

RETELE DE CALCULATOARE

1. Arhitectura rețelelor

TIPURI DE REȚELE:

- rețele locale
- rețele metropolitane
- rețele larg răspandite geografic
- rețele fără fir
- rețele casnice
- Inter-rețele

2. Tipuri de rețele (cu difuzare, punct la punct)

DUPA TRANSMISIE:

I-REȚEA PUNCT LA PUNCT:

Rețele punct la punct, care dispun de mai multe conexiuni între calculatoarele rețelei, ideal fiind cazul când fiecare calculator este legat de toate celelalte. Cum fizic (și nu numai) acest lucru este de obicei greu de realizat, rețeaua dispune de conexiuni între anumite calculatoare. Transmiterea mesajului se face prin parcurgerea unui traseu ce conține unul sau mai multe calculatoare; uneori există mai multe trasee posibile și atunci intervin algoritmi de dirijare care caută traseele cele mai scurte (sau cele mai ieftine, etc.).

II-REȚEA CU DIFUZARE:

Rețelele cu difuzare, care au un singur canal de comunicație care este accesibil tuturor calculatoarelor din rețea. La nevoie, fiecare calculator transmite mesaje scurte numite pachete, primite de toate celelalte calculatoare; o parte a mesajului (numită câmp de adresă) specifică destinatarul mesajului. Calculatoarele din rețea așteaptă mesaje, iar când un pachet apare în rețea, el este prelucrat de calculatorul care și-a recunoscut adresa, dar este ignorat de celelalte calculatoare din rețea. Această operație poartă numele de transmitere de mesaje.

3. Rețele locale, rețele metropolitane, rețele larg răspandite geografic (topologie, caracteristici)

REȚELE LOCALE:

Rețelele locale (Local Area Networks), denumite în general LAN-uri, sunt rețele private localizate într-o singură clădire sau într-un campus de cel mult câțiva kilometri.

LAN-urile se disting față de celelalte rețele prin trei caracteristici :

❖ 1. Marime

LAN-urile au dimensiuni restrânse, ceea ce înseamnă că timpul de transmisie în cazul cel mai defavorabil este limitat și cunoscut dinainte. Totodată, se simplifică administrarea rețelei.

❖ Tehnologie de transmisie

LAN-urile utilizează frecvent o tehnologie de transmisie care constă dintr-un singur cablu la care sunt atașate toate mașinile, așa cum erau odată cablurile telefonice comune în zonele rurale.

❖ Topologie

La rețelele locale sunt raspindite trei topologii majore: stea (toate calculatoarele sunt conectate la un calculator central), magistrala (bus, denumita și liniara, toate calculatoarele sunt conectate la un singur cablu-magistrala) și inel (fiecare calculator este conectat la doua calculatoare adiacente cu câte un segment de cablu, toate calculatoarele cu legaturile respective formînd un cerc fizic, fiecare mesaj este trimis într-un singur sens)

RETELE METROPOLITANE

O rețea metropolitană (Metropolitan Area Network), sau MAN (plural: MAN-uri) deservește un oraș. Cel mai bun exemplu de MAN este rețeaua de televiziune prin cablu disponibilă în cele mai multe orașe. Aceste rețele folosesc cel mai des tehnologia fără fir (wireless) sau fibra optică pentru a crea conexiuni. Standardul actual de comunicare al rețelei metropolitane este DQDB (de la Distributed-Queue Dual-Bus).

RETELE RASPANDITE LARG GEOGRAFIC:

În informatică, o rețea de arie largă (denumită și WAN, după eng. Wide Area Network) este orice rețea care conectează orașe, regiuni sau țări. De obicei rețelele de arie largă includ linii de telecomunicație publice și elementele de legătură și conectare necesare.

În majoritatea WAN-urilor, subrețeaua este formată din doua componente distincte:

- ❖ **liniile de transmisie** (transporta bitii între mașini: pot fi alc din fire de cupru, fibra optică sau chiar legături radio)
- ❖ **elementele de comutare**. (calculatoare specializate folosite ptr a conecta doua sau mai multe linii de transmisie)

4. Niveluri, protocoale, modelul de referință OSI (8 niveluri); descriere, caracteristici

NIVELURI:

❖ NIVEL FIZIC:

Funcție : transmitere a sirurilor de biți pe un canal de comunicație

Principalele probleme:

- codificarea zerourilor și a unităților
- stabilirea și desfășurarea conexiunilor fizice
- modul de transmisie (semiduplex sau duplex) etc.

Exemplu: transmiterea pe o linie telefonică

❖ Legătura de date

Funcție : realizează o comunicare sigură și eficientă între două noduri adiacente (conectate printr-un canal fizic de comunicație)

Probleme:

- Incadrare
- Transmisie transparentă
- Control erori
- Control flux
- Management legătură

❖ NIVEL RETEA

Funcție : dirijarea pachetelor transmise între oricare două noduri

Probleme:

- calculul tabelelor de dirijare
- alegerea legăturii următoare (dirijarea) – adresarea
- evitarea congestiei

❖ NIVEL TRANSPORT

Funcție:

- asigurarea unui transfer de date corect, eficient între sistemul sursă și sistemul destinatar

Oferta:

- un transfer sigur al datelor, chiar cu o rețea nesigură;
 - interfață uniformă pentru utilizatori, independent de tipul subrețelei utilizate.

