

CURS 1. MODEL ARHITECTURAL PENTRU SISTEME DESCHISE

Elaborarea standardelor de interconectare a sistemelor deschise a devenit imperios necesara din cel putin trei motive. Mai intai datorita marei diversificari a serviciilor si echipamentelor, retelele au devenit puternic heterogene. Diversificarea este in avantajul utilizatorilor, care pot alege variantele cele mai potrivite aplicatiilor lor, dar devine benefic doar in conditiile in care interconectarea sistemelor si echipamentelor din familii diferite este posibila. In al doilea rand, ea este necesara datorita existentei unor medii de comunicare diverse (retele telex, telefonice comutate, retele digitale cu comutare de circuite sau de pachete, retele cu valoare adaugata, retele digitale cu servicii integrate, retele locale, retele cu comunicare prin radio sau satelit), a caror interconectare este necesara pentru realizarea serviciilor multimediu avansate. In fine, standardizarea este necesara in realizarea protocoalelor unor aplicatii evolute, cum Sunt cele telematice, accesibile unui numar mare de utilizatori. Sunt cuprinse aici aspecte referitoare la conversiile de protocol, conversiile de mediu, multiadresarea, protocoale pentru retele cu valoare adaugata etc.

1.1. Modelul architectural ISO OSI

Preocuparile pentru realizarea unor retele mixte de calculatoare si interconectarea retelelor au determinat elaborarea si adoptarea de catre Organizatia Internationala de Standarde a unui model architectural de referinta pentru interconectarea sistemelor deschise ISO OSI (Open Systems Interconnection). Acesta creeaza cadrul conceptual necesar definirii standardelor internationale privind schimbul de informatie intre terminale, calculatoare, retele, procese. Entitatile enumerate Sunt "deschise" una altele si pot fi combinate neingradit in acelasi sistem, ca urmare a utilizarii doar a standardelor admise. Ne referim in continuare la principiile acestui model, la nivelurile sale functionale si la cele mai cunoscute standarde adoptate pana acum pe baza lui.

1.1.1. Principiile modelului ISO OSI

Modelul architectural are la baza trei elemente: procesele de aplicatie, care realizeaza prelucrarile de date; sistemele de calcul care gazduiesc procesele de aplicatie si care Sunt conectate printr-un mediu de comunicare; conexiunile logice care permit un schimb de informatie uniform intre procesele de aplicatie, indiferent de localizarea acestora in calculatoarele gazda.

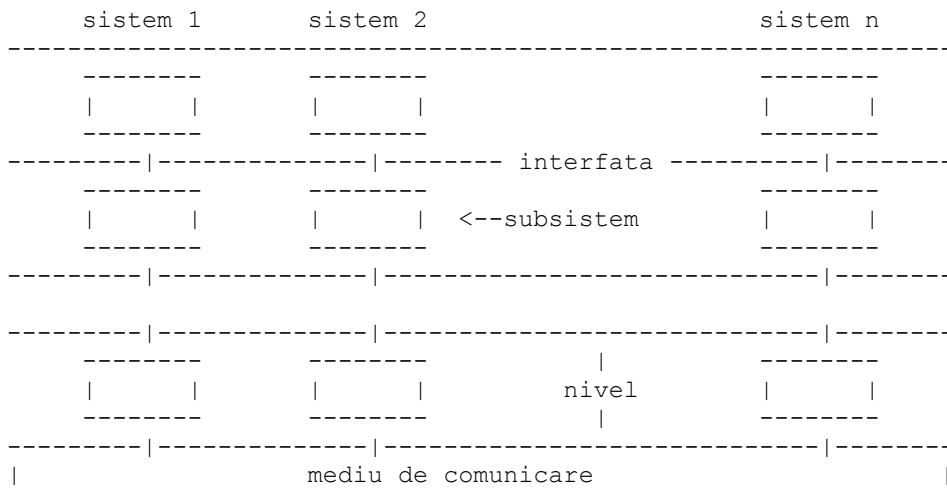


Figura 1.1.

