Capitolul 6. NIVELUL TRANSPORT

6.1. Caracterizare generala

Nivelul transport este inima ierarhiei de protocoale, avand rolul de asigurare a unui transfer de date corect, eficient intre sistemul sursa si sistemul destinatar. El prezinta multe similitudini cu nivelul retea:
- are servicii orientate si neorientate pe conexiuni;
- conexiunile au trei faze: stabilire, transfer de date si desfiintare;
- adresarea si controlul fluxului se fac similar.

De ce, atunci, un nivel special de transport? Nivelul retea este puternic dependent de subreteaua de comunicatie, performantele si caracteristicile sale fiind determinate de aceasta. Utilizatorul nu poate interveni in subretea pentru a-i mari performantele sau schimba caracteristicile. Se poate insa adauga subretelei un nivel care sa permita:
- un transfer sigur al datelor, chiar cu o retea nesigura;
- o interfata uniforma pentru utilizatori, adica un set standard de primitive de serviciu, independent de tipul subretelei utilizate.

Nivelul transport separa nivelele arhitecturale in doua categorii: nivelele 1-4, alcatuind furnizorul serviciului de transport si nivelele 5 - 7, alcatuind utilizatorul acestui serviciu.Primele au functii orientate spre comunicatii, iar celelalte spre organizarea dialogului si a transformarilor sintaxei unitatilor de date comunicate intre utilizatori.

Fiind un nivel de granita, transportul este important prin calitatea serviciilor oferite utilizatorilor, ea fiind rezultatul compunerii calitatii nivelelor inferioare. Calitatea este caracterizata prin urmatorii parametri, referitori la comunicarea datelor pe o conexiune de transport:
- intarzierea de stabilire a conexiunii;
- probabilitatea de esec la stabilirea conexiunii;
- productivitatea, reprezentand numarul maxim de octeti ce pot fi transferati intr-o secunda, in fiecare sens al conexiunii;
- intarzierea de transmisie;
- rata erorilor reziduale (teoretic zero, practic o valoare nenula foarte mica);
- probabilitatea de esec la transport, care arata diferenta intre rata anuntata a erorilor si cea reala;
- intarzierea de desfiintare a conexiunii;
- probabilitatea de esec la desfiintarea conexiunii;
- nivelul de protectie;
- prioritatea, conform careia anumite conexiuni sunt servite inaintea altora;
- rezilierea, care da probabilitatea ca nivelul transport sa termine o conexiune datorita unor probleme interne.

Calitatea serviciului se negociaza la stabilirea unei conexiuni: utilizatorul specifica valorile dorite si valorile acceptabile ale conexiunii. Daca valorile minime nu pot fi asigurate, conexiunea nu se stabileste.

6.2. Primitivele serviciului de transport

Sunt disponibile primitive pentru servicii orientate si neorientate pe conexiuni. Lista primitivelor, specificate de standardul ISO 8072 este prezentata in figura 6.1.

1. T-CONNECT.request (apelat, apelant, exp, calitate, date)
2. T-CONNECT.indication (apelat, apelant, exp, calitate, date)
3. T-CONNECT.response (calitate, respondent, exp, date)
4. T-CONNECT.confirm (calitate, respondent, exp, date)

5. T-DISCONNECT.request (date)
6. T-DISCONNECT.indication (motiv, date)

7. T-DATA.request (date)
8. T-DATA.indication (date)
9. T-EXPEDITED-DATA.request (date)
10. T-EXPEDITED-DATA.indication (date)

a) servicii orientate pe conexiuni

11. T-UNITDATA.request (apelat, apelant, calitate, date)
12. T-UNITDATA.indication (apelat, apelant, calitate, date)

b) servicii neorientate pe conexiuni

Figura 6.1.
(exp- solicita sau nu transport expeditiv; apelat, apelant, respondent - adrese ale unor puncte de acces la servicii de transport)

Observam absenta confirmarilor pentru date, justificata prin aceea ca serviciul de transport este sigur. De asemenea, lipseste o primitiva de resetare a conexiunii, prezenta in cadrul nivelului retea, considerandu-se ca transportul rezolva problema stabilirii unor noi conexiuni in cazul aparitiei defectelor. Cu toate acestea, erorile ireparabile pot provoca deconectarea, semnalata prin primitive T-DISCONNECT.indication generate de nivelul transport si adresate utilizatorilor.

Comportarea unui capat al conexiunii de transport este caracterizata de patru stari si se supune diagramei de tranzitii prezentata in figura 6.2 (pentru un serviciu orientat pe conexiuni).

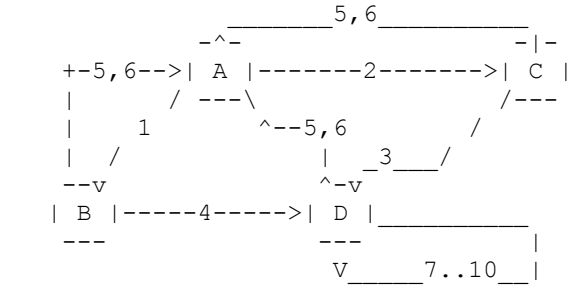


Figura 6.2.

