

## Capitolul 5. NIVELUL REȚEA

### 5.1. Serviciile nivelului rețea

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

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

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

Oricare ar fi politica de dirijare, performanțele rețelei pot fi afectate drastic în cazul supraincercării. De aceea, o altă funcție a nivelului rețea este evitarea congestiei rețelei, sau a supraincercării sale.

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

- identificatorul autorității și formatului,
- identificatorul domeniului,
- adresa în domeniu.

Funcțiile nivelului rețea sunt deci următoarele:

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

Serviciile oferite nivelului transport sint evidentiata prin enumerarea primitivelor de serviciu, care pot fi grupate in doua categorii: orientate pe conexiuni si neorientate pe conexiuni.

Primitivele orientate pe conexiuni permit:

- stabilirea sau desfiintarea unei conexiuni;
- transferul normal sau expeditiaal datelor;
- confirmarea datelor;
- resetarea unei conexiuni.

Utilizarea unei conexiuni permite pastrarea ordinii pachetelor comunicate intre sursa si destinatar. Ordinea poate fi incalcata doar prin transmiterea unor pachete expeditiv, care sint livrate cu prioritate fata de celelalte pachete. Receptia pachetelor poate fi confirmata, dar deoarece confirmarile nu au numere de secventa, ele nu au acelasi rol ca in cazul legaturii de date: ele dau informatii despre numarul de pachete ajunse la destinatar, dar nu si despre pachetele pierdute. Transferul expeditiv si confirmarea datelor sint negociabile la stabilirea conexiunii de retea, ca de altfel si calitatea serviciului, exprimata prin intirzierea transmisiei, rata de erori, costul si securitatea, precum si alte marimi caracteristice.

Stabilirea si resetarea conexiunilor sint servicii cu confirmare, cuprinzind toate cele patru categorii de primitive (cerere, indicare, raspuns, confirmare). Celelalte includ doar primele doua categorii.

Primitivele neorientate pe conexiuni permit:

- transferul individual al pachetelor;
- obtinerea unor informatii despre transferul pachetelor catre o anumita destinatie (de exemplu procentajul de pachete livrate);
- transmiterea unor rapoarte ale retelei catre nivelul transport, la producerea unor incidente.

O lista a acestor servicii si a parametrilor caracteristici, asa cum apar in standardul ISO 834, este prezentata in continuare, avindu-se in vedere impartirea lor pe diferite clase.

#### Servicii orientate pe conexiuni

-----  
Stabilirea unei conexiuni (N\_CONNECT):

N-CONNECT.request (dest,sursa,conf,exp,calitate,date)  
N-CONNECT.indication (dest,sursa,conf,exp,calitate,date)  
N-CONNECT.response (rasp,conf,exp,calitate,date)  
N-CONNECT.confirmation (rasp,conf,exp,calitate,date)

Rejectarea unei conexiuni:

N-DISCONNECT.request (orig,motiv,date,rasp)  
N-DISCONNECT.indication (orig,motiv,date,rasp)

Transmiterea datelor:

N-DATA.request (date)  
N-DATA.indication (date)  
N-EXPEDITED-DATA.request (date)  
N-EXPEDITED-DATA.indication (date)

Confirmarea receptiei datelor:

N-DATA-ACKNOWLEDGE.request ()  
. N-DATA-ACKNOWLEDGE.indication ()

Raportarea unor erori irecuperabile:

N-RESET.request (orig,motiv)  
N-RESET.indication (orig,motiv)  
N-RESET.response ()  
N-RESET.confirmation ()

Servicii neorientate pe conexiuni

-----  
Transmiterea datelor:

N-UNIT-DATA.request (sursa,dest,calitate,date)  
N-UNIT-DATA.indication (sursa,dest,calitate,date)

Determinarea caracteristicilor livrarii datelor:

N-FACILITY.request (calitate)  
N-FACILITY.indication (dest,calitate,motiv)

Raportarea unor erori:

N-REPORT.indication (dest,calitate,motiv)

S-au folosit urmatoarele notatii: sursa, dest, orig, rasp: adresele punctelor de serviciu de retea ale entitatilor sursa, destinatie, initiator si respondent, relative la actiunea specificata; conf: se doreste confirmarea datelor; exp: se solicita facilitatea de date expeditive; motiv: cauza evenimentului.

