4. Retele de calculatoare

In aceasta sectiune vom trece in revista cateva concepte de baza ale retelelor de calculatoare, punand accentul pe latura *informativa* , fara a intra in prea multe detalii ... Mai intai se va defini notiunea de retea si se va da o clasificare a acestora. In continuare vom vorbi, pe scurt, despre arhitectura standard a retelelor, protocoale de comunicatie, despre reteaua Internet si despre modalitatile practice de a realiza transferul de date. Aceasta sectiune contine urmatoarele capitole:

1. Definitia si clasificarea retelelor
2. Arhitectura unei retele
3. Reteaua Internet
4. Schimbul de date in retea

4.1 Definitia si clasificarea retelelor

4.1.1 Sa definim notiunea de retea !

INTERCONECTARE =

doua calculatoare se considera interconectate daca pot schimba date intre ele

MEDIU DE COMUNICATIE =

mediu fizic prin intermediul caruia se pot transmite date (cablu, fibra optica, radio, satelit)

RETEA DE CALCULATOARE =

ansamblu de calculatoare interconectate prin intermediul unor medii de comunicatie, asigurand folosirea in comun, de catre un mare numar de utilizatori, a tuturor resurselor fizice, logice si informationale ale ansamblului.

Simplificand putin definitia, putem privi reteaua ca fiind un grup de noduri interconectate, un nod putand sa contina:

- *calculator gazda* sau **host**
- *terminal video*
- *controler de comunicatie*
- *echipament periferic*

Folosirea unei retele determina urmatoarele avantaje:

- **Impartirea resurselor** - toate programele, datele si echipamentele sunt disponibile pentru orice utilizator al retelei, indiferent de localizarea fizica a resursei sau a utilizatorului;
- **Fiabilitate sporita**- prin accesul la mai multe echipamente de stocare alternative (fisierele pot fi stocate de doua-trei echipamente, asigurand accesul la date chiar daca unul dintre echipamente se defecteaza);
- **Extensibilitate** -reteaua se poate extinde usor prin conectarea altor echipamente, iar realizarea unui up-grade intr-o zona a retelei nu influenteaza negativ schimbul de date in celelalte zone;
- **Economie financiara**- o retea de calculatoare este mult mai fiabila si mai ieftina decat un supercalculator;
- **Mediu puternic de comunicatie** :
  - Posta electronica (e-mail)
  - Videoconferinte

- Divertisment interactiv

4.1.2 Clasificarea retelelor de calculatoare

Clasificarea retelelor trebuie sa ia in considerare doua aspecte foarte importante: tehnologia de transmisie si scara la care opereaza reteaua.
Din punct de vedere al tehnologiei de transmisie, retelele sunt de doua feluri:

1. **Retele cu difuzare**
   - Un singur canal de comunicatie este partajat de toate masinile din retea
   - Comunicatia se realizeaza prin intermediul unor mesaje scurte, numite pachete, care au in structura lor, printre altele, un camp pentru desemnarea expeditorului si unul pentru desemnarea destinatarului
   - Se pot trimite pachete catre toate masinile din retea, acest mod de operare numindu-se *difuzare*

2. **Retele punct-la-punct**
   - Dispun de numeroase conexiuni intre perechile de masini individuale ce formeaza reteaua
   - Pentru a ajunge la destinatie, un pachet de date trebuie sa treaca prin mai multe masini intermediare, fiind nevoie de algoritmi pentru dirijarea pachetelor pe un drum optim
   - Este un model folosit pentru retelele mari, in timp ce difuzarea se foloseste pentru retelele mici

Dupa marimea retelei, distingem trei tipuri:
- **Retele locale (LAN)**- retele localizate intr-o singura cladire sau intr-un campus de cel mult cativa kilometri; conectarea se face de obicei cu ajutorul unui singur cablu, la care sunt legate toate masinile
- **Retele metropolitane (MAN)**-retele care se pot intinde intr-o zona de pe suprafata unui intreg oras. Pentru conectare se folosesc doua cabluri unidirectionale la care sunt conectate toate calculatoarele, fiecare cablu avand un *capat de distributie* (dispozitiv care initiaza activitatea de transmisie)
- **Retele larg raspandite geografic (WAN)**- retele care ocupa arii geografice intinse, ajungand la dimensiunea unei tari sau a unui intreg continent;
-

4.2 Arhitectura unei retele

4.2.1 Topologia retelelor

Prin **topologia** unei retele se intelege modul de interconectare a calculatoarelor in retea. Folosirea unei anumite topologii are influenta asupra vitezei de transmitere a datelor, a costului de interconectare si a fiabilitatii retelei. Exista cateva topologii care s-au impus si anume: *magistrala, inel, arbore*. Pe langa acestea intalnim si alte modele topologice: *stea, inele intersectate, topologie completa si topologie neregulata* . In figura 1 puteti vedea reprezentarea , sub forma de grafuri, a acestor modele.

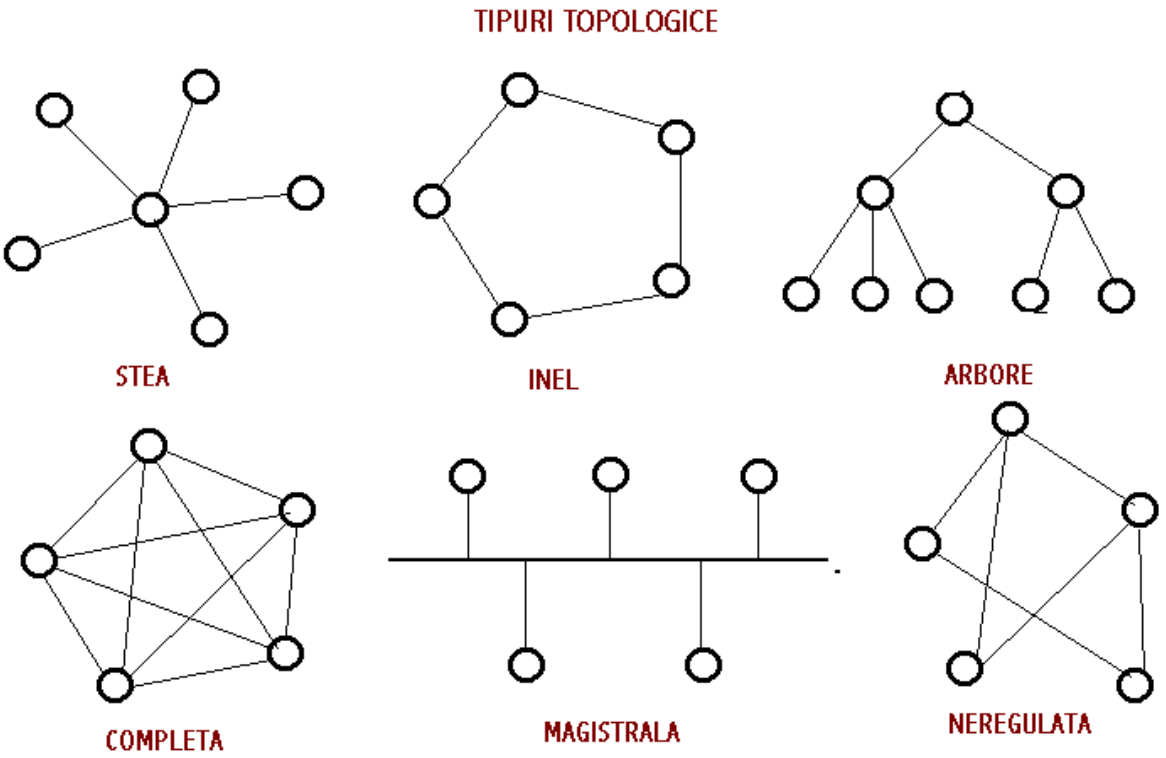


Fig.1: Topologii folosite frecvent

Topologia de magistrala este cea mai folosita atunci cand se realizeaza retele locale de mici dimensiuni, iar performantele nu trebuie sa fie spectaculoase. Acest model topologic se mai numeste si *magistrala liniara*, deoarece exista un singur cablu care leaga toate calculatoarele din retea. Avantajul este atat acela al costului mai scazut (se foloseste mai putin cablu), dar si acela ca, in cazul ruperii unui cablu sau defectarii unui calculator, nu se ajunge la oprirea intregii retele. Dezavantajul folosirii unui singur cablu este ca, atunci cand doreste sa transmita date, calculatorul trebuie sa "lupte" pentru a castiga accesul (trebuie sa astepte eliberarea cablului).

Topologia de inel conecteaza fiecare calculator de alte doua, imaginea fiind aceea a unor calculatoare asezate in cerc. Datele transmise de un calculator trec prin toate calculatoarele intermediare inainte de a ajunge la destinatie. Daca nu se folosesc cabluri suplimentare, oprirea unui calculator sau ruperea unui cablu duce la oprirea intregii retele. Performantele unei retele inel sunt ceva mai mari decat ale unei retele magistrala.

Topologia stea foloseste un calculator central care va fi conectat cu toate celelalte calculatoare prin cabluri directe. Toate transferurile de date se realizeaza prin intermediul calculatorului central. Daca se foloseste un calculator central de mare putere, atunci reteaua va avea performante ridicate, insa defectarea acestuia duce la oprirea retelei.

Se pot folosi topologii combinate, cum ar fi *lantul de stele* insa, orice topologie ar fi aleasa, exista un numar de probleme ce trebuiesc rezolvate (modul de obtinere a accesului este una dintre cele mai importante, trebuind eliminata posibilitatea ca un singur calculator sa "monopolizeze" mediul de transmisie). Apar probleme suplimentare atunci cand reteaua

noastra este *eterogena* (conecteaza diverse tipuri de calculatoare sau este formata din mai multe retele diferite ca tip).
Trebuie sa facem distinctie intre **topologia fizica**, despre care am discutat mai sus, si **topologia logica** (modul in care datele sunt transferate de la un calculator la altul).

4.2.2 Arhitectura retelelor

Un concept foarte important in retelele de calculatoare este acela de **protocol** .
PROTOCOL
=ansamblu de conventii si reguli pe baza carora se realizeaza transmiterea datelor
ARHITECTURA
=modul de interconectare a componentelor retelei, pentru a realiza un anumit mod de functionare

Arhitectura unui sistem trebuie sa ne dea informatii despre modul in care se conecteaza componentele sistemului si despre interactiunea dintre acestea, dar ofera si o imagine generala a sistemului. Stabilirea arhitecturii sistemului, fie ca este vorba despre o retea sau despre un produs software, este una dintre cele mai importante etape ale realizarii unui proiect. Este vital sa se stabileasca zonele critice ale sistemului, adica acele componente ce prezinta risc mare de defectare sau care, prin defectarea lor, pot provoca oprirea partiala sau totala a sistemului. Trebuiesc luati in considerare si factorii care ar putea avea influenta asupra sistemului (pana si conditiile atmosferice ar putea influenta functionarea unei retele).

Pentru reducerea complexitatii alcatuirii, majoritatea retelelor sunt organizate pe mai multe nivele (straturi), in sensul impartirii stricte a sarcinilor: fiecare nivel este proiectat sa ofere anumite servicii, bazandu-se pe serviciile oferite de nivelele inferioare. Atunci cand doua calculatoare comunica, in fapt, se realizeaza o comunicare intre nivelele de acelasi rang ale celor doua masini. Nivelul n al masinii A realizeaza schimb de date cu nivelul n al masinii B prin intermediul unui protocol numit *protocolul nivelului n* . In realitate datele nu sunt transmise de la nivelul n al unei masini catre nivelul n al alteia. In schimb, fiecare nivel realizeaza prelucrarile specifice asupra datelor si le transmit nivelului inferior, pana la nivelul fizic unde se realizeaza schimbul efectiv de date. Doar din punct de vedere logic se poate vorbi de o "conversatie" intre nivelele a doua masini.
Intre oricare doua nivele adiacente exista o *interfata* , care stabileste care sunt serviciile oferite nivelului superior. In momentul proiectarii arhitecturii retelei trebuie sa se specifice clar numarul de nivele si interfetele aferente. Multimea protocoalelor si a nivelelor reprezinta *arhitectura* retelei. Specificatiile arhitecturii (i.e. documentatia ce descrie arhitectura) trebuie sa fie destul de detaliate pentru a permite implementarea de aplicatii care sa se conformeze specificului fiecarui nivel.

4.2.3 Modelul arhitectural ISO-OSI

Vom lamuri pentru inceput intelesul unui cuvant, *standard*, care, alaturi de termeni precum *contor, tutorial, implementare, specificatie...* produc frisoane unei "anumite parti" a tineretului studios...

Odata cu aparitia unei noi tehnologii, se manifesta si un fenomen de proliferare a produselor ce utilizeaza tehnologia respectiva, fiecare producator dorind sa impuna pe piata propria realizare (mai buna sau mai proasta decat altele). Dupa un anumit timp, piata realizeaza o "selectie naturala", ramanand in competitie doar produsele de calitate (mai sunt si cateva exceptii, cum ar fi acea firma a carui nume incepe cu M, se termina cu T si are un

produs W...). Acest interval de timp duce la "maturizarea" tehnologiei respective si reprezinta un test al utilitatii ei. Urmeaza interminabile discutii si controverse intre firmele combatante, iar o comisie internationala incearca sa stabileasca un set de reguli si conventii obligatorii pentru toti cei ce dezvolta produse bazate pe tehnologia in discutie. Astfel se naste un *standard* . "Fizic", standardul se prezinta sub forma unui "metru cub" de documentatie, prea putin accesibila omului de rand, continand recomandari pe care nu toti le respecta sau ar fi imposibil ca, respectandu-le, sa "iasa" un produs functional (sfarsitul glumei). Standardul este important pentru unificarea diverselor variante ale tehnologiei respective si defineste un set de reguli generale, universal acceptate, contribuind la aparitia de produse portabile (na, a mai aparut un termen !). Standardele sunt aprobate de organizatii internationale, cum ar fi: **OSI** (International Standards Organisation), **ECMA** (European Computer Manufacturer's Association), **IEEE** (Institute of Electrical and Electronical Engineers), **ANSI**.

Elaborarea standardelor pentru retele a devenit necesara datorita diversificarii echipamentelor si serviciilor, care a condus la aparitia de retele eterogene din punctul de vedere al tipurilor de echipamente folosite. In plus, multitudinea de medii fizice de comunicatie a contribuit la decizia de a defini reguli precise pentru interconectarea sistemelor. ISO a elaborat un model arhitectural de referinta pentru interconectarea calculatoarelor, cunoscut sub denumirea de *modelul arhitectural ISO-OSI (Open System Interconnection)*.

Modelul ISO-OSI imparte arhitectura retelei in sapte nivele, construite unul deasupra altuia, adaugand functionalitate serviciilor oferite de nivelul inferior. Modelul **nu** precizeaza **cum** se construiesc nivelele, dar insista asupra serviciilor oferite de fiecare si specifica modul de comunicare intre nivele prin intermediul interfetelor. Fiecare producator poate construi nivelele asa cum doreste, insa fiecare nivel trebuie sa furnizeze un anumit set de servicii. Proiectarea arhitecturii pe nivele determina extinderea sau imbunatatirea facila a sistemului. De exemplu, schimbarea mediului de comunicatie nu determina decat modificarea nivelului fizic, lasand intacte celelalte nivele. In figura 2 puteti vedea cele sapte nivele ale modelului arhitectural OSI.

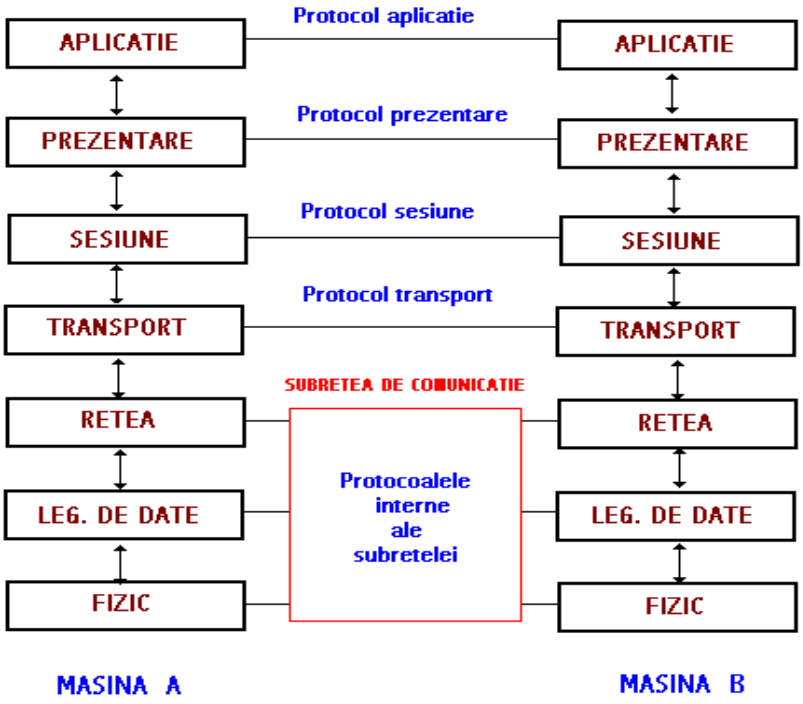


Fig.2: Modelul arhitectural ISO-OSI

In cele ce urmeaza voi prezenta cate ceva despre fiecare nivel:

1. **Nivelul fizic** are rolul de a transmite datele de la un calculator la altul prin intermediul unui mediu de comunicatie. Datele sunt vazute la acest nivel ca un sir de biti. Problemele tipice sunt de natura electrica: nivelele de tensiune corespunzatoare unui bit 1 sau 0, durata impulsurilor de tensiune, cum se initiaza si cum se opreste transmiterea semnalelor electrice, asigurarea pastrarii formei semnalului propagat. Mediul de comunicatie nu face parte din nivelul fizic.
2. **Nivelul legaturii de date** corecteaza erorile de transmitere aparute la nivelul fizic, realizand o comunicare corecta intre doua noduri adiacente ale retelei. Mecanismul utilizat in acest scop este impartirea bitilor in cadre (*frame*), carora le sunt adaugate informatii de control. Cadrele sunt transmise individual, putand fi verificate si confirmate de catre receptor. Alte functii ale nivelului se refera la fluxul de date (astfel incat transmitatorul sa nu furnizeze date mai rapid decat le poate accepta receptorul) si la gestiunea legaturii (stabilirea conexiunii, controlul schimbului de date si desfiintarea conexiunii).
3. **Nivelul retea** asigura dirijarea unitatilor de date intre nodurile sursa si destinatie, trecand eventual prin noduri intermediare (*routing*). Este foarte important ca fluxul de date sa fie astfel dirijat incat sa se evite aglomerarea anumitor zone ale retelei (*congestionare*). Interconectarea retelelor cu arhitecturi diferite este o functie a nivelului retea.
4. **Nivelul transport** realizeaza o conexiune intre doua calculatoare gazda (*host*) detectand si corectand erorile pe care nivelul retea nu le trateaza. Este nivelul aflat in mijlocul ierarhiei, asigurand nivelelor superioare o interfata independanta de tipul retelei utilizate. Functiile principale sunt: stabilirea unei conexiuni sigure intre doua masini gazda, initierea transferului, controlul fluxului de date si inchiderea conexiunii.
5. **Nivelul sesiune** stabileste si intretine conexiuni (*sesiuni*) intre procesele aplicatie, rolul sau fiind acela de a permite proceselor sa stabileasca "de comun acord" caracteristicile dialogului si sa sincronizeze acest dialog.
6. **Nivelul prezentare** realizeaza operatii de transformare a datelor in formate intelese de entitatile ce intervin intr-o conexiune. Transferul de date intre masini de tipuri diferite (Unix-DOS, de exemplu) necesita si codificarea datelor in functie de caracteristicile acestora. Nivelul prezentare ar trebui sa ofere si servicii de criptare/decriptare a datelor, in vederea asigurarii securitatii comunicatiei in retea.
7. **Nivelul aplicatie** are rolul de "fereastra" de comunicatie intre utilizatori, acestia fiind reprezentati de entitatile aplicatie (programele). Nivelul aplicatie **nu** comunica cu aplicatiile ci controleaza mediul in care se executa aplicatiile, punandu-le la dispozitie servicii de comunicatie. Printre functiile nivelului aplicatie se afla:
    - identificarea partenerilor de comunicatie, determinarea disponibilitatii acestora si autentificarea lor
    - sincronizarea aplicatiilor cooperante si selectarea modului de dialog
    - stabilirea responsabilitatilor pentru tratarea erorilor
    - identificarea constrangerilor asupra reprezentarii datelor
    - transferul informatiei

Primele trei nivele de la baza ierarhiei (fizic, legatura de date, retea) sunt considerate ca formand o *subretea de comunicatie* . Subreteaua este raspunzatoare pentru realizarea transferului efectiv al datelor, pentru verificarea corectitudinii transmisiei si

pentru dirijarea fluxului de date prin diversele noduri ale retelei. Acest termen trebuie inteles ca desemnand "subreteaua logica", adica multimea protocoalelor de la fiecare nivel care realizeaza functiile de mai sus. Termenul de subretea este utilizat si pentru a desemna liniile de transmisie si echipamentele fizice care realizeaza dirijarea si controlul transmisiei. Modelul OSI nu este implementat in intregime de producatori, nivelele sesiune si prezentare putand sa lipseasca (unele din functiile atribuite acestora in modelul OSI sunt indeplinite de alte nivele). Modelul OSI este un model orientativ, strict teoretic, realizarile practice fiind mai mult sau mai putin diferite. Ei, va zice unul dintre voi, pai ce-am batut campii atata cu un model teoretic ? Mie nu-mi foloseste asta la nimic ! Lucrurile nu stau chiar asa. Intelegerea unui alt model este mult usurata de studierea modelului ISO-OSI, motiv pentru care orice carte serioasa il prezinta detaliat.

Sa vedem cum se realizeaza un transfer de date intre doua masini gazda. Cel mai bun exemplu este modul in care putem citi o pagina web aflata pe un calculator situat la mare distanta:

- utilizatorul lanseaza un program pentru vizualizarea paginilor web (*browser*)
- browserul este entitatea aplicatie care va "negocia" pentru noi obtinerea paginii
- nivelul aplicatie va identifica existenta resursei cerute de client (clientul este browserul, care-l reprezinta pe utilizator in aceasta "tranzactie") si a posesorului acesteia (serverul-inteles ca fiind entitatea ce ofera resursa ceruta nu calculatorul central al unei retele; in cazul nostru avem de-a face cu un server de web). Se realizeaza autentificarea serverului (se verifica daca partenerul este intr-adevar cine pretinde ca este (cam ciudata chestie pentru o retea, nu ?)) si se stabileste daca acesta este disponibil (=poate si vrea sa ne satisfaca cererea).
- Nivelul sesiune va stabili o conexiune intre procesul client si procesul server
- Nivelul transport se va ocupa de intretinerea conexiunii si de corectarea erorilor netratate la nivelul retea
- nivelul retea va asigura transferul datelor in secvente (pachete), stabilind drumul acestora intre server si client

Lucrurile sunt ceva mai complicate decat in cele prezentate mai sus. Datele sosesc prin intermediul mediului de comunicatie ca un flux de biti. La nivelul legaturii de date, bitii sunt transformati in cadre, iar la nivelul retea in pachete (vom vedea mai tarziu cum arata un pachet). In cele din urma, datele ajung la nivelul aplicatie unde sunt preluate de browser si ne sunt prezentate. Fiecare nivel adauga sau sterge o parte din informatiile de control atasate datelor de celelalte nivele.

4.2.4 Modelul arhitectural TCP/IP

Modelul TCP/IP a fost utilizat de reteaua ARPANET si de succesorul acesteia, INTERNET, numele provenind de la protocoalele care stau la baza modelului:

- TCP (Transmission Control Protocol)
- IP (Internet Protocol)

Obiectivul central avut in vedere la proiectarea retelei a fost acela de a se putea interconecta fara probleme mai multe tipuri de retele, iar transmisia datelor sa nu fie afectata de distrugerea sau defectarea unei parti a retelei. In plus, arhitectura retelei trebuia sa permita rularea unor aplicatii cu cerinte divergente, de la transferul fisierelor si pana la transmiterea datelor in timp real (videoconferinte).
Modelul TCP/IP are doar patru nivele:

1. **Nivelul gazda-retea**
   Modelul nu spune mare lucru despre acest nivel, esentialul fiind acela ca, printr-un

anumit protocol (nu se zice nimic despre el), gazda trimite prin intermediul retelei pachete IP. Acest protocol misterios difera de la o retea la alta si subiectul nu este tratat in literatura de specialitate.