Starile au urmatoarele semnificatii:
A - liber, sunt posibile conexiuni initiate local sau de la distanta;
B - s-a comandat conectarea, asteptandu-se raspunsul entitatii distante;
C - s-a primit o solicitare de conectare si inca nu s-a transmis raspunsul;
D - s-a realizat o conexiune valida, putandu-se face transferul de date.
Numerele asociate tranzitiilor corespund primitivelor din figura 6.1.

6.3. Protocoale de transport

Ca si legatura de date, nivelul transport trebuie sa corecteze imperfectiunile nivelului inferior. De aceea, el are functii similare legaturii de date: detectarea erorior, controlul fluxului, gestiunea conexiunilor. Conditiile oferite de nivelul inferior sunt insa diferite de cele furnizate legaturii de date de nivelul fizic, influentind solutiile adoptate. Dam un singur exemplu, referitor la transportul unitatilor de date: in cazul unei legaturi de date, un cadru poate ajunge la destinatar sau poate fi pierdut; la nivelul transport, datele parcurg o subretea in care pot fi memorate temporar; ele pot intarzia in nodurile subretelei si pot apare la destinatar la momente inoportune. Tratarea unor astfel de cazuri necesita protocoale de o complexitate mai mare.

Functiile adaugate de nivelul transport depind de subreteaua de comunicatie. Putem grupa subretelele in trei categorii:
A - retele cu rata acceptabila de erori semnalate (de exemplu, resetari explicite ale conexiunilor de retea) si rata acceptabila de erori nesemnalate (erori de transmisie nedetectate si necorectate de retea);
B -retele cu rata inacceptabila de erori semnalate si rata acceptabila de erori nesemnalate;
C -retele cu rata inacceptabila de erori semnalate si nesemnalate.

Protocoalele de transport utilizate se impart in 5 clase.
-clasa 0 - pe retea A - protocol simplu, fara detectie de erori; el se ocupa de stabilirea si desfiintarea conexiunilor, transferul datelor, fragmentarea mesajelor lungi in mai multe unitati de date ale protocolului de transport (UDPT); restul (controlul fluxului, recuperarea erorilor etc) este realizat de retea;
-clasa 1 - pe retea B - protocol cu recuperarea erorilor semnalate; el trateaza resetarile generate de retea in cazul unor erori grave, prin stabilirea unor noi conexiuni si continuarea transportului datelor; in acest scop UDT au numere de secventa, caracteristica absenta la protocoalele din clasa 0;
-clasa 2 - pe retea A - fara detectie de erori dar cu multiplexare (mai multe conexiuni de transport pe o singura conexiune de retea) si control de flux;
-clasa 3 - pe retea B - cu recuperarea erorilor semnalate, control de flux si multiplexare;
-clasa 4 - pe retea C - cu recuperarea erorilor semnalate si nesemnalate, control de flux si multiplexare.

6.4. Gestiunea conexiunii de transport.
6.4.1. Adresarea

Cand un utilizator A doreste sa stabileasca o conexiune, el trebuie sa specifice utilizatorul B la care sa se conecteze.

Metoda uzuala este ca B sa se ataseze la un punct de acces la servicii de transport (PAST) si sa astepte cererea de conectare. La randul sau, A va specifica an cerere adresa PAST a lui B. Conectarea este realizata de entitatile transport din sistemele care gazduiesc utilizatorii si este mijlocita de nivelul retea. Un scenariu posibil de conectare este urmatorul:
- B se atasaza la PAST 100 si asteapta o cerere de conectare (T-CONNECT.indication);
- cand A doreste sa comunice cu B, face o cerere de conectare (T-CONNECT.request) in care specifica PAST 6 ca sursa si PAST 100 ca

destinatar;
- entitatea transport locala lui A selecteaza puncte de acces la servicii retea (PASR) locala si distanta si stabileste o conexiune retea intre ele;
- prin aceasta conexiune, entitatea transport locala lui A transmite un mesaj de conectare entitatii transport locale lui B;
- B genereaza T-CONNECT.indication la PAST 100 si, daca ea este confirmata pozitiv, conexiunea se stabileste.