Dintre parametrii mentionati, conf, exp si calitate se negociaza. Daca valorile din primitiva .request, nu sint acceptabile pentru retea, aceasta le modifica inainte de a-i transmite entitatii receptoare prin .indication. La rindul sau, entitatea receptoare transmite valorile acceptabile prin .response, aceste valori fiind livrate emitatorului cererii prin .confirm. Calitatea serviciului este prezentata de emitatorul cererii, sub forma a doua liste de valori: una a performantelor dorite,alta a celor minime acceptabile.

## 5.2. Organizarea interna a nivelului retea

Se utilizeaza doua tehnici diferite,numite circuit virtual, prin analogie cu circuitele fizice ale retelelor telefonice, respectiv datagrama, prin analogie cu telegramele.

Prima presupune transmiterea unui pachet initial de stabilire a circuitului, care este dirijat corespunzator intre nodul sursa si

nodul destinatar. Aceeasi ruta este folosita de toate celelalte pachete transmise pe acelasi circuit virtual. In acest scop, fiecare pachet contine in antet numarul circuitului logic, iar fiecare comutator pastreaza un tabel cu toate circuitele virtuale care il traverseaza. La receptia unui pachet, pe baza numarului circuitului virtual se determina o intrare a tabelului, in care este specificata legatura pe care pachetul va fi transmis.

La stabilirea unui circuit virtual, nodul sursa alege un numar de circuit disponibil in acel nod. El poate insa coincide cu numarul ales de nodul vecin pentru un alt circuit virtual. Pentru a evita ambiguitatile, un circuit virtual este renumerotat in fiecare nod prin care trece, corespondenta intre vechea si noua numerotare fiind pastrata in tabelul circuitelor virtuale, (vezi figura 5.2). Renumerotarea implica modificarea numarului circuitului virtual din antet, in fiecare nod traversat de pachet.

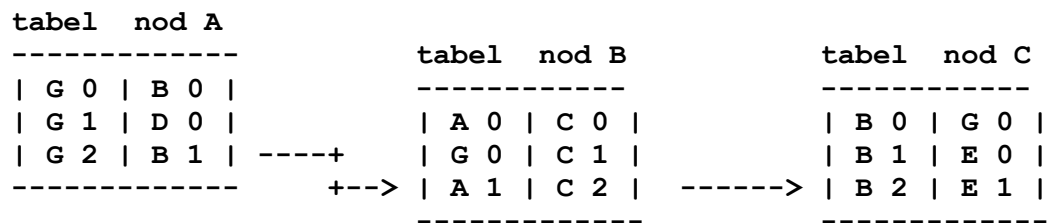


Figura 5.2.

(G semnifica gazda aplicatiei; fiecare intrare in tabele cuprinde nodul de intrare si numarul circuitului, nodul de iesire si numarul circuitului)

In cazul datagramelor, fiecare pachet este dirijat independent de predecesoarele sale. Pachetul trebuie sa contina adresa completa a destinatarului (care ocupa mai mult spatiu decit numarul circuitului virtual). Fiecare comutator are un tabel de dirijare, indicind legatura pe care trebuie transmis pachetul in functie de adresa destinatarului. Aceste tabele sint necesare si in cazul circuitelor virtuale, pentru a determina ruta pachetelor de stabilire a circuitelor. La receptia unui pachet, comutatorul inspecteaza adresa destinatarului, determina intrarea corespunzatoare din tabela de dirijare si de aici legatura pe care trebuie transmis in continuare pachetul.

### 5.3. Algoritmi de dirijare.

Asa cum s-a aratat, principala functie a unei retele cu comutare de pachete este preluarea pachetelor de la nodurile surse si livrarea lor nodurilor destinate. In acest scop, se alege o cale de transmitere a fiecarui pachet prin retea. Uzual, exista mai multe cai posibile, selectarea uneia din ele urmarind satisfacerea unor cerinte (uneori contradictorii) ale utilizatorilor si ale administratorilor retelei:

- transferul sa se faca corect si cu operativitate;
- sa nu existe utilizatori defavorizati;
- nodurile si legaturile sa fie folosite eficient.

Se cunosc mai multi algoritmi de dirijare, o clasificare, chiar sumara a lor fiind deosebit de utila. In functie de adaptarea la conditiile de trafic, dirijarea poate fi statica (atunci cind continutul tabelor de dirijare este fix), sau adaptiva (atunci cind continutul tabelor de dirijare se modifica in functie de traficul curent sau de topologia retelei).