Separa două categorii de nivele:

- furnizorul serviciilor de transport (nivele 1-4)
- utilizatorul serviciilor de transport (nivele 5-7)

Probleme:

- gestiunea conexiunilor
- transferul datelor
- controlul fluxului
- adresarea

❖ NIVEL SESIUNE

- Funcții:

- Gestiune servicii care se adaugă transportului datelor
- ControlDialog – client-server – peer-to-peer
- Sincronizare
- Gestiune activități
- Bazate pe gestiune jetoane (Tokens)

❖ NIVEL PREZENTARE

- Conversia datelor:
 - reprezentarea datelor transmise – sintaxa abstractă
 - sintaxa de transfer
- Compresie
- Securitate

❖ NIVEL APLICATIE

- Servicii comune unor categorii de aplicații:
 - Mesagerie
 - Transfer Fișiere
 - Terminal Virtual
 - Serviciu Directoare

5. Nivelul legatura de date

❖ Nivelul legatura de date

- **FUNCTII :**

- a) incadrarea(metode):
 - Caractere de control (BSC);
 - Numararea caracterelor(DDCMP);
 - Indicatori de incadrare(HDLC);
- b) transmisia transparenta;
- c) controlul erorilor;
- d) controlul fluxului;
- e) gestiunea legaturii;

6. Aspecte (caracteristici) ale proiectarii nivelului legaturii de date

Nivelul legătură de date are un număr de funcții specifice pe care trebuie să le îndeplinească.

Aceste funcții includ:

- Furnizarea unei interfețe bine-definite către nivelul rețea.
- Tratarea erorilor de transmisie
- Reglarea fluxului cadrelor în așa fel, încât receptorii lenți să nu fie inundați de către emițători rapizi.

Pentru a îndeplini aceste scopuri, nivelul legătură de date **primește pachete de la nivelul rețea**, pe care le încapsulează în cadre în vederea transmiterii. Fiecare cadru conține un antet, un câmp de informație utilă pentru pachet și încheiere . Gestionarea cadrelor reprezintă esența a ceea ce face nivelul legătură de date.

7. Detectarea si corectarea eoririlor (Hamming+CRC)

❖ PROTOCOL SIMPLEX FARA RESTRICII:

Cea mai simpla varianta de protocol are in vedere urmatoarele consideratii:

- a) utilizatorul **A** vrea sa transmita date lui **B** folosind o legatura sigura, simplex;
- b) **A** reprezinta o sursa inepuizabila de date, astfel incit transmitatorul nu trebuie sa astepte niciodata la preluarea datelor pentru transmisie;
- c) **B** reprezinta un consumator ideal, care preia imediat datele de la receptor ori de cite ori acesta i le da;
- d) canalul fizic de comunicatie este fara erori.

❖ PROTOCOL SIMPLEX START-STOP.

In a doua varianta, consideram in continuare canalul fara erori, dar utilizatorul B nu poate accepta date in orice ritm. Este necesar controlul fluxului, care in acest caz se realizeaza printr-o reactie a receptorului catre transmitator, permitindu-i acestuia din urma sa transmita urmatorul cadru. Reacția are forma unui cadru fictiv.

❖ PROTOCOALE CU FEREASTRA SUPRAUNITARA DE TRANSMISIE:

In cazul unor canale cu intirzieri mari de transmisie, așteptarea confirmării fiecarui cadru înainte de transmiterea urmatorului poate conduce la o eficienta scazuta. Astfel, la un canal cu o intirziere de propagare dus-intors de 500 msec si o viteza de 50 Kbps, transmiterea unui cadru de 1000 de biti dureaza efectiv doar 20 de

milisecunde din cele 520 necesare ciclului complet de transmitere si receptie a confirmarii. Erorile nu pot fi sesizate si corectate imediat de transmitator.

❖ **PROTOCOL CU RETRANSMITERE NESELECTIVA:**

Fereastra maxima a transmitatorului poate fi de MaxSecv cadre, desi exista $\text{MaxSecv}+1$ numere de secventa distincte. Justificam aceasta restrictie prin urmatorul scenariu, in care presupunem ca $\text{MaxSecv}=7$:

- a. Transmitatorul trimite cadrele 0..7;
- b. Toate cadrele sint receptionate si confirmate;
- c. Toate confirmarile sint pierdute;
- d. Transmitatorul retrimite la time-out toate cadrele;
- e. Receptorul accepta duplicatele.

Reglare a fluxului de date: cind utilizatorul are un pachet de transmis el provoaca un eveniment "ReteaPregatita"; cind o stație are MaxSecv cadre neconfirmate, ea interzice rețelei sa-i transmita alte pachete prin procedura "DezactivRetea"; cind numarul de pachete scade, ea anuleaza restrictia prin functia "ActivRetea".