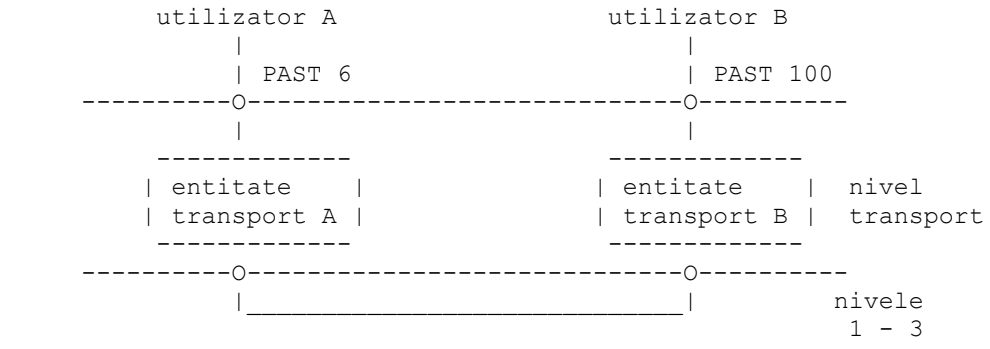


Figura 6.3

O problema importanta este: de unde cunoaste A adresa PAST 100 a lui B ? Solutia de a "lega" un utilizator de un anumit PAST cunoscut de toti ceilalti utilizatori este prea rigida. S-au

conturat, printre altele, doua alternative.

Schema folosita in ARPA defineste in fiecare masina care ofera servicii, un proces "server", care cunoaste toate serviciile puse la dispozitia utilizatorilor. Utilizatorul A incepe prin a se conecta la acest server, caruia ii adreseaza un mesaj despre serviciul solicitat. Serverul alege un PAST (de ex. PAST 100), pune in executie pe B comunicandu-i adresa PAST la care urmeaza sa se ataseze in asteptarea unei cereri de conectare si transmite aceeasi adresa (PAST 100) lui A. Apoi, A si B se vor conecta dupa scenariul precedent.

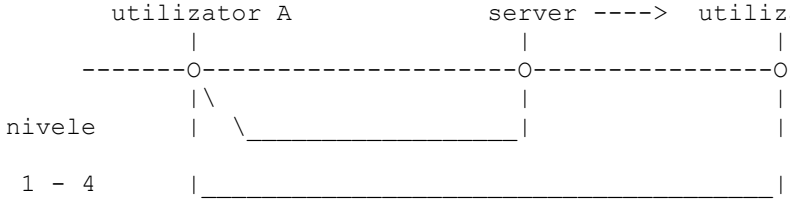


Figura 6.4

Intr-o a doua varianta, un server "director" pastreaza corespondenta dintre numele serviciilor cunoscute (de ex. B) si adresa lor PAST (de ex. PAST 100). Utilizatorul A se conecteaza la acest server, afla adresa lui B, dupa care se conecteaza la B. Orice serviciu nou isi anunta adresa serverului director (ca la serviciul de informatii al abonatilor telefonici).

Odata determinata adresa PAST 100 a lui B, entitatea transport locala lui A trebuie sa cunoasca in ce sistem este situata aceasta, pentru a putea stabili adresele PASR necesare conexiunii retea. Rezolvarea este mult usurata daca adresele PAST sunt structurate, avand campuri diferite pentru tara, retea, gazda, port. Cu aceasta schema, adresele PAST sunt o concatenare a

adreselor PASR si a unui numar de port in cadrul nodului gazda.

6.4.2. Gestiunea transferului datelor

Stabilirea unei conexiuni de transport necesita protocoale mai complicate decat pentru legatura de date. Sa presupunem ca un utilizator stabileste o conexiune cu o banca, cere transferul unei sume de bani si desfiinteaza conexiunea. Daca, din intamplare, fiecare pachet este duplicat si intarziat in retea, va avea loc un nou transfer, nedorit de bani.

O cale de a ocoli acest neajuns este de a asocia fiecarei conexiuni un identificator, ales de entitatea transport initiatoare si pus in antetul fiecarei unitati de date. La fiecare noua conexiune se aloca un nou identificator, diferit de cele ale conexiunilor anterioare (active sau desfiintate). Schema are dezavantajul ca trebuie retinute indefinit valorile identificatorilor utilizati, iar in cazul unei pene care determina pierderea informatiei, entitatea transport nu mai poate gestiona noile conexiuni.

Pentru a limita numarul identificatorilor memorati, se recurge la introducerea unui mecanism de distrugere a pachetelor vechi. Acesta foloseste fie un contor de noduri vizitate, fie o informatie despre momentul generarii, inclusa in antet. In practica este necesara garantarea distrugerii nu numai a pachetelor, ci si a confirmarilor. Fie T un multiplu al duratei maxime de viata a unui pachet (factorul de multiplicitate este dependent de protocol) in care pachetele si confirmarile lor sunt distruse cu siguranta.

Pentru a putea trata corect duplicatele, se echipeaza fiecare gazda cu un ceas. Ceasurile au forma unor numaratoare si nu sunt neaparat sincronizate intre ele. Unitatile de date au numere de secventa, valoarea lor maxima fiind mai mica decat valoarea maxima a ceasurilor. Ceasurile functioneaza chiar la caderea calculatoarelor gazda. Pentru a evita aparitia a doua unitati de date cu acelasi numar de secventa, la stabilirea unei conexiuni valoarea initiala a numerelor de secventa se stabileste la cea data de pozitiile mai putin semnificative ale ceasului. Plaja numerelor de secventa trebuie sa fie suficient de mare astfel ca la reluarea numerelor de secventa, pachetele anterioare cu aceleasi valori sa fie distruse.