Dupa locul unde se realizeaza calculele, dirijarea poate fi centralizata (un algoritme global utilizeaza informatii despre intreaga retea pentru a lua decizii optime de dirijare), izolata (care utilizeaza in fiecare nod informatii disponibile local) sau distribuita (care utilizeaza o combinatie de informatii locale si globale).

Algoritmii de dirijare pot fi clasificati si dupa obiectivele urmarite. O categorie o reprezinta algoritmi care asigura transmiterea pe calea cea mai scurta, pentru fiecare pereche sursa-destinatie. Algoritmii din a doua categorie minimizeaza intirziera medie globala de transmitere a pachetelor (considerind toate perechile sursa-destinatie din retea). Multe retele operationale includ algoritmi din prima categorie. Cei din a doua categorie conduc la o dirijare bifurcata, mai greu de gestionat practic. De aceea, ei sint folositi cu predilectie in proiectarea topologica a retelelor de calculatoare si mai putin ca metode efective de dirijare.

### 5.3.1. Calea cea mai scurta

Modelul topologic al unei retele este un graf in care nodurile corespund comutatoarelor de pachete, iar muchiile corespund liniilor de comunicatie. Asociind fiecarei muchii o lungime, se poate calcula calea cea mai scurta intre oricare doua noduri, deci cea mai indicata pentru dirijarea pachetelor intre nodurile respective (algoritmul lui Dijkstra).

Lungimea poate avea diverse semnificatii. Daca toate liniile au lungimea unu, gasim caile cu numar minime de noduri intermediare. Lungimea poate fi distanta geografica intre noduri, costul comunicatiei, intirziera medie masurata etc.

Algoritmul lui Dijkstra gaseste caile cele mai scurte de la o sursa la toate celelalte noduri. El trebuie sa dispuna de informatii topologice generale asupra retelei: listele nodurilor si legaturilor, costurile asociate legaturilor. Prin natura sa el este centralizat. Cu toate acestea, el are o varianta descentralizata (folosita in reseaua ARPA), in care fiecare nod pastreaza propria sa baza de date despre topologia retelei si

calculeaza singur drumurile cele mai scurte la celelalte noduri.

Algoritmul este iterati si calculeaza la fiecare iteratie ă cea mai scurta cale de la sursa la un nod al rețelei. Fie:

nnod - multimea nodurilor rețelei;  
sursa - nodul sursa;  
l[i][j]- costul legaturii (i,j), avind valorile  
0 daca i 1/2 j;  
lungmax daca i si j nu sint adiacente;  
o valoare intre 0 si lungmax in celelalte cazuri;  
D[i] - costul minime al legaturii de la sursa la i;  
S - multimea nodurilor deja selectate;  
A - tabloul de dirijare: V[i] este vecinul prin care se  
transmit date de la nodul curent la nodul i.

```
void Dijkstra (int sursa)
{ int i, j, k;
  for (i=1; i = nnod; i++)
  { S[i] = 0; / nodurile sint neselectate
    D[i] = l[sursa][i]; / distantele minime de la
                      / sursa

    if (D[i] < lungmax)
      V[i] = i; / initializeaza vecinii
    else
      V[i] = 0;
  }
  S[sursa] = 1; / selecteaza nodul sursa
  D[sursa] = 0;
  for (i=1; i = nnod; i++){
    gaseste nodul k neselectat cu D[k] minim;
    S[k] = 1;
    for (j=1; j = nnod; j++) / recalculeaza distantele
      if ((S[j] == 0) && (D[k] + l[k][j] < D[j]))
        { D[j] = D[k] + l[k][j];
          V[j] = V[k]; / modifica tabela de
        } / dirijare
  }
}
```

O varianta a drumului minime este dirijarea multicaei, in care fiecărei perechi de noduri ii corespund mai multe cai cele mai scurte, alegerea uneia facindu-se aleator cu o anumita probabilitate. Un exemplu de tabel de dirijare este tabelul 5.1.

Tabel 5.1.

destinatie	prima varianta leg. probab.	a doua varianta leg. probaâ.	a treia varianta leg. probab
a	a 0.63	f 0.27	c 0.10
c	c 0.7	e 0.2	a 0.10

d	c	0.6	e	0.2	f	0.10
e	e	0.65	a	0.25	c	0.10

---

La primirea unui pachet destinat lui d, nodul curent genereaza un numar aleator intre 0 si 1. Daca numarul este sub 0.6 alege legatura spre c, daca este intre 0.6 si 0.8 alege legatura spre e, altfel alege legatura spre f.