Data fiind complexitatea functiilor utilizate in realizarea cooperarii intre procesele de aplicatie, multimea acestora este structurata, folosindu-se tehnica stratificarii. Fiecare sistem este considerat ca o multime ordonata de subsisteme, totalitatea subsistemelor de acelasi rang formand un nivel al arhitecturii. Subsistemele adiacente comunica prin interfata lor comuna (figura 1.1). Fiecare subsistem este compus din una sau mai multe entitati. Entitatile care apare in aceluiasi nivel se numesc similare.

In orice retea, rolul unui nivel este de a oferi anumite servicii nivelelor superioare, degrevandu-le de detaliile implementarii efective a acestor servicii. Atunci cand o entitate nu poate furniza singura un serviciu, ea coopereaza cu alte entitati similare. Cooperarea este guvernata de protocoale specifice nivelului. Multimea de nivele cu serviciile si protocoalele acestora constituie arhitectura retelei.

1.1.2. Servicii si protocoale

Rolul unui nivel arhitectural este dat de cele doua taieturi care-l delimiteaza de nivelele adiacente, constand in diferenta intre serviciile oferite nivelului adiacent superior si serviciile puse la dispozitie de nivelul adiacent inferior. Fiecare taietura defineste un singur furnizor de servicii, avand in fiecare sistem unul sau mai multe puncte de interactiune, reprezentate de punctele de acces la servicii. Fiecare punct de acces la servicii este asociat unui singur utilizator.

Interactiunile utilizatorului cu furnizorul serviciilor Sunt descrise prin primitivele de serviciu. In figura 1.2. este dat un exemplu de interactiune, in care doi utilizatori stabilesc o conexiune, transfera date, dupa care desfiinteaza conexiunea.

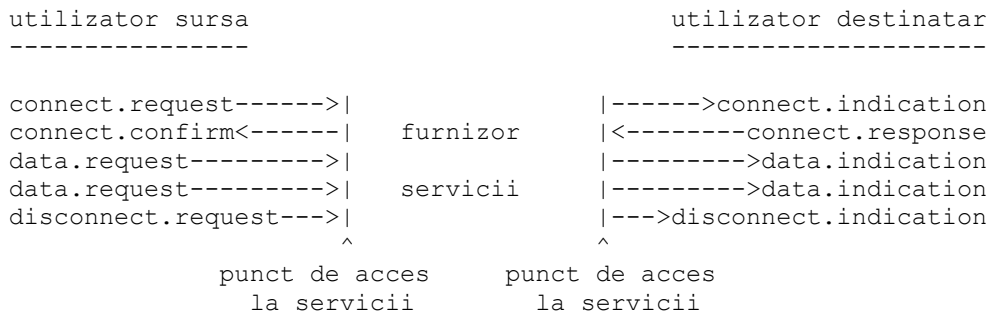


Figura 1.2.

Sunt aratate doua tipuri de servicii: confirmate (conectarea) si neconfirmate (transferul de date si deconectarea). In primul caz, se utilizeaza patru categorii de primitive de serviciu:

>M.ROSE>

serviciu.REQUEST, prin care utilizatorul sursa invoca furnizorul de servicii, serviciu.INDICATION, prin care furnizorul de servicii instiineaza utilizatorul destinatar despre cererea apelantului, serviciu.RESPONSE, prin care utilizatorul destinatar comunica furnizorului raspunsul privind acceptarea sau neacceptarea solicitarii, serviciu.CONFIRMATION, prin care utilizatorul sursa afla raspunsul dat de destinatar.

Serviciile neconfirmate folosesc doua primitive: serviciu.REQUEST si serviciu.INDICATION. Lipsa confirmarii inseamna o lipsa de sincronizare, dar nu si lipsa sigurantei de functionare. In cazul unor defectari, furnizorul serviciilor va instiinta ambii utilizatori.

Un alt serviciu este cel initiat de furnizor. El foloseste primitiva serviciu.INDICATION, data ambilor utilizatori de catre furnizor.

Figura 1.13 sugereaza derularea in timp a primitivelor. Ceea ce nu reiese de aici este faptul ca in modelul serviciilor se folosesc doua cozi, una in fiecare directie. Ca urmare, este posibil ca un utilizator sa initieze mai multe servicii succesiv, fara a astepta intre doua primitive. Primitivele corespunzatoare vor fi livrate celuiilalt utilizator in aceeasi ordine. Exceptie fac primitivele corespunzatoare unor servicii expeditiv, care sunt ordonate intr-o coada separata si pot fi tratate inaintea celor din coada normala, precum si serviciile disruptive, ABORT care au rolul de a goli coada continuand primitive inca nelivrate. Primitivele ABORT pot fi initiate de utilizator sau de furnizor.