La caderea calculatorului gazda, entitatea transport, care nu mai stie unde se gasea in spatiul numerelor de secventa, ar trebui sa astepte T secunde pentru distrugerea certa a pachetelor anterior transmise. Se poate ocoli aceasta asteptare daca se impune o restrictie asupra utilizarii numerelor de secventa. Pentru a o intelege, sa urmarim un exemplu.

Fie T=60 secunde si cadenta ceasului de un increment pe secunda. Presupunem ca la t=30 secunde, o unitate de date cu numar de secventa 80 a fost transmisa pe conexiunea 5, dupa care gazda cade si reporneste la scurt interval. La t=60 se redeschid conexiunile 0..4, iar la t=70 se redeschide conexiunea 5 cu numar initial de secventa 70. In urmatoarele secunde se transmit unitatile de date 70..80, astfel ca la t=85 exista doua unitati de date pentru conexiunea 5 cu numar de secventa 80.

Pentru a ocoli acest neajuns, putem interzice transmiterea unui numar de secventa in intervalul de T secunde care precede

posibila sa utilizare ca numar de secventa initial.

6.4.3. Stabilirea unei conexiuni

Metoda prezentata rezolva problema duplicatelor pentru unitatile de date. In cazul stabilirii conexiunilor, se poate recurge la un algoritm cu confirmare, in trei pasi. El nu cere ca ambele capete sa inceapa sa transmita cu acelasi numar de secventa. Desfasurarea normala a evenimentelor este prezentata a in figura 6.5.

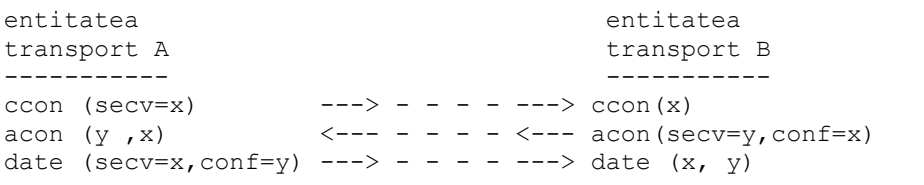


Figura 6.5.
(ccon - cerere conectare; acon - confirmare conectare)

Se arata aici mesajele schimbate intre entitatile de transport, x si y fiind numere de secventa initiale (eventual diferite) alese de cele doua entitati.

In cazul unui mesaj intarziat, care ajunge la B fara stirea lui A, B reactioneaza transmitand o confirmare, care reprezinta o verificare ca A a cerut intr-adevar conectarea. Primind a inapoi o rejectare, B renunta la stabilirea conexiunii (figura 6.6).

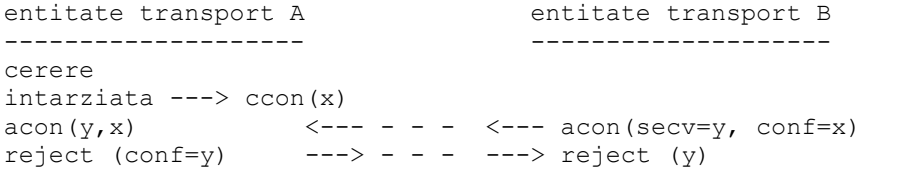


Figura 6.6.

In fine, cand atat cererea de conectare cat si datele sunt intarziate, avem situatia din figura 6.7.

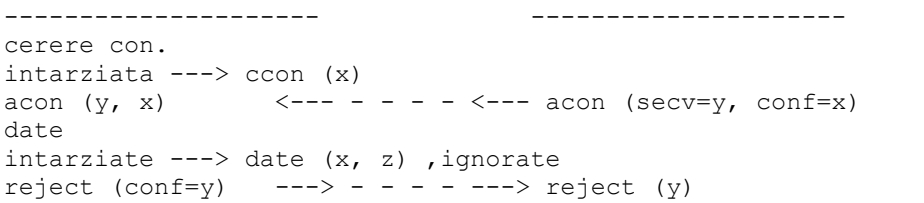


Figura 6.7.

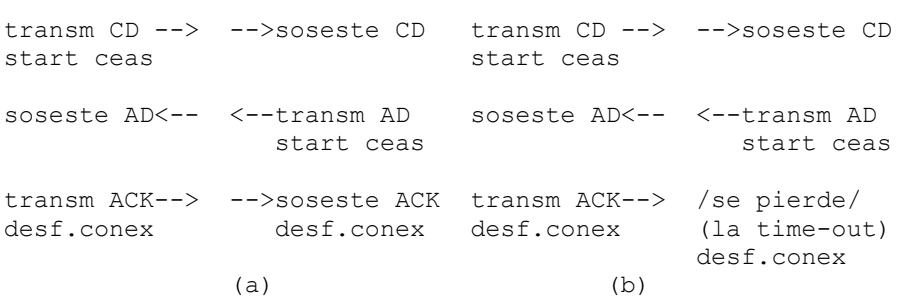
Ca si mai inainte, B primeste o cerere de conectare a intarziata, la care raspunde alegand un numar initial de secventa y, care nu mai poate apare in unitati de date sau in confirmari ale lor. Cand primeste date intarziate, cu numar de confirmare z si nu y, B stie ca acestea reprezinta duplicate mai vechi si le ignora.