Daca variantele au noduri disjuncte, fiabilitatea rețelei se imbunatateste: la defectarea unor noduri sau legaturi, pot fi utilizate variantele care nu contin nodurile sau legaturile respective. Calculul variantelor in aceasta situatie este simplu: se gaseste drumul minim, se elimina nodurile si legaturile corespunzatoare acestei variante si se trece la urmatoarea varianta.

### 5.3.2. Dirijarea centralizata

Varianta centralizata a algoritmului drumurilor minime (solutie propusa de Floyd) utilizeaza un tablou A al distantelor minime,  $A[i][j]$  fiind distanta minima de la nodul i la nodul j. Initial,  $A[i][j] = l[i][j]$  pentru orice i si j.

Calculul drumurilor minime se face iterativ. La iteratia k,  $A[i][j]$  va avea ca valoare cea mai buna distanta intre i si j, pe cai care nu contin noduri numerotate peste k (exceptind i si j). Se utilizeaza relatia:

$$A[i][j] \text{ /la pas } k/ = \min (A[i][j] , A[i][k] + A[k][j]) \text{ /pas } k-1/$$

Deoarece  $A[i][k] \text{ /la pas } k = A[i][k] \text{ /la pas } k-1/$   
si  $A[k][j] \text{ /la pas } k = A[k][j] \text{ /la pas } k-1/$

nici o intrare avind unul din indici egal cu k nu se modifica la iteratia k. Drept urmare, calculul se poate realiza cu o singura copie a tabloului A.

```
void Floyd()
{ int i,j,k;
  for (i=1; i = nnod; i++)
    for (j=1; j = nnod; j++)
      { A[i][j] = l[i][j];
        if (A[i][j] < lungmax)
          V[i][j] = j;
      }
  for (k=1; k = nnod; k++)
    for (i=1; i = nnod; i++)
      for (j=1; j = nnod; j++)
        if (A[i][j] > A[i][k] + A[k][j])
          { A[i][j] = A[i][k] + A[k][j];
            V[i][j] = V[i][k];
```

```
    }  
}
```

Tabloul A contine, de data aceasta, tabelele de dirijare ale tuturor nodurilor:  $V[i][j]$  este nodul vecin lui  $i$ , pe calea cea mai scurta de la  $i$  la  $j$ . Linia  $i$  a tabloului este tabelul de dirijare al nodului  $i$ , iar linia  $i$  a tabloului  $\bar{A}$  este tabelul costurilor minime corespunzatoare.

Algoritmul poate fi folosit pentru adaptarea dirijarii la modificarile de trafic sau de topologie (caderea unor legaturi sau noduri). Un centru de control al dirijarii primeste de la comutatoarele de pachete rapoarte periodice despre starea locala, calculeaza noile tabele de dirijare pentru fiecare comutator si le transmite acestora.

Deficientele acestei metode sint determinate de:

- vulnerabilitatea retelei, dependenta de functionarea centrului de control (se recurge la dublarea lui);
- supraincarcarea traficului prin transmiterea rapoarelor si a tabelelor de dirijare;
- utilizarea in noduri, in anumite perioade, a unor tabele necorelate, datorita receptiei la momente de timp distincte a noilor tabele.

### 5.3.3. Dirijarea izolata

Algoritmul "cartofului fierbinte" (hot potato). Pachetul receptionat de nod este plasat in coada cea mai scurta.

O varianta ia in considerare lungimea cozilor anumitor linii, selectate conforme cailor celor mai scurte.

Dirijarea delta este o combinatie a politicilor izolata si centralizata. Comutatoarele trimit rapoarte unui centru de control care calculeaza cele mai bune  $\delta$  rute. Caile sint echivalente daca lungimile lor difera intre ele cu o valoare mai mica decit un  $\delta$  specificat.

In algoritmul inundarii, pachetul este transmis pe fiecare legatura, cu exceptia celei de origine. Copiile sint distruse dupa traversarea unui anumit numar de noduri. Desi nepractic, algoritmul este folosit in aplicatii militare (datorita robustetii sale) sau in comparatii de performanta cu alte tehnici (deoarece are un timp de intirziere minim).

### 5.3.4. Dirijarea distribuita

O varianta modificata a algoritmului lui Dijkstra calculeaza drumurile minime de la toate nodurile catre o anumita destinatie. Ea conduce in mod natural la o varianta descentralizata, prezentata in cele ce urmeaza.



