UNIVERSITATEA PEDAGOGICĂ DE STAT
„ION CREANGĂ” DIN CHIŞINĂU

Olga Chirchina Zinaida Ghilan

REŢELE DE CALCULATOARE
(Suport didactic)

Chişinău 2014

CZU 004.7(075.8)
C43

RECENZENŢI:
Pavel Dorin, dr., conf. univ., Universitatea de Stat din Tiraspol (cu sediul la Chișinau)
Căpățină Gheorghe, dr., inginer, prof. univ., Universitatea de Stat din Moldova

Descrierea CIP a Camerei Naţionale a Cărţii.

Chirchina Olga
Reţele de calculatoare (Suport didactic)/ Olga Chirchina, Zinaida Ghilan : Univ. Pedagogică de Stat „Ion Creangă" din Chişinău.- Chişinău:S.n. 2014 (Tipogr. „Garomont-Studio").-222p.
Referinţe bibliogr.: p. 200-204 (111 tit.). – 15 ex.
ISBN 978-9975-115-38-4.

CZU 004.7(075.8)
C43

Cuprins

Introducere 7
Lista abrevierilor 9
Capitolul 1. Principii, clasificări şi modele de referinţă ale reţelor de calculatoare 14
1.1. Principii şi noţiuni fundamentale în transferul de date prin reţele de calculatoare 15
1.1.1. Componentele generale ale unei reţele 16
1.1.2. Încapsulare 17
1.2. Clasificarea reţelelor de calculatoare 18
1.3. Topologii ale reţelelor de calculatoare 27
1.4. Protocoale de comunicaţie în reţea 27
1.5. Modelul de referinţă *ISO-OSI* 29
1.5.1. Nivelul 7: Aplicaţie (*Application Layer*) 31
1.5.2. Nivelul 6: Prezentare (*Presentation Layer*) 32
1.5.3. Nivelul 5: Sesiune (*Session Layer*) 32
1.5.4. Nivelul 4: Transport (*Transport Layer*) 33
1.5.5. Nivelul 3: Reţea (*NetworkLayer*) 34
1.5.6. Nivelul 2: Legătură de date (*Data-Link Layer*) 35
1.5.7. Nivelul 1: Fizic (Physical Layer) 37
1.6. Modelul de referinţă *TCP/IP* 37
1.7. Comparaţia modelelor *ISO-OSI* şi *TCP/IP* 39
Capitolul 2. Tehnici şi medii de transmisie la nivel Fizic 41
2.1. Funcţiile definite în cadrul nivelului Fizic 42
2.1.1. Funcţiile de bază ale nivelului Fizic 42
2.1.2. Subnivelurile nivelului Fizic 44
2.2. Semnal şi zgomot în sistemele de comunicaţie 44
2.2.1. Semnale analogice 45
2.2.2. Semnale digitale 46
2.2.3. Fenomene care pot influenţa calitatea semnalului 46
2.3. Tehnica transmiterii semnalului 50
2.3.1. Multiplexare 50
2.3.2. Transmisia *baseband* şi *broadband* 51

2.4. Medii de transmisie 52
2.4.1. Fire de cupru 53
2.4.2. Fibra optică 58
2.4.3. Comparaţie între fibrele optice şi firele de cupru 60
2.4.4. Sistemele fără fir 61
Capitolul 3. Protocoale și tehnici de acces la nivelul Legătură de date. Sisteme de telefonie mobilă 66
3.1. Protocoale de acces la mediu de transmisie 67
3.2. Tehnici de acces la mediul de transfer în reţele locale (*LAN*) 69
3.2.1. Scheme de adresare folosite în telecomunicaţii 70
3.2.2. Structura generică a unui cadru de nivel 2 71
3.2.3. Protocoale de comunicaţie la nivelul Legătură de date 74
3.3. Tehnici de acces pentru reţele largi (*WAN*) 83
3.3.1. Comutaţie de circuite (*Circuit-switched*) 84
3.3.2. Comutaţie de pachete (*Packet-switched*) 85
3.3.3. *Cell-switched*. *ATM* (*Asyncronous Transfer Mode*) 86
3.3.4. *Dedicated digital*. Multiplexare în telefonie 87
3.3.5. *Analog services* 91
3.4. Sisteme de telefonie mobilă 94
3.4.1. *AMPS* (*Advanced Mobile Phone System*) 94
3.4.2. *D-AMPS* (*Digital Advanced Mobile Phone System*) 96
3.4.3. *GSM* (*Global System for Mobile Communications*) 97
3.4.4. *CDMA* (*Code Division Multiple Access*) 98
3.4.5. *EDGE* (*Enhanced Data Rates for GSM Evolution*) 101
3.4.6. *3G* (*Third Generation*) 102
3.4.7. *4G* (*Fourth generation*) 103
3.4.8. *5G* (*5th generation mobile networks or 5th generation wireless systems*) 103
3.5. *Bluetooth* 104
3.6. *Frame-Relay* 106
3.7. *GPS* (*Global Positioning System*) 108
Capitolul 4. Functii și protocoale la niveluri Reţea și Transport 112
4.1. Nivelul Reţea 113

4.1.1. Sisteme autonome. Clasificarea protocoalelor de rutare 114
4.1.2. Determinarea căii optime 117
4.2. Protocolul *IP* 118
4.2.1. Structura antetului *IP* 119
4.2.2. Adresa *IP* şi clasele de adrese 122
4.2.3. Masca de reţea 126
4.2.4. Subreţele (*host*-uri) 127
4.2.5. Prima şi ultima subreţea 129
4.2.6. *Supernetting* 131
4.3. Protocoalele de nivel Reţea 131
4.3.1. *ARP* (*Address Resolution Protocol*) 132
4.3.2. Alte protocoale în suita de protocoale Internet 135
4.4. Nivelul Transport. Protocoalele la nivel Transport 136
4.4.1. *UDP* (*User Datagram Protocol*) 136
4.4.2. *TCP* (*Transmission Control Protocol*) 139
Capitolul 5. Descrierea nivelelor: Sesiune, Prezentare, Aplicaţie 146
5.1 Nivelul Sesiune 146
5.2. Nivelul Prezentare 147
5.3. Nivelul Aplicaţie 149
5.3.1. *Telnet* (*Terminal Emulation Protocol*) 151
5.3.2. *File Transfer Protocol* (*FTP* - Protocol pentru transferul fişierelor) 151
5.3.3. *World Wide Web* 153
5.3.4. Poşta electronică 154
Capitolul 6. Dispozitivele retelelor de calculatoare. Modul de interconectare 157
6.1. Repetoare. *Hub*-uri 158
6.2. Punţile (poduri, *bridge*-uri). 161
6.2.1. Principiile de funcţionare a punţilor 162
6.2.2. Rolul punţii în comunicaţia din interiorul aceluiaşi segment 163
6.2.3. Rolul punţii în comunicaţia dintre segmente. 165

6.2.4. Cum îşi construieşte puntea tabela de comutare 166
6.3. Comutator (*Switch*) 168
6.3.1. Tipurile de comutare folosite de un comutator 171
6.3.2. Rolul comutatoarelor în implementarea conexiunilor *Ethernet half-duplex* 173
6.3.3. Rolul comutatoarelor în implementarea conexiunilor *Ethernet full-duplex* 173
6.4. Ruterele 174
6.4.1. Tabele de rutare 174
6.4.2. Clasificări ale rutelor 175
6.4.3. Efectul ruterelor asupra domeniilor de difuzare şi a domeniilor de coliziune 178
6.4.4. Tipurile ruterelor 179
6.5. Protocolul *STP* (*Spanning Tree Protocol*) 181
6.5.1. Prevenirea apariţiei avalanşelor de difuzări 182
6.5.2. Modul de funcţionare a *STP* 182
6.6. Placa de reţea (adaptor *LAN*). 186
6.7. Modemul 188
6.8. Intranetul 190
Capitolul 7. Administrarea reţelelor 192
7.1. Caracteristici ale reţelelor client-server 192
7.2. Securitatea reţelei 194
7.2.1. Modelul de securitate 194
7.2.2. Modalităţi de protecţie a informaţiei 196
7.2.3. Categorii principale de atacuri asupra informaţii 196
7.2.4. Programe distructive 198
Bibliografie 200
Anexa 1. Descrierea succintă a unor protocoale folosite în reţele de calculatoare 205
Anexa 2. Descrierea unor tipuri de adaptoare 218
Anexa 3. Costul porţilor pentru unele lăţimi de bandă 220
Anexa 4. Rată de transmisie a datelor 221

Introducere

Apariţia unor tehnici noi de transmitere a informaţiilor au condus la interconectarea calculatoarelor prin intermediul unor mijloace de comunicaţie şi la dezvoltarea reţelelor de calculatoare. Reţeaua de informare a adus un nou nivel de schimb de informaţii, ceea ce a facilitat crearea diferitor tipuri de reţele - de la reţelele private până la reţelele globale. Aplicaţiile reţelelor pot fi evidenţiate prin:

- accesul la programe complexe şi la baze de date;
- realizarea, prin reţele, a unui mediu complex de comunicaţii.

Reţele de calculatoare combină calculatoare şi dispozitive conectate în reţea în grupe, membrele cărora pot să interconecteze calculatoarele şi să trimită diferite tipuri de informaţii.

Formarea culturii de informaţie are loc în instituţiile superioare, prin explorarea unor noi domenii ale ştiinţei. Reforma sistemului educaţional presupune apariţia unor noi funcţii, în care procesul de învăţământ va fi mai complex de-a lungul perioadelor de studiu. Aceste domenii includ telecomunicaţiile, reţele locale şi globale, baze de date, calcule complexe, multimedia etc.

Punerea în aplicare a noilor tehnologii în procesul de învăţare necesită o reînnoire permanentă a conţinutului educaţiei universitare şi a cadrelor didactice. Utilizarea calculatorului în procesul de învăţare necesită nu numai un salt calitativ, dar şi o schimbare psihologică a studentului.

***

Lucrarea „Reţele de calculatoare” este propusă studenţilor care îşi fac studiile la Universitatea Pedagogică de Stat „Ion Creangă” în contextul integrării tehnologiilor informaţionale şi de comunicaţii, în formarea profesorilor de informatică, şi abordează următoarele aspecte:

- o viziune integrată asupra reţelelor de calculatoare;
- evoluţia reţelelor de comunicaţie;
- clasificarea reţelelor de calculatoare;

- topologia reţelelor de calculatoare;
- proiectarea funcţională a reţelelor de calculatoare;
- elemente de standardizare a echipamentelor reţelelor;
- analiza mediilor şi tehnicilor de comunicaţie;
- studierea protocoalelor de comunicaţie în reţea;
- sisteme de telefonie mobilă;
- rolul şi funcţiile dispozitivelor de interconectare a reţelelor;
- administrarea reţelelor.

În această lucrare sunt descrise principiile de funcţionare a sistemelor de transmitere, stocare şi de prelucrare a informaţiei [1, 2, 3, 4]. În partea a II-a va fi explicată partea practică cu privire la realizarea, documentarea şi deservirea reţelelor de calculatoare.

Abilitatea de a lucra cu reţelele, de a organiza reţeaua într-un mod topologic optim, precum şi capacitatea de a lucra cu echipamente de reţea este o cunoaştere importantă pentru profesorul de informatică, care trebuie să îndeplinească rolul administratorului de reţea în şcoală [5, 6].

Lista abrevierelor

3G	Third Generation
3GPP	Generation Partnership Project
ACK	ACKnowledge
ADSL	Asymmetric Digital Subscriber Line
AFP	Apple Filing Protocol
AMPS	Advanced Mobile Phone System
ANSI	American National Standards Institute
ARP	Address Resolution Protocol
ARPA	Advanced Research Projects Agency
ARPANET	Advanced Research Projects Agency NETwork
ARQ	Automatic Retransmition Query
AS	Autonomous System
ASCII	American Standard Code for Information Interchange
ASIC	Application Specific Integrated Circuit
ASMP	ASymmetric Multi-Processing
ASN.1	Abstract Syntax Notation 1
ASP	Apple Talk Session Protocol
ATM	Asynchronous Transfer Mode
AUI	Attachment Unit Interface
BGP	Border Gateway Protocol
BNC	Bayonet Neill Concelman
BPDU	Bridge Protocol Data Unit .
BSS	Basic Service Set
CAM	Content Adressable Memory
CDMA	Code Division Multiple Access
CHAP	Challenge Handshake Authentication Protocol
CIDR	Classless InterDomain Routing
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CTS	Clear To Send
D-AMPS	Digital Advanced Mobile Phone System
DARPA	Defense Advanced Research Projects Agency

DGPS	Diferenţial GPS
DMA	Direct Memory Access
DMSP	Distributed Mail System Protocol
DNA	Digital Network Architecture
DNS	Domain Name System
DOCSIS	Data Over Cable Service Interface Specification
DOS	Disk Operating System
DS0	Digital Signal 0
DSL	Digital Subscriber Line .
EBCDIC	Extended Binary Coded Decimal Interchange Code
ECTP	Ethernet Configuration Test Protocol
EGP	Exterior Gateway Protocol
EHF	Extra High Frequency
EIA	Electronic Industries Alliance
EIGRP	Enhanced Interior Gateway Routing Protocol
ESS	Extended Service Set
ETS	ElecTronic Serial number
FCS	Frame Check Sequence
FDDI	Fiber Distributed Data Interface
FEC	Forward Error Correction
FR	Frame Relay
FTP	File Transfer Protocol
GGP	Gateway-to-Gateway Protocol
GPS	Global Positioning System
GSM	Global System for Mobile communications
GUI	Graphic User Interface
HDLC	High-level Data Link Control
HDSL	High-bit-rate DSL;
HF	High Frequency
HTML	HyperText Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	HyperText Transfer Protocol/Secure
HUB	Host Unit Broadcast
IBSS	Independent Basic Service Set

ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronic Engineers
IEEE 802.11	Wi-Fi
IEEE 802.3	Ethernet
IEEE 802.5	Token Ring
ISP	Internet Service Provider
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IGRP	Interior Gateway Router Protocol
IMAP	Interactive Mail Access Protocol
IP	Internet Protocol
IPX	Internetwork Packet eXchange
IRC	Internet Relay Chat
ISDN	Integrated Services Digital Network
ISM	Industrial, Scientific and Medical radiobands
ISO	International Organization for Standardization
ISP	Internet Service Provider
ITU	International Telecommunications Union
LAN	Local Area Network
LCP	Link Control Protocol
LDAP	Lightweight Directory Access Protocol
LED	Light-Emitting Diode
LF	Low Frequency
LLC	Logical Link Control
LMI	Local Management Interface
LTE	Long Term Evolution
MAC	Media Access Control
MAN	Metropolitan Area Network
MCNS	Multimedia Cable Network System
MF	Medium Frequency
MILNET	MILitary NETwork
MIN	Mobile Identification Number
MSAU	MultiStation Access Unit
MTSO	Mobile Telephone Switching Office

MTU	Maximum Transfer Unit
NCP	Network Control Protocol
NFS	Network File System
OSI	Open System Interconnection
OSPF	Open Shortest Path First
PAP	Password Authentication Protocol
PHP	Personal Home Page
PLS	Physical signaLing Sublayer
PMA	Physical Medium Attachement
POP	Point Of Presence / Post Office Protocol
POTS	Plain Old Telephone Service
PPP	Point-to-Point Protocol
RADSL	Rate Adaptive DSL.
RARP	Reverse Address Resolution Protocol
RFC	Request For Comments
RIP	Router IP
RPC	Remote Procedure Call
RS-232	Recommended Standard 232
RTP	Real-time Transport Protocol
RTS	Request to Send
RTSP	Real Time Streaming Protocol,
SCP	Session Control Protocol .
SCTP	Stream Control Transmission Protocol
SDSL	Single-line DSL;
SFD	Start Frame Delimiter .
SHF	Super High Frequency
SID	System Identification Code
SIP	Session Initiation Protocol
SLIP	Serial Line IP
SMB	Server Message Block
SMTP	Simple Mail Transfer Protocol
SNA	Systems Network Architecture
SNMP	Simple Network Management Protocol
SONET	Synchronous Optical NETworking

SPF	Shortest Path First
SPX	Sequenced Packet eXchange
SQL	Structured Query Language
SSH	Secure SHell
SSL	Secure Sockets Layer
STP	Spanning Tree Protocol
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TDM	Time-Division Multiplexing
Telnet	Terminale virtuale
TIA	Telecommunications Industry Association
TLS	Transport Layer Security
TTL	Time To Live
UCAID	University Corporation for Advanced Internet Development
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
UTP	Unshielded Twisted Pair
VDSL	Very-high-bitrate DSL;
VHF	Very High Frequency
VLAN	Virtual Local Area Network
VLF	Very Low Frequency
VMTP	Versatile Message Transaction Protocol
WAN	Wide Area Network
WhoIs	„Who Is”
WLAN	Wireless Local Area Network;
WWW	World Wide Web
X.25	Packet Switching
XDR	eXternal Data Representation
xDSL	x (*for family of tehnologies*) Digital Subsriber Line
XML	eXtendable Markup Language
XMPP	eXtensible Messaging and Presence Protocol

Capitolul 1. Principii, clasificări și modele de referință ale reţelor de calculatoare

1.1. Principii şi noţiuni fundamentale în transferul de date prin reţele de calculatoare
- 1.1.1. Componentele generale ale unei reţele
- 1.1.2. Incapsularea

1.2. Clasificarea reţelelor de calculatoare
1.3. Topologii ale reţelelor de calculatoare
1.4. Protocoale de comunicaţie în reţea
1.5. Modelul de referinţă *ISO-OSI*
- 1.5.1. Nivelul 7
- 1.5.2. Nivelul 6
- 1.5.3. Nivelul 5
- 1.5.4. Nivelul 4
- 1.5.5. Nivelul 3
- 1.5.6. Nivelul 2
- 1.5.7. Nivelul 1

1.6. Modelul de referinţă *TCP/IP*
1.7. Comparaţia modelelor *ISO-OSI* şi *TCP/IP*

Reţelele de calculatoare - reprezintă cazuri particulare ale **reţelelor de telecomunicaţii**. O astfel de structură poate fi definită ca un ansamblu de echipamente de calcul, conectate între ele cu scopul de a prelucra şi transporta la distanţă diverse informaţii, reprezentate prin date.

Echipamentele reţelelor de calculatoare nu sunt neapărat numai calculatoarele, ci şi orice alt dispozitiv capabil să prelucreze date. Calculatoarele din reţele pot fi de tipuri diferite, atât ca *hard* cât şi ca *soft*. De exemplu, folosirea telefoniei mobile poate servi drept o reţea de date; televiziunea digitală de asemenea reprezintă avantajele tehnologiilor din reţelele de calculatoare; jocurile de calculator au schimbat modul de folosire a timpului liber.

1.1. Principii şi noţiuni fundamentale în transferul de date prin reţele de calculatoare

Până la începutul anilor ’80 din secolul trecut sistemele de calcul erau organizate în jurul unui calculator central capabil să rezolve problemele transmise de numeroşi utilizatori. Datorită tendinţei de trecere de la acest sistem centralizat, la soluţia instalării de calculatoare la fiecare utilizator şi asigurarea unor legături de comunicaţie eficientă între ele, a contribuit la dezvoltarea reţelelor de calculatoare, ca o parte integrantă a societăţii moderne.

O reţea de calculatoare (*computer network*) reprezintă un sistem de calcul complex, format din mai multe echipamente interconectate prin intermediul unui canal de comunicaţie (**cablu coaxial, fibră optică, linie telefonică, ghid de unde**) în scopul utilizării în comun de către mai mulţi utilizatori a tuturor resurselor fizice, logice şi informaţionale, asociate calculatoarelor din reţea. Calculatoarele conectate la reţea sunt denumite **noduri**.

Utilizarea calculatoarelor în reţea are o serie de avantaje: [7]

- accesul la toate resursele (echipamente, programe şi date) al oricărui utilizator indiferent de localizarea sa fizică;
- creşterea gradului de fiabilitate a sistemului de calcul, prin preluarea sarcinilor componentelor care apar de către alte componente disponibile în reţea;
- posibilitatea extinderii reţelei prin adăugarea de noi componente *hard* şi *soft* care să asigure creşterea performanţelor;
- implementarea diverselor aplicaţii cu aceleaşi investiţii de către mai mulţi utilizatori;
- crearea unor puternice medii de comunicaţie interumane.

Există mai multe tipuri de reţele, ele diferenţiindu-se prin distanţele pe care le acoperă, debitul utilizat în transmiterea informaţiei, tehnica de comutare folosită etc.

În Fig. 1.1. se reprezintă schema unei reţele locale (*LAN - Local Area Network*) interconectate *WAN* (*Wide Area Network*), unde *ISP - internet service provider.*

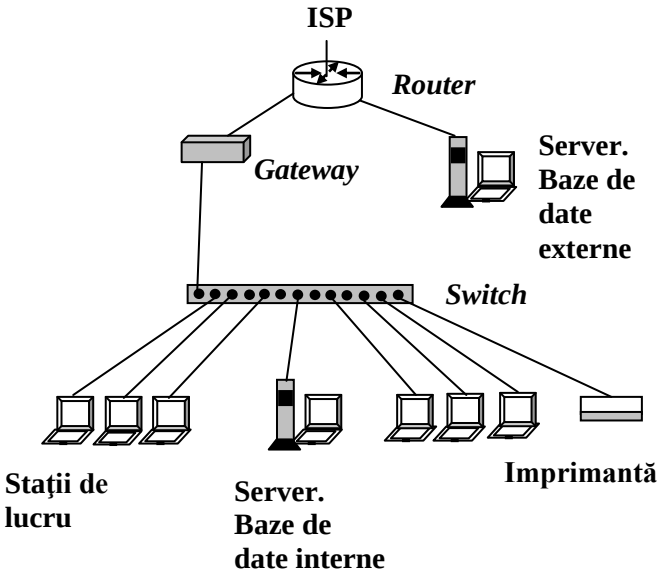


Fig. 1.1. Schema unei reţele locale (*LAN*) interconectate *WAN*

1.1.1. Componentele generale ale unei reţele sunt:

- **Calculatorul central** - denumit computer gazdă (*host computer*) sau *file-server*, este cel care gestionează funcţionarea întregii reţele şi concentrează o mare parte din resursele acesteia;

- **Staţiile de lucru** - numite şi nodurile reţelei, sunt echipamentele de calcul eterogene conectate la calculatorul central şi între ele, având posibilitatea să transmită şi să recepţioneze date şi să partajeze între ele resursele întregii reţele. În reţele pot fi conectate şi alte echipamente: imprimante, copiatoare, faxuri etc.;

- **Mediile de comunicaţie** - reprezentate de suporţii pe care sunt vehiculate pachetele de date între nodurile reţelei;

- **Echipamentele de adaptare**[1] - realizează compatibilitatea între calculatoarele din reţea şi mediile de comunicaţie;

- **Echipamente de control al comunicaţiilor** - optimizează traficul de mesaje dintre componentele reţelei şi asigură protecţia datelor. Ele utilizează o varietate de coduri de comunicaţie şi tehnici de transmisie.

- **Sistemul de operare al reţelei** - reprezintă pachetul de programe, instalat pe calculatorul central, care asigură coordonarea funcţiilor acesteia şi compatibilitatea între sistemele de operare instalate pe calculatoarele locale.

1.1.2. Încapsularea

Pentru ca mai mulţi utilizatori să poată **transmite simultan informaţii în reţea**, datele trebuie fragmentate în unităţi mici. Cu acest scop înainte ca datele să fie transmise, ele trec printr-un proces numit **încapsulare.** Încapsularea adaugă informaţii specifice prin elaborarea unui ***antet*** şi a unui ***trailer*** la fiecare nivel [8].

Prin încapsulare, protocoalele de pe fiecare nivel de transmisie pot comunica între sursă şi destinaţie independent de celelalte niveluri. Aceste unităţi reprezintă **unităţile de bază ale comunicaţiilor** în reţea şi în dependenţă de nivelul de transmisie sunt numite **segmente/ pachete/cadre**.

Componentele acestor unităţi sunt grupate **în trei secţiuni**:

Antetul - conţine un semnal de atenţionare, care indică faptul că se transmite un set de date; adresa sursă; adresa destinaţie; informaţii de ceas pentru sincronizarea transmisiei.