6.4.4. Desfiintarea conexiunilor

Desfiintarea unei conexiuni este mai simpla. Ea poate avea loc intr-unul din urmatoarele moduri:
- un utilizator (A) cere deconectarea prin T-DISCONNECT.request,

nivelul transport generand la celalalt capat (B) T-DISCONNECT.indication;
- ambii utilizatori (A si B) cer deconectarea prin T-DISCONNECT.request;
- nivelul transport determina deconectarea, provocand T-DISCONNECT.indication la ambele capete.

Toate variantele pot conduce la pierdere de date, daca un utilizator se deconecteaza inainte de primirea tuturor mesajelor de la cel de al doilea utilizator. Remediul este utilizarea unui algoritm cu confirmare, in trei pasi, care nu este infailibil, dar este satisfacator in practica (o metoda ideala ar cere ca un utilizator sa nu se deconecteze decat daca este sigur ca celalalt este pregatit sa faca ,de asemenea, acest lucru; aplicarea ei la un mediu cu erori face deconectarea imposibila !) Prezentam cateva scenarii de deconectare in trei pasi, evidentiind operatiile entitatilor de transport si unitatile de date de protocol de transport (UDPT) schimbate intre ele.



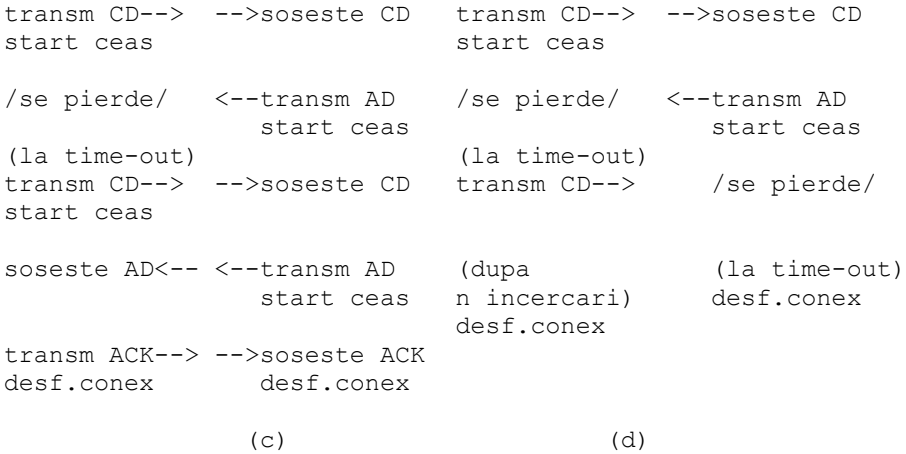


Figura 6.8.
(CD - cerere deconectare; AD - confirmare deconectare;
ACK - confirmarea confirmarii)

Secventa normala este cea din figura 6.8.(a). Daca se pierde ultima confirmare (ack), situatia este corectata prin ceas: la time-out, conexiunea este oricum desfiintata (figura 6.8.(b)). Figurile 6.8 (c) si (d) prezinta secventele de evenimente corespunzatoare pierderii confirmarii de deconectare (AD), respectiv a pierderii confirmarii AD si a cererii de deconectare CD, succesiv. In ambele situatii, ceasurile joaca un rol important in desfasurarea corecta a deconectarii. In cazul (d), dupa n incercari, entitatea transport din stanga desfiinteaza conexiunea, chiar daca nici una din cererile sale de deconectare nu au ajuns la cealalta unitate (situatie nefigurata). Aceasta desfiintare a unei jumatati de conexiune poate fi sesizata de

entitatea pereche prin absenta oricarei reactii la actiunile sale, o vreme indelungata.

6.5. Controlul fluxului.

Asemanarea cu nivelul legaturii de date este utilizarea protocoalelor cu fereastra glisanta (sau echivalente) pentru a impiedica un transmitator mai rapid sa supraincarce un receptor lent. Diferenta consta in numarul mult mai mare de conexiuni de transport decat cel al conexiunilor legaturii de date, ceea ce nu permite folosirea acelorasi strategii de memorare temporara a datelor.

In cazul unor subretele de tip B sau C, transmitatorul pastreaza datele pana la confirmarea receptiei lor. Chiar si in cazul retelelor de tip A, poate apare aceeasi cerinta, daca receptorul nu asigura memorarea datelor primite si , din lipsa de spatiu, ignora o parte a lor. O problema aparte, in acest ultim caz, este dimensionarea convenabila a zonelor tampon de transmisie si de receptie.