2. **Nivelul internet**
   Acest nivel este axul pe care se centreaza intreaga arhitectura, rolul sau fiind acela de a permite gazdelor sa emita pachete in retea si de a asigura transferul lor intre sursa si destinatie. Se defineste un format de pachet si un protocol (IP), nivelul trebuind sa furnizeze pachete IP la destinatie, sa rezolve problema dirijarii pachetelor si sa evite congestiile (lucreaza asemanator cu nivelui retea din modelul OSI).
3. **Nivelul transport**
   Este proiectat astfel incat sa permita dialogul intre entitatile pereche din gazdele sursa si destinatie, pentru aceasta fiind definite doua protocoale *capat-la-capat*: TCP si UDP. Protocolul de control al transmisiei (TCP) permite ca un flux de octeti emis de o masina sa fie receptionat fara erori pe orice alta masina din retea. TCP fragmenteaza fluxul de octeti in mesaje discrete pe care le paseaza nivelului internet. La destinatie, procesul TCP receptor reasambleaza mesajele primite, reconstituind datele initiale. TCP realizeaza controlul fluxului de date pentru a evita situatia in care un transmitator rapid inunda un receptor lent cu mai multe mesaje decat poate acesta sa prelucreze. TCP este un protocol *orientat pe conexiune*
   UDP (*User Datagram Protocol- protocolul datagramelor utilizator*) este un protocol nesigur, fara conexiuni, destinat aplicatiilor care doresc sa utilizeze propria secventiere si control al fluxului si nu mecanismele asigurate de TCP. Este un protocol folosit in aplicatii pentru care comunicarea rapida este mai importanta decat acuratetea transmisiei, asa cum sunt aplicatiile de transmitere a sunetului si imaginilor video.
4. **Nivelul aplicatie**
   Nivelul aplicatie contine protocoalele de nivel inalt, cum ar fi **terminalul virtual** (TELNET), **transferul de fisiere** (FTP) si **posta electronica** . Protocolul TELNET permite utilizatorului sa se conecteze pe o masina aflata la distanta si sa lucreze ca si cum s-ar afla intr-adevar langa aceasta. Pe parcurs s-au adaugat alte protocoale ca **DNS** (serviciul numelor de domenii), pentru stabilirea corespondentei dintre numele gazdelor si adresele retelelor, **NNTP**- folosit pentru transferul articolelor (stiri), **HTTP**-folosit pentru transferul paginilor web, e.t.c

Exercitiu: Care este corespondenta dintre nivelele modelului OSI si nivelel modelului TCP/IP ?

4.3 Reteaua Internet

4.3.1 Scurt istoric al retelei Internet

Istoria Internetului, desi sub acest nume va aparea mult mai tarziu, incepe in 1966 odata cu crearea Agentiei pentru Proiecte de Cercetare Avansata (ARPA). Obiectivul agentiei era crearea unei retele de comanda a trupelor SUA, care sa poata ramane operationala chiar si in cazul unui atac nuclear (de unde se vede ca multe lucruri bune apar din ratiuni militare). Reteaua a fost denumita ARPAnet si a fost operationala in 1969, cand lega 4 calculatoare din laboratoarele unor universitati. Agentia ARPA a finantat proiecte de cercetare in domeniul retelelor ale universitatilor americane, pe baza carora s-a dezvoltat reteaua. Realizarea "practica" a retelei a fost incredintata firmei BBN (constructia subretelei

de comunicatie). S-a hotarat ca subreteaua sa aiba ca router-e minicalculatoare IMP Honeywell DDP-316 special modificate (memorie de 12 KB, cuvant de 16 biti, fara discuri mobile considerate nesigure), conectate prin linii telefonice de 56 Kbps inchiriate de la diverse firme de telefoane. Fiecare nod al retelei era format dintr-un calculator gazda si un IMP (Interface Message Processors) aflate in aceeasi incapere. Pentru a spori siguranta comunicarii, fiecare IMP era conectat cu alte doua. Initial, ARPAnet a functionat experimental prin conectarea a patru universitati, iar in cativa ani s-a extins pe intreg teritoriul SUA. In 1983, ARPAnet a fost divizata in doua sisteme distincte: MILNET (retea destinata operatiunilor militare) si ARPAnet.

Reteaua ARPAnet a aratat cercetatorilor cat este de utila comunicarea rapida intre echipele de cercetatori aflate in diverse orase ale SUA si s-a dorit conectarea cat mai multor universitati. Procedura de conectare era restrictiva deoarece reteaua ARPAnet era finantata de ARPA (adica, pe scurt, de Pentagon). La sfarsitul anilor '70, din initiativa Fundatiei Nationale de Stiinta (NSF), a demarat proiectul de conectare a universitatilor care nu aveau contract de colaborare cu ARPA. Initial au fost oferite servicii de posta electronica. In 1986 a fost creata o subretea care conecta sase centre de supercalculatoare din sase orase americane, fiecare supercalculator conectat fiind legat de un minicalculator LSI-11 (FUZZBALL), numit si "fratele mai mic". Fuzzball-urile au format subreteaua la care au fost conectate, in timp, 20 de retele regionale. Reteaua a fost denumita NSFNET.
La mijlocul anilor '80, retelele ARPAnet si NSFNET au "fuzionat" si, odata cu marirea exponentiala a numarului cererilor de conectare, lumea a inceput sa perceapa colectia de retele ca fiind o uriasa conexiune de retele. Putem spune ca este momentul nasterii retelei Internet. Aceasta cuprindea, in 1990, 3000 de retele si 200 000 gazde pentru a ajunge astazi la cateva zeci de mii de LAN-uri si milioane de gazde. Expansiunea spectaculoasa a retelei a inceput in 1992, dupa ridicarea interdictiei de a desfasura activitati comerciale pe Internet si odata cu aparitia WWW. Una dintre activitatile profitabile este aceea de furnizor Internet (Internet provider= firma care ofera servicii de conectare la Internet.

4.3.2 Structura retelei Internet

In aceasta sectiune vom incerca sa aruncam o privire asupra structurii retelei Internet si sa lamurim anumiti termeni care apar foarte des, toata lumea ii foloseste dar mult prea putini stiu cu exactitate ce inseamna (desi nimeni n-ar recunoaste...). Fiindca Internetul este un conglomerat de retele, va fi util sa ne familiarizam si cu tipurile principale de retele locale conectate la Internet.

Dupa cum aminteam mai demult, o retea WAN este compusa dintr-o multime de noduri conectate prin intermediul unui mediu de comunicatie (cabluri coaxiale, cabluri torsadate, fibra optica, radio, satelit). Intr-un nod nu se afla neaparat un calculator gazda (host-sistem final), ci se poate amplasa un echipament periferic (display, imprimanta...) sau un *controler de comunicatie* (numit si *nod de comutare* sau ***ROUTER***). Gazdele sunt conectate printr-o *subretea de comunicatie* , a carei sarcina este sa transporte datele intre host-uri. In majoritatea retelelor WAN, subreteaua de comunicatie este formata din doua componente distincte:

- **liniile de transmisie** (*circuite / canale*)- transporta bitii de date intre masini;
- **elemente de comutare** - echipamente folosite pentru a conecta doua sau mai multe linii de transmisie;

Daca va mai amintiti, primele trei nivele ale modelului arhitectural OSI formau subreteaua de comunicatie, iar in modelul TCP/IP acest rol era indeplinit de nivelul gazda-retea si

nivelul internet. Datele ce trebuiesc transmise sunt divizate in "bucati" mai mici numite *pachete*. Cand un pachet este transmis de la un router la altul, prin intermediul unuia sau mai multor routere intermediare, pachetul este primit de fiecare router intermediar, este retinut acolo pina la eliberarea liniei cerute si apoi este retransmis mai departe. O subretea care functioneaza pa acest principiu se numeste *subretea cu comutare de pachete* sau *subretea punct-la-punct* sau *subretea memoreaza si transmite* . In figura de mai jos puteti vedea un fragment dintr-o retea WAN, fiind pusa in evidenta o parte a subretelei de comunicatie.

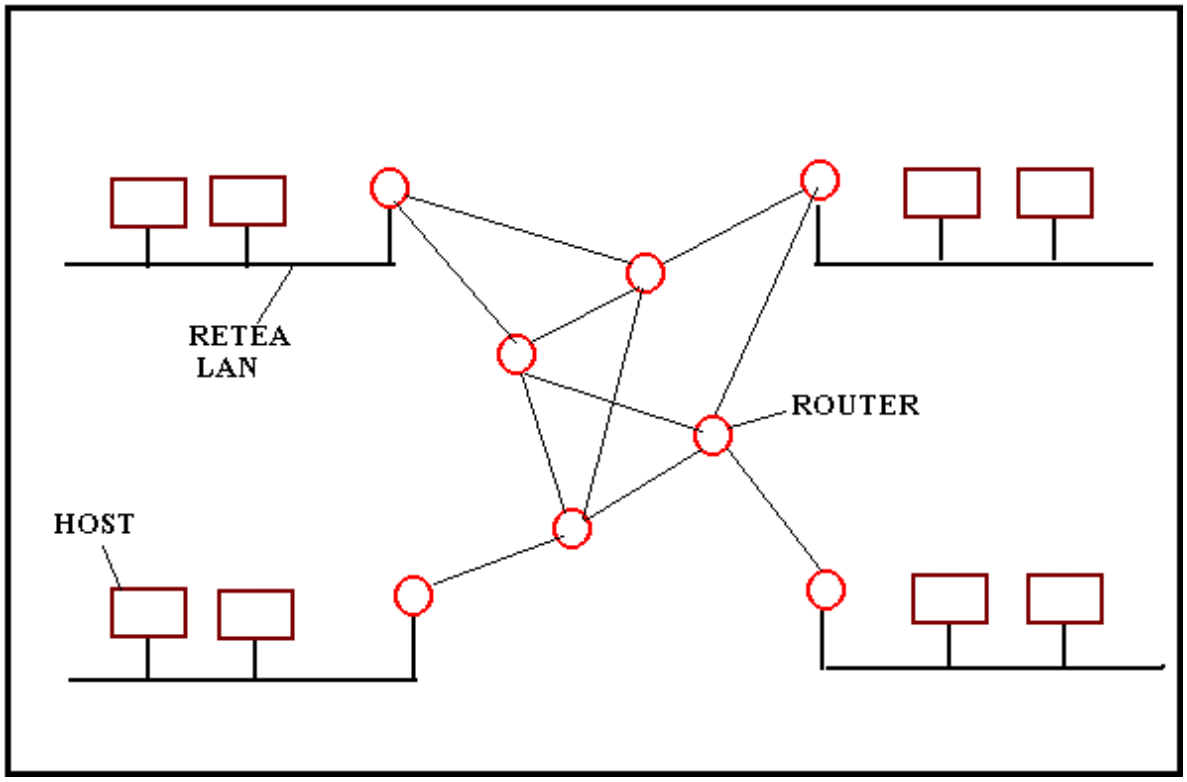


Fig. 3: Structura subretelei de comunicatie

Topologia de interconectare a routerelor este una dintre cele amintite in sectiunile precedente (stea, arborem inel, neregulata...). Topologia retelelor WAN este de obicei neregulata, in timp ce topologia unei retele LAN este una simetrica.

4.3.3 Retele locale si interconectarea mai multor retele LAN

Pentru conectarea fizica intre calculatoare s-a folosit la inceput topologia de magistrala, la care pe segmentul de mediu fizic (cablu coaxial gros, de culoare galbena- comparat de multe ori cu un furtun pentru udat gradina), se puteau conecta pana la 29 de echipamente pe o lungime de maxim 500 m. Conectarea echipamentelor se facea prin intermediul unui dispozitiv numit *transceiver* . Nu se mai realizeaza astazi asemenea retele, topologia de magistrala avand un cablu coaxial subtire, iar conexiunile fiind facute prin mufe T. Este vorba despre LAN-uri Ethernet.

Numele Ethernet este legat de ipoteza vehiculata mult timp de oamenii de stiinta, potrivit careia intre corpurile ceresti se afla zone "umplute" de un "gaz" misterios numit eter. Creatorul Ethernetului si-a imaginat o retea in care nu este important unde se afla

calculatoarele ci faptul ca acestea pot comunica fara restrictii si a ales, metaforic vorbind, eterul ca mediu de comunicatie.

Reteaua Ethernet a fost dezvoltata de Robert Metcalfe in 1973, pe vremea cand era angajat al companiei Xerox si a evoluat continuu, devenind cel mai popular tip de retea locala. Primele retele Ethernet au fost cele cu cablu coaxial gros si legatura prin transceivere, numite *Thick Ethernet* . Astazi se foloseste cablul coaxial subtire cu conexiuni T (Thin Ethernet), cablu torsadat sau fibra optica (Twisted Pair Ethernet, Fast Ethernet). Conexiunea cu mufe T are avantajul eliminarii transceiverului, mufa conectandu-se direct la placa de retea,"coada" T-ului, cablul continuind prin extremitatile T-ului catre calculatorul urmator.

Cea mai populara versiune Ethernet avea o viteza de transmisie a datelor de 10 Mb /s (Mb=mega bit pe secunda), iar Fast Ethernet de 100Mb/s. Se lucreaza astazi pentru realizarea unor retele cu viteza de transfer mai mare de 1000 Mb/s, numele vehiculat pentru acestea fiind Gigabit Ethernet.

Celelalte topologii, inel (*ring*) si stea (*star* sau *hub*) sunt mai putin raspandite la noi. Topologia de inel a fost utilizata de IBM pentru tipul de retea *Token Ring* , folosita astazi doar pentru conectari rapide la mare distanta cu fibra optica. Topologia stea s-a folosit in retelele *Arcnet* unde conectarile erau facute la un hub, in stea, fiind posibile si conexiuni intre hub-uri. Hub-ul este un dispozitiv in care intra un singur cablu si care are mai multe iesiri. (De exemplu, intr-un hub intra cablul ce pleaca de la server iar cablurile care ies conecteaza calculatoare sau alte hub-uri).

Retelele locale, indiferent de tipul lor, stabilesc o limita maxima a numarului echipamentelor ce se pot conecta la retea. Atunci cand este nevoie de conectarea mai multor calculatoare decat permite tipul de retea utilizat sau atunci cand avem mai multe retele locale, eventual de tipuri diferite, pe care dorim sa le conectam exista doua solutii: repetoarele si podurile. Un *repetor* (**REPEATER**) este un amplificator electric, care preia semnalul dintr-o parte si il trece in cealalta parte marind puterea acestuia (semnalul se pierde daca nu este amplificat). Repetorul nu poate fi folosit decat pentru legarea a doua retele de acelasi tip, iar folosirea unui numar mare de repetoare duce la scaderea vitezei de transmisie a datelor.

Atunci cand dorim sa conectam doua retele, mai ales daca sunt de tipuri diferite, vom utiliza un *pod* (**BRIDGE**). Acesta este conectat la doua sau mai multe retele locale, simultan. Podul stie sa preia pachetele de date dintr-o retea si sa le transmita in cealalta, realizand si anumite conversii ale pachetelor (pachetul de date ce "trece pe pod" va trebui "inteles" de calculatoarele din reteaua destinatie, deci vor trebui facute conversiile de structura; fiecare tip de retea foloseste pachete diferite ca structura, deci, de exemplu, un pachet Ethernet nu va fi "inteles" de o retea Token Ring. Se spune ca un pod "vorbeste" un protocol diferit la fiecare capat (regulile de transmisie a datelor sunt diferite de la o retea la alta). Podul nu copiaza un pachet dintr-o parte in alta decat daca destinatia sa se afla in alta retea decat calculatorul care l-a expediat, prevenind aglomerarea inutila a retelei. Exista si poduri care au un algoritm care le permite sa invete cand sa preia un pachet si cand nu (algoritm de invatare). In finalul discutiei despre poduri, voi preciza ca podul nu este altceva decat un calculator specializat si nu cine stie ce echipament misterios.

4.3.4 Nivelul retea si nivelul transport in Internet

Un lucru deosebit de important care trebuie precizat este acela ca Internetul nu conecteaza calculatoare ci retele. Daca mai punem la socoteal si faptul ca retelele au tipuri diferite, ne dam seama cat de importanta este stabilirea unor protocoale de comunicatie care

sa permita conectarea fara probleme a diverselor tipuri de retele si echipamente. La nivelul retea, Internetul poate fi vazut ca o colectie de subretele sau sisteme autonome interconectate. Nu exista o structura reala precisa dar se pot pune in evidenta cateva segmente principale ("coloane vertebrale")(vezi figura de mai jos).

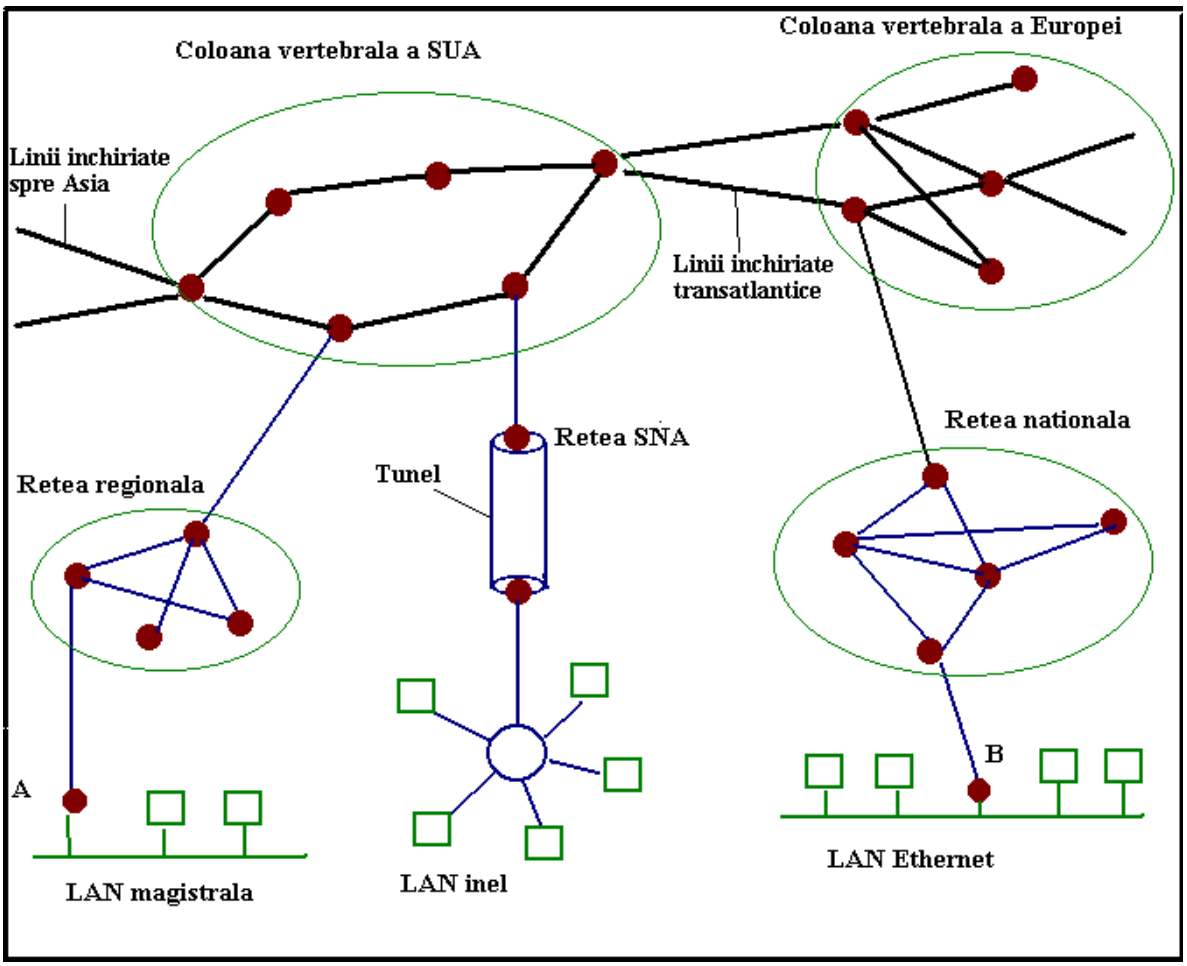


Fig. 4: Structura generala a retelei Internet

Coloanele vertebrale sunt construite din linii de mare capacitate si routere rapide, la care se conecteaza retelele regionale (de nivel mediu). Retelele regionale conecteaza LAN-uri din institutii, firme si ale furnizorilor de servicii Internet.

Liantul care tine Internetul la un loc este protocolul de nivel retea IP, special conceput pentru interconectarea retelelor. Sarcina sa este aceea de a oferi o cale de a transporta pachetele de date de la sursa la destinatie, fara a tine seama daca masinile expeditor si receptor (**SENDER si RECEIVER**) sunt de acelasi tip sau daca se afla in aceeasi retea ori mai sunt retele intre ele.

Comunicatia in Internet functioneaza astfel:

- nivelul transport preia fluxul de date si il sparge in pachete cu lungimea teoretica de maxim 64kb (practic 1500 bytes)
- fiecare pachet este transmis separat, putand fi fragmentat din nou pe drum
- pachetele ajunse la destinatie sunt reasamblate pentru a obtine datele originale
- datele reasamblate sunt pasate nivelului transport, care le insereaza in sirul de intrare al procesului receptor

Un pachet IP este compus dintr-un antet cu informatii de control si o parte de date, unde sunt stocate datele ce sunt transmise.
Antetul pachetului contine, printre altele, informatii despre:

- versiunea protocolului care a creat pachetul
- lungimea antetului
- lungimea zonei de date
- tipul derviciului dorit (fiabilitate si viteza)
- timpul de viata al pachetului: este un contor care numara cate salturi a facut pachetul (salt=trecerea dintr-un nod in altul), valoarea maxima fiind 255. Cand un pachet are valoarea contorului 0 (adica a petrecut cam mult timp pe drum), acesta este distrus iar hostul sursa este avertizat de pierderea datelor. Protocolul care se ocupa de acest lucru este **ICMP (Internet Control Message Protocol)**, un alt protocol al familiei TCP/IP.
- ce protocol de transport trebuie sa preia pachetul
- adresa sursei
- adresa destinatiei
- cat de secreta este informatia

Sa consideram cazul in care un host din reteaua A doreste sa transmita date unui host din reteaua B, dupa cum se vede in figura 4. Datele sunt transmise routerului A, care le trimite mai departe (salt) prin reteaua regionala la care este conectata reteaua locala, trece printr-o multime de routere intermediare din coloana vertebrala a SUA si Europei, ajunge in reteaua nationala si apoi in reteaua B, de unde vor fi dirijate catre hostul destinatie. Lucrurile par simple la prima vedere dar problemele care apar in transmiterea datelor de la sursa la destinatie sunt deosebit de complexe. Drumul ales de pachet de la sursa la destinatie se numeste *ruta* (**route**), iar operatia de alegere a unui drum dintre drumurile posibile la un moment dat se numeste *rutare* (**routing**). Nivelul retea trebuie sa aleaga drumul cel mai potrivit pentru fiecare pachet, astfel incat ruta sa fie cat mai ieftina si sa nu se produca aglomerari ale unor sectoare din retea (congestii). Sunt folositi diversi algoritmi pentru alegerea rutei optime, cum ar fi algoritmul lui Dijkstra (vedeti la ce sunt bune cunostintele despre grafuri ?) sau algoritmi ce se bazeaza pe anumite tabele actualizate dinamic (tabele de rutare).

Nu am luat inca in discutie un lucru foarte important: pentru a putea realiza transmisia datelor este nevoie sa identificam in mod unic fiecare gazda conectata in retea. Aceasta adresa va fi utilizata pentru localizarea gazdelor in Internet. O adresa IP este formata din patru numere intregi intre 0..255, despartite prin punct. De exemplu, adresa noastra este **192.168.1.1** . O adresa IP este formata dintr-un identificator al retelei (net ID) si un identificator (numar) de masina (host ID). Toate calculatoarele dintr-o anumita retea vor avea acelasi net ID
Adresele IP se pot clasifica in mai multe clase:

- Clasa A: net ID de 8 biti, host ID de 24 biti
- Clasa B: net ID de 16 biti, host ID de 16 biti
- Clasa C: net ID de 24 biti, host ID de 8 biti

Formatul A permite adresarea a 126 retele cu 16 milioane de gazde fiecare (valoarea primului octet nu poate depasi 127). Formatul B permite adresarea a 16383 de retele cu pana la 65000 gazde fiecare si 16000 subretele (primul octet ia valori intre 128 si 191). Formatul C adreseaza aproximativ 2 milioane de retele LAN cu pana la 255 gazde fiecare. Primul octet ia valori intre 192 si 222. Valorile de la 223 pana la 255 sunt rezervate pentru utilizari ulterioare.

4.3.5 Spatiul de nume in Internet (DNS)

Programele utilizate in mod curent se refera rareori la sistemele gazda, cutii postale si alte resurse prin adresa lor binara (IP), in locul acesteia fiind folosite siruri de caractere de forma:

nume_masina_gazda.subdomeniu1.subdomeniu2...subdomeniu_n.domeniu

Folosirea unor siruri de caractere in locul adreselor binare duce la utilizarea usoara a adreselor, fiind mult mai usor de retinut decat niste numere care nu spun mare lucru utilizatorilor obisnuiti. Va fi necesar un mecanism care sa permita convertirea unei adrese din format ASCII in format IP, singurul format recunoscut in retea. De exemplu, reteaua ARPANET avea un site ce continea un fisier text numit *host.txt* , care cuprindea toate sistemele gazda si adresele lor. Conversia intre adrese era realizata pe baza acestui fisier, insa aceasta modalitate este rezonabila doar intr-o retea ce contine cateva sute de masini gazda. In cazul Internetului s-a adoptat o alta solutie, numita DNS (Domain Name System-Sistemul Numelor de Domenii).

Internetul este divizat in cateva sute de zone de nivel superior, numite *domenii*, fiecare domeniu cuprinzand subdomenii sau/si sisteme gazda, rezultand o reprezentare arborescenta a DNS. Domeniile de pe nivelul unu al arborelui sunt de doua categorii:

1. **generice:**
   - com -comercial
   - edu -institutii de educatie
   - gov -guvernul SUA
   - mil -armata SUA
   - int -organizatii internationale
   - org -organizatii nonprofit
2. **de tari** : fiecare tara are alocat un domeniu (**ro** -Romania)

Fiecare adresa este data de drumul parcurs in arbore de la masina respectiva si pana in radacina arborelui, componentele fiind separate prin punct. Componentele numelor pot avea maxim 64 de caractere, iar intregul nume nu poate depasi 255 de caractere. Fiecare domeniu controleaza cum sunt alocate adresele in subdomeniile sale. Pentru a creea un subdomeniu (sa zicem ca am dori sa facem inca o retea, separata de cea pe care o avem) se cere permisiunea domeniului in care va fi inclus subdomeniul, astfel fiind evitate conflictele de nume. Fiecare domeniu primeste un anumit numar de adrese care pot fi alocate subdomeniilor sale.