Datele - reprezintă informaţiile care se transmit. Această componentă poate avea dimensiuni diferite, în funcţie de reţea.

[1] Unele echipamente de adaptare şi control pot lipsi sau pot fi montate în configuraţii care diferă de la o reţea la alta.

Postambulul (*trailer*) – depinde de protocolul utilizat. De obicei conţine o componentă de verificare a erorilor, numită *CRC* (*Cyclic Redundancy Check*) sau *FCS* **(***Frame Check Sequence*).

Segmente/pachete/cadre pot conţine mai multe tipuri de date printre care: informaţii (mesaje sau fişiere); anumite tipuri de date şi comenzi de control pentru calculator (solicitățile de servicii; codurile de control al sesiunii etc.). Dacă datele sunt fragmentate în segmente/pachete/cadre, transmisiile individuale vor fi accelerate, astfel încât fiecare calculator din reţea va putea transmite şi recepţiona date.

Verifică-ţi cunoştinţele:

1) Enumeraţi avantajele de utilizare a calculatoarelor în reţea.
2) Cu ce scop înainte ca datele să fie transmise, ele trec printr-un proces numit încapsulare?
3) Explicaţi funcţiile componentelor de bază (segmente/ pachete/cadre) ale comunicaţiilor în reţea.

1.2. Clasificarea reţelelor de calculatoare

Criteriile de bază ale reţelelor de calculatoare pot fi clasificate după: **tehnologia de transmisie** (modul de difuzare a datelor), **accesul la mediu, mărime**, **modul de interacţiune cu sistemele de operare [9]**.

În tabelul 1.1 se indică: modul de clasificare a reţelelor, tipurile de reţele, caracteristica reţelelor **[10, 11].**

Verifică-ţi cunoştinţele:

1) Enumeraţi modurile de clasificare a calculatoarelor în reţea.
2) Definiţi caracteristicile diferitor tipuri de reţea.

Tabelul 1.1. Modul de clasificare a reţelelor

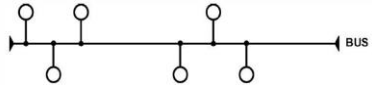
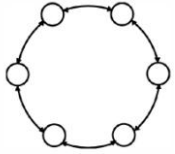
Nr.	Modul de clasificare	Tipuri de reţele	Caracteristica tipului de reţele
1	Tehnologia de transmisie	Reţele cu difuzare	Se caracterizează prin asigurarea unui **mediu comun** la care au acces toate dispozitivele din reţea, astfel oricare dintre mesajele trimise de un membru al acestui tip de reţea să poată fi recepţionat de toţi ceilalţi membri din reţea. Implementarea unei reţele bazate pe difuzare presupune şi asigurarea unui **mecanism de identificare** atât a celui ce a trimis, cât şi a destinatarului. Reţelele cu difuzare sunt mai uşor de implementat.
		Reţelele de tip punct-la-punct	Sunt alcătuite din **perechi de maşini** care comunică între ele. Pentru a parcurge traseul de la o sursă la destinaţie intr-o reţea de acest tip datele vor „călători" prin una sau mai multe staţii intermediare. Pot exista mai multe trasee intre o sursă şi o destinaţie, pentru care este necesară implementarea unor algoritmi specializaţi de dirijare.
2	Accesul la mediu	Alocarea statică	Fiecărei staţii sau fiecărui modul i se alocă **o cuantă de timp** (în cazul *TDMA - Time Division Multiple Access*) sau **o bandă de frecvenţă** (*FDMA - Frequency Division Multiple Access*). Această alocare este statică, în sensul că dacă jumătate din staţii nu transmit, cuantele alocate lor nu sunt reutilizate.

2	Accesul la mediu (continuare)	Alocarea dinamică	Se alocă **pe rând o cuantă de timp** staţiilor care vor să transmită. De exemplu, în cazul tehnologiilor de tip *TokenRing*, există o secvenţă de biţi, numit **jeton**. Acest jeton permite staţiei care îl deţine să transmită ce vrea. După ce a terminat de transmis, dă drumul la jeton care se „plimbă" pe reţea până ajunge la următoarea staţie.
		Alocarea aleatoare	Fiecare staţie procedează astfel: **ascultă** să vadă dacă nu cumva altă staţie transmite în acel moment. Dacă da, **aşteaptă** până când nu mai transmite nimeni. După ce aude că e „linişte", **se apucă de transmis.** Fiecare staţie procedează exact la fel, toate au **drept egal** de a începe transmisia. Există riscul că două staţii să asculte simultan şi când nimeni nu mai transmite, să înceapă ambele transmisia în acelaşi timp. În acest caz, mesajele celor două staţii se „ciocnesc" pe fir, dând naştere unei **coliziuni.**
3	Mărimea reţelelor	Reţele foarte restrânse	*VLAN* (*Virtual Local Area Network*) - reprezintă gruparea unor echipamente de reţea **după criterii logice** şi nu după topografia fizică, precum în cazul unui *LAN* obişnuit.
		Reţelele locale	*LAN* (*Local Area Network*) - componentele unei astfel de reţele sunt situate la **distanţe relativ mici** (de la câţiva metrii până la 5 km), amplasate într-o clădire sau un grup de clădiri învecinate. Reţelele locale sunt proiectate să realizeze următoarele lucruri: să permită

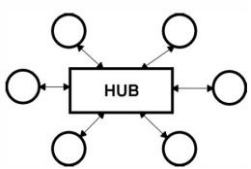
3	Mărimea reţelelor (continuare)		unui număr de utilizatori să acceseze media cu lăţime de banda mare; să furnizeze conectivitate permanentă la serviciile locale; să conecteze echipamente de reţea adiacente.
		Reţelele metropolitane	*MAN* (*Metropolitan Area Network*) - sunt reţele care acoperă aria **unui mare oraş** (metropolă) fiind folosite pentru conectarea *LAN*-urilor. Distanţele acoperite pot ajunge până la 75 km. În funcţie de arhitectura reţelei, viteza de transmisie poate fi mai mare pe distanţe mai mici. Acest tip de reţele este foarte asemănător cu categoria *WAN*.
		Reţelele largi	*WAN* (*Wide Area Network*) – acoperă **distanţe foarte mari**, deseori o ţară sau un continent, şi asigură utilizarea în comun a resurselor de calcul ale unor sisteme foarte complexe de către utilizatori. Este necesar ca informaţia să se transmită rapid şi eficient la diferite locaţii geografice: de exemplu Moldova cu Statele Unite. Această reţea aparţine de obicei unei companii de telefonie sau unui furnizor de servicii Internet (*ISP - Internet Service Provider*). Clienţii se conectează la această mare reţea folosind echipamente speciale şi plătind o taxă lunară *ISP*-ului. *WAN* folosesc protocoale şi tehnologii diferite decât *LAN*. Câteva dintre acestea sunt: *WAN modems*; *ISDN* (*Integrated Services Digital Network*); *DSL* (*Digital Subscriber*

			Line); *Frame relay*; *ATM* (*Asynchronous Transfer Mode*); *Carrier Series T* (*US*) şi *Carrier Series E* (*Europe*): *T1, E1, T3, E3*; *SONET* (*Synchronous Optical Network*) etc.
4	Modul de interacţiune al sistemului de operare	Bazate pe server (tehnologia *client-server*)	- Se bazează pe trimiterea, de către clienţi, a unei cereri prin care solicită unuia sau mai multor sisteme din reţea anumite informaţii. Un sistem care furnizează un serviciu de reţea se numeşte **server**. - Funcţiile principale ale unui sistem sunt: **administrarea şi procesarea datelor, prezentarea datelor către utilizator**. - Printre avantajele **tehnologiei *client-server*** se poate evidenţia reducerea costurilor prin partajarea resurselor, administrarea simplificată, bazată pe centralizare, scalabilitate.
		Egal-la-egal (*peer-to-peer*)	Tratează clienţii în mod egal. Oricare două echipamente configurate corespunzător au capacitatea de a solicita sau de a oferi servicii.

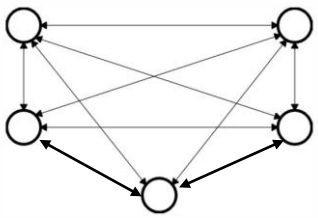
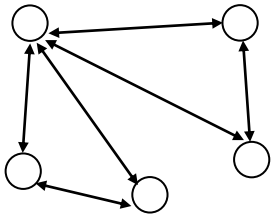
Tabelul 1.2. Tipurile de topologii şi descrierea lor succintă

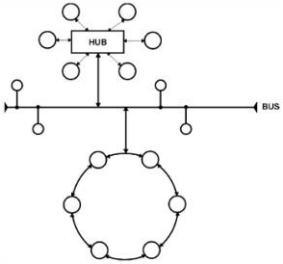
Nr.	Topologie	Descrierea
1	**Topologia *BUS* (magistrală)** 	Această arhitectură presupune existenţa unui **mediu fizic comun de comunicaţie**. Din acest motiv la un moment dat un singur nod poate transmite prin intermediul mediului partajat. *Atenţie*: Topologia *BUS* - necesită algoritmi şi dispozitive de arbitrare a accesului la mediul fizic comun. *Avantaje*: Resurse utilizate foarte puţine - o singură interfaţă *per nod*, un singur mediu fizic de transport. *Dezavantaje*: Fiabilitate scăzută - o defecţiune apărută la nivelul mediului fizic duce la căderea întregii reţele.
2	**Topologia *RING* (inel)** 	Se dezvoltă din structura *BUS*, fiind construită tot **în jurul unei resurse comune** (mediul fizic de comunicaţie), dar are o fiabilitate mai mare, deoarece la apariţia unei defecţiuni la nivelul mediului fizic partajat nu se va invalida întreg sistemul pentru că există totdeauna o rută (cale) alternativă de comunicare. Ambele arhitecturi *BUS* şi *RING* au în general o rată de transmisie a datelor cuprinsă în intervalul *1...10 Mbps*[2] (vezi Anexa 4), relativ mică.

[2] 1..10 megabiţi per second. Atunci când discutăm despre viteze de transmisie ale unor anumite tehnologii de

		Acestea se aplică la conectarea sistemelor de calcul în structuri mici (birouri). *Avantajul* constă în aceea că durata transmisiei unui mesaj între două puncte ale reţelei poate fi exact determinat. *Dezavantajul* constă în faptul că defectarea unei staţii atrage „căderea" întregii reţele.
3	**Topologia *STAR* (stea)**  	Fiabilitatea sistemului este dată de **fiabilitatea elementului central.** Defectarea unui nod sau a conexiunii aferente acestuia nu influenţează asupra funcţionalităţii reţelei. ***HUB*-**ul mai poate executa şi alte funcţii. Ele pot fi inteligente sau pasive, pot avea funcţii în formarea semnalelor sau în filtrarea zgomotelor etc. *Avantajele* topologiei - în rapiditatea transmisiilor şi fiabilitate în funcţionare *Dezavantajul* principal îl constituie complicarea construcţiei *hard* a *host*-computerului, la care trebuiesc cuplate fizic zeci sau chiar sute de staţii. De aceea, în practică staţiile de lucru sunt legate în stea la un echipament intermediar, numit *HUB* de reţea, iar acesta asigură conexiunea cu calculatorul central.

nivel 2, sau de viteze maxime pe care le suportă anumite tipuri de cablu, în general viteze la nivel 1 şi 2, discutăm în biţi sau megabiţi, dar nici într-un caz în *bytes*. Atunci când discutăm despre viteze raportate de aplicaţii (*browser, client FTP*, etc.), în general viteze la niveluri *OSI* superioare, discutăm în *bytes* sau octeţi. Există o convenţie conform căreia notaţia „*bps*" se referă la biţi, iar „*Bps*" la *bytes* [5].

4	**Topologia *MESH* (plasă)**  a) completă  b) incompletă	Această topologie presupune existenţa unei **conexiuni între nodurile din reţea (**completă - oricare două noduri, plasă – doar unele noduri). Este o structură ideală pentru cazul reţelelor locale. Introducerea unui nod nou în reţea se reflectă asupra tuturor nodurilor existente deoarece presupune apariţia la fiecare în parte a unei interfeţe şi a unei conexiuni fizice. *Avantaje:* 1) Un nod poate transmite oricând, el nefiind condiţionat de starea de activităţi a celorlalte noduri. Deci structura este caracterizată de disponibilitate maximă. 2) Fiabilitate maximă - nefuncţionarea unui nod sau a unei conexiuni nu influenţează asupra funcţionării globale a reţelei. *Dezavantaje***:** 1) Conexiunile nu sunt folosite optim. Rata lor de ocupare este minimă. 2) Consum maxim de resurse fizice (interfeţe, conexiuni etc.), deci in această situaţie vom avea costurile maxime. *Utilizare*: În general această topologie este utilizată în reţelele metropolitane, unde un nod este reprezentat de o structură *LAN - WAN - LAN*. In această situaţie costurile implicate de constituirea structurii sunt mici în comparaţie cu avantajul disponibilităţii canalului de comunicaţie.

5	**Topologia mixtă (hibridă)** 	Este cea mai larg întâlnită în structurile reale şi presupune **interconectarea mai multor structuri cu topologii de bază diferite**. Într-o asemenea situaţie se realizează o pondere a avantajelor şi dezavantajelor fiecărei topologii astfel încât să realizăm un optimum pentru o structură dată.
6	**Topologia arborescentă (*tree*)**	Este o **structură mixtă** dar este dezvoltată pe acelaşi principiu ca arhitectura *STAR*. Diferenţa este ca din punct de vedere ierarhic, elementul central al unei structuri stea poate fi privit ca nod în cadrul unei structuri de reţea stea de nivel ierarhic superior. Totuşi aceste structuri nu se pot cascada la infinit (o structură de reţea arborescentă nu poate avea un număr foarte mare de structuri ierarhic inferioare).

1.3. Topologii ale reţelelor de calculatoare

Prin topologia reţelelor vom înţelege **modul de conectare al nodurilor** ce comunică între ele.

În funcţie de necesităţile de comunicare şi de cerinţele impuse, s-au dezvoltat mai multe topologii de reţea. Topologia unei reţele se referă la structura acesteia, la modul de aşezare al nodurilor reţelei, precum şi la logica prin care acestea comunică. Nodurile pot fi calculatoare independente sau structuri *LAN* - *WAN*.

Topologiile se pot împărţi în două categorii: **topologii fizice şi topologii logice**. Cele **fizice** tratează aspectul spaţial şi organizarea fizică a staţiilor din reţea şi a cablurilor, pe când cele **logice** se referă la modul în care se realizează comunicarea în reţea.

Dintre tipurile de topologii existente, menţionăm: plasă (*mesh)*, magistrală (*bus*), inel, stea, mixtă (hibridă), arborescentă, etc. Acestea se referă atât la topologiile fizice, cât şi la cele logice.

În Tabelul 1.2 sunt descrise succint tipurile de topologii.

Verifică-ţi cunoştinţele:

1) Enumeraţi tipuri de topologii ale reţelelor.
2) Explicaţi deosebirea dintre topologii fizice şi topologii logice.
3) Numiţi avantajele şi dezavantajele diferitor tipuri de topologii.

1.4. Protocoale de comunicaţie în reţea

Un protocol într-o reţea de telecomunicaţii este o **descriere formală a regulilor şi convenţiilor** care stau la baza comunicării între dispozitivele ataşate la reţea. Protocolul determină **formatul sau structura mesajului**, **metodele** prin care dispozitivele din reţea schimbă informaţii privitoare la căile către alte reţele, temporizarea, **ordinea şi controlul erorilor** în comunicaţiile de date, **iniţierea şi finalizarea** sesiunii pentru transferul de date şi altele.

Cu alte cuvinte, protocolul reprezintă **un set de reguli** ce se referă la ceva concret (de exemplu, modul de funcţionare a *mail*-ul, paginile de *web* etc). Fără protocoale calculatoarele nu ar putea construi sau reconstrui în formatul original şirul de biţi (mesajul) transmis de la un alt calculator [12].

Protocoalele controlează toate aspectele comunicaţiilor de date, incluzând: Cum e construită fizic reţeaua? Cum sunt conectate între ele calculatoarele din reţea? Cum sunt formatate datele pentru transmitere? Cum sunt trimise datele? Ce se întâmplă când apar erori şi cum se pot corecta erorile?

Aceste reguli sunt sau au fost create şi dezvoltate permanent de diferite organizaţii şi comitete internaţionale. Printre acestea figurează *Institute of Electrical and Electronic Engineers* (*IEEE*), *American National Standards Institute* (*ANSI*), *Telecommunications Industry Association* (*TIA*), *Electronic Industries Alliance* (*EIA*) şi *International Telecommunications Union* (*ITU*)[3] [7].

---

[3] Pentru exemplificarea acestui proces se propune o istorie a apariţiei protocoalelor Internet. În perioada 1960 – 1970, *Advanced Research Projects Agengy* (*ARPA*) *of Department of Defense* (*DOD*) a sponsorizat dezvoltarea şi realizarea ***ARPANET***. *ARPANET* - include centre de cercetare, civile, militare şi universităţi, şi a fost creată să realizeze proiecte privind cercetarea militară şi din domeniul ştiinţei calculatoarelor. În prezent *ARPA* este numită *DARPA* cu litera D în faţă care vine de la *Defence*. În 1984 *DOD* a divizat *ARPANET* în două reţele: *ARPANET* pentru cercetări experimentale şi *MILNET* pentru domeniul militar. Reţeaua *ARPANET* consta din aproximativ 50 de calculatoare, care au fost legate între ele prin linii telefonice cu capacitatea de transmisie de *57.6 Kbps*. Calculatoarele de tip staţie (*host*) şi *gateway*-le din *ARPANET* au fost conectate la aceste 15 calculatoare. În 1980 a fost dezvoltată pentru *ARPANET* o nouă familie de protocoale, denumită ***DARP* Internet** şi în general este referit ca ***TCP/IP*** (*Transmission Control Protocol/Internet Protocol*). În 1987 *NSF* (*the National Science Fundation*) a fondat o reţea care conecta şase centre cu calculatoare.

De exemplu, în cazul Internetului, un protocol de comunicaţie reprezintă un set de reguli, pe care două dispozitive trebuie să le respecte atunci când comunică (transmit date) între ele. Întreaga suită de protocoale Internet poartă denumirea de familia de protocoale *TCP/IP*. Folosind aceste protocoale, mesajele de control sunt generate şi procesate de *software*-ul de reţea, fără implicarea utilizatorului.

În Anexa 1 sunt prezentate unele protocoale de bază, care asigură transmiterea corectă a informaţiilor prin reţele de calculatoare.

Verifică-ţi cunoştinţele:

1) Cum puteți explica necesitatea protocoalelor?
2) Enumeraţi funcţiile protocoalelor.

1.5. Modelul de referinţă *ISO-OSI*

În lumea reţelelor de calculatoare există foarte multe standarde, care impun anumite cerinţe şi restricţii funcţionale.

Standardul este un **document care stabileşte anumite reguli** despre desfăşurarea unei activităţi, sau nivelul de calitate a unui produs, sau impune unele cerinţe obligatorii pe care un anumit produs trebuie să le îndeplinească. Aceste norme se referă la **aspectele funcţionale**, şi nu la cele tehnologice.

În cazul, când vrem să standardizăm o transmisie de date între două calculatoare, această transmisie este un lucru complex şi nu poate fi tratată în cadrul unui singur protocol. Deaceia s-a intervenit la noţiunea de **stivă de protocoale**. Problema principală a fost împărţirea unei transmisii de date de la o staţie la alta în mai multe niveluri independente. Astfel, o **stivă de protocoale reprezintă o stivă de mai multe niveluri prin care trec datele** în cadrul unei transmisii de date [13].

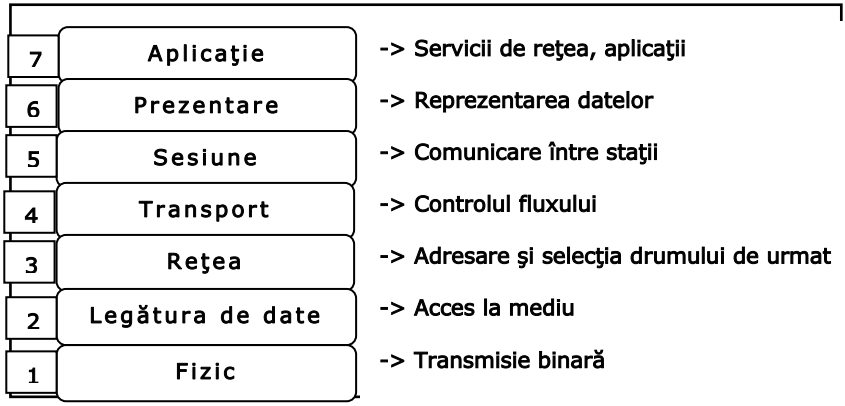


Fig. 1.2. Modelul *ISO-OSI*

Avantajele acestui gen de model sunt**:**

* Sparge comunicaţia în reţea, precum şi complexitatea acesteia şi numeroasele aspecte implicate în părţi mai mici, care pot fi studiate individual şi tratate separat;

* Standardizează componentele de reţea pentru a putea face posibilă dezvoltarea în sistem de concurenţă a dispozitivelor; astfel este stimulată şi activitatea de cercetare;

* Permite diferitelor tipuri de *hardware* şi *software* de reţea să comunice între ele;

* Modularitatea împiedică ca schimbările dintr-un nivel să producă modificări în alte niveluri; fiecare nivel este separat şi se poate dezvolta independent;

* Împarte problemele comunicării în reţea în părţi mai mici, pentru a putea fi înţelese şi explicate mai uşor;

* Permite existenţa unor dispozitive de interconectare mai ieftine şi mai eficiente, care nu cunosc decât protocoalele de pe câteva niveluri.

Organizaţia Internaţională de Standardizare (*ISO*) - una din cele mai importante organizaţii de standardizare - a studiat diferite tipuri de reţele existente în acea vreme (*DECnet, SNA, TCP/IP*) şi a propus în a. 1984 un model de referinţă numit ***OSI - Open System Interconnection***.

Deşi *OSI* nu este singurul model existent, este cel mai folosit în învăţământ, pentru că ilustrează cel mai bine separarea între niveluri şi împărţirea comunicaţiei în bucăţele mai mici, mai uşor de definit şi în consecinţă mai uşor de dezvoltat. *OSI* este un model teoretic structurat pe şapte niveluri: **Aplicaţie, Prezentare, Sesiune, Transport, Reţea, Legătură de date şi Fizic** (vezi Fig. 1.2) Fiecare dintre acestea ilustrând o funcţie particulară a reţelei. Separarea între funcţiile reţelei este denumită **nivelare** (*layering*).

Modelul *OSI* este un model de **arhitectură de reţea** şi nu specifică serviciile şi protocoalele utilizate la fiecare nivel. Fiecare nivel al modelului *OSI* are un set predeterminat de funcţii pe care le realizează pentru a duce la bun sfârşit comunicarea.

În continuare vom analiza succint nivelurile modelului *OSI*:

1.5.1. Nivelul 7: **Aplicaţie (*Application Layer*).** Nivelul Aplicaţie este situat cel mai aproape de utilizator şi oferă servicii de reţea aplicaţiilor utilizator. Diferă de celelalte niveluri *OSI* prin faptul că nu oferă servicii nici unui alt nivel, ci numai unor aplicaţii ce sunt situate în afara modelului *OSI* [11].

Nivelul Aplicaţie stabileşte **disponibilitatea unui calculator cu care se doreşte iniţierea unei conexiuni,** stabileşte procedurile ce vor fi urmate în cazul unor erori şi verifică integritatea datelor. De asemenea identifică dacă există suficiente resurse pentru a sprijini comunicaţia între parteneri. Exemple de astfel de aplicaţii sunt **editoare de texte, utilitare de calcul tabelar, terminale bancare,** etc.

Pentru a fi mai uşor să vă amintiţi despre acest nivel, gândiţi-vă la ***browsere* de *web*** folosit de programele de navigare (*browsere*) [11].

