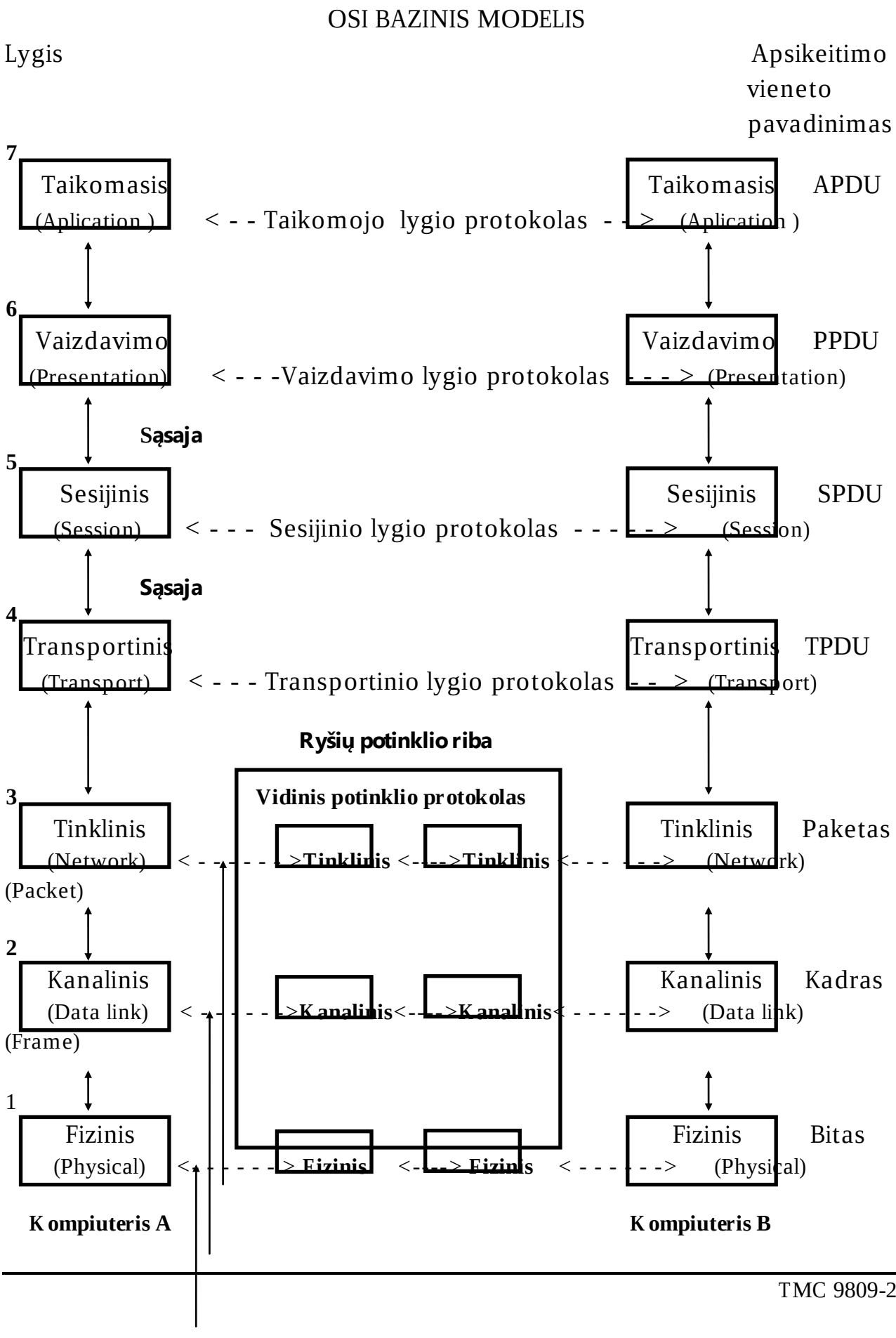
Pavadintas ISO OSI (Open Systems Interconnection) Reference Model. Atvirųjų sistemų tarpusavio sujungimo bazinis modelis.

Sudarytas iš 7 lygių (sluoksnių). Jų kūrimui naudoti principai:

1. Kiekvienas lygis turi vykdyti gerai apibrėžtas funkcijas.
2. Kiekvieno lygio funkcija turi būti pasirinkta orientuojantis į tarptautinių protokolų standartų sukūrimą.
3. Lygių ribos turi būti pasirinktos taip, kad minimizuotų informacijos srautą per sąsają.
4. Lygių skaičius turi būti pakankamai didelis, kad skirtingos funkcijos nepatektų be reikalo į tą patį lygį, ir pakankamai mažas, kad architektūra nepasidarytų griozdiška.

OSI modelis - tai ne tinklo architektūra, kadangi jis nenurodo tikslių paslaugų ir protokolų, kurie turi būti naudojami kiekviename lygyje. Jis tik pasako, ką kiekvienas lygis turi daryti. Tačiau ISO yra sukūrusi standartus visiems lygiams, nors jie nėra bazinio modelio dalis. Kiekvienas yra paskelbtas kaip atskiras tarptautinis standartas.

Pagrindinis ISO standartas OSI modeliui - ISO7498. Yra dar daug kitų ISO standartų, papildančių pagrindinį.


OSI BAZINIS MODELIS
Lygis
Apsikeitimo vieneto pavadinimas
7
Taikomasis (Aplication)
< - - Taikomojo lygio protokolas - - >
Taikomasis (Aplication)
APDU
6
Vaizdavimo (Presentation)
< - - -Vaizdavimo lygio protokolas - - - >
Vaizdavimo (Presentation)
PPDU
Sąsaja
5
Sesijinis (Session)
< - - - Sesijinio lygio protokolas - - - - - >
Sesijinis (Session)
SPDU
Sąsaja
4
Transportinis (Transport)
< - - - Transportinio lygio protokolas - - >
Transportinis (Transport)
TPDU
Ryšių potinklio riba
Vidinis potinklio protokolas
3
Tinklinis (Network)
< - - - - - - >Tinklinis <- - - >Tinklinis <- - - - - - >
Tinklinis (Network)
Paketas
(Packet)
2
Kanalinis (Data link)
< - - - - - - >Kanalinis <- - - >Kanalinis < - - - - - - >
Kanalinis (Data link)
Kadras
(Frame)
1
Fizinis (Physical)
< - - - - - - > Fizinis <- - - > Fizinis < - - - - - - >
Fizinis (Physical)
Bitas
Kompiuteris A
Kompiuteris B

—— Tinklinio lygio kompiuteris - maršrutizatorius protokolas
—— Kanalinio lygio kompiuteris - maršrutizatorius protokolas
—— Fizinio lygio kompiuteris - maršrutizatorius protokolas.

6 pav. OSI bazinis modelis

BAZINIAI MODELIAI

Fizinis lygis

Susijęs su bitų srauto perdavimu per ryšio kanalą. Kai viena pusė siunčia 1, tai kita turi gauti taip pat 1, o ne 0. Tipiški klausimai fiziniame lygyje. Kiek voltų reiškia 1, kiek voltų reiškia 0? Kokia bito trukmė? Ar perdavimas gali vykti į abi puses vienu metu? Kaip vykdomas pradinis susijungimas ir kaip jis užbaigiamas, kai abi pusės baigia siųsti? Kiek išvadų turi tinklinė jungtis, kam yra naudojamas kiekvienas išvadas? Projektavimas čia susijęs daugiausia su mechaninėmis, elektrinėmis ir procedūrinėmis sąsajomis, o taip pat su fizine perdavimo aplinka, kuri yra žemiau fizinio lygio.

Kanalinis lygis

Pagrindinė kanalinio lygio užduotis paimti neapdorotą perduodamą duomenų srautą ir transformuoti jį į eilutę, kuri patenka į tinklinį lygį be klaidų. Tai yra padaroma siunčiančioje dalyje suskaldant duomenis į duomenų kadrus, perduodant kadrus nuosekliai ir apdorojant patvirtinimo kadrus, atsiųstus atgal iš imtuvo. Kadangi fizinis lygis priima ir perduoda bitų srautą be jokio prasmės ar struktūros aiškinimosi, tai kanalinio lygio užduotis - sukurti ir atpažinti kadrų ribas. Tai gali būti padaryta, pridedant kadro pradžioje ir gale specialų bitų rinkinį. Jei tas rinkinys netyčia pasitaiko duomenyse, specialiomis priemonėmis jis atpažįstamas ir netraktuojamas kaip kadro riba.

Kanalinio lygio programinė įranga gali perduoti kadrą iš naujo, jei įvyko klaida ir nebuvo atsiųstas gavimo patvirtinimo kadras. Jeigu dingsta patvirtinimo kadras, gali būti išsiųstas duomenų kadro dublikatas. Taigi, kanalinis lygis turi išspręsti problemas, susijusias su kadrų pažeidimu,

praradimu, dubliavimu. Kanalinis lygis gali teikti keleto klasių paslaugas tinkliniam lygiui, skirtingos kokybės ir už skirtingą kainą.

Srauto valdymas (flow control), kai siųstuvas greitesnis už imtuvą. Paprastai srauto reguliavimas ir klaidų taisymas yra integruoti.

Transliaciniuose tinkluose įvedamas papildomas polygis MAC (Media Access Control - kreipties į aplinką valdymas) priėjimui prie bendro kanalo kontroliuoti.

BAZINIAI MODELIAI

Tinklinis lygis

Tinklinis lygis susijęs su potinklio (subnet) darbo valdymu. Svarbiausias dalykas yra apibrėžti, kaip paketai maršrutizuojami nuo šaltinio iki tikslo. Maršrutai gali būti paremti statinėmis lentelėmis, kurios retai keičiamos. Taip pat maršrutai gali būti nustatyti ryšio pradžioje. Galų gale jie gali būti dinaminiai, nustatomi kiekvienam paketui, atspindint realų tinklo apkrovimą tuo momentu.

Jei paketų labai daug potinklyje, jie gali sudaryti kamščius. Tokių perkrovimų reguliavimas priklauso tinkliniam lygiui.

Paketui pereinant iš vieno tinklo į kitą gali kilti daug problemų. Adresavimas, naudojamas antrajame tinkle, gali būti kitoks. Antroji pusė gali iš viso nepriimti paketo, kadangi jis per didelis. Protokolai gali skirtis ir t.t. Visas šias problemas turi išspręsti tinklinis lygis.

Transportinis lygis

Pagrindinė transportinio lygio funkcija - priimti iš sesijinio lygio duomenis, jei reikia padalyti į mažesnes dalis, perduoti į tinklinį lygį, užtikrinti, kad visos dalys teisingai pasiektų kitą pusę. Visa tai turi būti padaryta efektyviai, tokiu būdu, kad aukštesni lygiai būtų apsaugoti nuo neišvengiamų aparatūros pasikeitimų.

Paprastai transportinis lygis sukuria atskirą tinklinį sujungimą kiekvienam transportiniam sujungimui, kurio reikalauja sesijinis lygis. Jei reikia, transportinis lygis gali sukurti keletą tinklinių sujungimų vienam transportiniam. Ir atvirkščiai.

Populiariausias transportinio lygio sujungimas yra be klaidų kanalas taškas-taškas, kuris pristato pranešimus arba baitus tokia tvarka, kaip jie buvo išsiųsti. Tačiau viena iš transporto paslaugos rūšių gali būti izoliuotų pranešimų perdavimas be garantijos apie pristatymo eilę, o taip pat pranešimų tiražavimas į daugelį vietų.

Transportinis lygis yra tikras "end-to-end" lygis nuo šaltinio iki tikslo. Programa šaltinio kompiuteryje bendrauja su kita analogiška programa kitame sujungimo gale, naudodama pranešimo antraštes ir valdymo

pranešimus. Žemesniuose lygiuose protokolai yra tarp kompiuterio ir jo artimiausių kaimynų, o ne tarp šaltinio ir tikslo, kurie gali būti atskirti daugeliu maršrutizatorių.

BAZINIAI MODELIAI

Daugelis kompiuterių yra daugiaprogramiai. Tai reiškia, kad vienu metu gali egzistuoti daug sujungimų. Reikia nurodyti, koks pranešimas kokiam sujungimui priklauso. Transporto antraštė yra ta vieta, kur ši informacija patalpinama.

Reikalingas tam tikras vardų mechanizmas, nusakantis, koks procesas viename kompiuteryje nori bendrauti su kitu kitame kompiuteryje. Turi būti srauto valdymo mechanizmas, kad būtų suderintas apsikeitimas duomenimis tarp greito ir lėto kompiuterio (flow control).

Sesijinis lygis

Šis lygis leidžia vartotojams tarp skirtingų kompiuterių organizuoti sesijas. Sesijos leidžia paprastą duomenų padavimą kaip ir transportinis lygis. Be to jos užtikrina pagerintas papildomas paslaugas, reikalingas kai kuriems taikymams. Sesija gali būti panaudota vartotojui įeiti į nutolusią sistemą (log into) arba perduoti failą tarp dviejų kompiuterių.

Viena iš paslaugų yra valdyti dialogą. Sesijos gali leisti duomenims eiti abiem kryptim tuo pačiu metu arba tik viena kryptimi.

Dar viena paslauga - žymės valdymas. Kai reikia užtikrinti, kad sesijos metu nebus bandoma atlikti tą pačią operaciją iš abiejų pusių. Šiam procesui valdyti padaroma žymė (token). Tik ta pusė, kuri turi žymę, gali atlikti kritinę operaciją.

Kita paslauga - sinchronizacija. Pvz.: perduodant ilgą failą nepatikimu kanalu. Nutrūkus ryšiui, kitą kartą galima pratęsti siuntimą nuo tos vietos, kur jis buvo nutrauktas, jei į duomenų srautą yra įterpiami specialūs patikrinimo taškai.

BAZINIAI MODELIAI

Vaizdavimo lygis

Skirtingai nuo žemesniųjų lygių, kurie domisi tik patikimu bitų perdavimu iš vienos vietos į kitą, vaizdavimo lygis susijęs su perduodamos informacijos sintakse ir semantika.

Tipiška paslauga - duomenų kodavimas. Dauguma vartotojų programų nesikeičia atsitiktinėmis bitų eilutėmis. Jos keičiasi žmonių vardais, datomis, pinigų kiekiais ir t.t. Šie dalykai pateikiami kaip ženklų eilutės, sveiki skaičiai, skaičiai su slankiu kableliu ir t.t. Skirtingi kompiuteriai turi skirtingus kodus vaizduoti simbolių eilutes (pvz.: ASCII, UNICODE) ir t.t. Kad skirtingi kompiuteriai galėtų keistis duomenimis, duomenų struktūros turi būti pateiktos tam tikru abstrakčiu būdu. Vaizdavimo lygis valdo šias abstrakčias duomenų struktūras ir verčia kompiuteryje naudojamą pateikimo būdą į standartinį, naudojamą tinkle, ir atvirkščiai.

Taikomasis lygis

Vienas iš plačiai naudojamų taikymų - terminalo emuliavimas. Egzistuoja daugybė realių ir labai skirtingų terminalų. Terminalo emuliavimo programa leidžia sumodeliuoti tam tikrą abstraktų terminalą realiai egzistuojančiame. Visa virtualaus terminalo programinė įranga yra taikomajame lygyje.

Kita taikomojo lygio funkcija - failų perdavimas. Skirtinguose kompiuteriuose skirtingos failų sistemos. Taikomasis lygis turi išspręsti visas nesuderinamumo problemas. Šiam lygiui taip pat priklauso elektroninis paštas, užduočių įvedimas į nutolusius kompiuterius, katalogų peržiūra ir daug kitų taikomųjų dalykų.

BAZINIAI MODELIAI

TCP/IP bazinis modelis

Vienas iš sukūrimo tikslų - galimybė sujungti daug tinklų be aiškiai matomų siūlių. Kitas tikslas - ryšiai neturi nutrūkti tol, kol egzistuoja siuntiklis ir gaviklis, nors tuo metu didelė potinklio aparatūros dalis jau nedirba (išgyventi atominio karo metu). Reikalinga lanksti architektūra, nes numatomas labai platus taikymo spektras, pradedant failų siuntimu ir baigiant kalbos perdavimu realiame laike. TCP/IP pirmą kartą apibrėžtas 1974 m. Toliau buvo nuolat tobulinamas.

Tarptinklinis (Internet) lygis. Aukščiau minėti reikalavimai vedė prie tinklo su paketų komutacija su tarptinkliniu lygiu, neorientuotu į sujungimus. Šis lygis, kaip rišamoji medžiaga, laiko visą architektūrą. Jo darbas yra leisti kompiuteriams injektuoti paketus į tinklą. Paketai keliauja į tikslą nepriklausomai vienas nuo kito (galbūt į kitą tinklą). Jie gali atkeliauti kitokia eilės tvarka negu buvo išsiųsti. Aukštesniųjų lygių reikalas juos sustatyti į savo vietas.

Tarptinklinis lygis apibrėžia paketų formatą ir protokolą, pavadintą IP (Internet protocol). Pagrindiniai dalykai - paketų maršrutizavimas ir tinklo užsikimšimų likvidavimas (avoiding congestion). Labai panašu į OSI tinklinį lygį.

Transportinis lygis - tai lygis virš tarptinklinio TCP/IP modelyje. Jis leidžia bendrauti lygiaverčiams lygiams abiejose pusėse kaip OSI transportinis lygis. Čia apibrėžti du end-to-end protokolai. TCP (Transport Control Protocol - perdavimo valdymo protokolas) - patikimas, orientuotas į sujungimą protokolas. Jis fragmentuoja ateinantį baitų srautą į atskirus pranešimus ir kiekvieną jų perduoda į tarptinklinį lygį. Priimančioje pusėje atitinkamas TCP apjungia visus pranešimus. TCP taip pat palaiko srauto valdymą (flow control), kad greitas siųstuvas neperkrautų lėto imtuvo.

UDP (User Datagram Protocol - vartotojo deitagramų protokolas). Nepatikimas, be sujungimų protokolas taikymams, kuriems nereikia TCP eiliškumo tvarkymo ar srauto valdymo ir kurie linkę tvarkyti patys. Jis plačiai taikomas kliento-serverio tipo klausimas-atsakymas atvejams, taip pat taikymams, kai greitas pristatymas yra svarbesnis už pristatymą be klaidų.

BAZINIAI MODELIAI

	OSI	TCP/IP
7	Taikomasis	Taikomasis
6	Vaizdavimo	Modelyje neegzistuoja
5	Sesijinis	
4	Transportinis	Transportinis
3	Tinklinis	Tarptinklinis (Internet)
2	Kanalinis	Kompiuteris - tinklas (Host to network)
1	Fizinis	

7 pav. TCP/IP bazinis modelis

BAZINIAI MODELIAI

Taikomasis lygis. Čia visi aukšto lygio protokolai. Nuo seno ten yra virtualus terminalas (TELNET), failų perdavimas (FTP - File Transfer Protocol) ir elektroninis paštas (SMTP - Simple Mail Transfer Protocol). Veliau atsirado DNS (Domain Name Service), surišantis kompiuterių vardus su jų tinkliniais adresais. NNTP (Network News Transfer Protocol) - naujienų straipsnių perdavimui. HTTP (Hyper Text Transfer Protocol) - WWW puslapių perdavimui. Ir daug kitų protokolų.

Lygis kompiuteris-tinklas. Čia gana neapibrėžta situacija. Galima naudoti bet ką, kad tik IP paketai būtų perduoti per tinklą. Vėliau atsirado SLIP (Serial Line Internet Protocol) ir PPP (Point-to-Point Protocol). Tai protokolai atitinkantys OSI kanalinį lygį.

OSI ir TCP/IP bazinių modelių palyginimas

Lyginami baziniai modeliai, bet ne atitinkami protokolų stekai. OSI modelyje centriniai yra trys dalykai: paslaugos, sąsajos, protokolai. Jie labai aiškiai yra atskirti. Paslaugos apibrėžimas pasako, ką lygis daro, o ne kaip procesai virš jo jį pasiekia arba kaip lygis dirba. Lygių sąsaja pasako procesams virš lygio, kaip jį pasiekti. Jis lemia, kokie yra parametrai ir kokių rezultatų reikia tikėtis. Sąsaja taip pat nepasako, kaip lygis veikia.

Protokolai, naudojami lygyje, yra lygio reikalas. Protokolai gali būti įvairūs, svarbu, kad būtų suteikta reikalinga paslauga aukštesniajam lygiui. Pakeitimai protokole nekeičia programinės įrangos, dirbančios kituose lygiuose.

TCP/IP neturi aiškaus paslaugos, sąsajos ir protokolo atribojimo. Protokolai OSI modelyje labiau paslėpti ir gali būti palyginus lengvai pakeisti, kai keičiasi technologija. Tai yra labai svarbu.

OSI modelis buvo išrastas ankščiau negu protokolai. Jis nebuvo paremtas vienu kokiu nors protokolų steku. Dėl to OSI yra pakankamai bendro pobūdžio. Tačiau OSI kūrėjai neturėjo labai didelio patyrimo ir neturėjo aiškių minčių, į kokį lygį, kokias funkcijas dėti. Pvz. MAC polygis

kanaliniame lygyje iš pradžių visai neegzistavo. Jis neišvengiamai turėjo atsirasti plintant vietiniams tinklams, kurie yra transliacinio pobūdžio.

Su TCP/IP buvo atvirkščia situacija. Protokolai atėjo pirmieji, o modeliai buvo tik egzistuojančių protokolų aprašymas. Protokolai puikiai atitiko modelį, tačiau į modelį neįsipaišė jokie kiti protokolų stekai.

BAZINIAI MODELIAI

OSI turi 7 lygius, TCP/IP - keturis. Trys maždaug sutampa (tarptinklinis, transportinis, taikomasis). Kiti skiriasi.

Apie 1989 m. daugeliui ekspertų atrodė, kad OSI ir atitinkami protokolai greitai nustums visus kitus į šoną. Taip neatsitiko. Dabar atvirkščiai, atrodo, kad OSI bankrutuoja. Kol buvo bandoma realizuoti OSI, TCP/IP toliau masiškai plito, ypač, pradedant nuo 1990 metų. OSI modelio ir protokolų sudėtingumas privedė prie to, kad pradinės realizacijos buvo didžiulės, griozdiškos ir lėtos. Skirtingai nuo OSI, viena iš pirmųjų TCP/IP realizacijų buvo Berklio UNIX'o dalis ir pakankamai gera. Žmonės labai greitai pradėjo ją naudoti, vartotojų skaičius išaugo. Tai vedė prie įvairių patobulinimų, kas, savo ruožtu, dar padidino vartotojų bendruomenę.

Nežiūrint visų šių problemų, OSI modelis (minus sesijinis ir vaizdavimo lygis) yra labai naudingas, kalbant apie kompiuterių tinklus. Priešingai, OSI protokolai netapo populiariais. Su TCP/IP yra atvirkščiai. Modelis praktiškai neegzistuoja, bet protokolai yra plačiai naudojami.

ATM bazinis modelis

Didelė problema telefonų kompanijoms - tinklų įvairovė. Senieji telefonų tinklai, SMDS ir Frame Relay, naudojantys savo tinklą su paketų komutacija. DQDB skiriasi nuo jų, o vidinis telefonų kompanijos kvietimų valdymo tinklas (SSN 7) yra vėl kitoks. Šių visų tinklų palaikymas yra pakankamai sudėtingas dalykas.

Akivaizdus sprendimas - išrasti vieną naują tinklą ateičiai, kuris pakeistų visą telefonų sistemą ir visus specializuotus tinklus ir galėtų perduoti visų

rūšių informaciją. Šis tinklas turės žymiai didesnę spartą, lyginant su visais ankstesniais, ir galės duoti daug naujų paslaugų.

Nauja plataus masto paslauga vadinama B-ISDN (Broadband Integrated Services Digital Network - Plačiajuostis visuminių paslaugų skaitmeninis tinklas). Jis duos video pagal norus, tiesioginę televiziją iš daugelio šaltinių, elektroninį paštą su judančiais objektais, CD kokybės muziką, vietinių tinklų sujungimą, didelę duomenų perdavimo spartą, reikalingą moksle ir pramonėje ir daugelį kitų paslaugų, kurių dar neįmanoma įsivaizduoti. Viskas per telefono linijas.

Technologija, paverčianti B-ISDN realybe, yra vadinama ATM (Asynchronous Transfer Mode - Asinchroninė perdavimo moda).

BAZINIAI MODELIAI

Pagrindinė idėja - visą informaciją perduoti mažais fiksuoto ilgio paketais - ląstelėmis (cells). Ląstelės ilgis - 53 baitai, iš kurių 5 baitai skirti antraštei ir 48 baitai duomenims (8 pav.).

ATM yra tiek technologija (paslėpta nuo vartotojo), tiek paslauga (matoma vartotojui). Kartais paslauga vadinama ląstelių retransliacija (cell relay) pagal analogiją su frame relay. Ląstelių komutavimo technologija - didžiulis atotrūkis nuo 100 metų naudojamo grandinių komutavimo telefonų sistemos viduje.

Pagrindiniai argumentai ląstelių komutacijos naudai:

1. Technologija labai lanksti, palaiko tiek pastovios spartos srautą (audio, video), tiek kintamos spartos (duomenys) labai paprastai.
2. Esant labai didelėms spartoms (gigabitai/s) skaitmeninė ląstelių komutacija yra paprastesnė už tradicinę sutankinimo techniką, ypač naudojant šviesolaidžius.
3. Televizijai yra svarbu transliavimas (broadcasting). Tai gali daryti ląstelių komutacija, o ne grandinių komutacija.

ATM tinklai orientuoti sujungimams. Pirmiausiai įvyksta susijungimas, po to siunčiamos ląstelės tuo pačiu keliu. Pristatymas negarantuojamas, tačiau eilės tvarka garantuojama.

ATM tinklai organizuoti kaip tradiciniai WAN su linijomis ir komutatoriais (maršrutizatoriais). Tipinė sparta 155 Mb/s ir 622 Mb/s su

galimybe padidinti iki gigabitų. 155 Mb/s paimta dėl to, kad to reikia didelės raiškos televizijos signalui perduoti. Tikslus pasirinkimas - 155.52 Mb/s suderinimui su SONET/SDH.

ATM bazinis modelis skiriasi kaip nuo OSI, taip ir nuo TCP/IP modelio (9 pav.).

Fizinis lygis. Fizinė aplinka, įtampos, bitai ir kt. ATM ląstelės gali būti perduodamos į aplinką pačios arba gali būti kitų sistemų paketų viduje. Kitaip sakant, ATM yra sukonstruotas taip, kad būtų nepriklausomas nuo perdavimo aplinkos.

ATM lygis. Ląstelės ir jų perdavimas. Lemia ląstelių struktūrą, nusako antraštės laukų prasmę. Virtualiųjų grandinių sukūrimas ir atlaisvinimas. Perkrovos kontrolė (congestion control) taip pat čia.

Kadangi dauguma taikomųjų programų su ląstelėmis nedirba, tai lygis virš ATM yra tam, kad vartotojas galėtų siųsti paketus, didesnius už ląstelę. Jis segmentuoja paketus ir perduoda ląsteles, surenka paketus kitoje pusėje. Tai yra ATM adaptacijos lygis AAL (ATM Adaptation Layer).

BAZINIAI MODELIAI

Skirtingai nuo ankstesniųjų dvimačių bazinių modelių, ATM yra trimatis. Vartotojo plokštuma susijusi su duomenų perdavimu, srauto valdymu, klaidų taisymu ir kitomis vartotojo funkcijomis. Valdymo plokštuma susijusi su susijungimų valdymu. Lygio ir plokštumų valdymo funkcijos susiję su resursų valdymu ir tarplygine koordinacija.

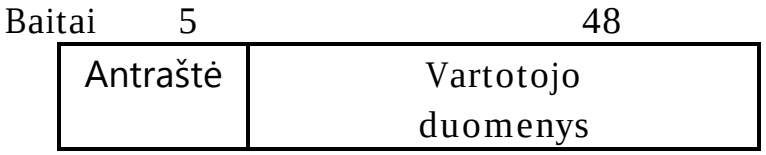
PDM (Physical Medium Dependent) polygis susijęs su realiu kabeliu. Jis perduoda bitus į kabelį, nustato bito trukmę. Skirtingiems nešliams ir kabeliams šis polygis skirtingas.

TC (Transmission Convergence) polygis. Kai ląstelės yra siunčiamos, TC polygis persiunčia jas kaip bitų srautą į PDM. Tai paprasta. Kitoje pusėje TC polygis gauna iš PDM bitų srautą, kurį reikia paversti į ląstelių srautą ATM lygiui. Tai gana sudėtinga. OSI modelyje ir daugelyje kitų suskirstymą į kadrus ir atvirkščiai atlieka kanalinis lygis.

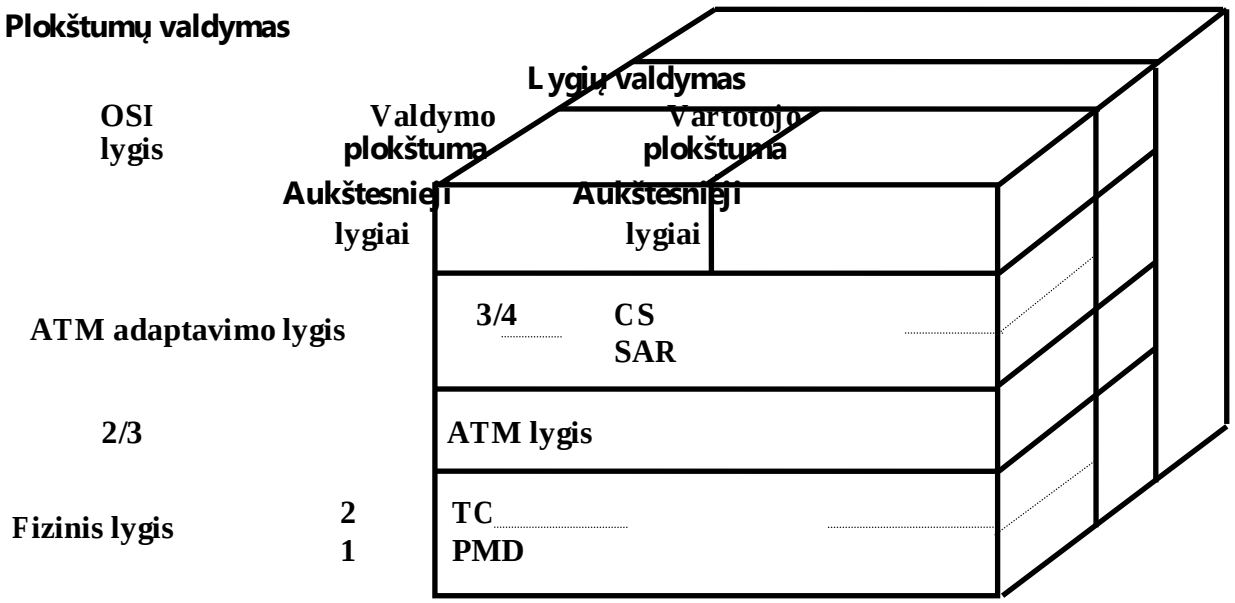
ATM lygis tvarko ląsteles, jas generuoja ir perduoda. Daugiausia įdomių ATM aspektų yra čia. Tai yra OSI tinklinio ir kanalinio lygio mišinys, tačiau į polygius jis nedalinamas.

AAL lygis padalintas į SAR (Segmentation and Reassembly) polygį ir CS (Convergence Sublayer). Žemesnysis polygis suskaldo paketus į ląsteles perduodančioje pusėje ir sulipdo atgal priimančioje pusėje. Aukštesnysis polygis suteikia galimybę ATM sistemoms teikti skirtingų rūšių paslaugas skirtingiems taikymams (pvz.: failų persiuntimas ir video pagal poreikį turi skirtingus reikalavimus klaidų taisymui ir kt.).

BAZINIAI MODELIAI



8 pav. ATM ląstelės struktūra.



9 pav. B-ISDN ATM bazinis modelis.

CS - konvergavimo polygis (Convergence),
SAR - segmentacijos ir surinkimo polygis (Segmentation and reassembly),
TC - perdavimo konvergavimo polygis (Transmission convergence),
PDM - polygis, priklausantis nuo fizinės aplinkos (Physical medium dependent).

TINKLŲ PAVYZDŽIAI

ARPANET

ARPANET - visų globaliųjų tinklų pirmtakas. Šešiasdešimtųjų viduryje, šaltojo karo įkarštyje JAV gynybos departamentui prireikė valdymo tinklo, kuris išgyventų atominį karą. Tradiciniai telefonų tinklai tam netiko. Departamentas kreipėsi į ARPA (Advanced Researchs Projects Agency), kuri buvo sukurta kaip atsakas į 1957 m. paleistą pirmąjį palydovą. Po diskusijų su įvairiais ekspertais ARPA nutarė, kad tinklas turi būti su paketų komutacija, susidedantis iš kompiuterių (hosts) ir potinklio (subnet). Potinklyje kiekvienas mazgas sujungiamas su mažiausiai dviem kitais mazgais. Jeigu linija yra pažeidžiama, paketai automatiškai nukreipiami į tikslą per kitus mazgus. Tinklas buvo sukurtas ir pavadintas ARPANET. Jam išsiplėtus ir susikūrus kitiems tinklams, iškilo jų apjungimo problema, kuri buvo išspręsta TCP/IP modelio ir protokolų pagalba 1974 m. Vėliau TCP/IP buvo integruotas į Berklio UNIX'ą, sukurta daug taikomųjų ir tinklo valdymo programų.