DNS consta intr-o schema ierarhica (arborescenta) de nume de domenii si dintr-un sistem de baze de date distribuite pentru implementarea schemei de nume. Spatiul de nume DNS este impartit in mai multe zone disjuncte, fiecare zona continand o parte a arborelul de adrese precum si numele serverelor care pastreaza informatiile referitoare la acea zona. O zona poate avea un server de nume (server DNS) primar, care preia informatiile dintr-un fisier de pe discul propriu, si mai multe servere de nume secundare, care iau informatia de pe discul serverului primar. Pentru mai multa siguranta, unele servere DNS sunt plasate in afara zonei pe care o administreaza.

Structura arborescenta a DNS permite utilizarea de domenii cu acelasi nume. Pentru a se stabili corespondenta intre nume si adresa IP se procedeaza astfel:

- programul de aplicatie apeleaza o precedura de biblioteca (*resolver*), transferandu-i ca parametru numele de domeniu

- resolver-ul trimite un pachet UDP la serverul local DNS, care cauta numele si returneaza adresa IP asociata acestuia
- avand adresa IP, programul apelant poate stabili o conexiune TCP cu destinatia...

4.4 Transferul datelor in Internet

4.4.1.1 Nivelul aplicatie-generalitati

Pana acum am discutat, mai mult sau mai putin amanuntit, despre subreteaua de comunicatie si despre nivelul transport, amintind cate ceva despre protocoalele care lucreaza la aceste nivele si care asigura stabilirea si intretinerea conexiunilor. Aceste nivele nu indeplinesc nici o functie concreta pentru utilizatori. *Nivelul aplicatie*, dupa cum spuneam in capitolele precedente, contine protocoale ce permit functionarea aplicatiilor reale, cele care interactioneaza cu utilizatorul.

Unul dintre aspectele de care trebuie sa se tina cont este *securitatea retelei* . Sunt necesare mecanisme care sa depisteze pe aceia care apeleaza la servicii pe care nu sunt autorizati sa le foloseasca si trebuie sa inlature posibilitatea ca, anumite persoane, curioase ori rauvoitoare, sa modifice sau sa poata citi mesajele adresate altor utilizatori. In momentul actual, securitatea datelor in Internet este doar un vis frumos. In ultimii ani s-au descoperit noi metode pentru criptarea datelor dar altii lucreaza pentru a "sparge" noile coduri, fie pentru a demonstra ca metodele descoperite nu sunt chiar eficiente, fie din motive mai putin morale (=furt). Furturile prin intermediul Internetului (caci nu se pot numi altfel) sunt principalul obstacol in calea dezvoltarii comertului electronic. Este foarte important sa denumim "actiunile" de obtinere a unor "foloase necuvenite" (cumpararea de bunuri prin utilizarea de carti de credit false ori sustrase, sustragerea datelor cu regim secret, efectuarea de convorbiri telefonice gratuite...)pur si simplu **furt**, desi unora li se pare a fi o distractie nevinovata. Legile sunt foarte aspre cu asemenea infractiuni (asta in tarile occidentale, fireste...) si, in curand, se vor introduce si la noi astfel de reglementari dure.
Deci, **atentie !!!**
In incheierea acestui paragraf cu tenta educativa, trebuie sa mai spun ca este bine sa nu folositi in mesajele electonice cuvinte si expresii care ar putea jigni pe cineva sau ar putea fi considerate calomnii sau probe intr-un proces...

Vom trece acum in revista cele mai importante protocoale ce activeaza la nivelul aplicatie, punand la dispozitia utilizatorilor un set complet de servicii de comunicare si transfer de date.

4.4.1.2 Conectarea pe o masina aflata la distanta

Conectarea la o masina aflata la distanta se realizeaza pe baza conceptului de **terminal virtual**. Senzatia utilizatorului este aceea ca se afla chiar langa host-ul apelat. Pentru a realiza o conexiune cu o masina ce lucreaza sub sistemul de operare Unix se foloseste comanda **rlogin** (*remote login*). Toate resursele (discuri, fisiere,...) de care dispune o masina poarta numele de **resurse locale** (*local resource*). Daca o anumita resursa poate fi accesata numai prin intermediul retelei, atunci avem de-a face cu o resursa la distanta sau **remote resource** . Folosirea comenzii rlogin determina utilizarea terminalului local ca si cum acesta ar fi conectat direct la masina aflata la distanta. Spunem ca se realizeaza o *sesiune de terminal virtual* . Sintaxa comenzii este:

```
rlogin nume_masina optiuni
```

, unde este numele unei statii disponibil in fisierul /etc/hosts sau adresa Internet a unei statii. Se poate folosi optiunea **-l** pentru a specifica si un nume de utilizator.
Spre exemplu, daca aveti un cont pe hostul *homer* , cont numit mybox, va puteti conecta la acesta de la distanta, prin comanda:

```
rlogin homer.hasdeu.bz.edu.ro  -l  mybox
```

Terminarea conexiunii cu masina de la distanta se face prin comanda **exit** .
Daca dorim sa realizam o conexiune de terminal virtual cu o masina care nu lucreaza in sistemul Unix, vom folosi comanda **telnet** , avand ca parametru numele masinii sau adresa acesteia. Comanda telnet este utilizata de voi la fiecare conectare la contul Linux personal (se realizeaza conectarea prin terminal virtual dintre un PC si o masina Linux).

4.4.1.3 Posta electronica (E-mail)

Trimiterea de mesaje (scrisori) a fost unul dintre primele servicii puse la dispozitia utilizatorilor unei retele si a avut un succes imens. Posta electronica a facut posibila comunicarea rapida intre angajatii firmelor (mai ales cand exista echipe mari de cercetare, care lucreaza in diverse tari), realizindu-se mari economii de timp in dezvoltarea proiectelor. Astazi, posta electronica a devenit un mijloc foarte rapid si comod de comunicare, fiind chiar mai ieftin decat sistemul postal clasic. Au fost creeate o multime de "liste de discutii" unde participantii pot discuta problemele care ii intereseaza. Cele mai raspandite sunt listele de discutii ce folosesc posta electronica. Fiecare mesaj trimis la adresa unei astfel de liste va fi retrimis tuturor utilizatorilor inscrisi. De multe ori exista si un moderator, care dirijeaza discutiile si verifica respectarea regulilor stabilite pentru acea lista (exista reguli de "comportament" care nu trebuie incalcate, una dintre cele mai importante fiind adresarea civilizata si fara insulte sau trivialitati; fireste, pentru a fi admis in anumite "cluburi de discutii", purtarea grosolana si comportamentul de "ghetou" este chiar o conditie de baza...). In general, utilizatorul trebuie sa respecte cateva reguli de comportare, care formeaza asa numita **NETicheta** .

Pentru a putea trimite un mesaj, se utilizeaza un program specializat (in Linux exista programul PINE). Fiecare retea trebuie sa aiba un server de posta electronica. Regulile de transmitere a mesajelor intre serverele de posta au fost definite in protocolul **SMTP** (*Simple Mail Transfer Protocol*). Nu intru in amanunte privitoare la acest protocol si nici nu voi descrie programele folosite pentru scrierea mesajelor, acestea putand fi teme de studiu pentru voi sau pentru alti "fericiti" ce va vor urma...

4.4.2 Transferul fisierelor

Unul dintre marile avantaje ale retelelor este acela de a pastra informatii in diverse hosturi si de a putea transfera o parte dintre acestea atunci cand este nevoie. Pentru a realiza transferul de date intre doua masini Unix, se foloseste comanda **rcp** (*remote copy*), care este o extensie a comenzii Unix **cp**.

Cea mai folosita modalitate de a transfera de date este aceea de a utiliza comanda **FTP**, care implementeaza protocolul cu acelasi nume (**File Transfer Protocol**).
Protocolul FTP este implementat pentru toate sistemele de operare (Unix, Dos, Windows).
Trebuie sa facem precizarea ca Internetul nu se reduce la World Wide Web (despre care

vom vorbi ceva mai tarziu), care este doar o mica parte a acestuia. Inainte de aparitia navigatoarelor grafice si a sistemelor hipertext, informatia era pastrata in arhive ce puteau fi transferate prin intermediul comenzii ftp (arhive ftp), arhivele fiind accesibile si astazi.

Accesul la informatie se face prin intermediul unui server FTP, care poate fi public sau privat. Serverul FTP public permite accesul oricarui utilizator, caruia i se cere ca nume "cuvantul " *anonymous* iar ca parola adresa E-mail. Toate arhivele disponibile sunt depuse intr-un director numit **pub** . Serverele private restrictioneaza accesul, fiind proiectate pentru utilizarea de catre un anumit grup de utilizatori (angajatii unei firme, de exemplu0.
Iata ce trebuie sa faceti pentru a transfera un fisier aflat intr-o arhiva FTP:

- Se tasteaza comanda ftp, iar pe ecran veti obtine promptul **ftp>** , unde sunteti invitat sa introduceti comenzi specifice protocolului ftp.
- Comanda **open** *adresa* va realiza conectarea cu masina specificata prin adresa. Urmeaza identificarea utilizatorului prin nume si parola.
- Odata ce primiti accesul, puteti accesa structura de subdirectoare din "pub" prin comenzile obisnuite (cd) pana ajungeti la fisierul dorit (toate serverele ftp contin si un index al fisierelor pe care le puteti gasi acolo).
- Stabiliti modul de transfer al informatiei: ASCII sau BINAR (prin comenzile **ascii** si **bin**). De multe ori este preferabila folosirea transferului binar.
- Comanda **get** *fisier* transfera fisierul specificat pe masina locala (se foloseste **mget** daca transferam o lista de fisiere).
  Daca dorim sa transferam un fisier de pe masina locala pe masina de la distanta, vom utiliza comenzile **put** fisier si **mput** *fisier*.
- Pentru inchiderea conexiunii se tasteaza comanda **close**, iar pentru terminarea sesiunii ftp se foloseste **quit**.

Arhivele FTP sunt disponibile si astazi, ele continand milioane de fisiere de date. Accesul la acestea este ceva mai dificil pentru utilizatorii obisnuiti sau lenesi ==> este posibil accesul la zonele ftp din Internet si prin utilizarea unor aplicatii cu interfata grafica, ce usureaza transferul de fisiere si automatizeaza acest proces.

4.4.3.1 World Wide Web

Incepand cu momentul standardizarii protocoalelor din familia TCP/IP s-au putut dezvolta retele de mare intindere, ce conectau mii de calculatoare bazate pe platforme hardware diferite si lucrand sub diverse sisteme de operare. Notiunea de comunicare era pusa intr-o noua lumina.

Odata cu dezvoltarea exploziva a retelei Internet, cantitatea de informatie pusa la dispozitia utilizatorilor a crescut exponential. Accesul la informatie nu era insa chiar usor pentru ca se lucra mai mult in linia de comanda (am vorbit mai devreme despre utilizarea protocolului FTP).

In 1989, adica intr-o perioada in care se mai foloseau produse ca Wordstar, a aparut un concept care avea sa dea un nou impuls dezvoltarii Internetului: World Wide Web. Personajul principal al "povestirii" este Tim-Berners Lee, informatician la CERN (Institutul European de Fizica a Particulelor), care a cautat un mod mai simplu de structurare a informatiei necesare proiectelor de cercetare pe care le dezvolta laboratorul. El a dorit sa creeze un mediu de comunicare rapid si usor de utilizat, care sa permita accesul la informatie din orice colt al lumii, prin intermediul retelei Internet.

Conceptul care a stat la baza WWW este conceptul de**hypertext** . Prin hypertext se intelege o colectie de documente legate intre ele prin legaturi (intre documente diferite sau

intre paragrafe ale aceluiasi document), permitand parcurgerea (*navigarea*) documentelor de-a lungul acestor legaturi (*link*), bidirectional.]

Ideea parcurgerii documentelor pe trasee neliniare a aparut inca din 1945, cand un inginer numit V. Bush a propus un sistem ce permitea urmarirea informatiilor in baze de date. Cel care a definit conceptul de hypertext, asa cum este inteles astazi, este Ted Nelson autorul sistemului *Xanadu* , care era bazat pe noduri ce contin informatie si legaturi catre alte noduri.

Posibilitatea de a realiza legaturi catre fisiere cu imagini si sunete au determinat modificarea conceptului de hypertext, astazi vorbindu-se despre *hypermedia* . Tim-Berners Lee a avut ideea crearii unor sisteme hypertext independente de sistemul de operare folosit. Link-urile puteau lega nu numai documente locale ci si documente aflate pe masini de la distanta, realizandu-se o "tesatura" la nivel mondial (*Web*). Traversarea legaturilor dintre documente (*navigare*= **browse**) se realiza prin intermediul unui program special denumit **browser** (navigator Web).
La inceputul anilor '90 a aparut primul navigator grafic, **Mosaic**, dezvoltat de Marc Andreessen (student la Universitatea din Illinois) si Eric Bina (programator la NCSA (National Center of Supercomputing Applications), care functiona sub XWindows. In 1993, Marc Andreessen si James Clark (unul dintre fondatorii firmai Silicon Graphics) au intemeiat compania Mosaic Communications Corp. (devenita ulterior Netscape Communications).

4.4.3.2 Protocolul HTTP

Numele este acronimul pentru HyperText Transfer Protocol, protocol ce stabileste regulile de transfer a documentelor hypermedia. Aplicatiile care folosesc protocolul- cei doi parteneri de la capetele unei conexiuni- sunt considerate niste entitati abstracte, din punctul de vedere al protocolului. Entitatile trebuie sa poata formula cereri si/sau receptiona raspunsuri (modelul client-server). Protocolul defineste reguli de comunicare, care permit interpretarea corecta a cererilor si raspunsurilor.

Unul dintre conceptele de baza- preluat si de alte protocoale- este cel de **resursa**, desemnand un calculator, o baza de date, un document, in general un serviciu. Resursa trebuie sa poata fi referita corect si fara echivoc. Pentru referirea unei resurse in Internet, se foloseste termenul generic **URI -Uniform Resource Identifier** , care poate specifica o locatie, caz in care se vorbeste de un **URL -Universal Resource Locator** sau de un nume, caz in care avem de-a face cu un **URN- Universal Resource Name** . Protocolul se bazeaza pe paradigma cerere/raspuns. Clientul cere accesul la o resursa, aceasta fiind identificata prin URI, iar serverul raspunde printr-o linie de stare (contine, printre altele, un cod de succes sau eroare si, in primul caz, urmeaza datele cerute). Cel mai simplu caz este acela cand conexiunea client-server se realizeaza prin intermediul unei singure conexiuni. In general, exista mai multi intermediari de-a lungul conexiunii. Acestia pot fi de trei feluri:

- **proxy**-este un intermediar sofisticat: primeste cereri adresate unei resurse identificate prin URI, rescrie anumite parti ale mesajului, dupa care retrimite cererea catre serverul adresat initial. El se substituie, practic, clientului initial, mesajul de raspuns va fi primit tot de el, iar proxy-ul trimite clientului raspunsul. Dinspre server nu se mai "vad" clientii adevarati, ei fiind reprezentati de serverul proxy. Serverul proxy poate realiza anumite verificari (de autentificare, de securitate,...), dificil de implementat pe toate masinile conectate la acel proxy. Serverul proxy trebuie inteles ca un reprezentant al unui grup intreg de clienti, negociind cererile acestora adresate "restului lumii".

- **gateway** - este similar unui proxy, dar pe partea de server. Este un fel de camera de primire pusa in fata unui server sau a unui grup de servere. Serverele de "dupa gateway" nu sunt vizibile, ele fiind reprezentate de gateway. Cererile sosite la gateway sunt dirijate spre serverul care poate raspunde cererii, sau celui mai liber dintre serverele ce pot raspunde, in dorinta de a utiliza eficient puterea de calcul. *Poarta* realizeaza si o conversie de protocol, serverul nefiind obligat sa "cunoasca" protocolul "http"
- **tunnel** - tunelul este un intermediar neinteligent: el transporta date pe care nu le intelege sau interpreteaza in nici un fel intre doua conexiuni. De obicei, la un capat al tunelului se afla un server gateway, iar la capatul celalalt un proxy.

Un mare avantaj al folosirii unui server proxy este acela ca, beneficiind de un cache intern, acesta poate memora cererile facute de clienti. Ulterior, la aparitia unei cereri care s-a mai facut, serverul proxy trimite raspunsul direct, daca acesta a ramas memorat in cache (nu se mai realizeaza conexiunea cu "detinatorul" resursei). Un proxy poate lucra cu mai multi clienti in acelasi timp, putand filtra cererile primite.
Adresarea unei resurse se face prin constructii de forma:

http://adresa_host [:port] /cale/subcale1/.../subcale_n/nume_document

unde:

- *http*- numele protocolului utilizat
- *adresa_host* identifica un server sau un gateway din retea, utilizand adresarea uzuala din DNS (vezi lectia precedenta)
- *:port* specifica portul de date la care se face conexiunea, implicit :80
- */cale/subcale1/.../subcale_n/* - calea absoluta pana la documentul *nume_document* de pe respectivul server

Observatie: resursa referita nu este neaparat un intreg document, ci poate fi doar o fractiune a documentului.
Serverul care raspunde cererilor privitoare la documente hypermedia se numeste server WWW si "cunoaste" protocolul http.

Lucrarea nr.4. Sectiunea 2.

4.5 Proiectarea retelelor Ethernet

Retea – Prin retea se întelege o colectie de calculatoare independente ce comunicã unul cu altul printr-un mediu de comunicatie comun.

LAN (Local Area Network) – sunt acele retele ce sunt limitate la o anumitã zonã geograficã, cum ar fi o clãdire sau un grup de clãdiri. Proiectarea unei astfel de retele locale nu este deloc simplã, ea putând contine chiar câteva sute de calculatoare.

WAN (Wide Area Network) – reprezintã o retea constituitã din mai multe retele locale separate din punct de vedere geografic. Legãtura între aceste retele locale se face prin diferite moduri: linii telefonice, legãturã prin satelit, etc.

Intranet – este o retea privatã bazatã pe protocolul Internet, dar disponibilã doar în interiorul respectivei organizatii.

Protocol – un protocol de retea este un standard pe baza cãruia comunicã douã sau mai multe calculatoare. Un protocol defineste modul de identificare al calculatoarelor din retea, forma în care sunt transmise datele, si modul de prelucrare a acestora odatã ce au ajuns la destinatie. În plus mai sunt tratate si cazurile de eroare sau pierdere a legãturii. Cele mai cunoscute protocoale sunt: IPX (pentru Novell NetWare), TCP/IP (pentru UNIX, Windows NT, Windows 95 si altele), DECnet (pentru calculatoare DEC), AppleTalk (pentru Macintosh) si NetBIOS/NetBEUI (pentru retele Windows NT).

4.5.1 Reteaua Ethernet

Este cel mai cunoscut nivel fizic de comunicatie pentru o retea localã, prin intermediu cãruia se transmite informatia între calculatoare la la viteza de 10 milioane de biti pe secundã (Mbps). Standardul Ethernet este definit de IEEE (Institute for Electrical and Electronic Engineers) ca IEEE 802.3. Acest standard defineste regulile pentru configurarea unei retele Ethernet precum si modul de interactiune între diferitele elemente ale unei astfel de retele.

Fiecare calculator echipat cu o placã de retea Ethernet, denumit si statie, functioneazã independent de toate celelalte statii din retea: nu existã control centralizat. Toate statiile atasate la retea sunt conectate la acelasi sistem de transport pentru semnal, denumit mediu de comunicatie. Informatia este transmisã serial, un bit la un moment dat, prin linia de comunicatie cãtre toate statiile atasate acesteia.

4.5.1.1 Elementele unei retele Ethernet

O retea Ethernet are trei elemente de bazã:

1. mediul fizic de comunicatie – folosit pentru transmiterea semnalului purtãtor de informatie între calculatoarele retelei;
2. protocolul de comunicatie – un set de reguli pentru controlul accesului la mediul de comunicatie respectat de fiecare interfatã, pe baza cãruia se arbitreazã accesul mai multor calculatoare la acest mediu;
3. cadrarea informatiei – un cadru Ethernet ce constã într-un set standardizat de biti folosit la transportul datelor prin retea.

L4 Retele de calculatoare

Mediul de comunicatie – Cabluri Ethernet

Termenul de 10BaseT este abrevierea de la 10 Mbps transmission, Baseband medium, Twisted pair (transmisie la 10 Mbps, în banda de bazã prin fire torsadate). Pentru o retea Ethernet sunt folosite urmãtoarele tipuri de cabluri:

- 10 Base 5 (sau cablu coaxial gros) – foloseste un singur cablu coaxial de 75 ohmi într-o topologie bus, connectând fiecare placã de retea printr-un dispozitiv ce se lega direct pe cablu. Lungimea maximã a unui segment este de 500 metri, terminatoare de 50 ohmi;

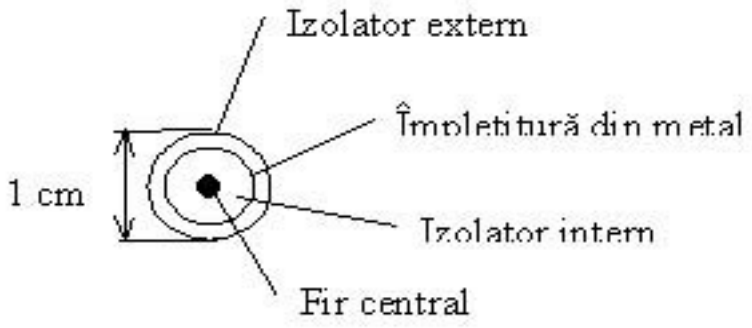


Fig. 5

- 10 Base 2 (sau cablu coaxial subtire) – foloseste un sir de cabluri coaxiale RJ-58 într-o topologie bus, cu conectori BNC T atasati fiecãrei plãci de retea, si având câte un terminator de 52 ohmi la fiecare capãt. Lungimea maximã pentru un segment este de 200 meteri, diametrul cablului este 0.5 cm, impedanta cablului 50 ohmi;
- 10 Base T – foloseste douã sârme torsadate (preferabil ecranate) într-o topologie stea, fiecare segment conectând un singur dispozitiv la un repetor, cunoscut sub numele de hub;
- 10 Base F – foloseste ca mediu de comunicatie cablul optic;
- 10 Broad 36 – singurul tip de mediu fizic cu transmisie în bandã largã, permite conectarea statiilor prin cablu TV cu circuit închis.

Distantele maxime pentru segmentele de retea Ethernet si numãrul de statii legate la o retea Ethernet depind de tipul cablului de transmisie folosit:

	Viteza	Tipul de cablu	Numãr maxim de statii pe segment	Distanta maximã dintre statii (m)
10 Base 5	10 Mbps	Coaxial gros	100	500
10 Base 2	10 Mbps	Coaxial subtire	30	185 (200)
10 Base T	10 Mbps	UTP 3	2	100
10 Base F	10 Mbps	Fibrã opticã	2	2000
10 Broad 36	10 Mbps	TV		

Tabelul 1

Cadrele Ethernet

Sunt folosite pentru transferul informatiei între statii. Un cadru constã din un numãr de biti organizati în câteva câmpuri. Acestea includ câmpurile cu adresele statiilor, câmpul pentru date având dimensiunea între 46 si 1500 biti, un câmp pentru controlul erorilor, etc. Adresele pe 48 de biti sunt unice pentru fiecare placã de retea, sunt atribuite de producãtor si nu pot fi modificate.

Protocolul CSMA/CD

Reteaua Ethernet foloseste protocolul numit CSMA/CD (Carrier Sense Multiple Access with Collision Detect). Termenul de acces multiplu ("Multiple Access") provine de la faptul cã fiecare statie este conectatã la acelasi mediu de comunicatie. "Carrier Sense" – înainte de a transmite date, o statie verificã linia pentru a vedea dacã nu existã nici o altã statie care transmite ceva. Dacã se constatã cã linia nu este ocupatã, statia poate începe sã transmitã date cu o vitezã de 10 Mbps, adicã câte un bit la 100 ns. Viteza luminii si a semnalului electric fiind de 300000 m/s, înseamnã cã electronii vor parcurge 0.3 m într-o ns. Astfel cã, dupã ce semnalul electric pentru primul bit a parcurs aproximativ 30 m, statia începe transmisia celui de-al doilea bit. Însã, un cablu Ethernet poate avea mai mult de 30 m. Dacã douã statii se aflã la o distantã de 100 m legate la acelasi cablu de retea, si ambele încep transmisia în acelasi timp (gãsind ambele linia liberã), atunci fiecare va fi în mijlocul transmiterii celui de-al treilea bit când semnalul de la fiecare statie ajunge la cealaltã. Acest exemplu explicã necesitatea pãrtii de "Collision Detect". Douã statii pot începe transmiterea datelor în acelasi timp, însã semnalele fiecãruia vor coliziona doar dupã câteva nanosecunde. Când apar astfel de coliziuni cele douã statii încetează transmisia si încearcã mai târziu dupã o perioadã de timp aleasã aleator.