Desi algoritmul este convergent, asigurind gasirea drumurilor minime intr-un numar finit de pasi, el poate fi utilizat doar pentru datagrame, din cel putin doua motive:

- sint posibile modificari ale cailor, pe durata transmiterii pachetelor, astfel incit pachetele trimise pe o cale pot ajunge pe o alta cale;
- pe durata intervalului de convergenta, algoritmul nu evita producerea buclelor (trecerea pachetelor de mai multe ori prin acelasi nod).

Pentru a putea realiza calculele, fiecare nod al retelei pastreaza trei tablouri:

C - tabloul distantelor; C[d][v] este lungimea (sau costul) drumului de la nodul curent la nodul destinat, prin nodul vecin v;

D - tabloul distantelor minime; D[d] este lungimea drumului minime de la nodul curent la nodul destinat d;

V - tabloul de dirijare; V[d] este nodul vecin prin care se transmit datele, pe drumul minim, spre destinatul d.

Oricare nod trebuie sa poata detecta modificarile lungimii legaturilor sale. La producerea unei modificari, fiecare nod care o sesizeaza actualizeaza tabloul distantelor minime si de dirijare. Toate modificarile tabelii de dirijare sint comunicate vecinilor, in forma unor mesaje de forma (s,d,Dsd), unde Dsd este distanta minima de la s la d.

In cazul adaugarii unei noi legaturi la retea, noului vecin i se transmite toata tabela de dirijare. Mesajele mentionate pot fi transmise izolat sau grupate in aceeasi unitate.

Primirea unui mesaj de forma (s,d,Dsd) declanseaza in receptor procesul de calcul si actualizare a tabelii de dirijare, modificarile fiind transmise, la rindul lor, tuturor vecinilor. Procesul este convergent, la un moment dat, schimbul de mesaje si actualizarea tabelii incheindu-se.

Algoritmul trateaza trei evenimente distincte:

- adaugarea unei noi legaturi;
  - sesizarea modificarii lungimii unei linii;
  - primirea unui mesaj de control de la un nod vecin.
- Corespunzator, aveme cele trei functii descrise in continuare.

```
void adauga_legatura (int m)
{
    /* se adauga legatura (crt,m), crt fiind nodul curent*/
    C[m][m] = 1[m][m];
    calculeaza p pentru care C[m][p] = min C[m][w], dupa w;
    V[m]=p;
    if (C[m][p] != D[m])
        {D[m] = C[m][p];
```

```

    transmite mesaj (crt,m,D[m]) tuturor vecinilor;
}
transmite mesajele (crt,a,D[a]),..., (crt,z,D[z]) nodului m;
}

void schimba_cost (int m,int delta_crt_m)
{
for (toate destinatiile d)
{ C[d][m] += delta_crt_m;
  calculeaza p a.i. C[d][p] = min C[d][w],dupa w;
  V[d] = p;
  if (C[d][p] != D[d])
    { D[d] = C[d][p];
      transmite mesaj (crt,d,D[d]) tuturor vecinilor;
    }
}
}

void receptie_mesaj (int s,int d,int cost_s_d)
{
if (d != crt)
{ C[s][d] = cost_s_d + l[s][crt];
  calculeaza p a.i. C[d][p] = min C[d][w],dupa w;
  V[d] = p;
  if (C[d][p] != D[d])
    {D[d] = C[d][p];
      transmite mesaj (crt,d,D[d]) tuturor vecinilor;
    }
}
}
}

```

Asa cum s-a aratat, algoritmul creeaza probleme prin dirijarea pachetelor in bucla, intre noduri vecine: daca A trimite pachetele catre C prin B si daca legatura B-C se defecteaza, B va trimite pachetele catre C prin A, care are o estimare mai buna. Dar A va trimite pachetele catre C lui B, recirculind pachetele in bucla (vezi figura 5.3).

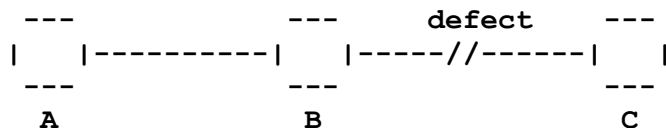


Figura 5.3,

Pentru a ocoli acest neajuns se foloseste principiul optimalitatii: daca B este pe ruta optima de la A la C, atunci calea cea mai buna de la B la C este inclusa in prima. Deci, caile catre C formeaza un arbore a carui cunoastere permite evitarea buclelor infinite.