Acestui nivel îi **corespunde protocoalele**: terminale virtuale - *Telnet*; transfer de fişiere - *FTP* (*File Transfer Protocol*); poşta electronică - *SMTP* (*Simple Mail Transfer Protocol*); *POP* (*Post Office Protocol*); Aplicaţii *web* (prezentare, baze de date etc.) cu *HTTP* (*Hyper Text Transfer Protocol*); Administrare şi monitorizare - *SNMP* (*Simple Network Management Protocol*) [9, 10] (vezi Anexa 1).

1.5.2. Nivelul 6: Prezentare (*Presentation Layer*) - se ocupă de **sintaxa şi semantica informaţiilor transmise** între aplicaţii sau utilizatori. Nivelul Prezentare asigură ca informaţia transmisă de nivelul Aplicaţie al unui sistem poate fi **citită şi interpretată** de către nivelul Aplicaţie al sistemului cu care acesta comunică. Dacă este necesar, nivelul Prezentare face **traducerea între diverse formate** de reprezentare, prin intermediul unui format comun. Tot nivelul Prezentare este responsabil cu eventuala compresie/decompresie şi criptare/decriptare a datelor.

Pentru a reţine nivelul Prezentare, gândiţi-vă la reprezentare şi la formatul comun al datelor [11].

Protocoalele utilizate: *XDR* (*eXternal Data Representation*), *ASN*.1 (*Abstract Syntax Notation 1*), *SMB* (*Server message block*), *AFP* (*Apple Filing Protocol*), *NCP* (*Network Control Protocol*) (vezi Anexa 1) [15, 16].

1.5.3. Nivelul 5: Sesiune (*Session Layer*). Prin sesiune se înţelege **dialogul între două sau mai multe entităţi**. Nivelul Sesiune se ocupă cu **stabilirea, menţinerea, gestionarea şi terminarea sesiunilor** în comunicarea dintre două staţii. Acest nivel asigură **expedierea datelor**, clase de servicii şi raportarea erorilor. Nivelul Sesiune oferă servicii nivelului Prezentare, realizează sincronizarea între nivelurile Prezentare ale două staţii şi gestionează schimbul de

date între acestea. În plus faţă de regularizarea sesiunilor, nivelul Sesiune oferă bazele pentru transferul eficient de date, pentru clase de servicii, pentru raportarea excepţiilor nivelurilor sesiune, prezentare şi aplicaţie. Acest mecanism este strâns legat cu noţiunea de port.

Dacă doriţi să reţineţi **nivelul Sesiune**, gândiţi-vă la dialog şi la conversaţii [11].

Protocoalele utilizate: *TLS* (*Transport Layer Security*), *SSH* (*Secure shell*), *RPC* (*Remote Procedure Cal*), *ASP* (*Apple Talk Session Protocol*), *NCP* (*Network Core Protocol*), *NFS* (*Network File System*) (vezi Anexa 1) [15, 16].

1.5.4. Nivelul 4: Transport (*Transport Layer*) - oferă **controlul fluxului de date**, **tratarea erorilor** şi este implicat in transmiterea şi recepţionarea pachetelor informaţionale fără erori, fără pierderi sau duplicări şi într-o ordine definita. Nivelul se ocupă de **împachetarea mesajelor, prin fragmentarea celor mari si gruparea celor mici** în scopul unei transmisii cât mai eficiente, **despachetarea datelor** la recepţie, reasamblarea mesajelor originale şi trimiterea mesajelor de **confirmarea recepţiei**. Oferă totodată suport nivelului Sesiune.

Nivelul Transport **segmentează datele** în sistemul sursă şi le **reasamblează** la destinaţie. Limita dintre nivelul Transport şi nivelul Sesiune poate fi văzută ca graniţa între protocoale aplicaţie şi protocoale de transfer de date. În timp ce nivelurile Aplicaţie, Prezentare şi Sesiune se preocupă cu probleme legate de aplicaţii, cele patru niveluri inferioare se ocupă cu probleme legate de transportul datelor. Nivelul Transport încearcă să ofere un serviciu de transport de date care să **izoleze nivelurile superioare** de orice specificităţii legate de modul în care este executat transportul datelor. Mai specific, **probleme de siguranţă** (*reliability*) sunt responsabilitatea nivelului Transport. În cadrul oferirii de servicii de comunicare, nivelul Transport iniţiază, gestionează şi închide **circuitele virtuale**.

Sarcina principală a nivelului Transport este aceea de refacere a fluxului de date (*flow control*) la destinaţie, deoarece un **pachet poate fi segmentat în mesaje mai mici**, cu rute diferite prin reţeaua de comunicaţii. Pentru a fi obţinută o comunicaţie sigură, servicii de detectare şi recuperare din erori sunt oferite tot la acest nivel.

Dacă doriţi să reţineţi **nivelul Transport** în cât mai puţine cuvinte, gândiţi-vă la *flow control*, la calitatea serviciilor şi la siguranţă [11].

Protocoalele utilizate *TCP* (*Transmission Control Protocol*), *UDP* (*User Datagram Protocol*), *RTP* (*Real-time Transport Protocol*), *SCTP* (*Stream Control Transmission Protocol*), *SPX* (*Sequenced Packet Exchange*) (vezi Anexa 1) [15, 16].

1.5.5. Nivelul 3: Reţea (*NetworkLayer*) - se ocupă de **controlul funcţionării subreţelei şi de transferul informaţiei organizate în pachete de date** între sursă şi destinaţie. Acesta este nivelul **cel mai important în cadrul Internetului,** asigurând posibilitatea interconectării diferitelor reţele. Tot la acest nivel se realizează **adresarea logică** a tuturor nodurilor din Internet.

Funcţia principală a acestui nivel constă în **dirijarea pachetelor între oricare două noduri de reţea**. Cu alte cuvinte, nivelul Reţea realizează „rutarea" (direcţionarea) pachetelor de date prin infrastructura de comunicaţii. Această operaţie fiind efectuată la nivelul fiecărui nod de comunicaţie intermediar. Nivelul Reţea asigură interfaţa între furnizorul de servicii şi utilizator, serviciile oferite fiind independente de tehnologia subreţelei de comunicaţie. Acest nivel oferă două categorii de servicii de transport: orientate pe conexiuni şi fără conexiuni[4]. La nivelul Reţea operează **ruterele**, dispozitivele cele mai importante în orice reţea de foarte mari dimensiuni.

[4] Termenul „fără conexiuni" înseamnă că atunci când o aplicaţie foloseşte *IPX* (*Internetwork Packet eXchange*) pentru a comunica cu alte aplicaţii din

Dacă doriţi să reţineţi **nivelul Reţea** în cât mai puţine cuvinte, gândiţi-vă la **selecţia drumului, rutare şi *IP*-uri** [11].

Protocoalele utilizate: *IP* (*Internet Protocol*), *ICMP* (*Internet Control Message Protocol*), *IGMP* (*Internet group management protocol*), *BGP* (*Border Gateway Protocol*), *EIGRP* (*Enhanced Interior Gateway Routing Protocol*), *ARP* (*Address Resolution Protocol*), *RARP* (*Reverse Address Resolution Protocol*), *X*.25 (*Packet Switching*) (vezi Anexa 1) [15, 16].

1.5.6. Nivelul 2: Legătură de date (*Data-Link Layer*) - gestionează transmisia **biţilor de date, organizaţi in cadre**, fără erori nedetectate, relativ la o anumită linie de transmisie.

Cadru de date reprezintă o structură logică, în care poate fi plasate (împachetate) date de transportat. Aceste secvente sunt marcate de **delimitatori de început/sfârşit**, delimitatorii care definesc astfel cadrul. Schimbul de cadre între sursă şi destinatar presupune trimiterea secvenţială a acestora urmată de cadre de confirmare a recepţiei.

Principalele atribuţii ale acestui nivel este **controlul erorilor, controlul fluxului informaţional** (*flow control*) şi **gestiunea legăturii**. Acest lucru presupune în cazul în care avem o conexiune *share-media* (în care mediul de transmisie este accesibil tuturor simultan şi este împărţit între staţii), detecţia şi corecţia cazurilor în care două staţii încearcă să transmită simultan (aşa-numitele coliziuni). Nivelul Fizic nu poate realiza acest lucru, deoarece nu putem vorbi despre nici un fel de date, ci numai despre biţi şi, mai exact, despre reprezentarea fizică a acestora (niveluri de tensiune, intensitatea luminii etc.). Pentru a realiza acest lucru, nivelul Legătură de date se ocupă cu **adresarea fizică, topologia reţelei, accesul la reţea**.

---

cadrul reţelei, între cele două aplicaţii nu se stabileşte nici o conexiune la nivelul „Legătură de date” (*OSI*, nivel 2).

Dacă doriţi să vă amintiţi **nivelul doi** în cât mai puţine cuvinte, gândiţi-vă la **cadre şi la controlul accesului la mediu** [11].

Nivelul 2 este împărţit în două subniveluri: ***LLC* şi *MAC*** - cu roluri diferite: **Subnivelul *LLC* (*Logical Link Control*)** - asigură **comunicarea** între nivelul Legătură de date şi nivelul Reţea. Acest subnivel este independent de tehnologie şi oferă funcţii ce sunt aceleaşi pentru orice variaţii ale nivelului Fizic şi ale subnivelului MAC.

Subnivelul *MAC* (*Media Acces Control*) - asigură **accesul ordonat la reţea**, controlează accesul şi delimitează cadrele, detectează erorile şi recunoaşte adresele, fiind inferior subnivelului *LLC*. ***MAC*** comunică direct cu interfata de reţea şi este responsabil pentru transportul fără erori al datelor intre două echipamente[5]. Acest subnivel este dependent de tehnologia *LAN* care este implementată[6].

Adresele *MAC* sunt asignate unic pe fiecare placă de reţea[7] şi nu pe fiecare calculator. Astfel, dacă unui calculator i se schimbă placa de reţea, adresa acestuia de *MAC* se va modifica. Adresele *MAC* nu pot fi modificate şi vor rămâne aceleaşi dacă calculatorul este mutat dintr-o reţea în alta.

Protocoalele utilizate *Ethernet*, *Token ring*, *FR* (*Frame relay)*, *ISDN* (*Integrated Services Digital Network*), *ATM* (*Asynchronous Transfer Mode*), *IEEE* 802.11 (*Wi-Fi*), *FDDI* (*Fiber Distributed Data Interface*), *ARP* (*Address Resolution Protocol*) (vezi Anexa 1) [15, 16].

---

[5] De exemplu, două staţii nu pot transmite în acelaşi timp, încercările de a transmite simultan sunt detectate în cadrul reţelelor de tip *broadcast*. În acest caz se realizează identificarea unui nod destinaţie şi se introduc delimitatorii necesari pentru separarea cadrelor, iar la recepţie, aceşti delimitatori se recunosc şi reconstituie cadrele.

[6] În cazul *Ethernet*-ului, este necesar un mecanism de detecţie a coliziunilor, iar în cazul *Token Ring* acest lucru nu mai este necesar.

[7] Un dispozitiv periferic care permite unui calculator să comunice cu alte dispozitive în reţea.

1.5.7. Nivelul 1: Fizic (*Physical Layer*) - defineşte specificaţiile **electrice, mecanice, procedurale şi funcţionale** pentru activarea, menţinerea şi dezactivarea legăturilor fizice între sisteme. În această categorie de caracteristici se încadrează **nivelurile de tensiune**, ***timing***-ul schimbărilor acestor niveluri, **ratele** de transfer fizice, **distanţele** maxime la care se poate transmite şi alte atribute similare care sunt definite de specificaţiile fizice: **fire de cupru, fibre optice, emiţătoare**, **receptoare** ce sunt folosite pentru a transmite date. Aceste date sunt de fapt ***mail*-uri, filme, *mp3*-uri, poze, fişiere text.** Datele sunt convertite în *biţi* care sunt transmişi prin aceste medii fizice. Fiecare dintre ele este definit de **lărgimea sa de bandă, întârziere, cost şi uşurinţa de instalare şi de întreţinere**.

Dacă doriţi să reţineţi **nivelul Fizic** în cât mai puţine cuvinte, gândiţi-vă la **semnale şi la mediu de transfer** [11].

Verifică-ţi cunoştinţele:

1) Definiţi noţiune de Standart.
2) Explicaţi cauzele elaborării modelului *ISO-OSI*.
3) Avantajul acestui gen de model.
4) Caracterizaţi nivelurile modelului *ISO-OSI*.

1.6. Modelul de referinţă *TCP/IP*

Deşi modelul *OSI* este general recunoscut, standardul istoric şi tehnic pentru Internet este *TCP/IP* (*Transmission Control Protocol*/ *Internet Protocol*). Modelul *TCP/IP* a fost creat de *US DoD* (*US Department of Defence* - Ministerul Apărării Naţionale al Statelor Unite) din necesitatea unei reţele care ar putea supravieţui în orice condiţii [17].

Nivelurile modelului *TCP/IP* :

Modelul TCP/IP are patru niveluri: **Aplicaţie**, **Transport**, **Reţea** (sau Internet) şi **Acces la reţea** (vezi Fig. 1.3).

Nivelul 4: Aplicaţie - nu este identic cu cel din modelul *ISO-OSI*, dimpotrivă, include ultimele trei niveluri superioare din stiva *ISO-OSI*. Acestea au fost comasate pentru a putea fi tratate la un loc toate problemele legate de protocoale de nivel înalt, fie de **reprezentare, codificare sau control al dialogului**.

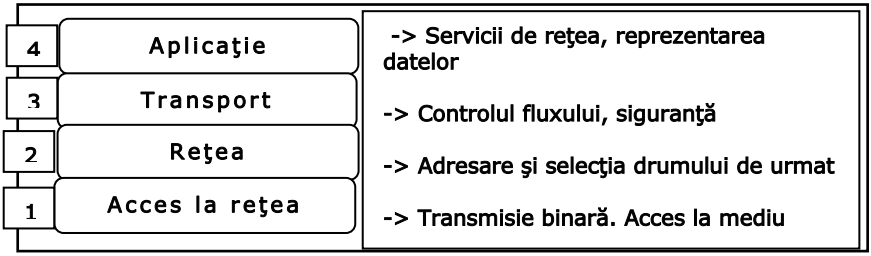


Fig. 1.3. Modelul *TCP/IP* [5]

Nivelul 3: Transport - este identic cu cel din modelul *ISO-OSI*, ocupându-se cu probleme legate de **siguranţă, control al fluxului şi corecţie de erori**.

Nivelul 2: Reţea. Scopul nivelului Reţea (Internet) este de a asigura **transmiterea pachetelor** de la orice sursă din reţea şi **livrarea** lor către o destinaţie independent de calea şi reţelele pe care le-a străbătut pentru a ajunge la destinator. **Determinarea drumului optim** şi comutarea pachetelor au loc la acest nivel.

Nivelul 1: Acces la reţea - se ocupă cu toate problemele legate de **transmiterea efectivă a unui pachet** *IP* pe o legătură fizică, incluzând şi aspectele legate de tehnologii şi de medii de transmisie, adică nivelurile **Legătură de date şi Fizic**.

Verifică-ţi cunoştinţele:

1) Explicaţi necesitatea elaborării modelului *TCP/IP*.
2) Avantajul acestui gen de model.
3) Caracterizaţi nivelurile modelului *TCP/IP*.

1.7. Comparaţia modelelor *ISO-OSI* şi *TCP/IP*

Deşi atât *ISO-OSI* cât şi *TCP/IP* modelează acelaşi lucru, şi anume procesul de comunicare între două entităţi, apare o întrebare: care din ele este mai bun? Cu acest scop evidenţiem deosebirile fiecărui model (vezi Fig. 1.4) [18].

Primă **deosebire** constă în aceea că *ISO-OSI* permite explicarea oricărui proces de comunicare, în timp ce *TCP/IP*-ul reuşeşte să modeleze perfect numai procesul de comunicare folosit în Internet.

1. O asemănare între cele două modele o reprezintă faptul că ambele conţin o stivă de niveluri care sunt legate între ele prin noţiunea de serviciu, interfaţă şi protocol.

Dacă *ISO-OSI* reuşeşte să facă o **distincţie clară** între aceste trei elemente, pentru *TCP/IP* ele nu reprezintă deloc un element vital.

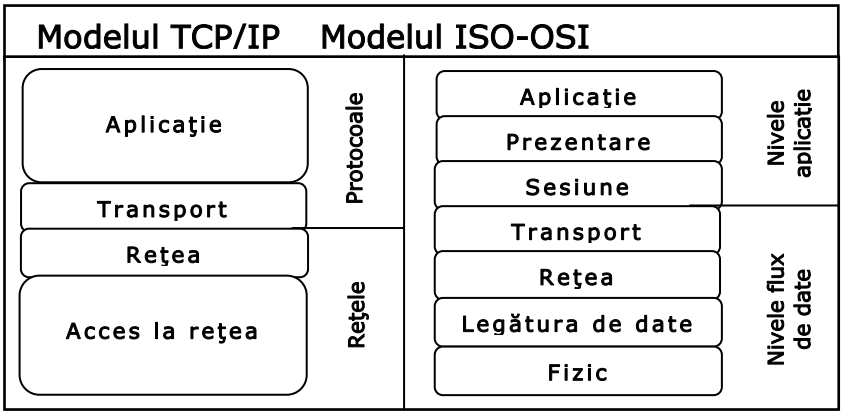


Fig. 1.4. Comparaţia modelelor *ISO-OSI* şi *TCP/IP*

2. Ambele modele au o răspândire largă. Modelul *ISO-OSI* permite **explicarea teoretică** a oricărui proces de comunicare, *TCP/IP*-ul joacă rolul de bază în reglarea **procesului referit la Internet**.

3. ***ISO-OSI*** - reprezintă un **model ideal**, deoarece ajută la realizarea unor paşi rapizi în evoluţia comunicării în plan teoretic. Pe de altă parte, ***TCP/IP*** se utilizează la descrierea unei **situaţii practice**.

4. Din **punct de vedere tehnic** o diferenţă evidentă dintre acele două modeluri o reprezintă faptul că nivelurile superioare prezente în *ISO-OSI* sunt **comasate** într-unul singur la *TCP/IP*. Acest lucru însă nu neagă existenţa unor niveluri ca Sesiune sau Prezentare, ci doar demonstrează că ele sunt specifice pentru diverse aplicaţii.

5. O altă deosebire de **ordin tehnic** care complică *ISO-OSI* e faptul că anumite operaţii, cum ar fi de exemplu verificările de integritate, sunt realizate de mai multe ori în cadrul unor niveluri diferite.

Verifică-ţi cunoştinţele:

1) Comparaţi modelele *ISO-OSI* şi *TCP/IP*
2) Descrieți deosebirile și asemănările modelelor.

Întrebările pentru autoevaluare:

1. Ce reprezintă reţele de calculatoare?
2. Enumeraţi tipurile de reţea.
3. Descrieți componentele generale ale unei reţele
4. Daţi definiţia procesului de încapsulare
5. Clasificați şi caracterizați tipurile de reţele
6. Descrieți topologiile reţelelor de calculatoare
7. Enumeraţi protocoale de comunicaţie în reţea
8. Explicaţi modelul de referinţă *ISO-OSI*
9. Analizaţi succint nivelurile modelului *ISO-OSI*
10. Modelul de referinţă *TCP/IP*
11. Comparaţi modelele *ISO-OSI* şi *TCP/IP*.

Capitolul 2. Tehnici şi medii de transmisie la nivel Fizic

2.1. Funcţiile definite în cadrul nivelului Fizic
- 2.1.1. Funcţiile de bază ale nivelului Fizic
- 2.1.2. Subnivelurile nivelul Fizic

2.2. Semnal şi zgomot în sistemele de comunicaţie
- 2.2.1. Semnalele analogice
- 2.2.2. Semnalele digitale
- 2.2.3. Fenomene care pot influenţa calitatea semnalului
  - a) Atenuare
  - b) Reflexia
  - c) Zgomotul
  - d) *Crosstalk*
  - e) Latenţa
  - f) Coliziuni

2.3. Tehnica transmiterii semnalului
- 2.3.1. Multiplexarea
- 2.3.2. Transmisia *baseband* şi *broadband*

2.4. Medii de transmisie
- 2.4.1. Firele de cupru
  - a) Cablurile torsadate (*UTP*, *STP*)
  - b) Cablul coaxial
- 2.4.2. Fibra optică
- 2.4.3. Comparaţia dintre fibrele optice şi firul de cupru
- 2.4.4. Sistemele fără fir

La baza tuturor reţelelor de calculatoare se află **nivelul Fizic**. Nivelul Fizic defineşte specificaţii **electrice, mecanice, procedurale şi funcţionale** pentru activarea, menţinerea şi dezactivarea legăturilor fizice între sisteme. Scopul principal al acestui nivel este de a **transmite o secvenţă de biţi** de la un calculator la altul utilizând diverse medii fizice. **Unitatea de date: *bit*-ul**.

2.1. Funcţiile definite în cadrul nivelului Fizic

Un standard de nivel Fizic defineşte 4 tipuri de caracteristici:

- **Mecanice** - forma şi dimensiunile conectorilor, numărul de pini;
- **Electrice** - modulaţia, debite (fluxuri) binare, codări, lungimi maxime ale canalelor de comunicaţie;
- **Funcţionale** - funcţia fiecărui pin;
- **Procedurale** - succesiunea procedurilor pentru activarea unui serviciu.

La nivelul Fizic se determină: cablaje şi mediul de transmisie, dispozitive de conectare la acestea, semnalele implicate în transmiterea/recepţia datelor, posibilitatea de a determina erorile de semnal la nivelul Fizic.

Nivelului Fizic nu are nici un mecanism pentru determinarea semnificaţiei biţilor pe care îi transmite sau îi primeşte, ci este preocupat exclusiv de **caracteristicile fizice ale tehnicilor de transmitere a semnalelor electrice şi/sau optice**.

2.1.1. Funcţiile de bază ale nivelului Fizic sunt [19, 20]:

1. Stabilirea tipului de transmitere şi recepţionare a şirurilor de biţi pe un canal de comunicaţii:

- **Transmisia asincronă**: semnalul de ceas al receptorului se **sincronizează pe semnalul de strat (de bază) transmis de emiţător**. Din această cauză, canalul de comunicaţie nu este utilizat eficient şi nu se pot obţine rate (cote) de transfer mari, de **maxim *115 Kbps***. Este frecvent utilizată pentru conectarea a două echipamente de reţea prin intermediul **cablurilor seriale sau a *modem*-urilor analogice**.

- **Transmisia sincronă**: şirurile de biţi se succed fără întrerupere, **fiecare echipament având nevoie de un semnal de sincronizare propriu**. De aceea, receptorul este mai complicat, însă se asigură o utilizare eficientă a canalului de comunicaţie şi se pot obţine **viteze mari de transfer (*2 Mbps*)**.

2. Definirea topologiilor de reţea şi în funcţie de topologie - **stabilirea tipului reţelei**:

- **Reţea *broadcast*** se stabileşte în cazul topologiilor de tip: **magistrală, stea, inel**. Pentru acest tip de reţea la acelaşi mediu de transmisiune pot fi ataşate mai multe echipamente de reţea, iar un **pachet de date transmis de o staţie este recepţionat de toate celelalte** (de exemplu, *Ethernet/Fast Ethernet, Token Ring*)

- **Reţele punct-la-punct** - în cazul topologiilor de tip: **stea, plasă.** Pentru acest tip de reţea la o conexiune fizică sunt ataşate numai două echipamente. Într-o reţea cu mai mult de două noduri, un pachet de date trebuie să tranziteze mai multe noduri intermediare pentru a ajunge la destinaţie.

3. Definirea tipurilor de medii de transmisiune: cablu coaxial, cablu *UTP*, fibră optică, linii de cupru etc.

4. Stabilirea modului de transmisie:

- ***simplex*** (un singur echipament poate transmite, iar corespondentul doar recepţionează);

- ***half-duplex*** (ambele echipamente pot să transmită şi să recepţioneze semnale, dar nu în acelaşi timp);

- ***full-duplex*** (ambele echipamente pot să transmită şi să recepţioneze semnale în acelaşi timp).

5. Definirea standardelor mecanice şi electrice ale interfeţelor seriale (*RS-232*, *V*.35, *G.703*) şi *LAN* (*BNC, AUI, RJ45*).