Prin conventie, un serviciu este desemnat printr-o combinatie nivel-verb, unde nivel desemneaza unul din cele sapte nivele ale modelului OSI, iar verb denota felul serviciului. De exemplu, N-data.request semnifica o cerere de transfer de date in nivelul retea (network).

<M.ROSE<

Fiecare primitiva este caracterizata de o serie de parametri. De exemplu, la conectare se poate specifica destinatarul (adresa punctului de acces la servicii la care acesta este conectat), tipul serviciului solicitat, lungimea maxima a mesajului etc. La indicare, se poate specifica sursa, tipul serviciului, lungimea maxima a mesajului. Ca raspuns, se poate preciza o contrapro- punere privind lungimea mesajului, furnizorul de servicii urmand sa rezolve conflictul.

Primitivele nu pot fi executate in orice ordine si cu orice valori pentru parametrii caracteristici. La un moment dat, primitivele permise si valorile parametrilor depind de istoria operatiilor precedente. Descrierea primitivelor de serviciu ale unui nivel constituie specificatia serviciilor. Natura acestei specificatii este conceptuala, ea precizand doar efectul fiecărei primitive, acelasi indiferent care ar fi sistemul in care este localizat punctul de acces la servicii. Aceste efecte pot fi realizate prin procedee care difera de la un sistem la altul si a caror descriere este data de specificatia de interfata.

Specificatia serviciilor precizeaza deci comportarea externa a unui nivel. Furnizarea unui serviciu reclama o cooperare intre mai multe entitati similare, conform unui protocol specific. Acesta precizeaza conventiile care guverneaza cooperarea intre entitatile unui nivel, in contextul existentei unor asocieri la diferite puncte de acces la servicii, descriind toate interactiunile posibile dintre entitati. Specificatia unui protocol include:

- lista tipurilor si formatele mesajelor schimbate intre entitai,
- regulile care guverneaza reactia fiecărei entitati la comenzile utilizatorului, la mesajele primite de la ale entitai si la evenimente interne,
- sugestii de implementare si marire a eficientei etc.

Protocolul trebuie incadrat in contextul sau, descrierea fiind completata cu:

- o prezentare generala a rolului nivelului arhitectural,
- specificatia serviciilor furnizate nivelului superior,
- specificatia serviciilor nivelului inferior,
- structura nivelului (entitati si relatii dinte entitati).

1.1.3. Unitati de date

Doua entitai pot schimba intre ele informatii daca aparein aceluiasi nivel si au stabilit intre ele o conexiune, sau apare in unor subsisteme adiacente ale aceluiasi sistem si Sunt atasate la acelasi punct de acces la servicii. Informatiile comunicate reprezinta date, informatii de control sau o combinatie a acestora. In transmiterea sa prin interfata, datele sufera transformările ilustrate in figura 1.3. Aici, unitatea de date de protocol (UDP) consta din informatia de control a protocolului (ICP), care este schimbata intre unitati similare, si din datele schimbate prin interfata cu nivelele adiacente, unitate de date de serviciu (UDS). In figura o UDS este preluata intr-o singura UDP. In cazul segmentarii, o UDS este impartita mai multor UDP, iar in cazul blocarii, mai multe UDS Sunt grupate intr-o singura UDP.

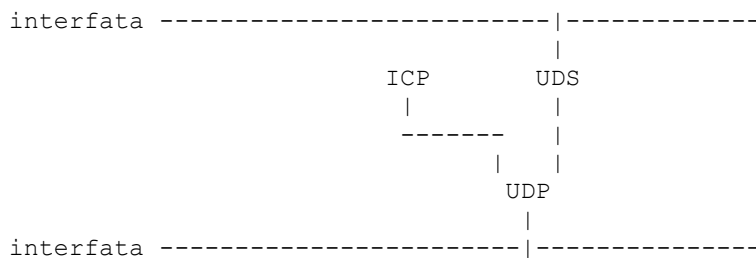


Figura 1.3.