❖ **PROTOCOL CU RETRANSMITERE SELECTIVA :**

- a. Transmitatorul trimite cadrele 0..6
- b. Cadrele sint receptionate si fereastra receptorului devine 7, 0, 1, 2, 3, 4, 5
- c. Toate confirmarile sint pierdute
- d. Transmitatorul retrimite cadrul 0 la time-out
- e. Receptorul accepta cadrul 0 aflat in fereastra si cere cadrul 7
- f. Transmitatorul afla ca toate cadrele sale au fost transmise corect si trimite cadrele 7, 0, 1, 2, 3, 4, 5
- g. Receptorul accepta cadrele, cu exceptia lui 0, pentru care are deja un cadru eceptionat. Ca urmare, ignora acest cadru, luind in locul lui dulpicatul cadrului 0 anterior.

Eroarea se produce deoarece ferestrele succesive ale receptorului au numere de secventa comune, neputindu-se face diferența între cadrele noi si duplicatele celor vechi.

8. Protocoale elementare pentru legatura de date (simplex fara restrictii, stop and wait, protocol cu confirmare si retransmitere) *Protocoale cu fereastra glisanta si protocoale cu revenire cu n pasi (Go back n).

❖ **PROTOCOALE CU FEREASTRA GLISANTA**

Transmitatorul mentine o lista cu numerele de secventa ale cadrelor transmise dar neconfirmate, alcatuind fereastra transmitatorului. Cind trebuie transmis un cadru, i se ataseaza numarul de secventa urmator (fereastra se maresteste). Cind se receptioneaza o confirmare, marginea inferioara a ferestrei se deplaseaza cu o unitate. Cadrele se pastreaza de transmitator pina la confirmarea lor.

Receptorul pastrează o lista cu numerele de secventa ale cadrelor pe care le poate accepta (fereastra receptorului). Cadrele din fereastra sint acceptate, celelalte sint ignorate. Cînd soseste un cadru cu numar de secventa egal cu marginea inferioara, fereastra este deplasata cu o pozitie (prin modificarea ambelor margini), pastrindu-se dimensiunea constanta.

❖ PROTOCOALE GO BACK N:

Există două moduri de bază de tratare a erorilor în prezenta benzii de asamblare. Un mod, nUnit revenire cu n pași (eng.: go back n), este ca receptorul să elimine pur și simplu cadrele care urmează, netrimțând confirmări pentru cadrele eliminate. Această strategie corespunde unei ferestre de recepție de dimensiune 1. Cu alte cuvinte, nivelul legatură de date refuză să accepte orice cadru exceptându-l pe următorul care trebuie livrat către nivelul rețea. Dacă fereastra emitatorului se umple înainte expirării contorului de timp, banda de asamblare va începe să se golească. În cele din urmă, timpul emitatorului va expira și se vor retransmite toate cadrele neconfirmate, în ordine, începând cu cadrul pierdut sau modificat. Dacă rata erorilor este mare, această abordare poate risipi o mare parte din lățimea de bandă.

9. Nivelul rețea.

Funcțiile nivelului rețea sunt următoarele:

1. dirijarea unitatilor de date (pachete) în nodurile de pe calea dintre sursă și destinație;
2. evitarea congestiei rețelei, prin supraincercarea anumitor legături;
3. reglementarea comunicării între surse și destinații aflate în rețele diferite interconectate.

Legatură de date asigură comunicarea corectă a cadrelor între oricare două noduri adiacente ale unei rețele de calculatoare. Transferul datelor între două noduri neadiacente utilizează mai multe legături, puse cap la cap. Rolul de releu între aceste legături este îndeplinit de nivelul rețea. Unitatea de date caracteristică acestui nivel este **pachetul**.

Principala funcție a **nivelului rețea** este dirijarea pachetelor transmise între oricare două noduri, pe cai convenabil alese. În acest scop, fiecare pachet recepționat de un nod este inspectat, determinându-se nodul destinatar. Se alege apoi legatură convenabilă, pachetul fiind transmis în continuare pe această legatură. În cazul în care legatură este ocupată, pachetul este pus într-o coadă de așteptare asociată legăturii, urmând a fi transmis mai târziu.

Pentru a realiza alegerea legăturii următoare, fiecare nod folosește o tabelă de dirijare V , fiecare intrare $V[i]$ specificând vecinul cărui i se transmite pachetul destinat nodului i . Conținutul tabelelor de dirijare (sau echivalent, setul cailor de dirijare) este stabilit prin algoritmi (sau politici) de dirijare,

O altă funcție importantă a serviciului rețea este furnizarea unui mecanism uniform de adresare pentru nivelul transport. Acesta ține cont de diversitatea rețelelor de comunicație (telefonice, telex, publice de date, ISDN etc.) aflate în proprietatea sau gestiunea unor autorități diferite, și de gruparea geografică a rețelelor pe țări, continente sau alte unități teritoriale. Formatul adreselor punctelor de serviciu de rețea are lungime variabilă și cuprinde câmpurile următoare:

- a. identificatorul autorității și formatului,
- b. identificatorul domeniului,
- c. adresă în domeniu.