### 5.3.5. Dirijarea ierarhica

Se utilizeaza pentru retele de mari dimensiuni la care tabelele de dirijare ar fi voluminoase. Comutatoarele sint grupate in regiuni, fiecare comutator cunoscind detaliat caile din regiunea proprie, dar necunoscind structura interna a altor regiuni. Doua regiuni sint legate prin conectarea unui anumit nod din prima regiune cu un anumit nod din a doua regiune. Tabela de dirijare se poate reduce, ea avind cite o intrare pentru fiecare nod din regiunea proprie si cite o intrare pentru fiecare din celelalte regiuni.

### 5.3.6. Dirijarea cu difuzare

Dirijarea unui pachet catre toate celelalte noduri se poate face prin tehnica inundarii. Pentru a evita degradarea performantelor rețelei se poate recurge la algoritmi descriși în continuare.

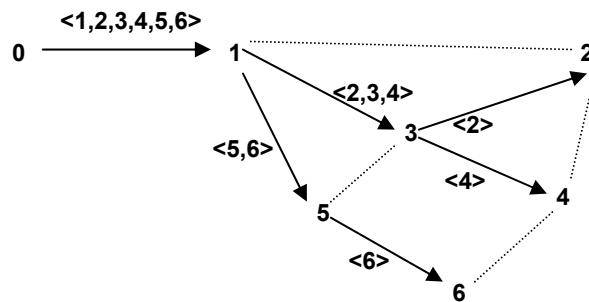


Figura 5.4.

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

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

### 5.4. Algoritmi de evitare a congestiei

\* Prealocarea zonelor tampon. Este aplicabilă circuitelor virtuale și constă în rezervarea unei sau mai multor zone tampon în fiecare nod intermediar, la deschiderea circuitului. În lipsa

de spatiu, se alege o alta cale sau se rejecteaza cererea de stabilire a circuitului.

\* Distrugerea pachetelor. Daca nu exista spatiul necesar memorarii, pachetul receptionat de un nod este ignorat. Deoarece prin aceasta se pot ignora pachete de confirmare, care ar duce la eliberarea spatiului ocupat de pachetele confirmate, se mentine cel putin un tampon de receptie pentru fiecare linie, permitinduse inspectarea pachetelor primite. De asemenea, se poate limita (inferior si superior) numarul zonelor tampon de transmisie ale fiecarei linii.

\* Pachete de permisiune. Se initializeaza reseaua cu pachete de permisiune (in numar fix). Cind un nod vrea sa transmita, el captureaza un pachet de permisiune si trimite in locul lui pachete de date. Receptorul regenereaza pachetul de permisiune. Se garanteaza astfel ca numarul maxime de pachete nu depaseste numarul de pachete de permisiune, fara a se asigura distribuirea lor conforme necesitatilor nodurilor. In plus, pierderea pachetelor de permisiune conduce la scaderea capacitatii retelei.

\* Pachete de soc. Sint transmise de comutatoare surselor de date pentru a mica rata de generare a pachetelor.

\* Evitarea blocarii definitive. Blocarea reprezinta o situatie limita a unei retele congestionate, cind lipsa de spatiu impiedica transmiterea vreunui pachet. O solutie de evitare a blocarii definitive este utilizarea in fiecare nod a  $m+1$  zone tampon,  $m$  fiind lungimea maxima a cailor retelei. Un pachet sosit de la calculatorul gazda local este acceptat in zona 0. In urmatorul nod trece in 1, apoi in 2 s.a.m.d. Zona " $m$ " a unui nod poate fi goala, poate contine un pachet pentru gazda locala, care este livrat, sau are un pachet pentru un nod distant, care este distrus. In toate cazurile zona " $m$ " se elibereaza, putind avansa un pachet din zona " $m-1$ ", apoi " $m-2$ " etc.

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

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

#### 5.5. Protocolul Internet al retelei ARPA

Internet Protocol (IP) a fost gindit, de la bun inceput, pentru a fi utilizat in sisteme interconectate de retele de

calculatoare, care folosesc comutarea de pachete. IP asigura transmiterea de pachete (datagrame) de la sursa la destinatie, sursa si destinatia fiind calculatoare gazda (host computers) identificate prin adrese de lungime fixa. IP asigura, de asemenea, fragmentarea pachetelor mai lungi decit dimensiunea maxima a unui cadru ce poate fi transmis intr-un anumit tip de retea. Nu trebuie uitat ca sursa si destinatia se pot gasi in retele diferite si ca un pachet poate strabate citeva retele pina sa ajunga la destinatie.