In orice caz, numarul mare de conexiuni impune alocarea dinamica a zonelor tampon si utilizarea unor tampoane de dimensiuni variabile. Alocarea dinamica atrage dupa sine necesitatea utilizarii unor ferestre de receptie de dimensiuni variabile: transmitatorul anunta numarul de mesaje pe care intentioneaza sa le trimita. Receptorul garanteaza initial cate poate si apoi, treptat, restul. Deoarece permisiunile (garantiile) nu au legatura cu confirmarile mesajelor corecte, cele doua se transmit separat, asa cum se arata in exemplul urmator, pentru o numarare modulo 16 a mesajelor.

A	mesaj	B	comentariu
1	>cere 8 tampoane>		
2	<ack=15,buf=4<		B garanteaza mesajele 0..3
3	>secv=0,data=m0>		
4	>secv=1,data=m1>		
5	>secv=2,data=m2>....		mesaj pierdut
6	<ack=1,buf=3<		B confirma 0,1 garanteaza 2..4
7	>secv=3,data=m3>		
8	>secv=4,data=m4>		A trebuie sa se opreasca
9	>secv=2,data=m2>		A retransmite la time-out
10	<ack=4,buf=0<		B confirma tot dar nu mai garanteaza nimic
11	<ack=4,buf=1<		A poate transmite 5
12	<ack=4,buf=2<		A poate transmite si 6
13	>secv=5,data=m5>		
14	>secv=6,data=m6>		A este iar blocat
15	<ack=6,buf=0<		A este inca blocat
16	...<ack=6,buf=4<		permisiunea este pierduta si A ramane blocat

Pentru a evita blocarea prin pierderea permisiunilor (ca in exemplul anterior), entitatile transport trebuie sa-si transmita periodic confirmarile si starea tampoanelor, pe fiecare conexiune.

Presupunand ca receptorul are suficienta memorie, o alta sursa a gatuirilor in retea este capacitatea limitata a subretelei. Daca subreteaua poate transmite c unitati de date/sec si durata unui ciclu, inclusiv confirmarea este r sec, fereastra

transmitatorului trebuie sa fie c*r, pentru o umplere completa a timpului de transmitere. Ea este ajustata periodic, prin masurarea parametrilor c si r. Evaluarea lui c se face simplu, prin numararea confirmarilor intr-o anumita perioada de timp.

6.6. Refacerea dupa caderea calculatoarelor gazda

Pentru a ilustra dificultatea refacerii dupa eroare, sa presupunem ca o conexiune de transport este folosita pentru transferul unui fisier. In mijlocul transferului, receptorul cade, iar la reluare cere tuturor celorlalte gazde sa transmita informatii despre conexiunile transport deschise. Daca se foloseste un protocol start-stop, transmitatorul poate avea sau nu un mesaj in transfer, inca neconfirmat. Intrebarea este daca trebuie sau nu trebuie sa transmita acest ultim mesaj.

Deoarece receptia unei unitati de date, confirmarea ei si scrierea sa in fisierul destinatar sunt operatii distincte, caderea putandu-se produce in orice moment pe parcursul lor, nu este posibila determinarea in nivelul transport a efectului ultimei operatii, deci daca fisierul destinatar a fost sau nu actualizat cu ultimul mesaj transmis. Ca urmare, recuperarea nu poate fi realizata de entitatile transport si nu poate fi transparenta nivelelor superioare.

6.9. Nivelul transport in retelele publice

Completam imaginea nivelului transport a
in conceptia OSI, prezentand elementele standardului ISO 8073
referitor la
protocoalele de transport.

Aceste protocoale admit 10 tipuri diferite de UDPT (unitati de date ale protocolului de transport). Fiecare UDPT are pana la patru parti:
- indicatorul de lungime a partilor fixa si variabila din antet;
- partea fixa a antetului, dependenta de tipul UDPT;
- partea variabila a antetului;
- date ale utilizatorului.

Formatul general al unei UDPT este prezentat in figura 6.12.

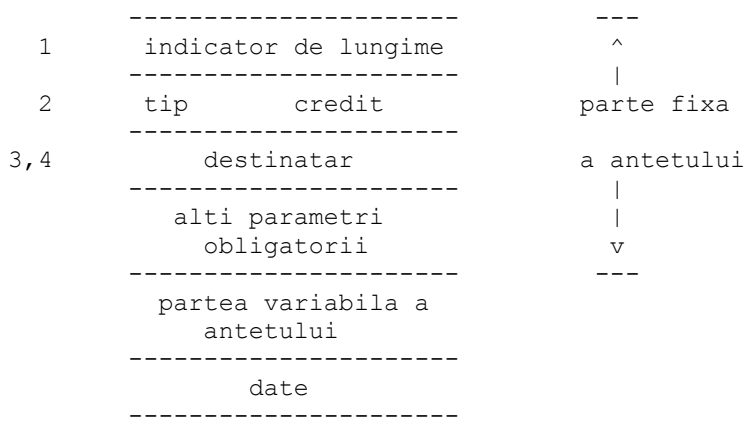


Figura 6.12.