1983 m. ARPANET turėjo 200 potinklio mazgų ir šimtus kompiuterių. Gana greitai tinklas išaugo, prisijungė daug vietinių tinklų. Pasidarė sunku surasti kompiuterį tinkle. Buvo sukurta DNS (Domain Naming System), sugrupuojant kompiuterius į grupes-domenus, surišant kompiuterio vardą su jo IP adresu. Nuo tada DNS tapo bendro pobūdžio paskirstyta duomenų bazė.

Apie 1990 metus ARPANET buvo pakeistas naujesniais tinklais. Išliko tik karinė dalis MILNET.

Internet

Tinklų, kompiuterių ir vartotojų, prisijungusių prie ARPANET kiekis augo labai greitai po to, kai TCP/IP tapo vieninteliu oficialiu protokolu 1983 m. sausio 1 d. Kai ARPANET ir NSFNET buvo sujungti, augimas pasidarė eksponentinis. Taip atsirado Internet, kuris iki 1990 m. išaugo iki 3000 tinklų ir 200000 kompiuterių. 1992 m. buvo prijungtas milijoninis kompiuteris. Internet dydis padvigubėja kasmet. Yra daugybė transatlantinių ryšių nuo 64Kb/s iki 2Mb/s.

TINKLŲ PAVYZDŽIAI

Rišančioji Internet medžiaga yra TCP/IP bazinis modelis ir TCP/IP protokolų stekas. Kompiuteris yra Internete, jeigu jame veikia TCP/IP protokolų stekas, jis turi IP adresą ir gali siųsti IP paketus visiems kitiems kompiuteriams Internete. Galimybės siųsti ir gauti elektroninį paštą neužtenka, kadangi paštas per šliuzus nueina ir į tinklus, neesančius Internete. Tie, kurie dirba per modemą, laikomi esančiais Internete tol, kol yra prijungti prie Interneto paslaugų tiekėjo maršrutizatoriaus.

1992 m. buvo sukurta Interneto bendruomenė (Internet Society), padedanti naudotis Internetu ir jį valdanti.

Tradiciškai egzistuoja keturi pagrindiniai Internet taikymai:

1. Elektroninis paštas (Email). Galimybė siųsti ir gauti laiškus egzistuoja nuo pirmųjų ARPANET dienų ir yra nepaprastai populiari.
2. Naujienos. Naujienų grupės - tai specializuoti forumai, kuriuose vartotojai, turį tam tikrų interesų, gali keistis pranešimais. Egzistuoja tūkstančiai naujienų grupių. Kiekviena iš jų - turi savo stilių ir taisykles, kurias pažeidinėti nevalia.
3. Prisijungimas prie nutolusio kompiuterio (Remote login). Naudojant Telnet, Rlogin ar kitas programas, vartotojai bet kur Internete gali prisijungti prie kompiuterio, kuriame yra užregistruoti.
4. Failų perdavimas.

Iki 80-tųjų pradžios Internetu daugiausia naudojosi akademiniai, valstybiniai ir pramoniniai tyrinėtojai. WWW (World Wide Web) iš esmės tą paskirstymą pakeitė - milijonai neakademinių vartotojų atėjo į Internetą. WWW išrado CERN'o fizikas Tim Berners-Lee. Tai nekeitė esminių Interneto savybių, o supaprastino naudojimąsi jomis. WWW puslapių peržiūros programa Mosaic parašyta superkompiuterių taikymo nacionaliniame centre (National Center for Supercomputer Applications). Per pirmus metus nuo Mosaic sukūrimo WWW serverių kiekis išaugo nuo 100 iki 7000. Augimas

tęsiasi. Manoma, kad tai bus Interneto technologijos ir naudojimo varomoji jėga sekančiame tūkstantmetyje.

TINKLŲ PAVYZDŽIAI

Gigabitiniai tinklai

Gigabitiniai tinklai praplečia dažnių juostą, tačiau mažai pagerina signalų vėlinimą. Pvz.: 1-Kbit paketas 1 Mb/s greičiu per JAV keliauja 20 ms. Taigi, 1 ms reikia išsiųsti bitus ir 20 ms sudaro vėlinimas. Viso 21 ms. 1Gb/s tinklas gali tai sumažinti tik iki 20.001 ms. Greitis šviesolaidyje ~200000 Km/s ir nepriklauso nuo duomenų perdavimo spartos. Taigi, ten, kur vėlinimas labai svarbus, gigabitai ne ką padeda. Tačiau yra daug taikymų, kur dažnių juostos plotis labai svarbus.

Telemedicina. Kai atsiranda rimta medicininė problema, laboratorijos testai, rentgeno nuotraukos ir kiti paveiksliukai gali būti pasiųsti specialistui, kuris gali diagnozuoti. Daktarai nenoriai diagnozuoja, jei perduotų nuotraukų kokybė blogesnė už originalą. Tai reiškia, kad perdavimui reikia 4K x 4K taškų su 24 bitais kiekvienam taškui. Daugeliui testų reikia iki 100 vaizdų (įvairūs organų pjūviai). Vienam tyrimui susidaro 40Gbitų. Jei viskas juda (plaka širdis), duomenų stipriai padidėja. Suspaudimas kažkiek gelbsti, tačiau prarandama kokybė, ko nenori daktarai. Visi vaizdai turi būti kaupiami ir saugomi daug metų, kad būtų galima pasinaudoti, kai tik prireiks. Taigi, reikalingi labai greiti duomenų perdavimo kanalai.

Virtualūs susirinkimai. Kiekviename kambaryje yra sferinė kamera ir vienas ar daugiau žmonių. Bitų srautai iš kiekvienos kameros apdorojami ir pateikiami taip, kad atrodo, jog visi yra viename kambaryje. Galima niekur nekeliauti. Tam reikalingi labai didelė duomenų perdavimo sparta.

Nuo 1989 m. buvo realizuota daug bandomųjų projektų, kur duomenų perdavimo sparta siekia gigabitus (Aurora, Blanca, CASA, Nectar, VISTAnet)

TINKLŲ STANDARTIZAVIMAS

Telekomunikacijos

1886 m. daugelio Europos vyriausybių atstovai susitiko ir įkūrė šiuolaikinio ITU (International Telecommunication Union) pirmtaką. Tada buvo tik telegrafijos standartizavimas. Kai tik atsirado telefonas, į ITU buvo įtrauktas ir jo standartizavimas. 1947 m. ITU tapo Jungtinių Tautų agentūra.

Trys pagrindiniai ITU sektoriai:

1. Radijo komunikacijų sektorius (ITU-R).
2. Telekomunikacijų standartizacijos sektorius (ITU-T).
3. Vystymo sektorius (ITU-D).

ITU-R susijęs su radijo dažnių skirstymu pasaulyje.

ITU-T susijęs su telefonų ir duomenų perdavimo sistemomis. Nuo 1956 m. iki 1993 m. buvo vadinamas CCITT.

ITU-T užduotis - kurti technines rekomendacijas apie telefono, telegrafo ir duomenų perdavimo sąsajas. Jos dažnai tampa tarptautiniu mastu pripažintais standartais. De jure - tai yra tik rekomendacijos, nebūtinos vykdyti kiekvienoje šalyje. De facto - tai standartai. Paruošiama apie 5000 puslapių rekomendacijų per metus.

Tarptautiniai standartai

ISO (International Standards Organization), įkurta 1946 m. Jos nariais yra nacionalinės standartų organizacijos iš 89 šalių narių. Tai ANSI (U.S.), BSI (Didžioji Britanija), AFNOR (Prancūzija), DIN (Vokietija) ir kitos.

Yra sukurta apie 5000 standartų, įskaitant OSI. ISO turi apie 200 techninių komitetų, padalintų į pakomitečius, kurie padalinti į darbo grupes. Realus darbas daromas šiose grupėse (apie 100000 žmonių).

Vienas iš svarbiausių OSI narių - ANSI. ANSI standartai dažnai adaptuojami kaip ISO tarptautiniai standartai. Adaptavimo procedūra prasideda, kai viena iš nacionalinių standartų organizacijų jaučia tarptautinio standarto poreikį kokioje tai srityje. Suformuojama darbo grupė, kuri sukuria CD (Commitee Draft). Apie 6 mėn. Tą CD svarsto visi nariai ir kritikuoja. Jei dauguma pritaria, revizuotas dokumentas pavadinamas DIS (Draft International Standard) ir paliekamas komentavimui bei balsavimui. Rezultate

yra paruošiamas galutinis tekstas kaip IS (International Standard), aprobuojamas ir publikuojamas. Procesas gali užtrukti keletą metų.

TINKLŲ STANDARTIZAVIMAS

NIST (National Institute of Standards and Technology) yra U.S. komercijos departamento agentūra. Anksčiau buvo žinomas kaip Nacionalinis standartų biuras (National Bureau of Standards).

Dar vienas svarbus pasaulinių standartų institutas - IEEE (Institute of Electrical and Electronics Engineers). Tai - didžiausia profesionali organizacija pasaulyje. Gerai žinomas IEEE 802 standartas - yra vietinių tinklų pagrindas.

Internetas

Turi savo standartizavimo mechanizmą, stipriai besiskiriantį nuo ITU-T ir ISO.

Sukūrus ARPANET, buvo įsteigta IAB (Internet Architecture Board). Kai prireikdavo naujo standarto, IAB nariai nuodugniai jį apsvarstydavo ir paskelbdavo. Realizuodavo daugiausia studentai. Bendravimas vyko per techninių ataskaitų serija, vadinamą RFC (Request for Comments). RFC sunumeruoti chronologine tvarka ir yra visiems prieinami. Šiuo metų jų yra virš 2000.

Internetui išaugus (1989 m.), IAB buvo perorganizuotas. Atsirado IRTF (Internet Research Task Force) ir IETF (Internet Engineering Task Force).

IRTF užsiima ilgalaikiais tyrimais. IETF - trumpalaikiais inžinieriniais. Sukurtas panašus į ISO standartizavimo procesas. Prieš sukuriant standartą - siūlymą (Proposed Standard), jo pagrindinė idėja turi būti pilnai išaiškinta RFC ir ja turi būti pakankamai susidomėta. Juodraštinis standartas (Draft Standard) sukuriamas tik po to, kai veikianti realizacija yra testuota 4 mėn. ne mažiau kaip dviejose vietose. Jei viskas gerai, yra paskelbiama apie naują Interneto standartą.

KANALINIS LYGIS

Paslaugos, teikiamos tinkliniam lygiui

Paslaugos, siūlomos tinkliniam lygiui, gali būti įvairios ir priklauso nuo konkrečios sistemos. Trys dažniausiai naudojamos galimybės:

1. Paslauga be pristatymo patvirtinimo ir be sujungimo.
2. Paslauga su pristatymo patvirtinimu, bet be sujungimo.
3. Orientuota į sujungimą paslauga su pristatymo patvirtinimu.

Pirmas variantas - nepatikima paslauga. Iš vieno kompiuterio į kitą yra siunčiami nepriklausomi kadrai be jų pristatymo patvirtinimo. Jei kadras dingsta, kanaliniame lygyje nebandoma jo atstatyti. Ši paslaugų klasė yra tinkama tada, kai klaidų tikimybė yra maža ir jų taisymas paliekamas aukštesniems lygiams. Taip pat gali tikti duomenų perdavimui realiame laike (kalba, video), kai pavėlavę duomenys yra blogesni už klaidingus. Dauguma vietinių tinklų naudoja šį paslaugos tipą kanaliniame lygyje.

Sekantis žingsnis, gerinant perdavimo patikimumą, yra paslauga su pristatymo patvirtinimu, be sujungimų. Šiuo atveju siuntiklis žino, ar kadras iki gaviklio nukeliavo sėkmingai. Jei per tam tikrą laiką gavimo patvirtinimas neateina, kadras gali būti išsiųstas pakartotinai. Ši paslauga naudinga, naudojant nepatikimus kanalus, tokius kaip bevielės sistemos.

Gavimų patvirtinimai kanaliniame lygyje yra daugiau optimizacija negu būtinybė. Transportinis lygis gali pasiųsti pranešimą ir laukti jo gavimo patvirtinimo. Nesulaukus, pranešimą galima pakartoti. Blogai yra tai, kad pranešimas gali būti sudarytas pvz.: iš 10 kanalinio lygio kadrų, iš kurių tik vienas dingo. Geriau pakartoti jį, negu iš naujo siųsti pranešimą transportiniame lygyje.

Pati sudėtingiausia - orientuota į sujungimą paslauga su pristatymo patvirtinimu. Šiuo atveju tarp dviejų kompiuterių yra organizuojamas sujungimas prieš siunčiant duomenis. Kiekvienas išsiųstas kadras yra numeruojamas ir kanalinis lygis garantuoja, kad kiekvienas išsiųstas kadras yra gaunamas. Taip pat garantuojama, kad kiekvienas kadras bus gautas tik vieną kartą ir kad visi kadrai gaunami eilės tvarka.

KANALINIS LYGIS

Kai naudojamas susijungimas, perdavimą galima suskirstyti į tris fazes. Pirmojoje fazėje įvyksta susijungimas, kurio metu abi pusės inicializuoja kintamuosius ir skaitliukus, kad būtų galima sekti, kurie kadrai buvo priimti, o kurie ne. Antrojoje fazėje persiunčiamas vienas arba keli kadrai. Trečiojoje fazėje atsijungiama, atlaisvinami kintamieji, buferiai ir kiti resursai, naudoti sujungimo palaikymui

Kadrų ribų nustatymas

Kanalinis lygis suteikia paslaugas tinkliniam lygiui, naudodamasis fizinio lygio paslaugomis. Fizinis lygis perduoda ir priima bitų srautą be garantijos, kad nebus klaidų. Kanalinio lygio užduotis - surasti klaidas ir, jei reikia, jas ištaisyti. Paprastai tai yra daroma, suskaidant bitų srautą į atskirus kadrus ir suskaičiuojant kontrolinę sumą kiekvienam kadrui. Gaviklyje suma suskaičiuojama iš naujo ir, jeigu nesutampa su įrašyta kadre, laikoma, kad kadras blogas ir jis atmetamas.

Kadrų atribojimui naudojami šie metodai:

1. Simbolių skaičiavimas.
2. Pradžios ir galo ženklai su simbolių įterpimu.
3. Pradžios ir galo vėliavos su bitų įterpimu.
4. Fizinio lygio kodavimo pažeidimai.

Pirmuoju atveju kadro antraštėje yra laukas, kuriame nurodyta, kiek simbolių turi kadras. Pagal šio lauko reikšmę gaviklyje nustatoma, kur yra kadro pabaiga. Kai simbolių skaičius antraštėje sugadinamas perdavimo metu, praktiškai neįmanoma teisingai nustatyti kadro ribas. Dėl šios priežasties metodas beveik nenaudojamas.

Antruoju atveju kiekvienas kadras prasideda dviejų ASCII simbolių seka DLE STX, o baigiasi DLE, ETX. (DLE - Data Link Escape, STX - Start of Text, ETX - End of Text). Šiuo atveju, praradus kadro pradžią, užtenka surasti sekančią porą DLE STX arba DLE ETX.

Rimta problema, kai siunčiami dvejetainiai duomenys, o ne tekstas. DLE STX, DLE ETX būna duomenyse. Tada prieš kiekvieną DLE duomenyse yra įdedamas dar vienas DLE. Gaviklyje papildomas DLE išmetamas.

KANALINIS LYGIS

Aprašytas būdas blogas tuo, kad yra susietas su konkrečiu 8 bitų ASCII simbolių kodavimu ir galima duomenis perdavinėti tik baitais, o ne bitais. Dėl to vis dažniau yra naudojami bitiniai protokolai, kai kadre gali būti bet koks bitų skaičius ir ženklai gali būti koduojami bet kokiu bitų skaičiumi. Tai jau trečias atvejis.

Trečiu atveju kiekvienas kadras prasideda ir baigiasi specialiu bitų rinkiniu 01111110, vadinamu vėliava. Kai duomenyse pasitaiko 5 vienas po kito einantys vienetai, išsiunčiant yra įterpiamas 0. Gaviklyje po 5 vienetų esantis nulis yra išmetamas. Tokiu būdu duomenyse esanti vėliavos bitų kombinacija visada būna su įterptu nuliu po 5 vienetų ir niekada nesumaišoma su kadro riba.

Paskutinis būdas yra įmanomas tuose tinkluose, kur kodavimas fiziniame lygyje yra perteklinio pobūdžio. Kai kuriuose vietiniuose tinkluose vienas bitas yra koduojamas dviem fiziniais bitais pvz.: vienetas yra perėjimas iš aukšto į žemą lygį, o nulis - perėjimas iš žemo lygio į aukštą. Kombinacijos žemas-žemas ir aukštas-aukštas duomenims nenaudojamos. Šis neteisingų fizinių kodų panaudojimas yra IEEE 802 standarto vietiniams tinklams dalis.

Dauguma kanalinio lygio protokolų naudoja pirmojo metodo kombinaciją su kuriuo nors kitu didesnio patikimumo užtikrinimui. Ženklų skaičiaus laukas yra naudojamas kadro galo suradimui. Jeigu ten randamas reikalingas kadrų atskyriklis ir kontrolinė suma teisinga, kadras laikomas geru.

Srauto valdymas (flow control)

Problema, kai siuntiklis siunčia kadrus dažniau negu gaviklis gali priimti. Ji sprendžiama kanaliniame ir kituose lygiuose. Dėl spartos nesuderinimo gali būti prarandami be klaidų nukeliavę kadrai.

Srauto valdymo mechanizmas turi perspėti siuntiklį, kad jis siųstų tam tikra sparta, ne didesne už gaviklio spartą. Protokole turi būti griežtos taisyklės, nusakančios, kada siuntiklis gali siųsti kitą kadrą. Pvz.: kai susijungiama, gaviklis gali pasakyti, kad dabar galima išsiųsti *n* kadrų, o po to reikia laukti, kol bus duotas leidimas tolimesniam siuntimui.

KANALINIS LYGIS

Klaidų suradimas ir taisymas

Dvi pagrindinės strategijos, naudojamos apdorojant klaidas:

1. Kiekvieną siunčiamą duomenų bloką papildyti pertekline informacija, pagal kurią gaviklis galėtų nustatyti koks iš tikrųjų turi būti atsiųstas kiekvienas simbolis.
2. Kiekvieną siunčiamą duomenų bloką papildyti pertekline informacija, pagal kurią būtų galima nustatyti, kad siunčiant atsirado klaida ir bloką reikia siųsti iš naujo.

Pirmoji strategija naudoja klaidų taisymo kodus, antroji - klaidų suradimo kodus.

Klaidų korekcijos kodai, gali būti naudingi tada, kai ryšys vienkryptis ir gaviklis negali informuoti siuntiklį apie klaidingą duomenų bloką.

Duomenų perdavimo sistemose dažniausiai yra naudojami klaidų suradimo kodai ir bloko siuntimo pakartojimas. Klaidų suradimui reikia žymiai mažiau papildomų bitų, negu klaidų taisymui.

Atskirų (vienbičių) klaidų suradimui galima naudoti lyginumo bitą, pridedamą prie perduodamo duomenų bloko.

Praktikoje dažniausiai naudojami cikliniai pertekliniai kodai (CRC - Cyclic Redundancy Code) arba polinominiai kodai. Bitų eilutės traktuojamos kaip polinomo koeficientai. Pvz.: 110001 atitinka polinomą $x^5 + x^4 + x^0$. Skaičiuojant CRC, naudojami taip vadinami generaciniai polinomai *G(x)*. Jeigu *G(x)* laipsnis yra *k*, tai perduodamos bitų eilutės gale prirašoma *k*+1 nulių ir eilutę atitinkantis dvejetainis skaičius dalinamas pagal dviejų modulį iš dvejetainio skaičiaus, atitinkančio *G(x)*. Dalybos liekana yra CRC, kuri pakeičia papildomai prirašytus prie bitų eilutės nulius. Gaviklis priima bitų eilutę su CRC gale ir viską dalina iš skaičiaus, atitinkančio tą patį generacinį polinomą *G(x)*. Jeigu dalinasi be liekanos, tai reiškia, kad klaidos duomenyse nėra.

Trys polinomai yra tapę tarptautiniais standartais:

CRC - 12 $= x^{12} + x^{11} + x^3 + x^2 + x^1 + 1$

CRC - 16 $= x^{16} + x^{15} + x^2 + 1$

CRC - CCITT $= x^{16} + x^{12} + x^5 + 1$

KANALINIS LYGIS

CRC - 12 naudojama, kai simbolio ilgis 6 bitai. Kiti du naudojami 8 bitų ilgio simboliams. CRC - 16 arba CRC - CCITT suranda visas viengubas ir dvigubas klaidas, visas klaidas su nelyginiu bitų skaičiumi, visas 16 ir mažiau bitų ilgio klaidas, 99.997% 17 bitų klaidų ir 99.998% 18 ir daugiau bitų klaidų.

Kanalinio lygio protokolų pavyzdžiai

Neapribotas vienkryptis (simplex) protokolas. Pats paprasčiausias protokolas. Duomenys siunčiami tik viena kryptimi. Siuntiklio ir gaviklio tinkliniai lygiai visada pasiruošę darbui. Apdorojimo laikas ignoruojamas. Buferio dydis neribotas. Ryšio kanale kadrai nedingsta. Naudojamos dvi procedūros. Viena siuntiklyje, kita gaviklyje. Siuntiklis yra begaliniame cikle ir perduoda duomenis į liniją maksimaliu greičiu. Cikle tik trys veiksmai: paimti paketą iš tinklinio lygio, sukonstruoti kadrą, išsiųsti kadrą. Gaviklis toks pat paprastas. Atėjus kadrui, jis paimamas iš fizinio lygio, panaikinamos kadro pradžios ir galo žymės, antraštė, ir duomenų paketas perduodamas į tinklinį lygį. Laukiama sekančio kadro.

Vienkryptis protokolas "sustoti ir palaukti". Viskas tas pats, išskyrus tai, kad tinklinis lygis negali priiminėti duomenis begaline sparta. Jeigu reikia Δt laiko paimti kadrą iš fizinio lygio ir perduoti jį į tinklinį, tai siuntiklis per Δt neturi išsiųsti daugiau negu vieną kadrą. Galima apriboti siuntiklio darbo spartą, įvedant papildomą vėlinimą. Tačiau kanalinio lygio pasiruošimo priimti kadrą trukmė gali būti atsitiktinė, ypač tuo atveju, kai lygis dirba su keliomis linijomis vienu metu. Įvedus maksimalų galimą vėlinimą, kanalo panaudojimas bus neoptimalus.

Daug geresnis sprendimas - suorganizuoti grįžtamąjį ryšį. Po paketo perdavimo į tinklinį lygį, gaviklis išsiunčia siuntikliui trumpą kadrą, kuris yra leidimas siųsti sekantį duomenų kadrą. Tokie protokolai vadinami "sustok ir lauk". Čia duomenų perdavimas vienkryptis, tačiau kadrai keliauja į abi puses. Reikalingas nevienalaikis dvikryptis kanalas (half-duplex).

Vienkryptis protokolas kanalui su triukšmais. Kadrai gali būti pažeisti arba visiškai prarasti. Pažeisti kadrai atpažįstami, suskaičiuojant kontrolinę

sumą. Siuntiklis kažkiek laiko laukia ir išsiunčia kadrą dar kartą. Kartojama tol, kol kadras nueina nepažeistas. Šio protokolo trūkumas - kadro dublikato perdavimas į tinklinį lygį, jeigu patvirtinantysis kadras dingsta.

KANALINIS LYGIS

Vienas iš problemos sprendimo būdų - siunčiamojo kadro antraštėje patalpinti kadro numerį. Gaviklis pagal tą numerį nustato, ar nėra kadras pakartotas be reikalo. Numeracijai užtenka vieno bito. Svarbu, kad skirtųsi dviejų vienas po kito siunčiamų kadrų numeriai. Po nulinio kadro turi būti gautas pirmas. Jeigu vėl ateina nulinis, jis atmetamas. Gaviklis siunčia kiekvieno kadro gavimo patvirtinimą. Po kadro su numeriu "1" yra laukiamas kadras su numeriu "0" ir t.t.

Protokolai, kuriuose siuntiklis laukia patvirtinančiojo kadro, prieš išsiųsdamas sekantį, dažnai vadinami PAR (Positive Acknowledgement with Retransmission - teigiamas gavimo patvirtinimas su siuntimo pakartojimu). Arba ARQ (Automatic Repeat reQuest). Duomenys keliauja tik viena kryptimi. Siuntiklio laukimo trukmė turi būti pakankama. Kitaip gali būti išsiųstas dublikatas anksčiau laiko. Po to atėjęs patvirtinantysis kadras bus suprastas kaip dublikato gavimo patvirtinimas. Jeigu po to sekantis kadras nenueis iki gaviklio, tai tikrasis dublikato gavimo patvirtinimas bus suprastas kaip po jo ėjusio ir prarasto kadro gavimo patvirtinimas. Dingęs kadras nebus pakartotas. Šito galima išvengti, įdedant į patvirtinančiuosius kadrus numerius, tuo pasakant, kurio kadro gavimo patvirtinimas atkeliavo.

Siuntiklyje po kadro išsiuntimo startuoja taimeris ir atskaičiuoja tiek laiko, kiek reikia kadrui nukeliauti iki gaviklio, jį ten apdoroti ir nusiųsti patvirtinantįjį kadrą į siuntiklį. Tik šiam laikui pasibaigus, galima kartoti siuntimą.

Slankiojančiojo langopro tokolai (Sliding Window Protocols)

Tą pačią liniją galima panaudoti ne tik patvirtinančiųjų, bet ir duomenų kadrų perdavimui. Šiuo atveju gavimo patvirtinimus galima siųsti kartu su duomenimis, pažymint tai specialiame *ACK* lauke. Gavus duomenų kadrą, patvirtinantysis kadras nesiunčiamas tuojau pat, o laukiama, kol iš tinklinio lygio bus perduotas duomenų paketas. Gavimo patvirtinimas išsiunčiamas kartu su šiuo kadru. Tai vadinama piggybacking. Taip geriau panaudojamas kanalas, nes nereikia konstruoti specialaus patvirtinančiojo kadro su savo

antrašte, atskyrikliais ir kontroline suma. *ACK* laukas duomenų kadre užima nedaug vietos.

KANALINIS LYGIS

Šiam protokolui realizuoti reikalingas dar vienas skaitliukas, kuri nustato, kiek laiko galima laukti duomenų iš tinklinio lygio prieš išsiunčiant *ACK* kartu su duomenų kadru. Jeigu duomenys iš tinklinio lygio laiku nepasirodo, išsiunčiamas atskiras patvirtinantysis kadras. Čia vis dar gali būti problemų, susijusių su priešlaikiniu kadro pakartojimu, kai vėluoja patvirtinantysis kadras. Žymiai stabilesni tuo požiūriu yra slankiojančiojo lango protokolai.

Visuose slankiojančiojo lango protokoluose kiekvienas kadras turi numerį nuo 0 iki tam tikro maksimalaus 2^n - 1. Slankiojančiojo lango protokolas "sustok ir lauk" naudoja n = 1. Bendru atveju gali būti bet koks n.

Bet kuriuo laiko momentu yra tam tikra seka numerių, atitinkančių kadrus, kuriuos galima išsiųsti (siuntimo langas). Gaviklyje, atitinkamai, yra priėmimo langas - kadrų numeriai, kuriuos leidžiama gauti. Bendru atveju siuntimo ir priėmimo langai yra nevienodi ir netgi gali keistis, kai kadrai išsiunčiami arba priimami.

Numeriai siuntimo lange atitinka išsiųstus, bet dar be gavimo patvirtinimo kadrus. Kai ateina paketas iš tinklinio lygio, jam suteikiamas didžiausias numeris ir lango viršutinė riba padidėja vienetu. Kai ateina gavimo patvirtinimas, padidinama vienetu apatinė lango riba. Taigi, bet kuriuo momentu lange yra visi išsiųsti ir dar be gavimo patvirtinimo kadrai.

Kadangi kadrai, esantys siuntimo lange gali būti pažeisti arba dingti linijoje, tai visi jie turi būti saugomi atmintyje galimam siuntimo pakartojimui. Kai langas išauga iki maksimalaus dydžio, tinklinis lygis turi būti pristabdytas, nes visi buferiai būna užpildyti.

Priėmimo lange yra visų laukiamų kadrų numeriai. Bet koks kadras, išeinantis iš šių ribų, yra atmetamas be komentarų. Kai gaunamas kadras, kurio numeris lygus apatinei lango ribai, jis perduodamas į tinklinį lygį, išsiunčiamas gavimo patvirtinimas ir langas pasisuka per vieną poziciją. Gaviklio langas visada išlaiko savo pradinį dydį.

Vieno bito slankiojančiojo lango protokolas. Siuntiklis išsiunčia kadrą ir laukia jo pristatymo patvirtinimo prieš išsiunčiant kitą kadrą. Kadrų numeracijai naudojamas 1 bitas.

Normaliomis sąlygomis vienas iš dviejų kanalinių lygių pradeda siųsti pirmas. Paimamas paketas iš tinklinio lygio ir su nuliniu numeriu išsiunčiamas. Gaviklis pagal numerį patikrina, ar čia yra laukiamasis kadras. Jei tai ne dublikatas, jis perduodamas į tinklinį lygį. Gavimo langas persislenka per vienetą.

KANALINIS LYGIS

Gavimo patvirtinimo lauke yra paskutinio, gauto be klaidų, kadro numeris. Jei šis numeris atitinka prieš tai išsiųsto kadro, saugomo buferyje, numerį, siuntiklis gali paimti sekantį paketą iš tinklinio lygio į buferį ir jį išsiųsti. Jei gauto kadro patvirtinimo lauke numeris ne toks, tai kadras, esantis buferyje, išsiunčiamas dar kartą.

A siunčia kadrus į **B**, o **B** į **A**. Jeigu **A** laukimo laikas yra mažas, tai **A** gali pakartoti siuntimą keletą kartų (kadro numeris *seq* = 0, paskutinio be klaidų priimto kadro numeris *ack* = 1). **B** gauna pirmą kadrą (*seq* = 0, *ack* = 1, atsako kadru (*seq* =0, *ack* = 0) ir laukia kadro su *seq* = 1, *ack* = 0. Tačiau visi dublikatai turi *seq* = 0, *ack* = 1. Taigi, jie yra atmetami ir yra siunčiamas kadras *seq* = 0, *ack* = 0 (kartojamas nulinis **B** kadras su nulinio **A** kadro gavimo patvirtinimu). Taip yra tol, kol **A** gauna iš **B** *seq* = 0, *ack* = 0. Dublikatai nepatenka į tinklinį lygį, paketai neprarandami, amžinojo ciklo nėra. Protokolas veikia gerai.

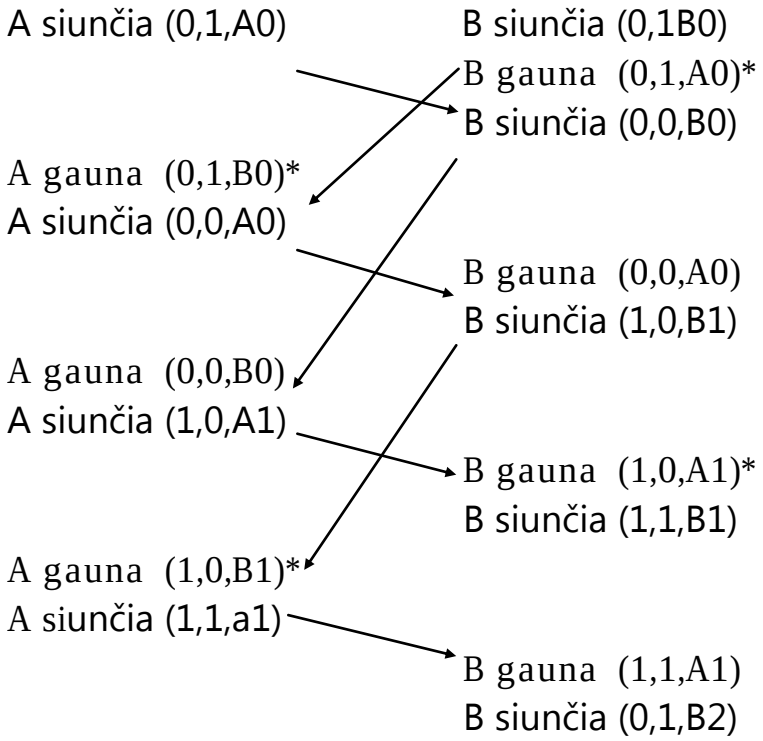
Kitas atvejis, kai tuo pačiu metu **A** siunčia kadrą į **B**, o **B** siunčia į **A**. Kiekvienas kadras yra bent vieną kartą pakartojamas, nors perdavimo klaidų nėra (10 pav.).

Grįžimas atgal per **n** ir selektyvusis pakartojimas. Kai kadro sklidimo laikas yra didelis (palydovinės sistemos), tai vieno bito slankiojančiojo lango protokolas labai neefektyviai išnaudoja kanalą. Ypač bloga yra tokia kombinacija: ilgas sklidimo laikas, plati dažnių juosta, trumpas paketas. Išeitis - leisti siuntikliui siųsti kadrus be pristatymo patvirtinimo tiek laiko, kiek reikia kadrui nukeliauti iki gaviklio ir atgal. Maksimalus siuntimo lango dydis lygus išsiųstų be patvirtinimo kadrų skaičiui.