IP nu garanteaza ajungerea la destinatie a pachetelor si nu asigura secventierea blocurilor de date. Aceste servicii sint realizate de protocolul situat la nivelul imediat urmator, si anume Transmission Control Protocol (TCP).

Amplasarea protocolului IP in ierarhia de protocoale cu care interactioneaza este prezentata in figura 5.5.

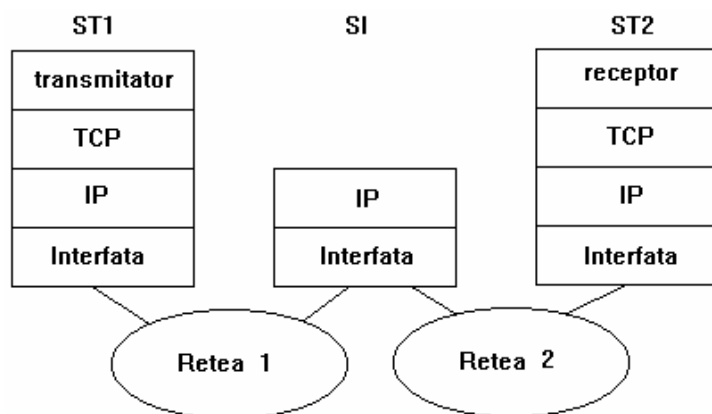


Figura 5.5.

Specificarea protocolului IP necesita definirea interfetei cu nivelul imediat superior (nivelul transport) si cu cel aflat sub el. Singura presupunere pe care o face IP in legatura cu protocolul legaturii de date este ca exista o modalitate de a transfera cadre de informatie de la un nod la altul adiacent, in aceeasi retea. Nu se face nici un fel de presupunere privind calitatea legaturii de date, tratarea si corectarea erorilor fiind atribute ale nivelului transport.

Protocolul IP indeplineste functiile, descrise in continuare.

#### 1. Adresarea

Pentru specificarea unei destinatii se folosesc trei elemente, la nivele distincte in ierarhia unei retele :

- numele destinatiei, la nivelul aplicatiei
- adresa destinatiei in cadrul retelei interconectate, la nivelul transport si
- adresa locala a destinatiei, la nivelul legaturii de date.

Functia de adresare a modulului Internet presupune stabilirea adresei locale pentru o adresa Internet. O adresa Internet are 32 biti. Ea cuprinde un numar de retea si o adresa locala. Exista mai multe clase de adrese internet, care difera prin lungimile celor doua cimpuri, asa cum se arata in figura 5.6.

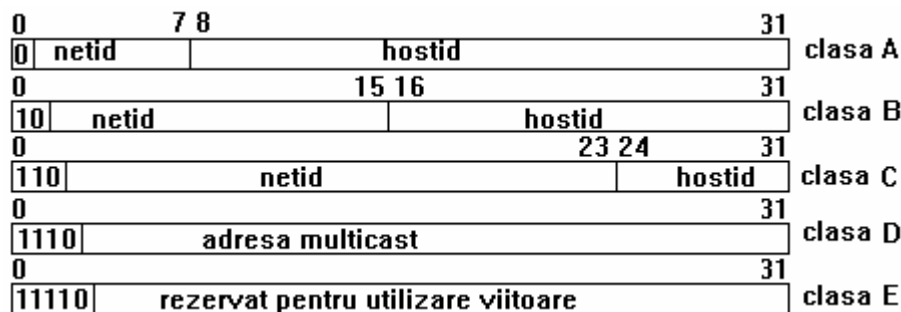


Figura 5.6.

Punerea in corespondenta a acestor adrese trebuie sa tina seama ca un acelasi nod al retelei poate avea mai multe adrese Internet, fiind multiplicat din punct de vedere logic, si ca un acelasi nod poate avea mai multe interfete de retea si deci mai multe adrese locale.

## 2. Fragmentarea

Fragmentarea unui pachet este necesara atunci cind el provine dintr-o retea cu o dimensiune mare a cadrului de date. Un pachet poate fi marcat "a nu se fragmenta". Un astfel de pachet va fi indrumat pe o cale ce evita fragmentarea, iar daca nu se poate va fi ignorat. Fragmentarea trebuie sa poata fi facuta intr-un numar arbitrar de cadre si trebuie prevazuta posibilitatea de reasamblare corecta, in secventa, la destinatie. Acest lucru se obtine prin interpretarea cimpurilor de Indicatori (Flags) si Deplasare (Offset), din cadrul antetelor Internet ale fragmentelor. Formatul antetului Internet este prezentat in figura 5.7.

