

Lucrarea 1

Modelul OSI; Conversii de date

1. Modelul de referinta OSI

In anii 80 cand ideea de retea a castigat din ce in ce mai multi adepti, inginerii au inteles ca fara o standardizare a tehnologiei retelelor dezvoltarea acestui concept este condamnata la esec. Specialistii de la Organizatia Internationala pentru Standardizare (ISO) au cercetat o multime de modele pentru diferite retele pentru a o alege pe cea care oferea cea mai buna interconectare. Astfel, in 1984 au creat un model de retea care sa poata ajuta companiile sa dezvolte retele capabile de a lucra impreuna. Modelul a fost numit modelul de referinta OSI (**Open Systems Interconnection Basic Reference Model**) si a devenit disponibil imediat.

ISO a decis crearea unui model care utilizeaza nivele (layer en.), fiecare nivel ocupandu-se cu alta actiune, toate fiind insa in legatura unul cu altul pentru ca este imposibila realizarea comunicarii fara parcurgerea tuturor pasilor necesari. In reprezentarea de mai sus exista numai patru pasi, dar ISO a ales o schema mult mai detaliata. Nivelele OSI (nu ISO - International Standards Organization) au o multime de avantaje; unul dintre acestea fiind faptul ca pot fi invatate usor. Nu exista pe lume inginer de retea profesionist care sa nu stie ceva despre nivelele OSI. Este imposibil sa dezvolti o retea fara sa le cunosti. Aceste nivele constituie baza unei retele. Modelul de referinta OSI va permite sa vedeti care sunt functiile retelei la fiecare nivel. Modelul de referinta OSI este cadrul care permite intelegerea felului in care datele sunt transmise printr-o retea.

In modelul OSI exista sapte nivele diferite, fiecare avand o functie specifica. Vom vedea in cele ce urmeaza ca TCP/IP simplifica modelul de referinta OSI, dar pentru inceput trebuie sa aflam cate ceva despre OSI.

1.1.Nivelele OSI

In imaginea de mai jos este realizata o reprezentare a nivelelor OSI:



Dupa cum puteti vedea, OSI este un model stiva. Acest model simplifica evolutia, deoarece orice schimbare a unui nivel nu ii afecteaza si pe ceilalti. De asemenea, standardizeaza reseaua si permite interoperabilitatea si modularizarea componentelor fabricate de diversi producatori. Haideti sa parcurgem nivelele existente, incepand cu cel mai apropiat de utilizatori.

7. Nivelul "Aplicatie" ofera servicii de retea pentru aplicatiile utilizatorilor. Avand in vedere ca se afla in varful stivei, nu ofera servicii pentru alte nivele ci pentru aplicatiile exterioare. Nivelul de aplicatii verifica partenerii de comunicare, integritatea si sincronizarea datele. Ganditi-va la ceea ce fac browserele.

6. Nivelul "Prezentare" ofera date pentru nivelul de aplicatii. Se ocupa cu "prezentarea datelor ". Cu alte cuvinte, verifica daca datele sunt intr-un format care poate fi inteles de ambele parti implicate in comunicare. Daca nu, le converteste la un format comun. De asemenea, negociaza pentru acest format comun. Este ca un translator intr-o conversatie.

5. Nivelul "Sesiune" stabileste, administreaza si termina sesiunile dintre partile implicate in comunicare. Dupa cum probabil ati ghicit, ofera date nivelului de prezentare. Ajuta doua gazde sa isi sincronizeze dialogul si administreaza schimbul de date. Ofera resursele pentru un transfer de date eficient, raportarea erorilor proprii sau ale nivelelor superioare.

4. Nivelul "Transport" se ocupa cu segmentarea datelor. Intai segmenteaza datele din sistemul gazdei care transmite si apoi le reassembleaza la destinatie si le transmite sistemului. Nivelul de transport poate fi perceptut ca un nivel de granita intre nivelele "gazda" si asa numitele nivele media. Nivelele de transport incearca sa asigure cea mai buna comunicare utilizand tehnici de depistare a erorilor de transmisie si de recuperare si control al fluxului. Mentine nivelele superioare departe de problemele de implementare.