6. Codificarea şi decodificarea şirurilor de biţi. De-a lungul timpului au existat numeroase forme de transport al informaţiei. Fiecare dintre aceste metode avea o anumită formă de **codare a informaţiei**. Telefoanele, fax-urile, radio *AM* şi *FM*, toate folosesc propriul lor sistem de codare electronică a informaţiei.

7. Modularea şi demodularea semnalelor purtătoare (*modem*-uri). De exemplu, transmisia de date în sistemele fără fir se realizează folosind **o undă purtătoare ca bază de frecvenţă**, urmând ca aceasta să fie **modificată prin modulare** pentru a codifica datele. În acest caz

pentru o undă purtătoare există trei mărimi care pot fi modificate pentru modulare: **amplitudinea** (rezultă modulare în amplitudine, *AM*); **frecvenţa** (modulare în frecvenţă, *FM*); **faza** (modulare în fază, *PM*).

2.1.2. Subnivelurile nivelul Fizic. Nivelul Fizic se împarte în două subnivelurile [21]:

- *PLS – Physical Signaling Sublayer* (Subnivelul de Semnalizare Fizic);
- *PMA – Physical Medium Attachement* (Subnivelul de Ataşare la Mediul Fizic).

Subnivelul *PLS* este responsabil cu **codificarea datelor ce sunt plasate în jetoane de la nivelul *MAC*** la o staţie care transmite. Codificarea datelor impune transformarea biţilor în semnale electrice pentru transmisia jetoanelor mediului fizic propriu-zis. La staţia destinaţie *PLS* decodifică semnalele recepţionate şi le transformă din nou în biţi de date ce sunt plasaţi spre subnivelul MAC.

Subnivelul *PMA* oferă servicii subnivelului *PLS*, realizând funcţia de adaptare între subnivelul *PLS* şi mediul de transmisie propriu-zis şi defineşte caracteristicile unui mediu particular de transmisie.

Exemple de **protocoale**: *IEEE 802.3 Ethernet*, *IEEE 802.5 Token Ring*, *IEEE 802.11* (fără fir).

Verifică-ţi cunoştinţele:

1) Enumeraţi tipurile de caracteristici definit de către un standard de nivel Fizic.
2) Caracterizaţi funcţiile de bază ale nivelului Fizic.
3) Analizaţi subnivelurile nivelul Fizic.
4) Daţi noţiunea de unitatea de date la nivel Fizic.

2.2. Semnal şi zgomot în sistemele de comunicaţie

Un semnal constă din mai multe **impulsuri electrice sau luminoase** ce sunt transmise de la un echipament la altul folosind ca suport un anumit mediu de transmisie.

Din punct de vedere al modului de transmisie şi al suportului folosit, **semnalele se împart în trei mari categorii** [22, 23]:

- **semnale electrice** - reprezentate de impulsuri electrice ce folosesc ca suport pentru transmisie fire de cupru;

- **semnale optice** - convertesc semnalul electric primit în impulsuri luminoase pe care le transmit folosind o fibră optică;

- **semnale *wireless*** (fără fir) - unde radio, microunde.

Indiferent de suportul de transmisie folosit, **semnalele pot fi analogice sau digitale.**

2.2.1. Semnalele analogice (vezi Fig. 2.1) - sunt sunete pe care le auzim (**vocea umană, ciripit de păsări, sgomot** etc.). Atunci când le reprezentăm le observăm graficul, vedem că seamănă cu nişte valuri mai mult sau mai puţin simetrice. Cel mai simplu exemplu de semnal analogic este o sinusoidă. În cadrul unui semnal analogic nu există treceri bruşte de la o valoare la alta.

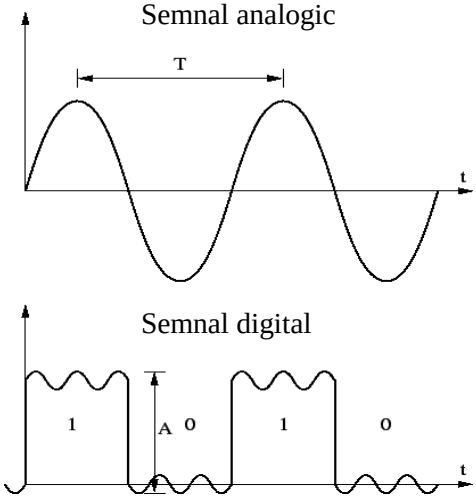


Fig. 2.1. Comparaţie între semnalele digitale şi analogice [5]

2.2.2. Semnalele digitale (vezi Fig. 2.1) - sunt cele folosite în tehnică şi au **la bază două valori logice, 0 şi 1**, care au fiecare câte o reprezentare în funcţie de modul în care sunt transmise. Impulsurile digitale (0 sau 1 logic) se numesc **biţi**.

Transmisia digitală este mai puţin afectată de zgomote. Tipurile de semnale cele mai folosite în reţele de calculatoare sunt în marea lor majoritate digitale. Există numeroase cazuri în care datorită interferenţelor prea mari se emite 0 şi se recepţionează 1 sau invers [24, 25].

2.2.3. Fenomene care pot influenţa calitatea semnalului:

În timpul transmisiei unui semnal apar diferite fenomene care pot influenţa calitatea semnalului: **Atenuarea, Reflexia, Zgomotul, *Crosstalk*-ul, Latenţa, Coliziunile**.

a) Atenuarea - se referă la reducerea puterii unui semnal şi este indiferent de tipul de semnal, analogic sau digital (vezi Fig. 2.2). Atenuarea afectează reţelele de calculatoare, limitează distanţa maximă şi din acest caz nu se va mai putea interpreta semnalul corect.

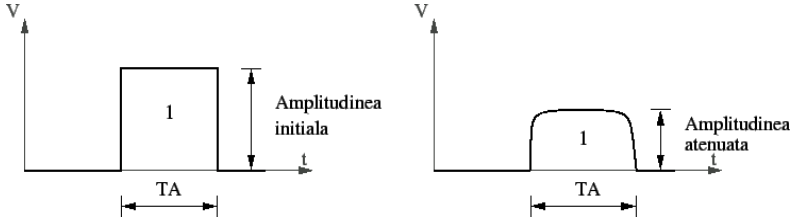


Fig. 2.2. Atenuarea semnalului [11]

Pentru transmisie la distanţe mai mari decât permite un tip de cablu, se folosesc anumite dispozitive, numite **repetoare**, care regenerează semnalul (din punct de vedere electric, optic sau *wireless*). Atenuarea afectează toate tipurile de medii de transmisie, însă are valori diferite pentru fiecare mediu în parte. Un semnal electric se atenuează mai repede transmis pe un fir de cupru decât un semnal optic pe o fibră optică.

Atenuarea în general se măsoară în decibeli (***dB***), iar atenuarea specifică unui anumit tip de cablu - în **decibeli/metru** sau **decibeli/kilometru**.

Fiecare tip de cablu are o atenuare specifică. Cu cât această atenuare specifică este mai mică, cu atât acel cablu este considerat mai bun. **Pentru determinarea distanţei** la care se poate „merge" cu transmisia se foloseşte formula:

$$\text{Distanţă maximă} = \frac{\text{aten. maxim permis de echipament} - \text{aten. introdus de conectori}}{\text{aten. specific mediului de transmisie}}$$

b) Reflexia - are loc atunci când un semnal întâlneşte o linie de separaţie între două medii. Atunci o parte din semnal se reflectă înapoi în mediul din care a venit şi o parte trece în mediul următor. În lumea reală, milioane de biţi sunt transmişi în fiecare secundă, iar această energie reflectată poate duce la multe transmisii nereuşite.

Reflexia poate avea loc şi în cazul **sistemelor optice**. Un semnal optic se reflectă ori de câte ori întâlneşte o discontinuitate în fibra de sticlă, ca de exemplu atunci când **ataşăm un conector**. De aceea este necesară o pregătire specială în cazul ataşării conectorilor de fibră optică, pentru a nu permite reflexia luminii înapoi în fibră.

c) Zgomotul (vezi Fig. 2.3) - este o cantitate de energie nedorită (**electrică, electro-magnetică sau radio**) care poate degrada calitatea semnalului transmis. Zgomotul apare atât în transmisiile **analogice** cât şi în cele **digitale**. În cazul semnalelor analogice, semnalul devine uşor deformat. În sistemele digitale, zgomotele afectează valorile biţilor transmişi (0 sau 1), iar la destinaţie aceştia pot fi „citiţi" greşit (adică 1 in loc de 0 şi invers).

Zgomotul poate avea mai multe cauze. Una dintre ele o reprezintă câmpurile electrice provenite de la **motoare electrice, lumina fluorescentă** (neoane) etc., toate provenite de la surse exterioare cablului afectat. Acest tip de zgomot se numeşte *EMI* (*Electromagnetic Interference* - Interferenţă Electromagnetică) dacă provine de la surse electrice sau *RFI* (*Radio Frequency Interference* -

Interferenţă Radio) când provine de la surse radio, radar sau microunde. Astfel, **fiecare fir dintr-un cablu poate acţiona ca o antenă**. Zgomotul mai poate proveni de la liniile de **curent alternativ** sau de la **fulgere**.

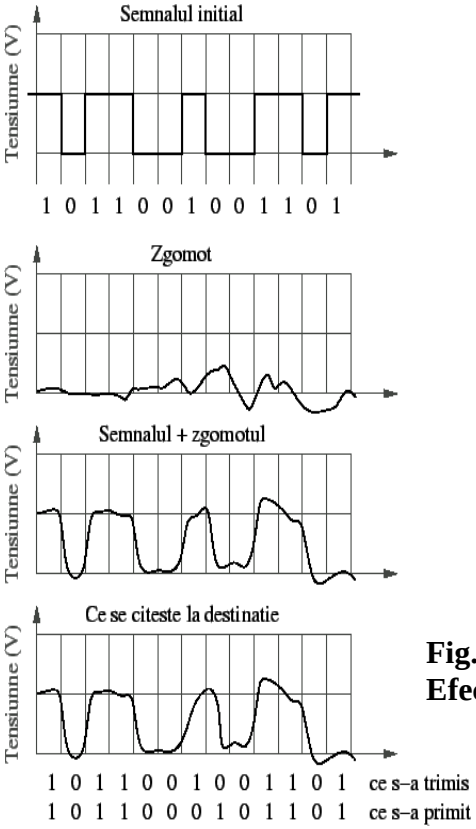


Fig. 2.3.
Efectul zgomotului

Sistemele optice şi *wireless* sunt afectate de unele dintre aceste tipuri de zgomot, însă sunt imune la altele. De exemplu, **transmisia optică este imună la interferenţele electrice**. Acest lucru le face **ideal pentru legăturile din exteriorul clădirii**, unde firele de cupru ar putea fi influenţate de fulgere, câmpuri electrice din alte surse etc.

d) *Crosstalk* - cablurile de cupru sunt afectate de **interferenţe electromagnetice** de la diferite surse din afara cablului. Cea mai importantă sursă de zgomot pentru cablurile de cupru o reprezintă „**scurgerea**” **unui semnal între două fire din interiorul aceluiaşi cablu**. Una dintre cele mai eficiente metode de prevenire a *crosstalk*-ului este **torsadarea (sucirea) firelor**. Prin torsadare, câmpurile electrice se anulează şi firele din celelalte perechi nu mai sunt influenţate de semnalul din perechea iniţială.

De multe ori apar probleme la **ataşarea conectorilor**: mai ales când este necesar de ataşat un conector la capătul unui cablu, trebuie întâi de detorsadat toate perechile din interiorul cablului. Dacă se lasă o bucată prea mare detorsadată, în acea zonă câmpurile electrice generate de fiecare fir dintr-o pereche nu se vor mai anula şi va apărea o interferenţă între fire, numită *Near-end crosstalk*. Acest parametru este specific fiecărui cablu.

e) Latenţa - numită şi **întârziere**, este de două tipuri: **(a) latenţa propagării prin mediul de transmisie (**este dat de timpul de propagare a **unui singur bit** de la sursă la destinaţie) **şi (b) latenţa trecerii prin echipamentele de reţea** (acest fenomen poate fi explicat prin faptul că fiecare echipament execută anumite operaţii, mai simple sau mai complexe, iar aceste operaţii introduc o anumită latenţă). **Cu cât este mai rapid modul de transmitere, cu atât este mai redusă latenţa.**

f) Coliziuni - are loc atunci când 2 biţi de la două staţii diferite care transmit se află pe acelaşi mediu de transmisie, în acelaşi timp. În cazul **firelor de cupru**, tensiunile celor două semnale binare se adună generând astfel un al treilea nivel de tensiune. Această variaţie a tensiunii nu este permisă în sistemul binar care înţelege doar două niveluri de tensiune. Aceşti biţi sunt în consecinţă „greşiţi” şi „distruşi”. Apariţia coliziunilor excesive poate încetini foarte mult viteza de transmisie. De aceea, unul dintre scopurile proiectării unei reţele este de a minimiza pe cât posibil prezenţa coliziunilor.

Există mai multe **metode de a aborda problema coliziunilor**. O variantă este de a permite apariţia lor, de a le detecta şi de a avea un set de reguli pentru tratarea acestora atunci când apar - cum este cazul *Ethernet*-ului. Altă variantă este de a evita apariţia coliziunilor, de exemplu, prin a permite unei singure staţii să transmită pe acelaşi mediu în acelaşi timp. Acest lucru necesită ca staţia să aibă un şablon sau tipar (*pattern*) special de biţi, numit **jeton** (*token*), pentru a transmite. Această situaţie o întâlnim în cazul *TokenRing*.

Verifică-ţi cunoştinţele:

1) Precizaţi categoriile în care din punct de vedere al modului de transmisie şi al suportului folosit se împart semnalele.

2) Caracterizaţi semnale analogice şi digitale.

3) Care fenomene pot apărea în timpul transmisiei unui semnal? Caracterizaţi-le.

2.3. Tehnica transmiterii semnalului

Prin **tehnica transmiterii semnalului** se înţelege maniera în care semnalele sunt transportate pe reţea.

2.3.1. ***Multiplexarea*** - este procedeul prin care **mai multe canale de date sunt combinate într-un singur canal fizic**. Demultiplexarea este procedeul de separare a canalelor iniţiale din canalul primit multiplexat [26, 27, 28].

Există numeroase **tehnici de *multiplexare*** dintre care menţionăm:

TDM - *Time Division Multiplexing* - informaţiilor din fiecare canal li se alocă o cuantă de timp predefinită, indiferent dacă pe acele canale se transmite sau nu.

ATDM - *Asynchronous time-division multiplexing* - informaţiilor din fiecare canal li se alocă o cuantă de timp variabilă, în funcţie de numărul de canale utilizate în acel moment.

FDM - *Frequency Division Multiplexing* - fiecărui canal i se alocă o anumită bandă de frecvenţă.

SM - *Statistical Multiplexing* - banda este alocată în mod dinamic fiecărui canal care are informaţii de transmis.

DWDM - *Dense Wavelength Division Multiplexing* - este o formă de *multiplexare* dezvoltată pentru transmisia pe fibră optică. *DWDM* este echivalentul optic al *multiplexării FDM*.

2.3.2. Transmisia *baseband* şi *broadband*. Tehnicile transmisiei sunt: ***baseband*** şi ***broadband***. Termenii de „*baseband*" (în bandă de bază) şi „*broadband*" (în bandă largă) descriu numărul de „canale" de comunicaţie folosite pe un anumit mediu de transmisie [29, 30, 31].

a) Semnalele ***baseband*** folosesc **întreaga bandă de frecvenţă** pentru transmiterea informaţiei, iar transmiterea simultană a mai multor seturi de date se face prin tehnica de ***multiplexare* în timp**. În cazul comunicaţiei *baseband*, pe mediul de transmisie **avem un singur semnal**. **Acest semnal poate avea mai multe componente**, însă din punct de vedere al firului (firului de cupru sau al fibrei optice), reprezintă un singur semnal (electric sau optic). Majoritatea comunicaţiilor în cazul *LAN*-urilor şi a sistemelor de telefonie fixă sunt *baseband*.

b) În sistemele de transmisie ***broadband*** semnale multiple (voce, date, semnal video) sunt transmise simultan pe acelaşi suport fizic folosindu-se **tehnica de *multiplexare* în frecvenţă**. Astfel, termenul de bandă largă este în general folosit pentru a descrie **accesul la Internet**. Prin bandă largă, se poate primi telefoane pe aceeaşi linie telefonică, care este folosită pentru conectarea la Internet, simultan cu navigarea pe Internet şi în acelaşi timp cu vizionarea unei emisiuni *TV*, de exemplu, *CATV* (*Community Antenna Television*) - sistemul de televiziune prin cablu.

Verifică-ţi cunoştinţele:

1) Ce se înţelege prin tehnica transmisiei semnalului?
2) Explicaţi noţiunea de *multiplexare*, transmisia *baseband* şi *broadband*.

2.4. Medii de transmisie

Medii de transmisie a datelor sunt folosite de diferitele tehnologii pentru transportul semnalelor care determină **ce transmitem, cât de mult, cât de departe.**

Medii de transmisie pot fi clasificate în două categorii mari: **medii ghidate si medii neghidate.** Mediile ghidate cuprind: **cablul de cupru şi fibrele optice**, iar cele neghidate, **undele radio** şi **laserul** [32].

Fiecare din mediile de transmisie sunt definite de o serie de caracteristici, care influenţează asupra alegerii suportului de transmisie: **lărgimea de bandă; întârzierile; costul cablării; facilităţile de racordare a echipamentelor; tip de conectivitate; fiabilitatea suportului; protecţia faţă de imunitatea la zgomot; facilităţile de instalare şi întreţinere,** etc.

Cablurile reprezintă un **suport fizic** pentru această transmisie, dar însă pot introduce limitări:

Limitări tehnologice - se referă la probleme de nivel Fizic, de exemplu, atenuarea.

Limitări de tehnologie - se referă la probleme de nivel Legătura de date, fiind independente de mediul de transmisie ales (de nivelul 1), de exemplu, lăţimea de bandă, care în cazul *FastEthernet*-ului este maxim ***100 Mbps*** indiferent de cablul ales.

Metodele de transmisie sunt în continuă dezvoltare şi deja foarte diverse, începând cu tot felul de cabluri metalice şi de fibră optică, chiar submarine, şi terminând cu legături fără fir prin unde radio cum ar fi ***Wi-Fi, WiMAX*** sau ***Bluetooth***, prin raze infraroşii sau prin intermediul sateliţilor de telecomunicaţii.

În continuare vor fi prezentate diferite **categorii de medii de transmisie** şi vor fi analizate limitările pe care acestea le impun unor tehnologii cunoscute.

2.4.1. Firele de cupru - firele de cupru reprezintă cel mai vechi suport utilizat. Marea parte a reţelelor de date folosesc fire de cupru în diferite forme, niveluri de calitate etc. **Transmisia pe fire de cupru se bazează pe propagarea unui semnal electric,** care trebuie să rămână între anumiţi parametri specificaţi de tehnologie, pe parcursul drumului între sursă şi destinaţie.

În funcţie de structura lor şi de parametrii specifici ai mediului de transmisie, cablurile de cupru se împart în două mari categorii: **torsadate şi coaxiale**.

a) Cablurile torsadate (*UTP, STP*) - previn interferenţele între câmpurile electrice cauzate de transmisia datelor la frecvenţe mai mari. **Un cablu torsadat** este format din mai multe perechi compuse din fire de cupru izolate, având o **grosime tipică de 1 mm**. Firele sunt împletite într-o formă elicoidală, pentru a reduce interferenţa electrică[8]. Interferenţele pot fi cauzate de câmpurile electrice induse de alte fire din interiorul aceluiaşi cablu sau de surse exterioare [11].

Metodele prin care se încearcă reducerea la minim a acestor interferenţe sunt mai multe, dintre care menţionăm [33]:

- torsadarea cablurilor două câte două, formându-se astfel mai multe perechi în interiorul cărora câmpurile electrice create de cele două fire se anulează;
- transmiterea semnalului în mod balansat[9] (vezi Fig. 2.4);
- ecranarea cablurilor.

[8] Două fire paralele constituie o antenă; dacă le împletim nu mai formează o antenă.

[9] Semnalul util se transmite ca fiind diferenţa între semnalele electrice dintre cele două fire din cadrul unei perechi; în acest fel, atunci când apar interferenţe electrice de la surse exterioare cablului, acestea afectează în mod egal ambele fire, astfel încât diferenţa dintre acestea rămâne constantă, semnalul fiind nealterat.

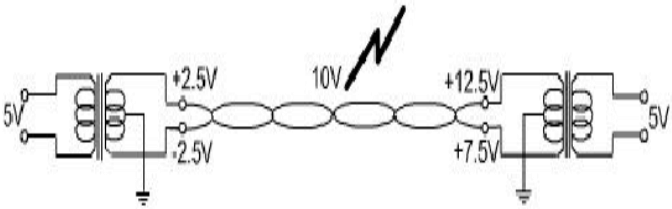


Fig. 2.4. Transmisie diferenţială [5]

Din punct de vedere al ecranării, există două categorii de cabluri torsadate:

- neecranate (*UTP - Unshieded Twisted Pair*)
- ecranate (*STP - Shielded Twisted Pair*).

Cablurile torsadate se împart în **solide şi liţate**. Cele solide conţin în interiorul celor 8 fire din cablu un singur fir de cupru şi sunt folosite la **cablările verticale** (acolo unde de obicei este nevoie de cabluri rigide) [34, 35].

Cablurile liţate au în interiorul celor **8 fire** mai multe fire foarte subţiri, numite **liţe**, ceea ce face acest tip de **cablu foarte flexibil** şi deci potrivit pentru **cablările orizontale** (de la priza de perete până la staţia utilizatorului), fiind şi mult mai uşor de sertizat.

- ***UTP* (*Unshieded Twisted Pair*)** (Fig. 2.5)**.** Cele **neecranate** se numesc ***UTP*** şi sunt **cele mai folosite** în cadrul reţelelor locale de calculatoare, fiind de altfel şi cele mai ieftine. Marea majoritate a cablurilor *UTP* conţin **4 perechi colorate** [36] .

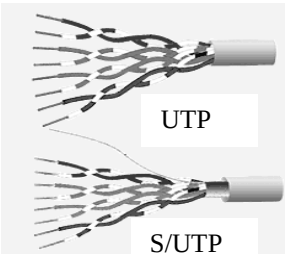
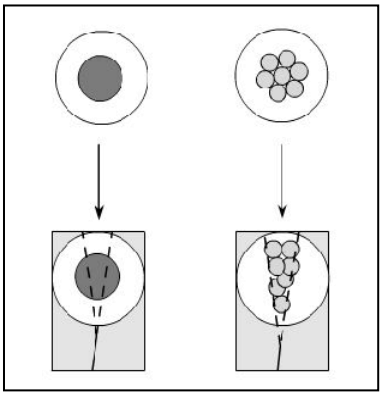


Fig. 2.5. Cablu *UTP*



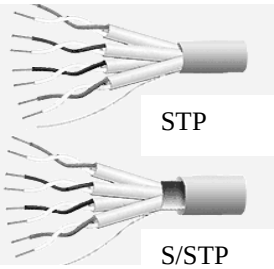


Fig. 2.6. Cablu *STP*

Conectorii folosiţi pentru cablurile torsadate sunt definiţi de standarde sub numele de ***8p8c*** (8 positions, 8 contact), însă sunt cunoscute mai ales sub numele de ***RJ45*** (*Registered Jack 45*), asemnător cu cel de la firul telefonic. Conectorul este construit

conform unui standard din industria telefonică, standard care precizează care fir trebuie să fie conectat pe un anumit pin al conectorului.

Referitor la parametrii impuşi de diferitele categorii de cabluri, o mare atenţie se acordă în ultima perioadă **normelor de siguranţă** pe care cablurile trebuie să le respecte. De exemplu, cablurile *UTP* sunt cele mai cunoscute, cu învelişul exterior din *PVC*. Acestea sunt mai ieftine, sunt rezistente la apă, însă au marele dezavantaj că atunci când iau foc degajă substanţe foarte toxice; ele nu pot fi folosite în exteriorul clădirilor, deoarece ar fi supuse unor posibile **şocuri electrice foarte mari**, care ar cauza defectarea echipamentelor conectate cu aceste cabluri.