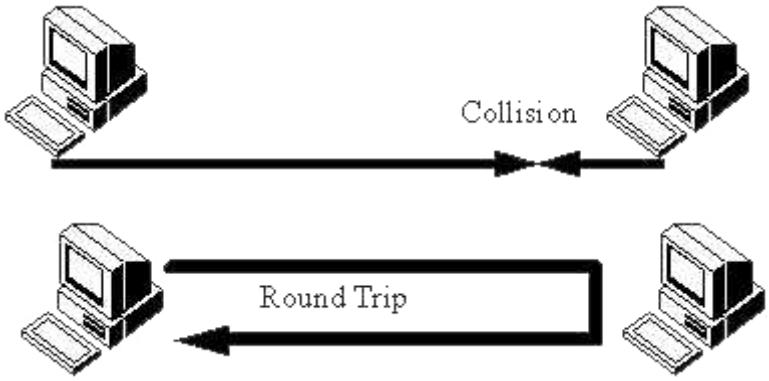


Fig. 6

Pentru ca sistemul de control al accesului la mediul de comunicatie sã functioneze corect (CSMA/CD), toate interfetele Ethernet trebuie sã rãspundã semnalului receptionat într-o perioadã datã de timp. Aceastã necesitate timp este bazatã pe perioada de timp necesarã unui semnal sã ajungã de la un capãt al mediului de comunicatie la celãlalt si înapoi, acest timp este cunoscut sub numele de timp de întoarcere ("round trip time"). Timpul de întoarcere maxim este strict limitat pentru a asigura ca fiecare interfatã sã poatã receptiona toate semnalele din linia de comunicatie într-o perioadã de timp specificatã. Cu cât este mai lung un segment de retea, cu atât mai mult timp îi ia unui semnal sã îl parcurgã.

La proiectarea unei retele Ethernet trebuie deci, sã se asigure ca timpul de întoarcere sã fie în limitele specificate, indiferent de combinatia de cabluri si echipamente folosite în retea.

4.5.1.2 Repetor (repeater) si comutator (switch)

Reteaua Ethernet a fost astfel proiectatã încât sã permitã o expandare usoarã, pe mãsura cerintelor de vitezã si de spatiu tot mai mari. Pentru extinderea unei retele Ethernet se pot folosi mai multe tipuri de dispozitive denumite hub-uri. Existã douã mari categorii de hub-uri: repetor (repeater) si comutator (switch).
Fiecare port al unui repetor leagã împreunã segmentele de cablu Ethernet individuale pentru a crea o nouã retea ce functioneazã ca un Ethernet independent si singular. Segmentele si repetoarele din aceastã nouã retea trebuie sã respecte limitãrile timpului de întoarcere.

Spre deosebire de un repetor, fiecare port al unui comutator conectează câte un segment de cablu care functioneazã ca o retea Ethernet distinctã. Deci, spre deosebire de un repetor ale cãrui porturi combinã segmentele de cablu pentru a forma un singur LAN, un comutator face posibilã divizarea unei retele Ethernet de dimensiuni mari, în mai multe retele Ethernet independente ce sunt legate printr-un mecanism de comutare a pachetelor (cadrelor Ethernet). Regulile pentru timpul de întoarcere nu se mai aplicã retelei globale ci doar retelelor Ethernet obtinute prin divizare.

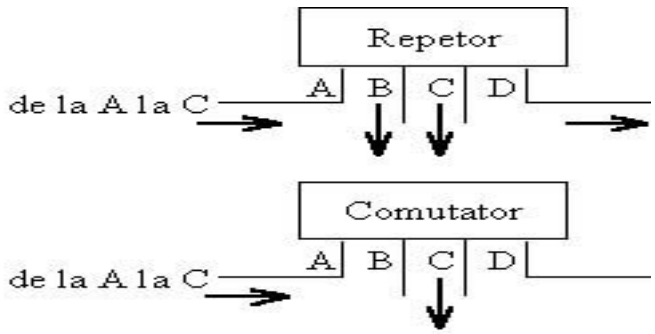


Fig. 6

Prin folosirea comutatoarelor se si pot lega mai multe retele Ethernet distincte.
O retea Ethernet poate fi constituitã:

- doar dintr-un singur cablu (coaxial) legând un numãr de calculatoare;
- dintr-un repetor conectând:
  - fiecare calculator printr-un segment de cablu (torsadat);
  - câte un segment continând mai multe calculatoare;

Mai multe astfel de retele Ethernet pot forma o retea extinsã prin utilizarea unui comutator de pachete. În timp ce o retea Ethernet simplã poate suporta un numãr de câteva zeci de statii, o retea extinsã poate lega câteva sute sau mii de statii.
Comutatoarele examineazã fiecare pachet receptionat pe fiecare port, îl procesează si îl transmite (dacã este cazul), pe baza unei baze de date initiale sau create dinamic, cãtre portul ce corespunde statiei destinatie. Pe când repetorul retransmite fiecare cadru primit pe toate porturile, fãrã nici un fel de prelucrare a pachetului.
În comutator se pãstrazã o bazã de date cu adresele Ethernet ale statiilor si portul din comutator corespunzãtor fiecãrei statii. Când comutatorul receptionează un cadru Ethernet,

foloseste adresele sursã si destinatie pentru a determina dacã cele douã statii se gãsesc pe acelasi segment de retea, caz în care pachetul este filtrat (este sters); iar dacã statiile se gãsesc pe segmente diferite, pachetul este trimis doar pe segmentul statiei destinatie.
Cu o astfele de functionare, un comutator împarte reteaua în domenii de coliziune distincte – câte unul pentru fiecare segment; spre deosebire de un repetor care formeazã un singur domeniu de coliziuni.

4.5.1.3 Topologii

Mediu de comunicatie al unei retele Ethernet este folosit în douã tipuri de configuratii generale (topologii): bus si stea. Aceste douã tipuri de topologii definesc modul în care sunt conectate statiile. (Fig. 7, 8)

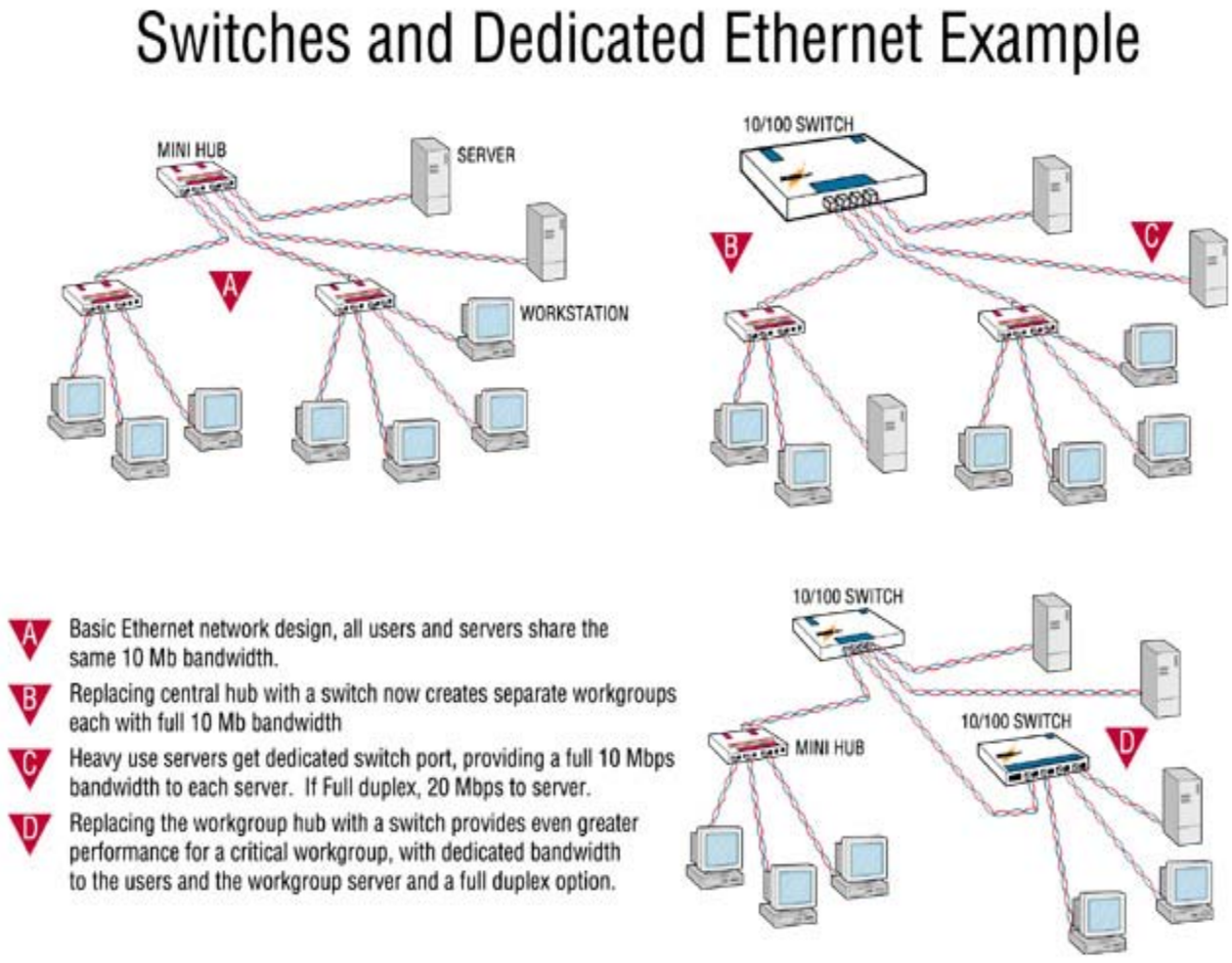


Fig. 7: Topologii Ethernet

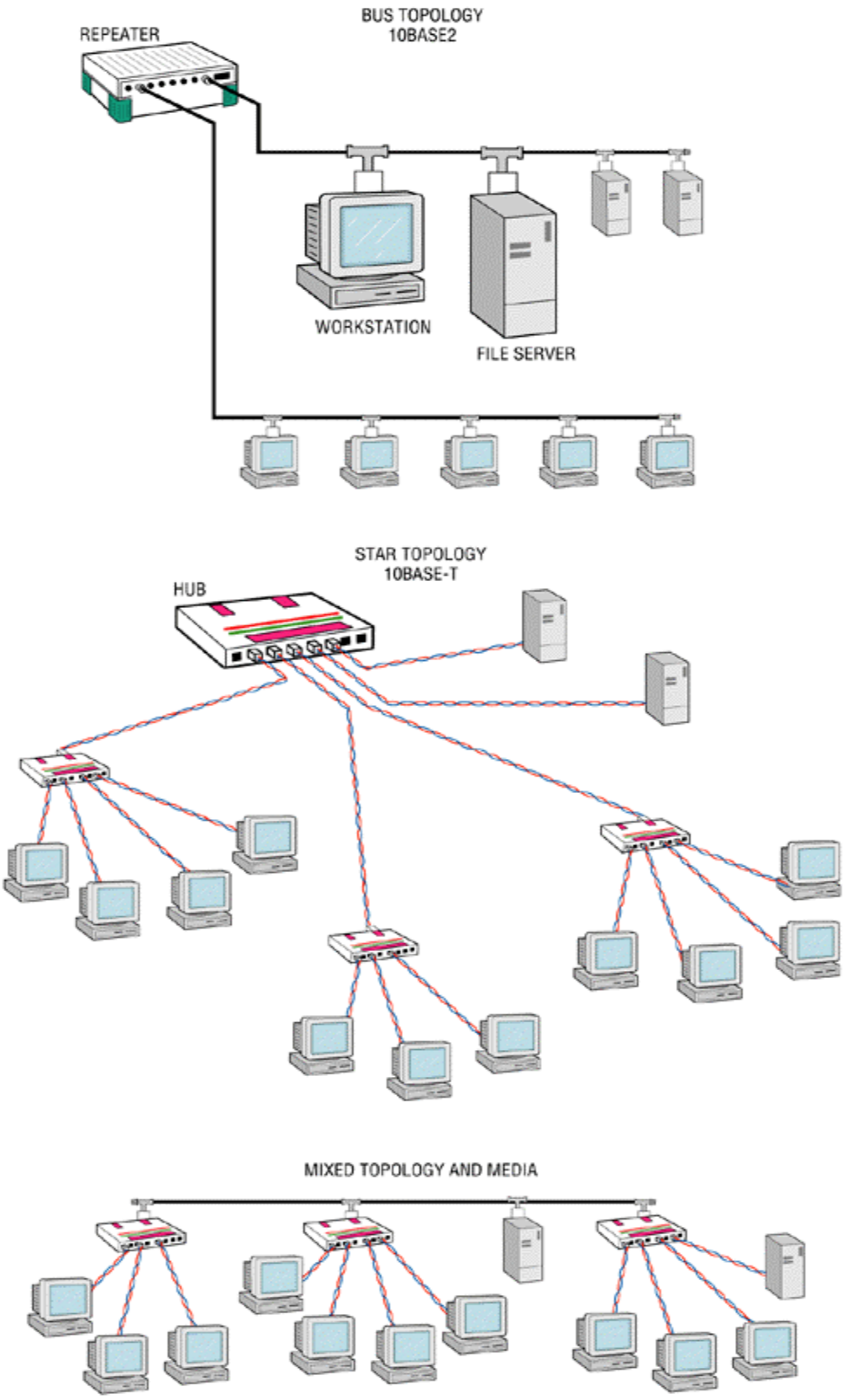

BUS TOPOLOGY
10BASE2
REPEATER
WORKSTATION
FILE SERVER
STAR TOPOLOGY
10BASE-T
HUB
MIXED TOPOLOGY AND MEDIA

Fig. 8

4.5.2 Reteaua Fast Ethernet

Pentru retelele Ethernet care necesitã viteze de comunicatie mai mari, a fost introdus standardul Fast Ethernet (IEEE 802.3u). Acest standard a ridicat limita de vitezã de la 10 Mbps la 100 Mbps cu modificări minimale la structura fizicã existentã, fiind foarte atractiv pentru îmbunãtãtirea retelelor Ethernet. Însã viteza ridicatã impune mai multe limitări în proiectarea acestor retele
Distantele maxime pentru segmentele de retea Fast Ethernet si numãrul de statii legate la o retea Fast Ethernet depind de tipul cablului de transmisie folosit:

	Viteza	Tipul de cablu	Numãr maxim de statii pe segment	Distanta maximã dintre statii (m)
100 Base T4	100 Mbps	UTP 3, 4 sau 5	2	100
100 Base TX	100 Mbps	UTP 5	2	100
100 Base FX	100 Mbps	Fibrã opticã	2	412

Tabelul 2

Determinarea întârzierii în propagare
Standardul Ethernet specificã lungimea minimã pentru un pachet la 512 biti. Astfel cã întârzierea introdusã de retea trebuie sã fie mai micã decât timpul de transmisie al celor 512 biti. Deoarece acest timp este foarte important în instalarea corectã a unei retele Fast Ethernet, este necesar a planifica reteaua pe baza lui:

- În primul pas se localizeazã cele douã noduri mai îndepãrtate unul de celãlalt;
- În pasul doi se determinã locul unde se va plasa hub-ul (sau hub-urile);
- Se însumeazã întârzierile introduse de fiecare dispozitiv si cablu (pe calea cea mai lungã) si se comparã cu 512. Dacã valoarea este mai micã reteaua este validã din punct de vedere teoretic.

Durata de propagare este mãsuratã în timp pe bit. Un timp bit este definit ca durata unui bit de date pe retea (pentru Fast Ethernet 10^{-8} s). Deoarece protocolul CSMA/CD cere ca primul bit al oricãrei transmisii sã ajungã în cel mai îndepãrtat punct al retelei înainte ca ultimul bit sã fie trimis, si dacã cel mai mic pachet are dimensiunea de 512 biti, rezultã cã reteaua trebuie proiectatã astfel încât în cel mai rãu caz sã avem întârzierea sub 512 timp bit.
Fiecare cablu si dispozitiv prin care trece semnalul de la un capãt la celãlalt va introduce o anumitã întârziere, dupa cum se poate urmari in tabelul 3:

Element	Întârzierea pe metru	Întârzierea maximã
Douã DTE TX/FX		100
Douã DTE T4		138
Un DTE T4 si unul TX/FX		127
Segment de cablu UTP 3	1.14	114 (100m)
Segment de cablu UTP 4	1.14	114 (100m)
Segment de cablu UTP 5	1.112	111.2 (100m)
Segment de cablu STP	1.112	111.2 (100m)
Segment de cablu optic	1.00	412 (412m)
Repetor de clasã I		140
Repetor de clasã II – TX/FX		92
Repetor de clasã II – T4		67
Convertor TX – FX		50-100

Tabelul .3

4.5.3 Proiectarea unei retele Ethernet

Proiectarea retelelor Ethernet si Fast Ethernet se bazează pe anumite reguli care trebuie urmate pentru ca acestea sã functioneze corect. Numãrul maxim de noduri, numãrul de repetoare si lungimile maxime ale segmentelor sunt determinate pe baza proprietãtilor electrice si mecanice ale fiecãrui tip de mediu Ethernet si respectiv Fast Ethernet. Dacã în proiectarea retelei nu se respectã regulile amintite, atunci nu vor fi respectate specificatiile pentru timpul de întoarcere, pierzându-se pachete si încãrcându-se traficul cu retransmiteri repetate.
La proiectarea unei retele Ethernet trebuie respectate urmãtoarele trei reguli (Fig. 9):

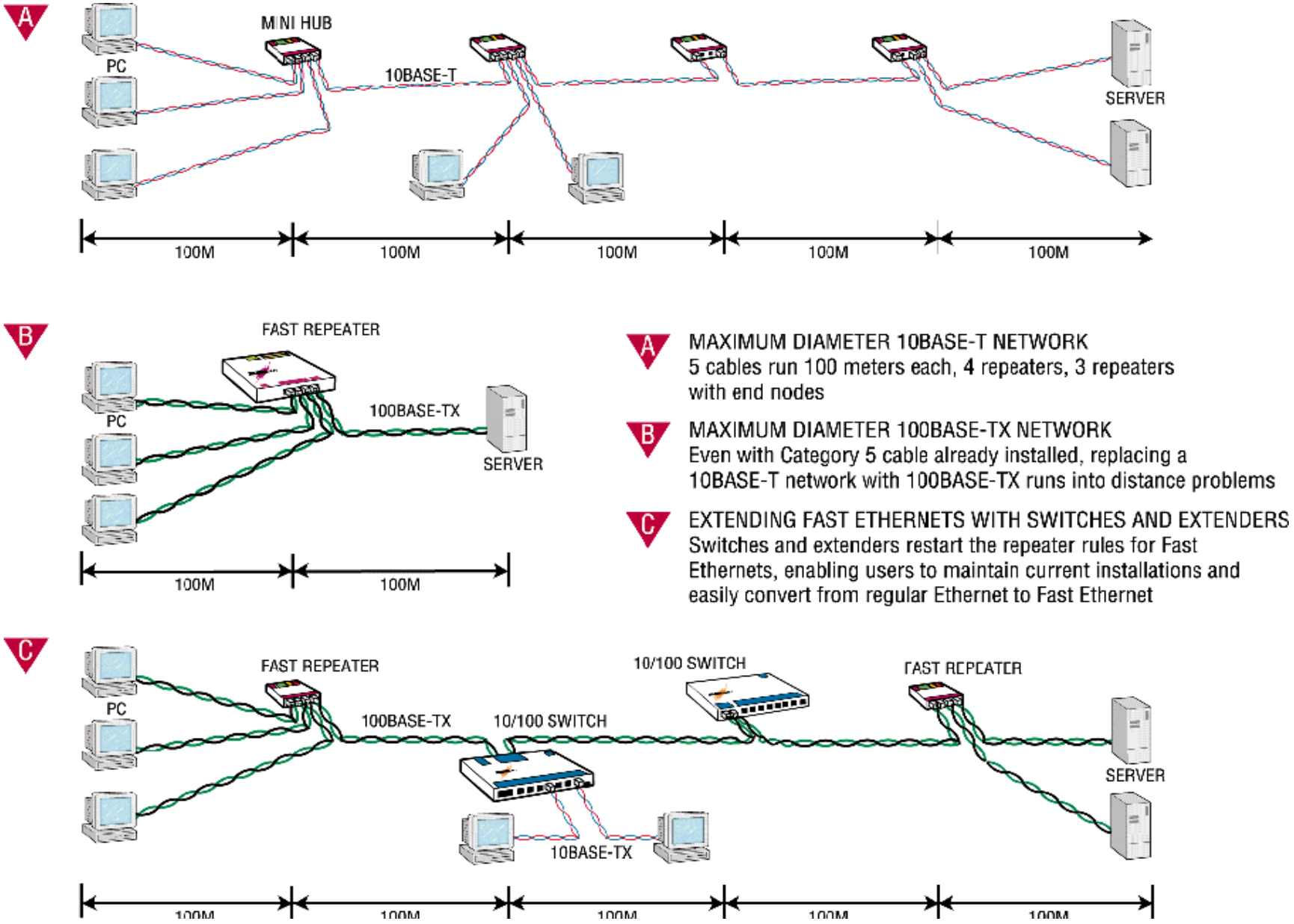


Fig. 9: Ilustrarea regulilor de proiectare a retelei Ethernet

- reteaua poate avea cel mult cinci segmente conectate (distanta maximã dintre nodurile retelei trebuie sã nu depãseascã 500m);
- se pot folosi doar patru repetoare;
- din cele cinci segmente, doar trei pot avea noduri atasate; celelalte douã trebuie sã fie legãturi între repetoare.

Fast Ethernet a modificat aceste reguli, deoarece unui pachet de dimensiune minimã îi ia mai putin timp propagarea prin mediul fizic decât în cazul unui Ethernet normal. Astfel cã pentru Fast Ethernet sunt permise mai putine repetoare. În retelele Fast Ethernet, existã douã tipuri de repetoare:

- repetoare de clasa I au o latentã de 0.7 μs si sunt limitate la un repetor pe retea;
- repetoare de clasa II au latenta de 0.46 μs si sunt limitate la douã repetoare pe retea

Proiectarea unei retele:

- cerintele de retea pentru fiecare statie;
- gruparea statiilor ce comunicã cel mai des între ele în acelasi segment;
- cãutarea modelelor de trafic pe departamente;
- evitarea gâtuirilor prin legãturi rapide pe acele portiuni;
- modificarea iterativã a statiilor în cadrul segmentelor pânã când toate nodurile ajung la o utilizare mai micã de 35%.

Cablarea retelei

Problemele cele mai dese care apar într-o retea Fast Ethernet sunt datorate instalãrii necorespunzãtoare a cablurilor. Astfel cã trebuie respectare câteva reguli si la cablarea retelei:

- pentru a se obtine o performantã maximã, trebuie folosite cabluri UTP de categoria 5;
- reteaua Fast Ethernet este foarte sensibilã la zgomotele electrice si la interferente, astfel cã trebuie sã se evite trecerea cablului de retea pe lângã linii de tensiune, lumini fluorescente sau orice alt echipament electric de putere.

Congestionarea retelei

Pe mãsurã ce creste numãrul de utilizatori, dimensiunea aplicatiilor si datelor vehiculate în retea, performantele acesteia se deteriorează datoritã folosirii mediului unic de comuniacatie.

Factorii care afectează eficienta unei retele:

- cantitatea traficului;
- numãrul de statii;
- dimensiunea pachetelor;
- dimensiunile fizice ale retelei.

Parametrii pentru mãsurarea eficientei unei retele Ethernet:

- raportul între încãrcarea maximã si cea medie;
- rata coliziunilor – procentajul pachetelor cu coliziuni din numãrul total de pachete;
- rata de utilizare – procentajul traficului total fatã de maximul teoretic pentru tipul de retea (10 Mbps).

Pentru determinarea acestor parametrii se pot folosi diferite utilitare de retea, luându-se în calcul atât valorile medii cât si cele maxime. O retea Ethernet functionează la parametrii optimi dacã rata coliziunilor nu depãseste 10% si dacã rata de utilizare este sub 35%. Timpul de rãspuns al retelei (performanta retelei transpusã în termenii utilizatorului) suferã pe mãsurã ce creste încãrcarea acesteia, iar la cresteri nesemnificative ale traficului (din punctul de vedere al utilizatorului) performanta descreste foarte mult. Aceasta deoarece în Ethernet, numãrul de coliziuni creste odatã cu cresterea încãrcãrii retelei, cauzând

retransmisii ce încarcã si mai mult reteaua, producând mai multe coliziuni. Supraîncãrcarea retelei îngreunând traficul considerabil.

Solutii pentru cresterea performantelor retelei:

- împãrtirea retelei în mai multe segmente ce întrã într-un repetor:
  - amplificarea semnalului;
- înlocuirea repetorului central cu un comutator:
  - conexiuni mai rapide la server(e);
  - izolarea traficului irelevant la fiecare segment de retea;
- adãugarea de comutatoare la backbone switched network – congestia unei retele comutate poate fi rezolvatã prin adãugarea de noi porturi de comutare si prin cresterea vitezei acestor porturi. Segmentele cu performantã scãzutã sunt identificate prin mãsurãtori de performantã si solutiile posibile sunt:
  - segmentarea în continuare a respectivei portiuni de retea;
  - conexiuni mai rapide (Fast Ethernet).

Modificãrile aduse unei retele sunt de cele mai multe ori evolutive si nu revolutionare; acestea fãcându-se încet si încercând a se pãstra cât mai mult din structura si echipamentele curente, înlocuindu-se doar cele învechite sau cele pentru care nu mai existã nici o altã solutie.

Fast Ethernet este foarte usor de adãugat la cele mai multe dintre retele. Un comutator sau o punte (bridge) permite conectarea unui Fast Ethernet la infrastructura Ethernet existentã pentru a îmbunãtãti viteza pe portiunile critice. Tehnologia mai rapidã este folositã pentru a conecta comutatoarele între ele pentru a se evita gâtuirile.