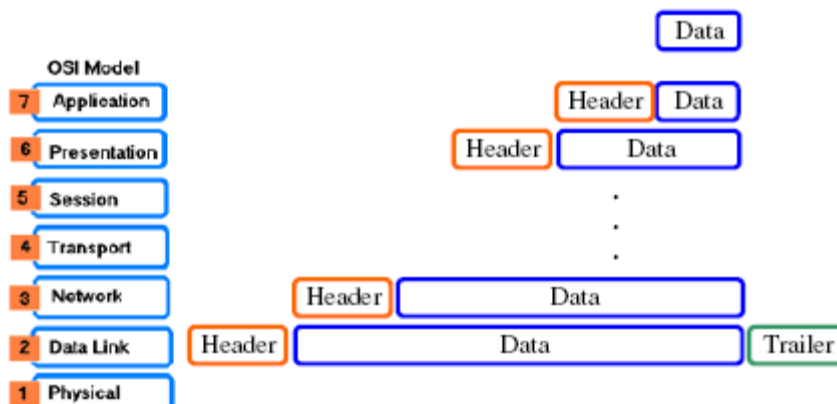
3. Nivelul "Retea" este cel mai complex nivel care conecteaza si asigura ruta potrivita dintre doua gazde aflate pe doua retele total diferite. Nivelul 3 se ocupa cu selectarea rutelor, rutarea si adresarea.

2. Nivelul "Legatura" transmite datele printr-o legatura fizica. Aceasta este partea fizica a unei retele care se ocupa cu adresarea "fizica", topologia retelei, accesul la retea, sesizarea erorilor, transportul cadrelor cerute si controlul fluxului.

1. Nivelul "Fizic" este nivelul de baza care asigura serviciile fizice. Nivelul fizic defineste specificatiile electrice, mecanice, procedurale si functionale pentru activarea, mentinerea si dezactivarea legaturii fizice intre sistemele finale. Este cel mai aproape de media si semnale. Totodata el controleaza specificatiile mediului de transmisie, controleaza tensiunea, semnalele, viteza de transmisie, distantele, conectorii.

1.2.Incapsularea datelor

Pentru a transmite un mesaj in retea se foloseste un procedeu numit incapsularea datelor.

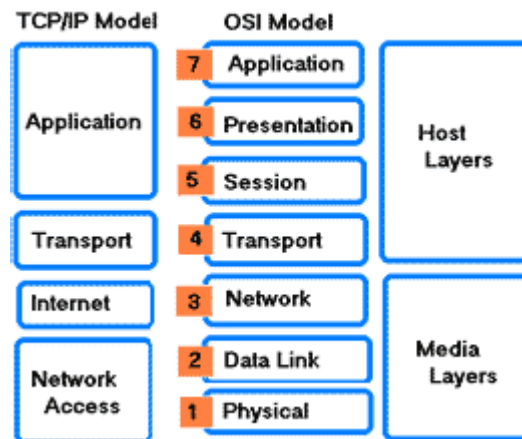


In cadrul unei retele datele sunt transmise de la o gazda la alta si fiecare nivel OSI comunica cu nivelul corespondent (corespondentul sau) de la destinatie. Forma de comunicare in cazul in care fiecare nivel realizeaza un schimb de date (asa numitul protocol data units - PDU) cu nivelul aflat la destinatie poarta numele de comunicare corespondent-corespondent (peer to peer). In cadrul unei retele fiecare nivel depinde de nivelul aflat dedesubt. Nivelul aflat cel mai jos incapsuleaza PDU-ul de la nivelul superior in campul sau de date, ii adauga headerele si trailererele proprii, iar datele trec la nivelul urmator. De exemplu nivelul 4 adauga mai multe informatii la datele provenite de la nivelul 5 si le grupeaza intr-un segment. Nivelul 3 (retea) trebuie sa transmita datele prin retea. Le ataseaza un header creand un PDU al nivelului 3. In acest moment headerul contine informatii logice, dar Nivelul 2 incapsuleaza intr-un cadru informatia despre adresa fizica necesara pentru ca transferul sa fie realizat. Nivelul de "legatura a datelor" asigura "serviciul" nivelului de retea prin incapsularea informatiilor acestuia din urma intr-un cadru. Nivelul fizic asigura de asemenea "serviciul" nivelului de "legatura a datelor". Nivelul fizic codeaza cadrul intr-un model de 1 si 0 (biti) pentru transmisia prin mediu la nivel fizic.