Partea fixa a antetului difera de la un tip de cadru la altul. Ea include parametri cum ar fi: identificatorul sursei, motivul cererii de deconectare, numarul de secventa al datelor confirmate, numarul de secventa al datelor curente etc. Partea variabila poate cuprinde urmatoarele optiuni:

- PAST local si distant;
- dimensiunea unitatii de date;
- suma de control a unitatii de date;
- clase alternative de protocoale acceptabile;
- parametrii de calitate.

Sunt disponibile urmatoarele tipuri de UDPT:
CR- cerere conectare,
CC- confirmare conectare,
DR- cerere deconectare,
DC- confirmare deconectare,
DT- date,
ED- date expeditive,
AK- confirmare date,
EA- confirmare date expeditive,
RJ- rejectare, utilizata pentru resincronizare dupa resetarea retelei,
ER- raportare erori de protocol,
UD- date, pentru serviciu fara conexiuni.

6.10. TCP - Transmission Control Protocol

TCP este un protocol fiabil de comunicare intre procese aflate in calculatoare interconectate, folosind comutarea de pachete. TCP este orientat pe conexiuni si se situeaza la nivelul transport din ierarhia OSI.

TCP presupune ca la nivelul imediat inferior (nivelul retea) exista o modalitate, chiar nefiabila de transmitere a pachetelor in retea. El a fost gandit ca avand la nivelul retea un modul Internet (IP), dar poate functiona foarte bine si cu alte protocoale.

Interfata dintre modulul TCP si modulele de nivel superior se face prin apeluri similare celor pe care un sistem de operare le ofera pentru manipularea fisierelor.

Protocolul TCP trebuie sa asigure urmatoarele:
- transfer de date:

Modulul TCP trebuie sa asigure transferul unui flux continuu de date in ambele directii, intre doua procese.

- fiabilitate:

Modulul TCP trebuie sa refaca datele din pachete eronate, sa tina cont de pachetele pierdute, de pachete duplicate sau trimise in alta ordine de sistemul de comunicatie de la nivelul retea. Pentru aceasta, fiecare pachet primeste un numar de secventa si necesita confirmare de primire. Daca nu este primita confirmarea intr-un interval maxim de timp, datele neconfirmate sunt retransmise. Numerele de secventa servesc atat la refacerea fluxului de date cat si la eliminarea pachetelor duplicate. Deci, atata timp cat modulele TCP vor functiona corect si sistemul de retele nu devine complet partitionat, erorile mediului fizic de propagare a datalor nu vor influenta corectitudinea fluxului de date.

- controlul fluxului de date:

Modulul TCP asigura o modalitate prin care receptorul poate controla cantitatea de date furnizata de transmitator. Acest lucru se realizeaza insotind fiecare confirmare, de o "fereastra" de permisiune care indica domeniul de numere de secventa in care transmitatorul poate furniza date, fara a primi o noua confirmare.

- multiplexare:
Prin multiplexare, se permite ca doua sau mai multe procese din acelasi calculator sa foloseasca simultan facilitatile TCP.
Pentru aceasta, modulul TCP furnizeaza in cadrul fiecarui calculator mai multe porturi de comunicatie. Adresa de port, concatenata cu adresa Internet formeaza un soclu (socket). O conexiune este identificata, in mod unic, de o pereche de socluri, fiind posibil ca acelasi soclu sa fie folosit de mai multe conexiuni. O practica uzuala este de a asocia anumitor servicii oferite de unele procese (servicii de "log-in", de acces la sistemul de fisiere, etc.) numere de port fixe, cunoscute de toti utilizatorii.

- servicii orientate pe conexiune:
Serviciile de fiabilizare si control al fluxului de date impun unui modul TCP sa mentina pentru fiecare flux de date anumite structuri de control (soclul, numere de secventa, dimensiunea ferestrei de comunicare, etc). O conexiune este definita de structurile de control din cele doua procese ce comunica prin fluxuri de date. Deci, pentru a stabili o conexiune intre doua procese ce doresc sa comunice, modulele TCP proprii trebuie mai intai sa stabileasca o conexiune (un canal de comunicatie). Stabilirea unei conexiuni presupune initializarea celor doua structuri de control, cu valori corelate. In acest scop, are loc un dialog prealabil intre cele doua module TCP, pentru initializarea structurilor de control. Un modul TCP pune la dispozitie doua functii de deschidere de conexiune, una activa, de initiere a conexiunii si alta pasiva, de raspuns la orice cerere de stabilire de conexiune.

- prioritati si securitate:
Utilizatorii modulelor TCP pot indica nivelele de prioritate si de securitate pentru transferul de date.

Pentru a transmite datele, modulele TCP folosesc pachete IP. Fiecare pachet va contine dupa antetul IP un antet TCP cu informatii specifice acestui protocol. Formatul antetului este prezentat in figura 6.13.