(UDS - unitate de date de serviciu; ICP - informatia de control a protocolului; UDP - unitate de date de protocol)

>M.ROSE>

Aceste conventii nu au nici un impact asupra implementarii. Mecanismul de comunicare intre nivele adiacente se bazeaza pe transferul de mesaje. Fiecare mesaj contine UDP si o informatie suplimentara de tip al mesajului. Oricum, primitivele REQUEST si RESPONSE pot fi asociate unui apel de procedura. Pentru INDICATION si CONFIRMATION exista cel putin doua solutii. Prima se bazeaza pe un mecanism asincron: furnizorul de serviciu apeleaza o procedura specificata de utilizator (un fel de rutina de intrerupere asincrona). A doua este un apel de procedura executat de utilizatorul destinatar, prin care acesta asteapta un anumit interval de timp generarea unei primitive de serviciu. La producerea acesteia, procedura reda controlul utilizatorului, odata cu informatia asupra primitivei date de furnizor.

<M.ROSE<

1.1.4. Servicii orientate si neorientate pe conexiuni

In conformitate cu modelul arhitectural, o varianta de comunicare a mesajelor este cea orientata pe conexiuni. Transferul presupune parcurgerea a trei etape. Prima etapa este stabilirea conexiunii, in care utilizatorii si furnizorul negociaza modul in care serviciul va fi utilizat; in cazul unei intelegeri, conexiunea se considera stabilita.

Aceasta permite punerea de acord a entitatilor in privinta caracteristicilor comunicarii pe care o incep: fiabilitatea schimbului de informatie, limbajul utilizat, regulile de sincronizare etc. Mesajele nu sunt transmise direct intre doua entitati similare dintr-un nivel, ci prin intermediul nivelurilor inferioare. Doar pe nivelul cel mai scazut se face transmiterea directa a mesajelor intre sistemele care gazduiesc entitatile corespondente. Ca urmare, stabilirea unei conexiuni pe un nivel presupune disponibilitatea unei conexiuni pe nivelul inferior.

A doua etapa este transferul de date. Cea de a treia este eliberarea conexiunii, in care se desfiinteaza legatura dintre utilizatori.

In contrast cu acest mod de comunicare se situeaza transferul unei singure unitati de date, cu sau fara confirmare, in afara oricarei conexiuni intre entitatile intre care transferul are loc (serviciu datagrama).

Cateva aplicatii folosind servicii orientate si neorientate pe conexiuni sunt prezentate in tabelul 1.1. In cazul sirului de mesaje, se pastreaza impartirea originara a unitatilor de date.

Tabel 1.1.

	serviciu	exemplu
orientat pe conexiune	sir de mesaje sigur	secventa de pagini
	sir de octeei sigur	conectare la noduri distante
	sir de octeei nesigur	voce digitizata
neorientat pe conexiune	datagrama nesigura	posta
	datagrama sigura	posta recomandata
	cerere/raspuns	interogare baza de date

1.1.5. Nivele arhitecturale

La baza stabilirii nivelelor arhitecturale ale modelului ISO OSI au stat o serie de principii generale, cum ar fi:

- crearea unui numar redus de nivele cu putine interactiuni intre ele;
- alegerea taieturilor dintre nivele in conformitate cu necesitatile de standardizare sau cu standardele deja existente;
- colectarea functiilor inrudite in acelasi nivel;
- crearea posibilitatii de modificare a functiilor unui nivel, fara afectarea celorlalte.

nivele OSI	nivele SNA	nivele DNA
aplicatie	tranzactii	utilizator
prezentare	servicii NAU	aplicatii retea
sesiune	control flux date	control sesiune si
transport	control transmisie	transport
retea	controlul rutei	dirijare
leg. date	legatura de date	legatura de date
fizic	fizic	fizic

Figura 1.4.