10. Aspecte (caracteristici, cerințe) ale proiectării nivelului rețea

❖ Cerințele de proiectare ale nivelului rețea

- a. comutarea de pachete de tip Memorează-si-Retransmite(Store and Forward)
- b. servicii furnizate nivelului transport
- c. implementarea serviciului neorientat pe conexiune
- d. implementarea serviciilor orientate pe conexiune

a. Componente majore:

- echipamentul companiei de telecomunicații (routere conectate prin linii de transmisie)
- echipamentul clientului

Acest echipament este folosit după cum urmează:

- ✱ O gazdă care are de transmis un pachet îl transmite celui mai apropiat router, fie în aceeași rețea LAN, fie printr-o legătură punct la punct cu compania de telecomunicații.
- ✱ Pachetul este memorat acolo până ajunge integral, astfel încât să poată fi verificată suma de control.
- ✱ Apoi este trimis mai departe către următorul router de pe traseu, până ajunge la destinație, unde este livrat.
- ✱ Acest mecanism reprezintă comutarea de pachete de tip memorează-și-retransmite.

b. Serviciile nivelului rețea au fost proiectate având în vedere următoarele scopuri:

1. Serviciile trebuie să fie independente de tehnologia routerului.
2. Nivelul transport trebuie să fie independent de numărul, tipul și topologia routerelor existente.
3. Adresele de rețea disponibile la nivelul transport trebuie să folosească o schemă de numerotare uniformă, chiar în cadrul rețelelor LAN și WAN

Obiectivele fiind stabilite, proiectantul nivelului rețea are o mare libertate în a scrie specificațiile detaliate ale serviciilor oferite nivelului transport. Această libertate degenerază adesea pareri diferite între două tabere opuse.

Problema centrală a discuției este dacă nivelul rețea trebuie să furnizeze:

1. servicii orientate pe conexiune
2. servicii neorientate pe conexiune

c. Implementarea serviciului neorientat pe conexiune

Dacă este oferit un serviciu neorientat pe conexiune, atunci pachetele sunt trimise în subrețea individual și dirijate independent de celelalte.

- ✱ Nu este necesară nici o inițializare prealabilă.
- ✱ În acest context, pachetele sunt numite frecvent datagrame (datagrams) (prin analogie cu telegramele), iar subrețeaua este numită subrețea datagramă (datagram subnet).

Dacă este folosit serviciul orientat conexiune, atunci, înainte de a trimite pachete de date, trebuie stabilită o cale de la routerul sursă la routerul destinație.

- ✱ Aceasta conexiune este numita VC - virtual circuit, prin analogie cu circuitele fizice care se stabilesc în sistemul telefonic, iar subrețeaua este numita subrețea cu circuite virtuale (virtual-circuit subnet)

d. Implementarea serviciilor orientate pe conexiune

Pentru serviciile orientate conexiune, avem nevoie de o subrețea cu circuite virtuale Să vedem cum funcționează:

- ✱ Ideea care se stă la baza circuitelor virtuale este evitarea alegerii unei noi căi (rute) pentru fiecare pachet trimis
- ✱ În schimb, atunci când se stabilește o conexiune, se alege o cale între mașina sursă și mașina destinație, ca parte componentă a inițializării conexiunii și aceasta este memorată în tabelele routerelor.
- ✱ Acea cale este folosită pentru tot traficul de pe conexiune, exact în același mod în care funcționează sistemul telefonic.
- ✱ Atunci când conexiunea este eliberată, este închis și circuitul virtual.
- ✱ În cazul serviciilor orientate conexiune, fiecare pachet poartă un identificator care spune cărui circuit virtual îi aparține.
- ✱ Pentru evitarea conflictelor routerele trebuie să poată înlocui identificatorii de conexiune în pachetele care pleacă. În unele contexte, aceasta se numește comutarea etichetelor (label switching).

11. Algoritmi de dirijare (calea cea mai scurtă, inundarea, dirijarea centralizată, dirijarea izolată, dirijarea distribuită, dirijarea ierarhică)

❖ Dirijare pe calea cea mai scurtă

Să începem studiul algoritmilor de dirijare posibili cu o tehnică des utilizată în multe forme deoarece este simplă și ușor de înțeles. Ideea este de a construi un graf al subrețelei, fiecare nod al grafului fiind un router, iar fiecare arc al grafului fiind o linie de comunicație (numită adesea legătură). Pentru a alege o cale între o pereche dată de routere, algoritmul trebuie să găsească în graf calea cea mai scurtă dintre ele

- O modalitate de a măsura lungimea căii este numărul de salturi
- O altă metrică este distanța geografică în kilometri sunt posibile multe alte metrici în afară de salturi și distanța geografică