Naudojant šį protokolą nepatikimiems kanalams, iškyla rimtų problemų. Pvz.: kai sugadinamas arba dingsta kadras perduodamų kadrų srauto viduje. Po jo einantys geri kadrai pasieks gaviklį greičiau, negu siuntiklis sužinos apie vieną sugadintą ir atmestą kadrą. Kaip užtikrinti perduodamų į tinklinį lygį kadrų eilės tvarką?

Vienas iš būdų, vadinamas "grįžti atgal per ***n***", atmeta visus po blogo einančius kadrus. Tai atitinka gavimo lango dydį lygų vienetui. Kanalinis lygis priima tik tą kadrą, kurį gali perduoti į tinklinį lygį eilės tvarka. Siuntiklis turi išsiųsti iš naujo visus kadrus, kurių gavimo patvirtinimo nesulaukė per tam tikrą laiką. Jeigu klaidos yra dažnos, kanalas išnaudojamas neefektyviai.

KANALINIS LYGIS



10 pav. Abi pusės pradeda siųsti vienu metu. Skliaustuose (*seq*,*ack*,kadro Nr.). Kadrai, perduoti į tinklinį lygį, pažymėti žvaigždute.

KANALINIS LYGIS

Kitas problemos sprendimo būdas vadinamas selektyviuoju pakartojimu. Čia visi geri kadrai, einantys po sugadinto, išsaugojami kanaliniame lygyje. Kai siuntiklis sužino, kad įvyko klaida, jis pakartotinai išsiunčia tik blogą kadrą. Gaviklio kanalinis lygis jį priima ir turi daug einančių eilės tvarka kadrų, kurie gali būti perduoti į tinklinį lygį. Po to išsiunčiamas kadro, turinčio didžiausią numerį, gavimo patvirtinimas.

Šis būdas atitinka gavimo langą, didesnį už 1. Kiekvienas kadras lango viduje gali būti priimtas ir laikomas buferyje tol, kol visi prieš jį einantys bus perduoti į tinklinį lygį. Tai reikalauja papildomos atminties.

Kiekvienam išsiųstam kadrui yra skaičiuojamas atskiras laukimo intervalas. Gavus *n* - tojo kadro gavimo patvirtinimą, laikoma, kad visi kadrai iki *n* - tojo taip pat sėkmingai nukeliavo adresatui.

Protokolas HDLC (High - level Data Link Control)

Visa grupė protokolų yra išvestiniai iš IBM SNA kanalinio lygio protokolo SDLC (Synchronous Data Link Control). ANSI modifikavo šį protokolą ir pavadino ADCCP (Advanced Data Communication Control Procedure). ISO taip pat modifikavo ir pavadino HDLC. CCITT adaptavo ir modifikavo HDLC savo protokolui LAP (Link Access Procedure), kuris yra X.25 dalis. Vėliau jį vėl modifikavo į LAPB, kad būtų suderinamas su vėlesne HDLC versija.

Visi šie protokolai remiasi tais pačiais principais. Visi jie yra bitiniai (bit-oriented). Kadro struktūra parodyta 12 pav.

Adreso laukas apibrėžia pirminę arba antrinę stotį, dalyvaujančią konkretaus kadro perdavime. Kiekviena stotis turi unikalų adresą. Nesubalansuotoje sistemoje adresų laukai komandose ir atsakymuose turi antrinės stoties adresą. Subalansuotose konfigūracijose komandinis kadras turi gaviklio adresą, o atsakymo kadre yra perduodančiosios stoties adresas. Linijose taškas-taškas adresas kartais naudojamas atskirti komandas nuo atsakymų.

KANALINIS LYGIS

Bitai

8	8	8	≥ 0	16	8
01111110	Adresas	Valdymas	Duomenys	Kontrolinė suma	01111110

12 pav. Bitinių protokolų kadro struktūra.

KANALINIS LYGIS

Valdymo laukas yra naudojamas eilės numeriams, gavimo patvirtinimams ir kitiems tikslams.

Duomenų lauke gali būti bet kokio ilgio informacija. Tačiau, didėjant kadro ilgiui, kontrolinės sumos efektyvumas mažėja. Kontrolinė suma skaičiuojama, naudojant CRC-CCITT generacinį polinomą.

HDLC vėliava taškas-taškas tipo linijose kartojama tol, kol linija neužimta.

Yra trys kadrų rūšys: informaciniai, valdantieji (Supervisory) ir nenumeruotieji (Unnumbered). Valdymo lauko turinys šiuose kadruose parodytas 13 pav. Protokolas naudoja slankiojantjjį langą su 3 bitų eilės numeriais. *Seq* laukas yra kadro eilės numeris. *Next* laukas yra naudojamas gavimo patvirtinimui. Čia įrašomas pirmojo negauto kadro numeris.

P/F reiškia Poll/Final (Apklausa/pabaiga). Naudojamas, kai kompiuteris (arba koncentratorius) apklausinėja grupę terminalų (stočių). P/F vadinamas P bitu, kai jį naudoja pirminė stotis ir F bitu, kai naudoja antrinė stotis. F=1 reiškia, kad atsakymo kadras yra paskutinis. P=1 reiškia, kad iš terminalo turi būti siunčiami duomenys.

Kai kuriuose protokoluose P/F bitas naudojamas priversti kitą kompiuterį siųsti valdymo kadrą (pvz.: gavimo patvirtinimą) tuoj pat, nelaukiant duomenų kadro.

Valdantieji kadrai atskiriami pagal tipo lauko reikšmę.

Tipas 0 yra patvirtinantysis kadras (RECEIVE READY - pasiruošęs priimti), parodantis, kad yra laukiamas sekantis kadras. Jis naudojamas, kai nėra atgalinio duomenų srauto gavimo patvirtinimams pernešti.

Tipas 1 yra gavimo nepatvirtinimas (REJECT - atmetimas). Parodo, kad buvo perdavimo klaida. *Next* laukas parodo pirmojo, neteisingai priimto, kadro numerį. Siuntiklis turi pakartotinai išsiųsti visus kadrus, pradedant nuo *Next*.

Tipas 2 yra RECEIVE NOT READY - nepasiruošęs priimti. Patvirtina visų kadrų iki *Next* gavimą, bet pasako siuntikliui, kad reikia nustoti siųsti. Kai problemos gaviklyje pašalinamos, jis pasiunčia RECEIVE READY, REJECT ar kokį kita valdantjjį kadrą.

KANALINIS LYGIS

Tipas 3 yra SELECTIVE REJECT - selektyvusis atmetimas. Jis reikalauja tik nurodyto kadro pakartojimo. Trečio tipo kadras nenumatytas SDLC ir LAPB. HDLC ir ADCCP jį turi.

Nenumeruotieji kadrai. Kartais naudojami valdymo tikslais. Taip pat gali būti panaudoti duomenų perdavimui nepatikimos be sujungimo paslaugos atveju. Čia bitiniai protokolai vienas nuo kito skiriasi stipriai. Kadro tipui skirti 5 bitai, tačiau ne visi 32 variantai yra naudojami.

Visi protokolai turi komandą DISC (DISConnect). Kompiuteris paskelbia, kad jis yra išjungiamas. Taip pat yra komanda SNRM (Set Normal Response Mode - pereiti į normalaus atsako modą). Tai leidžia tik ką įjungtai mašinai pranešti apie save. Čia yra nesubalansuota (asimetrinė) moda, kai viena pusė yra vedančioji, o kita pavaldi pirmajai. Terminalas praneša apie save valdančiam jį kompiuteriui. Kad protokolą būtų galima naudoti, kai abi pusės lygios, HDLC ir LAPB turi papildomą komandą SABM (Set Asynchronous Balanced Mode), kuri perveda liniją į pradinę būseną ir paskelbia abi puses lygiomis. Taip pat yra komandos SABME ir SNRME, kurios yra tokios pat kaip SABM ir SNRM, išskyrus tai, kad jos leidžia praplėstąjį kadro formatą, naudojantį 7 bitų eilės numerius.

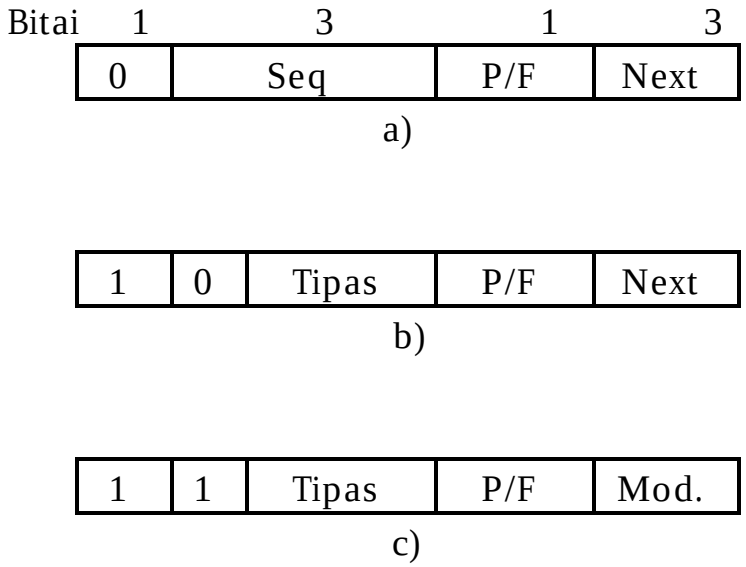
Trečioji komanda, palaikoma visų protokolų, yra FRMR (Frame Reject - kadras atmestas), naudojama parodyti, kad atėjo kadras su gera kontroline suma, bet nesuprantama semantika (prasme). Pvz.: 3-čiojo tipo valdantysis kadras protokole LAPB, kadras, trumpesnis už 32 bitus, nelegalus valdymo kadras, išeinančio iš lango ribų kadro gavimo patvirtinimas. FRMR kadrai turi 24 bitų duomenų lauką, pasakantį, kas buvo blogai kadre. Duomenyse įrašomas blogojo kadro valdantysis laukas, lango parametrai ir bitų rinkinys, nurodantis specifines klaidas.

Valdantieji kadrai gali būti prarasti arba pažeisti kaip ir duomenų kadrai. Taigi, jiems taip pat reikalingas gavimo patvirtinimas. Šiam tikslui yra skiriamas specialus valdantysis kadras UA (Unnumbered Acknowledgement - nenumeruotasis patvirtinimas).

Kiti valdantieji kadrai skirti inicializacijai, apklausai, būsenos aprašymui. Taip pat yra valdantysis kadras, kuriame gali būti bet kokia informacija. UI

(Unnumbered Information - nenumeruotoji informacija). Šie duomenys neperduodami į tinklinį lygį, o panaudojami gaviklio kanaliniame lygyje.

KANALINIS LYGIS



13 pav. Valdymo laukas informaciniame kadre (a), valdančiajame kadre (b), nenumeruotajame kadre (c).

KANALINIS LYGIS

Kanalinis lygis Internete

Dažniausiai naudojami taškas-taškas tipo sujungimai tarp maršrutizatorių, prie kurių gali būti prijungti vietiniai tinklai. Kitas atvejis, kai yra jungiamasi prie Interneto paslaugų tiekėjo per komutuojamą liniją ir modemą. Abiem atvejais reikalingas taškas-taškas kanalinis protokolas (kadrų ribų sukūrimas, klaidų aptikimas ir kt.). Internete yra naudojami du tokie protokolai. Tai SLIP ir PPP.

SLIP Serial Line Internet Protocol. Sukurtas 1984 m. Sun darbo stočių pajungimui prie Internet per telefono liniją, naudojant modemą. Protokolas, aprašytas RFC 1055, yra labai paprastas. Darbo stotis siunčia IP paketų srautą per liniją su specialiu vėliaviniu baitu paketo gale kadrų atskyrimui. Jei vėliava pasitaiko duomenyse, naudojamas vienas iš simbolių įterpimo metodų. Kai kuriose SLIP realizacijose vėliavinis baitas yra kiekvieno IP paketo pradžioje ir gale.

Naujesnės SLIP versijos naudoja TCP ir IP antraščių suspaudimą, kadangi dauguma vienas po kito einančių paketų turi vienodus laukus antraštėse. Persiunčiami ne pilni laukai, o tik jų reikšmių skirtumai. Šios optimizacijos aprašytos RFC 1144.

SLIP turi keletą rimtų trūkumų. Neieško ir netaiso klaidų. Palaiko tik IP paketus. Dėl to netinka tinklams, kurie IP nenaudoja (Novell NetWare ir kt.). Abi pusės turi iš anksto žinoti kitos IP adresą. Adresų trūksta, negalima kiekvienam kompiuteriui namuose duoti pastovų IP adresą. SLIP neatlieka jokio autentifikavimo. Tai nėra labai svarbu, kai naudojamos skirtinės linijos. Tačiau yra būtina, kai turime komutuojamą liniją su modemu. Galų gale, SLIP nėra patvirtintas Interneto standartu. Dėl to egzistuoja daug skirtingų ir nesuderinamų versijų.

PPP - Point-to-Point Protocol. Anksčiau minėtos problemos buvo išspręstos, sukūrus PPP, kuris tapo oficialiu Interneto standartu (RFC 1661, 1662, 1663). PPP detektuoja klaidas, palaiko daugelį protokolų, leidžia

dinamiškai nustatyti IP adresą, vykdo autentifikavimą ir turi daug kitų pagerinimų, lyginant su SLIP.

KANALINIS LYGIS

Trys svarbiausios PPP dalys:

1. Kadrų atskyrimas, vienareikšmiškai nurodant, kur yra vieno pabaiga ir kito pradžia. Klaidų suradimas.
2. Kanalo valdymo protokolas LCP (Link Control Protocol).
3. Tinklo valdymo protokolas NCP (Network Control Protocol), atskiras kiekvienam palaikomam tinklinio lygio protokolui.

PPP veikimo pavyzdys - namų kompiuterio įsijungimas į Internetą. Kompiuteris skambina paslaugų tiekėjo maršrutizatoriui per modemą. Po to, kai maršrutizatoriaus modemas atsako ir užmezgamas fizinis ryšys, kompiuteris siunčia maršrutizatoriui seriją LCP paketų viename arba keliuose PPP kadruose. Šių paketų ir atsakymų į juos pagalba parenkami reikalingi PPP parametrai. Toliau siunčiama serija NCP paketų tinklinio lygio sukonfigūravimui. Tipiniu atveju naudojamas TCP/IP protokolų stekas, taigi, reikalingas IP adresas. Kompiuteriui dinamiškai skiriamas IP adresas NCP pagalba. Nuo šio momento kompiuteris yra Internete ir gali siųsti IP paketus. Kai vartotojas baigia darbą, NCP pagalba IP adresas atlaisvinamas. Tada naudojamas LCP protokolas kanalinio lygio sujungimui nutraukti. Galų gale kompiuteris duoda komandą modemui, kad šis padėtų ragelį ir nutrauktų fizinio lygio sujungimą.

PPP kadro formatas artimas HDLC kadro formatui. Didžiausias skirtumas tarp jų yra tai, kad PPP - baitinis protokolas, o HDLC - bitinis. Komutuojamose modeminėse linijose PPP, panašiai kaip SLIP, naudoja simbolių įterpimo metodą. PPP kadrai gali būti siunčiami per SONET arba per bitines HDLC linijas. Kadro struktūra parodyta 14 pav.

KANALINIS LYGIS

Baitai 1	1	1	1 arba 2	Kintamas	2 arba 4	1
Vėliava 01111110	Adresas 11111111	Valdymas 00000011	Protokolas	Duomenys	Kontrolinė suma	Vėliava 01111110

14 pav. PPP kadro formatas (nenumeruotoji veikimo moda).

KANALINIS LYGIS

Adreso lauke visada 11111111, visos stotys gali priimti šį kadrą. Valdymo lauke pagal nutylėjimą yra 00000011. Tai reiškia nenumeruotąjį kadrą (nenaudoja eilės numerio ir gavimo patvirtinimo kadrų). Tačiau triukšminėse aplinkose gali būti panaudotas patikimas perdavimo būdas (Numbered Mode). Šio panaudojimo detalės aprašytos RFC 1663.

Kadangi pagal nutylėjimą adreso ir valdymo laukai yra fiksuoti, tai LCP leidžia abiem pusėms susitarti ir tų dviejų baitų visai nenaudoti. Kiekviename kadre sutaupoma po du baitus.

Protokolo lauke yra nurodoma, koks tinklinio lygio paketo tipas yra duomenyse. Yra nustatyti kodai LCP, NCP, IP, IPX, Apple Talk ir kitiems protokolams. Protokolai, kurių kodai prasideda nuliniu bitu, yra iš tinklinio lygio. Vienetu prasideda LCP ir NCP protokolų kodai. Pagal nutylėjimą protokolo laukas yra dviejų baitų, tačiau, abiem pusėms susitarus, gali būti tik vieno baito.

Duomenų laukas yra kintamo ilgio. Jei dėl maksimalaus ilgio nesitariama LCP pagalba linijos inicializavimo metu, tai naudojamas 1500 baitų maksimalus ilgis.

Kontrolinės sumos laukas yra 2 baitai. Gali būti susitarta ir naudojama 4 baitų kontrolinė suma.

Apibendrinant galima pasakyti, kad PPP yra daugiaprotokolinis kadrų sudarymo mechanizmas, naudojamas komutuojamose modeminėse linijose, HDLC bitinėse linijose, SONET'e ir kitose fizinio lygio realizacijose. Jis suranda klaidas, užtikrina susitarimo galimybę duomenų perdavimo parametrams nustatyti, vykdo antraštės suspaudimą ir, jei reikia, užtikrina patikimą perdavimą su kadrų numeracija ir gavimo patvirtinimais.

TINKLINIS LYGIS

Tinklinis lygis turi užtikrinti paketo pristatymą per daugelį potinklio mazgų. Tai galima padaryti, žinant viso potinklio topologiją ir parenkant reikalingą kelią potinklyje. Kelio parinkimas reikalingas ir tam, kad kuriame nors mazge neįvyktų tinklo perkrova, kai kiti tuo metu stovės be darbo. Galų gale, kai šaltinis ir tikslas yra skirtinguose tinkluose, tinklinis lygis turi išspręsti su tuo susijusias problemas.

Tinklinio lygio organizavimo principai

Paslaugos, teikiamos transportiniam lygiui. Tinklinio lygio paslaugos organizuojamos, vadovaujantis trimis pagrindiniais dalykais:

1. Paslaugos turi būti nepriklausomos nuo potinklio technologijos.
2. Transportinis lygis turi būti užekranuotas nuo egzistuojančių potinklių kiekio, tipo ir technologijos.
3. Tinklo adresai, prieinami tinkliniam lygiui, turi naudoti vieningą numeracijos būdą netgi vietiniuose ir globaliuose tinkluose.

Tinklinio lygio realizacijos, nepažeidžiančios šių principų, gali būti labai įvairios. Tačiau dažniausiai čia visos diskusijos susiveda į tai, ar tinklinis lygis turi teikti patikimas, orientuotas į sujungimus paslaugas, ar nepatikimas be sujungimų.

Internet atstovai, naudodamiesi 30 metų patirtimi eksploatuojant realiai veikiančius tinklus, teigia, kad potinklio darbas yra duomenų perdavimas ir tiek. Potinklis yra nepatikimas nepriklausomai nuo to, kaip jis padarytas. Taigi, kompiuteriai turi atlikti klaidų paiešką ir taisymą, taip pat ir srauto valdymą. Iš to seka, kad tinklinės paslaugos turi būti be sujungimų su primityvais SEND PACKET ir RECEIVE PACKET. Neturi būti paisoma paketų eilės tvarkos ir srauto valdymo, kadangi kompiuteriai vis vien tai atlieka. Paketai turi turėti pilną tikslo adresą, nes jie keliauja nepriklausomai vienas nuo kito.

Telefonų kompanijų atstovai, naudodamiesi 100 metų telefono sistemos eksploatavimo patirtimi, teigia, kad tinklinio lygio paslaugos turi būti

patikimos ir orientuotos į sujungimus. Sujungimai turi pasižymėti tokiomis savybėmis:

TINKLINIS LYGIS

1. Prieš siunčiant duomenis tinklinio lygio procesas turi susijungti su lygiaverčiu procesu gaviklyje. Sujungimui priskiriamas tam tikras identifikatorius. Sujungimas atlaisvinamas po to, kai visi duomenys išsiųsti.
2. Po susijungimo du procesai veda derybas dėl paslaugos parametrų, kokybės ir kainos.
3. Srauto valdymas yra vykdomas automatiškai.

Kitos savybės, tokios kaip garantuotas pristatymas, pranešimas apie pristatymą, didelio prioriteto suteikimas kai kuriems paketams, yra reikalingos bet nebūtinos.

Šie du požiūriai iš principo skiriasi tuo, kur turi būti sudėtingiausioji duomenų perdavimo dalis. Tinkliniame lygyje (potinklyje) ar transportiniame lygyje (kompiuteriuose). Internet naudoja nepatikimą be sujungimų tinklinį lygį. ATM naudoja patikimą, orientuotą į sujungimus tinklinį lygį. Kai ATM potinklis naudojamas IP paketų perdavimui, gaunasi nelabai efektyvus variantas, kadangi ATM tinklinis lygis garantuoja paketų pristatymą iš eilės, o virš IP lygio esantis TCP protokolas taip pat turi pilną paketų eilės tvarkos užtikrinimo mechanizmą. Detaliau apie IP virš ATM galima rasti RFC 1577.

Sujungimai, naudojami potinklyje, vadinami virtualiosiomis grandinėmis pagal analogiją su fizinėmis grandinėmis telefonų sistemoje. Nepriklausomi paketai, kai sujungimai nenaudojami, vadinami deitagramomis pagal analogiją su telegramomis.

Pirmu atveju paketų siuntimo kelias nustatomas susijungimo metu. Toliau visi paketai siunčiami tuo pačiu keliu.

Antru atveju jokie keliai iš anksto nenustatomi. Kiekvienas paketas yra nukreipiamas tam tikru keliu, nepriklausomai nuo kitų. Nors deitagramų potinkliai atlieka daugiau darbo, maršrutizuodami paketus, tačiau jie gali lengviau susidoroti su gedimais, lyginant su virtualiųjų grandinių potinkliais.

TINKLINIS LYGIS

E-mail	FTP	.....
TCP		
IP		
ATM		
Kanalinis lygis		
Fizinis lygis		

15 pav. TCP/IP protokolų stekas virš ATM potinklio.

TINKLINIS LYGIS

Pavadinimas	Deitagramų potinklis	Virtualiųjų grandinių potinklis
Grandinės inicializavimas	Nereikalinga	Reikalinga
Adresavimas	Kiekvienas paketas turi pilną šaltinio ir tikslo adresą	Kiekvienas paketas turi trumpą virtualiosios grandinės numerį
Būsenos informacija	Potinklis nelaiko būsenos informacijos	Kiekvienai virtualiajai grandinei reikia vietos potinklio lentelėse
Maršrutizavimas	Kiekvienas paketas maršrutizuojamas nepriklausomai	Kelias parenkamas virtualiosios grandinės sukūrimo metu. Visi paketai nukreipiami tuo pačiu keliu
Maršrutizatoriaus gedimas	Prarandami tik tie paketai, kurie sugedimo metu buvo maršrutizatoriuje. Kitų paketų tai neveikia	Visos virtualiosios grandinės, naudojančios maršrutizatorių, nutraukiamos
Tinklo perkrovų valdymas	Sudėtingas	Paprastas, jei kiekvienai virtualiajai grandinei gali būti iš anksto iškirta pakankamai vietos buferyje

16 pav. Deitagramų ir virtualiųjų grandinių potinklių palyginimas

TINKLINIS LYGIS

Maršrutizacijos algoritmai

Maršrutizacijos algoritmas yra tinklinio lygio programinės įrangos dalis, atsakinga už atėjusio paketo nukreipimą į reikiamą išėjimo liniją. Maršrutizavimas virtualiųjų grandinių potinklyje vadinama sesijiniu maršrutizavimu, kadangi paketų perdavimo kelias yra nustatomas visai vartotojo darbo sesijai.

Algoritmai gali būti sugrupuoti į dvi pagrindines klases: neadaptyvūs ir adaptyvūs. Neadaptyviųjų algoritmų atveju visi reikalingi keliai apskaičiuojami iš anksto ir persiunčiami į maršrutizatorius tinklo pakrovimo (inicializavimo) metu. Ši procedūra kartais vadinama statiniu maršrutizavimu.

Adaptyvūs algoritmai kelius nustato dinamiškai, įvertindami potinklio topologijos pokyčius ir duomenų srauto dydį (traffic) duotu momentu. Jie skiriasi pagal tai, kokiu būdu surenka informaciją, kada keičia maršrutizacijos lenteles ir koks parametras naudojamas optimizacijai (atstumas, perėjimų skaičius, numatoma perdavimo trukmė).

Trumpiausias kelias. Vienas iš trumpiausio kelio nustatymo būdų - suskaičiuoti tarpinių mazgų (perėjimų, šuolių) skaičių. Kitas kriterijus - geografinis atstumas kilometrais. Trumpiausias kelias gali būti nustatomas pagal testinio paketo laukimo eilėse trukmę ir perdavimo trukmę. Galimi ir kiti trumpiausio kelio nustatymo kriterijai arba jų kombinacija.

Užtvindymas (Flooding). Paketas perduodamas į visas išėjimo linijas. Taip vieno paketo kopijos gali užpildyti visą tinklą. Kadangi tokiu būdu yra išbandomi visi įmanomi variantai, tai realizuojasi pats geriausias iš jų. Gaviklis priima greičiausiai atėjusią paketo kopiją. Kad kopijų kiekis tinkle neišaugtų iki begalybės, apribojama jų gyvavimo trukmė arba nustatomas maksimalus perėjimų skaičius, kurį išnaudojus, paketas atmetamas. Kitas būdas, apribojantis paketo kopijų skaičių, yra padaugintų paketų numeracija. Jeigu ateina antra paketo kopija į maršrutizatorių, ji yra atmetama. Šiek tiek praktiškesnis yra selektyviojo užtvindymo būdas. Kiekvieną gautą paketą maršrutizatorius perduoda tik į tas išėjimo linijas, kurios apytikriai atitinka kryptį į tikslą.

Nors užliejimo algoritmas nėra praktiškas daugeliu atvejų, tačiau jis naudingas karinėse sistemose tais atvejais, kai daug maršrutizatorių sugenda tuo pačiu metu. Taip pat šis algoritmas gali būti naudojamas palyginimui su kitais algoritmais, nes iš visų galimų kelių išrenka patį optimaliausią.

TINKLINIS LYGIS

Maršrutizavimas, įvertinant tinklo topologiją ir apkrovą (Flow-Based Routing). Kai kuriuose tinkluose duomenų srautas tarp atskirų mazgų keičiasi nedaug ir vidutinė jo reikšmė yra žinoma iš anksto. Linijos pralaida taip pat yra žinoma. Tada, naudojant eilių teoriją, galima suskaičiuoti paketo vėlinimo trukmę šioje linijoje. Žinant visų linijų vėlinimo trukmes, galima suskaičiuoti vidutinę vėlinimo trukmę visame potinklyje. Tada maršrutizavimo problema suvedama į paiešką algoritmo, kuris minimizuoja minėtą trukmę. Visus galimus variantus galima suskaičiuoti ir pasirinkti tą, kuris duoda minimalų vidutinį vėlinimą potinklyje. Skaičiavimams gali prireikti nemažai laiko. Dėl to šis būdas tinka statinių maršrutizavimo lentelių sudarymui.

Maršrutizavimas pagal atstumo vektorių (Distance Vector Routing). Tai dinaminis maršrutizavimo algoritmas. Kiekvienas maršrutizatorius turi lentelę (t.y. vektorių), duodančią geriausią žinomą atstumą iki kiekvieno mazgo. Maršrutizatoriai periodiškai keičiasi informacija ir tokiu būdu papildo lenteles.

Atstumo vektoriaus algoritmas kartais vadinamas Bellman-Ford algoritmu arba Ford-Fulkerson algoritmu. Jis buvo pradinis maršrutizavimo algoritmas ARPANET'e. Buvo naudojamas Internete RIP pavadinimu, o taip pat pradinėse DECnet ir Novell IPX versijose. AppleTalk ir Cisco maršrutizatoriai naudoja patobulintus atstumo vektoriaus protokolus. Kaip atstumo kriterijus (metrika) gali būti tarpinių mazgų skaičius, vėlinimo trukmė, bendras eilėje stovinčių paketų skaičius ar kažkas panašaus.

Laikoma, kad maršrutizatorius žino "atstumą" iki kiekvieno iš savo kaimynų. Jeigu metrika yra vėlinimo trukmė, tai ji gali būti išmatuota specialiu ECHO paketu, kuriame gaviklis įrašo gavimo laiką ir išsiunčia atgal kaip galima greičiau.

Sudarant maršrutizacijos lentelę pasinaudojama informacija apie "atstumus" iki artimiausių kaimynų. Sudarytoje lentelėje yra "atstumas" iki kiekvieno maršrutizatoriaus ir linijos, kuri turi būti panaudota, numeris.

Atstumo vektoriaus algoritmas turi labai rimtą trūkumą - "skaičiavimo iki begalybės" problemą. Jeigu mazgas sugenda, tai "atstumas" iki jo laikomas lygiu begalybei. Paprastai reikia daug kaitos informacija tarp maršrutizatorių ciklų, kol visi sužino apie išjungtą mazgą. Tuo tarpu informacija apie mazgo įjungimą sklinda potinkliu žymiai greičiau. Per vieną kaitos ciklą pasislenkama

per vieną mazgą tolyn. Yra daug pasiūlymų, kaip spręsti šią problemą. Tačiau nei vienas iš jų neduoda visiško problemos sprendimo.

TINKLINIS LYGIS

Maršrutizavimas pagal kanalo būseną (Link State Routing). Nuo 1979 metų ARPANET naudoja šį algoritmą vietoj atstumo vektoriaus algoritmo.

Kanalo būsenos algoritmas susideda iš 5 dalių. Kiekvienas maršrutizatorius privalo:

1. Surasti savo kaimynus ir sužinoti jų tinklinius adresus.
2. Išmatuoti vėlinimo trukmę (kainą) iki kiekvieno kaimyno.
3. Sukonstruoti paketą, pasakantį visiems, ką nustatė.
4. Pasiųsti paketą visiems kitiems maršrutizatoriams.
5. Apskaičiuoti trumpiausią kelią iki kiekvieno maršrutizatoriaus.

Tokiu būdu eksperimentiškai nustatoma tinklo topologija, išmatuojamos vėlinimų trukmės ir visa tai perduodama visiems maršrutizatoriams.

Yra keletas šio algoritmo realizacijų. Labiausiai paplitęs yra OSPF (Open Short Path First). Kitas svarbus kanalo būsenos protokolas yra IS-IS (Intermediate System-Intermediate System) - tarpinė sistema-tarpinė sistema. Jis buvo sukurtas DECnet'ui ir vėliau adaptuotas ISO naudojimui su tinklinio lygio protokolu CLNP. Jis buvo modifikuotas, kad galėtų palaikyti kitus protokolus, pvz.: IP. IS-IS naudojamas daugelyje Internet magistralių (backbones) ir kai kuriose skaitmeninėse korinėse sistemose. Novell NetWare naudoja supaprastintą IS-IS variantą (NLSP) IPX paketų maršrutizavimui.

Hierarchinis maršrutizavimas. Didėjant tinklams, maršrutizacijos lentelės didėja proporcingai. Reikia vis daugiau laiko tų lentelių apdorojimui. Galų gale pasiekiamas taškas, kai neįmanoma į lentelę sudėti informaciją apie kiekvieną maršrutizatorių.

Maršrutizatoriai sudalinami į regionus, kurių viduje kiekvienas žino, kaip nukreipti paketus i bet kurį kitą. Tačiau maršrutizatoriai nieko nežino apie kitų regionų vidinę struktūrą. Dideliems tinklams dviejų lygių hierarchija gali būti nepakankama. Tada regionai grupuojami į klasterius, klasteriai į zonas, zonos į grupes ir t.t.

Tokiu atveju maršrutizacijos lentelės žymiai sutrumpėja. Jose būna informacija apie kiekvieną maršrutizatorių regiono viduje, o taip pat po eilutę kiekvienam kitam regionui. Nurodoma, i kokį vidinį maršrutizatorių persiųsti paketus, kad jie patektų į kita regioną. Dėl to gali šiek tiek pailgėti kelias,

perduodant paketus iš vieno regiono į kitą, tačiau tą pilnai kompensuoja maršrutizacijos lentelių sumažėjimas.

TINKLINIS LYGIS

Tinklo perkrovos

Kai tinkle yra perdaug paketų, jo darbo našumas krinta (17 pav.).

Keletas perkrovos priežasčių. Paketai iš keleto įėjimo linijų turi būti perduoti į vieną išėjimą. Sudaroma eilė. Jei pritrūksta eilei atminties, paketai prarandami. Atmintį galima praplėsti, tačiau netgi, esant begalinei atminčiai, situacija nepagerėja, nes labai išauga stovėjimo eilėje trukmė. Išsiunčiami paketų dublikatai, kurie dar labiau užkemša tinklą.

Lėti procesoriai. Ribota duomenų sparta linijose.