LUCRAREA NR. 5

Simularea unei rețele de tip token-ring

5.1 Tema lucrării

Lucrarea de față este dorită a fi o simulare a unei rețele LAN bazată pe protocolul Token-ring prin implementarea mecanismului de priorități al acestui protocol. Simularea trebuie să reflecte fenomenele petrecute într-o rețea Token-ring.

5.2 Noțiuni teoretice cu privire la rețelele Token-ring

5.2.1 Arhitectura rețelei

Un inel poate fi format nu numai dintr-un mediu unic de difuzare, ci şi din mai multe legături punct-la-punct, care formează un inel. Legăturile punct-la-punct pot fi de diferite tipuri: fire torsadate, cablu coaxial sau fibre optice. Deşi construcția inelului este în majoritate numerică, există şi părți analogice (ca de exemplu detecția coliziunilor la 802.3). Ideea utilizării unei structuri de inel real, nu numai logic, cu un canal având cunoscută limita superioară a debitului, a convenit firmei IBM care a adoptat-o pentru LAN - urile produse de ea. În consecință a apărut un al treilea standard pentru LAN – uri IEEE 802.5 **token-ring**, deşi există mai multe feluri de inel. O noțiune de bază în analiza şi proiectarea rețelelor inel este cea de **lungime fizică a bitului**. Dacă debitul datelor pe inel este de D Mbps, un bit este emis la fiecare 1/D μs, ceea ce reprezintă durata unui bit. La o viteză tipică de propagare de 200 m/μs, fiecare bit va ocupa 200/D metri de inel. Aceasta înseamnă că la un debit de 1Mbps, un inel poate conține la un moment dat doar 5 biți.

Inelul real constă dintr-o colecție de interfețe de inel conectate prin linii punct la punct. Fiecare bit ce soseşte la o interfață este copiat într-un tampon de 1 bit şi apoi recopiat înapoi în inel. Aflat în tampon, bitul este testat şi poate fi modificat înainte de a fi transmis din nou pe linie. Această etapă de copiere introduce o întârziere suplimentară de 1 bit la fiecare interfață. Configurația de inel este prezentată în figura 1.

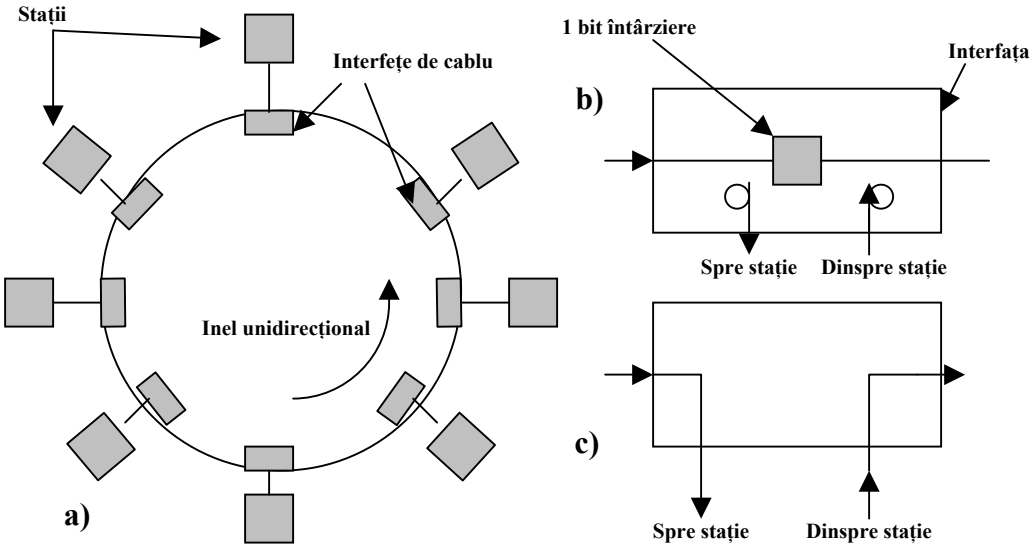


Fig. 1: Protocolul Token-ring 802.3 a)Reţea inel; b)Ascultare; c)Transmisie

În protocolul token-ring, de-a lungul inelului circulă o structură specială de biţi şi anume **token**-ul, de câte ori staţiile sunt inactive. Când o staţie vrea să transmită un cadru ea trebuie să capteze tokenul şi să-l trimită înapoi pe inel înainte de transmisia propriu-zisă. Deoarece există un singur token, numai o singură staţie poate transmite la un moment dat, rezolvându-se astfel problema accesului la canal la fel ca în protocolul token-bus.

Este necesar ca lungimea inelului să fie suficientă pentru ca pe el să încapă un token complet când staţiile sunt inactive. Întârzierea are două componente: întârzierea de 1 bit introdusă de fiecare staţie şi întârzierea cauzată de timpul de propagare. În majoritatea reţelelor de tip inel, staţiile pot fi deconectate pentru diferite perioade, mai ales noaptea. Dacă interfeţele sunt alimentate electric de la inel, deconectarea staţiei nu are nici un efect asupra interfeţei, dar dacă este alimentată extern, atunci ea trebuie proiectată să conecteze intrarea la ieşire când se deconectează tensiunea, eliminând astfel întârzierea de 1 bit. Pe inelele scurte, trebuie prevăzută o întârziere suplimentară pe timp de noapte, pentru a asigura că tokenul să fie conţinut în inel.

Interfeţele de inel au două moduri de lucru: **ascultare** şi **transmisie**. În modul ascultare biţii de intrare sunt pur şi simplu copiaţi la ieşire, cu o întârziere egală cu durata unui bit. În modul transmisie, care interesează doar dacă tokenul a fost achiziţionat de staţie, interfaţa întrerupe conexiunea între intrare şi ieşire, pe inel fiind lansate datele staţiei. Pentru a reuşi să comute din starea ascultare în starea transmisie într-un interval de timp egal cu durata unui bit, la interfeţele obişnuite este mai bine să se memoreze una sau mai multe cadre în interfaţă, decât ca

interfața să le extragă din stație într-un timp atât de scurt. Pe măsură ce biții transmişi pe inel se întorc înapoi, ei sunt eliminați de pe inel de transmițător. Stația care i-a transmis îi poate fie memora, pentru a-i compara cu secvența originală şi a verifica astfel fiabilitatea rețelei, fie să-i distrugă.

Această structură de inel nu impune limite asupra dimensiunii cadrelor, deoarece un cadru nu apare niciodată complet în inel. După ce stația termină de transmis ultimul bit al cadrului, ea trebuie să regenereze tokenul. Când ultimul bit al cadrului a fost transmis pe inel şi s-a întors el trebuie eliminat şi interfața trebuie comutată înapoi în modul ascultare, pentru a permite extragerea tokenului care ar urma dacă nici o altă stație nu l-a extras.

Confirmările sunt simplu de manevrat la token-ring. Formatul cadrului are doar un bit pentru confirmări, inițial pus pe zero. Când stația destinație a recepționat cadrul, ea inversează bitul. Dacă se foloseşte şi suma de control în mecanismul de confirmare, bitul trebuie să urmeze suma de control şi interfața de inel trebuie să fie capabilă să verifice suma de control până în momentul sosirii ultimului bit. Dacă se intenționează difuzarea cadrului la mai multe stații, mecanismul de confirmare este ceva mai complicat, dacă se utilizează unul pentru toate.

Când traficul este scăzut, tokenul îşi va consuma cea mai mare parte din timp circulând nefolosit pe inel. Ocazional, o stație îl captează, transmite un cadru şi relansează tokenul. Dimpotrivă, dacă traficul este încărcat, astfel încât se crează şiruri de aşteptare în fiecare stație, de îndată ce o stație îşi termină transmisiunea şi regenerează tokenul, acesta va fi preluat de următoarea stație din aval. Astfel, permisele de transmisie se rotesc încet pe inel. Eficiența rețelei este aproape de 100% la încărcări mari.

5.2.2 Mecanismul de priorități la token-ring

Standardul 802.5 include o specificație pentru un mecanism de priorități opțional. Sunt suportate 8 niveluri de prioritate furnizate de două câmpuri a câte 3 biți în fiecare cadru de date şi token: un câmp de prioritate şi un câmp de rezervare. Pentru a explica acest algoritm se definesc următoarele variabile:

P_f = prioritatea cadrului de transmis de către stație

P_s = prioritatea serviciului: prioritatea tokenului curent

P_r = valoarea lui P_s din ultimul token recepționat de către stație

R_s = valoare rezervată în tokenul curent

R_r = valoarea rezervată cea mai mare în cadrele recepționate de stație, din timpul ultimei rotiri a tokenului.

Se poate realiza o organigramă care să reflecte mecanismul de priorități la rețelele Token-Ring. Organigrama este prezentată în continuare.

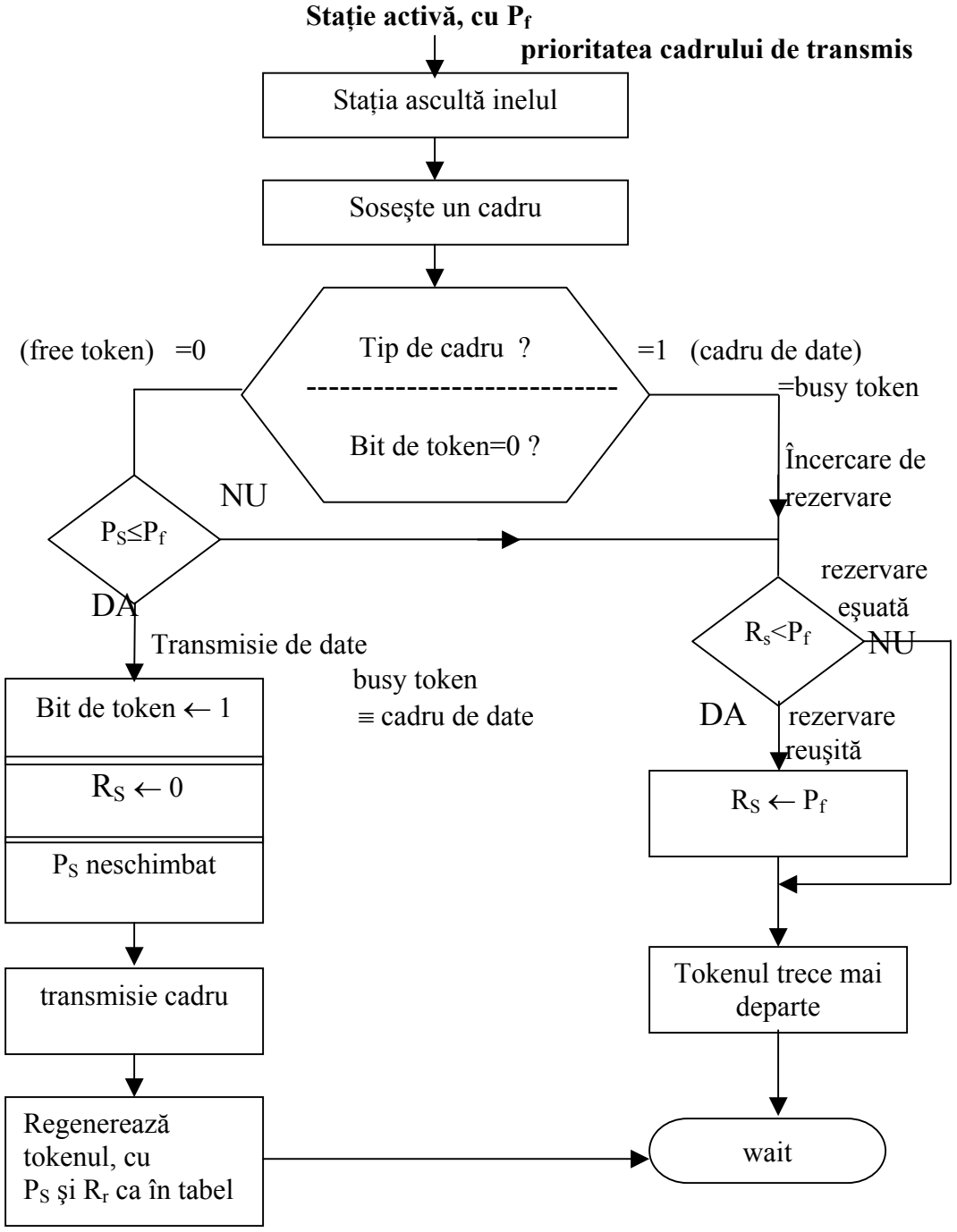


Fig.2: Organigramă a mecanismului de priorități

Schema lucrează astfel:

1. O stație care dorește să transmită trebuie să aştepte după un token cu $P_s \leq P_f$.
2. Când aşteaptă, stația poate rezerva un token viitor cu prioritatea sa P_f. Dacă trece un cadru de date şi câmpul de rezervare e inferior priorității sale, ($R_s<P_f$) atunci stația îşi înscrie prioritatea sa în câmpul de rezervare al cadrului ($P_f \rightarrow R_s$). Dacă trece un token şi ($R_s < P_f$ şi $P_f < P_s$), atunci stația îşi înscrie propria sa prioritate în câmpul de rezervare ($P_f \rightarrow R_s$). Rezultatul este surmontarea oricărei rezervări de prioritate inferioară.
3. Când stația captează un token, pune bitul de token pe 1, marcând astfel începutul unui cadru de date, pune 0 în câmpul de rezervare al cadrului de date şi lasă neschimbat câmpul de prioritate P_f (la fel ca al cadrului de token sosit).
4. După transmiterea unuia sau mai multor cadre de date, stația generează un nou token, cu câmpurile de rezervare şi priorităţile ca în tabelul de mai jos:

Condiții	Acțiuni
(Cadru disponibil) şi ($P_s \leq P_f$)	Transmite cadrul
(Cadru indisponibil sau THT expirat) şi ($P_r \geq$ MAX [R_r , P_f])	Transmite un token cu: $P_s \leftarrow P_f$ $R_s \leftarrow$ MAX[R_r , P_f]
(Cadru indisponibil sau THT expirat) şi ($P_r <$ MAX [R_r , P_f]) şi ($P_r > S_x$)	Transmite un token cu: $P_s \leftarrow$ MAX[R_r , P_f] $R_s \leftarrow 0$ Introduce în stiva S_r pe P_r Introduce în stiva S_x pe P_s
(Cadru indisponibil sau THT expirat) şi ($P_r <$ MAX [R_r , P_f]) şi ($P_r = S_x$)	Transmite un token cu: $P_s \leftarrow$ MAX[R_r , P_f] $R_s \leftarrow 0$ Scoate din stiva S_x Introduce în stiva S_x pe P_s
((Cadru indisponibil) sau (Cadru disponibil şi $P_f < S_x$)) şi ($P_s = S_x$) şi ($R_r > S_r$)	Transmite un token cu: $P_s \leftarrow R_r$ $R_s \leftarrow 0$ Scoate din stiva S_x Introduce în stiva S_x pe P_s
((Cadru indisponibil) sau (Cadru disponibil şi $P_f < S_x$)) şi ($P_s = S_x$) şi ($R_r \leq S_r$)	Transmite un token cu: $P_s \leftarrow R_r$ $R_s \leftarrow 0$ Scoate din stiva S_r; Scoate din stiva S_x

Fiecare stație e responsabilă de faptul ca nici un token să nu circule la infinit din cauză că are o prioritate prea mare. Memorând prioritatea transmisiei precedente, o stație poate detecta această condiție şi să degradeze prioritatea sau rezervarea la valoarea anterioară, mai scăzută. Aceasta înseamnă că atunci când stația vede un token liber de prioritate maximă, toate stațiile de prioritate maximă şi-au încheiat transmisia.

Pentru implementarea schemei de degradare, fiecare stație menține două stive:

- una pentru rezervare
- una pentru priorități

S_x – stiva pentru memorarea noilor valori de prioritate a tokenului

S_r – stiva pentru memorarea vechilor valorii de prioritate a tokenului

5.3 Desfăşurarea lucrării

5.3.1 Inițializarea inelului

Pentru a vizualiza modul în care se realizează inițializarea inelului se deschide fişierul *Token Ring Initialisation.html*, unde se marchează opțiunea cu comentariu pentru a putea beneficia de avantajele oferite de o fereastră de comentariu în care toate evenimentele petrecute în rețea sunt comunicate utilizatorului. Se poate varia viteza simulării după dorință, dar pentru o simulare foarte clară se recomandă o viteză a simulării de 120.

Se vor nota de către studenți evenimentele din rețea care duc la inițializarea rețelei, încercându-se înțelegerea fenomenului.

5.3.2 Simularea funcționării rețelei

Se va deschide fişierul *Token-Ring Simulation.html* şi se marchează opțiunea de comentariu. Se recomandă să se ruleze mai multe simulări, pentru cantități diferite de mesaje, de priorități şi mărimi diferite, pentru toate cazurile posibile de mediu, viteză a rețelei, etc.

Descrierea simulării

Simularea arată un număr de computere (stații) conectate prin cablu pentru a forma o rețea LAN (local area network) Token Ring. Afişajul stațiilor arată modul curent al stației. Modurile alternative sunt după cum urmează:

Această stație este disponibilă 

;

Această stație în mod curent recepționează 

;

Această stație in mod curent transmite 

;

Fiecare stație are un port de intrare/ieşire arătând astfel: ○ ataşat la stație.

Stațiile sunt conectate printr-un cablu conectat între două porturi de stații arătând astfel: ○——○ . Mesajele care trec între porturi de-a lungul cablului arată astfel: ○—■—○ , cu mesajul ■ mergând de-a lungul cablului. Un mesaj duce o etichetă cu el pentru a descrie natura şi conținutul mesajului.

Această simulare foloseşte o mărime a cadrului mesaj de 3 octeți (egal cu mărimea unui Token într-o rețea Token Ring). Fiecare mesaj ■ reprezintă 3 octeți de date.

O listă de diferite mesaje folosite în simulare este după cum urmează:

■**T** indică un token liber

■**Ms** indică primii 3 octeți ai unui cadru de date

■**M** indică partea de mijloc a unui cadru de date (3 octeți)

■**Me** indică ultimii 3 octeți ai unui cadru de date

■**DAT** indică un cadru de test de adresă duplicată

■**SMP** indică un cadru de standby monitor prezent

■**CT** indică un cadru de cerere de token

■**PRG** indică un cadru de curățare (purge)

■**AMP** indică un cadru monitor activ prezent

■**BCN** indică un cadru de test de avertizare (beacon)

Afişajul de lângă stație asigură informație adițională despre starea curentă a stației.

Următorii parametri sunt folosiți pentru a descrie starea curentă:

Transmise - numărul mesajelor complete pe care această stație le-a transmis cu succes.

Recepționate - numărul mesajelor complete pe care această stație le-a recepționat cu succes.

Aşteptând - numărul mesajelor pe care această stație aşteaptă să le transmită.

Prioritate - prioritatea curentă a stației aşteptând să transmită.

Adresa - adresa stației în rețea.

UNA - adresa vecinului din amonte al stațiilor.

AMP - stațiile văd al cărei adrese de stație este monitorul activ prezent curent.

Configurația aleasă pentru simulare este cazul în care o stație transmite la stația opusă pe diagonală.

Simularea este controlată de toolbar-ul din partea de jos a ecranului:

Bara de scroll selectează viteza simulării şi poate fi modificată pe parcursul simulării. Viteza este afişată în stânga şi variază de la 1 la 999. Viteza simulării va depinde de platforma folosită şi de simulare. În general o simulare rapidă, dar vizibilă, cere o viteza in jurul a 40, în timp ce o simulare mai lentă, dar clară, cere o viteză în jurul a 120.

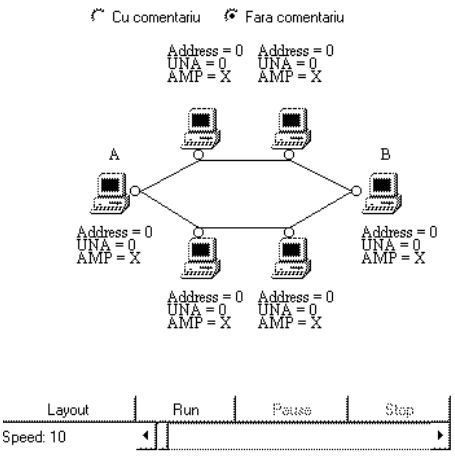
Simularea se porneşte apăsând butonul "Run" si poate fi pusă în pauză cu butonul "Pause" şi oprită cu butonul "Stop". O simulare pusă în pauză poate fi repornită cu butonul de "Restart" care înlocuieşte butonul "Pause" după ce a fost apăsat. Apăsând butonul "Stop" se termină simularea curentă. Dacă parametri sunt modificați în simulare (de ex. numărul stațiilor), butonul "Layout" redesenează imaginea simulării. Un re-layout este de asemenea realizat când o simulare este repornită.

Anexe

A1- Inițializarea inelului
a)începutul inițializării

Initializare Token Ring

Retele de Comunicatii de Date



Descriere

Toate statiile emit test de adresă duplicată (DAT)

La întoarcerea lui DAT, statiile emit monitor prezent (SMP)

Adresa vecinului din amonte (upstream neighbour address (UNA)) este determinată

La întoarcerea lui SMP statiile reclamă tokenul(CT)

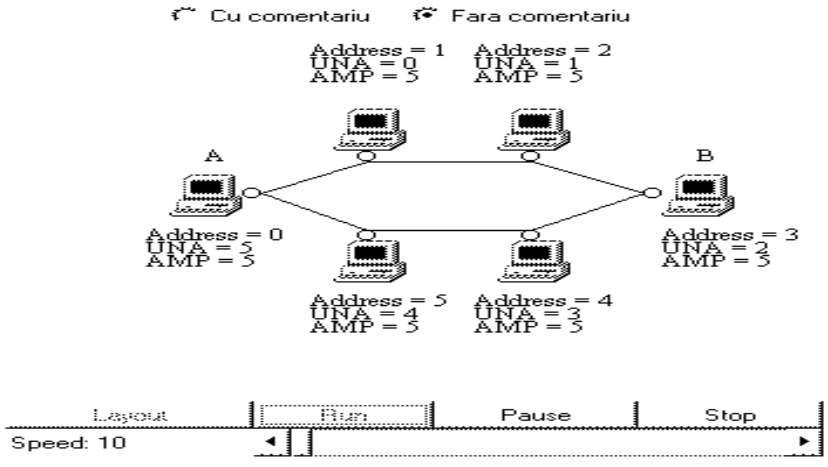
Dacă o statie de prioritate mai mare a emis o cerere de token nu se mai emite CT

Dacă CT se intoarce la emitătorul ei, această statie devine monitor activ

Monitorul activ emite un cadru purge (PRG)

Se emite monitor activ prezent(AMP) si token (T)

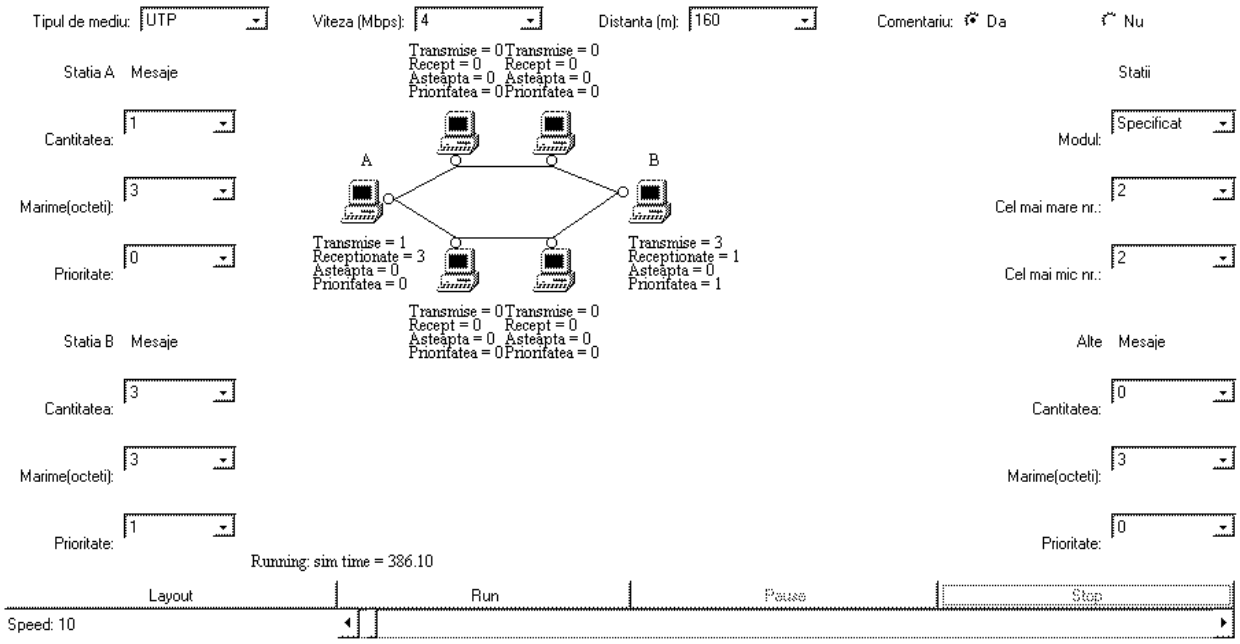
b)Inelul este inițializat



Statia 0 emite uun cadru DAT.
Statia 1 emite uun cadru DAT.
Statia 2 emite uun cadru DAT.
Statia 3 emite uun cadru DAT.
Statia 4 emite uun cadru DAT.
Statia 5 emite uun cadru DAT.
Statia 0 receptioneaza un cadru DAT de la statia 5.
Statia 1 receptioneaza un cadru DAT de la statia 0.
Statia 3 receptioneaza un cadru DAT de la statia 2.
Statia 4 receptioneaza un cadru DAT de la statia 3.
Statia 2 receptioneaza un cadru DAT de la statia 1.
Statia 5 receptioneaza un cadru DAT de la statia 4.
Statia 0 receptioneaza un cadru DAT de la statia 4.
Statia 1 receptioneaza un cadru DAT de la statia 5.
Statia 2 receptioneaza un cadru DAT de la statia 0.
Statia 3 receptioneaza un cadru DAT de la statia 1.