Se cunosc mai mulți algoritmi pentru calculul celei mai scurte căi între două noduri dintr-un graf. Cel mai cunoscut este cel propus de Dijkstra (1959). Fiecare nod este etichetat (în paranteze) cu distanța de la nodul sursă până la el, de-a lungul celei mai bune căi cunoscute. Inițial nu se cunoaște nici o cale, așa că toate nodurile vor fi etichetate cu infinit. Pe măsură ce se execută algoritmul și se găsesc noi căi, etichetele se pot schimba, reflectând căi mai bune. O etichetă poate fi fie temporară, fie permanentă. Inițial toate etichetele sunt temporare. Atunci când se descoperă că o etichetă reprezintă cea mai scurtă cale posibilă de la sursă către acel nod, ea devine permanentă și nu se mai schimbă ulterior.

❖ Inundarea

Un alt algoritm static este inundarea (flooding), în care fiecare pachet recepționat este trimis mai departe pe fiecare linie de ieșire, cu excepția celei pe care a sosit. Este evident că inundarea generează un mare număr de pachete duplicate, de fapt un număr infinit dacă nu se iau unele măsuri pentru a limita acest proces. O astfel de măsură este păstrarea unui contor de salturi în antetul fiecărui pachet, contor care este decrementat la fiecare salt și care face ca pachetul să fie distrus când contorul atinge valoarea zero.

Ideal ar fi ca acest contor să fie inițializat cu lungimea căii de la sursă la destinație. Dacă emițătorul nu cunoaște lungimea căii, poate inițializa contorul la valoarea cea mai defavorabilă, adică diametrul subrețelei.

O metodă alternativă pentru limitarea inundării este identificarea pachetelor care au fost deja inundate, pentru a preîntâmpina trimiterea lor a doua oară

O variantă a algoritmului de inundare, care este și ceva mai practică, este inundarea selectivă (selective flooding). În acest algoritm ruterele nu trimit fiecare pachet recepționat pe fiecare legătură de ieșire, ci doar pe acele linii care duc aproximativ în direcția potrivită.

❖ Dirijare cu vectori de distanță

Algoritmul de dirijare cu vectori distanță (distance vector routing) presupune că fiecare ruter menține o tabelă (de exemplu un vector) care păstrează cea mai bună distanță cunoscută spre fiecare destinație și linia care trebuie urmată pentru a ajunge acolo. Aceste tabele sunt actualizate prin schimbul de informații între nodurile vecine.

În dirijarea pe baza vectorilor distanță, fiecare ruter păstrează o tabelă de dirijare conținând câte o intrare pentru fiecare ruter din subrețea. Această intrare are două părți: linia de ieșire preferată care se folosește pentru destinația respectivă și o estimare a timpului sau distanței până la acea destinație. Metrica folosită poate fi numărul de salturi, întârzierea în milisecunde, numărul total de pachete care așteaptă în cozi de-a lungul căii, sau ceva asemănător.

Dirijarea folosind vectori distanță funcționează în teorie, însă în practică are o limitare importantă: deși ea converge spre rezultatul corect, o face foarte lent. În particular, ea reacționează **rapid** la veștile bune, dar foarte **lent** la cele rele.

❖ Dirijare folosind starea legăturilor

Ideea algoritmului bazat pe starea legăturilor este simplă și poate fi formulată în 5 puncte. Fiecare ruter trebuie să facă următoarele:

- a. Să descopere care sunt vecinii săi și afle adresele de rețea ale acestora.
- b. Să măsoare întârzierea sau costul până la fiecare din vecinii săi.
- c. Să pregătească un pachet prin care anunță pe toată lumea că tocmai a terminat de cules datele despre vecini.
- d. Să trimită acest pachet către toate celelalte rutere.
- e. Să calculeze cea mai scurtă cale spre fiecare ruter.

❖ Dirijarea centralizata

Deficiențele acestei metode sunt determinate de:

- vulnerabilitatea rețelei, dependența de funcționarea centrului de control (se recurge la dublarea lui);
- supraîncărcarea traficului prin transmiterea rapoarelor și a tabelelor de dirijare;
- utilizarea în noduri, în anumite perioade, a unor tabele necorelate, datorită recepției la momente de timp distincte a noilor tabele.

❖ Dirijarea izolata

Algoritmul "cartofului fierbinte" (hot potato). Pachetul recepționat de nod este plasat în coada cea mai scurtă.

O variantă ia în considerare lungimea cozilor anumitor linii, selectate conforme cailor celor mai scurte.

❖ Dirijarea delta

Este o combinație a politicilor izolată și centralizată. Comutatoarele trimit rapoarte unui centru de control care calculează cele mai bune rute. Caile sunt echivalente dacă lungimile lor diferă între ele cu o valoare mai mică decât un delta specificat.