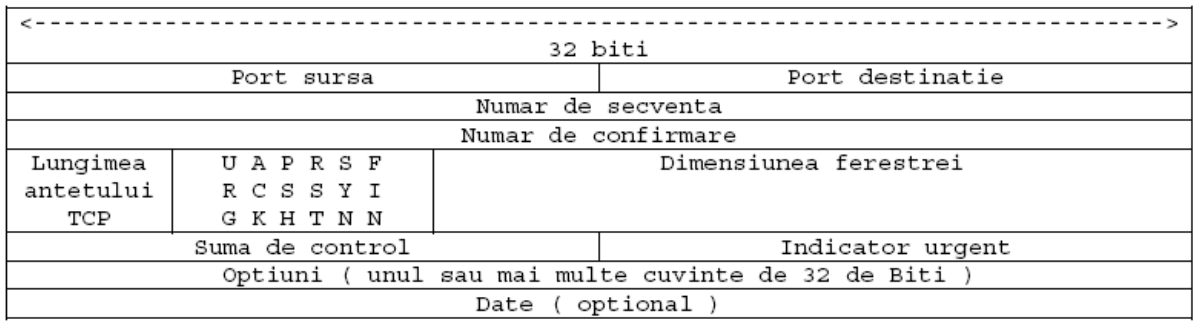


Figura 6.13.

Semnificatia campurilor este urmatoarea:

Portul sursa (16 biti) - Numarul portului sursa. Impreuna cu adresa IP a sursei (din antetul de retea) formeaza identificatorul complet al capatului sursa al conexiunii TCP. Adresa IP identifica o interfata de retea (mai precis, o masina din retea), iar portul identifica unul din procesele (software) aflate la acea adresa IP (care ruleaza deci pe masina respectiva).

Portul destinatar (16 biti) - Numarul portului destinatar. Selecteaza procesul din calculatorul destinatar cu care s-a stabilit o conexiune.

Numarul de secventa (32 biti) - Reprezinta numarul primului octet de date din cadrul segmentului de date curent. Daca bitul de control SYN este setat, campul reprezinta numarul de secventa initial (NSI) iar primul octet de date are numarul de secventa NSI+1 (vezi semnificatia bitilor de control).

Numarul de confirmare (32 biti) - Daca bitul de control ACK este setat, contine valoarea urmatorului numar de secventa pe care receptorul il asteapta.

Ofset date (4 biti) - Contine lungimea antetului TCP in cuvinte de 32 biti, indicand astfel de unde incep datele.

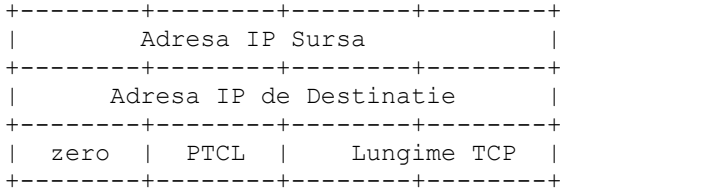
Rezervat (6 biti) - Sunt initializati cu 0

Biti de control (6 biti)

URG - Campul "pointer imediat" este semnificativ
ACK - Valideaza numarul de confirmare.
PSH - Cere functia PUSH (cere anuntarea imediata a utilizatorului destinatie de primirea mesajului)
RST - Cere resetarea conexiunii.
SYN - Cere sincronizarea numerelor de secventa.
FIN - Anunta terminarea fluxului de date de la transmitator.

Fereastra (16 biti) - Reprezinta numarul de octeti, incepand cu cel indicat prin numarul de confirmare, pe care cel ce trimite mesajul il poate receptiona.

Suma de control (16 biti) - Este o suma de control calculata pentru toate cuvintele din antet, din blocul de date si dintr-un pseudo-antet cu urmatorul format:



PTCL este numarul de protocol pentru TCP (=6) iar Lungime TCP reprezinta lungimea segmentului TCP (antet si date, fara pseudo-antet). Includerea pseudo-antetului in calculul sumei de control permite o protectie (la nivel TCP) impotriva segmentelor dirijate gresit. Mecanismul reprezinta totodata o incalcare a principiului separarii nivelelor arhitecturale prin includerea la nivelul transport a unei informatii dependente de nivelul retea (adresele IP sursa si destinatie).

Daca numarul de octeti de date este impar se completeaza cu un octet nul pentru calculul sumei de control. Octetul nul nu se transmite destinatarului.

Algoritmul de calcul pentru suma de control este urmatorul:
la transmisie: aduna cuvinte de 16 biti in complement fata de 1
complementeaza rezultatul
scrie rezultatul in antet
la receptie aduna cuvinte de 16 biti
rezultatul trebuie sa fie zero.

Pointer urgent (16 biti) - Reprezinta ofset-ul fata de numarul de secventa al datelor ce trebuie trimise urgent.

Optiuni (xx biti) - Au lungimi diferite, apar sau nu in antet.