- ***STP* (*Shielded Twisted-Pair*)** (Fig. 2.6)**.** Pentru a evita neajunsurile cablurilor *UTP*, atunci în exteriorul clădirilor se poate folosi cablu ecranat *STP* sau *ScTP* (*screened twisted pair*). *ScTP* are un singur înveliş de ecranare exterior şi o dimensiune puţin mai mare decât *UTP*. *STP*-ul are, pe lângă învelişul de ecranare identic cu cel de la *ScTP* de asemenea are şi un **înveliş separat pentru fiecare pereche**. Acest lucru îl face **mult mai rezistent la intereferenţele electrice exterioare**, dar în acelaşi timp **mai scump**, mai mare ca dimensiuni şi în consecinţă mai greu de utilizat[10] [37].

b) Cablul coaxial a fost folosit încă de la începutul reţelelor de

[10] Cablul *STP* de 100 Ohm folosit în reţelele *Ethernet*, oferă rezistenţă atât la interferenţele electromagnetice, cât şi la cele radio fără a fi un cablul prea gros. În reţelele *Token Ring* se foloseşte cablul *STP* de 150 Ohm, în care fiecare pereche de fire torsadate este izolată cu un înveliş protector pentru a se reduce posibilitatea transferului semnalului în alte fire (*crosstalk*). Învelişul protector folosit în cablul de 150 Ohm nu face parte din circuit aşa cum se întamplă în cazul cablului coaxial.
Chiar dacă este mai scump decat *UTP*, cablul *STP* oferă protecţie împotriva tuturor tipurilor de interferenţe. O conectare incorectă face ca învelişul protector să acţioneze ca o antenă, absorband semnalele electrice din cablurile aflate în vecinătate.

calculatoare, fiind **foarte uşor de instalat**. În zilele noastre, cablul coaxial nu mai este implementat în reţele locale (deşi mai este încă găsit în multe „reţele de bloc”), însă este în continuare folosit în transmisia video şi *CATV* (televiziune prin cablu) [38, 39].

Un cablu coaxial este format (vezi Fig. 2.7) dintr-o **sârmă de cupru dură** (D), protejată de un **material izolant** (C). Acest material este încapsulat într-un **conductor circular** (B), de obicei sub forma unei plase strâns întreţesute. Conductorul exterior este acoperit cu un înveliş de plastic **protector** (A), acesta fiind şi proveniența denumirii de „*co-axial*” (datorită acestei axe unice date de miezul de cupru).

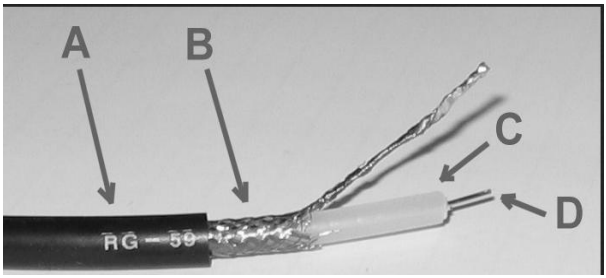


Fig. 2.7. Cablu coaxial

Datorită structurii sale şi a izolării foarte bune, cablul coaxial prezintă **două avantaje majore** faţă de alte tipuri de cablu de cupru: în primul rând o **comportare foarte bună în frecvenţă**, în al doilea rând **poate acoperi o bandă foarte largă**, de la frecvenţe joase până la *UHF* (*Ultra High Frequency*). În televiziunea analogică, există mai multe benzi de frecvenţă, pe care „emit” posturile TV[11].

[11] Când căutaţi manual un post *TV*, sunteţi pe o anumită bandă de frecvenţă, care poate fi *VHF* (*Very High Frequency*), *UHF* etc. Dintre acestea, *UHF* este cea mai mare, însă sunt relativ puţine posturi care emit pe *UHF*, ceea ce îl face ideal pentru transmisii de video analogic (televiziune prin cablu), însă şi pentru

Dezavantajul îl constituie faptul că nu suportă pentru *Ethernet* **o lăţime de bandă mai mare de *10 Mbps***, ceea ce este mult prea puţin pentru cerinţele reţelelor actuale, motiv pentru care în acest domeniu **a fost înlocuit cu cablul torsadat**. Un alt dezavantaj constă in aceea că **este un mediu partajat** (*shared-media*) şi nu poate oferi un grad minim de **securitate**[12]; pentru o imunitate bună la interferenţele electromagnetice **cablul trebuie împământat doar la un capăt**.

Dintre **conectorii** folosiţi pentru cablurile coaxiale pot fi menţionate *BNC* (*Bayone-Neill-Concelman*), folosit pentru reţele de calculatoare şi aplicaţii video, şi *type-F*, folosit pentru *CATV*.

2.4.2. Fibra optică - este mediul care asigură transmiterea luminii, modulată la o **anumită frecvenţă**. Comparativ cu alte medii de transmisie, fibra optică este cea mai costisitoare, dar **nu este susceptibilă la interferenţe electromagnetice şi în plus asigură rate de transfer mult mai ridicate decat celelalte categorii de medii** [40, 41]. **Sursa de lumină** pentru cablul de fibră optică este o diodă luminescentă, iar datele sunt codificate prin varierea intensităţii luminii.

Interiorul fibrei optice este format din *core* (miez) şi *clading*, două tuburi concentrice de sticlă, inseparabile, având indici de reflexie diferiţi. Propagarea semnalului se bazează pe fenomenul de reflexie totală. ***Clading*-ul,** foarte subţire, cu **diametrul de 125 microni**, este învelit în trei straturi protectoare: un strat numit *buffer* (teacă), de obicei colorat, un înveliş rezistent de protecţie fabricat din *kevlar* şi o jachetă

---

tehnologii digitale moderne de transmisie de date.

[12] Există mai multe tipuri de cabluri coaxiale, utilizate în diferitele domenii menţionate anterior. De exemplu, pentru *Ethernet 10Base2*, folosim un cablu coaxial numit *RG-58*, având impedanţa de *50 Ohm*, lungimea maximă fiind de 185 de metri, iar viteza maximă de transmisie este de *10 Mbps*. Cablurile coaxile *RG-59* sunt folosite în transmisiile *TV*, cu singură menţiune că impedanţa acestora este de *75 Ohm*.

exterioară din *PVC* (înveliş). Aceste trei învelişuri au rol de protecţie pentru partea din sticlă care este foarte fragilă.

În funcţie de modul de transmisie şi de dimensiunea *core*-ului, fibrele optice se împart în două categorii: *single mode* (Fig. 2.8) şi *multimode.*

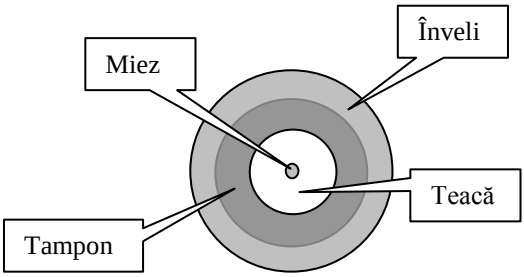


Fig. 2.8. Structura fibrei optice

Fibra optică *single-mode* (monomodallă) are o dimensiune a *core*-ului de **10 microni (mai nou - între 5 şi 8 microni),** acesta acţionând ca un ghidaj pentru raza luminoasă a semnalului care se transmite astfel aproape **fără reflexie**. Fibra optică *single-mode* permite distanţe mai mari de transmisie decât cea *multi-mode*, însă este mult mai scumpă şi impune precauţii speciale.

Echipamentele terminale folosesc lasere pentru a emite semnal luminos cu **lungimi de unda de 1310 sau 1550 nanometri.** Deoarece laserul emite o undă luminoasă foarte puternică şi focalizată, aceste echipamente pot produce **leziuni grave ochiului**. De asemenea, echipamentele de *single-mode* sunt mai scumpe decât cele de *multi-mode*.

Fibra *multi-mode* are dimensiunea ***core*-ului de 50 sau 62,5 microni,** acest lucru permiţând transmiterea semnalului prin reflexie în pereţii *core*-ului. Acest tip de fibră permite **distanţe mai mici decât cea *single-mode*** (deoarece lumina are un drum mai lung de parcurs), însă este mai ieftină şi mai uşor de folosit (mai uşor de terminat cu conectori şi de sudat). De asemenea, echipamentele care emit semnal pe fibra

optică *multi-mode* sunt mai ieftine, deoarece folosesc *LED*-uri (*light emiting diode*), folosind **lungimi de undă de 850 sau 1300 nanometri**. Aceste echipamente cu *LED*-uri nu sunt periculoase omului.

În cazul legăturilor de fibră optică avem de-a face cu legături punct la punct, unde transmisia este *full-duplex* şi nu există posibilitatea apariţiei coliziunilor, **limitarea distanţei maxime** la care se poate întinde un segment de fibră optică este dată **numai de puterea de emitere a dispozitivelor terminale,** putând ajunge în cazul transmisiei *single-mode* şi la **120 de Km** pentru *FastEthernet* şi mai mult pentru alte tehnologii.

2.4.3. Comparaţie între fibrele optice şi firul de cupru. La începuturile apariţiei fibrei optice au existat păreri conform cărora în „câţiva" ani, firele de cupru vor fi înlocuite cu fibră optică în totalitate. Acest lucru este greşit. Printre **avantajele pe care le prezintă firele de cupru** menţionăm: preţul scăzut, uşurinţa în instalare, nu necesită atenţie sporită în utilizare. Aceste avantaje fac firele de cupru mediul ideal pentru cablări în reţele mici şi mijlocii în interiorul clădirilor, unde nu se justifică fibra optică.

Dintre **dezavantajele majore are firelor de cupru** menţionăm: sunt susceptibile la interferenţe electrice şi pot fi folosite pe distanţe relativ mici - oricum mult, mult mai mici decât echivalentul lor în fibră optică.

Fibra are multe **avantaje**. În primul rând, lărgimea de bandă pe care o suportă este mai mare decât a cuprului. Un singur cablu de fibră optică *multi-mode* poate purta acum **aproape 5 milioane de convorbiri telefonice simultane**. Fibra are **avantajul** că nu este afectată de şocurile electrice, de interferenţa câmpului electromagnetic sau de căderile de tensiune. De asemenea, nu este afectată de substanţele chimice corozive din aer, fiind ideală pentru mediile aspre din fabrici.

Companiile de telefoane preferă fibra şi din alt motiv: este subţire şi foarte uşoară. Canalele cu cabluri sunt în general pline până la refuz, iar prin înlocuirea cuprului cu fibră se golesc canalele, iar cuprul are o valoare foarte bună pe piaţă. În plus, **900 de cabluri torsadate de 1 km lungime cântăresc 7250 kg.** Un cablu ce conţine **24 fibre şi are aceeaşi capacitate cântăreşte doar 60 kg,** acest lucru reducând drastic necesitatea unor echipamente mecanice scumpe care trebuie întreţinute.

În fine, fibrele nu pierd lumina şi sunt foarte dificil de interceptat. Acest lucru le oferă **o excelentă securitate**. Pe de altă parte, fibra este o tehnologie nefamiliară şi necesită o pregătire pe care mulţi ingineri nu o au. Terminarea fibrei (adică ataşarea conectorilor) este un procedeu care necesită multă pregătire şi experienţă.

Fibra optică nu poate fi folosită ca un cablu *UTP* - să fie îndoită prea tare, să fie călcată, să fie strânsă după piciorul mesei, etc. Deoarece transmisia optică este prin natura ei unidirecţională, comunicaţiile bidirecţionale necesită fie două fibre, fie două benzi de frecvenţă diferite pe aceeaşi fibră. Nu în ultimul rând, interfeţele pentru fibră costă mult mai mult decât interfeţele electrice.

2.4.4. Sistemele fără fir - ca cel de transmisie **radio terestră,** au apărut ca un prim nivel de *broadcasting* de sunet şi ca un substitut al telefonului fix. Mai târziu, **lansarea sateliţilor** de comunicaţie a făcut posibilă eliminarea necesităţii unei linii de vizibilitate directă între receptori şi sursa serviciilor, pentru unde radio spaţiale.

Sistemul de telefonie mobilă a satisfăcut nevoia de comunicare permanentă a utilizatorilor în mişcare, iar reţelele locale fără fir au apărut la fel de natural, pentru a conecta utilizatorii în reţele de date, fără a pune la punct o infrastructură complexă şi costisitoare de cablu de cupru sau fibră [42, 43].

Undele electromagnetice generate de electronii în mişcare sunt

caracterizate prin frecvenţă şi lungime de undă. Proprietăţile fizice ale undelor electromagnetice influenţează decisiv transmisia de date fără fir. Astfel, **un *Hertz* poate codifica unul sau mai mulţi biţi**.

În general, **comunicaţiile folosesc benzi de frecvenţă înguste**, pentru o alocare a puterii cât mai concentrată şi o recepţie mai bună.

Din spectrul electromagnetic (vezi Fig. 2.9), **undele radio, microundele, undele infraroşii şi lumina vizibilă sunt baza comunicaţiilor** din zilele de astăzi, datorită lungimilor de undă destul de mari pentru a face faţă absorbţiei în atmosferă. Odată cu creşterea frecvenţei undelor, în domeniul ultravioletelor, al **razelor X şi gamma**, **acestea sunt absorbite uşor în aer şi sunt periculoase pentru om**.

În domeniul undelor radio şi microundelor (benzile cele mai folosite pentru comunicaţii fără fir) sunt: *VLF* (*Very Low Frequency*), *LF* (*Low Frequency*), *MF* (*Medium Frequency*), *HF* (*High Frequency*), *VHF* (*Very High Frequency*), *SHF* (*Super High Frequency*), *EHF* (*Extra High Frequency*) - care formează o bază de transmisie de date dependentă de proprietăţile lor fizice:

a) Undele de **frecvenţă joasă şi medie** (*VLF*, *LF*, *MF*) (**3 KHz - 3 MHz**): sunt numite şi **unde terestre**, pentru că se propagă la suprafaţa Pământului, ghidate de ionosferă (vezi Fig. 2.10) în lungul curburii Pământului; depăşeşte obstacolele şi se propagă uşor prin clădiri datorită lungimilor de undă mari, care fac ca difracţia să fie maximă; sunt folosite în **comunicarea la distanţe mari** şi pentru transmisiile **radio** de lungime de **undă mare şi medie**;

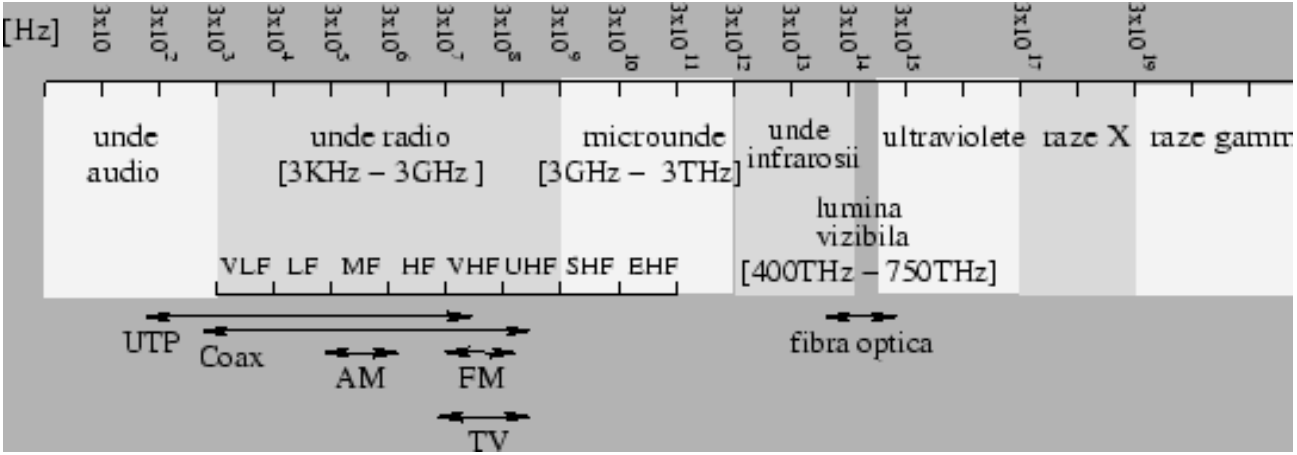


Fig. 2.9. Spectrul electromagnetic [11]

b) undele de **frecvenţă înaltă** (*HF*) (**3-30 MHz**): sunt numite şi **unde celeste**, pentru ca nu tind să se propage la suprafaţa Pământului, dar nici nu se disipă în spaţiu; propagarea undelor se face prin reflexie repetată de straturile înalte ale ionosferei, astfel că undele pot parcurge distanţe mari şi pot fi recepţionate în afara liniei directe de vizibilitate; sunt folosite în **comunicarea la distanţe medii** (**între 500 şi câteva mii de kilometri**) şi pentru transmisiile **radio** de lungime de **undă scurtă**;

c) Undele de **frecvenţă foarte înaltă** (*VHF*) (**30-300 MHz**): sunt numite **unde spaţiale** pentru ca nu urmează linia Pământului şi nu se reflectă de ionosferă, deci pot fi recepţionate **numai în linia de vizibilitate directă;** sunt folosite pentru **comunicaţii de rază mijlocie** (**70-100 Km la sol, câteva sute de Km în aer**), comunicaţii **mobile** şi transmisii de **sunet**;

d) Undele de **frecvenţă ultra înaltă** (*UHF*) (**300-3000 MHz**): sunt **unde spaţiale** (**numite microunde sau unde centimetrice**) intens folosite în sistemele de **comunicaţie actuală pentru transmisii de rază mică, transmisii TV şi legături punct la punct**;

e) Undele de **frecvenţă super înaltă** (*SHF*) (**3-30 GHz**): sunt **unde spaţiale** folosite în **comunicaţiile pe bază de sateliţi, în sistemele radar şi pentru legături punct la punct.**

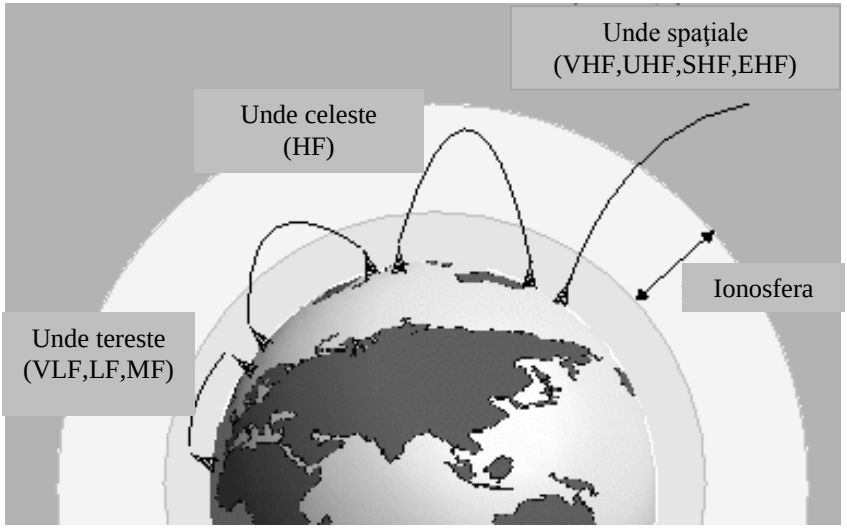


Fig. 2.10. Ionosfera [11]

Pentru microundele de frecvenţă mare, absorbţia în aer este destul de puternică pentru a face undele cu **frecvenţa mai mare de 400GHz inutilizabile**.

Pentru toate undele din clasa microundelor**, absorbţia în aer este un factor important, fapt care limitează raza de transmisie.**

Verifică-ţi cunoştinţele:

1) Factorii care influenţează asupra alegerea mediilor de transmisie.
2) Categoriile mediilor de transmisie şi limitările pe care acestea le impun unor tehnologii.
3) Categoriile cablurilor de cupru şi caracteristicele lor.
4) Categoriile fibrelor optice şi proprietăţile lor.
5) Comparaţi avantajele firelor de cupru şi fibrelor optice.
6) Prin ce se caracterizează undele electromagnetice?
7) Care factorii limitează raza de transmisie a microundelor?

Întrebările pentru autoevaluare:

1. Enumeraţi funcţiile definite în cadrul nivelului Fizic
2. Definiţi noţiunea de semnal şi zgomot în sistemele de comunicaţie
3. Tehnica de transmitere a semnalului
4. Caracterizaţi mediu de transmisie prin fire de cupru
5. Caracterizaţi mediu de transmisie prin fibra optică
6. Caracterizaţi mediu de transmisie prin sistemele fără fir
7. Descrieţi avantajele şi dezavantajele fiecărui mediu de transmisie

Capitolul 3. Protocoale şi tehnici de acces la nivelul Legătură de date. Sisteme de telefonie mobilă

3.1. Protocoalele de acces la mediu de transmisie
3.2. Tehnici de acces la mediul de transfer în reţele locale (*LAN*)
  3.2.1. Scheme de adresare folosite în telecomunicaţii
  3.2.2. Structura generică a unui cadru de nivel 2
  3.2.3. Protocoale de comunicaţie la nivelul Legătură de date
3.3. Tehnici de acces pentru reţele largi (*WAN*)
  3.3.1. Comutaţie de circuite (*Circuit-switched*)
  3.3.2. Comutaţie de pachete (*Packet-switched*)
  3.3.3. *Cell-switched. ATM* (*Asyncronous Transfer Mode*)
  3.3.4. *Dedicated digital*. Multiplexare în telefonie
    a) Fluxuri *E1* şi *T1*
    b) *xDSL* (*Digital Subsriber Line*)
    c) *PPP* (*Point-to-Point Protocol*)
    d) *SDH* **(***Synchronous Digital Hierarchy***)** şi *SONET* (*Synchronous Optical Network*)
  3.3.5 *Analog services*
3.4. Sisteme de telefonie mobilă
  3.4.1. *AMPS* (*Advanced Mobile Phone System*)
  3.4.2. *D-AMPS* (*Digital Advanced Mobile Phone System*)
  3.4.3. *GSM* (*Global System for Mobile Communications*)
  3.4.4. *CDMA* (*Code Division Multiple Access*)
  3.4.5. *EDGE* (*Enhanced Data Rates for GSM Evolution*)
  3.4.6. *3G* (*Third Generation*)
  3.4.7. *4G* (*Fourth generation*)
  3.4.8. *5G* (*5th generation mobile networks or 5th generation wireless systems*)
3.5. *Bluetooth*
3.6. *Frame-Relay*
3.7. *GPS* (*Global Positioning System*)

Nivelul Legătură de date se ocupă cu adresarea fizica, cu topologia reţelei, accesul la reţea şi controlul fluxului fizic (*flow control*). Acest nivel transferă unităţi adresabile de informaţie, cadre (*frames*), face verificarea erorilor (CRC - *Cyclic Redundancy Check*) şi retransmite cadrele recepţionate incorect.

Nivelul Legătură de date furnizează un transport sigur, fiabil, al datelor de-a lungul unei legături fizice, realizând [44]:

- Controlul **erorilor** de comunicaţie;
- Controlul **fluxului de date**;
- Controlul **legăturii**;
- **Sincronizarea** la nivel de cadru.

Datele sunt **împărţite în cadre formatate şi sincronizate** pentru transmitere prin nivelul fizic. Se asigură astfel **un canal virtual**, fără erori, pentru nivelul superior, cel de reţea. **Unitatea de date: cadrul**.

3.1. Protocoalele de acces la mediu de transmisie

Protocolul de acces reprezintă metoda pe care fiecare staţie de lucru o utilizează pentru a obţine **accesul în cadrul transmisiei datelor**.

In cadrul unui *LAN* dat, **toate staţiile trebuie să folosească acelaşi protocol de acces**.

Cele mai uzuale protocoale de acces la mediu sunt [45]:

a) acces multiplu cu ascultarea mediului ***CSMA/CA*** (*Carrier Sense Multiple Access with Collision Avoidance*);

b) ascultarea mediului şi detectarea coliziunilor ***CSMA/CD*** (*Carrier Sense Multiple Access with Collision Detection*);

c) inel cu **jeton** (*Token Pasing*).