În algoritmul inundării, pachetul este transmis pe fiecare legătură, cu excepția celei de origine. Copiile sunt distruse după traversarea unui anumit număr de noduri. Deși nepractic, algoritmul este folosit în aplicații militare (datorită robusteții sale) sau în comparații de performanță cu alte tehnici (deoarece are un timp de întârziere minim).

❖ Dirijarea distribuita

O variantă modificată a algoritmului lui Dijkstra calculează drumurile minime de la toate nodurile către o anumită destinație. Ea conduce în mod natural la o variantă descentralizată

Deși algoritmul este convergent, asigurând găsirea drumurilor minime într-un număr finit de pași, el poate fi utilizat doar pentru datagrame, din cel puțin două motive:

- sunt posibile modificări ale cailor, pe durata transmiterii pachetelor, astfel încât pachetele trimise pe o cale pot ajunge pe o altă cale;
- pe durata intervalului de convergență, algoritmul nu evită producerea buclor (trecerea pachetelor de mai multe ori prin același nod).

Algoritmul tratează trei evenimente distincte:

- a. adăugarea unei noi legături;
- b. sesizarea modificării lungimii unei linii;
- c. primirea unui mesaj de control de la un nod vecin.

❖ Dirijarea ierarhica

Se utilizează pentru rețele de mari dimensiuni la care tabelele de dirijare ar fi voluminoase.

Comutatoarele sunt grupate în regiuni, fiecare comutator cunoscând detaliat caile din regiunea proprie, dar necunoscând structura internă a altor regiuni. Două regiuni sunt legate prin conectarea unui anumit nod din prima regiune cu un anumit nod din a doua regiune. Tabela de dirijare se poate reduce, ea având câte o intrare pentru fiecare nod din regiunea proprie și câte o intrare pentru fiecare din celelalte regiuni.

Atunci când se folosește dirijarea ierarhică, ruterele sunt împărțite în ceea ce vom numi regiuni (regions), fiecare ruter știind toate detaliile necesare pentru a dirija pachete spre destinație în cadrul

regiunii sale, dar neștiind nimic despre organizarea internă a celorlalte regiuni. Când rețele diferite sunt interconectate, este natural să privim fiecare rețea ca pe o regiune, pentru a elibera ruterele dintr-o rețea de sarcina de a cunoaște structura topologică a celorlalte.

❖ Dirijarea cu difuzare

Dirijarea unui pachet către toate celelalte noduri se poate face prin tehnica inundării. Pentru a evita degradarea performanțelor rețelei se poate recurge la algoritmi descriși în continuare.

În dirijarea multidestinație, pachetul conține o listă cu adresele destinatarilor. Când pachetul ajunge la un comutator, acesta determină, pe baza adreselor, pe ce linie trebuie să transmită în continuare pachetul, partitionând totodată lista adreselor între duplicatele transmise pe aceste linii.

În dirijarea cu difuzare se poate utiliza ca traseu orice arbore de acoperire minimal. Ca alternativă, algoritmul se poate baza pe urmărirea căilor inverse: când un pachet ajunge la un comutator, dacă el a fost recepționat pe legătura folosită de obicei pentru a transmite către sursa acestui pachet, atunci el a sosit pe calea cea mai scurtă și este de obicei prima copie recepționată de comutator. Ca urmare, ea este acceptată, iar comutatorul o transmite în continuare pe fiecare linie cu excepția celei pe care a sosit.

12. Controlul și evitarea congestiei și a blocării

❖ Algoritmi de evitare a congestiei

- ✱ Prealocarea zonelor tampon. Este aplicabilă circuitelor virtuale și constă în rezervarea uneia sau mai multor zone tampon în fiecare nod intermediar, la deschiderea circuitului. În lipsa de spațiu, se alege o altă cale sau se rejectează cererea de stabilire a circuitului.
- ✱ Distrugerea pachetelor. Dacă nu există spațiul necesar memorării, pachetul recepționat de un nod este ignorat. Deoarece prin aceasta se pot ignora pachete de confirmare, care ar duce la eliberarea spațiului ocupat de pachetele confirmate, se menține cel puțin un tampon de recepție pentru fiecare linie, permitându-se inspectarea pachetelor primite. De asemenea, se poate limita (inferior și superior) numărul zonelor tampon de transmisie ale fiecărei linii.
- ✱ Pachete de permisiune. Se inițializează rețeaua cu pachete de permisiune (în număr fix). Când un nod vrea să transmită, el capturează un pachet de permisiune și trimite în locul lui pachete de date. Receptorul regenerează pachetul de permisiune. Se garantează astfel ca numărul maxim de pachete nu depășește numărul de pachete de permisiune, fără a se asigura distribuția lor conforme necesităților nodurilor. În plus, pierderea pachetelor de permisiune conduce la scăderea capacității rețelei.
- ✱ Pachete de soc. Sunt transmise de comutatoare surselor de date pentru a micșora rata de generare a pachetelor.
- ✱ Evitarea blocării definitive. Blocarea reprezintă o situație limită a unei rețele congestionate, când lipsa de spațiu împiedică transmiterea vreunui pachet. O soluție de evitare a blocării definitive este utilizarea în fiecare nod a $m+1$ zone tampon, m fiind lungimea maximă a căilor rețelei. Un pachet sosit de la calculatorul gazdă local este acceptat în zona 0. În următorul nod trece în 1, apoi în 2 ș.a.m.d. Zona "m" a unui nod poate fi goală, poate conține un pachet pentru gazdă locală, care este livrat, sau are un pachet pentru un nod distant, care este distrus. În toate cazurile zona "m" se eliberează, putând avansa un pachet din zona "m-1", apoi "m-2" etc.