Ele au condus la un model cu sapte nivele functionale, ale caror principale caracteristici sunt prezentate in continuare. Figura 1.4 prezinta comparativ nivelele OSI cu cele ale arhitecturilor SNA (IBM) si DNA (Digital). Folosit drept cadru pentru elaborarea de standarde, modelul este util si pentru sistematizarea predarii retelelor de calculatoare. Modelul este adoptat drept cadru de prezentare in mai toate cartile si manualele ce au ca subiect retelele. Descriem in continuare rolul diverselor niveluri arhitecturale, enumerand totodata problemele mai importante ce vor fi abordate in cursul de fata.

Nivelul fizic asigura transmiterea datelor binare codificate intre diferite sisteme, prin mediul fizic de interconectare, pastrand ordinea bitilor fara a garanta corectitudinea transmisiei. Aspectele principale se refera la:

- modul de codificare a datelor binare;
- modul de transmisie (duplex sau semiduplex);
- modul de stabilire si desfiintare a conexiunilor fizice.

Transmisia este transparenta, semnificatia datelor sau modul lor de grupare in unitati logice fiind neimportant.

Nivelul legaturii de date trateaza erorile de transmisie produse la nivelul fizic, realizand o comunicare corecta intre doua noduri adiacente. Mecanismul utilizat in acest scop este impartirea sirului de biti in cadre, carora le sunt adaugate informatii de control (coduri de verificare, numere de secventa etc). Cadrele sunt transmise individual putand fi verificate si confirmate de catre

receptor. Alte functii ale nivelului se refera la: controlul fluxului datelor (astfel incat transmitatorul sa nu furnizeze date mai rapid decat le poate accepta receptorul) si gestiunea legaturii (stabilirea legaturii, controlul schimbului de date si desfiintarea legaturii).

In transferul datelor se utilizeaza doua strategii: una la care transmiterea fiecarui cadru se face dupa confirmarea cadrului precedent (protocoale start-stop) si alta la care confirmarea se poate face simultan pentru mai multe cadre (protocoale cu fereastra glisanta). Protocoalele legaturii de date mai pot fi clasificate dupa lungimea cadrului, in protocoale orientate pe caracter (lungimea in biti fiind multiplu de 8) si protocoale orientate pe bit (cu o lungime oarecare de cadre).

Controlul fluxului este realizat uneori prin transmiterea unor cadre de control speciale, de "permisiune", de la receptor la sursa, dar mai frecvent prin mecanismul confirmarii cadrelor. Astfel, confirmarea receptiei unui anumit numar de cadre poate insemna implicit o invitatie pentru transmiterea cadrelor urmatoare.

Modelele algoritmice ale celor doua categorii de protocoale constituie "miezul" prezentarii. Pornind de la ele sunt discutate si alte modele (tranzitionale, hibride) ale protocoalelor si sunt analizate proprietatile lor.

Un loc aparte este dedicat prezentarii limbajului LOTOS, de specificare a protocoalelor. LOTOS (Language Of Temporal Ordering Specification) este un limbaj aplicabil in egala masura sistemelor distribuite si concurente. In acest limbaj, un sistem este vazut ca o multime (ierarhie) de procese care interactioneaza si schimba date intre ele sau cu mediul. Obiectivul realizarii sale este obtinerea unor specificatii standard OSI, care sa fie:

- neambigue, precise, complete, independente de implementare;
- intelese cu usurinta de cel care le utilizeaza, implementeaza sau testeaza;
- folosite ca baza pentru verificarea si validarea standardelor, precum si pentru testele de conformanta ale implementarilor lor.

O specificatie LOTOS permite analiza riguroasa a unui protocol, chiar cand acesta se afla la inceputul dezvoltarii sale, reducand propagarea erorilor .

Tehnica de descriere nu este legata de logici temporale, asa cum ar lasa sa se inteleaga denumirea limbajului, ci de algebre de procese, mai precis de CCS (A Calculus of Communicating Systems, introdus de Milner) si de TCSP (A Theory of Communicating Sequential Processes elaborata de Brookes, Hoare si Roscoe). Controlul programelor este descris prin expresii algebrice numite comportamente sau expresii de comportare. Comunicarea si sincronizarea se fac prin rendez-vous, fara memorie partajata.

LOTOS include o a doua componenta, pentru descrierea structurilor de date si a expresiilor de valoare. In acest scop, el foloseste formalismul ADT (Abstract Data Types) pentru manipularea tipurilor de date. Acest formalism, preluat din limbajul ACT-ONE, permite descrierea obiectelor prin indicarea proprietatilor pe care trebuie sa le satisfaca datele si operatiile in orice realizare, fara a impune restrictii de implementare ne-necesare.