Yra skirtumas tarp perkrovų valdymo ir srauto valdymo. Perkrovos susijusios su viso potinklio veikla. Srauto valdymas susijęs tik su perdavimu tarp dviejų taškų per juos jungiančią liniją. Tinklas gali būti neperkrautas, o gaviklis tiesiog nespėja priiminėti duomenis iš siuntiklio. Kitas atvejis, kai kiekvienoje poroje gaviklis spėja priimti duomenis, tačiau tinkle yra daug kompiuterių, kurie siunčia daug mažesne sparta už maksimalią linijos spartą, tačiau tinklas yra perkrautas, nes nesugeba palaikyti tokio didelio paketų skaičiaus.

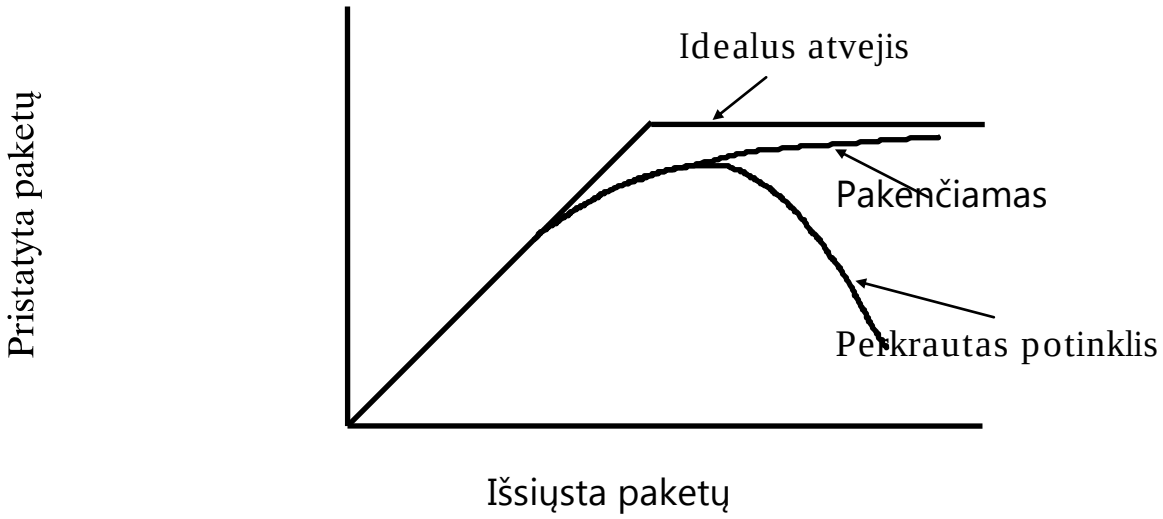
Tarptinkliniai ryšiai (Internetworking)

Tarptinkliniai ryšiai gali būti tokių tipų: LAN - LAN, LAN - WAN, WAN - WAN, LAN - WAN - LAN. Jiems realizuoti naudojami įrenginiai:

1. Kartotuvai.
2. Tiltai (kanalinio lygio kadrų perdavimas tarp VT).
3. Daugiaprotokoliai maršrutizatoriai. Perduoda paketus tarp skirtingų tinklų.
4. Transportiniai šliuzai. Perduoda duomenų srautus transportiniame lygyje.
5. Taikomieji šliuzai. Tarptinkliniai ryšiai virš ketvirtojo lygio.

Tuneliavimas - tai duomenų perdavimas iš to pačio tipo VT į kitą VT per WAN. VT paketas įdedamas į WAN voką ir perduodamas į kita VT.

TINKLINIS LYGIS



17 pav. Pristatytų paketų skaičiaus priklausomybė nuo išsiųstų paketų skaičiaus, esant skirtingam tinklo apkrovimui.

TINKLINIS LYGIS

Tinklinis lygis Internete

Internetą sudaro daug įvairių tinklų. Rišamoji medžiaga - IP (Internet protocol). Iš pat pradžių šis protokolas buvo orientuotas į tarptinklinius ryšius.

Transportinis lygis paima duomenų srautą, sukarpo jį į deitagramas. Teoriškai jų dydis iki 64 k baitų. Praktiškai dažniausiai būna apie 1500 baitų. Kiekviena deitagrama perduodama Internete. Galima jų fragmentacija. Tinkliniame lygyje jos surenkamos iš atskirų gabalų ir perduodamos į transportinį lygį, kuris, savo ruožtu, atiduoda duomenis atitinkamam taikomajam procesui.

IP protokolas

IP deitagrama sudaryta iš antraštės (18 pav) ir tekstinės dalies. Antraštė turi fiksuotą 20 baitų dalį ir kintamą nebūtinąją dalį.

Versijos laukas (4 bitai) leidžia perduoti deitagramas iš kompiuterių, naudojančių senesnes IP protokolo versijas.

IHL (4 bitai) - antraštės ilgis 32 bitų žodžiais. Mažiausia yra 5 žodžiai (20 baitų), daugiausia - 15 žodžių (60 baitų). Nebūtinoji dalis 40 baitų).

Paslaugos tipo laukas dažniausiai nenaudojamas. Susideda iš keleto laukų. Prioriteto laukas - 3 bitai. 0 - normalus paketas, 7 - tinklo valdymo paketas. Trys vėliaviniai bitai: D (Delay), T Throughput), R (Reliability). Taip nurodoma, kas yra svarbiausia - vėlinimas, pralaida, patikimumas. Maršrutizatorius pagal tai gali parinkti paketo perdavimo kelią. 2 bitai nenaudojami.

Bendras ilgis (2 baitai) - viso paketo ilgis (antraštė + duomenys).

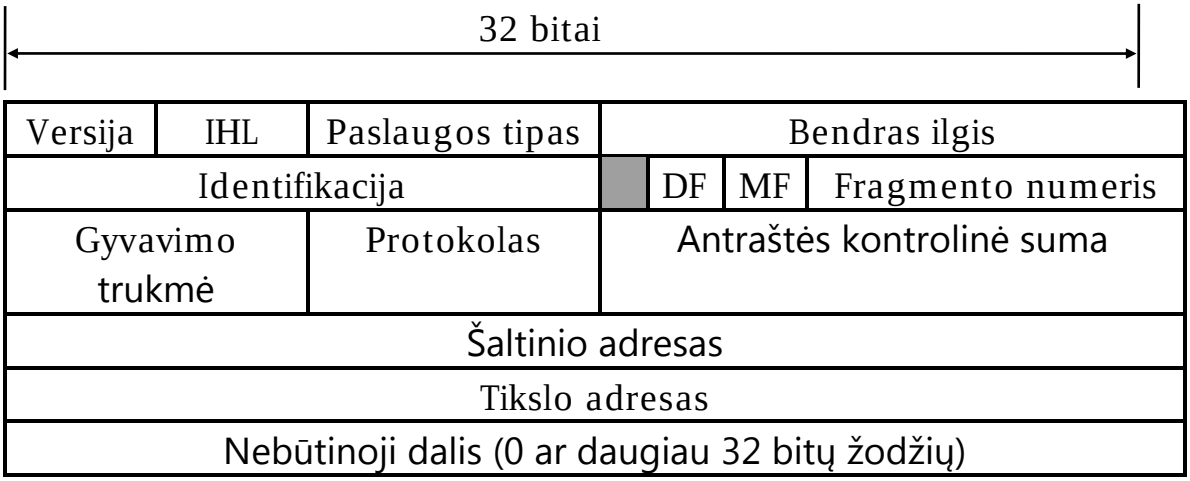
Identifikacijos laukas. Visi deitagramos fragmentai turi tą pačią identifikacijos reikšmę.

DF (1 bitas) Don't Fragment (nefragmentuoti). Kai gaviklis neturi galimybių surinkti fragmentuotas deitagramas.

MF (1 bitas) - More Fragments. Vienetas visuose fragmentuose, išskyrus paskutinį.

Fragmento numeris (Fragment offset). Visų fragmentų ilgiai, išskyrus paskutinio, turi būti kartotiniai 8 baitams. 13 bitų duoda 8192 fragmentus vienai deitagramai. Iš viso 8192 *8=65536 baitai. Vienu baitu daugiau už bendrąjį ilgį.

TINKLINIS LYGIS



18 pav. IP paketo antraštės struktūra

TINKLINIS LYGIS

Gyvavimo trukmė (1 baitas). Maksimali trukmė - 255 s. Gyvavimo trukmės reikšmė kiekviename maršrutizatoriuje sumažinama priklausomai nuo eilės dydžio. Realiai yra skaičiuojamas perėjimų skaičius. Kiekvieno perėjimo metu gyvavimo trukmė sumažinama vienetu. Kai tampa lygi nuliui, paketas atmetamas. Išsiunčiamas paketas, informuojantis siuntiklį apie paketo atmetimą.

Protokolo laukas nurodo, koks reikalingas aukštesnio lygio protokolas TCP, UDP ar kitas. Protokolų numeracija yra vienoda visame Internete, aprašyta RFC 1700.

Antraštės kontrolinė suma yra perskaičiuojama kiekviename maršrutizatoriuje, kadangi bent vieno lauko reikšmė pasikeičia (gyvavimo trukmė).

Adresų laukuose yra tinklo numeris ir kompiuterio tinkle numeris.

Nebūtinoji dalis (Options). Padaryta tam, kad naujesnės protokolo versijos galėtų patalpinti reikalingą informaciją. Eksperimentuojama, tikrinamos naujos idėjos. Susideda iš atskirų laukų, kurių ilgis yra keturių kartotinis. Kiekvienas laukas prasideda identifikaciniu baitu. Kai kurios opcijos turi vieno baito opcijos ilgio baitą ir po jo einančius duomenų baitus. Apibrėžtos penkios opcijos.

Saugos (Security) opcija pasako, kokio slaptumo perduodama informacija.

Tikslus kelias nuo šaltinio. Įrašomas visas deitagramos kelias kaip IP adresų seka. Deitagrama turi keliauti šiuo keliu. Naudinga sistemos administratoriams, siunčiant paketus, kai yra sugadintos maršrutizacijos lentelės arba kai reikia išmatuoti perdavimo trukmę.

Loose source routing reikalauja, kad paketas praeitų per nurodytus maršrutizatorius, tačiau leidžia praeiti ir per kitus maršrutizatorius.

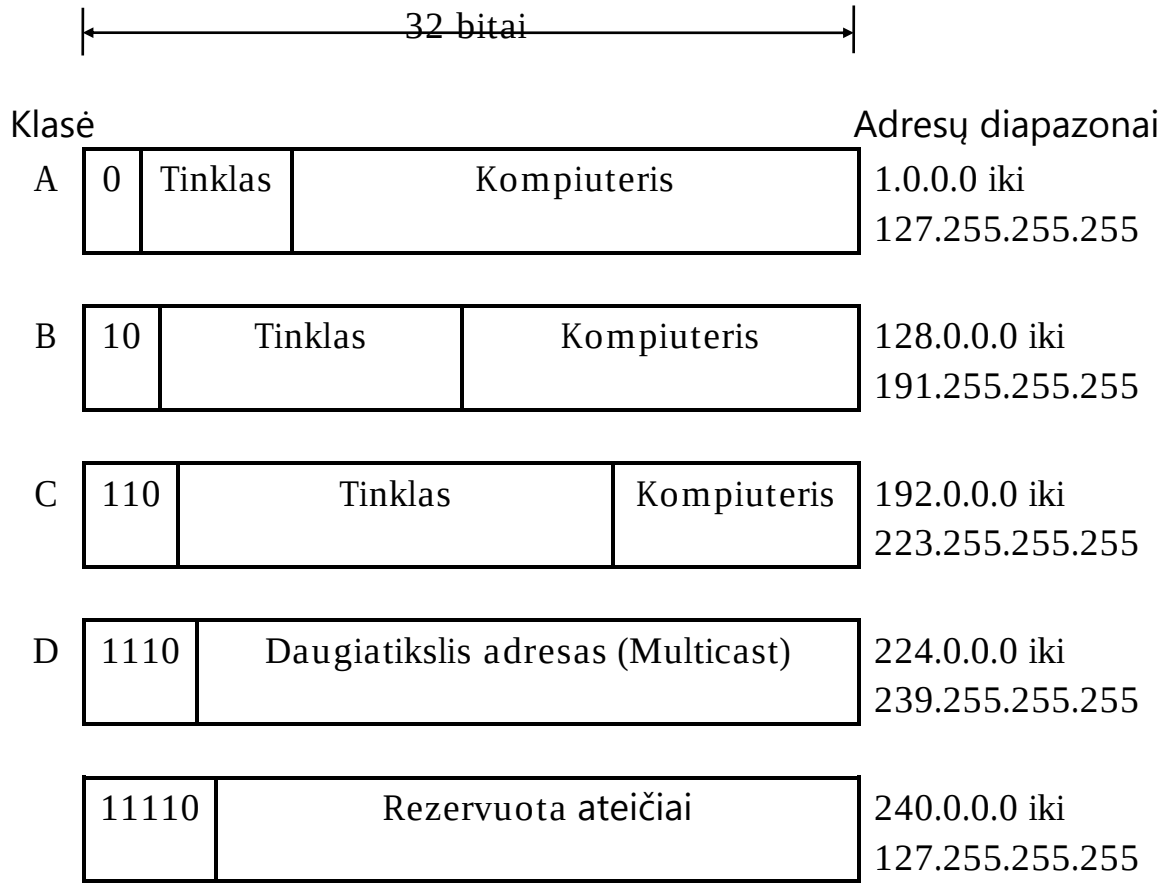
Įrašyti kelią. Nurodo maršrutizatoriams, kad jie įrašytų savo adresą į opcijos lauką. Tai reikalinga tinklų derinimui, maršrutizacijos klaidų taisymui ir t.t.

Laiko opcija. Kiekvienas maršrutizatorius įrašo ne tik savo IP adresą, bet ir laiką.

IP adresai

Kiekvienas kompiuteris ir maršrutizatorius Internete turi savo IP adresą, kuriame yra tinklo numeris ir kompiuterio numeris. Kompiuteriai, prijungti prie kelių tinklų, turi keletą IP adresų, atitinkančių kiekvieną iš tinklų. IP adreso struktūra parodyta 19 pav.

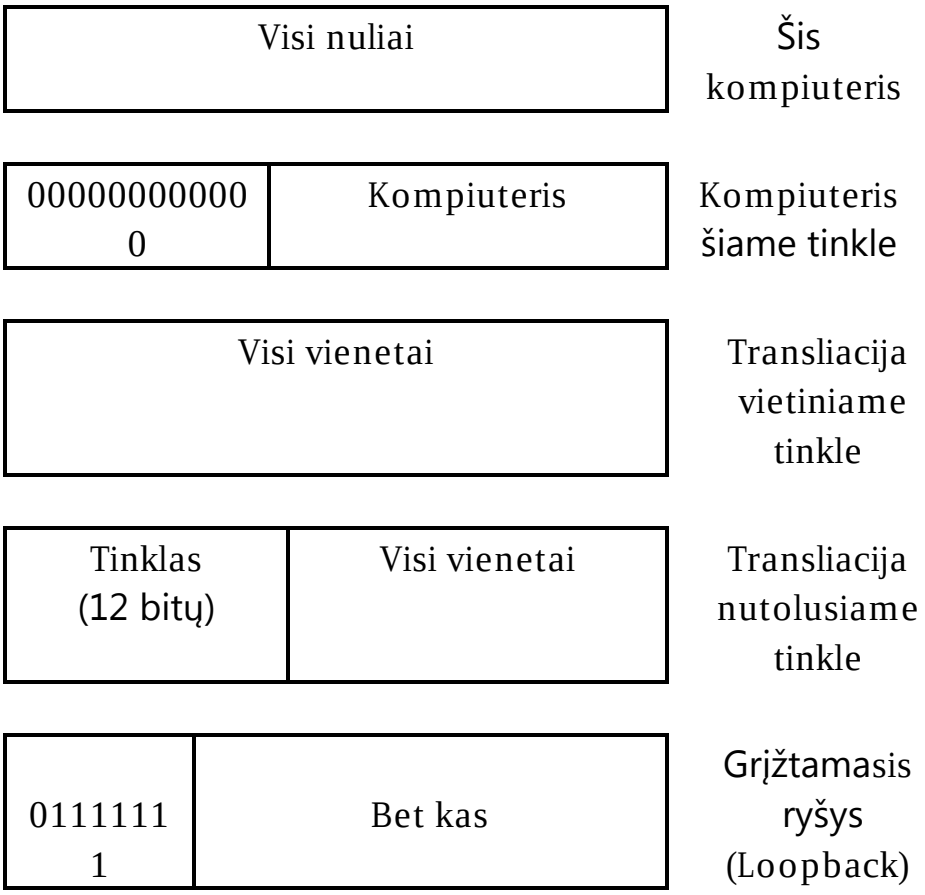
TINKLINIS LYGIS



19 pav. IP adresų formatas.

A - 126 tinklai su 16 milijonų kompiuterių kiekviename tinkle.
B - 16382 tinklai su 64k kompiuterių kiekviename tinkle.
C - 2 milijonai tinklų su 254 kompiuteriais kiekviename tinkle.

TINKLINIS LYGIS



20 pav. IP adresų vartojimo ypatumai.

TINKLINIS LYGIS

Tinklų numeriai skiriami tinklų informacijos centre NIC (Network Information Center).

Nuliai ir vienetai adrese turi specialią reikšmę (20 pav.). 0 - reiškia kompiuterį šiame tinkle. 1 - transliacinis adresas, reiškiantis visus kompiuterius nurodytame tinkle.

Adresą 0.0.0.0 naudoja kompiuteriai, kai kraunasi.

Adresų diapazonas 127.xx.yy.zz rezervuotas testavimui (Loopback testing). Paketai, išsiųsti šiuo adresu neišeina lauk. Jie apdorojami vietoje kaip priimti iš išorės paketai. Naudojama tinklų programinės įrangos derinimui.

Potinkliai

Suskirstymas į klases nėra tobulas. Išaugus tinklui, gali neužtekti C klasės adresų. Galima sukurti antrą tinklą ir gauti jam atskirą adresą, tinklus sujungti tiltu. Tačiau tai yra papildomos problemos.

Didelis vietinių tinklų kiekis taip pat ne išeitis. Kai tik naujas tinklas sukuriamas, sistemos administratorius turi susisiekti su NIC ir gauti naują tinklo numerį. Šis numeris turi būti paskelbtas visam pasauliui. Perkeliant kompiuterį iš vieno tinklo į kitą, reikia pakeisti jo IP adresą, pakeisti konfigūracinius failus, paskelbti apie adreso pakeitimą visiems. Jeigu kompiuteris gauna neseniai atlaisvintą IP adresą, tai kurį laiką jis gali gauti paštą ir duomenis, skirtus kitam kompiuteriui, kol adreso pakeitimas nenukeliauja per pasaulį.

Problemos sprendimas - padalinti tinklą į dalis vidiniam naudojimui. Šios dalys vadinamos potinkliais (subnets). Čia yra kita žodžio subnet reikšmė. Anksčiau jis reiškė maršrutizatorių ir perdavimo linijų visumą tinkle.

Pvz.: jei naudojamas B klasės adresas, tai, atsiradus naujam tinklui, galima 16 bitų kompiuterio numerio lauką padalyti į du. 6 bitai potinklio numeriui ir 10 bitų kompiuterio numeriui (21 pav.). Iš viso gali būti 62 vietiniai tinklai kaip potinkliai ir kiekviename iš jų po 1022 kompiuterius.

Tinklo išorėje potinkliai yra nematomi. Taigi, sukuriant naują potinklį, nereikia kreiptis į NIC.

TINKLINIS LYGIS

10	Tinklas	Potinklis (6 bitai)	Kompiuteris (10 bitų)

Potinklio kaukė

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0

21 pav. Vienas iš būdų, kaip sudalinti B klasės tinklą į potinklius.

TINKLINIS LYGIS

Maršrutizatorius *k* potinklyje žino, kaip nusiųsti paketą visiems kitiems potinkliams ir visiems kompiuteriams savo potinklyje. Jis nežino kitų potinklių struktūros detalių. Kiekvienas maršrutizatorius atlieka operaciją AND su adresu atėjusiame pakete ir potinklio kauke. Taip sužinoma, kokiam potinkliui paketas siunčiamas. Jei tai yra to paties potinklio paketas, jis perduodamas kompiuteriui. Jei yra skirtas kitam potinkliui, persiunčiamas į to potinklio maršrutizatorių. Pvz.: adresas pakete 130.50.15.6. Jis patenka į potinklį 5. Ten po AND operacijos gaunama 130.50.12.0. Tai reiškia, kad paketas skirtas 3 - čiam potinkliui.

Interneto valdymo protokolai

Interneto valdymo pranešimų protokolas. ICMP (Internet Control Message Protocol). Skirtas įvairių nenumatytų atvejų tinkle apdorojimui. Kiekvienas ICMP pranešimas yra įdėtas į IP paketą.

Tikslas nepasiekiamas (DESTINATION UNREACHABLE). Naudojamas, kai potinklis arba maršrutizatorius negali surasti kompiuterio arba paketas su DF bitu negali būti nusiųstas, nes pakeliui yra "mažų paketų" tinklas.

Viršytas laikas (TIME EXCEEDED) pasiunčiamas tada, kai pasibaigia paketo gyvavimo laikas. Tai gali reikšti paketo užsiciklinimą, tinklo perkrovą arba užduotą per mažą gyvavimo trukmę.

PARAMETER PROBLEM (Problema su parametrais). Antraštėje aptikta neleistina lauko reikšmė. Tai gali reikšti kompiuterio arba maršrutizatoriaus IP programinės įrangos klaidą.

SOURCE QUENCH (šaltinio "nuraminimas"). Siuntiklis, gavęs šį paketą, turi pamažinti siuntimo spartą. Naudojamas retai, nes tinklo perkrovos atveju dar labiau ją padidina. Perkrovų valdymas Internete yra atliekamas transportiniame lygyje.

REDIRECT (Nukreipimas, krypties pakeitimas). Išsiunčiama siuntikliui, kai maršrutizatorius nustato, kad paketas buvo nukreiptas neteisingu keliu.

ECHO REQUEST, ECHO REPLY naudojama nustatyti, ar kompiuteris yra pasiekiamas, ar jis veikia. Į ECHO REQUEST pranešimą gaviklis atsako ECHO REPLY.

TIMESTAMP REQUEST, TIMESTAMP REPLY. Į TIMESTAMP REQUEST užklausą išsiunčiamas atsakymas TIMESTAMP REPLY su įrašytu gavimo ir išsiuntimo laiku.

TINKLINIS LYGIS

Kiti keturi pranešimai susiję su adresavimu Internet'e. Leidžia kompiuteriams sužinoti savo tinklo numerį ir realizuoti situaciją, kai keli tinklai naudojasi vienu IP adresu. ICMP apibrėžtas RFC 792.

Adreso nustatymo protokolas (Address Resolution Protocol). Kanalinis lygis nesupranta IP adresų. Ethernet atveju šis lygis supranta 6 baitų tinklo kortos numerį, kuris yra unikalus.

IP adresai turi būti susieti su kanalinio lygio adresais (pvz.: Ethernet). Galima sudaryti lentelę. Tačiau jos palaikymas yra gana sudėtingas, kai tinkle tūkstančiai kompiuterių.

Geriau yra pasiųsti pranešimą visiems kompiuteriams vietiniame tinkle su klausimu, kam priklauso šis IP adresas. Šio klausimo išsiuntimas ir atsakymo gavimas yra vykdomas, naudojant ARP. Jį naudoja beveik kiekvienas kompiuteris Internete. Apibrėžtas RFC 826.

Galimi įvairūs ARP patobulinimai. Gautą ARP protokolu adresą kompiuteris gali laikinai saugoti savo atmintyje ir juo vėl pasinaudoti po kurio laiko. Jeigu į ARP paketą įdedamas klausiančiojo IP, tai kitas kompiuteris taip pat gali laikinai saugoti tą informaciją ir ja pasinaudoti po kurio laiko. Kadangi ARP paketas patenka į visus kompiuterius, tai visi jie gali išsaugoti siunčiančiojo IP ir kanalinio lygio adresus savo atmintyje.

Kiekviena mašina krovimosi metu išsiunčia visiems užklausą apie savo IP adresą. Jei atsakymo negauna, tai reiškia, kad joks kitas kompiuteris tinkle šio IP adreso nenaudoja. Priešingu atveju turi būti informuotas sistemos administratorius ir naujas kompiuteris nepakraunamas. Kaip šalutinis šio proceso efektas yra tai, kad pranešimą gauna visi kompiuteriai ir gali pakoreguoti savo laikinas lenteles. Paprastai įrašai tose lentelėse saugomi ne

daugiau kaip keletą minučių, kadangi vienas kompiuteris gali būti pakeistas kitu, kuris turi kitą tinklo kortos numerį.

Atgalinis adreso nustatymo protokolas (Reverse Adress Resolution Protocol). Kartais pagal tinklo kortos adresą reikia sužinoti IP adresą. To gali prireikti bediskio kompiuterio pasikrovimo metu iš nutolusio serverio. Naudojamas RARP (RFC 903). Pasikrovimo metu kompiuteris išsiunčia visiems pranešimą su savo tinklo kortos numeriu ir klausimu: "Koks yra šią kortą atitinkantis IP adresas?" RARP serveris gauna šį pranešimą, peržiūri konfigūracinius failus ir išsiunčia reikalingą IP adresą.

TINKLINIS LYGIS

RARP panaudojimas yra geriau, negu užkrauti sistemą su gatavu IP adresu. Pakrovimo procedūra ir pakraunama informacija į atmintį yra vienoda visiems kompiuteriams.

RARP trūkumas yra tai, kad jis naudoja visus vienetus tikslo adrese (apribota vietiniu tinklu transliacija) tam, kad pasiektų RARP serverį. Ši transliacija nepraeina per maršrutizatorius. Dėl to RARP serveris reikalingas kiekviename tinkle. Šią problemą apeina protokolas BOOTP (RFC 951, 1048, 1084). Jis naudoja UDP pranešimus, kurie praeina per maršrutizatorius. Bediskei stočiai yra suteikiama papildoma informacija, įskaitant failų serverio, turinčio pakrovimo failus, IP adresą, maršrutizatoriaus pagal nutylėjimą (default) IP adresą, potinklio kaukę.

Transliacija Internete (Internet Multicasting)

Vienas siuntiklis - daug gaviklių. IP palaiko transliaciją, naudodamas D klasės adresus. Kiekvienas D klasės adresas identifikuoja kompiuterių grupę. 28 bitai skirti grupių identifikavimui - virš 250 milijonų grupių gali egzistuoti tuo pačiu metu. Yra palaikomos dvi grupinių adresų rūšys: pastovūs adresai ir laikini adresai. Keletas pastovių grupių adresų:

224.0.0.1 - visos sistemos vietiniame tinkle;
224.0.0.2 - visi maršrutizatoriai vietiniame tinkle;
224.0.0.5 - visi OSPF maršrutizatoriai vietiniame tinkle;

Laikinos grupės turi būti sukurtos prieš jas panaudojant. Procesas gali paprašyti savo kompiuterio prisijungti prie specifinės grupės. Taip pat gali paprašyti palikti grupę. Kai paskutinis procesas kompiuteryje išeina iš grupės, ši grupė kompiuteryje nustoja egzistuoti. Kiekvienas kompiuteris saugo informaciją apie tai, kokiai grupei jo procesai duotu momentu priklauso.

Transliacija realizuojama specialiais transliaciniais maršrutizatoriais, kurie gali būti kartu su paprastu maršrutizatoriumi. Maždaug vieną kartą per minutę kiekvienas toks maršrutizatorius pasiunčia kanalinio lygio pranešimą

visiems kompiuteriams vietiniame tinkle (224.0.0.1), prašydamas pranešti apie visas grupes, kurioms priklauso procesai. Kiekvienas kompiuteris pasiunčia atsakymą apie visus jį dominančius D klasės adresus.

TINKLINIS LYGIS

Šie užklausos ir atsakymo paketai naudoja IGMP (Internet Group Management Protocol - Interneto grupių valdymo protokolas) protokolą. Naudojami tik du paketų tipai: užklausa ir atsakymas. Kiekvienas iš jų turi paprastą fiksuotą formatą, turintį tam tikrą valdančią informaciją pirmajame duomenų lauko žodyje ir D klasės adresą antrajame. Aprašytas RFC 1112.

Transliacija vykdoma, naudojant gaubiančiuosius medžius. Kiekvienas transliacinis maršrutizatorius keičiasi informacija su savo kaimynais, naudodamas modifikuotą atstumo vektoriaus protokolą. Kiekvienam maršrutizatoriui sukonstruojamas gaubiantysis medis, apimantis visus grupės narius.

CIDR -Classless Inter Domain Routing (beklasis tarpdomeninis maršrutizavimas)

IP adresų pradeda neužtekti. 1987 keletas įžvalgių žmonių numatė, kad Internetas išaugs iki 100000 tinklų. Daugelis ekspertų juos išjuokė. 100000-tasis tinklas buvo prijungtas 1996 metais. Iš principo egzistuoja virš 2 milijardų adresų, bet dėl jų suskirstymo į klases milijonai adresų prarandami. Daugeliui organizacijų A klasės tinklas per didelis. C klasės per mažas. Tačiau praktiškai ir B klasės adresai yra per daug dideli daugeliui organizacijų (65536 kompiuteriai tinkle). Tyrimai parodė, kad daugiau kaip pusė B klasės tinklų turi mažiau negu 50 kompiuterių.

Kita problema yra maršrutizacijos lentelių katastrofiškas didėjimas. Maršrutizatoriams IP adresų erdvė yra dviejų lygių hierarchija su tinklų numeriais ir kompiuterių numeriais. Pusė milijono C klasės tinklų duoda atitinkamo dydžio lenteles, kurias ir apdoroti ir perduoti yra labai sudėtinga.

Vienas iš sprendimo būdų yra CIDR (RFC 1519). Pagrindinė idėja - likusius beveik 2 milijonus C klasės adresų padalinti į kintamo dydžio blokus. Pvz.: jeigu reikia 2000 adresų, tai yra duodamas 2048 adresų blokas (8 C klasės tinklai), bet ne pilnas B klasės adresas.

Pasaulis buvo padalintas į keturias zonas. Kiekvienai iš jų skirta C klasės adresų erdvės dalis:

194.0.0.0 - 195.255.255.255 - Europa;
198.0.0.0 - 199.255.255.255 - Šiaurės Amerika;
200.0.0.0 - 201.255.255.255 - Centrinė ir Pietų Amerika;
202.0.0.0- 203.255.255.255 - Azija ir Ramusis vandenynas.

TINKLINIS LYGIS

Kiekvienam regionui skirta apie 32 milijonus adresų. Likusieji 230 milijonų C klasės adresų nuo 204.0.0.0 iki 223.255.255.255 rezervuoti ateičiai. Rezultate 32 milijonai adresų dabar yra suspausti į vieną įrašą maršrutizacijos lentelėje. Regiono ribose adresai paskirstomi į grupes. Del to maršrutizacijos lentelė regiono viduje taip pat sumažėja. Kiekviena adresų grupė turi savo grupinę kaukę (mask). Adreso ir kaukės AND duoda grupės numerį. Lentelėje kiekvienai grupei skiriamas vienas įrašas.

Ta pati idėja gali būti panaudota visiems adresams. Senieji A, B, C tinklai daugiau nenaudojami maršrutizavimui. Dėl to vadinama beklasiu maršrutizavimu.

IPv6

CIDR šiek tiek atitolina problemą, tačiau jos nesprendžia iki galo. Senojo IP (IPv4) dienos suskaičiuotos. Dėl to IETF 1990 m. pradėjo kurti naują IP versiją, kurią naudojant adresų niekada nepritruks. Pagrindiniai naujos versijos bruožai:

1. Palaiko milijardus kompiuterių net jeigu adresų erdvė naudojama neefektyviai.
2. Sumažina maršrutizacijos lenteles.
3. Supaprastina protokolą, maršrutizatoriai gali greičiau apdoroti paketus.
4. Užtikrina didesnį duomenų saugumą.
5. Daugiau kreipia dėmesio į paslaugos tipą, dalinai realaus laiko duomenims.

6. Transliacija, leidžianti nustatyti diapazonus
7. Galimybė keliauti su kompiuteriu, išliekant Internete ir nekeičiant IP adreso.
8. Leidžia protokolui tobulėti (evolve) ateityje.
9. Leidžia senam ir naujam protokolui egzistuoti daug metų.

1992 m. buvo atrinkta keletas rimčiausių pasiūlymų. Iš jų Deering ir Francis pasiūlymas vadinamas SIPP (Simple Internet Protocol Plus). Dabar jam priskirtas ženklas IPv6. Nesuderinamas su IPv4, bet suderinamas su TCP, UDP, ICMP, IGMP, OSPF, BGP, ir DNS. Reikalingos nedidelės modifikacijos, susijusios su adresavimu. Daugiau informacijos yra RFC 1883 - 1887.

TINKLINIS LYGIS

Pirmiausia - 16 baitų adresai. Antra - antraštės supaprastinimas. Tik 7 laukai (vietoj 13-kos IPv4). Tai leidžia maršrutizatoriams greičiau apdoroti paketus ir padidinti pralaidą. Trečia - geriau palaikomos opcijos. Maršrutizatoriai greičiau jas apdoroja arba praleidžia, kai reikia. Ketvirta - padidintas informacijos slaptumas. Autentifikavimas ir privatumas - vieni iš svarbiausių naujame IP. Penkta - daugiau dėmesio kreipiama į paslaugos tipą.