O alta varianta pastreaza pentru fiecare pachet o informatie de vechime. La comunicarea dintre doua noduri A si B putem intilni situatiile urmatoare (presupunem ca A are de transmis lui B un pachet mai vechi decit B catre A):

- B are un tampon liber si poate primi cel mai vechi pachet al lui A catre B;
- B nu are un tampon liber, dar are un pachet pentru A si poate primi, prin schimb, cel mai vechi pachet al lui A catre B;
- B nu are nici un tampon liber si nici un pachet catre A; in acest caz, B este fortat sa transmita lui A un pachet la alegere si sa primeasca cel mai vechi pachet al lui A catre B.

13. Protocolul IP (descriere)

Protocolul IP este un protocol prin care datele sunt trimise de la un calculator la altul prin intermediu Internetului.

Fiecare calculator (cunoscut sub denumirea de „gazdă”), are pe Internet cel puțin o adresă IP unică, care îl identifică între toate computerele din rețea. Când cineva trimite sau primește informații mesajul este împărțit în pachete (pana la 64 KO). Fiecare pachet cuprinde adresa expeditorului și pe cea a destinatarului.

Fiecare pachet este trimis, prima dată la un calculator-pasarelă. Calculatorul pasarelă citește destinația pachetelor și trimite pachetele către o altă pasarelă, și așa mai departe, până ce pachetul ajunge la pasarela vecină cu computerul destinatar.

Adresa IP este utilizată la nivelul programelor de prelucrare în rețea. În schimb, la nivelul utilizatorilor cu acces la Internet, identificarea calculatoarelor se face printr-un nume de gazdă gestionat de sistemul DNS.

O datagramă IP (un pachet) constă dintr-o parte de antet și o parte de text. Antetul are o parte fixă de 20 octeți și o parte opțională de lungime variabilă.

Fiecare gazdă și ruter din internet are o adresă IP, care codifică adresa sa de rețea și de gazdă. Combinația este unică: în principiu nu există două mașini cu aceeași adresă IP. Toate adresele IP sunt de 32 biți și sunt folosite în câmpurile „Adresă sursă” și „Adresă destinație” a pachetelor IP. Este important de observat că o adresă IP nu se referă la o gazdă. Se referă, de fapt, la o interfață de rețea. Cu alte cuvinte, dacă o gazdă este în două rețele, trebuie să folosească două adrese IP . Rețelele sunt dinamice și este posibil ca 2 pachete IP de la aceeași sursă să plece pe căi diferite (BGP – protocolul porților de graniță) și să ajungă la aceeași destinație. Pachetele IP (dupa cum s-a mai spus) nu au garanția că vor ajunge la destinație, acest lucru fiind lăsat în seama protocoalelor adiacente (TCP UDP etc).

14. / 15. Nivelul Transport si caracteristici ale acestuia

Nivelul transport

- Servicii furnizate
 - transfer de date eficient, sigur si cu raport cost performanta bun
 - interfața uniforma cu utilizatorii
- Caracteristici
 - servicii capăt la capăt (host to host)
 - doua tipuri de servicii:
 - orientate pe conexiune (connection oriented)
 - fără conexiune (connectionless)

Model simplificat

PRIMITIVA	Unitatea de date trimisă	Explicații
listen	(nimic)	Se blochează până când un proces încearcă să se conecteze
connect	CON. , REQ.	Încearcă să stabilească conexiunea
send	DATA	Transmite informație
receive	(nimic)	Se blochează până când primește date trimise
disconnect	DISCON. REQ.	Trimisă de partea care vrea să se deconecteze
accept	CON. ACCEPTED	Trimisa de partea care acceptă să se conecteze

Adresare:

PORT	PROTOCOL USE
21	FTP File transfer
23	Telnet
25	SMTP e-mail
69	TFTP Trivial File Transfer Protocol
79	Finger Lookup info about a user
80	HTTP World Wide Web
110	POP-3 Remote e-mail access
119	NNTP USENET news

16. Protocolul TCP

Corectitudinea segmentelor TCP

1. Suma de control (checksum) din antet - include

- antet + incarcatura segment TCP
- pseudo-antet
- adresele IP sursa si destinatie
- protocolul (6 pentru TCP)
- lungime segment TCP (include antetul)

Algoritm:

❖ la transmisie

- aduna cuvinte de 16 biti in complement fata de 1
- completeaza rezultatul
- scrie rezultatul in antet