Analizăm funcţionarea acestor protocoale [46]:

a) *CSMA/CA* (vezi Fig. 3.1) - este un protocol de acces la mediu care **ascultă mediul pentru a evita coliziunile**. **Distribuirea informaţiilor** de rezervare a mediului se face prin interschimbarea de

către staţiile care vor să converseze unele **cadre de tip *RTS*** (*Request to Send*) **şi *CTS*** (*Clear to Send*). Aceste două tipuri de cadre conţin un **câmp de durată**, care specifică perioadele de timp pentru transmisia datelor, cadrului de confirmare pozitivă ***ACK*** (*Acknowledge*) de la terminarea conversaţiei, şi a tuturor intervalelor de timp dintre cadrele trimise [47].

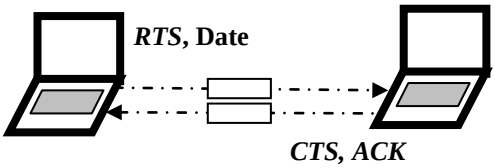


Fig. 3.1. *CSMA/CA* distribuit

Toate staţiile care se află în reţea, în raza de transmisie a staţiei emiţătoare, care trimite cadrul *RTS,* sau a staţiei receptoare, care trimite cadrul *CTS,* **vor afla despre rezervarea mediului**. Dacă mai mult de o staţie transmite în acelaşi timp pe canal, se produc coliziuni.

b) Tehnica „ascultă, transmite şi ascultă transmisia" **(*CSMA/CD*)** îmbunătăţeşte pe cea precedentă prin **detectarea fermă a coliziunilor** - îşi reglează transmisia de date odată ce coliziunile s-au produs. Sursa, după ce transmite pachetul, **aşteaptă un interval foarte scurt de timp (μs)**, după care îşi **ascultă propria transmisie**. Dacă detectează o diferenţă între informaţia transmisă şi cea recepţionată, transmite un mesaj ca toate sursele implicate să fie informate. Staţiile îşi **replanifică transmisia** folosind un algoritm special de reluare. Aceste coliziuni **nu produc aspecte negative** în reţea (pierderi de date), ci fac parte din **logica de partajare a canalului.** Majoritatea coliziunilor într-o reţea *Ethernet* neaglomerată se rezolvă în μs (sau 10^{-6} secunde) [48].

Dacă apar mai multe coliziuni pentru acelaşi *frame*, intervalul de μs de aşteptare creşte. **După 16 coliziuni consecutive pentru acelaşi pachet**, acesta nu se va mai transmite, deoarece se consideră că există o supraîncărcare a reţelei pe un interval mare de timp sau reţeaua este întreruptă.

c) Inel cu jeton (*Token Passing*) - metoda dreptului de control circulant, constă în utilizarea unei **combinaţii specifice de biţi** (jeton de control sau *token*) identificată de reţea, care în absenţa traficului circulă continuu pe inel. Posesorul jetonului are acces la inel. Dacă nu are nimic de transmis, trimite mai departe jetonul; în caz contrar, **jetonului i se anexează mesajul**.

Înlăturarea mesajului de pe inel se realizează de nodul sursă (cel care a transmis mesajul), care în acest moment verifică identitatea datelor trimise cu cele sosite. Poate fi realizat cu un singur mesaj sau cu un şir întreg de mesaje.

O **situaţie particulară** apare la pornirea sistemului, sau la alterarea jetonului, cînd fiecare nod va asculta inelul pe o durată proprie, după care prima dintre ele va genera un jeton.

Verifică-ţi cunoştinţele:

1) Cu ce se ocupă nivelul Legătură de date?
2) Precizaţi destinaţia protocoalelor de acces la mediu.
3) Enumeraţi şi caracterizaţi cele mai uzuale protocoale de acces la mediu.

3.2. Tehnici de acces la mediul de transfer in retele locale (*LAN*)

Adresele folosite de nivelul Legătură de date se numesc **adrese *MAC* sau adrese fizice**. Acestea au **48 de biţi** exprimaţi în **12 cifre hexazecimale: FF.FF.FF.FF.FF.FF**. Termenul de adresă fizică este folosit pentru a distinge adresa fizică şi adresa logică [49].

Adresa fizică este atribuită în procesul de fabricaţie unui

dispozitiv de reţea, iar cea **logică** este atribuită de administrator şi poate fi schimbată cu uşurinţă. **Adresele fizice** sunt stocate în memoria *ROM*, şi sunt încărcate în *RAM* în momentul iniţializării plăcii de reţea. Din această cauză adresele fizice mai sunt numite şi *burned-in addresses* (*BIAs*) [50]. Instituţia ce administrează adresele fizice este *IEEE* (*Institute of Electrical and Electronic Engineers*). Problema este că *IEEE* nu poate monitoriza direct atribuirea fiecărei adrese fizice, astfel încât **transferă această responsabilitate producătorilor** [51].

Din cei **şase octeţi** ce compun adresa fizică, **primii trei** vor fi folosiţi pentru **identificarea fabricantului**, acest câmp fiind denumit *Organizational Unique Identifier* (*OUI*). Prin urmare, ***IEEE* distribuie producătorilor fâşii din spaţiul de adrese**, urmând ca aceştia la rândul lor să atribuie fiecărui dispozitiv de reţea nou creat una sau mai multe adrese fizice.

3.2.1. Scheme de adresare folosite în telecomunicaţii.

Tehnicile de acces la mediul de transfer în reţele locale se definesc prin **scheme de adresare** folosite în telecomunicaţii şi **protocoale de comunicaţie** *IEEE 802.3, IEEE 802.5, IEEE 802.11*.

Există două scheme de adresare folosite în telecomunicaţii: (a) **adresarea plată** şi cea (b) **ierarhică**.

a) În cazul **distribuţiei plate** spaţiul de adrese este ocupat treptat şi complet, adică, dacă am atribuit adresa „**n**”, următoarea adresă pe care trebuie să o atribuim va fi neapărat „**n+1**”.

Avantajul acestui tip de adresare constă în **folosirea eficientă a spaţiului** de adrese. **Dezavantajul** constă în **imposibilitatea implementării unor algoritmi eficienţi de căutare**, deoarece spaţiul adreselor va reprezenta o mulţime neordonată. **Adresarea plată este mai puţin întâlnită**. Exemplul unei distribuţii plate poate fi

numerotarea bancnotelor sau a biletelor de transport.

b) **Adresarea ierarhică** se utilizează de la **codul de bare de pe produse până la numerele de telefon**. Acesta presupune înglobarea în adresă a unei **informaţii suplimentare** ce va permite identificarea mai întâi a **grupului de adrese** căreia îi aparţine adresa destinaţie şi abia apoi, în interiorul acestui grup, **identificarea adresei destinaţie**. De exemplu, un număr de telefon din Moldova, va conţine adresa ţării, adresa raionului şi abia în final adresa postului telefonic destinaţie. **Avantajul** adresării ierarhice este **posibilitatea ordonării spaţiului** de adrese, dar totuşi se pierde o parte din spaţiul de adrese.

Schema de adresare folosită la nivelul Legătură de date combină **dezavantajele ambelor scheme de adresare**: mulţimea adreselor fizice este o mulţime neordonată, care în plus nu va putea folosi integral spaţiul de adrese. Toate acestea nu afectează însă principala funcţie a adreselor fizice, şi anume asigurarea unicităţii.

3.2.2. Structura generică a unui cadru de nivel 2. O parte importantă în cadrul comunicaţiei în reţea o constituie procesul de **încapsulare** ce are loc la nivelul Legătură de date (vezi Fig. 3.2).

Acesta este **prima formă de organizare a şirurilor de biţi transmise la nivelul Fizic**: şirurile de biţi primiţi/transmişi de la/către nivelul Fizic sunt **organizaţi în cadre (*frame*)**. Aceasta înseamnă că nivelul Legătură de date îşi adaugă ***header*-ul** (antetul) şi ***trailer*-ul** propriu pachetelor primite de la nivelul reţea, creând astfel cadrele ce sunt plasate sub forma de şiruri de biţi nivelului Fizic. De asemenea, biţii primiţi sunt reasamblaţi în cadre; *header*-ul şi *trailer*-ul cadrelor primite sunt îndepărtate şi interpretate, iar pachetele obţinute sunt trimise nivelului reţea [52, 53]. Împărţirea în cadre permite obţinerea de **informaţii care nu se puteau transmite prin şiruri de biţi şi anume**: calculatoare care comunică între ele; când începe şi când se termină comunicarea între anumite calculatoare individuale; ţinerea

evidenţei erorilor care au apărut în comunicare; al cui este rândul să transmită în cadrul comunicaţiei, etc.

Transpunerea în cadre este **ultima fază a încapsulării**, înainte ca informaţia să fie codificată în biţi şi transmisă prin mediul de comunicare.

Fiecare standard are propria lui structură a cadrului, adică pentru diferite tipuri de reţele (*Ethernet*, *Token Ring* etc.) vom avea formate

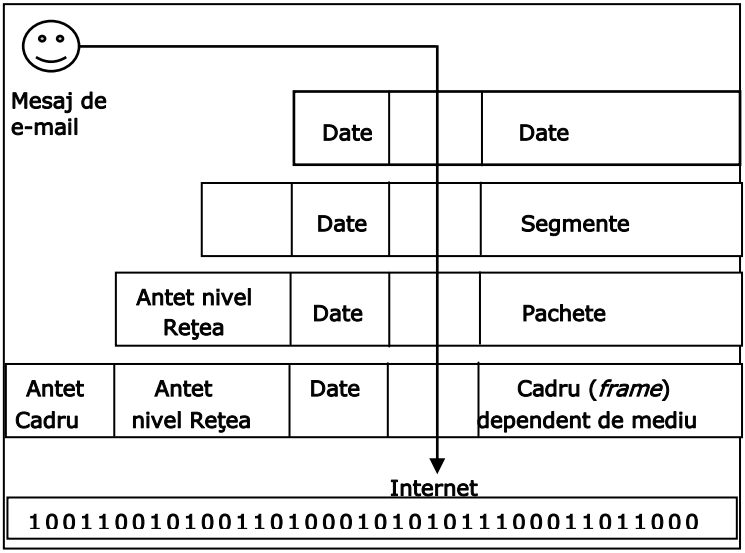


Fig. 3.2. Structura procesului de incapsulare

de cadre diferite. Mai jos se reprezintă **structura generică a unui cadru de nivel 2** (vezi Fig. 3.3**)**, care conţine anumite câmpuri compuse din unul sau mai mulţi octeţi:

Început de cadru	**Adresa**	**Lung/Tip**	**Date**	***FCS***

Fig. 3.3. Structura generică a unui cadru de nivel 2

- Primul câmp (**Început de cadru**) anunţă începerea unui cadru,

conţinând o secvenţă de semnalizare specifică fiecărei tehnologii în parte.

- Adresarea (**Adresa**) este esenţială pentru a şti cui se adresează acel cadru şi de la cine provine. Fiecare protocol are propriul lui tip de adresare; de exemplu, în cazul *Ethernet*-ului adresarea se realizează prin intermediul adreselor *MAC*.

În cadrul *header*-ului sunt înscrise, pe lângă alte date, adresele *MAC* - sursă şi destinaţie, fără de care reţeaua nu ar putea funcţiona, pentru că staţiile sau dispozitivele de interconectare nu vor putea identifica la primirea unui cadru, dacă le este destinat lor sau nu.

Subnivelul *MAC* conţine **protocoalele (*FDDI, FR, HDLC,* etc.) care determină într-o reţea locală care staţie are dreptul să transmită la un moment dat,** adăugând verificarea erorilor. Aceste protocoale organizează comunicarea şi gestionează modul şi momentul în care fiecare staţie are acces la mediul de transmisie.

- Anumite tipuri de cadre (***Lung/Tip***) conţin lungimea cadrului, iar altele conţin un câmp special, numit „*Protocol field*" sau „*Type Field*" prin intermediul căruia se **specifică protocolul de nivel superior căruia i se adresează**. O staţie primeşte un cadru, decapsulează datele şi trimite nivelului superior informaţia din acel cadru. Dacă pe nivelul superior se află un singur protocol, cum ar fi *IP*-ul, atunci e simplu, dacă utilizăm două protocoale, cum ar fi *IP* şi *IPX*, atunci intervine „*Protocol field*", care specifică protocolul de nivel 3 cui i se adresează datele.

- În timpul unei transmisii datele sunt susceptibile de eroare, unde în structura unui cadru de nivel 2 este prevăzut un **câmp *FCS* (***Frame Check Sequence*) care determină dacă a apărut o eroare: în momentul în care staţia sursă încapsulează cadrul, calculează *FCS*[13] şi ataşează rezultatul la cadru. Staţia care recepţionează cadrul, calculează şi ea la rândul ei acest *FCS* şi îl compară cu cel citit de la

[13] În funcţie de tehnologia folosită există mai multe variante de a calcula *FCS*

sfârşitul cadrului primit. Dacă nu coincid, înseamnă că a avut loc o eroare pe parcursul transmisiei.

Controlul erorilor, se realizează în două moduri:

1. *FEC* (*Forward Error Correction*) - foloseşte biţii de control pentru detectarea şi corectarea erorilor;

2. *ARQ* (*Automatic Retransmition Query*) – este utilizat numai pentru detectare, nu şi pentru corectarea erorilor, ca mijloc de alertare a sursei, că informaţia nu a fost recepţionată corect.

3.2.3. Protocoale de comunicaţie la nivelul Legaturii de date. Reţelele locale lucrează într-unul din două moduri fundamentale: sesizarea **coliziunilor** sau trecerea mesajului ***token***. Referind la aceasta, există două **categorii de acces la mediul de transmisie**:

***Ethernet*:** reţea cu sesizarea coliziunilor, *IEEE 802.3*. **Categoria de acces la mediul de transmisie:** nedeterminist - utilizează o abordare de tipul: primul venit, primul servit. **Protocol:** CSMA/CD (*Carrier Sense Multiple Access with Collision Detection).*	***TokenRing*:** reţea cu trecerea mesajului *token, IEEE 802.5*. **Categoria de acces la mediul de transmisie:** determinist - fiecare staţie ştie exact când va transmite: fiecare staţie are dreptul să transmită pe rând prin plasarea unui jeton (*token*). **Protocol:** CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance).*

a) **Utilizarea protocolului *Ethernet*** [14]- este cea mai răspândită tehnologie de LAN, având numeroase **avantaje** cum ar fi: uşurinţă de instalare şi întreţinere, capacitatea de a introduce noi tehnologii (de la *10 Mbps la 10 Gbps*), fiabilitatea şi costul relativ scăzut de instalare şi *upgrade*. *Ethernet*-ul nu este de fapt o tehnologie, ci este mai mult o familie de tehnologii care include: *LegacyEthernet, FastEthernet, GigabitEthernet*.

[14] Ideea originală de la care a plecat această tehnologie a apărut în anii 1970, când la o universitate din Hawaii se punea problema accesului mai multor utilizatori la o reţea fără ca semnalele lor să se amestece.

Ethernet-ul este situat pe **două niveluri ale stivei *OSI*** şi anume: **partea de jos a nivelului Legătură de date (subnivelul *MAC*) şi nivelul Fizic.**

Primul sistem se numea *Alohanet* şi a devenit mai târziu baza unei **metode de acces la mediu numită *CSMA/CD***, metodă folosită de tehnologia *Ethernet*. [15] Pe o reţea *Ethernet*, datele sunt trimise în toate direcţiile, cu rata de transfer de ***10 Mbps***. Pachetele de date (*frame*) sunt primite de toate calculatoarele, dar numai cele cărora le sunt adresate (corespunzător adresei de destinaţie a pachetului) răspund cu o confirmare.

Cele mai multe din **problemele de transmisie** ale unei reţele *Ethernet* se datorează **cablurilor defectoase sau funcţionării eronate a plăcilor adaptoare la reţea**.

IEEE 802.3 - este un standard *LAN*, similar cu *Ethernet*. Prima sa ediţie apare în 1985. Diferenţele între cele două standarde *Ethernet* apar în zona arhitecturii de reţea şi a formatului pachetului de date (*frame*). Arhitectura de reţea *IEEE 802.3* face distincţie între nivelurile *MAC* şi *LLC*; adevăratul protocol *Ethernet* pune toate aceste niveluri împreună în nivelul Legăturii de date. De asemenea, *Ethernet* defineşte o configuraţie *ECTP* (*Ethernet Configuration Test Protocol*) care lipseşte din standardul 802.3. Diferenţele importante între cele două protocoale constă **între tipul şi lungimea câmpurilor,** care

[15] Primul standard *Ethernet* a fost publicat în 1980 de un consorţiu format din firmele *DEC*, *Intel* şi *Xerox*, consorţiu numit *DIX*. *Ethernet*-ul funcţiona atunci pe un suport de cablu coaxial gros, numit *thicknet*, şi atingea viteze de până la ***10Mbps***. În 1985, *IEEE* (*Institute of Electrical and Electronics Engineers*) au publicat o serie de standarde pentru *LAN*, serie care începea cu *802.x*. Standardul pentru *Ethernet* este *802.3* şi a adus ceva modificări faţă de standardul iniţial propus de *DIX*, însă modificările sunt atât de mici, încât în linii mari cele două standarde sunt aproape identice.

constituie pachetul de date (*frame*). Aceste diferenţe pot conduce la incompatibilitatea celor două protocoale.

b) Descrierea pachetului original de date *Ethernet* (vezi Fig. 3.4) [53]**:**

- Preambul – este folosit pentru **sincronizare şi încadrare**, are lungimea de 8 octeţi şi conţine întotdeauna tiparul de biţi 10101010, în primii 7 octeţi, cu 10101011 în ultimul octet (al 8-ulea).

Preambul	**Adresa destinaţie**	**Adresa sursă**	**Lung/Tip**	**Date**	***FCS***
8 oct.	**6 oct.**	**6 oct.**	**2 oct.**	**46-1500 oct.**	**4 oct.**

Fig. 3.4. Structura cadrului *Ethernet*

- **Adresa de destinaţie.** Ocupă **6 octeţi** şi conţine **adresa staţiei** de lucru ce va primi acest pachet de date. **Primul bit** (cel mai din stânga) al primului octet are o semnificaţie specială, anume, dacă este **egal cu 0**, adresa de destinaţie este o **adresă fizică unică** în universul *Ethernet*. Ca rezultat al unei scheme de denumire administrate de corporaţia *Xerox*, primii trei octeţi sunt o adresă de grup asignată de *Xerox*, iar ultimii trei sunt asignaţi local. Dacă bitul cel mai din stânga **este 1**, el reprezintă un **pachet de date de transmis**. Ca urmare, restul adresei de destinaţie se poate referi la un grup de staţii de lucru înrudite logic sau la toate staţiile de lucru din reţea (toate 1-uri).
- **Adresa sursă.** Prezenţa adresei sursă în cadru se explică prin faptul că orice comunicaţie este bidirecţională, în sensul că orice cadru transmis are de obicei ca urmare emiterea unui cadru de răspuns. Acest câmp ocupă **6 octeţi** şi identifică **staţia de lucru** emiţătoare a pachetului de date. Cel mai din stânga **bit** al primului octet este **întotdeauna 0**.
- **Lungime/Tip.** Câmpul Lungime/Tip poate fi interpretat în două feluri: dacă valoarea acestuia este **mai mică de 1536 (0…600 în hexazecimal: $16^0*0+16^1*0+16^2*6=1536$)** - atunci el reprezintă

lungimea. Dacă este **mai mare de 1536** - el reprezintă **protocolul de nivel superior** folosit. Conţine **2 octeţi de date** ce identifică **tipul protocolului de nivel superior** care a emis (sau vrea să recepţioneze) acest pachet de date. Câmpul Tip este asignat de *Xerox* şi nu este interpretat de *Ethernet*. El face posibil ca protocoalele multiple de nivel înalt (denumite niveluri client – *Client Layers*) să împartă reţeaua fără a intra unul în mesajele celuilalt.

- **Porţiunea de date.** Ea reprezintă **mesajul de date** pe care se intenţionează ca pachetul să le transporte la destinaţie. Câmpul de date trebuie să fie **mai mare de 46 de octeţi**. Dacă datele sunt de lungime mai mică, atunci i se adaugă o „umplutură" numită *padding* pentru a ajunge la dimensiunea de 46 octeţi. Acest câmp nu are voie să depăşească valoarea de *MTU - Maximum Transmission Unit* - care pentru *Ethernet* este 1500 octeţi.

- **Câmpul de control *FCS*** este adăugat în cadru pentru a determina dacă nu cumva a avut loc o eroare în cadrul transmisiei.

Putem menţiona că un pachet întreg de date *Ethernet* are între 64 şi 1518 octeţi (suma tuturor octeţi fără Preambul) şi că dimensiunea minimă a unui mesaj de date este de 46 octeţi.

c) **Tratarea coliziunilor în protocolul *Ethernet*. Domeniul de coliziune** este acea zonă dintr-o reţea care va fi afectată de apariţia unei coliziuni în interiorul ei[16]. Reţelele ***Ethernet*** sunt de tip *share-media*, deci orice cadru transmis de către o staţie va fi recepţionat de către toate celelalte staţii din reţeaua locală. Toate calculatoarele, la recepţionarea unui cadru valid, vor verifica dacă adresa *MAC* înscrisă în cadrul câmpului destinaţie din *header*-ul cadrului primit este identică cu adresa *MAC* proprie. Dacă nu se stabileşte că cele două adrese sunt identice,

[16] Dispozitivele din categoria *hub*-urilor şi repetoarelor propagă coliziunea. Reţeaua locală poate fi împărţită în domenii de coliziune separate prin intermediul unor dispozitive din categoria *bridge*-urilor şi *switch*-urilor.

cadrul este ignorat şi nu va fi transmis către nivelul reţea.

d) Utilizarea pachetului de date *IEEE 802.3 Ethernet* [54]:

- Preambul. Câmpul ocupă 7 octeţi de date de sincronizare. Fiecare octet are acelaşi tipar de biţi 10101010.

- *SFD* (*Start Frame Delimiter*) **- delimitator de început de pachet.** *SFD* constă dintr-un singur octet care are tiparul de biţi 10101011 (Câmpurile Preambul şi *SFD* ale pachetului *IEEE 802.3* se potrivesc cu câmpul Preambul al celui *Ethernet*).

- Adresa de destinaţie. Câmpul conţine 2 sau 6 octeţi în funcţie de tipul de reţea *IEEE 802.3* şi indică **staţia de lucru** căreia îi este destinat pachetul. Ca urmare, toate adresele dintr-o reţea trebuie să fie adrese de 2 sau 6 octeţi.

Cel mai răspândit tip de protocol *IEEE 802.3*, numit *10BASE5*, specifică adrese de 6 octeţi. Primul bit al adresei de destinaţie este bitul individual/de grup (**I/G**). Acesta are **valoarea 0**, dacă adresa se referă la o **singură staţie de lucru**, sau **valoarea 1**, dacă reprezintă **un grup de staţii de lucru** (un mesaj de transmis).

- Adresa sursă. Este adresa de 2 sau 6 octeţi a staţiei emiţătoare.

- Lungimea. Ocupă 2 octeţi ce exprimă lungimea porţiunii de date a pachetului.

- Porţiunea de date. Câmpul variază între 0 şi 1500 octeţi de date. Dacă este mai mic de 46 octeţi, atunci câmpul următor este utilizat pentru a umple pachetul până la o dimensiune acceptabilă (minimă).

- Câmpul tampon. Conţine suficienţi octeţi de umplere pentru a asigura o anumită dimensiune minimă a pachetului de date. Dacă porţiunea de date este suficent de mare câmpul tampon nu apare în pachet (are lungimea 0).

- *CRC.* Conţine 4 octeţi ca la *Ethernet*.