Palaikomos deitagramos, didesnės už 65535 baitus. Vadinamos jumbogramomis. Jų ilgiui skirti 4 baitai. Svarbu superkompiuteriams gigabitinių duomenų kiekių perdavimui Internetu.

TRANSPORTINIS LYGIS

Paslaugos, teikiamos aukštesniesiems lygiams

Pagrindinis tikslas - teikti efektyvias, patikimas ir nebrangias paslaugas vartotojams (procesams taikomajame lygyje). Tam panaudojamos tinklinio lygio paslaugos. Aparatinės/programinės priemonės transportinio lygio viduje vadinamos transporto esybe (transport entity). Tai gali būti realizuota operacinės sistemos branduolyje, atskirame vartotojo procese, tinklinių taikymų bibliotekose arba tinklo kortoje. Atskirais atvejais transportinio lygio organizavimui gali būti naudojamos specialios interfeisinės mašinos potinklio krašte. Prie jų jungiami kompiuteriai.

Kaip ir tinkliniame lygyje, yra orientuotos į sujungimus ir be sujungimų transportinės paslaugos. Vartotojai nevaldo potinklio, kur veikia tinklinio lygio protokolai. Paslaugos kokybės pagerinimui reikalingas lygis virš tinklinio. Tai ypač svarbu, kai tinklinis lygis nepatikimas ir be sujungimų.

Transportinio lygio dėka taikomosios programos gali būti parašytos, naudojant standartinį primityvų rinkinį, ir gali dirbti daugybėje tinklų, nesirūpindamos apie skirtingas potinklių sąsajas ir nepatikimą perdavimą.

Kartais pirmieji 4 lygiai yra vadinami transporto paslaugos tiekėju. Aukštesnieji lygiai - transporto paslaugos vartotojai. Transportinis lygis yra skiriamoji riba tarp patikimo duomenų perdavimo tiekėjo ir vartotojo.

Paslaugos kokybė. Galimi parametrai:

1. Sujungimo įvykdymo vėlinimas (Connection establishement delay).
2. Nesusijungimo tikimybė (Connection establishement failure probability).
3. Pralaida. Vartotojo baitų kiekis, perduodamas viena kryptimi per tam tikrą laiką.
4. Perdavimo trukmė (Transit delay). Laiko tarpas tarp pranešimo išsiuntimo vienoje pusėje ir priėmimo kitoje.
5. Liekamasis klaidų santykis (Residual error ratio). Prarastų ir išsiųstų pranešimų santykis.
6. Apsauga (Protection). Nuo neautorizuoto duomenų skaitymo arba keitimo.

7. Prioritetas (Priority). Leidžia transporto vartotojui nurodyti, kad vienas iš jo naudojamų sujungimų yra svarbesnis už kitus.
8. Resilience. Tikimybė, kad transportinis lygis pats panaikins sujungimą dėl savo vidinių problemų.

TRANSPORTINIS LYGIS

QoS (Quality of Service) parametrai yra nustatomi sujungimo užklausos metu. Nurodomos reikalingos ir minimalios priimtinos reikšmės. Sujungimo užklausa gali būti atmesta, jei kokio nors parametro reikšmės negalima užtikrinti. Dėl konkrečių paslaugos kokybės parametrų reikšmių susitariama derybų dėl opcijų (Option negotiation) metu.

Transporto protokolų elementai

Daug kuo transporto protokolai panašūs į kanalinio lygio protokolus. Užsiima klaidų kontrole, eilės tvarka, srauto valdymu ir kt. Pagrindiniai skirtumai - aplinka, kurioje protokolai veikia. Kanaliniame lygyje du maršrutizatoriai keičiasi duomenimis per fizinį kanalą. Transportinio lygio atveju šis kanalas pakeistas visu potinkliu. Kanaliniame lygyje kiekviena linija jungia tik du maršrutizatorius, adresai neturi didelės prasmės. Transportiniame lygyje reikalingas pilnas adresas. Transportinio sujungimo organizavimas yra labiau komplikuotas negu kanalinio lygio.

Kitas svarbus skirtumas tarp kanalinio ir transportinio lygio yra tai, kad paketas potinklyje gali būti užlaikytas ir pristatytas žymiai vėliau. Taip pat gali atsirasti nereikalingi paketų dublikatai. Tai gali pareikalauti specialių protokolų.

Buferizavimas ir srauto valdymas reikalingi abiejuose lygiuose, tačiau didelis, dinamiškai besikeičiantis sujungimų skaičius transportiniame lygyje reikalauja kitokių būdų kaip tai realizuoti.

Adresavimas. Taikomasis procesas, norėdamas bendrauti su procesu kitoje pusėje, turi nurodyti to proceso adresą. Reikalingas transportinio lygio adresas. Bendras pavadinimas TSAP (Transport Service Access Point). Internete - tai IP adreso ir vietinės prieigos (porto) pora. ATM tinkluose - tai AAL-SAP.

UNIX'iniai kompiuteriai Internete naudoja pradinį sujungimo protokolą (initial connection protocol). Kiekvienas kompiuteris, kuris teikia paslaugas nutolusiems kompiuteriams, turi specialų procesų serverį, kuris veikia kaip proxy mažiau apkrautiems naudojamiems serveriams. Jis "klauso" visą eilę

prieigų tuo pačiu metu, laukdamas TCP susijungimo užklausos. Potencialūs vartotojai savo užklausoje nurodo reikalingą TSAP adresą (TCP prieigą). Jeigu tuo adresu joks serveris neveikia, tai jie sujungiami su procesų serveriu, kuris pažadina reikalingą serverį ir perduoda jam sujungimą su vartotoju.

TRANSPORTINIS LYGIS

Tačiau kai kurios paslaugos egzistuoja nepriklausomai nuo procesų serverio ir nėra kuriamos tuo momentu, kai jų prireikia. Dėl to dažniau naudojama kita schema. Čia egzistuoja specialus procesas, vadinamas vardų serveriu arba katalogų serveriu. Vartotojas susijungia su vardų serveriu, kuris "klauso" gerai žinomą prieigą, nusiunčia pranešimą su paslaugos vardu, o vardų serveris nusiunčia TSAP adresą. Tada vartotojas atsijungia nuo vardų serverio ir susijungia su reikalingu serveriu.

Susijungimas. Atsiranda problemos, kai tinklas praranda, sukaupia (store) ir dubliuoja paketus.

Potinklis gali būti perkrautas ir patvirtinimai ateina pavėluotai. Tada kiekvienas paketas išsiunčiamas pakartotinai. Nesiimant specialių priemonių, tai gali būti suprasta kaip naujo sujungimo užklausa. Jeigu tai yra pinigų pervedimas iš banko, tai jis bus pakartotas keletą kartų be šeimininko žinios.

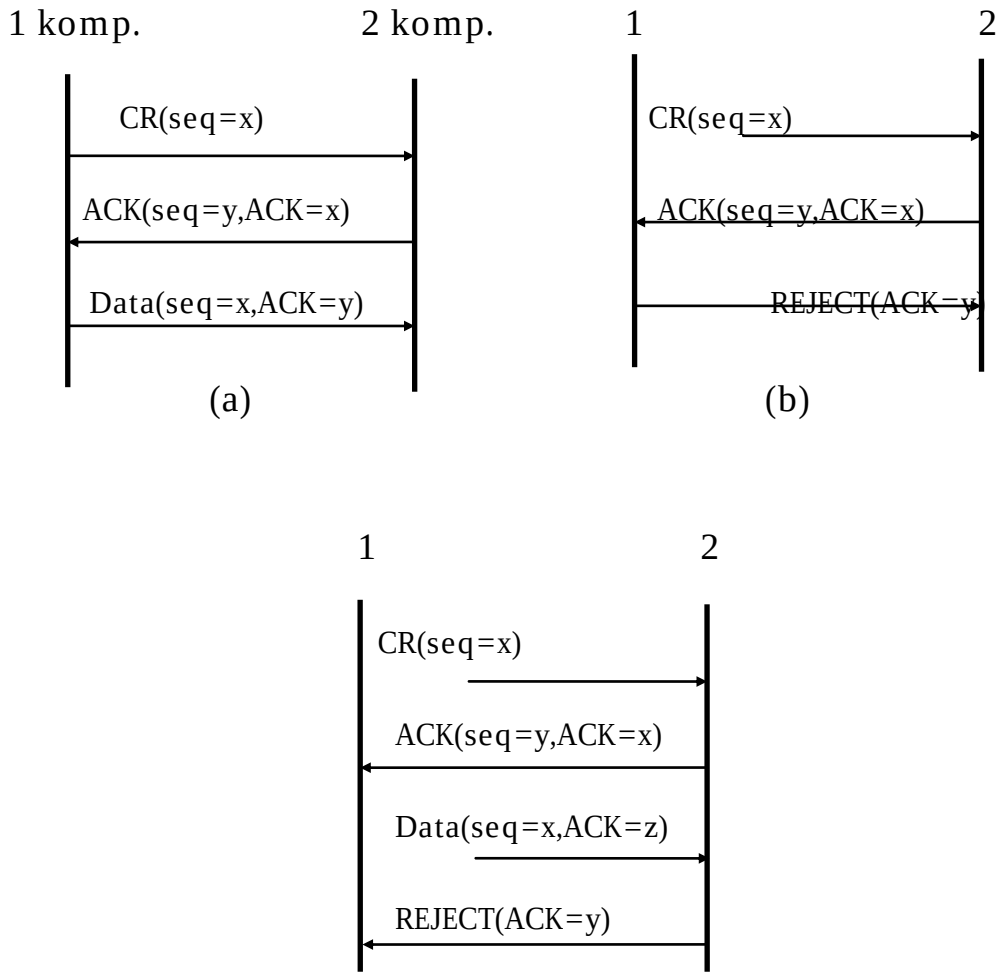
Geriausias būdas šito išvengti - apriboti paketų gyvavimo trukmę tinkle. Praktiškai reikia garantuoti ne tik tai, kad paketo gyvavimo laikas pasibaigė, bet kad baigėsi ir visų jo patvirtinančiųjų paketų gyvavimo laikas. Paketai numeruojami ir skaičiuojamas laikas nuo jų išsiuntimo. Skaitliukai turi veikti net ir tada, kai kompiuteris "nulūžta" ir paskui vėl pakraunamas iš naujo. Paprastai po pakrovimo atskaičiuojamas pilnas laiko tarpas τ, per kurį visi nereikalingi paketai turi išnykti tinkle. Tačiau sudėtingame tinkle šis τ gali būti labai didelis.

Sujungimo organizavimui naudojamas Tomlinson (1975) trijų etapų "rankos paspaudimas" (three-way handshake) (22 pav.).

Sujungimo atlaisvinimas. Asimetrinis ir simetrinis. Pirmu atveju sujungimas atlaisvinamas, kai bent viena pusė apie tai paskelbia. Antru atveju kiekviena kryptis turi būti atlaisvinta atskirai. Asimetriniu atveju gali būti prarasti duomenys, siunčiami tos pusės, kuri dar nieko nežino apie kitos atsijungimą. Simetrinis atvejis geresnis.

Naudojamas trijų etapų rankos paspaudimas su taimeriais.

TRANSPORTINIS LYGIS



22 pav. Trys susijungimo scenarijai, naudojant trijų etapų "rankos paspaudimą". CR- CONNECTION REQUEST, ACK - CONNECTION ACCEPTED. (a) normali situacija. (b), (c) - kai egzistuoja pavėlavę dublikatai.

TRANSPORTINIS LYGIS

Transportiniai protokolai Internete (TCP, UDP)

TCP (Transmission Control Protocol). Sukurtas patikimai perduoti baitų srautą iš vieno kompiuterio į kitą (end-to-end) per nepatikimą tinklų rinkinį. Vidiniai tinklai gali skirtis savo topologija, pralaida, vėlinimais, paketų dydžiais ir kitais parametrais. TCP turi dinamiškai prisitaikyti prie viso tinklo savybių ir būti tvirtas įvairių gedimų atveju.

TCP pradžia - RFC 793. Įvairūs patobulinimai ir klaidų taisymai išdetalizuoti RFC 1122. Plėtiniai pateikti RFC 1323.

TCP paslaugos modelis. Siuntiklis ir gaviklis sukuria galinius taškus, vadinamus jungtimis (sockets). Kiekviena jungtis turi numerį, susidedantį iš IP adreso ir 16 bitų vietinės prieigos (porto) numerio. Vienai jungčiai tuo pačiu metu gali priklausyti keletas sujungimų. Sujungimai identifikuojami jungčių numeriais abiejose pusėse.

Prieigų numeriai, mažesni už 1024, vadinami gerai žinomais ir yra rezervuoti standartinėms paslaugoms. Norint pasinaudoti FTP, reikia susijungti su kito kompiuterio 21 prieiga. Telnet'ui naudojama 23-čia prieiga. Gerai žinomų prieigų numeriai surašyti RFC 1700.

Visi TCP sujungimai yra vienalaikiai dvikrypčiai taškas-taškas tipo. TCP transliacijos nepalaiko.

TCP sujungimas yra baitų srautas, o ne pranešimų srautas. Pranešimų pabaigos nepažymimos. Pvz.: jeigu siunčiantysis procesas perduoda keturias grupes po 512 baitų, tai duomenys gali būti pristatyti priimančiajam procesui kaip dvi 1024 baitų grupės arba viena 2048 baitų grupė. Gaviklis neturi būdo nustatyti, po kiek baitų buvo perduota į transportinį lygį siuntiklyje.

Kai taikomasis procesas perduoda duomenis į TCP, TCP gali pasiųsti juos tuoj pat arba padėti į buferį ir taip surinkti daugiau duomenų išsiuntimui. Kartais procesai reikalauja duomenis išsiųsti tuoj pat. Pvz.: įėjimas (login) į nutolusį kompiuterį. Po komandos užrašymo ir ENTER paspaudimo visa eilutė turi būti perduota į nutolusį kompiuterį. PUSH vėliavėlės pagalba galima nurodyti, kad TCP neužvėlintų išsiuntimo.

Dar viena svarbi TCP paslaugos galimybė - skubių duomenų išsiuntimas (urgent data). Pvz.: kitame kompiuteryje paleistos programos nutraukimui DEL arba CTRL-C turi būti perduota tuoj pat.

TRANSPORTINIS LYGIS

Kai skubūs duomenys yra gaunami, gaunantysis procesas yra pertraukiamas ir turi priimti skubius duomenis. Skubių duomenų pabaiga yra pažymėta, pradžia ne.

TCP protokolas. Kiekvienas baitas TCP sujungime turi savo 32 bitų eilės numerį. Siunčiant 10 Mb/s sparta, eilės numeriai apsisuka maždaug per valandą. Praktiškai tai užima daug daugiau laiko. Eilės numeriai yra naudojami tiek patvirtinimams tiek lango mechanizmui palaikyti. Tam yra 2 keturių baitų laukai.

TCP duomenų apsikeitimo vienetas yra segmentas. Sudarytas iš 20 baitų antraštės (plius nebūtinoji dalis) ir nulio arba daugiau baitų duomenų. Koks turi būti siunčiamojo segmento dydis sprendžia TCP programinė įranga. Maksimalus dydis toks, kad iš jo pagaminus IP deitagramą nesigautų daugiau už 65535. Kiekvienas tinklas turi maksimalų perdavimo vienetą MTU (Maximum Transfer Unit). Praktiškai MTU yra apie 2000 baitų.

Pagrindinis protokolas, naudojamas TCP, yra slankiojančiojo lango protokolas. Kai išsiunčiamas segmentas, startuoja taimeris. Gaviklis, gavęs segmentą, išsiunčia kitą segmentą, nešantį patvirtinimo numerį, lygų sekančio laukiamo baito numeriui. Jeigu taimeris apsisuka iki ateinant patvirtinimui, siuntiklis išsiunčia segmentą pakartotinai.

Iš tikrųjų viskas nėra taip paprasta. Segmentai gali būti fragmentuojami. Vieni fragmentai gali būti prarasti, kiti gali atkeliauti ne eilės tvarka. Segmentai gali vėluoti taip daug, kad siuntiklis išsiųs jų dublikatus. Jeigu jie keliauja kitu keliu ir yra kitaip fragmentuoti, originalo ir dublikato dalys susimaišys gaviklyje. Reikia specialių priemonių, norint viską išdėlioti eilės tvarka. Tinklai, esantys segmento kelyje, gali būti perkrauti.

TCP turi efektyviai spręsti visas minėtas problemas.

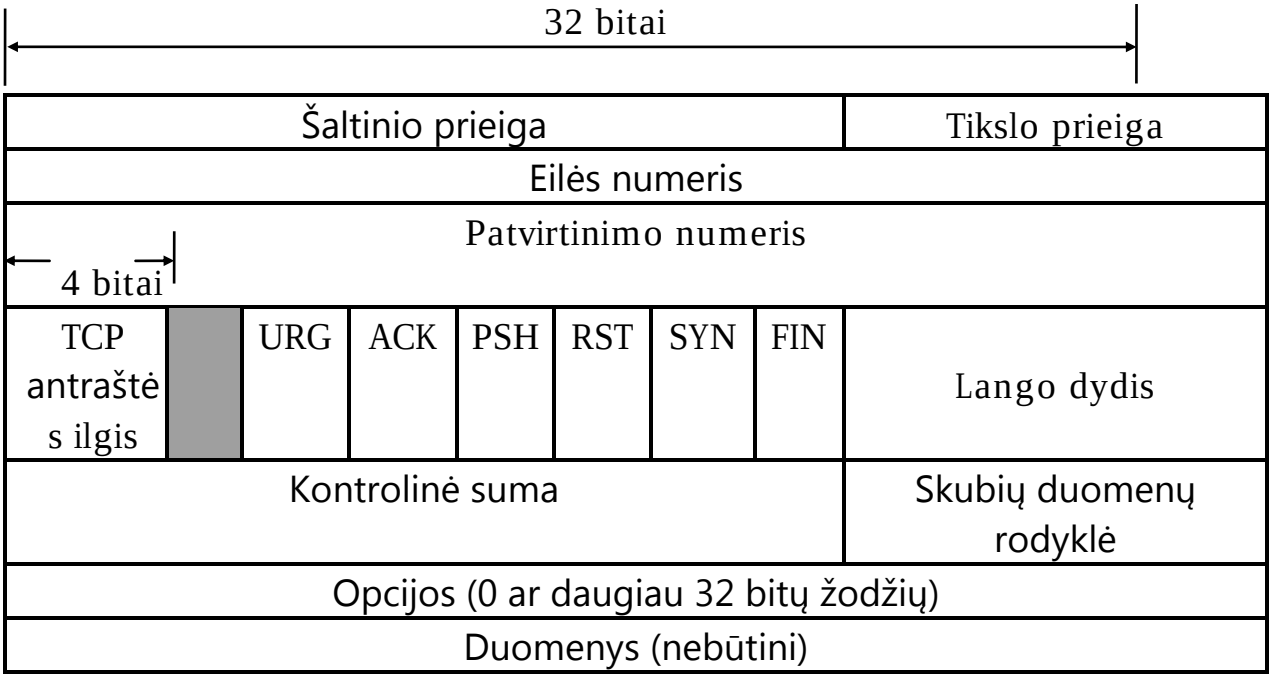
TCP segmento antraštė (23 pav). 20 baitų antraštė (fiksuota). Opcijų iki 40 baitų. Duomenų 65535-20-20=65495 baitai. Segmentai be duomenų paprastai naudojami patvirtinimams ir valdymo pranešimams.

Kiekvienas kompiuteris gali nuspręsti pats, kaip jam sunumeruoti prieigas, pradedant nuo 1024.

TCP antraštės ilgis reikalingas dėl to, kad yra opcijos.

URG = 1, kai yra naudojama skubių duomenų rodyklė, kuri parodo, kokioje vietoje, skaičiuojant nuo eilės numerio pradžios, prasideda skubūs duomenys.

TRANSPORTINIS LYGIS



23 pav. TCP segmento struktūra.

TRANSPORTINIS LYGIS

ACK = 1 parodo, kad patvirtinimo numeris yra veikiantis. ACK = 0 - patvirtinimo laukas ignoruojamas.

PSH - PUSHED duomenys. Prašoma, kad gaviklis perduotų duomenis taikomajam procesui, nelaukdamas, kol bus užpildytas buferis.

RST - naudojamas panaikinti sujungimą, kuris yra sugadintas dėl kokių nors priežasčių. Taip pat naudojamas blogiems segmentams atmesti arba neigiamai atsakyti į sujungimo užklausą.

SYN - naudojamas sujungimo užklausoje. SYN = 1, ACK = 0. Atsakymas į užklausą turi SYN = 1, ACK = 1.

FIN - sujungimo nutraukimas. Po jo dar galima priimti duomenis, kol kita pusė pasiųs FIN. SYN ir FIN segmentai turi eilės numerius, kas garantuoja, kad jie bus apdoroti eilės tvarka.

TCP palaiko srauto valdymą, naudodamas kintamo dydžio slankiojantįjį langą. Lango dydžio laukas lygus nuliui parodo, kad visi baitai, įskaitant ACK Nr-1, yra gauti, tačiau daugiau duomenis siųsti nereikia. Leidžiama siųsti, kai yra gaunamas segmentas su tuo pačiu ACK Nr. ir nenuliniu lango dydžiu.

Kontrolinė suma skaičiuojama nuo antraštės, duomenų ir taip vadinamos pseudoantraštės (24 pav).

Protokolo numeris TCP atveju lygus 6. Pseudoantraštė padeda surasti ne ten pristatytus paketus (misdelivered), tačiau tai pažeidžia protokolų hierarchiją, kadangi IP adresai priklauso IP lygiui, o ne TCP.

Opcijos. Viena iš svarbiausių opcijų leidžia nurodyti kompiuteriui, kokį maksimalų duomenų kiekį segmente jis gali priimti. Susijungimo metu abi pusės paskelbia apie maksimalų duomenų kiekį. Jeigu kompiuteris šios opcijos nenaudoja, tai reiškia 536 baitus pagal nutylėjimą. Visi Interneto kompiuteriai turi priimti TCP segmentus 536+20=556 baitų ilgio.

Linijos su didele pralaida ir (arba) dideliu vėlinimu 64 KB langas dažnai yra problema. Pvz.: T3 (44.736 Mb/s) linijoje reikia 12 ms išsiųsti visus 64 KB. Jeigu sklidimo į ten ir atgal trukmė 50 ms, tai ¾ laiko siuntiklis lauks patvirtinimų. Palydovinėse sistemose situacija dar blogesnė. Dėl to RFC 1323 yra pasiūlytas lango ilgio padidinimo būdas, leidžiantis siuntikliui ir gavikliui susitarti dėl lango dydžio. Lango dydžio lauko reikšmė gali būti perstumta iki

14 bitų į kairę. Taigi, lango dydis gali būti iki 2^{30}. Dauguma TCP realizacijų šiuo metu palaiko šią opciją.

TRANSPORTINIS LYGIS

Šaltinio adresas (IP)		
Tikslo adresas (IP)		
00000000	Protokolas=6	TCP segmento ilgis

24 pav. Pseudoantraštė, naudojama skaičiuojant kontrolinę sumą.

TRANSPORTINIS LYGIS

Kita opcija, pasiūlyta RFC 1106 ir dabar plačiai naudojama, leidžia selektyvųjį pakartojimą vietoj grįžimo atgal per *n*.

TCP sujungimų valdymas

Susijungiama, naudojant trijų etapų "rankos paspaudimą". CONNECT primityvas nusiunčia TCP segmentą su SYN=1, ACK=0 ir laukia atsakymo. Jeigu yra procesas su nurodyta segmente prieiga, tai jam perduodamas atėjęs segmentas. Procesas gali priimti arba atmesti siūlymą susijungti. Kai atmeta, išsiunčia RST=1. Kai priima, išsiunčia SYN=1, ACK=1 (25 pav.).

Jeigu du kompiuteriai tuo pačiu metu bando užmegzti ryšį, tai įvyksta tik vienas sujungimas (abiejų bandymų jungtys tos pačios).

TCP sujungimai yra dvikrypčiai nevienalaikiai. Sujungimo atlaisvinimo metu geriau tai įsivaizduoti kaip du vienkrypčius sujungimus. Kiekvienas iš jų turi būti atlaisvintas atskirai. Pasiunčiamas segmentas su FIN bitu. Kai ateina jo patvirtinimas, pusė sujungimo atlaisvinama. Kita pusė atlaisvinama analogiškai. Segmentai su FIN bitu gali būti pasiųsti vienu metu. Jie abu bus patvirtinti ir visas sujungimas atlaisvintas.

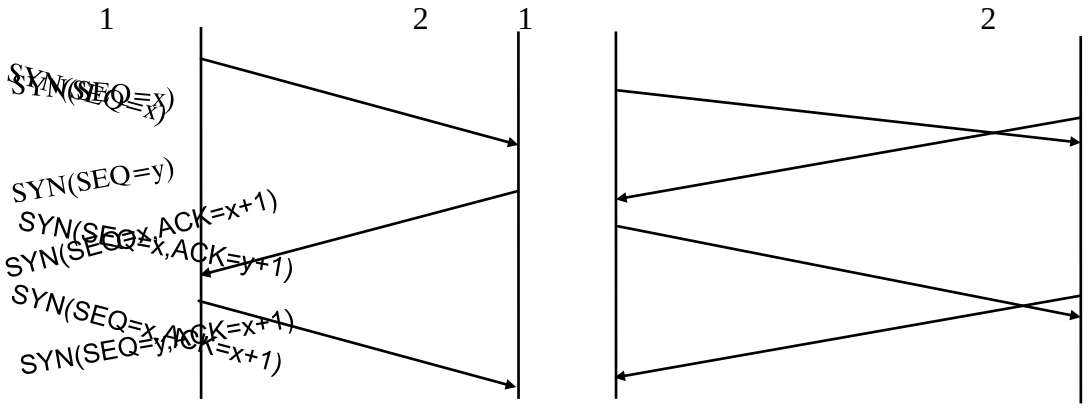
Jei atsakymas į FIN neateina per dvi gyvavimo trukmes, sujungimas atlaisvinamas. Kita pusė pastebi, kad jos daugiau niekas neklauso ir taip pat atlaisvina sujungimą.

TCP perdavimo politika

Tarkime gaviklis turi 4096 baitų buferį, kaip parodyta 26 pav. Jeigu siuntiklis išsiunčia 2048 baitų segmentą, kuris teisingai priimamas, gaviklis segmentą patvirtina. Nuo šio momento jis turi tik 2048 baitus laisvos vietos buferyje (tol kol procesas paima tam tikrą kieki duomenų iš buferio) ir paskelbia 2048 baitų langą (WIN=2048), pradedant nuo sekančio laukiamo baito (ACK=2048). Siuntiklis vėl perduoda kitus 2048 baitus, kurie yra patvirtinami, tačiau paskelbtas lango dydis lygus nuliui, kadangi gaviklio buferis yra užpildytas. Siuntiklis turi palaukti, kol taikomasis procesas gaviklio

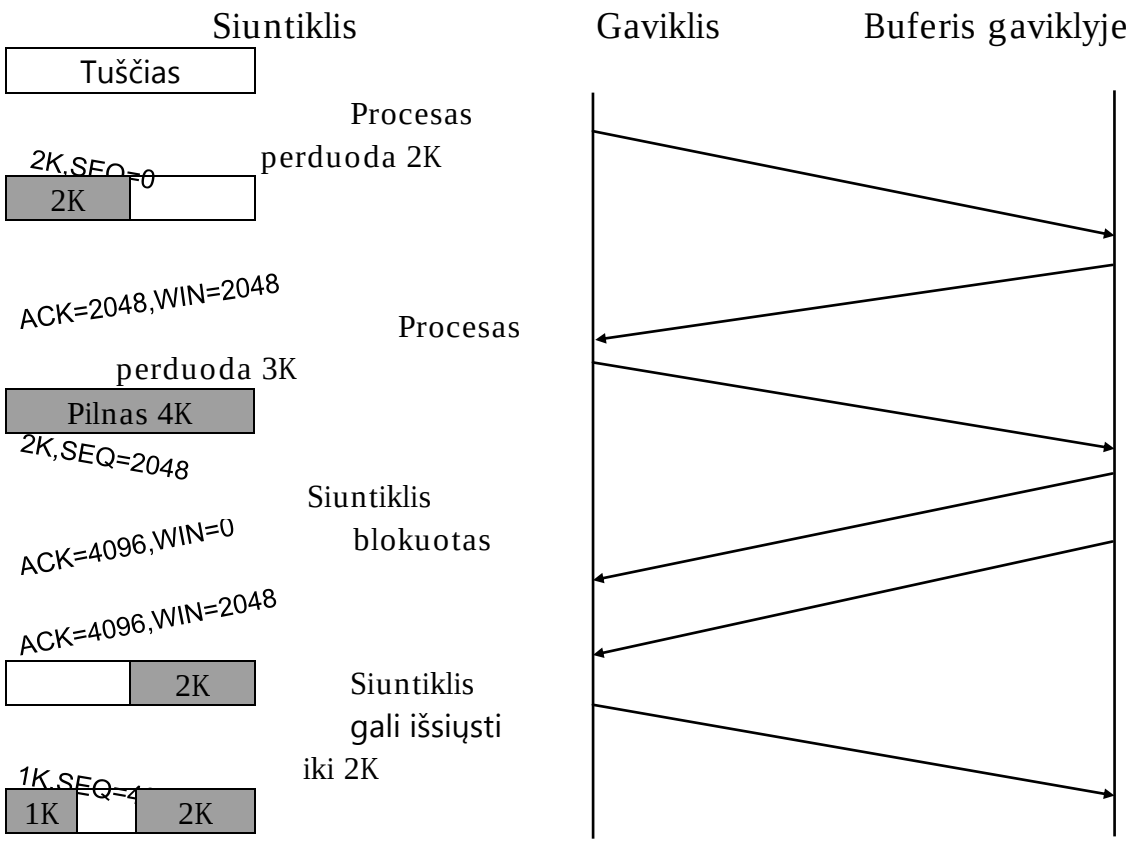
kompiuteryje paims kažkiek duomenų iš buferio ir bus paskelbtas didesnis už nulį langas.

TRANSPORTINIS LYGIS



25 pav. TCP sujungimas normaliu atveju (a), sujungimo užklausų kolizija (b).

TRANSPORTINIS LYGIS



26 pav. Lango dydžio valdymas TCP.

TRANSPORTINIS LYGIS

Kai WIN=0, gali būti išsiųsti skubūs duomenys (pvz.: išjungti procesą nutolusiame kompiuteryje). Taip pat gali būti išsiųstas vieno baito segmentas tam, kad gaviklis iš naujo paskelbtų, kokio baito laukia ir koks yra lango dydis. Tai yra naudinga tuo atveju, kai prieš tai išsiųstas patvirtinimas su lango dydžiu kur nors dingsta.

Nei siuntiklis neprivalo išsiųsti duomenis tuoj pat, kai tik juos gauna iš taikomojo lygio, nei gaviklis turi tuoj pat perduoti duomenis taikomajam procesui. Duomenys gali būti kaupiami buferiuose iki pilno jų užpildymo. Patvirtinimas išsiunčiamas, kai pilnas buferis perduodamas į taikomąjį lygį.

Panagrinėkime Telnet sujungimą, kai reaguojama į kiekvieną klavišo paspaudimą. Blogiausiu atveju, kai simbolis patenka į siuntiklio TCP, TCP sukuria 21 baito segmentą ir perduoda į IP išsiųsti kaip 41 baito deitagramą. Gaviklyje TCP iš karto siunčia 40 baitų patvirtinimą (dvi antraštės po 20 baitų). Kai nuskaitomas baitas iš buferio į taikomąjį lygį, išsiunčiamas lango informacijos atnaujinimo baitas (lango dydis padidėja vienu baitu). Kai taikomoji programa apdoroja gautą simbolį, pasiunčiamas 41 baito paketas. Iš viso 162 baitai iš keturių segmentų vienam klavišo paspaudimui. Labai neefektyvus darbas.

Vienas iš būdų, pagerinančių situaciją, yra palaukti 500 ms, tikintis duomenų iš taikomojo lygio, ir su jais išsiųsti lango informaciją ir patvirtinimą. Baitų sumažėja dvigubai, tačiau siuntiklis vis tiek vienam baitui išsiųsti naudoja 41 baito paketus. Nagle algoritmas leidžia sumažinti nenaudingai perduodamų baitų skaičių. Idėja paprasta. Pirmas baitas išsiunčiamas, o kiti kaupiami buferyje tol, kol ateina pirmojo baito patvirtinimas. Tada išsiunčiama viskas, kas yra buferyje, vienu segmentu. Kol ateina segmento patvirtinimas, vėl duomenys kaupiami buferyje. Jeigu klaviatūra spaudinėjama pakankamai greitai, o tinklas lėtas, tai į segmentą gali patekti gana daug simbolių. Algoritmas leidžia išsiųsti naują paketą, kai pvz.: yra užpildoma pusė lango.

Šis algoritmas plačiai naudojamas, tačiau kartais geriau jį išjungti. Kai terminalas dirba grafinėje aplinkoje ir naudoja pelę, tai duomenų kaupimas ir siuntimas dideliais paketais privers pelę šokinėti ekrane.