❖ la receptie

- aduna cuvinte de 16 biti – rezultatul trebuie sa fie zero

2. Acknowledgement number :urmatorul octet asteptat

Controlul congestiei

- **Fluxul de date transmis pe o conex. TCP limitat de minimul dintre:**
 - a. dimensiunea fereastrei receptorului
 - b. capacitatea retelei (fereastra de congestie)
- **Algoritm de stabilire fereastră de congestie**
 - a. transmite un segment de dimensiune maxima pe conexiunea stabilita
 - b. se dubleaza dimensiunea (rafala de segmente) la fiecare transmisie confirmata la timp
 - c. la primul timeout se opreste procedeul si fereastra ramane la valoarea ultimei transmisii confirmate la timp (fara timeout)
- **Algoritmul de control al congestiei**
 - a. foloseste un prag (threshold)
 - b. la un timeout pragul setat la jumatate din fereastra de congestie se aplica procedeul de crestere (exponentiala) a fereastrei de congestie pana se atinge pragul
 - c. peste prag se aplica o crestere liniara (cu cate un segment maxim o data)

17. Elemente de performanta

Cresterea performantei canalelor de comunicatie (1 Gbps a devenit o realitate frecventa) a facut ca gaturile in performanta retelelor sa se mute la nivelul calculatoarelor gazda. Tratarea lor reclama modificari (uneori importante) in solutia de sistem adoptata sau in implementarea protoalelor.

Un exemplu din prima categorie este alegerea plasamentului pentru anumite module de retea, in spatiul utilizator sau in spatiul nucleu. De regula, sosirea unui pachet provoaca o comutare de la procesul aflat in executie la nucleu unde este tratata receptia si o a doua comutare de la nucleu la procesul receptor pentru livrarea datelor.

18. Nivelul Aplicatie

Trei elemente de baza:

- O schema de adresare a documentelor in Internet (URL – Uniform Resource Locator)
- Un limbaj de formatare a documentelor (HTML – HyperText Markup Language)
- Un protocol pentru transportul mesajelor specializate prin retea (HTTP – HyperText Transfer Protocol)

Interacțiunea client - server

- Browser - determina URL
- Browser - cere DNS-ului adresa IP pentru www.w3.org
- DNS - raspunde cu 18.23.0.23
- Browser - deschide o conexiune TCP la port 80 pe 18.23.0.23
- Browser - trimite o comanda GET /hypertext/www/TheProject.html

- Server www.w3.org - trimite fisierul TheProject.html
- Conexiunea TCP este inchisa
- Browser - afișează conținutul din TheProject.html
- Browser - extrage si afiseaza toate imaginile din TheProject.html (se deschide o noua conexiune TCP pentru fiecare imagine)

URL – Uniform Resource Locator

scheme://host[:port#]/path/.../[;url-params][?query-string][#anchor]

scheme protocol (http, ftp etc.)

host nume / adresa IP a serverului Web

port# numar port server Web (80 pentru http)

path calea de la radacina serverului la document

url-params pentru identificarea sesiunii

query-string valori din formular HTML

anchor referinta la un marcaj pozitional din document exemplu

http://www.situlmeu.ro/cv/test?id=8079?name=valentin&x=true#aici

HTML - HyperText Markup Language

- Definit ca o aplicatie SGML – Standard Generalized Markup Language
- Aplicatia are patru parti
 - Declaratia SGML – caractere si delimitatori
 - DTD (Document Type Definition) – constructiile de marcare valide (sintaxa)
 - Specificarea semanticii asociate
 - Instante de documente cu continut si markup

HTTP

- Protocol “stateless”
- Foloseste paradigma request/response
 - clientul si serverul comunica direct sau prin proxy-uri
 - structura mesajelor:
- linia de comanda / raspuns
- linii de antet
- linie blank
- corp mesaj

19.DNS, mail, www, ftp (o aplicatie la alegere).

DNS (Domain Name System) este un sistem distribuit, cu o structura ierarhica, de pastrare si interogare a unor date arbitrare. DNS-ul este folosit pentru a traduce nume de domenii in adrese IP si invers.

Majoritatea serviciilor din internet se bazeaza pe DNS pentru buna lor functionare, astfel daca DNS nu functioneaza sau este prea lent atunci paginile web nu vor putea fi accesate si nici e-mail-urile nu vor putea fi transmise.

Denumirea domeniilor faciliteaza localizarea unei resurse din retea de catre utilizatori fiind mult mai usor sa retina o adresa de tip `www.mywebserver.com` decat adresa numerica de 32 de biti, spre exemplu `192.168.100.101`. Translatia nume/ip este transparenta pentru utilizator.

DNS-ul este practic compus din 3 componente: **Resolver DNS**, **Servere DNS**, **date DNS** sub forma de inregistrari de resurse (**Resource Records - RR**). Internetul, ca intreg, contine milioane de RR-uri care sunt impartite pe zone. Fiecare zona are un numar de servere DNS autorizate care raspund de zona respectiva.