Statia 4 receptioneaza un cadru DAT de la statia 2.
Statia 5 receptioneaza un cadru DAT de la statia 3.
Statia 0 receptioneaza un cadru DAT de la statia 3.
Statia 1 receptioneaza un cadru DAT de la statia 4.
Statia 2 receptioneaza un cadru DAT de la statia 5.
Statia 3 receptioneaza un cadru DAT de la statia 0.
Statia 4 receptioneaza un cadru DAT de la statia 1.
Statia 5 receptioneaza un cadru DAT de la statia 2.
Statia 0 receptioneaza un cadru DAT de la statia 2.
Statia 1 receptioneaza un cadru DAT de la statia 3.
Statia 2 receptioneaza un cadru DAT de la statia 4.
Statia 3 receptioneaza un cadru DAT de la statia 5.
Statia 4 receptioneaza un cadru DAT de la statia 0.
Statia 5 receptioneaza un cadru DAT de la statia 1.
Statia 0 receptioneaza un cadru DAT de la statia 1.
Statia 1 receptioneaza un cadru DAT de la statia 2.
Statia 2 receptioneaza un cadru DAT de la statia 3.
Statia 3 receptioneaza un cadru DAT de la statia 4.
Statia 4 receptioneaza un cadru DAT de la statia 5.
Statia 5 receptioneaza un cadru DAT de la statia 0.
Statia 0 receptioneaza un cadru DAT de la statia 0.
Statia 1 receptioneaza un cadru DAT de la statia 1.
Statia 2 receptioneaza un cadru DAT de la statia 2.
Statia 4 receptioneaza un cadru DAT de la statia 4.
Statia 5 receptioneaza un cadru DAT de la statia 5.
Statia 3 receptioneaza un cadru DAT de la statia 3.
Statia 0 nu a gasit nici o adresa duplicata.
Statia 2 nu a gasit nici o adresa duplicata.
Statia 1 nu a gasit nici o adresa duplicata.
Statia 3 nu a gasit nici o adresa duplicata.
Statia 0 a transmis un cadru SMP.
Statia 2 a transmis un cadru SMP.
Statia 5 nu a gasit nici o adresa duplicata.
Statia 4 nu a gasit nici o adresa duplicata.
Statia 3 a transmis un cadru SMP.
Statia 5 a transmis un cadru SMP.
Statia 4 a transmis un cadru SMP.
Statia 1 a transmis un cadru SMP.
Statia 0 a receptionat un cadru SMP de la statia 5.
Statia 1 a receptionat un cadru SMP de la statia 0.
Statia 2 a receptionat un cadru SMP de la statia 1.
Statia 4 a receptionat un cadru SMP de la statia 3.
Statia 5 a receptionat un cadru SMP de la statia 4.
Statia 3 a receptionat un cadru SMP de la statia 2.
Statia 1 isi reinoieste UNA.
Statia 2 isi reinoieste UNA.
Statia 4 isi reinoieste UNA.
Statia 3 isi reinoieste UNA.
Statia 0 isi reinoieste UNA.
Statia 5 isi reinoieste UNA.
Statia 0 a receptionat un cadru SMP de la statia 4.
Statia 1 a receptionat un cadru SMP de la statia 5.
Statia 2 a receptionat un cadru SMP de la statia 0.
Statia 3 a receptionat un cadru SMP de la statia 1.
Statia 4 a receptionat un cadru SMP de la statia 2.
Statia 5 a receptionat un cadru SMP de la statia 3.
Statia 0 a receptionat un cadru SMP de la statia 3.
Statia 1 a receptionat un cadru SMP de la statia 4.
Statia 2 a receptionat un cadru SMP de la statia 5.
Statia 3 a receptionat un cadru SMP de la statia 0.
Statia 4 a receptionat un cadru SMP de la statia 1.
Statia 5 a receptionat un cadru SMP de la statia 2.
Statia 0 a receptionat un cadru SMP de la statia 2.
Statia 1 a receptionat un cadru SMP de la statia 3.
Statia 2 a receptionat un cadru SMP de la statia 4.
Statia 3 a receptionat un cadru SMP de la statia 5.
Statia 4 a receptionat un cadru SMP de la statia 0.
Statia 5 a receptionat un cadru SMP de la statia 1.
Statia 0 a receptionat un cadru SMP de la statia 1.
Statia 1 a receptionat un cadru SMP de la statia 2.
Statia 2 a receptionat un cadru SMP de la statia 3.
Statia 3 a receptionat un cadru SMP de la statia 4.
Statia 4 a receptionat un cadru SMP de la statia 5.
Statia 5 a receptionat un cadru SMP de la statia 0.
Statia 0 a receptionat un cadru SMP de la statia 0.
Statia 1 a receptionat un cadru SMP de la statia 1.
Statia 2 a receptionat un cadru SMP de la statia 2.
Statia 3 a receptionat un cadru SMP de la statia 3.
Statia 4 a receptionat un cadru SMP de la statia 4.
Statia 5 a receptionat un cadru SMP de la statia 5.
Statia 0 a receptionat un cadru SMP de la statia 5.
Statia 1 a receptionat un cadru SMP de la statia 0.
Statia 2 a receptionat un cadru SMP de la statia 1.
Statia 3 a receptionat un cadru SMP de la statia 2.
Statia 4 a receptionat un cadru SMP de la statia 3.
Statia 5 a receptionat un cadru SMP de la statia 4.
Statia 0 a receptionat un cadru SMP de la statia 4.
Statia 1 a receptionat un cadru SMP de la statia 5.
Statia 2 a receptionat un cadru SMP de la statia 0.
Statia 3 a receptionat un cadru SMP de la statia 1.
Statia 5 a receptionat un cadru SMP de la statia 3.
Statia 2 nu a receptionat un cadru AMP deci transmite CT.
Statia 5 nu a receptionat un cadru AMP deci transmite CT.
Statia 1 nu a receptionat un cadru AMP deci transmite CT.
Statia 0 nu a receptionat un cadru AMP deci transmite CT.
Statia 4 a receptionat un cadru SMP de la statia 2.
Statia 4 nu a receptionat un cadru AMP deci transmite CT.
Statia 3 nu a receptionat un cadru AMP deci transmite CT.
Statia 0 isi da seama ca statia 5 a facut o oferta.
Statia 1 isi da seama ca statia 0 a facut o oferta.
Statia 2 isi da seama ca statia 1 a facut o oferta.
Statia 3 isi da seama ca statia 2 a facut o oferta.
Statia 4 isi da seama ca statia 3 a facut o oferta.
Statia 5 isi da seama ca statia 4 a facut o oferta.
Statia 0 inainteaza o oferta de adresa mai mare de la statia 5.
Statia 2 are o oferta mai mica de la statia 1 deci incheie transmisia lui CT.
Statia 3 are o oferta mai mica de la statia 2 deci incheie transmisia lui CT.
Statia 5 are o oferta mai mica de la statia 4 deci incheie transmisia lui CT.
Statia 1 are o oferta mai mica de la statia 0 deci incheie transmisia lui CT.
Statia 4 are o oferta mai mica de la statia 3 deci incheie transmisia lui CT.
Statia 1 isi da seama ca statia 5 a facut o oferta.
Statia 1 inainteaza o oferta de adresa mai mare de la statia 5.
Statia 2 isi da seama ca statia 5 a facut o oferta.
Statia 2 inainteaza o oferta de adresa mai mare de la statia 5.
Statia 3 isi da seama ca statia 5 a facut o oferta.
Statia 3 inainteaza o oferta de adresa mai mare de la statia 5.
Statia 4 isi da seama ca statia 5 a facut o oferta.
Statia 4 inainteaza o oferta de adresa mai mare de la statia 5.
Statia 5 isi da seama ca statia 5 a facut o oferta.
Statia 5 atunci devine monitor activ.
Monitorul activ transmite un cadru PRG pentru a curata inelul.
Monitorul activ transmite un cadru AMP.
Monitorul activ transmite un token liber T.
Intializare completa.
Transmisia normala poate incepe.
Cadru periodic AMP transmis de monitorul activ.

A2 Exemplu de funcționare a rețelei



- **Stația A emite tokenul, care este capturat de către stația B**

Statia 0 este monitorul activ.
Statia 0 elibereaza token cu prioritatea=0.

- **Stația B deține tokenul şi transmite primul pachet**

Statia 3 transmite cu prioritatea tokenului=0.

- **Stația B primeşte pachetul trimis care a efectuat o rotație completă pe inel şi emite un token**

Statia 3 a crescut prioritatea tokenului la1.
Se introducee noua prioritate=1 in Sx.
Statia 3 a crescut prioritatea tokenului la 1.
Statia 3 pune in stiva: Sx = 1 Sr = 0.
S-a realizat eliberare normala de token la statia 3.
Token eliberat cu prioritatea=1.
Token eliberat cu rezervarea=0.
Statia 3 este o statie care lucreaza cu stiva.

- **Întrucât $P_s > P_f$, stația A nu poate capta tokenul pentru a emite , astfel încât acesta revine la stația B, care are din nou dreptul de emisie (pentru cel de-al 2-lea pachet)**

Prioritatea din stiva Sr este egala sau mai mare decat prioritatea rezervata.
Prioritatea tokenului a fost redusa la 0.
Scoate prioritatea din stiva Sx
Statia 3 nu mai este o statie care lucreaza cu stiva.
Statia 3 transmite cu prioritatea tokenului=0.

- **Stația B primeşte pachetul trimis care a efectuat o rotație completă pe inel şi emite un token**

Statia 3 a crescut prioritatea tokenului la1.
Se introducee noua prioritate=1 in Sx.
Statia 3 a crescut prioritatea tokenului la 1.
Statia 3 pune in stiva: Sx = 1 Sr = 0.

S-a realizat eliberare normala de token la statia 3.
Token eliberat cu prioritatea=1.
Token eliberat cu rezervarea=0.
Statia 3 este o statie care lucreaza cu stiva.

- **Întrucât $P_s>P_f$, stația A nu poate capta tokenul pentru a emite , astfel încât acesta revine la stația B, care emite ultimul cadru**

Prioritatea din stiva Sr este egala sau mai mare decat prioritatea rezervata.
Prioritatea tokenului a fost redusa la 0.
Scoate prioritatea din stiva Sx
Statia 3 nu mai este o statie care lucreaza cu stiva.
Statia 3 transmite cu prioritatea tokenului=0.

- **Stația B primeşte pachetul trimis care a efectuat o rotație completă pe inel şi emite un token cu prioritatea 1**

Statia 3 a crescut prioritatea tokenului la1.
Se introducee noua prioritate=1 in Sx.
Statia 3 a crescut prioritatea tokenului la 1.
Statia 3 pune in stiva: Sx = 1 Sr = 0.
S-a realizat eliberare normala de token la statia 3.
Token eliberat cu prioritatea=1.
Token eliberat cu rezervarea=0.
Statia 3 este o statie care lucreaza cu stiva.

- **Tokenul revine la stația B, care nu mai are însă nici un pachet de transmis. Prin urmare stația B decrementează prioritatea tokenul şi emite un nou token cu prioritate =0**

Prioritatea din stiva Sr este egala sau mai mare decat prioritatea rezervata.
Prioritatea tokenului a fost redusa la 0.
Scoate prioritatea din stiva Sx
Statia 3 nu mai este o statie care lucreaza cu stiva.

- **Acest token poate fi capturat de către stația A, întrucât $P_s=P_f$. Stația A capătă astfel dreptul la emisie şi emite un cadru de informație**

Statia 0 transmite cu prioritatea tokenului=0.

- **Stația A primeşte pachetul trimis care a efectuat o rotație completă pe inel şi emite un token cu prioritatea 0**

S-a realizat eliberare normala de token la statia 0.
Token eliberat cu prioritatea=0.
Token eliberat cu rezervarea=0.

Lucrarea nr. 5

Adresele Internet

Internet-ul se poate defini ca o retea virtualã construitã pe interconectarea retelelor fizice prin intermediul unor sisteme dedicate numite gateway. În aceastã lucrare vom discuta despre adresare - ingredientul principal care ajutã software-ul TCP/IP sã ascundã detaliile retelelor fizice pe care le parcurge, si face posibilã conceperea internet-ului ca o entitate singularã si uniformã.

O **retea** se defineste ca o colectie de sisteme gazdã (hosts) capabile sã comunice între ele, bazându-se pe serviciile unor sisteme dedicate care transferã datele între participanti. Prin **host** se întelege de cele mai multe ori un calculator, dar acest termen nu se referã doar la calculatoare; alte sisteme gazdã ar mai putea fi terminale X, imprimante de retea si altele. Micile grupări de sisteme gazdã se numesc **site**-uri.

5.1 Generalitãti

5.1.1 Identificatori universali

Se spune despre un sistem de comunicatie cã oferã un serviciu de comunicatie universal dacã permite fiecãrui sistem gazdã (host) sã comunice cu orice alt sistem. Pentru a face sistemul nostru de comunicatie sã fie universal, trebuie sã stabilim o metodã global acceptatã de identificare a calculatoarelor.
Identificarea unei statii se poate face prin:

1. nume (ce este un obiect);
2. adresã (unde se gãseste obiectul);
3. rutã (cum se poate ajunge la el).

Fiecare dintre atributele de mai sus se reduc la niste simpli identificatori. Pentru om cel mai simplu mod de identificare îl reprezintã numele. Însã în alegerea (pentru internet) a unui identificator universal putea fi ales oricare dintre atributele amintite. Totusi a fost ales modul de identificare binarã prin adresã a unei anumite statii pentru a eficientiza luarea deciziilor de rutare.

5.1.2 Clasele de adrese IP

Internet-ul se poate gândi ca si orice altã retea fizicã, cu diferenta cã internet-ul este o structurã virtualã implementat în întregime în software. Astfel proiectantii nu au fost constrânsi în alegerea formatului si dimensiunii pachetelor, a adreselor, de nici o caracteristicã (limitare) hardware. Pentru adrese, proiectantii TCP/IP au ales o schemã analoagã cu modul de adresare din retelele fizice, în fiecare statie are atribuit un unic numãr întreg numit "adresã internet" sau "adresã IP". Însã aceste adrese au fost alese astfel încât sã facã rutarea pachetelor cât mai eficientã. Si anume, o adresã IP codificã informatia despre reteaua fizicã de care apartine o anumitã statie si informatia de identificare a statiei în cadrul retelei.

Fiecare statie gazdã din internet are atribuitã o unicã adresã internet pe 32 de biti care este folositã în toate comunicatiile cu statia respectivã. Aceastã adresã este o pereche de tipul (*netid*, *hostid*) unde *netid* este un identificator de retea, iar *hostid* identificã o statie

din cadrul retelei *netid*. În practicã fiecare adresã IP are una dintre formele prezentate în figura 1.

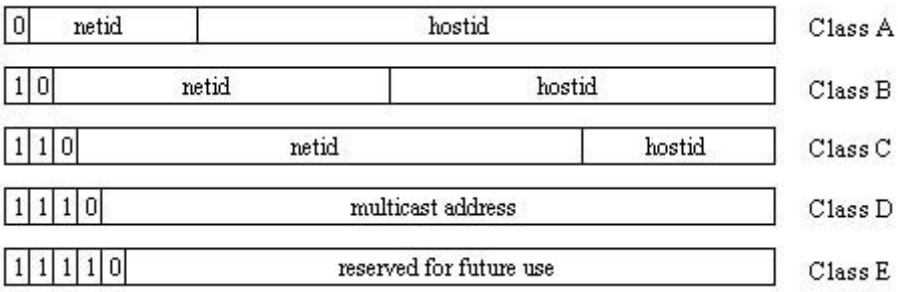


Fig. 1: Formatul adreselor IP

Pentru o anumitã adresã IP datã se poate determina clasa din care face parte pe baza celor mai semnificativi trei biti din adresã. Adresele de clasã A sunt folosite pentru retelele de dimensiuni foarte mari, care au mai mult de 2^{16} statii, au 7 biti pentru *netid* si 24 biti pentru *hostid*. Adresele de clasã B, care sunt folosite pentru retelele de dimensiuni medii având un numãr de statii între 2^8 si 2^{16}, au alocati 14 biti pentru *netid* si 16 biti pentru *hostid*. Iar adresele de clasã C, folosite în retelele de dimensiuni mici cu pânã la 2^8 statii, au alocati 21 biti pentru *netid* si 8 biti pentru *hostid*. Trebuie sã remarcãm cã adresele IP au fost astfel concepute pentru a se putea extrage cât mai simplu si rapid identificatorii *netid* si *hostid*.

5.1.3 Adresele specificã conexiunile la retea

Orice adresã IP identificã o unicã statie din retea, dar o statie poate avea alocate mai multe adrese IP. Astfel pentru fiecare conexiune a unei statii la o anumitã retea trebuie alocatã o adresã IP distinctã. Acest lucru este impus de codificarea adresei retelei în adresa IP.

5.1.4 Adresele de retea, de broadcast si de loopback

Un alt avantaj al codificãrii informatiei de retea în adresa IP este acela cã se poate face referire la o retea în acelasi mod ca si la o statie. Prin conventie o valoare 0 de *hostid* nu este atribuitã nici unei statii dintr-o retea. Însã o adresã IP cu câmpul *hostid* 0 este folositã pentru a identifica reteaua.

De asemenea avantaj semnificativ al acestui tip de adresare este cã include o "adresã de broadcast", care identificã toate statiile unei retele. Astfel o adresã a cãrei *hostid* are "1" pe toate pozitiile este rezervatã pentru broadcast. Acest tip de adresare se numeste adresare de broadcast directã, deoarece contine atât adresa de retea cât si cea de statie.
Un alt mod de adresare de tip broadcast, numitã adresare de broadcast limitatã, oferã o adresã de broadcast pentru reteaua localã, indiferent de adresa IP alocatã ei. Adresa de broadcast localã este formatã din 32 biti de "1". Acest tip de adresare poate fi folosit în rutina de boot-are pentru a afla adresa IP a retelei locale.

Adresa IP de clasã A 127.0.0.0 este rezervatã pentru "loopback" si este folositã în testarea comunicatiilor interprocese de pe masina localã. Dacã un program foloseste adresa de loopback pentru a trimite date, software-ul pentru protocol al calculatorului returneazã datele fãrã a le mai trimite pe nici o retea.

5.1.5 Notatia zecimalã cu punct

În interfata cu utilizatorul adresele IP sunt scrise ca patru numere întregi zecimale separate prin punct, unde fiecare întreg reprezintã valoarea zecimalã a unui octet din adresa IP. Astfel adresa IP

11000001.11100010.00001000.01101111

se scrie

193.226.8.111

5.1.6 Slãbiciunile adresãrii internet

Codificarea informatiei despre retea în adresa IP are câteva dezavantaje. Primul mare dezavantaj este cã o adresã este atribuitã unei conexiuni si nu unei statii. Astfel, dacã o statie se mutã de pe o retea pe alta, trebuie sã i se modifice si adresa IP.
O altã slãbiciune a acestui mod de adresare apare în cazul unei retele de clasã C, dacã numãrul de statii din retea depãseste 255. În aceastã situatie trebuie obtinutã o adresã de clasã B si trebuie modificate toate adresele IP din retea la noua adresã. Aceastã operatie este destul de mare consumatoare de timp.

Cea mai importantã lipsã a adresãrii internet apare în procesul de rutare. Deoarece rutarea foloseste portiunea de retea din adresa IP, calea pe care o urmeazã un anumit pachet IP în drumul lui cãtre o anumitã statie cu adresare IP multiplã depinde de adresa folositã în comunicatie. Astfel pentru situatia din figura 2.2 dacã statia A transmite un pachet statiei B, identificatã prin adresa corespunzãtoare conexiunii I_4, pachetul va urma urmãtoarea cale:
Statia A -> Conexiunea I_3 -> Reteaua 1 -> Conexiunea I_4 -> Statia B.
Iar dacã statia A transmite un pachet statiei B la adresa corespunzãtoare conexiunii I_5, acesta va urma calea:
Statia A -> Conexiunea I_3 -> Reteaua 1 -> Conexiunea I_1 -> Gateway -> Conexiunea I_2 ->Reteaua 2-> Conexiunea I_5 -> Statia B

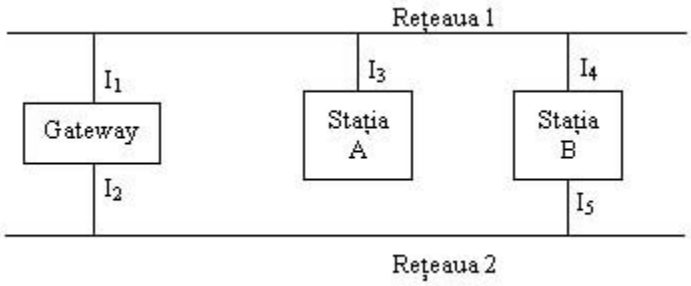


Fig. 2: Ilustarea deficientelor formatului IP

O altã comportare stranie apare dacã se întrerupe (defectează) una dintre conexiunile la retea al statiei B. Presupunem cã se întrerupe conexiunea I_4 iar statia B este functionalã si la fel este si conexiunea I_5 a acesteia. Atunci dacã statia A trimite un pachet pe adresa I_4 acesta va fi pierdut si programele care folosesc aceastã adresã nu pot comunica cu statia B, în schimb programele care folosesc adresa I_5 vor putea comunica cu statia B.

5.1.7 Ordinea octetilor în retea

Pentru a crea o retea globalã independentã de hardware trebuie definit un mod standard de reprezentare a datelor. Acest standard este necesar datoritã ordinii diferite de stocare a cuvintelor în memorie. Astfel unele sisteme salveazã numerele cu octetul mai putin semnificativ la adresele mai mici (Little Endian), iar altele cu octetul mai semnificativ la adresele mai mici (Big Endian).

Standardizarea ordinii octetilor pentru numere este importantã deoarece pachetele internet contin numere binare care reprezintã informatii specifice cum ar fi adresa destinatiei, lungimea pachetului, etc. Aceste date trebuie sã fie corect întelese de atât de statia care trimite pachetul cât si de cea care îl receptioneazã. Protocolul TCP/IP rezolvã aceastã problemã prin definirea unui standard al ordinii octetilor în retea, care este folosit de toate masinile pentru citirea câmpurilor binare dintr-un pachet. Acest standard foloseste stilul Big Endian.

5.2 Maparea adreselor IP în adrese fizice

5.2.1 Rezolutia adreselor

Sã considerãm douã masini A si B care sunt legate la o aceeasi retea fizicã. Fiecare are asignatã câte o adresã IP I_A si I_B si câte o adresã fizicã P_A si P_B. Scopul nostru ar fi sã vedem care este rolul software-ului de nivel scãzut care ascunde adresele fizice si permite programelor de nivel înalt sã lucreze cu adresele IP ale statiilor. Presupunem cã statia A doreste sã transmitã un pachet statiei B prin reteaua fizicã la care sunt ambele statii legate, dar statia A cunoaste doar adresa IP I_B a statiei B. Întrebarea care se ridicã este: Cum se mapeazã o adresã IP în adresa fizicã corespunzãtoare statiei destinatie.
Problema mapãrii adreselor de nivel înalt în adrese fizice este cunoscutã sub numele de "problema rezolutiei adreselor" si a fost rezolvatã în câteva moduri. Unele soft-uri de protocol mentin tabele pe care fiecare masinã care contin perechi de adrese de nivel înalt si fizice.

O altã solutie ar fi codificarea adreselor de nivel înalt în adrese fizice pe baza unei functii. Aceastã a doua metodã este mai simplã si este aplicabilã retelelor fizice care au formatul de adresã scurt si usor configurabil pentru o statie. În acest caz avem nevoie de o functie f care mapeazã adresele IP în adrese fizice, astfel încât

$$P_A = f(I_A)$$

iar adresele fizice se aleg pe baza relatiei de mai sus.

5.2.2 Rezolutia prin legare dinamicã

În acest subcapitol vom lua drept exemplu cazul retelelor Ethernet. O placã de retea Ethernet are adresa pe 48 de biti, adresã stabilitã de cãtre producãtorul plãcii, iar aceastã adresã nu poate fi modificatã. Ca o consecintã, dacã o placã de retea se defectează si aceasta se înlocuieste, masina în cauzã va avea o altã adresã fizicã. Mai mult, deoarece adresa Ethernet este pe 48 de biti, nu este nici o posibilitate de a o codifica pe cei 32 de biti ai adresei IP.

Solutia aleasã permite ca o nouã masinã sã fie adãugatã în retea fãrã a recompila codul, si nu necesitã mentinerea unei baze de date centralizatã. Pentru evitarea mentinerii unui tabel de mapare centralizat, proiectantii Internet au ales un protocol de nivel scãzut

care leagã adresele dinamic. Acest protocol este cunoscut sub numele de "Address Resolution Protocol" (ARP).