Dar viena problema - kvailo langо sindromas. Kai siunčiami duomenys dideliais blokais, o taikomoji programa gaviklyje nuskaito po vieną baitą iš buferio. Buferis užpildomas, nuskaitomas vienas baitas, išsiunčiama informacija apie 1 baito langą. Siuntiklis išsiunčia vieną baitą, langas uždaromas. Vėl nuskaitomas iš buferio tik vienas baitas ir t.t.

TRANSPORTINIS LYGIS

Sprendimas (Clark algoritmas) - neleisti gavikliui siuntinėti informaciją apie 1 baito langa dydį, palaukti kol atsilaisvins pusė buferio arba tiek, kad galėtų tilpti maksimalus segmentas, apie kurį buvo susitarta susijungimo metu. Siuntiklis taip pat gali nesiųsti informacijos tol, kol lango dydis pasidarys pakankamas.

Nagle algoritmas ir Clark algoritmas papildo vienas kitą ir gali veikti kartu. Esmė ta, kad siuntiklis neturi siųsti mažų paketų, o gaviklis jų reikalauti.

Tinklo perkrovų valdymas

Pirmas žingsnis perkrovų valdyme - atpažinti perkrovą. Timeout'as gali būti dėl dviejų priežasčių: 1. Triukšmas perdavimo linijoje. 2. Tinklo perkrova. Atskirti šiuos du atvejus labai sunku.

Šiais laikais paketai linijose prarandami vis rečiau (bevieliai tinklai - kitas reikalas). Dėl to dauguma timeout'ų Internete yra dėl perkrovų.

Pirmiausia, ką galima padaryti, kad perkrova neįvyktų? Yra tam tikra tinklo talpa ir gaviklio buferio talpa. Kiekvienas siuntiklis turi du langus: gaviklio suteikiamas langas ir perkrovos langas. Mažiausias iš jų parodo, kiek baitų siuntiklis gali išsiųsti.

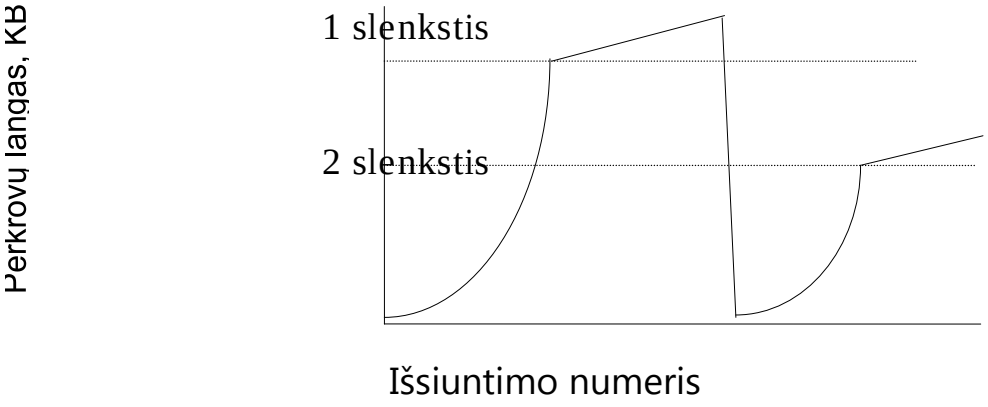
Kai įvyksta susijungimas, siuntiklis nustato perkrovos langą maksimalaus segmento dydžio. Išsiunčia vieną maksimalaus dydžio segmentą. Jei patvirtinimas ateina anksčiau negu taimeris apsisuka, perkrovos lango dydis padidinamas dar vienu maksimaliu segmentu. Išsiunčiami du segmentai. Jei jie laiku patvirtinami, lango dydis padvigubinamas ir t.t.

Perkrovos langas didėja eksponentiškai tol, kol apsisuka taimeris arba yra pasiekiamas gaviklio lango dydis. Pvz.: jeigu 4096 baitų langas dirba gerai, o 8192 duoda timeout'ą, tai lango dydis turi būti palaikomas 4096 lygyje, kad nebūtų tinklo perkrovų. Algoritmas vadinamas lėto starto algoritmu.

Interneto perkrovų algoritmas naudoja trečią parametrą - slenkstį, kurio pradinė reikšmė 64K (27 pav.). Kai persisuka taimeris (timeout), slenkstis prilyginamas pusei tuo metu egzistuojančio perkrovų lango. Perkrovų lango dydis sumažinamas iki maksimalios vieno segmento reikšmės. Toliau naudojamas lėto starto algoritmas, kol pasiekiamas slenkstis. Nuo šio

momento perkrovos langas didėja tiesiškai, pridedant po vieną maksimalų segmento dydį kiekvienam duomenų išsiuntimui. Jeigu timeout'ų nėra, perkrovos lango dydis auga iki gaviklio lango dydžio. Kai ICMP SOURCE QUENCH paketas ateina į siuntiklio TCP, jis yra traktuojamas kaip timeout'as.

TRANSPORTINIS LYGIS



27 pav. Tinklo perkrovų valdymo algoritmas Internete.

TRANSPORTINIS LYGIS

Yra įvairūs perkrovos valdymo mechanizmo patobulinimai. Perkrova numatoma dar prieš timeout'ą. Tai leidžia padidinti pralaidą nuo 40 iki 70%.

TCP taimerių valdymas

Vienas iš svarbiausių yra pakartotinio išsiuntimo taimeris. Jis įjungiamas, kai išsiunčiamas segmentas. Kokia turi būti timeout'o trukmė?

Transportiniame lygyje ši problema yra žymiai sudėtingesnė, negu kanaliniame lygyje. Pastaruoju atveju vėlinimo trukmė yra nesunkiai numatoma (pasiskirstymo dispersija maža). Kadangi kanaliniame lygyje didelis vėlinimas yra labai retas, tai patvirtinimo neatėjimas nustatytu laiku dažniausiai reiškia, kad patvirtinantysis kadras dingo.

Transportiniame lygyje tikimybės tankio kreivė žema ir plati (dispersija didelė). Netgi, kai yra žinomas vėlinimas (į ten ir atgal), nustatyti timeout'o dydį gana sunku. Pasiskirstymo kreivė gali keistis labai greitai, ypač, kai atsiranda tinklo perkrova.

Išeitis - naudoti dinaminį algoritmą, pakeičiantį timeout'o reikšmę priklausomai nuo tinklo pralaidos (performance) nuolatinių matavimo rezultatų. Paprastai naudojamas Jacobson'o (1988) algoritmas. Kiekvienam sujungimui TCP turi kintamąjį RTT(Round Trip Time) kuris yra geriausias numatomas duotu momentu vėlinimas. Matuojama kiekvieno patvirtinimo vėlinimo trukmė ir RTT perskaičiuojamas pagal formulę:

$$RTT = \alpha RTT + (1 - \alpha)M,$$

čia α - svorio koeficientas (tipiška reikšmė 7/8), M - paskutinė išmatuota vėlinimo trukmė. TCP naudoja timeout'o trukmę βRTT. Pradinėse realizacijose β buvo visą laiką lygus 2. Tačiau buvo nustatyta, kad šis variantas nelankstus, sunkiai reaguoja į greitus pokyčius tinkle. Jacobson'as pasiūlė palaikyti β proporcingą standartiniam nuokrypiui, paskaičiuotam iš patvirtinimų vėlinimo laikų pasiskirstymo funkcijos. Standartinis nuokrypis D aproksimuojamas tokia išraiška:

TRANSPORTINIS LYGIS

$$D = \alpha_1 D + (1 - \alpha_1)|RTT - M|,$$

čia α_1 - svorio koeficientas, $|RTT - M|$ - skirtumas tarp numatomos ir tikros vėlinimo trukmės duotu momentu. Jacobson'as parodė, kad D galima paskaičiuoti gana greitai, naudojant tik sveikųjų skaičių sudėtį, atimtį ir poslinkius. Dauguma TCP realizacijų naudoja šį algoritmą ir timeout'ą skaičiuoja taip:

$$Timeout = RTT + 4D.$$

Koeficientas 4 parinktas eksperimentiškai.

Karn'as papildė šį algoritmą atvejui, kai timeout'as išsenka ir segmentas išsiunčiamas iš naujo. Pasiūlė neperskaičiuoti RTT tiems segmentams, kurie yra pakartotinai išsiųsti. Tiesiog timeout'as yra tokiu atveju padvigubinamas.

Antras taimeris - atkaklumo taimeris (persistent timer). Naudojamas tam, kad ryšys "nepakibtų", kai iš pradžių gaviklio langas = 0 ir siuntiklis laukia. Po to gaviklis išsiunčia segmentą su nauja lango reikšme ir tas segmentas dingsta. Abu laukia iki begalybės. Kad begalybės nebūtų, sukasi minėtas taimeris. Laikui pasibaigus, siuntiklis išsiunčia bandomąjį paketą. Jeigu atsakyme WIN=0, tai laukia toliau. Jei WIN≠0, galima siųsti duomenis.

Trečias taimeris - gyvybės palaikymo taimeris (keepalive timer). Kai sujungimas apmiršta ilgam laikui, nusiunčiamas segmentas įsitikinti, ar antra pusė dar gyva. Jei atsakymo nėra, sujungimas nutraukiamas. Kartais tokiu būdu gali būti nutrauktas gyvas sujungimas.

Paskutinis taimeris - naudojamas įsitikinti, prieš uždarant sujungimą, kad visi paketai, susiję su tuo sujungimu, jau mirę. Taimeris sukasi dvigubą paketo gyvavimo trukmę.

UDP (User Data Protocol)

Protokolas be sujungimų. Dauguma kliento-serverio taikymų, turinčių vieną užklausą ir vieną atsakymą į ją, naudoja šį protokolą. Aprašytas RFC 768.

UDP ilgis - antraštė (29 pav.) su duomenimis. Kontrolinė suma, kaip ir TCP, naudoja pseudoantraštę.

TRANSPORTINIS LYGIS

29 pav. UDP antraštė.

TAIKOMASIS LYGIS

DNS - Domain Name System (Domenų vardų sistema)

Programos retai naudoja skaitmeninius kompiuterių, pašto dėžučių ir kitų resursų numerius. Dažniausiai naudojamos simbolių eilutės. Tačiau tinklas supranta tik skaitmeninius adresus.

ARPANET'e tai buvo paprastas tekstinis failas hosts.txt, kuriame buvo išvardyti visi kompiuteriai ir jų IP adresai. Kiekvieną naktį visi kompiuteriai pasiimdavo tą failą iš atitinkamo serverio. Kol buvo keli šimtai kompiuterių tinkle, ši sistema veikė gerai.

Išaugus kompiuterių skaičiui, failas yra perdaug didelis, kad jį būtų galima efektyviai naudoti. Be to, jei jis saugojamas viename serveryje, tai serveris gali būti perkrautas, kai jį norės nuskaityti daugelis kitų. Kaip išvengti vardų pasikartojimo, jeigu nėra centralizuoto valdymo sistemos, kuri sunkiai įsivaizduojama dideliame tarptautiniame tinkle? Šios problemos buvo išspręstos, sukuriant DNS (Domain Name System).

DNS pagrindas yra hierarchinė, paremta domenais, vardų schema ir paskirstyta duomenų bazė, realizuojanti šią vardų schemą. Pradžioje buvo naudojama kompiuterių vardams ir elektroninio pašto adresams sutapatinti su atitinkamais IP adresais. Tačiau gali būti naudojama ir kitiems tikslams. DNS apibrėžta RFC 1034 ir RFC 1035.

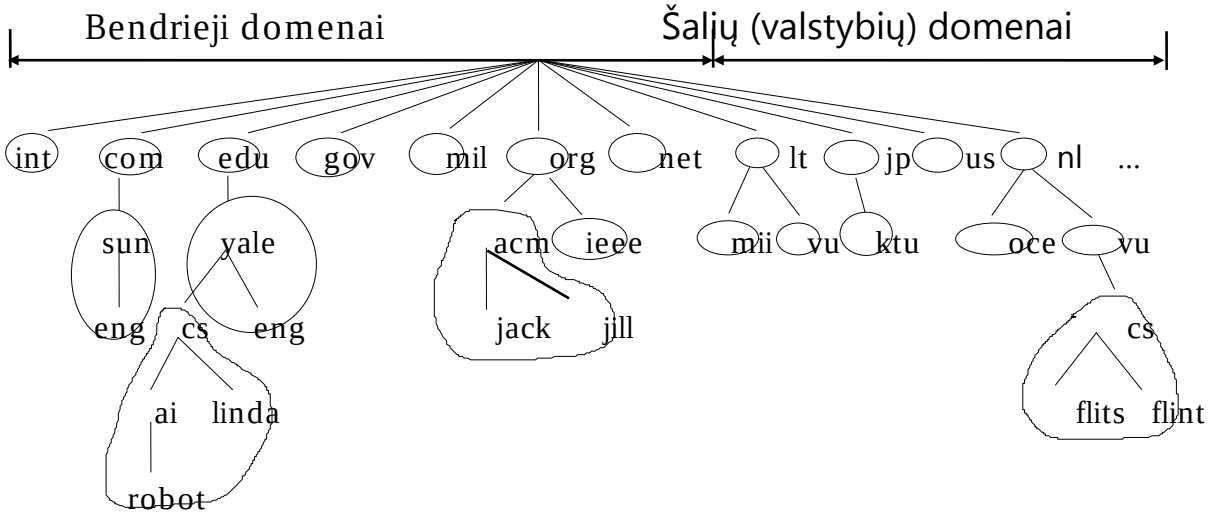
DNS naudojama taip. Taikomoji programa iškviečia procedūrą, vadinamą resolver, perduoda jai kompiuterio vardą kaip parametrą. Resolveris nusiunčia UDP paketą į vietinį DNS serverį, kuris pagal vardą suranda IP adresą ir grąžina jį resolveriui, kuris perduoda taikomajai programai. Toliau, žinodama IP adresą, programa gali užmegzti TCP ryšį su kompiuteriu arba siųsti jam UDP paketus.

DNS vardų erdvė

Internetas yra padalintas į kelis šimtus aukšto lygio domenų. Kiekvienas domenas apima daug kompiuterių. Domenai sudalyti į subdomenus, šie taip pat sudalyti į dalis ir t.t. Visi jie gali būti pavaizduoti kaip medis (29 pav.). Kiekviena šaka užsibaigia domenu, kuris neturi subdomenų, tačiau turi

kompiuterių. Aukščiausio lygio domenai skirstomi į dvi grupes: bendro pobūdžio ir šalių domenai.

TAIKOMASIS LYGIS



29 pav. Dalis DNS vardų erdvės, parodantis padalinimą į zonas

TAIKOMASIS LYGIS

Com (commercial), edu (educational institutions), gov (U.S. federalinė vyriausybė, int (kai kurios tarptautinės organizacijos). Šalių domenai - po vieną kiekvienai šakiai, kaip nustatyta ISO 3166.

Kiekvienas domenas pavadintas pagal vardus, sutinkamus, einant nuo jo į viršų iki neįvardintos šakninės dalies. Komponentės atskiriamos taškais. Pvz.: ff.vu.lt, robot.ai.cs.yale.edu.

Domenų vardai gali būti rašomi tiek didžiosiomis, tiek mažosiomis raidėmis. Komponenčių vardai gali būti iki 63 simbolių, pilnas vardas neturi viršyti 255 simbolių.

Dažniausiai visos organizacijos U.S. yra priskirtos bendriesiems domenams, o visur kitur - savo valstybių domenams. Ta pati organizacija gali priklausyti keliems aukšto lygio domenams, tačiau tai beveik nepraktikuojama.

Kiekvienas domenas valdo kitus, esančius jo viduje. Jei reikia sukurti naują domeną, reikalingas leidimas iš to domeno, kuriame yra kuriamas naujas. Kiekvienas naujas domenas gali kurti kitus savo viduje, neklausdamas nieko iš aukštesnio lygio domenų. Taip yra išvengiama vardų dubliavimo.

Įrašai apie resursus. Kiekvienas domenas turi rinkinį resursų įrašų. Paprastam kompiuteriui tai gali būti tik įrašas apie jo IP adresą. Kai resolveris perduoda domeno vardą į DNS, tai gauna atgal resursų įrašus, skirtus domenui. Įrašo formatas:

Domain_name Time_to_live Class Type Value

Kiekvienai Domain name reikšmei gali būti priskirta daug įrašų. Tai yra raktinis laukas, pagal kurį vykdoma paieška.

Time to live parodo, kiek stabilus yra įrašas. 86400 (sekundžių skaičius paroje). Kuo didesnis skaičius, tuo įrašas stabilesnis.

Class. Internetinei informacijai jo reikšmė visada IN. Neinternetinei informacijai žymėti gali būti skirtos kitokios reikšmės. Tačiau tai pasitaiko labai retai.

TAIKOMASIS LYGIS

Type parodo įrašo tipą. Svarbiausi įrašų tipai:

SOA - start of Authority. Vardų serverio zonos parametrai, administratoriams. Vardų serverio zonos parametrai, administratoriaus el.pašto adresas, unikalus serijinis numeris, įvairios vėliavėlės ir timeout'ai.

A - Address. Svarbiausias tipas. Jame yra kompiuterio IP adresas.

MX - Mail Exchange. Apibrėžia kompiuterio, kuris priima duoto domeno paštą, vardą. Dažniausiai naudojama neinternetiniams kompiuteriams gauti paštą iš Interneto. Iš nurodyto kompiuterio paštas gali būti paimtas, naudojant bet kokį abiem pusėms priimtiną protokolą.

```
electrobrain.com  86400  IN  MX  1mailserv.cs.ucla.edu
```

Electrobrain domeno, užregistruoto .com domene, bet neturinčio išėjimo į Internetą, paštas kaupiamas mailserv.cs.ucla.edu.

NS - Name Server. Kiekviena DNS duombazė turi NS įrašą kiekvienam iš aukšto lygio domenų.

CNAME leidžia sukurti pseudonimus (Aliases). Pvz.:

```
cs.mit.edu  86400  IN  CNAME  lcs.mit.edu
```

Rašant adresu cs.mit.edu, laiškai eis į kompiuterį lcs.mit.edu. cs.mit.edu yra pseudonimas. Realiai tokio vardo nėra.)

PTR duoda nuorodą į kitą vardą. Beveik visada naudojamas susieti vardą su IP adresu. Pagal IP adresą surandamas mašinos vardas.

HINFO - nurodo, koks yra kompiuterio ir operacinės sistemos tipas.

TXT - patogumui. Tokio įrašo reikšmė - teksto eilutė.

Value - reikšmė. Tai gali būti skaičius, domeno vardas arba simbolių eilutė. Prasmė priklauso nuo įrašo tipo.

TAIKOMASIS LYGIS

Vardų serveriai. DNS vardų erdvė yra suskirstyta į nepersiklojančias zonas. Kiekviena zona apima tam tikrą medžio dalį ir turi serverius, laikančius "authoritative information" apie zoną. Paprastai zona turi vieną pirminį vardų serverį, kuris paima savo informaciją iš failo diske, ir keletą antrinių vardų serverių, kurie gauna informaciją iš pirminio vardų serverio. Patikimumo padidinimui kai kurie zonos serveriai gali būti ne zonos ribose.

Resolver'is perduoda užklausą į vieną iš vietinių vardų serverių. Jeigu domenas papuola į to serverio zoną, tai atsiunčiami resursų įrašai (authoritative - pirminis). Pirminis - tas, kuris gaunamas iš įrašą valdančio šaltinio ir visada yra teisingas. Įrašai, patalpinti į kašę (cache), gali būti pasenę.

Jeigu užklaustas domenas yra iš kitos zonos, vardų serveris pasiunčia užklausą į aukštesniojo lygio domeną. Taip keliaujama medžiu, kol nueinama į vardų serverį, kuris turi informaciją apie užklaustąjį domeną. Tada įrašai tuo pačiu keliu keliauja atgal, kol pasiekia zonos vardų serverį. Jis perduoda užklausiusiajam kompiuteriui ir tuos duomenis padeda į kašę. Kadangi pakeitimai apie duomenų vardus neišsiuntinėjami į visas kašes, tai jau čia gaunasi "not authoritative" informacija, kuri turi būti saugoma tiek laiko, kiek yra parašyta time_to_live lauke.

Aprašytoji užklausa vadinama rekursine. Galima ir kitokia schema. Kai užklausa neįvykdoma lokaliai, tai užklausėjui perduodamas sekančio vardų serverio vardas. Toliau kompiuteris jau klausia šio serverio ir t.t. taip yra geriau valdomas paieškos procesas.

TAIKOMASIS LYGIS

Elektroninis paštas (email)

Elektroninis paštas egzistuoja virš 20 metų. Pirmosios email sistemos naudojo failų perdavimo protokolą. Pirmoje siunčiamojo pranešimo (failo) eilutėje turėjo būti gavėjo adresas. Pagrindiniai tokios sistemos trūkumai:

1. Vieno ir to pačio pranešimo išsiuntimas grupei žmonių labai nepatogus. O to dažnai reikia.
2. Pranešimai neturėjo vidinės struktūros. Dėl to juos buvo sunku apdoroti kompiuteriu. Pvz.: jeigu toliau persiunčiamas (forwarded) pranešimas buvo įdedamas į kito vidų, tai persiunčiamos dalies išėmimas iš gauto pranešimo buvo sudėtingas.
3. Siuntėjas niekada nežinojo ar laiškas pateko adresatui.
4. Sunku suorganizuoti laikiną pašto tvarkymo perdavimą kitam asmeniui (pvz.: sekretorei, išvykus viršininkui).
5. Vartotojo sąsaja blogai integruota į perdavimo sistemą. Iš pradžių naudojamas redaktorius laiškui parašyti. Po to jis uždaromas ir leidžiama failų perdavimo programa.
6. Nebuvo įmanoma sukurti ir pasiųsti pranešimus, kuriuose yra tekstas bei failai saugantys grafinę ir kitokią informaciją.

Kaupiantis patyrimui buvo pasiūlytos geresnės el. pašto sistemos. 1982 buvo paskelbti ARPANET el. pašto pasiūlymai RFC 821 (perdavimo protokolas) ir RFC 822 (pranešimo formatas). Jie tapo interneto standartais de facto. Po keleto metų CCITT paskelbė savo X.400 rekomendaciją.

Šiuo metu pašto sistemos, paremtos RFC 822, yra labai plačiai naudojamos, žymiai plačiau nei X.400. Taip yra dėl to, kad X.400 taip sukonstruotas ir toks sudėtingas, kad gerai jį įdiegti beveik neįmanoma. Taigi dauguma pasirinko internetinį el. paštą, pagrįstą RFC 822.

TAIKOMASIS LYGIS

Architektūra ir paslaugos

Paprastai pašto sistemos sudarytos iš dviejų posistemių: vartotojo agentai (programos), kurie leidžia skaityti ir siųsti paštą, ir pranešimų perdavimo agentai, perduodantys pranešimus adresatui. Vartotojo agentai - tai vietinės programos, leidžiančios vienu ar kitu būdu bendrauti su pašto sistema. Perdavimo agentai - tai sisteminės programos, dirbančios foniniame režime ir perduodančios paštą per sistemą.

Paprastai pašto sistemos turi penkias bazines funkcijas:

1. Rašymas (composition). Pranešimų ir atsakymų rašymas. Nors galima naudoti bet kokį teksto redaktorių pranešimų kūrimui, tačiau sistema gali padėti adresuojant laišką ir įdedant įvairias kitas antraštes. Pvz.: atsakant į laišką, pašto sistema gali paimti to laiško siuntėjo adresą ir įdėti į reikiamą vietą atsakyme.
2. Perdavimas (transfer). Pranešimų persiuntimas iš siuntėjo gavėjui. Dažniausiai tam reikia susijungti su adresatu arba kokiu nors tarpiniu kompiuteriu, išsiųsti pranešimą ir atsijungti. Tai yra daroma automatiškai.
3. Ataskaita (reporting). Siuntėjui pranešama ar laiškas pristatytas, atmestas, prarastas. Kartais tai yra labai svarbu.
4. Atsiųstų pranešimų rodymas ekrane. Kartais reikalingi specialūs konverteriai, peržiūros programos atsiųstiems vaizdams pamatyti, garsams išgirsti.
5. Gautų pranešimų patalpinimas. Pvz.: išmesti prieš skaitant, išmesti perskaičius, išsaugoti ir t.t. Taip pat turi būti galimybė skaityti išsaugotus laiškus, persiųsti kitam adresatui ar apdoroti laišką kokiu tai būdu.

Be šių pagrindinių paslaugų el. pašto sistemos gali suteikti ir kitų papildomų paslaugų. Pvz.: žmogui išvykus, automatiškai persiųsti laiškus kitu adresu. Svarbi yra galimybė išsiųsti laišką grupiniu adresu (visiems adresatams, užregistruotiems į grupės sąrašą).

Kitos papildomos galimybės - "kalkinės" kopijos (carbon copies), didelio prioriteto el. paštas, slaptas paštas, alternatyvūs adresatai, jei pagrindinis yra nepasiekiamas, galimybė sekretorėms apdoroti savo viršininkų paštą.

TAIKOMASIS LYGIS

Pagrindinė idėja visose šiuolaikinėse el. pašto sistemose yra skirtumas tarp voko (envelope) ir jo turinio. Į voką įdedamas pranešimas. Vokas yra visa informacija, reikalinga perduodant pranešimą. Tai gavėjo adresas, prioritetas, slaptumo lygis. Pranešimo perdavimo agentai naudoja voką maršrutizavimui panašiai, kaip tai yra daroma paprastame pašte. Pranešimas voke sudarytas iš dviejų dalių : antraštės ir kūno. Antraštėje yra informacija, reikalinga vartotojo agentui. Kūne tiesiog yra pats laiškas.

Vartotojo agentas

El. pašto išsiuntimas. Vartotojas turi paruošti pranešimą, nurodyti gavėjo adresą ir kitus galimus parametrus (prioritetą, slaptumo lygį). Adresas turi būti tokiame formate, kad jį suprastų vartotojo agentas. Dažniausiai yra naudojami DNS adresai. Pvz.: juozas.vysniauskas@ff.vu.lt. X.400 sistemoje adreso formatas visai kitoks. Dažnai vietoj pilno adreso galima naudoti pseudonimus (aliases). Dauguma pašto sistemų palaiko adresų sąrašus. Parašius vieną laišką ir nurodžius sąrašo pavadinimą, laiškai nusiunčiami kiekvienam, esančiam sąraše. Sąrašai gali būti palaikomi vartotojo agento. Tada siuntėjo agentas išsiunčia tiek laiškų, kiek yra nurodyta adresų sąraše. Jeigu sąrašas palaikomas nutolusioje pašto stotyje, tai vartotojo agentas išsiunčia tik vieną laišką, kuris išskleidžiamas jau pašto stotyje.

Pašto skaitymas. Paprastai, paleidus vartotojo agentą, jis pasižiūri į vartotojo pašto dėžutę (ar neatėjo naujų laiškų?) prieš ką nors parodydamas ekrane. Po to praneša apie laiškų skaičių dėžutėje arba parodo trumpą informaciją apie kiekvieną laišką (kiekvienam laiškui po eilutę). Kiekvienoje eilutėje yra po keletas laukų, paimtų iš voko arba pranešimo antraštės. Sudėtingesnėse sistemose vartotojas gali pasirinkti, kuriuos laukus rodyti ekrane. Dažniausiai viename iš laukų yra talpinami požymiai (naujas, skaitytas, atsakytas, persiųstas kitam adresatui ir t.t.). Kituose laukuose yra nurodoma, iš kur gautas laiškas, kokio ilgio, data, laikas, laiško tema (subject).

Kiekvieną laišką parodytą pašto dėžutėje, galima perskaityti, perkelti į kitą dėžutę, ištrinti. Patogu gauti laiškus, išskirstytus į atskiras dėžutes pagal temos pavadinimą. Kai kurios pašto sistemos teikia tokias paslaugas. Plačiai

naudojama yra programa Eudora, kuri be daugelio kitų paslaugų, rūšiuoja laiškus pagal temas.

TAIKOMASIS LYGIS

Pranešimo formatai. Pagrindinis el. pašto pranešimų formatas aprašytas RFC 822. Vėliau atsirado įvairūs RFC 822 plėtiniai.

RFC 822. Pranešimas sudarytas iš primityvaus voko (RFC 821), keleto antraštės laukų, tuščios eilutės ir pranešimo teksto. Kiekvienas antraštės laukas yra viena eilutė ASCII teksto, sudaryta iš lauko vardo, dvitaškio ir reikšmės. RFC 822 yra senas standartas ir nepilnai skiria voką nuo antraštės laukų, kaip tą daro naujesni standartai. Paprastai vartotojo agentas paruošia pranešimą ir perduoda jį pranešimų persiuntimo agentui, kuris naudoja kai kurias antraštės laukų reikšmes tikrojo voko konstravimui. Pagrindiniai antraštės laukai, susiję su pranešimo persiuntimu parodyti 30 pav.

Lauke *To:* talpinamas pirminio gavėjo DNS adresas. Gali būti keli adresai. *Cc:* lauke talpinami antrinių gavėjų adresai (*Cc* - carbon copy - kalkinė kopija). Pranešimo perdavimo požiūriu šie adresai niekuo nesiskiria nuo pirminių. Laiškas pristatomas visiems. Tai daugiau psichologinis skirtumas nuo pirminio. *Bcc:* (blind carbon copy - paslėpta kalkinė kopija) yra tas pats kaip *Cc:* išskyrus tai, kad ši eilutė neperduodama pirminiams ir antriniams gavėjams. Šie gavėjai nežino, kad laiškas išsiųstas dar kažkam.

Laukai *From:* ir *Sender:* nusako kas parašė ir kas išsiuntė laišką. *Sender:* reikšmė gali būti praleista, jeigu parašė ir išsiuntė tas pats asmuo. Jei pranešimo negalima pristatyti, tai yra aišku, kokiu adresu jį grąžinti atgal.

Eilutę su lauku *Received:* įrašo kiekvienas perdavimo agentas pakeliui į adresatą. Ten yra agento identifikatorius, pranešimo gavimo data ir laikas, taip pat kita informacija, reikalinga maršrutizacijos klaidoms aptikti.

Return-Path: užpildo paskutinis perdavimo agentas kelyje, kad būtų aišku, kaip surasti siuntėją. Dažniausiai ten yra įrašomas siuntėjo adresas.

RFC 822 antraštėje gali būti visa eilė kitokių laukų (31 pav.)

Reply-To: laukas naudojamas tada, kai atsakymą reikia išsiųsti trečiam asmeniui (nei siuntėjas, nei laišką parašęs asmuo nenori gauti atsakymo).

RFC 822 leidžia vartotojams kurti savo privačias antraštes. Šiuo atveju laukų vardai turi prasidėti simboliais *X-*. Pvz.: *X-kaip-malonu-gauti-laiškus:*

Po antraštės rašomas pranešimo tekstas.

TAIKOMASIS LYGIS

Antraštė	Reikšmė
To:	Pirminių gavėjų adresai
Cc:	Antrinių gavėjų adresai
From:	Asmuo ar asmenys, kurie sukūrė pranešimą
Bcc:	Adresai paslėptųjų, "kalkinių" kopijų
Sender:	Tikrojo siuntėjo el. pašto adresas
Received:	Šia eilutę prideda kiekvienas perdavimo agentas visame maršrute
Return-Path:	Gali būti panaudotas kelio atgal pas siuntėją nuorodai

30 pav. RFC 822 antraštės laukai, susiję su pranešimo perdavimu.

TAIKOMASIS LYGIS

Antraštė	Reikšmė
Date:	Išsiuntimo data ir laikas
Reply-To:	Adresas, kuriuo reikia siųsti atsakymus
Message-Id:	Unikalus šio pranešimo numeris
In-Reply-To:	Pranešimo, į kurį yra atsakoma, identifikatorius
References:	Kiti reikalingi identifikatoriai
Keywords:	Vartotojo naudojami raktiniai žodžiai
Subject:	Vienoje eilutėje suformuluota pranešimo tema

31 pav. Kai kurie laukai, naudojami RFC 822 antraštėje

TAIKOMASIS LYGIS

MIME - Multipurpose Internet Mail Extensions (daugiatiksliai Internet pašto plėtiniai).

RFC 822 tinka tik tekstiniams pranešimams, naudojantiems ASCII kodavimą. Dėl to atsiranda problemų siunčiant pranešimus kalbomis, naudojančiomis kirčiuotas raides, ne lotynišką abėcėlę, nenaudojančiomis jokios abėcėlės. Taip pat yra problemų, siunčiant pranešimus, kuriuose nėra teksto (audio, video).

Sprendimas pasiūlytas RFC 1341, papildytas RFC 1521 ir pavadintas MIME (Multipurpose Internet Mail Extensions). Pagrindinė idėja yra ta, kad lieka galioti RFC 822, tačiau yra struktūrizuojamas pranešimo kūnas ir apibrėžiamos netekstinių pranešimų kodavimo taisyklės. Visi MIME pranešimai gali būti perduodami, naudojant egzistuojančias pašto programas ir protokolus. Turi būti šiek tiek pakeistos išsiuntimo ir priėmimo programos.

MIME apibrėžia penkias naujas pranešimo antraštes, kaip parodyta 32 pav. Pirmoji iš jų pasako vartotojo agentui, priimančiam pranešimą, kad tai MIME pranešimas ir kokia MIME versija yra naudojama. Bet koks pranešimas, neturintis MIME-*Version:* lauko, yra traktuojamas kaip angliškas tekstas.