1.3. Modelul de referinta TCP/IP

Am discutat despre modelul OSI, care are cateva avantaje importante. Cu toate acestea, standardul TCP/IP este folosit in acest moment pentru transmisiile de date din cea mai mare retea existenta - Internetul. Din punct de vedere istoric, modelul de referinta TCP/IP a fost creat de Ministerul Apararii din SUA pentru a deveni retea suprema - o retea capabila sa supravietuiasca in orice conditii, chiar si intr-un razboi nuclear. Era extrem de important sa fie creata o retea capabila sa opereze cu o infrastructura distrusa in proportie de peste 90%, fara sa aiba vreo importanta starea fizica a unor anumite segmente ale retelei. Imaginati-va ca Internetul are aceasta capacitate - doua gazde pot comunica chiar daca aproape toate serverele dintre ele nu au putut sa le raspunda cererii; sunt disponibile cai redundante. La urma urmei un uptime de 100% este imposibil. Chiar si un uptime de 99,9%, cu care se lauda multe companii de hosting este foarte greu de realizat. Asa ca cei de la US DoD (Departament of Defence) doreau o retea inteligenta.

Atentie: Modelul de referinta TCP/IP are patru nivele si, desi exista doua nivele cu acelasi nume ca la modelul OSI, nu le confundati cu acelea pentru ca fiecare nivel are functii total diferite.



4. Nivelul "Aplicatie" - Include primele trei nivele aflate in varful modelului OSI: de Aplicatie, Presentare si Sesiune. Se ocupa cu procesarea logica ca reprezentarea, codarea, dialogul. La fel ca in modelul OSI nivelul pentru aplicatii pregateste datele pentru nivelul urmator.

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 reassembleaza 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.

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).

1. Nivelul "Acces la Retea" - La fel ca Nivelul de Legatura a Datelor si ca cel Fizic din modelul OSI se ocupa in principal cu toate aspectele legaturilor fizice.