Nivelul retea asigura dirijarea unitatilor de date intre nodurile sursa si destinatar, trecand eventual prin noduri intermediare. Decizia este luata astfel incat sa nu existe in acelasi timp legaturi supraincercate si legaturi neutilizate, evitandu-se deci congestiunea retelei. O alta functie importanta a nivelului retea este cea de interconectare a retelelor cu arhitecturi diferite.

In cazul retelelor cu comutarea pachetelor, serviciile mentionate se bazeaza pe utilizarea circuitelor virtuale (comutate sau permanente). Un circuit virtual (vezi figura 1.5) "parcurge" mai multe comutatoare de pachete intre doi utilizatori si permite schimbul unor secvente de pachete de date, simultan in

ambele sensuri. El pastreaza ordinea pachetelor si permite controlul fluxului datelor, astfel incat comunicarea este posibila si intre terminale sau calculatoare de viteze diferite. Circuitele comutate se stabilesc si se desfiinteaza dinamic, la cererea utilizatorilor, la fel ca circuitele telefonice comutate dintre abonati. Spre deosebire de acestea, un circuit virtual nu ocupa in exclusivitate canalele fizice de comunicatie, doua sau mai multe circuite virtuale putand utiliza acelasi canal. In plus, comutatoarele de pachete memoreaza temporar datele aflate in tranzit, spre deosebire de comutatoarele din retelele telefonice care nu au capacitati de memorare.

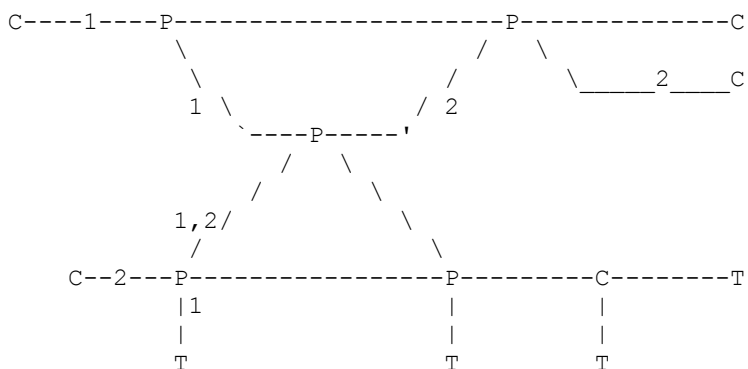


Figura 1.5.

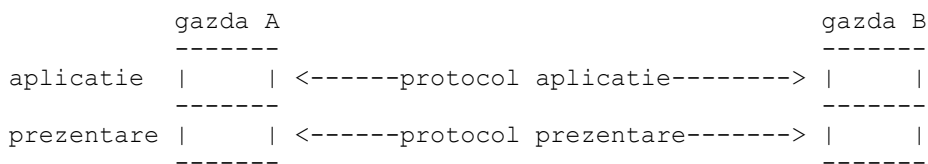
(T - terminal; C - calculator; P - comutator de pachete; 1 si 2 - circuite virtuale)

O alta posibilitate este transmiterea individuala a pachetelor, dupa modelul transmiterii scrisorilor in serviciul postal. La aceasta varianta, numita datagrama, stabilirea traseului se face pentru fiecare pachet in parte, doua pachete diferite care au aceeasi sursa si aceeasi destinatie putand fi dirijate pe trasee diferite.

In ambele variante, retea poate transmite expeditiv pachetele prioritare, dar nu asigura intotdeauna receptia datelor expeditive sau normale.

In curs sunt prezentati algoritmi de dirijare, precum si alte probleme legate de: adresare, fragmentarea si reasamblarea datelor, interconectarea retelelor, interfata X.25. O atentie deosebita se acorda proiectarii topologice a retelelor de calculatoare, subiect ce este prezentat gradat, in ordinea crescanda a complexitatii problemelor de proiectare: stabilirea capacitatii, stabilirea fluxului, stabilirea topologiei.