Content-Transfer-Encoding: nurodo, kaip yra užkoduotas pranešimas perdavimui tinkle. Yra pasiūlyta keletas kodavimo būdų. Vienas iš plačiausiai naudojamų, perduodant dvejetainius failus, yra base64 kodavimas. Šiuo atveju 24 bitų grupės yra skirstomos į 6 bitų dalis. Kiekviena dalis yra persiunčiama kaip ASCII ženklas. Nulis koduojamas "A", vienetas "B" ir t.t. Taip pat yra panaudojamos mažosios raidės ir "+", "/" (62, 63). Du lygybės ženklai "==" parodo, kad paskutinėje grupėje tik 8 bitai. Vienas lygybės ženklas reiškia, kad paskutinėje grupėje yra 16 bitų. "CR" ir "LF" kodai yra ignoruojami, taigi, jie gali būti panaudojami eilučių sutrumpinimui užkoduotame faile. Tokiu būdu bet koks dvejetainis failas gali būti sėkmingai persiųstas.

Pranešimams, kurie beveik pilnai yra ASCII, išskyrus keletą simbolių, base64 kodavimas gaunasi neefektyvus. Tada geriau naudoti quoted-printable kodavimą. Ženklai, kurių kodai viršija 127, koduojami lygybės ženklu ir po jo einančiu kodu, užrašytu dviejų šešioliktainių skaitmenų pavidalu.

Kai netinka nė vienas iš minėtų kodavimo būdų, galima nurodyti ir naudoti kokį nors savo kodavimą.

TAIKOMASIS LYGIS

Antraštė	Reikšmė
MIME-*Version*:	Identifikuoja MIME versiją
Content-Description:	Eilutė, nusakanti pranešimo turinį
Content-Id:	Unikalus identifikatorius
Content-Transfer-Encoding:	Pranešimo kodavimo būdas
Content-Type:	Pranešimo tipas

32 pav. RFC 822 antraštės, pasiūlytos MIME.

TAIKOMASIS LYGIS

Content-Type: lauke nurodomas pranešimo tipas. RFC 1521 yra apibrėžti septyni tipai, kurie gali turėti vieną ar daugiau potipių. Tipas nuo potipio atskirti "/". Pvz.: *Content-Type*: *video/mpeg*.

Potipis turi būti nurodytas išreikštai, nėra reikšmių pagal nutylėjimą. Pradinis tipų ir potipių sąrašas, apibrėžtas RFC 1521, parodytas 33 pav. Nuo to laiko šis sąrašas buvo papildytas daugeliu kitų tipų ir potipių. *Text* tipas yra tiesiog tekstas. *Text/plain* yra paprastas pranešimas, kuris gali būti parodytas toks, koks yra gautas, be iškodavimo ir tolimesnio apdorojimo. Tai leidžia siųsti paprastus pranešimus MIME formatu.

Text/richtext leidžia panaudoti paprastą žymėjimo kalbą tekste. Ši kalba duoda nepriklausomą nuo sistemos būdą parodyti paryškintą tekstą, kursyvą, didesnius ir mažesnius šriftus, pastraipos pirmos eilutės atitraukimą nuo krašto, kraštų išlyginimą, paprastą puslapio planavimą (page layout). Žymėjimo kalba paremta SGML (Standart General Markup Language), kuri yra WWW HTML pagrindas.

Image tipas yra naudojamas persiųsti paveiksliukais. Be *GIF* ir *JPEG* gali būti ir kiti formatai.

Audio ir *video* tipai yra garsams ir judantiems paveiksliukams perduoti. *Video* yra tik vaizdo informacija, bet ne garso kanalas. Jeigu perduodamas failas su garsu, tai vaizdo ir garso dalys perduodamos atskirai.

Application tipas yra formatams, kurie reikalauja išorinio apdorojimo, neapibrėžto kituose tipuose. *Octet-stream* yra neinterpretuojamų baitų srautas. Priimant šį srautą, vartotojo agentas turi pasiūlyti vartotojui nukopijuoti viską į failą. Tolimesnis failo apdorojimas yra vartotojo reikalas.

Postscript reiškia, kad failas yra *PostScript* formate. Vartotojo agentas gali iškviesti išorinį *PostScript* interpretatorių ir parodyti atsiunčiamus *PostScript* pranešimus. Tai gali būti pavojinga, nes rodant failą yra vykdoma jame esanti programa, kuri gali ne tik rodyti failą, bet ir skaityti, keisti arba trinti vartotojo failus ir t.t.

Message tipas leidžia pilnai įdėti vieną pranešimą į kitą. Tai yra naudinga persiunčiant gautą pranešimą kitam adresatui. *Rfc822* potipis reiškia, kad viduje vieno pranešimo yra kitas pilnas RFC 822 formato pranešimas.

Partial potipis duoda galimybę įdėtą pranešimą suskaidyti į dalis ir persiųsti jas atskirai. Parametrai leidžia pranešimą atkurti iš gautų dalių.

TAIKOMASIS LYGIS

Tipas	Potipis	Aprašymas
Text	Plain	Neformuotas tekstas
	Richtext	Tekstas su paprastomis formavimo komandomis
Image	Gif	Paveiksliukas GIF formate
	Jpeg	Paveiksliukas JPEG formate
Audio	Basic	Garsai
Video	Mpeg	Filmas MPEG formate
Application	Octet-stream	Neinterpretuojama baitu seka
	Postscript	Postscriptinis spausdinamas dokumentas
Message	Rfc822 Partial External-body	MIME RFC 822 pranešimas Pranešimas padalintas į dalis Pranešimas gali būti paimtas per tinklą
Multipart	Mixed	Nepriklausomos dalys norodyta tvarka
	Alternative	Tas pats pranešimas skirtingais formatais
	Parallel	Dalys turi būti peržiūrimos tuo pačiu metu
	Digest	Kiekviena dalis yra pilnas RFC 822 pranešimas

33 pav. MIME tipai ir potipiai, apibrėžti RFC 1521.

TAIKOMASIS LYGIS

External-body potipis gali būti panaudotas labai ilgiems pranešimams (pvz.: video filmams). Vietoj MPEG failo pranešime yra duodamas FTP adresas ir gavėjo vartotojo agentas gali paimti pranešimą per tinklą tada, kai reikės. Labai naudinga tada, kai filmas siunčiamas ištisam sąrašui žmonių ir tik keletas iš jų tą filmą panori pasižiūrėti.

Multipart tipo atveju pranešimas gali būti sudarytas iš keleto dalių su aiškiai apibrėžta kiekvienos dalies pradžia ir pabaiga.

Mixed potipis leidžia skirtingas atskiras dalis. Priešingai, *alternative* potipis reiškia, kad visose dalyse yra tas pats pranešimas tik kitokiame pavidale. Pvz.: pranešimas gali būti pasiųstas kaip *ASCII text*, *richtext* ir *PostScript*. Gerai sukonstruotas vartotojo agentas turėtų parodyti pranešimą *PostScript* formate. Jeigu tai neįmanoma, tai rodoma kaip *richtext*. Jeigu ir tai neįmanoma, rodomas kaip paprastas tekstas.

Alternative potipį galima panaudoti siunčiant tekstą ir muzikinį jo variantą.

Parallel potipis naudojamas tada, kai visas dalis reikia demonstruoti tuo pačiu metu. Pvz.: įgarsinto filmo vaizdo ir garso dalys.

Digest potipis naudojamas, kai daug pranešimų sudedama į vieną ir išsiunčiama.

Pranešimų perdavimas

SMTP - Simple Mail Transfer Protocol (paprastas pašto perdavimo protokolas). Internete el. paštas pristatomas iš šaltinio į tikslą, naudojant TCP sujungimą su 25-ta prieiga tinklo kompiuteryje. Šios prieigos "klauso" el. pašto demonas (daemon) - programa suprantanti SMTP. Šis demonas priima ateinančius kvietimus susijungimui ir kopijuoja pranešimus į atitinkamas pašto dėžutes. Jei pranešimas negali būti pristatytas, siuntėjui grąžinama pirmoji pranešimo dalis su klaidos nuoroda.

SMTP yra paprastas ASCII protokolas. Po TCP susijungimo su 25 prieiga, siunčiantysis kompiuteris, dirbantis kaip klientas, laukia pranešimo iš priimančiojo kompiuterio, dirbančio kaip serveris. Serveris išsiunčia eilutę

teksto identifikuodamas save ir pasakydamas ar yra pasiruošęs priimti paštą. Jeigu nepasiruošęs, klientas atlaisvina sujungimą ir bando siųsti vėliau.

TAIKOMASIS LYGIS

Jeigu serveris pasiruošęs priimti paštą, klientas praneša, kas ir kam siunčia paštą. Jeigu toks gavėjas egzistuoja, serveris leidžia klientui siųsti pranešimą. Pranešimas išsiunčiamas, serveris patvirtina, kad pranešimas gautas. Kai įvyksta pilnas pasikeitimas paštu iš abiejų pusių, susijungimas atlaisvinamas.

Nors SMTP protokolas yra gerai apibrėžtas (RFC 821), tačiau kartais gali kilti problemų. Viena iš jų susijusi su pranešimų ilgiu. Kai kurios senesnės realizacijos nepalaiko ilgesnių už 64 KB pranešimų. Kita problema, susijusi su timeout'ais, kai jie nevienodi kliente ir serveryje. Kuri nors pusė gali nelauktai atlaisvinti sujungimą. Retai pasitaikančiose situacijose gali kilti nesibaigiančios pašto audros. Taip gali atsitikti, kai abi pusės palaiko skirtingus adresų sąrašus, tačiau su nuorodomis į kiekvieną iš jų. Bent vieno pranešimo persiuntimas sukelia nesibaigiantį el. pašto srautą.

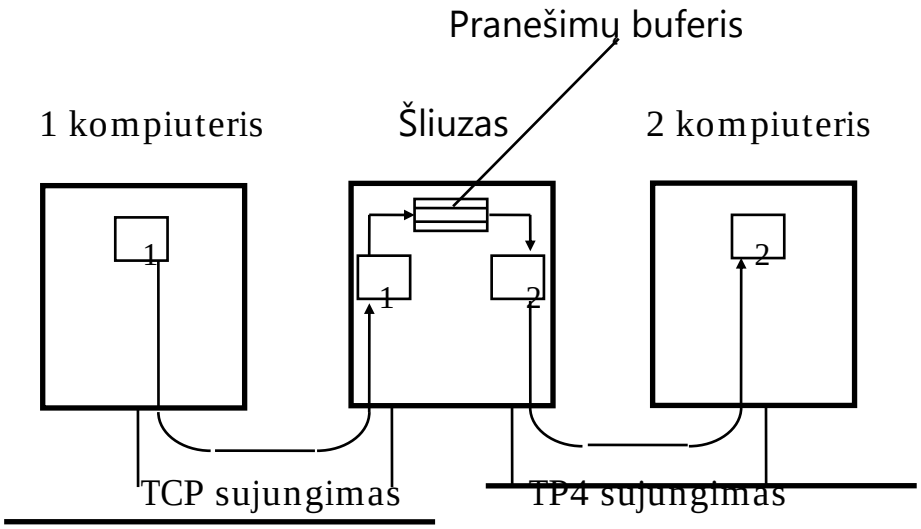
Šios problemos sprendžiamos, naudojant išplėstąjį SMTP (ESMTP), apibrėžtą RFC 1425. Klientai, norintys naudoti šį protokolą, turi išsiųsti pradžioje ne HELO, o EHLO. Jeigu tai yra atmetama tai serveris palaiko tik SMTP ir apsikeitimas vyksta šiuo protokolu.

Elektroninio pašto šliuzai. Kai siuntiklis ir gaviklis yra Internete ir gali naudoti TCP sujungimus, apsikeitimas pranešimais vyksta be didesnių problemų. Tačiau ne visi kompiuteriai yra Internete, kai kurie tinklai specialiai nejungiami prie Interneto dėl saugumo sumetimų, organizuojamos "ugnies sienos"(firewalls), apribojančios priėjimą prie kompiuterių iš Interneto pusės. Nežiūrint to, apsikeitimas el. pašto pranešimais yra reikalingas. Kita problema, kai vienas kompiuteris supranta SMTP, o kitas tik X.400.

Visos šios problemos sprendžiamos, panaudojant taikomojo lygio el. pašto šliuzus (gateways). Tipiškas atvejis pavaizduotas 34 pav. Kompiuteris 1 supranta TCP/IP ir RFC 822, o kompiuteris 2 supranta tik OSI TP4 ir X.400. Nežiūrint to, paštas tarp jų veikia per šliuzą, kuris su pirmuoju organizuoja TCP sujungimą, RFC 822 formatą paverčia į X.400 ir TP4 sujungimo pagalba pranešimą perduoda antrajam (TP4 yra TCP ekvivalentas OSI protokolų steke). Viskas atrodo paprasta, tačiau iš tikrųjų taip nėra. Pirma problema yra ta, kad

Interneto ir X.400 adresai skiriasi iš principo. Antra problema - vokas arba antraštės laukai, egzistuojantys vienoje sistemoje, gali neegzistuoti kitoje. Pvz.: viena sistema reikalauja prioriteto, o kita to neturi. Gali būti prarasta labai svarbi informacija.

TAIKOMASIS LYGIS



34 pav. Elektroninio pašto perdavimas, naudojant taikomojo lygio šliuzą

TAIKOMASIS LYGIS

Dar blogiau, kai yra skirtinga pranešimo struktūra. Ką gali padaryti šliuzas su pranešimu iš Interneto, kur nurodytas garso failas, atsiunčiamas per FTP, jeigu kita pusė šios koncepcijos nepalaiko? Ką daryti, kai X.400 nurodo pristatyti pranešimą tam tikru adresu ir, jei to padaryti nepavyksta, pasiųsti pranešimą faksu? RFC 822 modelis fakso nepalaiko. Iš tikrųjų čia nėra paprastų sprendimų. Problemų nekyla, jeigu perduodamas paprastas ASCII tekstas.

Galutinis pristatymas. Dažniausiai galiniai vartotojai negali išsiųsti arba priimti paštą. Tam reikalingas pašto serveris, surenkantis iš vartotojų pranešimus ir išsiunčiantis už įstaigos ribų. Šiuo atveju reikalingas tam tikras galinio vartotojo ir serverio bendravimo protokolas.

Paprastas protokolas, naudojamas paimti paštą iš nutolusios pašto dėžutės, yra POP3 (Post Office Protocol), aprašytas RFC 1225. Jis turi komandas prisijungti prie pašto stoties (log in), atsijungti (log off), paimti pranešimus, ištrinti pranešimus. Pats protokolas yra panašus į SMTP. Pagrindinis jo tikslas - paimti pranešimus iš nutolusios pašto dėžutės ir išsaugoti vartotojo kompiuteryje.

Žymiai sudėtingesnis pristatymo protokolas yra IMAP (Interactive Mail Access Protocol), aprašytas RFC 1064. Patogus vartotojams, kurie dirba su keletu kompiuterių ir nereikia pašto kopijuoti į kiekvieną iš jų. Visi laiškai saugomi pašto serveryje kaip reliacinė duomenų bazė, prie kurios įrašų galima prieiti, naudojant įvairius požymius (Pvz.: "duokite man pirmąjį Jono laišką").

Dar vienas pristatymo protokolas yra DMSP (Distributed Mail System Protocol), kuris yra PCMAIL sistemos dalis, aprašytas RFC 1056. Pranešimai gali būti ne viename serveryje. Paštą galima atsisiųsti į savo kompiuterį ir atsijungti.

Nepriklausomai nuo to, ar paštas pristatomas tiesiai į vartotojo kompiuterį ar į nutolusį serverį, daugelis sistemų turi įvairių papildomų galimybių pranešimų apdorojimui. Ypatingai geras dalykas yra filtrų nustatymas. Nurodomos taisyklės, pagal kurias apdorojami atsiųsti

pranešimai. Tokiu būdu galima atsikratyti nereikalingų pranešimų, o kitus pateikti tam tikru pavidalu (Pvz.: nurodomas šriftas, jo dydis ir t.t.).

Kita svarbi pristatymo galimybė - persiųsti ateinantį el. paštą kitu adresu.Tas kitas adresas gali būti kompiuteris, perduodantis *Subject:* lauko reikšmę į vartotojo pranešimų gaviklio (pager) ekraną.

TAIKOMASIS LYGIS

Dar viena galimybė - pasinaudoti "atostogų demonu" - programa, kuri į visus laiškus, gautus atostogų metu, atsako pranešimu apie tai, kad gavėjas atostogauja. Programa gali turėti įvairių kitų savybių, kaip pvz.: nesiųsti atsakymų į laiškus, išsiųstus pagal adresų sąrašą. (Kitaip iš kiekvieno sąraše esančio adreso gali būti gautas pranešimas apie atostogas. O jeigu sąraše keli tūkstančiai adresų?).

Elektroninio pašto slaptumas (privatumas, konfidencialumas)

Nenaudojant specialių užslaptinimo priemonių, laiškai gali būti perskaityti, pakoreguoti, sugadinti bet kurioje tarpinėje pašto stotyje. Pranešimų slaptumo užtikrinimas - svarbi ir sudėtinga problema. El. pašto saugumui užtikrinti yra naudojamos įvairios sistemos. Dvi iš jų - PGP ir PEM.

PGP - Pretty Good Privacy (Phil Zimmermann, 1995). Ši sistema užtikrina privatumą, autentifikavimas, skaitmeninius parašus ir suspaudimą. Pilnas paketas, į kurį įeina visi išeities kodai (source code), laisvai platinamas Internete, skelbimų lentose ir komerciniuose tinkluose. Dėl savo kokybės, nulinės kainos ir labiausiai paplitusių operacinių sistemų palaikymo (DOS/Windows, Unix, Macintosh) yra labai plačiai naudojamas šiuo metu.

PEM - Privacy Enchanced Mail. Tai yra oficialus Interneto standartas, aprašytas RFC 1421- RFC 1424. Apibrėžia privatumą ir autentifikavimą RFC 822 el. pašto sistemose.

USENET naujienos

Tai naujienų grupių (newsgroups) sistema, vadinama net news. Dažnai net news (tinko naujienos) vadinamos USENET. USENET ir Internet yra ne tas pats. Kai kurie Internete esantys kompiuteriai negauna tinklo naujienų, o kiti jas gali gauti nebūdami Internete.

TAIKOMASIS LYGIS

USENET vartotojo požiūriu

Naujienų grupė - tai yra pasaulinis diskusijų forumas tam tikra tema. Galima prisiregistruoti prie tam tikros naujienų grupės (subscribe). Prisiregistravusieji gali panaudoti specialų vartotojo agentą - naujienų skaitymo programą visiems grupės pranešimams perskaityti. Taip pat galima rašyti straipsnius į grupę. Kiekvienas straipsnis automatiškai pristatomas visiems prisirašiusiems kur jie bebūtų.

Naujienų grupių yra labai daug (dešimtys tūkstančių). Dėl to jos turi tam tikrą hierarchinę struktūrą. Kitaip tiek daug grupių būtų sunku valdyti. 35 pav. parodyta "oficialiosios" hierarchijos viršūnė. Kiekviena iš 35 pav. parodytų kategorijų yra suskirstytos į subkategorijas ir t.t. Pvz.: rec.sports yra apie sportą. Rec.sports.basketball - apie krepšinį. Rec.sports.basketball.women - apie moterų krepšinį.

Egzistuoja daugybė naujienų skaitymo programų. Beveik visada, paleidus programą, yra peržiūrimas failas, kuriame surašyta prie kokių grupių vartotojas yra prisiregistravęs. Tada parodoma kiekvieno neperskaityto straipsnio reziume. Straipsnius galima skaityti, ištrinti, išsaugoti, atspausdinti ir t.t.

Naujienų skaitymo programos leidžia prisiregistruoti prie tam tikrų grupių ir išsiregistruoti. Tai yra tik prisiregistravimo informacijos pakeitimas vietiniame faile. Programos leidžia parašyti straipsnį ir jį patalpinti naujienų grupėje. Vieną ir tą patį straipsnį galima patalpinti į keletą grupių. Galima apriboti straipsnio skelbimo geografiją ir t.t.

Naujienų grupės gali būti moderuojamos. Tai reiškia tam tikrą cenzūrą. Moderatorius atrenka gerus ir atmeta blogus straipsnius.

Kadangi kiekvieną dieną prie įvairių grupių prisijungia tūkstančiai naujų vartotojų, tai grupės dažnai publikuoja FAQ (Frequently Asked Question), kad nereikėtų naujokams vėl ir vėl uždavinėti tuos pačius klausimus.

Straipsnius į grupes galima siųsti anonimiškai. Tam yra anoniminiai pašto persiuntimo serveriai (anonymous remailers). Pranešimuose jie pakeičia *From:*, *Sender:*, *Reply To:* laukų reikšmes tam tikrais siuntėjo identifikaciniais numeriais.

TAIKOMASIS LYGIS

Vardas	Temos
Comp	Kompiuteriai, informatika, kompiuterių pramonė
Sci	Fizikiniai ir techniniai mokslai
Humanities	Literatūra ir humanitariniai mokslai
News	Diskusijos apie patį USENET
Rec	Poilsis, įskaitant sportą ir muziką
Misc	Viskas, kas nepritampa prie kitų temų
Soc	Socialinės temos
Talk	Polemikos, debatai
Alt	Alternatyvus medis, apimantis beveik visas temas

35 pav. USENET hierarchijos viršūnė santykio signalas/triukšmas didėjimo tvarka

TAIKOMASIS LYGIS

Kaip USENET yra realizuotas

Mažos naujienų grupės gali būti suorganizuotos kaip adresų sąrašai (mailing lists). Pranešimas persiunčiamas sąrašui ir jį gauna visi esantys sąraše. Didelės grupės taip organizuoti neįmanoma, nes labai apkraunami pašto serveriai. Paprastai kiekvienas ISP (Internet Service Provider), organizuoja, įsteigia, kaupia naujienas savo serveryje, suskirstydamas pagal grupes į atskirus katalogus. Iš ten kiekvienas gali pasiimti naujienas, kai jam reikia. Taigi, serveryje saugoma tik viena naujienų kopija, kuri ištrinama praėjus tam tikram laikui.

Serveris yra USENET'e, jei jis gali gauti naujienas iš kito serverio. Taip susiformuoja ištisas tinklas, kuri gali išeiti iš Internet ribų.

Periodiškai kiekvienas serveris paklausia savo naujienų tiekėją, ar nėra naujų naujienų. Jeigu yra, jos atsiunčiamos. Taip naujienos plinta tinklu. Šiuo metu dažniausiai būna taip, kad naujienų tiekėjas pats pasiūlo naujienas, kai jų prisikaupia pakankamas kiekis. Ne kiekvienas serveris gauna visas naujienų grupes. Kitaip būtų labai apkrautas tinklas ir reikėtų skirti daug diskinės atminties.

Naujienų straipsniai turi tokį pat formatą, kaip ir RF C822 pranešimai. Pridėta dar keletas antraščių. Naujienų antraštės aprašytos RFC 1036.

Path: antraštė parodo, kiek tinklo mazgų praėjo straipsnis nuo siuntėjo iki gavėjo.

Newsgroups: antraštė parodo, kokiai naujienų grupei priklauso pranešimas. Gali būti keleto grupių pavadinimai. Dėl to yra *Followup-To:* antraštė, kur nurodoma į kokią grupę siųsti atsakymus ir komentarus.

Distribution: antraštė pasako, kur platinti straipsnį. Ten gali būti vienos arba keleto valstybių kodai, specifinių serverių arba tinklų pavadinimai. Žodis "world" reiškia, kad naujiena skirta visam pasauliui.

Nntp-Posting-Host: antraštė analogiška RFC 822 Sender: antraštei. Parodo, kuriame kompiuteryje patalpintas straipsnis net jeigu jis parašytas visai kitur.

References: antraštė parodo, kad tai yra atsakymai (reakcija) į kitą ankstesnį straipsnį. Įrašomas to straipsnio identifikatorius.

Organization: antraštė parodo, kokiai organizacijai priklauso straipsnio autorius.

Lines: antraštė parodo pranešimo kūno ilgį. Antraštės eilutės ir tuščia eilutė, skirianti antraštę nuo pranešimo kūno, neskaičiuojamos.

TAIKOMASIS LYGIS

Komanda	Reikšmė
LIST	Duok man visų naujienų grupių ir straipsnių sąrašą
NEWSGROUPS *date time*	Duok man sąrašą naujienų grupių, sukurtų nuo *date/time*
GROUP *grp*	Duok man visų straipsnių *grp* sąrašą
NEWNEWS *grp date time*	Duok man naujų straipsnių iš nurodytų grupių sąrašą
ARTICLE *id*	Duok man nurodytą straipsnį
POST	Aš turiu straipsnį tau, kuris yra paskelbtas čia
IHAVE *id*	Aš turiu straipsnio identifikatorių. Ar tau jo reikia?
QUIT	Baigti sesiją

36 pav. Pagrindinės NNTP komandos, naudojamos naujienų sklidimui.

TAIKOMASIS LYGIS

NNTP - Network News Transfer Protocol (tinklo naujienų perdavimo protokolas). Aprašytas RFC 977. Kuriant buvo siekiama dviejų tikslų. Pirmiausia, naujienos turi sklisti nuo vieno kompiuterio prie kito per patikimą sujungimą (TCP). Antra, užtikrinti naujienų skaitymą nutolusiems vartotojams. Realizuota ir viena ir kita.

Klientas organizuoja TCP sujungimą su vieno iš naujienų tiekėjo 119 prieiga. Ši prieiga yra arba NNTP demonas, kuris visą laiką laukia klientų arba sukuriamas tada, kai jo prireikia. Po sujungimo klientas ir serveris siunčia vienas kitam komandų ir atsakymų sekas. Komandos ir atsakymai naudojami tam, kad būti tikram, jog klientas gauna visus reikalingus straipsnius, o ne dublikatus, nepriklausomai nuo to, kiek naujienų tiekėjų jis naudoja. Pagrindinės komandos parodytos 36 pav.

LIST ir NEWSGROUPS komandos parodo, ką turi serveris. Atsakymai į komandas yra ASCII formato sąrašas, kur vienai grupei skirta viena eilutė. Eilutėje taip pat yra požymis, parodantis, ar galima skelbti straipsnius toje grupėje.

GROUP ir NEWNEWS komandų pagalba galima sužinoti, kokie straipsniai yra serveryje. ARTICLE pagalba galima gauti reikalingą straipsnį. IHAVE komanda parodo, kad klientas yra gavęs straipsnį iš kito tiekėjo. POST reiškia, kad straipsnis yra paskelbtas pas klientą. Serveris gali paimti tą straipsnį arba jo atsisakyti. Sesija baigiama QUIT komanda.

Naujienų perdavimą gali inicijuoti kaip klientas taip ir serveris. Abiem atvejais yra naudojamas "sustok ir lauk" principas. Tipiškai atsakymo yra laukiama apie 100 ms. Tokiu būdu, siunčiant daug straipsnių, prarandama nemažai laiko.

Pasaulinis voratinklis (The World Wide Web (WWW))

WWW leidžia pasiekti susietus tarpusavyje dokumentus, esančius tūkstančiuose kompiuterių visame Internete. Per 5 metus WWW taip paplito, kad daugelis žmonių galvoja, jog tai ir yra Internetas.

Web'as buvo sukurtas 1989 m. Europos branduolinių tyrimų centre CERN. Pradžioje tai buvo tiesiog geras būdas keistis duomenimis, piešiniais, nuotraukomis fizikams, gyvenantiems daugelyje pasaulio šalių.

TAIKOMASIS LYGIS

Pirminę Web idėją suformulavo CERN fizikas Tim Berners-Lee 1989 m. kovo mėnesį. Pirmas (tekstinis) variantas pradėjo veikti po 18 mėnesių. 1991 m. gruodį įvyko pirmoji vieša demonstracija Hypertext '91 konferencijoje San Antonio, Texas. 1993 m. vasario mėn. buvo išleista pirmoji grafinė sąsaja Mosaic.

Mosaic buvo tokia populiari, kad po metų jos autorius Marc Andreessen išėjo iš Superlaidumo taikymų nacionalinio centro ir sukūrė kompaniją Netscape Communications Corp., kurios tikslas buvo kurti klientų, serverių ir kitą Web programinę įrangą.

1994 m. CERN ir MIT (Masačiusetso technologijos institutas) įkūrė konsorciumą, užsiimantį tolimesniu Web'o vystymu ir protokolų standartizavimu. Berners-Lee tapo direktoriumi. Nuo to laiko prie konsorciumo prisijungė daugybė universitetų ir kompanijų. Konsorciumo tinklalapis yra http://www.w3.org. Ten galima rasti visą dokumentaciją, susijusią su konsorciumo veikla.

Web yra kliento-serverio sistema. Vieną ir kitą sistemos dalį panagrinėsime plačiau.

Kliento pusė

Vartotojo požiūriu Web susideda iš pasaulinės dokumentų, vadinamų tiklalapiais, puslapiais (pages) kolekcijos. Kiekvienas puslapis gali turėti nuorodas į kitus puslapius, esančius bet kur pasaulyje. Vienu pelytės mygtuko spustelėjimu galima patekti į nurodytą puslapį ir taip peržiūrėti šimtus tarpusavyje susijusių puslapių. Sakoma, kad tokie puslapiai naudoja hipertekstą.

Puslapiai peržiūrimi programa, vadinama tinklo naršykle (browser). Žinomiausios naršyklės - Mosaic, Netscape, Internet Explorer, Opera. Naršyklė paima nurodytą puslapį, interpretuoja tekstą ir formatavimo komandas, esančias puslapyje, parodo puslapį ekrane. Eilutės, turinčios nuorodas į kitus puslapius, vadinamos hipernuorodomis (hiperlinks). Jos paprastai būna paryškintos arba išskirtos iš viso teksto kitokia spalva. Šiuolaikinės tinklo naršyklės turi daugybę galimybių, palengvinančių darbą su puslapiais.

Be teksto ir hiperteksto puslapiuose gali būti ikonos, piešiniai, paveiksliukai ir t.t. Kiekvienas iš jų gali būti susietas su kitu puslapiu. Kai kuriuose puslapiuose gali būti garso failai, videoklipai ir kt. Kai hipertekstiniai puslapiai yra papildyti kitokia "media" (terpe), tai yra vadinama hypermedia.

TAIKOMASIS LYGIS

Naršyklė gali dirbti, jeigu kompiuteris yra Internete, kadangi puslapių skaitymui reikia TCP sujungimo. Atsiųstus puslapius naršyklė užrašo į "kašę" (cache) diske. Puslapius, esančius kašėje, galima perskaityti ir atsijungus nuo Interneto.

Puslapius taip pat galima atsisiųsti elektroniniu paštu arba faksu.

Serverio pusė

Kiekvienoje Web stotyje (site) yra serverio procesas, kuris priklauso TCP 80-tosios prieigos, į kurią iš klientų (dažniausiai tinklo naršyklių) ateina užklausos susijungimui. Po susijungimo klientas siunčia vieną užklausą, o serveris siunčia vieną atsakymą. Po to sujungimas atlaisvinamas. Protokolas, kuris apibrėžia teisingas užklausas ir atsakymus, vadinamas HTTP (Hyper Text Transfer Protocol). Toliau paprastas HTTP darbo pavyzdys. Tarkime yra kreipiamasi į nuorodą, kurios vardas (URL - Uniform Resource Locator) yra http.www.w3.org/hypertext/www/TheProject.html. URL turi tris dalis: protokolo vardą (http), kompiuterio, kuriame yra reikalingas puslapis, vardą (www.w3.org) ir failo, kuriame yra puslapis, vardą (hypertext/www/TheProject.html). Veiksmai, kurie yra atliekami tarp puslapio iškvietimo, paspaudžiant pelytės klavišą, ir jo parodymo ekrane:

1. Naršyklė nustato URL.
2. Naršyklė klausia DNS www.w3.org IP adreso.
3. DNS atsako 18.23.0.23.
4. Naršyklė organizuoja TCP sujungimą su 80 prieiga kompiuteryje, kurio IP adresas 18.23.0.23.
5. Naršyklė nusiunčia GET /hypertext/www/TheProject.html komandą.
6. www.w3.org serveris siunčia failą TheProject.html.
7. TCP sujungimas atlaisvinamas.
8. Naršyklė parodo visą tekstą TheProject.html faile.
9. Naršyklė paima ir parodo visus paveiksliukus TheProject.html faile.

Dauguma naršyklių būsenos eilutėje parodo, koks veiksmas vykdomas duotu momentu. Kai darbas lėtas arba iš viso nevyksta, vartotojas mato, kad neatsako DNS, neatsako serveris arba tiesiog tinklas perkrautas.

TAIKOMASIS LYGIS

Kiekvienam paveiksliukui puslapyje, naršyklė organizuoja naują TCP sujungimą su reikalingu serveriu. Tai nėra labai efektyvu, kai viskas yra tame pačiame serveryje. Naujesnės protokolo versijos turėtų šį procesą optimizuoti.

Kadangi HTTP yra ASCII protokolas, yra gana paprasta vartotojui bendrauti su Web serveriu, naudojantis terminalu. Galima pasinaudoti Telnet programa. Tokio seanso pradžia atrodytų taip:

C: telnet www.w3.org 80
T: Trying 18.23.0.23...
T: Connected to www.w3.org
T: Escape character is '^]'
C: GET /hypertex/www/TheProject.html HTTP/1.0
C:
S: HTTP/1.0 200 Document follows
S: Mime-Version: 1.0
S: Server: Cern/3.0
S: Content-Type: text/html
S:Content-Length: 8247
.....
.....