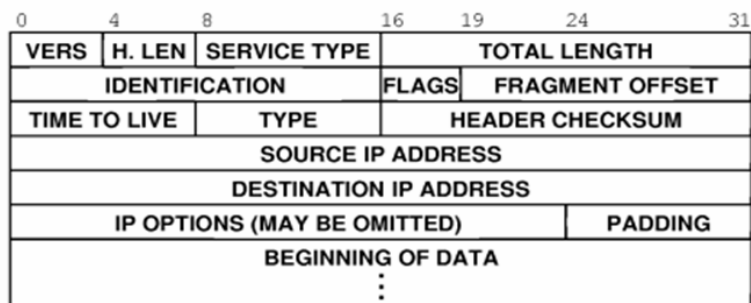


Figura 5.7.

Descrierea cimpurilor este urmatoarea:

**VERS (4 biti) - Selecteaza formatul antetului Internet (Formatul prezentat corespunde versiunii 4).**

**H.LEN (4 biti) - Lungimea antetului (headerului) Internet (exprimat in cuvinte de 32 biti).**

**SERVICE TYPE (Tip serviciu 8 biti) - Conform RFC 791 Oferă o indicatie asupra calitatii serviciului dorit, in conformitate cu conventiile din figura 5.8.**

biti 0-2: precedenta (defineste prioritatea pachetului)  
bit 3: intarziere 0 = normala, 1 = redusa  
bit 4: productivitate 0 = normala, 1 = ridicata  
bit 5: fiabilitate 0 = normala, 1 = ridicata  
biti 6-7: rezervati pentru utilizari viitoare

#### **Figura 5.8.**

Campul este folosit acum (conform RFC 3168) pentru servicii diferite (DiffServ) si notificare explicita a congestiei (ECN - Explicit Congestion Notification).

**TOTAL LENGTH (Lungime totala 16 biti) - Lungimea pachetului masurata in octeti, inclusiv lungimea antetului. Pentru o functionare corecta destinatia trebuie sa fie pregatita pentru a primi tot pachetul.**

**IDENTIFICATION (16 biti) - Identificarea fragmentului. Valoare setata de transmitator pentru a facilita reasamblarea fragmentelor unui pachet.**

**FRAGMENT OFFSET (13 biti) - Indica pozitia fragmentului in pachet.**

**FLAGS (3 biti) - Indicatori de control, cu semnificatiile:**

**Rezervat**

**DF = Don't Fragment - nu fragmenta pachetul**

**MF = More Fragments - nu este ultimul fragment din pachet**

**TIME TO LIVE (8 biti) - Timpul maxime pe care poate sa-l petreaca un pachet in sistemul Internet (in secunde). Este decrementat la fiecare trecere printr-un modul Internet, pentru a preveni aglomerarea retelei cu pachete ce nu pot ajunge la destinatie.**

**TYPE (Protocolul 8 biti) - Indica protocolul de la nivelul superior (TCP, UDP, etc.). Cateva valori posibile:**

- 1: Internet Control Message Protocol (ICMP)
- 2: Internet Group Management Protocol (IGMP)
- 6: Transmission Control Protocol (TCP)
- 17: User Datagram Protocol (UDP)
- 89: Open Shortest Path First (OSPF)
- 132: Stream Control Transmission Protocol (SCTP)

**HEADER CHECKSUM (Suma control antet 16 biti) - Calculata doar pentru antet. Este verificata de fiecare modul Internet. Daca este invalida pachetul este neglijat.**

**Adresa sursa (32 biti) - Adresa IP a sursei**

**Adresa destinatie (32 biti) - Adresa IP a destinatiei**

**IP OPTIONS (Optiuni un numar variabil de biti) - Au lungime diferita, apar sau nu in antet. Exista optiuni privind:**

    Securitate - cat de secreta este datagrama

    Strict source routing - arata calea completa de parcurs

    Loose source routing - arata ruterele care nu trebuie sarite

    Record route - fiecare ruter trebuie sa-si adauge adresa IP

    Timestamp - fiecare ruter adauga amprenta de timp.