Nivelul transport realizeaza o comunicare sigura intre doua calculatoare gazda, detectand si corectand erorile pe care nivelul retea nu le trateaza. El este important nu numai prin pozitia pe care o ocupa, la mijlocul ierarhiei de nivele, dar si prin functia sa, de a furniza nivelelor superioare o interfata independenta de tipul retelei utilizate. Functiile sunt realizate de entitati situate in sistemele gazda, fara concursul altor entitati similare intermediare, motiv pentru care se numesc functii "capat la capat" (capetele fiind cele doua entitati de transport corespondente). Pentru a sublinia mai bine acest lucru, prezentam in figura 1.6 repartizarea nivelelor arhitecturale in nodurile gazda si in nodurile de comutare.



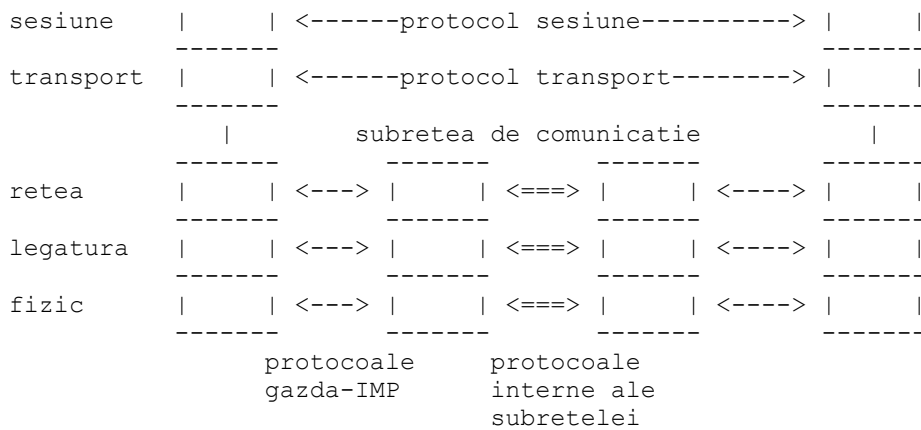


Figura 1.6.

Pentru primele trei subniveluri, protocoalele se refera la legatura dintre terminal (gazda) si subretea (IMP - Interface Message Processor), sau intre nodurile subretelei, constituind astfel doua categorii distincte de protocoale. Incepand cu nivelul transport, protocoalele sunt capat la capat, entitatile acestora neavand corespondente in subretea.

In functie de caracteristicile traficului generat, utilizatorii nivelului transport (entitatile sesiune) pot cere, la stabilirea unei conexiuni de transport, o anumita calitate a serviciului furnizat. Calitatea este exprimata prin valorile unor parametri cum sunt: productivitatea (cantitatea de date transferata intr-o unitate de timp), intarzierea de transmisie, durata de stabilire a unei conexiuni, erorile netratate, probabilitatea de esec la tratarea erorilor, posibilitatea de transmitere expeditiva a datelor etc.

Atingerea acestor performante depinde in mare masura de tipul subretelei utilizate, in functie de care nivelul transport trebuie sa realizeze un numar mai mare sau mai redus de functii. Astfel, in cazul unor subretele fara erori, nivelul transport nu trebuie sa realizeze corectii ale transmisiei, in timp ce in cazul unor erori, semnalate sau nesemnalate de retea, aceste corectii sunt necesare. Este deci normal sa existe mai multe clase de servicii de transport si, corespunzator, mai multe clase de protocoale.

Una din functiile importante ale nivelului transport este multiplexarea conexiunilor. Multiplexarea "in sus" consta in utilizarea unei conexiuni retea, ca suport al mai multor conexiuni de transport. Ea urmareste utilizarea mai eficienta a subretelei, in cazul in care traficul pe fiecare conexiune de transport este redus. Multiplexarea "in jos" consta in utilizarea mai multor conexiuni de retea pentru o singura conexiune de transport. Ea urmareste marirea vitezei de transmitere a datelor, in cazul unui flux ridicat.

La fel de important este controlul fluxului datelor, aspect intalnit si la nivelul legatura de date. Complexitatea acesteia in contextul nivelului transport este determinata de numarul mult mai mare de conexiuni gestionate si de intervalele de timp mult mai mari, in care mesajele trebuie pastrate pentru eventuale retransmiteri. Ea reclama solutii diferite de cele adoptate la nivelul legaturii de date, bazate pe decuplarea mecanismului de control al fluxului de cel de confirmare a receptiei mesajelor.