Čia C: - kliento pranešimas; T: - tai, kas rodoma ekrane; S: - serverio pranešimas

Šis pavyzdys parodo, kad: pirmiausia susijungiama su reikalingu kompiuteriu, toliau prašoma atsiųsti puslapį, nurodant protokolą ir versiją (HTTP/1.0). 7-toje eilutėje serveris atsako, kad protokolas, kurį jis naudoja, yra toks pat, kaip ir kliento. Kodas 200 reiškia OK. Toliau eina RFC 822 MIME pranešimas ir t.t.

Ne visi serveriai supranta HTTP. Daugelis senesnių serverių naudoja FTP, Gopher ir kitus protokolus. Kadangi daug naudingos informacijos yra FTP ir Gopher serveriuose, tai Web yra sukonstruotas taip, kad ši informacija būtų prieinama Web varotojams. Vienas iš sprendimų - Web naršyklę papildyti FTP ir Gopher protokolais. Tačiau negalima visų įmanomų protokolų sudėti į naršyklę. Šiuo atveju yra naudojami proxy serveriai.

TAIKOMASIS LYGIS

Proxy serveris- tai šliuzas, kuris naudoja HTTP, bendraujant su naršykle, ir FTP ar Gopher bendraujant su serveriu. Jis priima HTTP užklausas, paverčia jas į, sakykim, FTP užklausas. Taigi, naršyklė žino tik HTTP ir to užtenka. Proxy serveris gali būti programa, veikianti tame pačiame kompiuteryje kaip ir naršyklė, bet gali būti ir nutolusiame kompiuteryje, aptarnaujančiame daugelį naršyklių.

Proxy serveriai taip pat turi daug kitų svarbių savybių. Viena iš jų - rašymas į kašę (cashing). Toks serveris visus puslapius, praeinančius per jį, deda į kašę. Kai vartotojas prašo tam tikro puslapio, pirmiausia pažiūrima ar nėra jo kašėje. Jeigu jis yra, tai jis labai greitai pateikiamas vartotojui iš kašės.

Dažnai proxy serveris yra kartu su "ugnies siena" (firewall) tame pačiame kompiuteryje. Tokiu būdu ribojamas išėjimas į Internetą arba, dažniau, priėjimas prie vietinių resursų iš Interneto.

HTTP - Hyper Text Transfer Protocol (Hiperteksto perdavimo protokolas)

Kiekviena kliento-serverio sąveika susideda iš vienos ASCII užklausos ir vieno RFC 822 MIME atsakymo. Nors dažniausiai yra naudojami TCP sujungimai, tačiau tai nėra standarto reikalavimas. Gali būti panaudoti ir kiti: pvz.: AAL5 ATM tinkluose.

Visos naujesnės HTTP versijos palaiko dvi užklausų rūšis : paprastoji užklausa ir pilnoji užklausa. Paprastoji užklausa - tai tiesiog GET eilutė, nenurodant protokolo. Šio tipo užklausos dabar beveik nenaudojamos, reikalingos senesnių versijų palaikymui.

Pilnoje užklausoje yra nurodomas protokolas ir jo versija. Pirmas žodis užklausos eilutėje yra metodo pavadinimas (komanda). Pagrindiniai metodai pateikti 37 pav. Reikalui esant, gali būti įvesti ir kiti metodai.

GET metodas prašo serverio atsiųsti puslapį MIME formate. Jei po GET užklausos yra *If-Modified-Since:* antraštė, tai serveris atsiunčia puslapį, jeigu jis buvo pakeistas po nurodytos datos.

HEAD metodas reikalauja tik puslapio antraštės. Sąlyginės HEAD užklausos nėra.

PUT metodas yra atvirkščias GET. Puslapis įrašomas, o ne nuskaitomas.

POST panašus į PUT. Tik šiuo atveju puslapis yra papildomas, o ne pilnai įrašomas. Vienu iš kitų atvejų turi būti leidimas operaciją atlikti.

TAIKOMASIS LYGIS

Metodas	Aprašas
GET	Web puslapio perskaitymo užklausa
HEAD	Puslapio antraštės nuskaitymo užklausa
PUT	Puslapio užrašymo (store) užklausa
POST	Papildyti nurodytą puslapį
DELETE	Ištrinti puslapį
LINK	Sujungia du egzistuojančius resursus
UNLINK	Nutraukia egzistuojantį sujungimą tarp dviejų resursų

37 pav. Pagrindiniai HTTP užklausų metodai

TAIKOMASIS LYGIS

Vardas	Kam naudojamas	Pavyzdys
HTTP	Hypertext'o perdavimo protokolas	http://www.vu.lt
FTP	FTP (failų perdavimo protokolas)	ftp://ftp.vu.lt/pub/readme
file	Vietinis failas	/usr/Petras/prog.c
news	Naujienų grupė	news:comp.os.unix
news	Naujienų straipsnis	news:AA013422311@news.takas.lt
gopher	Gopher	gopher://gopher.tc.tp.edu/12/libraries
mailto	Pašto išsiuntimas	mailto:petras@kt.cl.org
telnet	Prisijungimas prie nutolusio kompiuterio	telnet:/www.w3.org:80

38 pav. Kai kurie tipiniai URL

TAIKOMASIS LYGIS

DELETE trina puslapį. Turi būti autentifikavimas ir leidimas trinti.

Atsakymas į kiekvieną užklausą turi būsenos eilutę ir galimą papildomą informaciją. Būsenos eilutėje gali būti įvairūs kodai: 200 (OK), 304 (nemodifikuojamas), 400 (bloga užklausa), 403 (uždrausta).

HTTP standartai detaliai aprašo pranešimų antraštes ir kūnus. Savo struktūra jie yra labai panašūs į RFC 822 MIME.

URLs - Uniform Resource Locators
(Universali informacijos pasiekimo sistema)

URL - turi tris dalis: protokolą, kompiuterio, kuriame yra puslapis, DNS vardą, vietinį unikalų puslapio vardą. Protokolas parodo, kokiu būdu puslapis yra pasiekiamas, kompiuterio vardas - kur yra puslapis, puslapio vardas - kaip puslapis pavadintas. Kai kurių naudojamų protokolų pavadinimai pateikti 38 pav.

News protokolas leidžia Web vartotojui paimti naujienų straipsnį kaip Web puslapį. Tai reiškia, kad Web naršyklė yra ir naujienų skaitymo programa. Yra du naujienų formatai. Pirmasis reiškia naujienų grupę ir gali būti panaudotas straipsnių sąrašui atsiųsti. Antrasis reikalauja straipsnio identifikatoriaus. Naršyklė paima straipsnį, naudodama NNTP protokolą.

Gopher protokolą naudoja Gopher sistema, kuri buvo sukurta Minesotos universitete ir pavadinta pagal sporto komandos pavadinimą - Golden Gophers. Gopher atsirado keletu metų anksčiau už Web'a. Tai yra informacijos paėmimo schema panaši į Web, bet palaikanti tik tekstą. Kai vartotojas prisijungia prie Gopher serverio, jis gauna failų ir katalogų meniu, kiekvienas iš kurių gali būti surištas su kitu Gopher meniu bet kur pasaulyje.

Didelis Gopher privalumas prieš Web yra tai, kad jis gerai dirba su 25x80 ASCII terminalais ir yra labai greitas, nes dirba tik su tekstu. Yra tūkstančiai Gopher serverių visame pasaulyje. Naudojant Gopher protokolą, Web vartotojai gali pasiekti Gopher ir gauti kiekvieną meniu kaip Web puslapį.

Mailto protokolas leidžia vartotojui pasiųsti el. paštą iš Web naršyklės.

Telnet protokolas leidžia iškviesti Telnet programą.

Taigi, Web naršyklė apima visas Interneto paslaugas, naudojant tą pačią vartotojo sąsają. Specializuotos FTP, Gopher ir kitos programos darosi nebereikalingos.

TAIKOMASIS LYGIS

Nežiūrint visų šių gerų savybių URL turi ir trūkumų. URL yra nuoroda tik į vieną kompiuterį. Naudinga turėti kopijas tų puslapių, kurie yra labai dažnai lankomi, kituose kompiuteriuose. Tokiu būdu būtų galima mažinti tinklo apkrovimą. Tačiau kol kas negalima nurodyti puslapio vardo, nenurodant kur jis yra. IETF kuria sistemą URI (Universal Resource Indentifiers). Tai bus kaip URL apibendrinimas.

HTML - Hypertext Markup Language (Hiperteksto žymėjimo kalba)

HTML yra ISO standarto 8879 SGML (Standart Generalized Markup Language) panaudojimas Specializuota hipertekstui ir pritaikyta Web'ui. Šios kalbos pagalba yra nurodoma, kaip tekstas turi būti suformatuotas (šriftai, jų dydžiai, paryškinimai ir t.t.) Yra keletas HTML versijų, kurios skiriasi pagal savo galimybes. Pvz.: naujesnės versijos palaiko duomenų įvedimo formas užklausoms, senesnės nepalaiko. Naršyklių kūrėjai gali pasiūlyti savo HTML kalbos plėtinius, kuriuos galės išnaudoti tik specializuotos naršyklės.

Įvairių judančių objektų demonstravimui Web puslapiuose naudojama universali programavimo kalba Java.

Telnet protokolas

Telnet protokolas leidžia programai viename kompiuteryje (Telnet klientas) prieiti prie kito kompiuterio (Telnet serveris) resursų taip, lyg klientas būtų prijungtas prie serverio lokaliai (39 pav.)

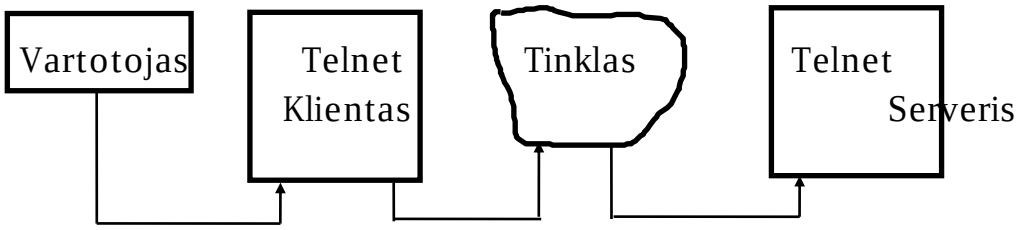
Telnet standartas remiasi virtualaus tinklo terminalo (Network Virtual Terminal, NTV) idėja. Virtualus - tai įsivaizduojamas prietaisas, turintis standartines terminalo charakteristikas. Vartotojo ir serverio prietaisai turi sutapatinti savo terminalų charakteristikas su virtualaus terminalo aprašymu.

Telnet protokolas leidžia bendraujantiems kompiuteriams susitarti dėl daugelio ryšio parametrų, naudojamų sesijos metu. Serveris ir klientas turi naudoti standartinį procedūrų rinkinį parametrams nustatyti.

Yra visa serija RFC, kur aprašyti parametrai, dėl kurių susitaria klientas ir serveris prieš pradedant sesiją. Ne visos terminalinės programos palaiko

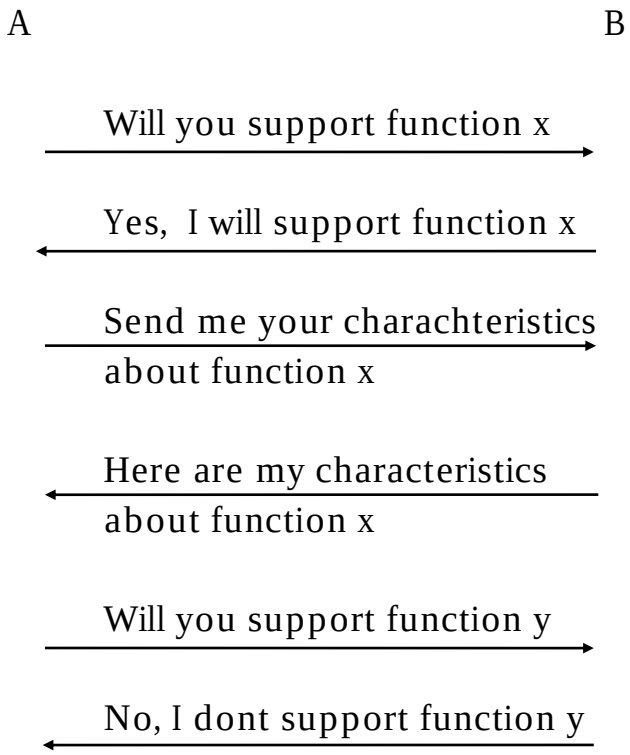
pilną parametrų sąrašą. Sesijos metu turi galioti tik tie parametrai, kuriuos palaiko abi pusės. Derybų schema pavaizduota 40 pav.

TAIKOMASIS LYGIS



39 pav. Telnet modelis

TAIKOMASIS LYGIS



40 pav. Telnet derybos dėl perdavimo parametrų.

TAIKOMASIS LYGIS

1 baitas	2 baitas	3 baitas (nebūtinas)
Interpret as Command (IAC)	Komandos kodas	Parametro, dėl kurio tariamasi, kodas

41 pav. Telnet komandos formatas

TAIKOMASIS LYGIS

Telnet duomenų apsikeitimo vienetas vadinamas komanda, kurios formatas nurodytas 41 pav.

Telnet naudoja TCP sujungimą. Po derybų tarp kliento ir serverio dėl ryšio parametrų, kliento ekrane atsiranda nutolusio kompiuterio operacinės sistemos komandinė eilutė. Įvedinėjant atitinkamas komandas, galima naudotis nutolusio kompiuterio resursais. Dažniausiai, prieš prasidedant ryšio seansui, reikia įvesti vartotojo vardą ir slaptažodį.

Paprastas failų perdavimo protokolas TFTP

TFTP (Trivial File Transfer Protocol) yra paprastas, neturi tiek daug galimybių kaip FTP, naudoja nedaug kompiuterio resursų. Dėl to juo galima naudotis, turint negalingą kompiuterį.

TFTP nereikalingi slaptažodžiai ir vartotojo identifikacijos procedūra. Jis nėra labai patikimas, nes naudoja ne TCP, o UDP. Nežiūrint to, yra tikrinamas duomenų perdavimo vientisumas, palaikomi taimeriai ir duomenų perdavimo pakartojimas, siunčiami pranešimai apie klaidas.

Siuntiklis siunčia 512 baitų bloką ir laukia patvirtinimo prieš siunčiant kitą bloką. Tai vadinama flip-flop protokolu.

Blokai yra nuosekliai sunumeruoti, o patvirtinimo lauke yra patvirtintojo bloko numeris. Paskutinis blokas yra nepilnas (0-511 baitų). Taip žymima pranešimo pabaiga.

TFTP protokolas nėra “tvirtas” įvairių trikdžių atžvilgiu. Praktiškai bet kokia problema gali nutraukti sujungimą. Dėl to jis nėra plačiai naudojamas, ypač ilgų failų siuntimui. Tačiau trumpų failų siuntimui jis yra tinkamas ir gali būti panaudotas maršrutizatoriaus konfigūracinio failo persiuntimui.

Failų perdavimo protokolas FTP

FTP (File Transfer Protocol) apibrėžia failų perdavimo procedūras tarp dviejų kompiuterių. Išsiskiria iš kitų protokolų tuo, kad naudoja du loginius sujungimus tarp kompiuterių. Vienas sujungimas yra reikalingas įėjimui (login)

į kitą kompiuterį ir naudoja Telnet protokolą. Kitas sujungimas yra naudojamas duomenų perdavimui.

TAIKOMASIS LYGIS

FTP leidžia perdavimą tarp dviejų kompiuterių, veikiančių kaip serveriai, siuntimą inicijuojant iš trečio kompiuterio (kliento). Klientas prašo leidimo siųsti duomenis iš vieno serverio į kitą. Jei prašymas patenkinamas, vienas serveris organizuoja TCP sujungimą su kitu serveriu ir perduoda failus.

Perduodamų duomenų tipai: ASCII, EBCDIC, binary (bitų srautas). Paskutinis tipas yra svarbus perduodant netekstinius failus.

FTP naudoja daug komandų pradinei identifikacijai, slaptažodžiams, vykstančio duomenų perdavimo valdymui. Taip pat yra visa serija atsakymų (reakcijų) į komandas. Atsakymai susideda iš trijų skaitmenų skaičiaus ir po jo einančio paaiškinamojo teksto. Pirmas skaitmuo turi reikšmes nuo 0 iki 5. Tai atitinka pagrindines atsakymų grupes. Kiti skaitmenys detalizuoja atsakymą. Atsakymo kodų aprašymą galima rasti RFC 959.

Operacijų seka FTP sesijos metu

Prisijungimas prie nutolusio kompiuterio (remote host login) (1). Ši operacija turi būti įvykdyta prieš pradedant duomenų perdavimą, nes reikia įsitikinti, kad vartotojas turi teisę naudotis kito kompiuterio resursais. Kaip minimum turi būti įvestas teisingas vardas ir slaptažodis. Yra FTP serverių, kurie leidžia naudotis tam tikrais resursais visiems vartotojams. Pakanka vietoj vardo įvesti "anonymous", o vietoj slaptažodžio savo el. pašto adresą.

Katalogo pasirinkimas (directory definition)(2). Dažnai reikia nurodyti katalogo, kuriame yra reikalingi failai, vardą (padaryti tą katalogą darbiniu) Tai yra padaroma CWD komanda.

Failo pasirinkimas (3). Nurodoma, kokį failą reikia perduoti. Dažniausiai yra naudojamos GET ir PUT komandos. Pirmoji kopijuoja failą iš nutolusio kompiuterio į vietinę failų sistemą. Antroji - atvirkščiai.

Perdavimo modos (režimo) nustatymas (4). Nurodoma, kaip duomenys yra organizuoti ir kaip turi būti perduodami bitai. Tam skirta keletas komandų.

Block. Šis parametras reiškia, kad failas bus perduodamas blokais.

Stream. Tai yra duomenų perdavimo moda pagal nutylėjimą. Duomenys siunčiami kaip bitų srautas, nereikalinga blokų valdymo informacija.

TYPE. Ši moda naudojama su IMAGE, ASCII ir EBCDIC parametrais.

ASCII. Šis parametras naudojamas pagal nutylėjimą. Naudojama, siunčiant tekstinius failus.

TAIKOMASIS LYGIS

EBCDIC. Naudojama siunčiant failus tarp kompiuterių, palaikančių EBCDIC kodavimą.

IMAGE. Perduodamas bitų, supakuotų į 8 bitų baitus, srautas. Dažniausiai naudojamas perduodant dvejetainius failus.

Duomenų perdavimo pradžia (5). Šis žingsnis gali prasidėti su daugeliu FTP komandų (GET, PUT ir t.t.).

Duomenų perdavimo pabaiga (6). Paprasčiausia tai galima padaryti, įvedus QUIT komandą, kuri panaikina sujungimą su nutolusiu kompiuteriu. Tačiau galima pasinaudoti komanda CLOSE, kuri nutraukia duomenų perdavimą, tačiau nepanaikina sujungimo. FTP lieka aktyvus ir kitas vartotojas gali pradėti naują FTP sesiją, pasinaudodamas OPEN komanda.

Šiuo metu yra sukurta nemažai programų, palaikančių grafinę FTP sąsają (WS-FTP, Cute FTP, FTP Explorer). Šiuo atveju praktiškai nereikia žinoti FTP komandų, norint atsisiųsti failus iš nutolusių kompiuterių. Tai žymiai palengvina darbą su gana daug galimybių turinčia FTP sistema.

IP telefonija

IP telefonija - tai ne tik balso perdavimas per IP tinklus. Į šią sąvoką taip pat įeina faksas ir multimedija per IP. Tačiau šiuo metu daugiausia dėmesio yra skiriama balso perdavimui Internete. IP telefonijos rinka stipriai plečiasi. 42 pav. parodyta rinkos struktūra su prognoze sekantiems metams. Iš lentelės matyti, kad ši duomenų perdavimo sritis yra aktuali ir ryšių operatoriams.

IP telefonija debiutavo 1995 m. kai firma VoicalTec pristatė klientą Internet Phone. Iš pradžių tai buvo tik galimybė kalbėtis vietiniame tinkle. Greitai programos galimybės buvo išplėstos. Dviejų multimedijinių kompiuterių vartotojai galėjo kalbėtis per Internetą. Šios sistemos trūkumas buvo tai, kad, prieš pradedant tokį pokalbį, reikėjo paskambinti kolegai telefonu ir paprašyti, kad jis įsijungtų kompiuterį ir paleistų reikalingą programą. Be to, įvairių gamintojų programinė įranga dažnai buvo nesuderinama. Dėl šių ir dėl kitų priežasčių IP telefonija buvo tik egzotinis būdas pakalbėti su pažystamu, naudojant kompiuterį.

TAIKOMASIS LYGIS

Metai	Mėgėjai	Korporacijos	Ryšių operatoriai
1995	94%	2%	4%
1996	90%	3%	7%
1997	75%	5%	20%
1998	55%	10%	35%
1999	45%	15%	40%

42 pav. IP telefonijos rinkos struktūra.

TAIKOMASIS LYGIS

1996 m. buvo pagaminti pirmieji šliuzai, kurių pagalba tiek asmeninių kompiuterių, tiek paprastų telefonų vartotojai galėjo susijungti per Internetą. Šliuzai apjungia tokius tinklus su paketų komutacija kaip Internet ir tradicinius telefoninius tinklus.

Šliuzai jungiami prie telefonų tinklo iš vienos pusės ir prie Interneto iš kitos. Skambinant iš telefono į kompiuterį, telefoninis kvietimas perduodamas per telefonų tinklą į šliuzą. Jeigu telefoninis signalas analoginis, šliuzas paverčia jį skaitmeniniu, suspaudžia ir suformuoja paketus perdavimui per IP. Po to šliuzas perduoda kvietimą reikalingam kompiuteriui.

Jei abu galiniai įrenginiai yra telefonai, tai balso signalas per telefonų tinklą patenka į artimiausią IP telefonijos šliuzą, po to Internetu perduodamas į antrą šliuzą netoli kito abonento. Šliuzas priima kvietimą ir perduoda jį telefonų tinklu atitinkamam abonentui. Tradiciniam balso perdavimui reikia 64 kb/s, suspaustam užtenka 8-12kb/s.

Šliuzai gali būti patalpinti įvairiose vietose, tame tarpe telefoninio aptarnavimo centruose ir filialuose.

1996 m. pradžioje grupė kompanijų įkūrė forumą "Balsas per IP" (Voice over IP, VOIP). Forumo tikslas - nustatyti standartinę VOIP realizaciją, kurios prisilaikytų visi gamintojai. Forumas palaiko standartą H.223, apibrėžiantį multimedijos perdavimą per IP.

IP telefonija - tai duomenų perdavimas realiame laike per Internetą, kur galimos tinklo perkrovos, dideli vėlinimai ir paketų priėmimas ne eilės tvarka. Paprastų duomenų perdavimui tai didelės reikšmės neturi. To negalima pasakyti apie balso perdavimą realiame laike. Paketai turi pasiekti adresatą eilės tvarka ir minimaliai vėluodami. Jeigu balso paketai kelyje praeina per daugelį maršrutizatorių, tai vėlinimo tikimybė išauga.

Tyrimai parodė, kad žmogaus ausis nepakenčia vėlinimų didesnių už 300-600 ms. Laikoma, kad 150 ms. vėlinimas - tai labai gera kokybė. Viršijus 300 ms. balsas pradeda skambėti taip, kaip esant blogam koriniam ryšiui.

Jeigu vėlinimų mažinimas gerina balso atgaminimo kokybę, tai reikalingos pralaidos užtikrinimas garantuoja paketų pristatymą adresatui. Kadangi IP telefonijos duomenų srautas yra susimaišęs su paprastų duomenų

srautu, tai tinklų administratoriai turi paskirstyti tinklų pralaidą taip, kad balso duomenys turėtų prioritetą.

TAIKOMASIS LYGIS

Vienas iš būdų pralaidai valdyti yra tokių protokolų kaip RSVP (Resource Reservation Protocol) panaudojimas.

Kadangi patvirtinimai ir siuntimo pakartojimai perduodant balsą yra nepageidautini, tai yra daug ekonomiškiau vietoj TCP protokolo naudoti UDP. Dėl to daugelis kompanijų kaip pagrindinį transportinį protokolą balsui perduoti naudoja UDP. Tradiciškai UDP srautas turi didesnį prioritetą, be to, maršrutizatorius nesunku atitinkamai sukonfigūruoti.

Viena iš IP telefonijos vystymosi priežasčių yra ta, kad tokiu būdu žymiai pigiau atsieina tarptautiniai pokalbiai telefonu. Tuo ypatingai suinteresuotos didelės kompanijos su filialais įvairiuose pasaulio kraštuose. IP telefonijos vystymasis turi duoti naudos ne tik vartotojams, bet ir ryšių operatoriams ir Internet paslaugų tiekėjams. Jau atsirado nauja paslaugų tiekėjų rūšis Internet telefonijos paslaugų tiekėjai (Internet Telephony Service Provider, ITSP).

IRC (Internet Relay Chat - kalbėjimas Internete)

Pirmasis IRC parašė Jarkko Oikarinen 1988 metais. Po sėkmės Suomijoje, IRC buvo pradėtas naudoti visame pasaulyje. IRC - tai daugiavartotojiška pokalbių sistema, kurioje žmonės gali susitikti kanaluose (kambariai, kuriose dažniausiai kalbama kažkokia tema), kalbėtis grupėmis arba privačiai su kažkuo. IRC nėra jokių taisyklių, ribojančių vartotojų kiekį arba sukurtų kanalų skaičių.

Paprastas vartotojas turi naudoti IRC kliento programą, kurios pagalba jis gali prisijungt prie IRC serverio. Visi serveriai yra sujungti tarpusavyje ir perduoda vieno vartotojo pranešimus kitam per IRC serverių tinklą. Vienas IRC serveris gali būti susijungęs su keletu kitų ir su šimtais klientų. Pasaulyje egzistuoja keletas didesnių ir mažesnių IRC tinklų. Didžiausi iš jų : Efnet, IRCnet, Undernet ir Dalnet dažniausiai aptarnauja po 20000 klientų vienu metu. Daugelis kitų IRC tinklų yra mažiau apkrauti, tačiau jie yra kiek stabilesni ir paprastesni.

IRC klientas nuskaito vartotojo komandas ir jo pokalbių frazes ir viską apdoroja. Parinkinėdamas komandas IRC klientas įvykdo reikiamus veiksmus

ir ką reikia perduoda IRC serveriui. IRC serveris gali tuo pačiu metu aptarnauti daugelį kitų žmonių.

TAIKOMASIS LYGIS

Serveris laiko informaciją apie sukurtus kanalus ir žmones juose, kaip ir kitą reikalingą informaciją. Jis taip pat atsakingas už duomenų maršrutizavimą kitiems vartotojams. IRC tinklas, savo ruožtu, sudarytas iš daugelio tarpusavyje sujungtų serverių.

Yra daug skirtingų IRC klientų : WSIRC, InteRfaCe, ChatMan, Virc, tačiau patys populiariausi mIRC ir Pirch.

Pirmą kartą paleidus IRC klientą, vartotojas turi įvesti duomenis apie save ir IRC serverį, į kurį jis norės jungtis. IRC klientas taip pat gali paprašyti įvesti prieigą (dažniausiai IRC serverio prieiga yra 6667, tačiau ji gali būti nuo 6665 iki 6670.Būna ir kitokių, Pvz.: 7000, 7777), slaptažodį (password, jei manoma kad į serverį įleidžiami visi norintys vartotojai, tai šis laukas paliekamas tuščias. Dažniausiai serveriai nereikalauja jokio slaptažodžio), tikrąjį savo vardą (Real Name, įvesti tikrą vardą nėra būtina), el. pašto adresą (nėra būtina), pseudonimą (nickname, galima pasirinkti bet kokį pseudonimą, neilgesnį už 9 simbolius. Jame patartina nenaudoti specialiųjų ASCII simbolių. Kartais kitas žmogus naudoja tą patį pseudonimą, tokiu atveju vartotojas turi pakeisti savo pseudonimą), IP adresą. Dažniausiai IRC kliente šias opcijas galima rasti File/Setup meniu.

Kai vartotojas prisijungė prie IRC serverio jis gali įvedinėti komandas. Visos IRC komandos prasideda ženklu "/" ir daugumą sudaro vienas žodis. Parašius : */help* - serveris vartotojui parodys trumpą informaciją apie komandas, */name* - vartotojui bus parodytas kitų vartotojų pseudonimų sąrašas, */list* - vartotojui bus parodytas visų kanalą sąrašas. Vieno kanalo aprašas atrodys maždaug taip :

"*#mirc* 27 Geriausiais kanalas!"

Čia *mirc* yra kanalo vardas, o "#" yra kanalo vardo prefiksas. 27 - reiškia žmonių, esančių kanale, skaičių. "Geriausias kanalas" yra #*mirc* kanalo tema (topic).

IRC kanalas - tai yra vieta kur vyksta grupiniai žmonių pokalbiai. Žmonės gali prisijungti prie to pačio kanalo ir "pamatyti" vienas kitą (pamatyti - reiškia pamatyti pseudonimus visų žmonių esančių kanale). Kanale gali būti

labai daug vartotojų, tai priklauso nuo pašnekesio temos ir laiko. Kanalai taip pat gali būti gana chaotiški arba ramūs, taip pat jie gali būti atidaryti visiems, o kai kurie tik siauram draugų ratui. IRC kanalai yra dinamiški ta prasme, kad bet kas gali sukurti naują kanalą, o kanalas uždaromas kai iš jo išeina paskutinis vartotojas.

TAIKOMASIS LYGIS

Visų IRC kanalų vardai prasideda # arba &. # kanalai prieinami visiems vartotojams, o & kanalai tik serverio, kuriame jie yra sukurti vartotojams. Kai kurie vartotojai IRC kanale turi prefiksą @ prieš savo pseudonimą. Jie yra to kanalo operatoriai. Kanalo operatorius - tai žmogus kuris valdo kanalą, jis nusprendžia ar dalinsis savo valdžia su kitais (ar suteiks kam nors kanalo operatoriaus statusą). Pirmasis vartotojas, užėjęs į kanalą, automatiškai gauna kanalo operatoriaus statusą. Jeigu vartotojas nori išeiti iš kanalo, jis turi pasinaudoti komanda "*/part*". 1 pav. pavaizduotos visos pagrindinės IRC komandos.

Dažnai IRC kanaluose pasitaiko botų (bot - sutrumpinimas nuo angliško žodžio robot). Botas - tai yra tam tikra komandų seka, leidžiama iš kliento, arba atskira programa, parašyta perl, C arba kokia nors kita programavimo kalba. Botą būtų galima sulyginti su žmogum, kuris visą parą sėdi IRC kanale (tuo neleisdamas kanalui užsidaryti) ir gali vykdyti visas IRC komandas. Kiekvienas botas turi savo šeimininką, kuris jį gali visapusiškai valdyti.

TAIKOMASIS LYGIS

Komanda	Trumpas aprašymas
/help	Parodo bendrą žinyną arba informaciją apie nurodytą komandą. Pvz.: "/help part"
/list	Parodys visų kanalų sąrašą
/join	Prisijungs prie nurodyto kanalo. Pvz.: "/join #smagu"
/part	Išeis iš nurodyto kanalo. Pvz.: "/part #lietuva"
/quit	Atsijungs nuo IRC serverio
/nick	Pakeisti savo pseudonimą. Pvz.: "/nick naujas"
/away	Visiems kurie į jus kreipiasi, praneša kad jūsų nėra. Pvz.: "/away pietauju..."
/invite	Pakviečia kažką į nurodytą kanalą. Pvz.: "/invite #smagu"
/kick	Išmeta nurodytą asmenį iš kanalo. Šia komandą gali vartoti tik kanalo operatoriai. "/kick naujas"
/topic	Parodys nurodyto kanalo temą. Pvz.: "/topic #smagu"

1 pav. Pagrindinės IRC komandos

TAIKOMASIS LYGIS

ICQ - (I Seek You - Aš ieškau jūsų)

Lapkričio 15 d., 1996-ais metais bendra Amerikos ir Izraelio kompanija Mirabilis išleido naują programą, skirtą žmonių bendravimui per Intreneta - ICQ. Specialistai, iškart pavadino šią programą revoliucine. Ši programa suteikia vartotojui galimybę bendrauti su kitais žmonėmis iš viso pasaulio realiame laike (On-line), keistis žinutėmis, siųsti failus ir Web adresus (URL) ir netgi žaisti. Programa veikia vizualiame rėžime ir neužima daug operatyvinės atminties. Tuo metu, kai vartotojas dirba su kitomis programomis, ICQ paskelbia, kas iš jo draugų šiuo metu yra On-line, ar atėjo naujų laiškų į jo el. pašto dėžutę ir t.t.

Taip pat ICQ turi dar vieną labai gerą funkciją - tai paieška naujų draugų per Interneta! Galima ieškoti pagal el. pašto adresą, vardą, pavardę arba pseudonimą (nickname) arba ICQ numerį, kuris yra vadinamas UIN. Šį numerį gauna kiekvienas užsiregistravęs į ICQ tinklą vartotojas.