În cazul ambelor protocoale (atât *Ethernet*, cât şi *IEEE 802.3 Ethernet*), dimensiunea pachetului de date fără preambul şi *SFD* este

aceeaşi: 64 – 1518 octeţi. Totuşi, la *IEEE 802.3* este permis ca aplicaţia, sau un nivel superior de protocol să trimită o zonă de date mai mică de 46 octeţi, deoarece pachetul de date este completat automat de nivelul *MAC*. La adevăratul protocol *Ethernet*, pachetele de date prea mici sunt considerate cazuri de eroare.

e) Utilizarea protocolului *IEEE 802.5* (*Token Ring*) [55]: Pentru reţeaua *IEEE 802.5 Token Ring* sunt definite trei formate de pachete de mesaje: mesaje *token*, pachete de date (*frame*) şi secvenţe de abandonare (*abort sequences*). În reţea, mesajele *token* circulă de la o staţie la alta ca printr-un inel. O staţie de lucru trimite un pachet de date **unităţii de acces multistaţie** *MSAU* (*MultiStation Access Unit*), care îndrumă pachetul spre următoarea staţie.

Fiecare placă adaptoare pentru reţea recepţionează un pachet de date de la vecin, regenerează semnalele electrice construind pachetul şi transmite rezultatul către următoarea staţie de lucru. Cu toate că diferenţele sunt invizibile, nu toate staţiile de lucru din reţea sunt egale. Una dintre staţii este desemnată ca **monitor activ**, adică îşi **asumă responsabilităţi suplimentare pentru a controla inelul.**

Monitorul activ menţine controlul temporizării în inel, emite noi mesaje *token* (dacă este cazul) ca funcţionarea să continue şi, în anumite condiţii, generează pachete de date pentru diagnoză. Monitorul activ **este ales în momentul iniţializării inelului** şi poate fi oricare dintre staţiile din reţea. Dacă acesta se defectează, există un mecanism prin care celelalte staţii (*monitoare standby*) pot decide care dintre ele va fi noul monitor activ.

f) Reţele locale fără fir *IEEE 802.11* [56]:

Reţelele locale fără fir (*WLAN*) oferă utilizatorilor aceleaşi facilităţi ca şi reţelele locale bazate pe infrastructura de cablu, dar fără limitarea impusă de fire. Standardizarea impusă reţelelor fără fir de *IEEE* şi *Wi-Fi Alliance* a permis **interoperabilitatea echipamentelor**,

ceea ce a dus în final la **scăderea costurilor şi la un proces de dezvoltare mai rapid**. În momentul de faţă viteza de transfer a datelor într-o reţea locală fără fir atinge ***54 Mbps***, iar costurile de instalare a reţelei fără fir sunt considerabil mai mici, ceea ce face ca instalarea unui *LAN* fără fir să fie o soluţie viabilă, nu numai în cazul utilizatorilor mobili, ci şi ca un substitut al *LAN*-urilor clasice.

IEEE 802.11 este o familie de protocoale care **defineşte nivelul Fizic şi subnivelul *MAC*** al nivelului Legatură de date. Standardul stabileşte ca medii de transmisie - **benzi de unde din domeniul infraroşu şi radio** (incluzând microundele). În domeniul radio sunt specificate **trei tipuri de transmisie** folosind unde radio din benzile nelicenţiate de frecvenţe *ISM* (***The industrial, scientific and medical*** *radio bands*) de ***2.4GHz*** **şi** ***5GHz***:

802.11a lucrează în banda de ***5GHz*** şi atinge viteze de ***54 Mbps****;* din cauza benzii diferite de transmisie este incompatibil cu *802.11b*, dar lucrează într-o bandă de frecvenţe mai puţin aglomerată şi oferă viteze de transmisie comparabile cu cele oferite de reţelele de cupru;

802.11b este primul standard lansat în domeniul reţelelor *LAN* fără fir, şi cea mai populară tehnologie astăzi; lucrează în banda de *2.4GHz* şi atinge viteze de ***11Mbps***; problemele de care s-a lovit acest standard au fost încărcarea benzii ***ISM*** **de** ***2.4GHz*** (în care lucrează multe alte sisteme, cum sunt *Bluetooth* şi cuptoarele cu microunde) şi viteza de transfer relativ mică, în condiţiile în care *FastEthernet* este implementat din ce în ce mai mult;

802.11g (ratificat în 2003) specifică o viteză de transfer de ***54 Mbps*** (egală cu viteza *802.11a*); fiind compatibilă cu tehnologia *802.11b* şi oferind viteze de vârf.

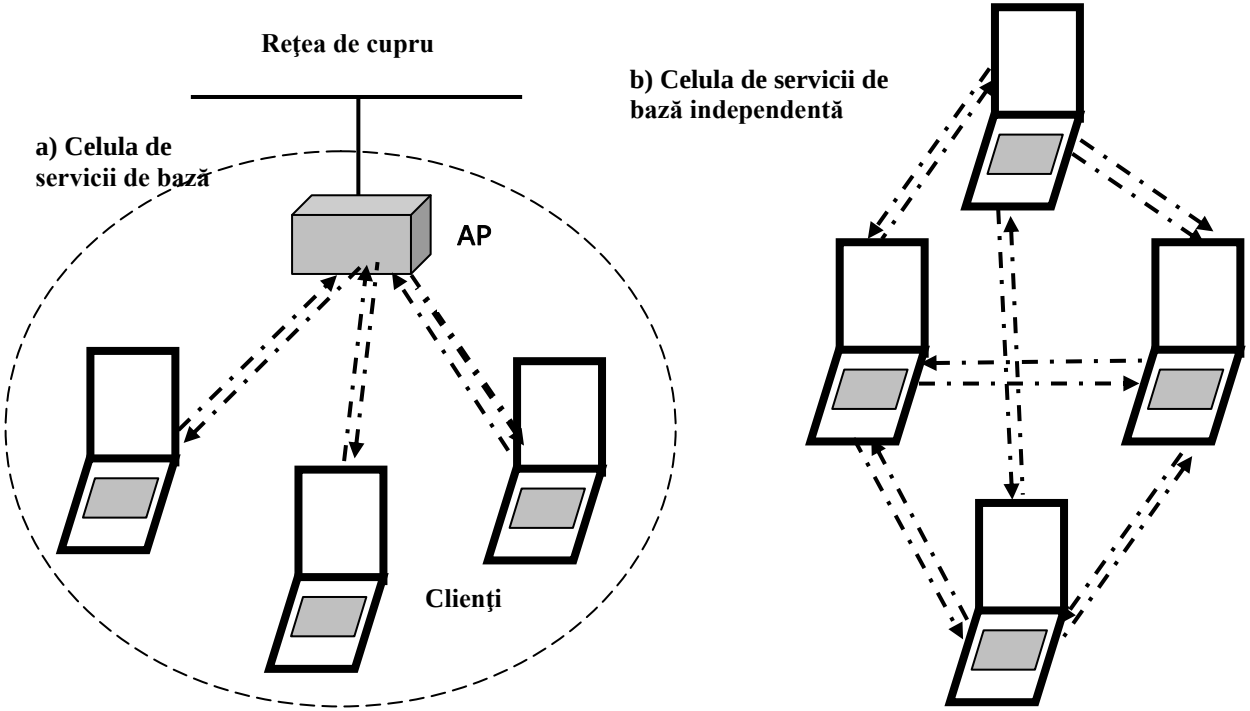


Fig. 3.5 a) *Basic Service Set* (*BSS*); b) *Independent Basic Service Set* (*IBSS*)

Echipamentele necesare implementării unei reţele fără fir *802.11* sunt: **adaptoare de reţea, care înlocuiesc plăcile de reţea tradiţionale** pentru calculatoare fixe sau mobile; un ***acces point*** (*AP*), care este punctul central al unei reţele fără fir, dar care poate funcţiona şi ca un repetor sau poate asigura conectivitatea între o reţea fără fir şi una clasică**; *bridge***-uri care conectează două reţele la distanţă.

O celulă de servicii de bază *BSS* (*Basic Service Set*) (vezi Fig. 3.5a) este formată dintr-un ***access point* (*AP*)** şi un număr de **clienţi** pentru care acesta este punctul central de conectivitate.

Conexiunea cu reţeaua de cupru este asigurată de *AP*. **Toţi clienţii comunică numai cu *AP*-ul**, şi nu între ei, ceea ce înseamnă că schimbarea poziţiei unui client în celulă trebuie să fie făcută astfel încât distanţa dintre client şi *AP* să nu depăşească o valoare maximă de acoperire a semnalului radio.

O celulă de servicii de bază independentă (numită şi celula *ad-hoc*, *IBSS* (vezi Fig. 3.5b) sau *Independent Basic Service Set*) este o topologie formată numai din staţii care comunică direct între ele. Minimul necesar pentru a pune la punct o asemenea topologie este numai de două staţii.

Cele trei tipuri de reţele fără fir specificate de *802.11* ating lăţimi de bandă între ***1 Mbps*** şi ***54 Mbps***, în funcţie de modalitatea de transmisie a datelor şi modularea lor.

Verifică-ţi cunoştinţele:

1) Cum se numesc adresele folosite de nivelul Legătură de date?
2) Cine atribuie adresa fizică şi cea logică a dispozitivelor de reţea?
3) Care este modul de definire a tehnicilor de acces la mediul de transfer în reţele locale?
4) Explicaţi diferenţa dintre distribuţia plată şi cea ierarhică.
5) Descrieţi structura procesului de incapsulare.
6) Explicaţi avantajul împărţirii în cadre.

7) Structura generică a unui cadru de nivel 2.
8) Protocoale de comunicaţie la nivelul Legaturii de date şi particularităţile lor.
9) Pe care niveluri ale stivei *OSI* este situat *Ethernet*-ul?
10) Structura cadrului *Ethernet.*
11) Structura pachetului de date *IEEE 802.3 Ethernet.*
12) Utilizarea protocolului *IEEE 802.5* (*Token Ring*).
13) Reţele locale fără fir *IEEE 802.11*.
14) Arhitectura *IEEE 802.11*.

3.3. Tehnici de acces pentru retele largi (*WAN*)

Accesul fizic la Internet poate fi prin linie de telefon comutată (*dial-up*), access prin linie închiriată, linie de telefon *ISDN (Integrated Services Digital Network)*, linie de telefon *ADSL (Asymmetric Digital Subscriber Line)*, cablu (de *TV*), radio, sistemele de telefonie mobilă *GSM (Global System for Mobile Communications)*, *UMTS* (*Universal Mobile Telecommunications System*), satelit etc.

Retelele pe arie extinsa *WAN* functionează după protocoale proprii [57]. Viteza şi disponibilitatea conexiunii Internet **împarte serviciile** pentru utilizatorii finali în două categorii: **pe linie comutată** şi **de bandă largă (de mare viteză): conexiunile pe linie comutată** (*dial-up*) necesită o linie telefonică; **conexiunile de bandă largă** pot fi *ISDN*, Radio, Cablu, *DSL*, Internet prin *Satelit* sau direct *Ethernet*. Legătura de bandă largă e mai rapidă şi este disponibilă permanent, dar în acelaşi timp este şi mai scumpă.

Legăturile între diverşii *ISP* (*Internet Service Provider*) sau între punctele de prezenţă ale unui *ISP* sunt făcute de obicei printr-o reţea care transmite un volum imens de informaţii, folosind deseori fibră optică.

Tehnologiile *WAN* cele mai importante sunt împărţite in 5 grupe: *circuit-switched, packet-switched, cell-switched, dedicated digital, analog services*.

3.3.1. Comutaţie de circuite (*circuit-switched*) - numită şi **comutare sincronă**, este o tehnologie de telecomunicaţii care asigură pentru fiecare comunicaţie un ***debit* (administrarea *flux*-urilor) constant**, prin unele **canale temporale prin care informaţia circulă periodic**. Spre deosebire de tehnica de comutaţie de pachete (*packet switching*), în reţelele cu comutaţie de circuite **traseul de conexiune este fix** pe durata comunicaţiei şi este alocat **exclusiv pentru o comunicaţie** [58] .

O proprietate importantă a metodei comutării de circuite este necesitatea de **iniţializare a conexiunii**, **adică de a stabili o cale de la un capăt la altul înainte ca informaţiile să fie transmise**.

În cazul telefoniei**, intervalul de timp** dintre momentul formării numărului şi până se aude sunând telefonul poate dura chiar şi **zece secunde** (de exemplu în cazul convorbirilor internaţionale). Odată ce conexiunea a fost stabilită, singurele întârzieri sunt date de durata de propagare a informaţiei de la un capăt la altul, fără să apară pauze sau blocaje în trafic.

La reţelele cu comutaţia de circuite pot fi referite:

- *POTS* (*Plain Old Telephone System*) -** este numele de reţeaua de telefonie veche (comparativ cu „noi" tehnologii: *VoIP, ATM, ISDN*).

POTS oferă: transmisia de date *Full duplex* de sunet cu o gamă de frecvenţă de *300-3400 Hz.*

Avantajele: preţul avantajos al dispozitivelor (telefon); o gamă largă de echipamente cu standardele utilizate (*dial-up, fax*, voce); cerinţele scăzute pentru calitatea sistemelor de cablare. **Dezavantajele**: o linie de comutaţie oferă o singură conexiune la un

moment dat; viteza de transmisie extrem de scăzute (în jurul *64Kbit/s*); pentru fiecare canal client - postul telefonic trebuie să imparte o pereche de fire de cupru; nu este prevăzută corectarea erorilor etc.

Deşi *POTS* nu reprezintă un serviciu pentru comunicaţia de date între calculatoare, este inclus în această categorie din două motive: pe de o parte, multe din tehnologiile pe care le include fac astăzi parte din infrastructura aflată în continuă creştere; pe de altă parte, este considerat un model de încredere, uşor de folosit.

- ***Narrowband ISDN*** (*Servicii Integrate Digital Network*) - **bandă îngustă** *ISDN* - a reprezentat primul serviciu digital de tip *dial-up*. Servicii *ISDN,* iniţial funcţionează prin conectarea unei linii telefonice de cupru standard. **Avantajele** - aceasta aparţine acum la o reţea digitală stabil şi foarte de încredere. Utilizarea sa depinde de la o ţară la alta, asigurând o lăţime de bandă *128 Kbps-3 Mbps*.

3.3.2. Comutaţie de pachete (*Packet-switched*) - sau **comutaţia asincronă** este o tehnică de comunicaţii digitale, care constă în **separarea mesajelor de la o gazdă în blocuri de dimensiuni reduse - denumite pachete**, pentru a fi mai apoi transmise individual prin reţea, într-o succesiune rapidă. **Pachetele sunt transportate unul câte unul** prin reţea şi depozitate la gazda receptoare, unde sunt reasamblate în forma mesajului iniţial şi furnizate procesului receptor [59].

Comutaţia de pachete este o **tehnică mai avansată** decât simpla comutaţie de mesaje, prin aceea că fixează **o limită de dimensiune a blocurilor transmise**, astfel încât să nu blocheze linia *ruter-ruter* minute întregi.

Comutaţia de pachete **se deosebeşte** de o altă tehnică importantă de reţea, **comutaţia de circuite**, în multe privinţe:

1. Comutaţia de pachete **nu necesită iniţializarea conexiunii** (circuitului) între transmiţător şi receptor.

2. **La comutaţia de circuite** - se alocă lăţime de bandă pentru întreg traseul comunicaţiei, iar toate pachetele de informaţie urmează succesiv această cale prestabilită. Din contră, la comutaţia de pachete, **pachetele pot urma o cale diferită, urmând doar ca la final să poată fi recompus mesajul iniţial**. Consecinţa practică a acestui lucru este faptul că traficul prin reţelele cu comutaţie de pachete este **taxat pe unitate de informaţie (*bit*),** în timp ce comutaţia de circuite se taxează la intervale de timp.

3. Comutaţia de pachete poate lua **trei forme**, în funcţie de tehnica de dirijare a pachetelor în reţea:

- comutaţie **pe circuit virtual** (orientată pe conexiune): pachetul de date are **asociată o etichetă de identificare a canalului temporal** alocat în cadrul multiplexului temporal;

- **autodirijare**, presupune utilizarea de **etichete** care descriu explicit direcţiile succesive;

- datagramă[17] **(fără conexiune**), utilizează **pachete de date** care conţin o etichetă de identificare a destinatarului.

La reţelele cu comutaţia de pachete pot fi referite:

- ***X.25*** **(***Packet Switching***) -** deşi este o tehnologie veche, se mai foloseşte încă. Oferă siguranţă în transmiterea datelor, dar **lăţimea de bandă** este limitată la *2 Mbps*.

- ***FR*** (*Frame Relay*) - este versiunea bazată pe comutarea pachetelor a *Narrowband ISDN* (*Integrated Services Digital Network*). A devenit cea mai populară tehnologie *WAN* asigurând o lăţime de bandă maximă de *1,544 Mbps*.

3.3.3. *Cell-switched. ATM* (*Asyncronous Transfer Mode*) - similară tehnologiei *broadband ISDN*, a devenit una din cele mai importante tehnologii *WAN*, cu o **lăţime de bandă** de *622 Mbps*.

[17] Termenul „datagramă" desemnează faptul că fiecare pachet este tratat ca o entitate individuală care nu are nici o relaţie secvenţială cu alte pachete.

ATM este o **tehnologie completă**, în sensul că implementează mecanisme de la **echivalentul nivelului 2 *ISO-OSI* pana la nivelul 7, şi** revine foarte puternic ca o soluţie potrivită în reţelele mixte de date, voce şi video [60].

Tehnologia *ATM* **se bazează pe comutarea de celule**, care pot fi văzute **ca nişte cadre foarte mici** de dimensiune constantă (**53 de octeţi din care 7 – este antet**). Datorită acestei proprietăţi de dimensiune fixă, *ATM* se pretează **bine la transportul fluxurilor de date cu comportament predictibil** (lăţime de bandă puţin variabilă), cum ar fi vocea şi video. Operatorii de telefonie mobilă din unele ţări îşi bazează întreaga reţea de voce pe tehnologie *ATM*.

3.3.4. Dedicated digital. Multiplexare în telefonie:

a) **Fluxuri *E1, E3* şi *T1, T3*** [53]. Industria telefonică foloseşte pe scară largă **multiplexarea**, ceea ce permite existenţa **mai multor convorbiri simultane**. Prima „formă" de telefonie a fost cea analogică, apoi a apărut cea digitală, care, datorită numeroaselor avantaje tehnice, a evoluat extrem de rapid.

Canalul cu **lăţimea de bandă de *64Kbps*** este folosit pentru transmiterea vocii umane în format digital. Acest canal a fost numit ***DS0*** (***Digital Signal 0***) (şi echivalentele lui ***E0*** şi ***J0***) şi reprezintă **unitatea fundamentala în telefonia digitală** în Statele Unite, Europa, Japonia, precum şi în sistemele moderne, cum ar fi sincrone *SDH / SONET*.

Primul astfel de flux se numeşte ***T1*, şi conţine 24 de canale *DS0***. În Europa, s-a standardizat fluxul **E1, care conţine 32 de canale *DS0***. Seriile *T* în Statele Unite ale Americii şi *E* în Europa au devenit cele mai importante tehnologii *WAN*. **Lăţimile de bandă corespunzătoare** sunt: *T1 – 1,544 Mbps*; *T3 – 44,736 Mbps*; *E1 – 2,048 Mbps*; *E3 – 34,368 Mbps*.

Pentru a limita numărul de cabluri necesare implicat în schimbul

de apeluri de voce, un sistem a fost construit în mai multe *DS0*-s care sunt **multiplexate împreună pe circuitele de capacitate mai mare**. În acest sistem, 24 canale *DS0*-s sunt multiplexate într-un semnal ***DS1*** - corespunzător cu ***T1*** (sârmă de cupru); 28 canale ***DS1***-s sunt multiplexate într-un ***DS3*** - corespunzător cu ***T3*** (sârmă de cupru).

b) ***xDSL* (*Digital Subsriber Line, x - for family of tehnologies*).** Serviciile tradiţionale de telefonie (numite şi *POTS - Plain Old Telephone Service*) folosesc **cabluri de cupru torsadate pentru a transmite vocea umană**. Aceste sisteme de telefonie au fost gândite pentru voce, drept care sunt optimizate pentru frecvenţe între *300 şi 3000Hz*. Cu toate acestea, **cablurile în sine permit şi implementarea unor soluţii mai performante**, astfel că a apărut o nouă tehnologie numită *DSL* (*Digital Subscriber Line*).

Prima bandă de frecvenţă (până în *20KHz*) să fie folosită **pentru telefonie**, iar restul frecvenţelor să fie folosite **pentru date**. În acest fel, pe acelaşi cablu de telefon putem avea şi telefonie normală, iar pe frecvenţele de la *25KHz* în sus se foloseşte *DSL*.

Putem spune, că tehnologie *WAN* este dedicată în special *homeuser*-ilor. Sunt incluse aici: *HDSL - high-bit-rate DSL*; *SDSL - single-line DSL*; *ADSL - asymmetric DSL*; *VDS*L - *very-high-bitrate DSL*; *RADSL - rate adaptive DSL*. De exemplu, *ADSL* se referă la **natura asimetrică a conexiunii**, adică **lăţimea de bandă folosită** pentru *download* este mult mai mare decât cea folosită pentru *upload*. Acest lucru este un avantaj pentru cei care intenţionează să-şi instaleze *ADSL* pentru a naviga pe Internet, însă nu este foarte convenabil pentru cei care doresc să ţină o pagină de *web online*.

Avantajele majore sunt că, această conexiune *DSL* este activă în permanenţă (nu este nevoie de sunat ca la *dial-up*), iar partea de telefonie poate fi folosită pentru convorbiri prin telefon obişnuite. Alt avantaj este viteza relativ mare (de ordinul a câţiva *Mbps*) comparativă cu o conexiune *dial-up* (vezi Anexa 4).

Dezavantajul este că abonatul trebuie să fie aproape de centrala telefonica, pentru că în cele două capete ale liniei (la client şi la centrala telefonică) se află câte un modem *DSL* care funcţionează la **frecvenţe mari şi limitează distanţele la care pot funcţiona**. Atunci când sunt pornite aceste două modemuri, ele negociază între ele viteza la care pot comunica. Deşi în cazul *ADSL* viteza teoretică este în jur de *8 Mbps*, practic viteza reală negociată depinde de distanţa la care se află cele două modemuri (o viteză obţinută foarte frecvent şi cu uşurinţă este *1 Mbps*). Un alt dezavantaj este faptul că viteza la care se sincronizează modemurile depinde mult de calitatea liniei telefonice.

c) Dacă se doreşte interconectarea mai **multor echipamente produse de firme diferite,** atunci se optează de multe ori pentru folosirea **încapsulării *PPP*** (*Point-to-Point Protocol*) - este un protocol de nivel 2 folosit pentru a încapsula date pe interfeţele seriale sincrone. *PPP* prezintă numeroase **avantaje** faţa de alte încapsulări existente, dintre care menţionăm:

- Este standardizată şi implementată la fel de toţi producătorii de echipamente;
- Permite folosirea pe acelaşi ruter a mai multor protocoale de nivel 3;
- Poate fi folosită pe interfeţele seriale sincrone, pe cele asincrone (atunci când facem *dial-up* folosind un modem), şi pe interfeţe *ISDN*;
- Este posibilă autentificarea.

Să detaliem funcţionarea *PPP*-ului. În primul rând, acesta are o structură ierarhică, şi anume conţine două sub-protocoale:

LCP - *Link Control Protocol* - pentru stabilirea conexiunii punct la punct;

NCP - *Network Control Protocol* - folosit pentru configurarea anumitor protocoale de nivel 3 (de exemplu, cu ajutorul *NCP*-ului primim automat un *IP* - o adresă de nivel 3 - atunci când facem *dial-up* la un *ISP* - *Internet Service Provider*).

Protocolul *PPP* suportă **compresia**, ceea ce este extrem de util atunci când avem un procesor mai puternic însă lăţimea de bandă mai mică.

Una dintre cele mai importante facilitaţi ale *PPP*-ului o reprezintă **autentificarea**. Atunci când se încearcă conectarea (fie prin *dial-up*, fie două rutere între ele prin serială sincronă) se foloseşte un protocol de autentificare care verifică dacă acea conectare este autorizată. Cele două metode de autentificare suportate de *PPP* sunt:

***PAP* (*Password Authentication Protocol*)** - Clientul (*dial-up* sau ruter) trimite combinaţia user/parolă, necriptate, până când serverul îl acceptă (dacă combinaţia e corectă) sau până când conexiunea se închide (dacă combinaţia nu e bună). Este o metodă slabă de autentificare, pentru că nu criptează parola şi pentru că **clientul este cel care trimite când vrea combinaţia**, el este cel care „începe" autentificarea.