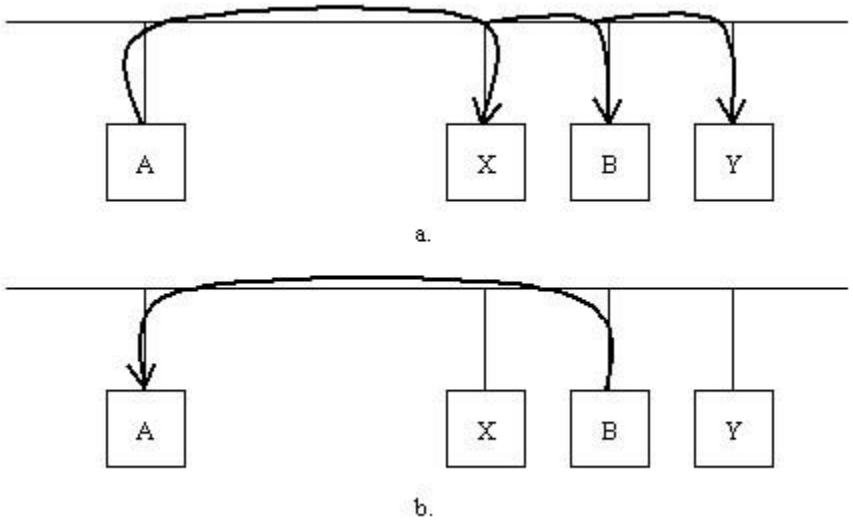


Fig. 3: Rezolutia dinamica ARP

Ideea pe care se bazează rezolutia dinamicã cu ARP este simplã si este prezentatã schematic în figura 3. Dacã o statia A doreste sã rezolve adresa IP a statiei B I_B, trimite un pachet special prin broadcast la toate statiile din retea, prin care cere statiei cu adresa IP I_B sã rãspundã cu adresa sa fizicã P_B. Toate statiile receptioneazã pachetul, dar doar statia B îsi recunoaste propria adresã IP si rãspunde statiei A, cãreia îi cunoaste adresa fizicã chiar din pachetul receptionat. Dupã ce statia A a receptionat rãspunsul va trimite pachetele statiei B folosind adresa ei fizicã. ARP permite unei statii sã afle adresa fizicã a unei alte statii conectate la aceeasi retea fizicã, furnizând doar adresa IP a statiei destinatie.

Pentru a reduce comunicatiile inutile, statiile care folosesc ARP mentin în cache cele mai recente adresele IP rezolvate si adresele fizice corespunzãtoare. Când o statie primeste un rãspuns la o cerere ARP, ea salveazã în cache adresa IP a masinii si adresa fizicã corespunzãtoare, pentru cãutãrile ulterioare. Când statia doreste sã transmitã un pachet, ea se uitã prima datã dacã are în cache adresa fizicã pentru adresa IP doritã, dacã o are o foloseste pe aceasta, iar dacã nu o gãseste trimite un pachet ARP, si asteaptã rãspunsul cu adresa fizicã.

5.2.3 Implementarea ARP

Din punct de vedere functional ARP este împãrtit în douã pãrti. O parte determinã adresele fizice prin trimiterea de pachete, iar cealaltã parte rãspunde cererilor de la alte statii.

Dându-se o adresã IP destinatie, statia A care doreste sã trimitã un pachet consultã cache-ul pentru a vedea dacã cunoaste maparea adresei IP în adresa fizicã. Dacã gãseste adresa fizicã pentru statia destinatie B, o foloseste pe aceasta în construirea cadrului pentru Ethernet, încapsuleazã informatia si o transmite statiei destinatie. Dacã nu reuseste sã mapeze adresa IP, atunci trebuie sã trimitã în broadcast o cerere ARP si asteaptã un rãspuns. Aici apar însã câteva probleme. Dacã statia destinatie nu este pornitã sau nu rãspunde cererii ARP, atunci statia A nu va primi nici un rãspuns si ar trebui sã existe un

timp de timeout după care să retransmită cererea ARP. Acest timp de timeout este necesar si pentru cazul în care, datoritã coliziunilor, s-ar pierde pachetul cu cererea sau pachetul cu rãspunsul.

La nivelul statiei care receptioneazã o cerere ARP, se executã urmãtoarele operatii: se extrage din pachetul primit adresa IP si adresa fizicã corespunzãtoare statiei A, verificã dacã acestea existã în cache-ul propriu si în caz cã nu existã le salveazã iar dacã existã se suprascrie noua pereche. Dacã adresa IP nu este egalã cu propria adresã IP acest pachet este ignorat. Dacã, însã adresa IP este cea a statiei, aceasta construieste un pachet rãspuns pentru statia A, pachet care contine si adresa fizicã a statiei B, si-l trimite statiei A. Statia A preia pachetul, extrage informatia de mapare a adreselor de care are nevoie, îsi actualizeazã cache-ul cu ea si trimite pachetul de date statiei B.

Un mesaj ARP pentru a putea fi trimis trebuie încapsulat într-un cadru fizic. Aceastã încapsulare este prezentatã în figura 4.

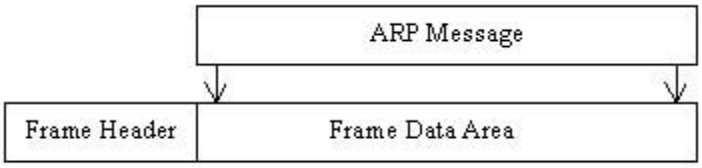


Fig. 4

Structura unui pachet ARP nu este fixã. Astfel cã acest protocol poate fi folosit pe mai multe tipuri de retele fizice si pentru maparea mai multor tipuri de adrese de nivel înalt. În figura 5 este prezentat un pachet ARP care foloseste adresele IP pe o retea Ethernet.

HARDWARE TYPE		PROTOCOL TYPE
HLEN	PLEN	OPERATION
SENDER HA (octets 0-3)		
SENDER HA (octets 4-5)		SENDER IP (octets 0-1)
SENDER IP (octets 2-3)		TARGET HA (octets 0-1)
TARGET HA (octets 2-5)		
TARGET IP (octets 0-3)		

Fig. 5

HARDWARE TYPE - tipul interfetei cu reteaua (1 pentru Ethernet);
PROTOCOL TYPE - tipul adreselor de nivel înalt pentru care se face maparea (0800h pentru adrese IP);
OPERATION

- - 1 pentru cerere ARP
- - 2 rãspuns ARP
- - 3 cerere RARP
- - 4 rãspuns RARP;

HLEN, PLEN - specificã lungimea adresei fizice si respectiv cea a adresei de nivel înalt;
SENDER HA - adresa fizicã a statiei care a initiat cererea;

SENDER IP - adresa de nivel înalt a statiei care a initiat cererea;
TARGET HA - adresa fizicã a statiei care a primit cererea;
TARGET IP - adresa de nivel înalt a statiei care a primit cererea;

5.2.4 Determinarea adresei IP

Existã situatii în care o satie, dupã boot-are nu îsi cunoaste propria adresã IP (este cazul statiilor fãrã harddisk care comunicã prin TCP/IP cu server-ul). Pentru acestea existã "Reverse Address Resolution Protocol" (RARP) prin care este posibilã obtinerea adresei IP pe baza adresei fizice a statiei. RARP functioneazã asemãnãtor cu ARP, si foloseste acelasi tip de pachete (fig. 2.5). Deosebirea fatã de ARP constã în aceea cã într-o cerere RARP se completeazã atât informatiile despre "sender" cât si cele despre "target" cu adresa fizicã a statiei care trimite cererea. O cerere RARP este primitã de toate statiile din retea dar vor rãspunde la ea doar acele statii care au fost configurate ca server-e RARP.

5.3 Subretele

Structura standard a unei adrese IP poate fi modificatã local prin folosirea bitilor pentru adresa statiei ca biti suplimentari pentru adresa de retea. Prin aceasta se crează mai multe retele, prin reducerea numãrului maxim de statii ce apartin fiecãrei retele nou create. Aceste retele nou create se numesc subretele.
Folosirea subretelelor:
- permite un management decentralizat al adresării statiilor;
- rezolvarea diferentelor hardware si a limitării distantelor.

Din punct de vedere conceptual, împãrtirea în subretele schimbã doar interpretarea unei adrese IP. Astfel, în loc de a împãrti adresa IP într-un prefix corespunzãtor retelei si un sufix pentru adresa statiei, adresa se împarte într-o portiune corespunzãtoare retelei si una corespunzãtoare statiei. Partea de retea fizicã se tratează doar local; doar gateway-ul local stie cã sunt mai multe retele fizice si rutează traficul între ele.

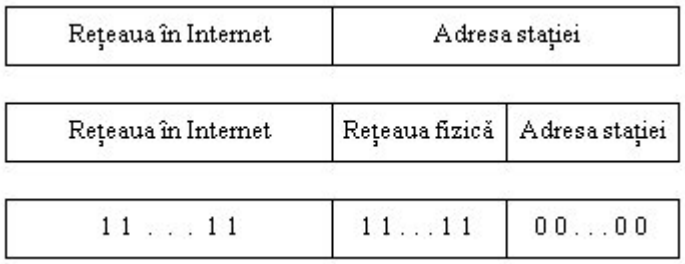


Fig. 6: Impartirea retelei in subretele

Standardul specificã cã pentru un site care foloseste subretelele trebuie specificatã câte o mascã pentru fiecare subretea. În aceastã mascã sunt setati pe 1 bitii corespunzãtori adresei retelei si sunt pe 0 pentru portiunea corespunzãtoare adresei statiei.

5.4 Rezumat

Protocolul TCP/IP foloseste adrese binare pe 32 de biti ca si identificatori universali. Aceste adrese IP sunt împãrtite în trei clase (Clasa A: 7 biti pentru retea si 24 biti pentru statie; Clasa B: 14 biti pentru retea si 16 biti pentru host si Clasa C cu 21 biti pentru retea si 8 biti pentru host).

Adresele IP identificã o conexiune si nu o statie, astfel cã o statie cu mai multe conexiuni de retea va avea mai multe adrese IP.
Adresele IP speciale sunt prezentate în figura 7.

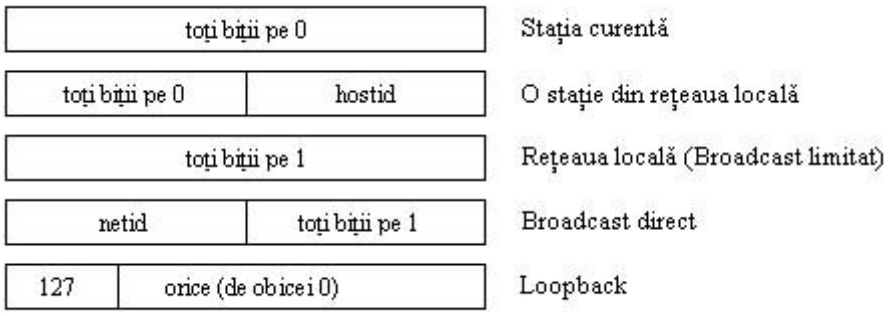


Fig. 7: Adrese IP speciale

ARP este un protocol de nivel scãzut care ascunde adresarea fizicã a retelei, permitându-ne sã alocãm adrese IP la alegerea noastrã fiecãrei statii.
RARP este un protocol care permite unei statii sã-si afle adresa IP pe baza adresei sale fizice.

Lucrarea 6

DISPOZITIVE DE INTERCONECTARE PENTRU RETELELE LOCALE

In cele ce urmeaza voi utiliza o clasificare a dispozitivelor de interconectare in trei categorii: repetoare, punti si rutere - chiar daca exista o serie de producatori care incearca sa-si impuna produsele ca pe o noua clasa de dispozitive. Aceasta tipologie prezinta avantajul de a urmari de aproape stiva de protocoale OSI, prezentand astfel actiunile specifice fiecarui nivel.

Repetorul este dispozitivul de interconectare ce functioneaza la nivel fizic. Deoarece la nivelul fizic nu exista date ci doar biti, repetorul nu este preocupat de identificarea destinatiei sau de verificarea unui cod de corectie, ci doar de semnalul electric pe care-l primeste si de regenerarea acestuia.
Principala sa functie este aceea de a extinde suprafata acoperita de o retea locala cu un cost si o latenta foarte scazute.
Sirul de biti generat de o placa de retea este clar, respectand strict nivelurile de tensiune standardizate. Cu cat sirul de biti calatoreste mai mult prin cablu, semnalul electric se deterioreaza si devine din ce in ce mai slab. Pentru a opri deteriorarea semnalului peste o limita ce l?ar face de nerecunoscut pentru destinatie, repetorul ia sirul de biti, il aduce la treptele de semnalizare standardizate si il amplifica.
Deprecierea semnalului nu apare doar cand acesta calatoreste prin mediul de cupru, dar si cand atasam prea multe dispozitive la mediul de transmisie, deoarece fiecare nou dispozitiv atasat la mediu va provoca o mica degradare a semnalului.

Exista repetoare pentru toate mediile de transmisie pe cupru - de la cablul coaxial de diferite impedante pana la cel torsadat. Cele mai des intalnite retele locale sunt totusi fara indoiala si in Romania retelele Ethernet. Din aceasta cauza ma voi referi in continuare cu precadere la acestea.

In retelele Ethernet intalnim deseori repetoare multiport numite huburi. Huburile vor transmite datele primite pe unul dintre porturi pe toate celelalte porturi. Pentru mediul torsadat acestea indeplinesc o functie suplimentara si anume asigura conectarea tuturor nodurilor la un mediu de transmisie distribuit.

Initial au existat doua tipuri de huburi: pasive si active. Huburile pasive ofera posibilitatea interconectarii la acelasi mediu de transmisie a mai multor dispozitive, fara a regenera semnalul la trecerea prin ele. Huburile active vor oferi in plus fata de primele regenerarea semnalului. Datorita scaderii extrem de rapide a preturilor si avantajelor ce le ofera aceasta regenerare de semnal huburile pasive au disparut de pe piata inca de la sfarsitul anilor '80, din aceasta cauza in continuare prin huburi vom intelege huburi active.

Una din componentele esentiale ale protocolului Ethernet este detectia coliziunilor. Ne intereseaza care este efectul unui repetor asupra coliziunilor. Inainte de a merge mai departe, ar fi bine insa sa definim doua notiuni pe care le vom mai intalni deseori pe parcursul acestei carti.

L6 Dispozitive de interconectare pentru retelele locale

Ce sunt domeniile de coliziune?

Un domeniu de coliziune reprezinta acea sectiune dintr-o retea in care se va propaga o coliziune.

Ce sunt domeniile de difuzare?

Un domeniu de difuzare (domeniu de broadcast) reprezinta acea sectiune dintr-o retea in care se va propaga un pachet de difuzare (broadcast).

Pentru un repetor nu exista notiunea de coliziune, dupa cum nu exista notiunea de pachet de date. Deci repetoarele extind atat domeniile de coliziune cat si pe cele de difuzare.

Repetoarele impart reteaua in microsegmente. Aceasta denumire nu este general acceptata, cateodata fiind folosit termenul de segment pentru cele doua sau mai multe seturi de calculatoare pe care le conecteaza un repetor.

Exista o regula foarte importanta pentru proiectarea retelelor Ethernet: regula 5-4-3.

Regula 5-4-3: Comunicatia dintre oricare doua calculatoare sau dispozitive dintr-o retea nu trebuie sa treaca prin mai mult de:

5 microsegmente,

4 repetoare consecutive,

3 microsegmente populate.

De unde vine aceasta regula?

Exista o fereastra de timp pentru transmiterea unui bit. Pentru Ethernet, ce ofera o viteza de 10 Mbps, durata transmiterii unui singur bit este de 100 de nanosecunde. Dimensiunea minima a cadrului Ethernet este de 64 de octeti. Rezulta ca timpul necesar transmiterii cadrului de dimensiune minima este de 51,2 microsecunde.

De ce ne intereseaza acest timp? Pentru ca aparitia unei coliziuni trebuie detectata inainte de expirarea acestui interval de timp. Altminteri, aparitia unei coliziuni va fi interpretata ca o coliziune la cel de-al doilea cadru si nu pentru primul.

Latenta introdusa de mediul de transmisie va fi data de viteza de propagare a semnalului electric, aceasta fiind aproximativ doua treimi din viteza luminii. Rezulta ca propagarea pe un segment de 100 de metri va dura aproximativ 0,5 microsecunde. Comparativ cu latenta introdusa de un repetor Ethernet de aproximativ 5,6 microsecunde, latenta introdusa de mediul de conectare este cu un ordin de marime mai mica, deci neglijabila.

Cel mai defavorabil caz se obtine cand sursa si destinatia se afla la distanta maxima, iar coliziunea apare langa destinatie, astfel incat coliziunea ce trebuie detectata si de sursa trebuie sa parcurga de doua ori distanta maxima. Daca vom considera acum ca intre sursa si destinatie se afla cinci repetoare, vom determina ca in cel mai defavorabil caz detectia coliziunii va fi posibila doar dupa cel putin 56 de microsecunde, asta insemnand ca un alt doilea pachet deja a fost trimis.

Ce se intampla daca nu respectam regula 5-4-3?

In primul rand se va cere retransmisia unui cadru corect, in vreme ce cel pierdut in urma coliziunii va fi considerat ca ajuns la destinatie intact. Astfel responsabilitatea integritatii datelor va fi pasata nivelului superior si anume nivelului retea. Din pacate, acest nivel nu are posibilitatea manipularii de cadre, si va determina ca intregul pachet din care face parte si cadrul eronat este incorect, cerand retransmiterea pachetului. Aceasta practica, desi va asigura integritatea datelor, introduce o latenta semnificativa.

Desi la o incalcare nu prea violenta a acestei reguli (cum ar fi folosirea a cinci repetoare in loc de patru) deprecierea performantelor este mica, daca vrem sa garantam ca

reteaua Ethernet instalata va oferi o latime de banda de 10 Mbps, atunci cinci repetoare inseamna deja un repetor in plus.

Doar pentru a sesiza flexibilitatea enorma a regulii 5-4-3 sa consideram cazul transmiterii cadrelor Ethernet de lungime maxima, adica 1500 de octeti. Pentru acest caz este necesara o fereastra de timp de 1,2 milisecunde, astfel incat performantele pentru o retea Ethernet ce ar folosi doar cadre de lungime maxima nu ar fi afectate nici in cazul folosirii a 100 de repetoare. Cu toate acestea in realitate se folosesc si cadre de lungime mica si, desi cadrele de lungime minima nu sunt folosite prea des, corectarea erorilor de catre nivelul retea poate constitui un cost prea mare in termeni de latenta. Desi nu este principala sa functie, repetorul ofera, precum orice dispozitiv de interconectare, posibilitatea legarii de grupuri de calculatoare ce difera prin mediul de transmisie folosit, adica prin nivelul fizic. Altfel spus, numeroase dintre huburile actuale ofera pe langa porturile RJ-45 si un port BNC, sau mai rar chiar un port AUI.

6.1 Puntea

Puntea sau bridge-ul este primul dispozitiv de interconectare ce poate lua decizii logice. Pentru el semnalele electrice se transforma in octeti si in date.
Puntea este dispozitivul de interconectare ce functioneaza la nivelul legatura de date. Puntile sunt folosite si la interconectarea a grupuri de calculatoare ce difera prin protocolul folosit la nivelul legatura de date sau a mediului de transmisie. Astfel, exista punti ce conecteaza retele Ethernet cu retele Token Ring, sau retele Token Ring cu retele Token Bus.
Care sunt mecanismele ce ii permit puntii sa ia decizii logice? Cele doua mecanisme ce fac din punte un dispozitiv de interconectare "inteligent" sunt: incapsularea datelor la nivel legatura de date si folosirea unei scheme de adresare pentru livrarea acestora.
Gruparea datelor nu se face la nivel de bit, ci la nivel de cadru, un cadru putand contine pana la 1500 de octeti in cazul cadrului Ethernet, sau chiar 8000 de octeti.

6.1.1 Principiile de functionare a puntilor

Puntea interconecteaza doua sau mai multe segmente de retea. In plus fata de un simplu calculator, care la nivelul legatura de date se preocupa doar de incapsularea datelor in cadre, o punte trebuie sa ia decizia spre ce segment sa trimita cadrul primit.
Va regenera puntea semnalul electric?
Da! In cazul in care pe una dintre interfete primeste un sir de biti ale caror valori nu sunt 0,85V sau -0,85V (in cazul Ethernetului), va incerca sa-si dea seama care au fost valorile initiale a acestor biti pentru a putea intelege cadrul primit. Odata obtinut un cadru valid, adica dupa corectarea bitilor ce nu mai aveau niveluri de tensiune corecta, puntea va desface antetul cadrului si va analiza informatiile legate de adresa destinatie. Dupa determinarea interfetei pe care trebuie trimis cadrul, placa de retea il va transforma in biti, trecandu-l la nivelul fizic. Placa de retea poate genera doar cateva niveluri de tensiune, astfel incat nici nu ar fi posibila trimiterea sirului de biti depreciat.

- **Principala functie a unei punti este filtrarea traficului pe baza adresei fizice.**

Sa ne aplecam un pic asupra procesului prin care putea ia decizii de comutare a unui cadru. Pentru a putea lua astfel de decizii puntile folosesc o tabela, numita tabela de comutare (bridging / switching table) in care fiecarei adrese fizice ii este asociata una dintre interfetele sale. In figura de mai jos avem o astfel de tabela.

Interfata	Adresa MAC
E0	00.48.C2.01.78.12
E0	00.00.2E.00.59.91
E1	00.00.54.91.01.4A

Tabela de comutare cu 3 intrari

De exemplu, prima intrare are urmatoarea semnificatie: destinatia 00.48.C2.01.78.12 se afla pe segmentul conectat pe interfata E0 a puntii (E0 este prescurtarea de la Ethernet 0, prima interfata Ethernet).

Care este rolul puntii in comunicatia din interiorul aceluiasi segment?
Protocolul Ethernet ofera un mediu de comunicatie distribuit, adica comunicatia dintre doua statii va accesibila nivelului legatura de date a oricarei alte statii conectate pe acelasi segment. Pentru fiecare cadru primit de o statie, nivelul legatura de date va verifica daca aceasta statie este sau nu destinatia. In cazul afirmativ cadrul va fi pasat nivelului retea, altminteri va fi ignorat.

Pentru cazul comunicatiei in interiorul aceluiasi segment, sa consideram reteaua din figura de mai jos. Presupunem ca statia A1 vrea sa transmita date statiei A2.

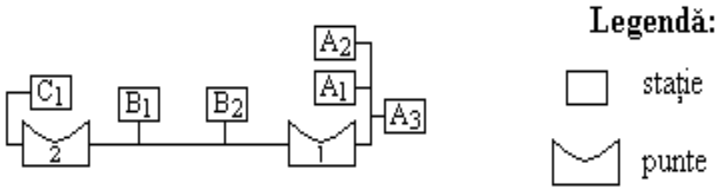


Fig. 1: Retea segmentata cu punti

Suntem intr-o retea Ethernet, asa ca primul lucru pe care-l va face statia A1 va fi ascultarea mediului. Daca mediul este liber va incepe transmisia datelor. Cadrul emis de A1 se va propaga catre toate statiile conectate pe acest segment, inclusiv catre punte. Statia A2 va trece cadrul catre nivelul retea, statia A3 il va ignora. Odata ajuns la punte cadrul este despachetat si adresa destinatie este cautata in tabela de comutare a puntii. Puntea va decide ca destinatia se afla chiar pe interfata pe care a primit cadrul. In acest caz puntea ia decizia ca acest cadru nu mai trebuie transmis, deoarece retransmiterea cadrului ar duce la o duplicare a acestuia la destinatie.

Cum va actiona puntea 1 in cazul comunicatiei intre B1 si B2?
Ambele punti (desi vor receptiona cadrele) vor lua decizia de a nu le mai retransmite.
Dar sa presupunem ca cele doua comunicatii apar simultan: atat A1 transmite catre A2, cat si B1 catre B2. Va aparea in acest caz o coliziune? Daca in loc de puntea 1 am fi folosit un repetor, cu siguranta am fi avut o coliziune. In cazul nostru, de vreme ce nici un cadru din comunicatia dintre A1 si A2 nu va ajunge pe segmentul B, si nici un cadru din comunicatia dintre B1 si B2 nu va ajunge pe segmentul A, este imposibil sa apara o coliziune.

- **Puntea izoleaza comunicatia intre statii aflate in acelasi segment la nivelul segmentului.**

Consecintele acestui fapt sunt extrem de importante. In primul rand, puntea va margini domeniile de coliziune. Totodata ea va oferi mai multa banda disponibila, deoarece comunicatia in interiorul aceluiasi segment nu va consuma din banda disponibila a intregii retele.

O alta consecinta o reprezinta minimizarea riscurilor de securitate legate de atacurile din interiorul retelei locale. Unul dintre cele mai populare atacuri este ascultarea liniei (sniffing attack), prin care pe una dintre statiile conectate la mediul distribuit se forteaza nivelul legatura de date sa trimita spre nivelurile superioare toate cadrele - inclusiv cele ce nu sunt destinate acestei statii. Cu ajutorul unor aplicatii dedicate datele sunt reasamblate si astfel va fi monitorizat tot traficul ce traverseaza segmentul de retea. Prin folosirea puntilor putem izola de restul retelei statiile ce prezinta un risc de securitate.

Care este rolul puntii in comunicatia dintre segmente?

Pentru acest caz vom considera aceeasi retea din figura precedenta si un trafic intre statia A1 si B1. Statia A1 va asculta mediul si cand acesta va fi liber va transmite un cadru. Cadrul se va propaga spre statiile A2, A3 si spre puntea 1. Statiile vor ignora cadrul, acesta nefiind adresat lor, in schimb puntea va cauta adresa destinatie in tabela sa de comutare. Va determina interfata pe care trebuie trimis cadrul si apoi va decide ca aceasta interfata este diferita de cea pe care cadrul a fost primit. Astfel incat puntea va transmite cadrul primit din segmentul A, doar pe segmentul B. Cadrul va fi receptionat atat de B1, cat si de B2, dar doar B1 il va prelucra.