In fine, o alta problema se refera la lungimea unitatilor de date, care se poate modifica prin blocare (colectarea mai multor mesaje intr-o singura unitate de date de protocol) sau segmentare (impartirea unui mesaj in mai multe unitati de date de protocol).

Desi nivelul transport asigura conexiuni utilizabile simultan in ambele sensuri (duplex), multe aplicatii necesita o coordonare a dialogului, in care doar unul din interlocutori poate transmite la un moment dat. Controlul dialogului este

una dintre functiile nivelului sesiune. Acesta mai permite impartirea sirului de mesaje in grupuri numite activitati, care pot fi gestionate independent unele de altele, precum si includerea in sirul de mesaje a unor puncte de sincronizare, de la care se poate face reluarea operatiilor in cazul aparitiei unor erori.

Aceste functii au la baza utilizarea unui mesaj special, numit jeton (token), care poate fi trecut de la un utilizator la altul si a carui posesie ofera detinatorului anumite privilegii: de a transmite date, de a stabili puncte de sincronizare, de a stabili inceputul unei activitati etc. Nivelul sesiune ofera toate serviciile pentru gestiunea jetoanelor, lasand la latitudinea utilizatorilor semnificatiile asociate acestora.

Modelul client-server constituie unul din principalele subiecte discutate in aceasta parte a cursului.

Nivelul prezentare realizeaza transformari ale reprezentarii datelor, astfel incat sa se pastreze semnificatia lor, rezolvandu-se totodata diferentele de sintaxa. Functiile principale se refera la codificarea standard a datelor transmise intre calculatoare cu conventii de reprezentare diferite, la comprimarea textelor, precum si la criptarea/decriptarea acestora in vederea protectiei si securitatii lor.

Nivelul aplicatie, cel mai inalt nivel al arhitecturii, are rolul de fereastră de comunicare prin care se fac toate schimburile de date între utilizatori. Fiind nivelul care furnizeaza servicii direct aplicatiilor, el cuprinde toate functiile pe care acestea le pot solicita:

- identificarea utilizatorilor cooperanti, autentificarea lor si determinarea disponibilitatii acestora,
- stabilirea calitatii serviciului,
- sincronizarea aplicatiilor cooperante si selectarea modului de dialog,
- stabilirea responsabilitatilor pentru tratarea erorilor,
- identificarea constrangerilor asupra sintaxei datelor,
- transferul informatiei.

Dintre acestea, o parte sunt comune tuturor protocoalelor de aplicatie si sunt realizate prin proceduri care constituie elementele de serviciu comune aplicatiilor. Ele includ, de exemplu, functii de stabilire si terminare a unei asociatii între procese de aplicatie. In afara lor, nivelul aplicatie include o serie de servicii de interes general, cum ar fi cele de terminal virtual, fisier virtual, posta electronica, acces la noduri indepartate etc. care sunt specifice anumitor clase de aplicatii.

Terminalele folosite in retelele de calculatoare au caracteristici diverse, referitoare la: modul de afisare (defilare sau pagina), setul de caractere utilizat, semnificatia unor caractere de control, lungimea liniei sau dimensiunea ecranului, modul de utilizare a cursorului si altele. Pentru a preveni utilizarea lor defectuoasa de catre programele de aplicatie, se utilizeaza protocoale care mascheaza particularitatile terminalelor fata de utilizatori. Un astfel de protocol face corespondenta între terminalul real si un terminal virtual, ale carui caracteristici se conformeaza anumitor standarde. Probleme similare apar in cazul fisierelor.

Aceasta diferentiere de functii conduce la impartirea nivelului aplicatie in subniveluri, doua dintre ele fiind deja mentionate anterior. Un al treilea subnivel ar corespunde functiilor strict legate de o anumita aplicatie, cum ar fi schimbul de documente informatizate. Evident, chiar si in cazul acestor functii, modelul OSI se refera la acele aspecte care implica transferul de date, deci la formatul datelor si la procedurile de transfer.