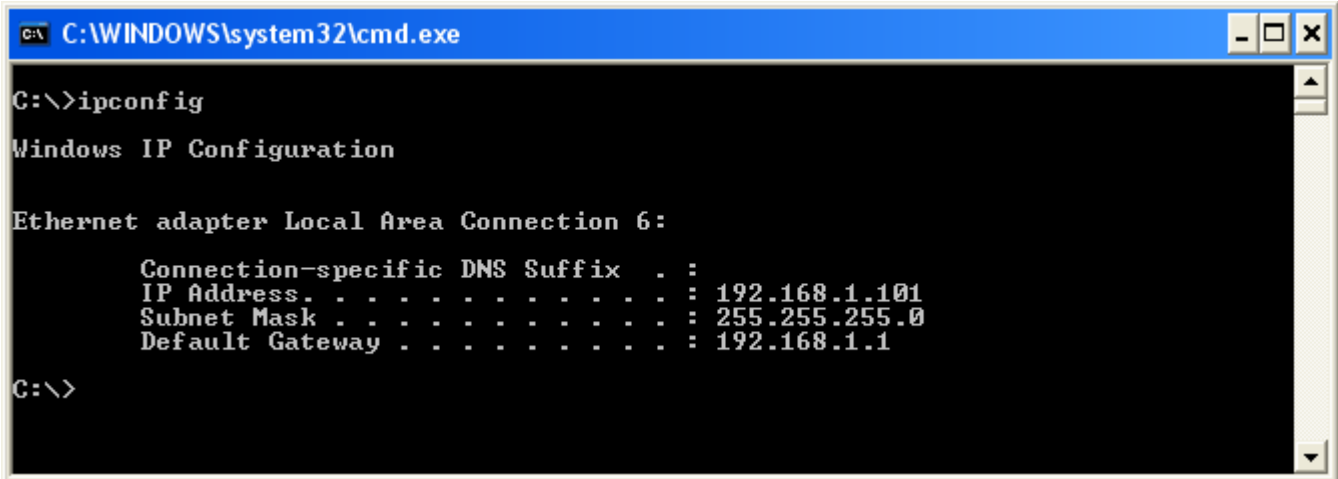
2. Adresa IP, Port, Socket

2.1. Cum este identificat un calculator în retea ?

Orice calculator gazda conectat la Internet este identificat în mod unic de adresa sa IP (IP este acronimul de la Internet Protocol). Aceasta reprezinta un numar reprezentat pe 32 de biti, uzual sub forma a 4 octeti, cum ar fi de exemplu: 193.226.26.231 si este numit adresa IP numerica. Corespunzatoare unei adrese numerice exista si o adresa IP simbolica, cum ar fi www.google.com. Mai multe informatii despre acestea intr-un laborator viitor.

Pentru a afla adresa IP curenta a unui calculator scrieti urmatoarea comanda in consola sistemului:

ipconfig



```
C:\WINDOWS\system32\cmd.exe

C:\>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection 6:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .                : 192.168.1.101
    Subnet Mask . . . . .              : 255.255.255.0
    Default Gateway . . . . .          : 192.168.1.1

C:\>
```

Rezultatul acestei comenzi este o lista a adaptoarelor de retea existente si, pentru fiecare adaptor de retea, informatiile necesare pentru conectarea la reseaua logica:

- Adresa IP (IP Address) – adresa unica a echipamentului in cadrul retelei
- Masca (Subnet Mask) – masca folosita pentru a delimita reseaua locala
- Gateway (Default Gateway) – adresa echipamentului care face legatura retelei locale cu o retea externa

Atunci cand un echipament trimite un pachet catre un alt echipament identificat cu o adresa IP se va verifica daca destinatia apartine retelei locale (SI logic intre adresa IP a sursei si masca este identic cu SI logic intre adresa IP a destinatiei si aceiasi masca). Daca Echipamentul apartine retelei locale, datele se trimit direct catre acesta. Altfel, datele sunt trimise catre Gateway care le va transmite mai departe.

2.2. Ce este un port ?

Un calculator are în general o singura legatura fizica la retea. Orice informatie destinata unei anumite masini trebuie deci sa specifice obligatoriu adresa IP a acelei masini. Insa pe un calculator pot exista concurrent mai multe procese care au stabilite conexiuni în retea, asteptând diverse informatii. Prin urmare datele trimise catre o destinatie trebuie sa specifice pe lângă adresa IP a calculatorului si procesul catre care se îndreapta informatiile respective. Identificarea proceselor se realizeaza prin intermediul porturilor.

Un port este un numar de 16 biti care identifica în mod unic procesele care ruleaza pe o anumita masina. Orice aplicatie care realizeaza o conexiune în retea va trebui sa ataseze un numar de port acelei conexiuni. Valorile pe care le poate lua un numar de port sunt cuprinse între 0 si 65535 (deoarece sunt numere reprezentate pe 16 biti), numerele cuprinse între 0 si 1023 fiind însa rezervate unor servicii sistem si, din acest motiv, nu trebuie folosite in aplicatii

3. Adrese socket

Pentru a fi independent de platformă, socket permite utilizarea mai multor **familii de adrese**, o astfel de familie definind un stil de adresare. Toate host-urile din aceeași familie de adrese înțeleg și folosesc aceeași schemă pentru adresarea capetelor socket-ului. Familiile de adrese sunt identificate printr-un număr întreg, numele acestora începând cu AF (Adress Family):

- AF_INET: definește adresarea în domeniul Internet, fiind și cea mai importantă familie;
- AF_UNIX: reprezintă sistemul local (protocoale interne UNIX);
- AF_NS: pentru protocoale Xerox NS;
- AF_IMPLINK: pentru IMP(Interface Message Processor).