Care este suprafata maxima pe care o poate ocupa o retea ce foloseste doar punti?

Suprafata maxima pe care se poate intinde o retea folosind doar punti nu face obiectul nici unei reglementari explicite. Cu toate acestea, in plus fata de avantajele prezentate mai sus, puntea aduce si o serie de dezavantaje, facand astfel ca procesul de proiectare a unei retele locale sa fie un lucru foarte delicat.

In comparatie cu repetorul, puntea inlatura limitarile impuse de regula 5?4?3, izoleaza traficul din interiorul unui segment la nivelul segmentului si ofera posibilitatea interconectarii unor segmente de retea ce folosesc protocoale de nivel legatura de date diferite. Puntile vor extinde domeniile de difuzare, desi le limiteaza pe cele de coliziune. In acelasi timp, costul unei punti este cu cel putin un ordin de marime mai mare decat cel al unui repetor.

Inlocuirea repetoarelor cu punti duce o crestere a latentei in retea cu 10-30 %, datorita timpului necesar prelucrarii informatiei de nivel legatura de date. In cazul unui trafic intens intre statii aflate in segmente diferite puntea poate duce la o gatuire a traficului. Pentru a putea functiona eficient o punte trebuie sa aiba la dispozitie o tabela de comutare ce contine cate o intrare pentru fiecare dintre statiile din acea retea locala. Cautarea in aceasta tabela este o cautare secventiala, deci extrem de ineficienta pentru o dimensiune prea mare a tabelei. Astfel dimensionarea optima a retelei ce foloseste doar punti, desi nu se supune nici unei restrictii de lungime, va fi puternic influentata de numarul de statii, precum si de tipul traficului, mai exact de latenta pe care ne-o putem asuma pentru diferite tipuri de trafic.

Cum isi construieste puntea tabela de comutare?

In exemplele anterioare am presupus ca tabela de comutare era deja construita. Aceasta tabela este pastrata bineinteles in memoria RAM a puntii, prin urmare se va pierde daca

reinitializam puntea. In plus, o punte trebuie sa fie in stare sa includa dinamic in tabela de comutare informatii despre o noua statie conectata in retea.

Sa consideram reteaua din figura de mai jos, unde puntea 1 a fost reinitializata, si statia A1 vrea sa comunice cu statia B1.

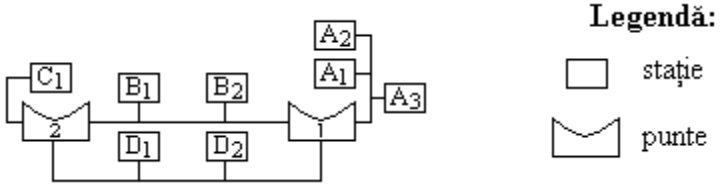


Fig. 2: Construirea tabelei pentru puntea I

Statia A1 asculta mediul, iar cand acesta este liber trimite un cadru ce are ca destinatie statia B1. Statiile A2 si A3 vor ignora cadrul. Puntea 1 va primi cadrul si va incerca sa gaseasca adresa destinatie in tabela sa de comutare. Puntea nu va reusi sa gaseasca destinatia, deoarece tabela sa de comutare era goala, astfel incat va retransmite cadrul pe toate segmentele la care este ea conectata, in afara de segmentul de pe care a fost primit cadrul. Inainte de a retransmite cadrul puntea va verifica daca adresa sursa este prezenta in tabela sa de comutare. In cazul nostru ea nu este, astfel incat puntea va crea prima intrare in tabela de comutare ce va contine adresa fizica a statiei A1 si interfata ce conecteaza segmentul A. Cadrul va ajunge atat pe segmentul D, unde statiile D1 si D2 vor determina ca nu acesta nu le este adresat lor, deci il vor ignora; cat si pe segmentul B, la statia B1, B2 si puntea 2. Puntea 2 va determina ca destinatia este in acelasi segment din care a primit cadrul si va decide sa nu?l mai retransmita, iar statia B1 va determina ca ea este destinatarul cadrului.

Chiar si comunicatia intre doua statii aflate in acelasi segment poate afecta latimea de banda din intreaga retea daca puntea nu a apucat sa-si construiasca tabela de comutare.

Dupa cadrul trimis catre statia B1, sa consideram ca statia A1 va trimite un cadru pentru A2. Cadrul va ajunge la destinatie fara ajutorul puntii, dar puntea, neidentificand destinatia in tabela sa de comutare, va retransmite cadrul atat pe segmentul B, cat si pe segmentul D.

Datorita dificultatii cautarii intr-o multime neordonata, in tabela de comutare nu se vor pastra toate adresele statiilor din reteaua locala, ci doar a celor ce au o probabilitate mare sa transmita in viitorul apropiat, mai exact a ultimilor statii ce au transmis. Pentru implementarea acestui concept, o intrare intr?o tabela de comutare va avea, pe langa adresa MAC si interfata, si o eticheta de timp. Aceasta eticheta de timp este actualizata la o noua primire a unui cadru cu aceeasi adresa sursa. Acest mecanism permite inlaturarea intrarilor invechite si deci restrangerea dimensiunii tabelei de comutare. Pretul platit pentru aceasta este consumul din latimea de banda a tuturor segmentelor din retea in cazul in care o statie nu transmite nici un cadru un interval de timp.

6.2 Comutatorul

L6 Dispozitive de interconectare pentru retelele locale

O punte multiport se numeste comutator sau switch. Fata de punti, comutatoarele in general implementeaza metode de comutare mai rapide. Uneori comutatorul este privit ca un dispozitiv de interconectare ce actioneaza atat la nivel fizic, cat si la nivel legatura de date. Aceasta nu se datoreaza unei latente mai mici sau unui cost mai scazut comparativ cu o punte, ci datorita faptului ca in retelele Ethernet ce folosesc mediul torsadat comutatorul preia functia principala a hubului, si anume aceea de a asigura conectarea tuturor nodurilor la un mediu de transmisie.

Exista doua paradigme in retelele de calculatoare: arhitecturi bazate pe magistrala si indirect pe difuzare si arhitecturi bazate pe comutare. Optarea pentru una dintre cele doua paradigme se traduce in decizia de a folosi un comutator sau un hub.
O retea bazata pe huburi are un cost mai scazut si o latenta mai mica. Principalul avantaj al inlocuirii huburilor cu comutatoare nu il reprezinta inlaturarea restrictiilor impuse de regula 5-4-3, ci reducerea numarului de utilizatori ce partajeaza aceeasi latime de banda. Comutatoarele vor oferi protectie impotriva atacurilor prin ascultare a liniei.
"Razboiul" hub versus comutator opune costul si latenta mai scazute, pe de o parte, cu cerintele crescande de latime de banda disponibila si de securitate, pe de alta parte.
Am lamurit care sunt diferentele dintre comutatoare si repetoare, *dar care sunt diferentele intre punti si comutatoare?* Numarul de interfete sau porturi este fara indoiala cea mai usor de observat diferenta. Diferenta cea mai importanta consta totusi in modul de comutare a pachetelor: comutatoarele folosind comutarea directa. In acelasi timp cerintele de latenta pentru o punte cu doua interfete sunt mult mai relaxate decat pentru un comutator. Din aceasta cauza puntile, in general, comuta pachete folosind componente software, in vreme ce comutatoarele vor lua toate deciziile la nivel hardware.
Dupa cum am vazut, puntea reface semnalul la nivel de bit, pentru a obtine un cadru, apoi despacheteaza cadrul, foloseste informatiile din campul adresa destinatie pentru a filtra sau nu cadrul, iar adresa sursa va fi folosita pentru construirea tabelei de comutare. Dar una dintre functiile nivelului legatura de date este acela de a oferi mecanisme de corectie a datelor la nivel de cadru. *Ofera comutatoarele astfel de mecanisme?*

Octeti	7	1	6	6	2	<1500	4
	Preambul	Delimitator start cadru	Adresa destinatie	Adresa sursa	Lungime camp date	Date	Suma de control

Fig. 3: Structura cadrului Ethernet

In figura de mai sus este prezentata structura cadrului Ethernet. Este important de remarcat ca informatiile de detectie sau corectie a erorilor se afla in finalul cadrului. Asta inseamna ca, pentru a putea detecta erorile dintr-un cadru, trebuie mai intai asteptata receptionarea integrala a acestuia. Problema care apare in acest caz este ca latenta pentru un cadru de dimensiune maxima pentru Ethernet va fi de 1,2 milisecunde. Aparent, mutarea campului de control in antetul cadrului ar rezolva problema. Practic, acest lucru este imposibil, deoarece suma de control nu este un sir continuu de biti aflati dupa zona bitilor de date, ci bitii ce compun cei 2 sau 4 octeti ai sumei de control sunt intercalati cu bitii de date.

Care sunt tipurile de comutare folosite de un comutator?

L6 Dispozitive de interconectare pentru retelele locale

Exista doua metode de comutare a pachetelor: comutare directa (cut through) si comutare dupa stocare (store and forward).

Metoda de comutare dupa stocare se bazeaza pe receptionarea intregului cadru inainte de a incepe retransmisia acestuia. Latenta acestei metode creste odata cu dimensiunea campului de date. Cu toate acestea, performantele metodei de comutare dupa stocare pot fi superioare celor oferite de comutarea directa, mai ales in cazul linilor expuse unor interferente puternice. Mecanismele de detectie a erorilor pe care le ofera aceasta metoda de comutare permite asigurarea unei conexiuni sigure la nivelul legatura de date. Aparent, metoda de comutare dupa stocare ridica si problema asigurarii memoriei pentru stocarea cadrelor. Sa luam exemplul unui comutator cu 24 de porturi. Acesta va trebui sa poata gestiona 12 comunicatii simultane, care in cel mai defavorabil caz posibil vor transfera cadre de lungime maxima. Am ajuns astfel la o dimensionare a memoriei RAM pentru stocarea cadrelor de aproape 18 ko. Putem trage concluzia ca dimensionarea memoriei RAM folosite pentru stocarea cadrelor a incetat sa mai fie o problema de actualitate.

Comutarea directa presupune ca puntea sa inceapa transmiterea cadrului pe portul destinatie imediat ce adresa destinatie a fost trecuta prin tabela de comutare si interfata de plecare a fost determinata. Cel mai adesea se intampla ca transmisia cadrului sa inceapa inainte de receptionarea integrala a cadrului. Astfel comutatorul va primi pe una dintre interfete octeti ce compun cadrul, transmitand in acelasi timp pe portul destinatie octeti din acelasi cadru primiti mai devreme.

Pentru comutarea directa nu este necesara nici macar receptionarea integrala a antetului cadrului, adresa destinatie fiind suficienta. Aceasta metoda se numeste comutare directa rapida (fast forward) si ofera o latenta de aproximativ 21 de microsecunde. Datorita faptului ca retransmisia cadrului incepe imediat dupa citirea adresei destinatie, cadrele eronate vor fi transmise cu erori. Desi aceste cadre sunt respinse la nivelul legatura de date al destinatiei (de catre placa de retea), traficul generat de retransmisia lor poate, in cazul unui mediu de transmisie cu multe erori, sa duca la o depreciere severa a performantelor retelei.

Al doilea tip de comutare directa este comutarea fara fragmente (fragment free). Pentru aceasta metoda de comutare vor fi filtrate fragmentele de cadre rezultate in urma unei coliziuni. Intr-o retea ce respecta specificatiile standardului Ethernet dimensiunea fragmentelor de coliziuni nu poate depasi 64 de octeti. Pentru comutarea fara fragmente, comutatorul va determina ca sirul de octeti receptionati nu fac parte dintr-un fragment de coliziune si abia apoi va incepe retransmisia pe portul destinatie. Latenta in acest caz este de minim 51,2 microsecunde, ceea ce reprezinta timpul necesar receptionarii a 64 de octeti.

Care sunt tipurile de comutare folosite de o punte?

In general puntile implementeaza doar comutarea dupa stocare, aceasta fiind una din principalele deosebiri fata de comutatoare care nu vor mai fi preocupate de detectia erorilor, ci de filtrarea pachetelor si de asigurarea unei latente cat mai scazute.

Care este rolul comutatoarelor in implementarea conexiunilor Ethernet half-duplex?

Comunicatia semi-duplex (half-duplex) permite doar unui singur nod sa transmita date. In Ethernet aceasta este controlata cu ajutorul coliziunilor. Daca doua sau mai multe statii incearca sa comunice simultan rezultatul va fi o coliziune.

L6 Dispozitive de interconectare pentru retelele locale

Pe interfetele unui comutator putem conecta o statie sau un segment intreg. Cu toate acestea, retelele comutate sunt raspunsul pentru cerintele crescande de securitate si de latime de banda pentru fiecare nod. Retelele comutate vor folosi cate un port pentru fiecare statie, reducand dimensiunea domeniilor de coliziune la doar doua noduri (unul fiind placa de retea din respectiva statie, iar cel de-al doilea portul din comutator ce o conecteaza pe aceasta).

Altfel spus, comutatoarele ofera suportul pentru implementarea retelelor comutate, retele in care domeniile de coliziune nu depasesc doua noduri.

Care este rolul comutatoarelor in implementarea conexiunilor Ethernet full-duplex?

Ethernetul full-duplex permite trimiterea si receptionarea simultana.

Ce tipuri de arhitecturi pot fi implementate cu ajutorul comutatoarelor?

STP

O bucla de nivel legatura de date apare intr-o retea cand intre doua dispozitive ale acesteia exista doua sau mai multe legaturi active, fiecare conexiune folosind doar dispozitive de interconectare ce pot analiza cel mult informatii de nivel legatura de date.

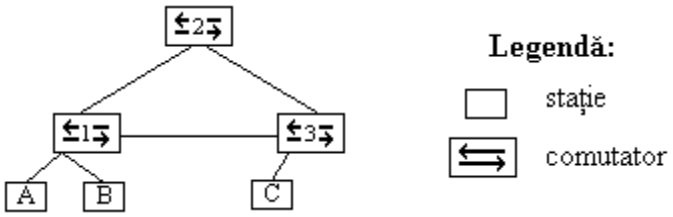


Fig. 4: Retea in care s-a creat o bucla

Care este efectul aparitiei buclelor de nivel legatura de date?

Aparitia buclelor de nivel legatura de date este corelata cu faptul ca puntile si comutatoarele nu filtreaza pachetele de difuzare si duc la o depreciere semnificativa a performantelor retelei prin determinarea unor avalanse de difuzari (broadcast storm).

Sa consideram reteaua din figura de mai sus. Presupunem ca statia A1 trimite un cadru de difuzare. Comutatorul 1 nu va gasi adresa destinatie in tabela sa de comutare, astfel incat va transmite cadrul pe celelalte segmente: segmentul ce contine statia B, segmentul dintre comutatoarele 1 si 2, si segmentul dintre comutatoarele 1 si 3. Statia B va examina cadrul, va decide ca ii este adresat si il va trece spre nivelul legatura de date. Comutatorul 2 va lua decizia de a transmite cadrul pe toate interfetele sale, cu exceptia celei de pe care a primit cadrul. Am ajuns ca sa avem in retea doua cadre destinate statiei FF.FF.FF.FF.FF.FF, adica doua cadre de difuzare. Indiferent de ordinea in care acestea ajung la comutatorul 3, acesta va determina ca nu cunoaste adresa destinatie si le va retransmite catre statia C, dar si catre celelalte comutatoare.

Avalansa de difuzari consuma din banda utila a retelei, ducand la o micsorare a bandei efective disponibile. O avalansa de difuzari se va opri doar in cazul intreruperii buclei.

Cum putem preveni aparitia avalanselor de difuzari?

Solutia triviala ar fi sa instruim puntile si comutatoarele sa nu retransmita cadrele de difuzare. Din pacate acest lucru nu este posibil, deoarece o serie de protocoale folosesc cadre de difuzare pentru a functiona corect, unul dintre acestea fiind chiar ARP - Address

Resolution Protocol. Altfel spus, filtrarea cadrelor de difuzare de catre punti ar presupune rescrierea protocoalelor fundamentale ce asigura suportul de comunicatie.
Solutia valida presupune identificarea buclelor si intreruperea lor. Protocolul ce realizeaza aceasta se numeste STP - Spanning Tree Protocol, si presupune construirea unui arbore de acoperire pe graful determinat de dispozitivele de interconectare si de conexiunile dintre acestea.

Cum functioneaza STP?
Functionarea acestui protocol se bazeaza pe crearea topologiei retelei folosind niste cadre speciale numite cadre BPDU (Bridge Protocol Data Unit). Aceste cadre speciale sunt folosite intens la initializarea comutatoarelor; ulterior, la fiecare doua secunde vor fi schimbate cadre BDPU, pentru a verifica daca nu au aparut modificari. Totodata sunt definite cinci stari in care se poate afla o interfata a comutatorului: starea blocat, de ascultare, de invatare, de comutare de cadre si nefunctional (blocking, listening, learning, forwarding, disabled). In starea blocat nu se accepta decat cadre BPDU, in cea de ascultare se primesc si cadre, dar acestea nu sunt retransmise. In starea de invatare, in plus fata de starea de ascultare, este inspectata adresa sursa a cadrelor primite, permitand astfel construirea tabelei de comutare. In starea de comutare cadrele primite sunt retransmise, iar tabela de comutare este actualizata. In starea nefunctional nu se vor accepta nici cadre BPDU.

Pentru construirea arborelui de acoperire sunt necesare aproximativ 30 de secunde, timp in care toate porturile comutatoarelor sunt in starea blocat. Exista trei pasi ce trebuie urmati pentru construirea arborelui de acoperire: mai intai trebuie aleasa radacina arborelui (root bridge), apoi trebuie alese porturile radacina, pentru ca in final sa fie determinate porturile active.

Prioritatea puntii este o valoare numerica pastrata in memoria nevolatila a fiecarei punti. Pe baza compararii prioritatilor tuturor puntilor din retea se va determina puntea cu prioritatea cea mai scazuta, aceasta devenind radacina arborelui de acoperire.
Prioritatea puntii are o valoare implicita atribuita de producator, valoare ce poate fi modificata ulterior. In cazul folosirii mai multor echipamente produse de aceeasi firma, se intampla adesea sa existe mai multe punti ce vor avea aceeasi prioritate.

Cum vom putea decide care dintre doua sau mai multe punti cu aceeasi prioritate sa devina radacina arborelui?
Pe baza adresei fizice. Puntea cu cea mai mica adresa fizica va deveni radacina arborelui de acoperire.

Pasul al doilea presupune identificarea cailor redundante dintre fiecare punte si puntea radacina, apoi selectarea unei sigure cai intre respectiva punte si radacina si, in final, dezactivarea celorlalte.

Pentru evaluarea unei cai vom determina costul caii, care va fi definit ca suma a costurilor porturilor prin care trece calea. Costul unui port este definit pe latimea de banda pe care o ofera portul, sau uneori chiar pe mediul de transmisie folosit pentru conectarea la port. De exemplu, pentru comutatoarele CISCO costul portului este determinat prin impartirea lui 1000 la latimea de banda pe care o ofera portul, astfel incat un port Ethernet va avea costul 100.

Pentru alegerea porturilor radacina vor avea prioritate porturile conectate direct la radacina arborelui de acoperire. In cazul in care nu exista nici un port cu o conexiune directa spre puntea radacina, sau cand avem mai mult de un singur port cu conexiune directa spre radacina, va fi ales portul ce are cel mai scazut cost al caii spre radacina.

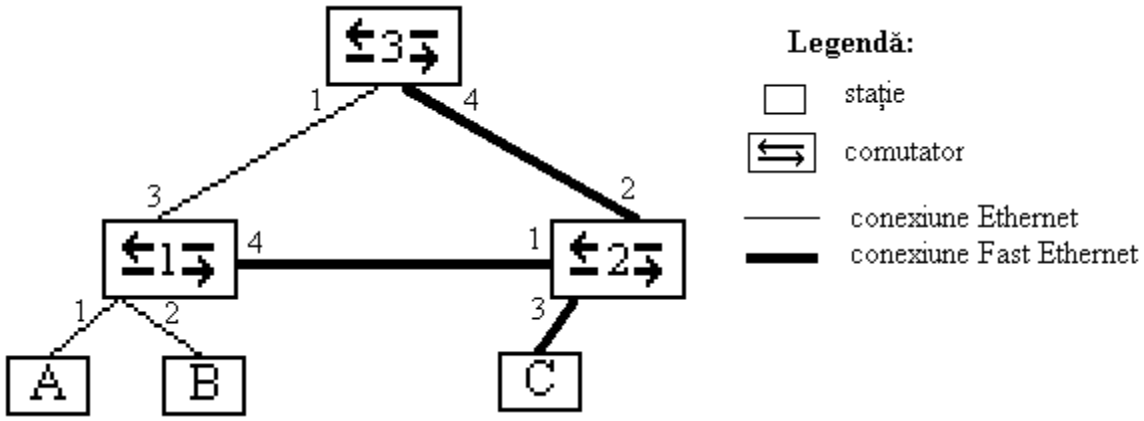


Fig. 5: Construirea arborelui de acoperire

Fie reteaua din figura de mai sus. Vom urmari pentru aceasta retea etapele construirii arborelui de acoperire.

Prima intrebare pe care trebuie sa ne-o punem este: care este prioritatea fiecarui comutator? Sa consideram ca toate cele trei comutatoare sunt produse de acelasi fabricant si in plus sunt abia scoase din cutie. Asta inseamna ca toate comutatoarele vor avea aceeasi prioritate. In acest caz va trebui sa aflam adresele fizice.

comutatorul 1	00.C2.45.26.57.A1
comutatorul 2	00.C2.45.2E.08.EF
comutatorul 3	00.C2.45.A2.11.49

Din analiza tabelului de mai sus, rezulta ca radacina arborelui de acoperire va fi comutatorul 1. In continuare vom determina pentru restul comutatoarelor costurile porturilor ce ofera cai spre comutatorul radacina. Pentru comutatorul 2 costul portului 1 va fi 10 (=1000/100), iar pentru portul 2 va fi 110 (10 + costul portului 1 din comutatorul 3). Pentru comutatorul 3, portul 1 va avea costul 100, iar portul 4 costul 20.
Pentru comutatorul 2 portul radacina va fi portul 1, astfel incat portul 1 trece in starea de comutare, in vreme ce portul 2 va ramane in starea de blocat. Pentru comutatorul 3 portul radacina va fi portul 1, deoarece, chiar daca are un cost mai mare decat portul 4, este direct conectat la radacina, astfel incat portul 1 va trece in starea de comutare.

6.3 Ruterul

Ruterul este dispozitivul de interconectare ce are rolul de a determina calea ce trebuie urmata de un pachet pentru a ajunge la destinatie, de a interconecta si a schimba pachete intre retele diferite. Ruterul este un dispozitiv de interconectare ce poate fi intalnit mai ales la nivelWAN, dar si la nivelul retelei locale, una din functiile sale principale fiind si acea de a oferii posibilitatea conectarii LAN-urilor la WAN.

L6 Dispozitive de interconectare pentru retelele locale

Procesul de rutare sau de determinare a cai optime se bazeaza pe construirea si mentinerea unei tabele de rutare. O intrare intr-o tabela de rutare se numeste ruta si este compusa din minim 3 elemente: adresa de retea, masca de retea, adresa urmatorului ruter si/sau interfata de plecare.

Ce se intampla cu un pachet ajuns la un ruter?

Antetul de nivel legatura de date este despachetat. Acesta va contine doar adresa logica a destinatiei si nu si masca de retea. Ruterul va verifica mai intai daca adresa destinatie nu este cumva una dintre adresele sale. Daca este atunci cadrul va fi trecut la nivelul superior, daca nu ruterul va verifica daca adresa destinatie nu este in accesi retea cu interfata de pe care a primit pachetul. Daca este atunci ruterul nu va mai face nimic cu acest pachet si va trece la urmatorul. Daca este atunci va abandona prelucrarile asupra respectivului pachet si va lua urmatorul pachet. In cazul in care destinatia nu este nici el si nici nu se afla pe accesi interfata de unde a primit pachetul, atunci va incepe procesarea tabelei de rutare. Va extrage prima ruta din tabela si va aplica masca de retea adresei destinatie continuta in antetul pachetului. Rezultatul il va compara cu adresa de retea a respectivei rute. Daca cele doua coincid pachetul va fi trimis pe interfata specificata de ruta. Daca nu este extrasa o noua ruta din tabela. Procesul se repeta pana la ultima ruta din tabela sau pana la gasirea primei potriviri. Daca pachetul nu corespunde nici ultimei rute atunci acesta este abandonat si se trece la pachetul urmator. Inainte de a trimite pachetul sau il abandona tabela ARP a interfetei pe care a sosit pachetul va fi actualizata folosindu-se adresa MAC si cea IP a sursei.

Care este efectul ruterelor asupra domeniilor de difuzare si a domeniilor de coliziune?

La nivelul legatura de date puntile detectau coliziunile si nu le transmiteau mai departe, dar cadrele de Detectia coliziunilor este una din principalele functii ale nivelului legatura de date, iar ruterul insusindu-si toate functiile primelor niveluri va face atat regenerarea semnalului cat si detectia coliziunilor. In plus ruterele, spre deosebire de punti au acces si la informatiile de nivel retea aceasta permitandu-le controlul difuzarilor si a pachetelor de multicast. In mod implicit ruterele nu transfera pachetele de difuzare sau de multicast. Concluzionand putem spune ca ruterele marginesc atat domeniile de coliziune, cat si pe cele de difuzare.