Generic, structura unei adrese este următoarea:

```
struct sockaddr{
    u_short sa_family; //valoare AF_xxx
    char sa_data[14]; //dependent de familie
}
```

unde,

sa_family: indică familia de adrese folosită;
sa_data: interpretat în funcție de familie.

OBS. fiecare familie de adrese are definită structura proprie de adresă care se suprapune peste cea generică în momentul transmiterii ei către nucleu.

Adresele din familia AF_INET sunt de tip struct sockaddr_in și sunt definite în <Winsock2.h> astfel:

```
struct in_addr {
    union {
        struct { u_char s_b1,s_b2,s_b3,s_b4; } S_un_b;
        struct { u_short s_w1,s_w2; } S_un_w;
        u_long S_addr;
    } S_un;
};
```

```

struct sockadrr_in{
    short sin_family;           //valoarea AF_INET
    u_short sin_port;          //numarul de port
    struct in_addr sin_addr;    //adresa IP
    caht s in_zero[8];         //nefolosit
}

```

4. Functii C folosite pentru manipularea datelor

4.1. Conversii ale reprezentărilor întregilor

La nivelul comunicațiilor prin socket, întregii sunt tratați ca fiind fără semn. Este definită o reprezentare „universală” a lor numită **reprezentare de rețea**. Folosind fișierul header <Winsock2.h> conversia întregilor din reprezentarea de rețea se face cu ajutorul următoarelor funcții:

```

u_long htonl (u_long I)      //host to network long
u_short htons (u_short I)    //host to network short
u_long ntohl (u_long I)     //network to host long
u_short ntohs (u_short I)   //network to host short

```

Funcțiile transformă argumentul I și întorc întregul convertit astfel:

- funcția `htonl` convertește un întreg lung local I într-un întreg lung reprezentat rețea;
- funcția `htons` convertește un întreg scurt local I într-un întreg scurt de rețea;
- funcția `ntohl` convertește un întreg lung de rețea I într-un întreg lung local;
- funcția `ntohs` convertește un întreg scurt de rețea I într-un întreg scurt local.

4.2. Rutine de manevrare a șirurilor de octeți

În cele ce urmează prezint pe scurt patru funcții ce efectuează o serie de operații utile asupra șirurilor de octeți:

```

memcpy (char *d, char *s, int n)
    //copiază n octeți de la adresa s la adresa d

memset (char *d, int v, int n)
    //depune începând cu adresa d, n octeți cu
    //conținut identic, valoarea v

strcpy (char *d, char *s)
    //copiază un șir de octeți de la adresa s la
    //adresa d

int strcmp (char *a, char *b)
    //compară șirurile a și b și întoarce o valoare
    //negativa daca a<b, pozitiva daca a>b si 0 daca
    //a=b

```

5. Cerinte laborator

Scrieti o functie in C (C++) pentru conversia unui numar scris pe 4 octeti (de ex. un numar de tip `int` sau `long` daca folositi Visual C++ 2005) din format Little Endian in format Big Endian precum si o functie care face conversia inversa (care e cea mai scurta varianta a acestei functii inverse ?). Scrieti un program in C (C++) care citeste 5 numere scrise pe 4 octeti dintr-un fisier de intrare (de preferat text), le afiseaza pe ecran in format hexa, le converteste in format Big Endian, afisaza rezultatul conversiei pe ecran si apoi scrie rezultatul intr-un fisier de iesire. Pentru conversie folositi mai intai functia creata anterior si apoi una din functiile existente in biblioteca *winsock2*