***CHAP* (*Challenge Handshake Authentication Protocol*)** - este folosit atât la stabilirea conexiunii cât şi după aceea, periodic, după un timp aleator, pentru a verifica identitatea clientului. Cum funcţionează autentificarea: serverul trimite clientului un mesaj de „încercare" numit „*challange*". Clientul preia acest mesaj şi parola configurată, trimite un răspuns serverului. Serverul calculează un răspuns pe baza mesajului trimis şi a parolei pe care o are configurată şi compară rezultatul cu răspunsul primit de la client. Dacă mesajele coincid, înseamnă că parola pe care a folosit-o clientul pentru a genera răspunsul este identică cu parola folosită de server pentru verificare, deci identitatea clientului este verificată şi se stabileşte conexiunea. Dacă răspunsul nu se potriveşte, atunci conexiunea este închisă. Pentru a fi sigur că la celălalt capăt se află mereu clientul autentificat iniţial, serverul trimite din când în când astfel de mesaje de *challenge* şi procedura explicată mai sus se repetă.

d) Există **două sisteme de transmisie cu multiplexare sincrone**: 1) ***SDH* (*Synchronous Digital Hierarchy*)** - Ierarhia digitală sincronă, şi 2) ***SONET* (*Synchronous Optical* NET*work*).**

1) **Sistemul *SDH*** este practic sistemul European, iar sistemul SONET este sistemul American. Cele două sisteme utilizează acelaşi algoritm de multiplexare, au aceleaşi informaţii de control, au cadre de transport cu dimensiune şi structură asemănătoare. Cadrele de transport de bază nu sunt identice (au dimensiuni diferite) [61].

2) ***SONET*** - este proiectată pentru medii bazate pe **fibra optică,** poate fi implementată şi în cazul firelor de cupru. Oferă lăţimi de bandă de la *51,84 Mbps* la *9952 Mbps*. *SONET* este un standard al *ANSI* (*American National Standards Institute*) pentru transmisii de **date sincrone** pe medii optice. SONET oferă standarde pentru debite de linie de până la *39,808 Gbps*. *SONET*-ul are o serie de **avantaje** faţă de sistemele asincrone. Tehnica sa de multiplexare permite o **tactare sincronă simplificată**. Configuraţia de tip *hub* adaugă o bună **flexibilitate sistemului**, permiţând **convergenţa unor protocoale de reţea** (*ATM, IP*).

3.3.5. *Analog services* – este reprezentat prin: *Dial-up* de acces; *Cable modems; Wireless* legăturile. Analizăm aceste posibilităţi ale serviciul analogic [62]:

1) *Dial-up* de acces la distanţă - serviciu care permite computerului, utilizând un modem şi o reţea de telefonie, să se conecteze la un alt calculator pentru a **iniţializa sesiune de date**. De obicei, cu acest scop se folosesc în două puncte protocol *PPP*. Conexiune telefonică **prin intermediul unui modem** nu are nevoie de orice infrastructură suplimentară, în afară de reţeaua de telefonie. În unele ţări, acces *dial-up* la Internet rămâne cauza principală a costului ridicat de acces în bandă largă, şi, uneori o lipsă a cererii de servicii în rândul populaţiei.

Dial-up are nevoie **de timp pentru a stabili o conexiune** (de câteva secunde, în funcţie de locaţie). Costul de acces la Internet prin *dial-up* este determinat deseori de timpul petrecut de utilizator în reţea, mai degrabă decât volumul de trafic.

2) Primul sistem asimetric de ***cable modems*** de mare viteza a fost dezvoltat, a demonstrat şi brevetat de reţele hibride în 1990. Dezvoltat de *CableLabs* standard a fost numit *DOCSIS* (*Data Over Cable Service Interface Specification*) - de transmisie a datelor prin cablurile coaxiale. *Cable modems* foloseşte ca mediu de transmisie cablul TV şi asigură o lăţime de bandă maximă de *10 Mbps.*

Standardul *DOCSIS* a fost destinat să înlocuiască standardele veche, bazate pe protocoalele incompatibile între ele, şi să asigure interoperabilitatea echipamentelor de la diferiţi producători.

3) ***Wireless* legăturile** - în acest caz sunt de două tipuri: **terestre** sau prin satelit şi pentru utilizatorii „**mobili**".

- *Wireless LAN* (*Wireless Local Area Network*; *WLAN*) – reprezintă o reţea locală fără fir. Cu această metodă transmiterea datelor în reţea se efectuează prin intermediul undelor. Cele mai răspândite metode de construire a reţelelor sunt ***Wi-Fi* si *WiMAX***. Aceste tehnologii conţin multe caracteristici asemănătoare (standardele dezvoltate în baza *IEEE*, ambele încep cu „*802.*"), dar aceste tehnologii sunt destinate la rezolvarea problemelor diferite.

- *WiMAX* - un sistem de **rază mare de acţiune**, care acoperă **mile de spaţiu**, care foloseşte de obicei, frecvenţele autorizate de spectru pentru a oferi conexiune la **Internet punct-la-punct**. *WiMAX* este văzut ca o soluţie pentru accesul la Internet în **mediul rural**. Raza de acţiune a emiţătorului este de aproximativ **30-50 de kilometri**. Viteza de accesare a Internetului este de până la *70-75 Mbps,* iar costurile sunt destul de mici.

WiMAX utilizează un **mecanism bazat pe o legătură între staţia de bază şi dispozitivul de utilizator**. Această tehnologie a fost proiectată să ofere acces fără fir de bandă largă în reţele

metropolitane cu performanţe comparabile cu cablul tradiţional, *DSL* şi *T1*. **Avantajele**: abilitatea de a porni rapid acest serviciu chiar şi în zone unde ar fi greu de ajuns cu interfeţe pe bază de cablu, evitarea costurilor mari de instalare, şi posibilitatea de a depăşi limitările fizice ale infrastructurilor tradiţionale cu conexiune prin fir.

- ***Wi-Fi*** - un sistem de **rază scurtă de acţiune**, care acoperă **zeci de metri**, care utilizează **benzile de frecvenţă fără licenţă**. *Wi-Fi* este numele comercial pentru tehnologiile construite pe baza standardelor de comunicaţie din familia ***IEEE 802.11*** utilizate pentru realizarea de reţele *LAN* fără fir (*wireless, WLAN*) **la viteze echivalente cu cele ale reţelelor cu fir** electric de tip *Ethernet*. *Wi-Fi* este folosit de către utilizatori pentru a accesa reţeaua proprie.

În cazul în care *WiMAX* poate fi comparat cu telefoane mobile, *Wi-Fi* este mai mult ca un telefon fix fără fir. Datorită ieftinătăţii şi uşurinţei de instalare, *Wi-Fi* este adesea utilizat pentru a oferi clienţilor acces rapid la Internet prin diverse organizaţii. **Suportul pentru *Wi-Fi*** este furnizat de diferite dispozitive *hardware*, şi de aproape toate sistemele de operare moderne pentru calculatoarele personale (*PC*), rutere, telefoane mobile şi cele mai avansate console de jocuri.

Verifică-ţi cunoştinţele:

1) Cum poate fi realizat accesul fizic la internet?
2) Cum se efectuează conexiunile pe linie comutată şi conexiunile de bandă largă?
3) Enumeraţi şi caracterizaţi grupele în care sunt împărţite tehnologiile *WAN*.
4) Fluxurile ***E1, E3* şi *T1, T3 şi caracteristicele lor.***
5) Descrieţi modul de utilizare a cablurilor de cupru torsadate.
6) Explicaţi modul de interconectare a mai multor

echipamente produse de diferite firme.

7) Enumeraţi sistemele de transmisie cu multiplexare sincrone.
8) Cum poate fi reprezentat *analog services* şi analizaţi posibilităţile lui.

3.4. Sisteme de telefonie mobilă

Telefoanele mobile nu sunt decât transmiţătoare radio ceva mai complexe. Nici una din tehnologii de telefonie mobilă **nu a fost standardizată**. Analizăm unele din tehnologii mai detaliat.

3.4.1. *AMPS* (*Advanced Mobile Phone System*) - primul sistem mobil utilizat extins, odată cu care apare noţiunea de **telefonie celulară analogică**.

O celulă dispune de o staţie de bază (formată numai dintr-o antenă plasată pe un turn şi un echipament de calcul) şi este relativ restrânsă în dimensiune (**diametrul nu depăşeşte 20Km**).

Staţiile de bază ale celulelor dintr-o arie geografică se conectează la un singur centru, numit *MTSO* (*Mobile Telephone Switching Office*) (vezi Fig. 3.6). Într-o arie geografică mai mare, centrele *MTSO* se ierarhizează pe câteva niveluri, pentru ca în final toată reţeaua fără fir să fie conectată la reţeaua de telefonie fixă.

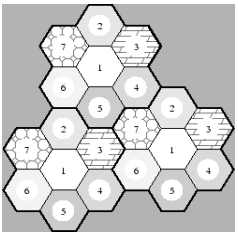


Fig. 3.6. Sistemul mobil celular [11]

Sistemul lucrează în **banda de frecvenţă *UHF* de *800MHz*, folosind 832 de canale duplex** pentru a separa conversaţiile (prin divizare în frecvenţă, *FDM* - vezi Fig. 3.7); fiecare asemenea canal

duplex are alocate **două canale simplex cu lăţimea de *30 KHz***, unul de transmisie în banda *824 - 849 MHz* şi unul de recepţie în banda *869 - 894 MHz*. **Lăţimea de bandă** a unui canal a fost aleasă astfel încât **calitatea transmisiei de voce să fie comparabilă cu transmisia de voce din sistemul de telefonie fixă**.

Canalele disponibile **se împart în şapte părţi egale** şi se asignează celulelor, pentru reutilizarea frecvenţelor în sistem. Chiar şi după împărţire, nu toate canalele transmit date (**voce în formă analogică**) în cadrul conversaţiilor; există şi: **canale de comandă**, pe care staţiile de bază le folosesc pentru managementul clienţilor, şi pe care transmisia se face digital; **canale de acces**, pentru alocarea canalelor de date; **canale de semnalizare**, pentru anunţarea apelurilor.

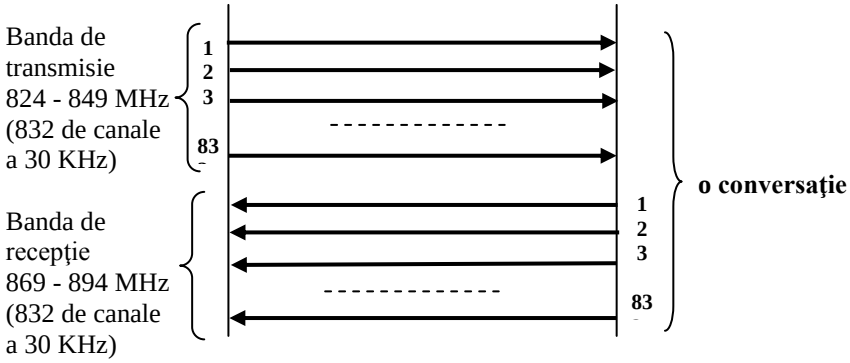


Fig. 3.7. *FDM* în sistemul *AMPS*

Fiecare telefon mobil are scris în *hardware* un **număr serial de 32 de biţi, numit *ETS* (*Electronic Serial Number*)** echivalent cu adresa *MAC* a unui calculator într-o reţea. Pe lângă acesta, „**adresa" sa logică este un număr de telefon format din 10 cifre, numit *MIN*** (*Mobile Identification Number*). La fel, fiecare serviciu de telefonie mobilă are alocat **un cod de recunoaştere de 5 cifre**, **numit *SID*** (*System Identification Code*).

La intrarea în reţea telefonul ascultă canalul de control până când aude codul de recunoaştere al serviciului, *SID*, care trebuie să fie egal cu codul serviciului programat în telefon (în cazul când clientul nu dispune de *roaming*). **Pentru a fi activat**, telefonul trimite pe canalul de comandă o cerere de înregistrare, care conţine numărul său serial *ETS* şi numărul de telefon *MIN*; staţia de bază informează centrul *MTSO* de poziţia utilizatorului, care este înregistrat în sistem. Astfel, staţia centrală *MTSO* memorează într-o bază de date celula în care se găseşte utilizatorul, pentru a putea ruta apoi convorbirile către el. Dacă *SID*-ul codat în telefon nu este egal cu cel transmis pe canalul de control, atunci conversaţiile clientului nu se pot desfăşura decât prin roaming. **Serviciul de roaming** presupune un contact între centrele *MTSO* local şi cel originar al clientului.

La generarea unui apel, telefonul client trimite pe un canal de acces datele sale de identificare şi numărul apelat. Pentru primirea unui apel, *MTSO* determină poziţia clientului destinaţie în sistem şi cererea este trimisă staţiei de bază a celulei respective; aceasta trimite un mesaj pe canalul de semnalizare conţinând numărul de telefon al staţiei destinaţie, care răspunde la mesaj. **După confirmarea prezenţei clientului destinaţie**, staţia de bază îi trimite numărul canalului pe care se va desfăşura conversaţia, care apoi poate începe.

3.4.2. *D-AMPS* (*Digital Advanced Mobile Phone System*) - este varianta digitală, care a succedat *AMPS* şi se foloseşte în *SUA* şi în Japonia. *D-AMPS* este proiectat astfel încât să fie **perfect compatibil cu varianta sa analogică** şi să poată coexista în aceeaşi celulă.

În cazul încărcării sistemului în urma creşterii numărului de utilizatori, pentru *D-AMPS* s-a alocat un set nou de canale în intervalele *1880 - 1910 MHz* pentru transmisie şi *1930 - 1990 MHz* pentru recepţie. Astfel, cele două benzi de frecvenţe, *800 MHz* şi *1900 MHz*, coexistă.

Diferenţa de bază între *D-AMPS* şi *AMPS* este faptul că în

noua tehnologie semnalul de date captat de microfon pentru transmisie este **transformat în semnal digital în telefonul clientului**. Digitizarea este urmată de compresie, prin care se micşorează cantitatea de date transmisă pentru a **transmite cel puţin trei conversaţii pe aceeaşi pereche de canale**, asigurând fiecărui utilizator o bandă de voce de *8 Kbps*.

3.4.3. *GSM* (*Global System for Mobile Communications*) – se foloseşte pe scară largă în Europa: legătură prin radio, de la un telefon celular de tip *smartphone*, de la un calculator portabil sau, mai general, de la un dispozitiv *Internet* mobil la antena celulară terestră, utilizând tehnicile *GSM* sau *UMTS* (*Universal Mobile Telecommunications System*)[18]. În ţările acoperite de *GSM*, acelaşi telefon poate opera în orice locaţie, după schimbarea *provider*-ului de servicii prin schimbarea cartelei *SIM* (*Subscriber Identification Module* - un modul care memorează datele de identificare şi de conectare la serviciu pentru un anumit *provider*) de identificare a serviciului.

Asemănările între cele două sisteme (*D-AMPS* şi *GSM*) sunt mai evidenţiate decât deosebirile, însă **cele două tehnologii rămân incompatibile**.

La fel ca *D-AMPS*, *GSM* foloseşte două benzi de frecvenţă în jurul a *900 MHz* care găzduiesc perechile de canale simplex. Banda alocată la *890.2 - 914.8 MHz* este împărţită în ***124* de canale** de transmisie de *200 KHz*, la fel ca banda alocată între *935.2 - 959.8 MHz*, pentru recepţie. Pentru *D-AMPS* există o bandă suplimentară la *1800 MHz* formată din două intervale pentru transmisie (*1710-1785 MHz*), respectiv recepţie (*1805-1880 MHz*).

[18] Este unul din standardele generaţiei a treia de comunicaţie radio mobilă *3G*. Pentru a diferenţia *UMTS* din celelalte tehnologii de reţea, *UMTS* mai este numit şi *3GSM*, subliniind combinaţia dintre *3G* şi standardele *GSM*.

Lărgimea de bandă mai mare a canalului face ca, în final, banda de voce alocată unui client să ajungă la *13 Kbps*, pentru o calitate mai bună a transmisiei.

3.4.4. *CDMA* (*Code Division Multiple Access*) – completează tehnologiile digitale pe larg folosite în *SUA* în paralel cu *D-AMPS*. În esenţă, *CDMA* este o tehnologie de **transmisie în spectru larg**, spre deosebire de *AMPS* şi *GSM*, care împart spectrul alocat în canale înguste.

Coliziunile, care în **domeniul digital** înseamnă însumarea amplitudinilor, nu sunt tratate ca nefolositoare, ci sunt folosite pentru a „extrage" din secvenţa de date suprapuse datele trimise de staţii diferite.

Pentru a evita coliziunea, transmiterea semnalelor poate fi divizată în timp (*divizia de timp*), în benţi de frecvenţă (*frequency division*) şi folosind **codurile de acces la mediu** (*divizia de cod*). *CDMA* utilizează ultimul mod de evitare a confuziilor.

În tehnologia *CDMA* **fiecare bit de informaţii este înlocuit de un cod**, în forma unei secvenţe de 64 sau 128 de biţi (vezi Fig. 3.8). Această serie este luată de la o secvenţă de pseudo-aleatoare. Prin înlocuirea fiecărui bit cu codul lui respectiv (presupunem *10 bps*), putem multiplica semnalul de ieşire pe 10, obţinând *100 kbit/s*. Astfel, transmiterea semnalului va fi efectuată de 10 ori mai repede.

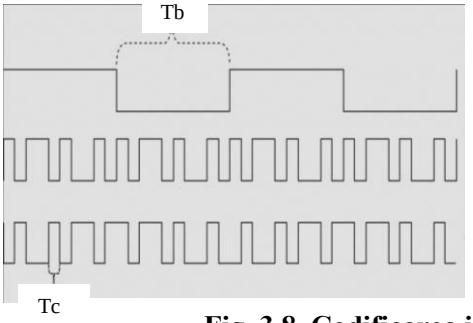
Codificatorul *CDMA* este alcătuit din două părţi, prima - **împarte secvenţa de biţi** generate de un encoder (opţional) în ***M* seturi** (de exemplu, în cazul în care succesiunea a fost +1-1+1, va fi ***M* ori de „+1", *M* ori de „-1",** etc.), şi a doua parte - **înmulţeşte fiecare set construit** (numit **codul de canalizare de lungime *M*) la codul informaţiei de date**.[19] La recepţie semnalurile pot fi decodate

[19] Pentru ca sistemul să funcţioneze, **secvenţele de acces ale tuturor**

doar de cei care au codul **exact cu codul de canalizare**.

Cu privire la codificarea *CDMA*, pot exista două cazuri, **multiplexare sincronă şi asincronă**:

Multiplexare sincronă - secvenţele generate se **înmulţesc la codurile de fiecare utilizator primite exact în acelaşi timp** de la toţi utilizatorii. Dacă în aplicarea codurilor de canalizare, toate codurile ortogonale sunt considerate 0, atunci semnalul primit reprezintă un zgomot termic. De exemplu, şir binar 1011 este reprezentat de vectorul (1, 0, 1, 1). Vectorii pot fi înmulţite scalar[20], prin însumarea scalarilor de componentele lor respective (de exemplu, dacă $a = (a, b)$ şi $v = (c, d)$, apoi produsul lor scalar $u \cdot v = ac + bd$.



Datele

T_c - Codul pseudo-aleator generat de un encoder

T_b - Codul transmis

Fig. 3.8. Codificarea informaţiei în *CDMA*

Multiplexare asincronă - secvenţele nu sunt sincronizate între

---

staţiilor din sistem trebuie să fie ortogonale reciproc. Dacă această condiţie este îndeplinită, atunci dintr-o secvenţă de date rezultată din coliziunea mai multor transmisii se poate recupera secvenţa transmisă de o anumită staţie, cunoscând codul ei de acces, făcând un simplu produs între vectorul rezultat prin coliziune şi codul de acces respectiv (vezi exemplul prezentat mai jos).

[20] În cazul în care produsul scalar este egal cu zero, doi vectori sunt ortogonale reciproc. Ortogonalitatea garantează că modificarea efectului tehnic produs de o componentă a unui sistem nici nu creează, nici nu propagă efecte secundare în alte componente ale sistemului.

ele şi, pe lângă zgomotul termic, vor fi semnalele de la alte canale cu un număr mare de coduri (**ortogonate reciproc**). Spre deosebire de *CDMA* sincrone, semnalele de alţi utilizatori vor apărea ca zgomot şi din această cauză este necesar să fie cât mai puţine.

Următorul exemplu demonstrează cum semnalul fiecărui utilizator poate fi codificat şi decodificat [63]:

În acest exemplu se utilizează coduri cu doar 2 biţi. Fiecărui utilizator i se asociază un cod diferit. Un *bit 1* este reprezentat de un cod pozitiv, şi *bit 0* este reprezentat de un cod negativ.

Dacă codul pseudo-aleator este $v=(v_0, v_1) = (1, -1)$ şi data de transmisie =(1, 0, 1, 1), atunci datele codificate sunt:
(v, –v, v, v) = $(v_0, v_1, -v_0, -v_1, v_0, v_1, v_0, v_1)$ = (1, –1, –1, 1, 1, –1, 1, –1).
Encode=M_*(*data*) - (1,1,1,1).

step	Codificare (*encode*) sender 0	Codificare (*encode*) sender 1
0	*code0* = (1,–1), *data 0* = (1,0,1,1)	*code1* = (1,1), *data 1* = (0,0,1,1)
1	*encode0* = M_*(*data*) - (1,1,1,1)= 2_*(1,0,1,1)–(1,1,1,1) = (1,–1,1,1)	*encode1* = M_*(*data*) - (1,1,1,1)= 2_*(0,0,1,1)–(1,1,1,1) = (–1,–1,1,1)

Pentru că *signal0* şi *signal1* sunt transmise în acelaşi timp, acestea se adaugă pentru a produce un semnal comun:
(1,–1,–1,1,1,–1,1,–1) + (–1,–1 –1,–1,1,1,1,1) = (0,-2,-2,0,2,0,2,0)
Acest semnal comun este numit un **model de interferenţă**. Tabelul următor explică funcţionarea lui şi arată că semnalele nu interferează unul pe altul:

Step	*Decode sender0*	*Decode sender1*
0	*code 0* = (1,–1), *signal* = (0,–2,–2,0, 2, 0, 2, 0)	*code1* = (1, 1), *signal* = (0,–2,–2,0,2,0,2,0)
1	*decode0 = signal$_*$cod0(vector)*	*decode1 = signal$_*$cod1(vector)*

2	*decode0* = ((0,–2),(–2,0),(2,0),(2,0))∗(1,–1)	*decode1* = ((0,–2), (–2,0),(2,0),(2,0)) ∗ (1,1)
3	*decode0* = ((0+2),(–2+0),(2+0), (2+0)) = (2,–2,2,2)	*decode1* = ((0 –2), (–2+0), (2+0), (2+0)) = (–2,–2,2,2)
4	*decode0* --> *data0* (2,–2,2,2) --> (1,0,1,1)	*decode1* --> *data1* (–2,–2,2,2) --> (0,0,1,1)

După decodare, toate valorile **mai mari decât 0** sunt interpretate ca 1, în timp ce toate valorile **mai mici decât 0** sunt interpretate ca 0. De exemplu, după decodificare, este *data0* (2, -2, 2, 2), dar receptorul interpretează acest lucru ca (1, 0, 1, 1). **Valorile exact 0** înseamnă că expeditorul nu transmite date, cum este prezentat în tabelul următor.

Presupunem *signal0* = (1, –1, –1, 1, 1, –1, 1, –1) este transmis singur. Următorul tabel arată decodare la receptor:

Step	Decode sender0	Decode sender1
0	*code0* = (1, –1), *signal* = (1,–1, –1, 1, 1, –1, 1, –1)	*code1* = (1, 1), *signal* = (1, –1, –1, 1, 1, –1, 1, –1)
1	*decode0 = signal∗cod0*	*decode1 = signal∗cod1*
2	*decode0* = ((1,–1),(–1,1),(1,–1),(1,–1))∗(1,–1)	*decode1* = ((1,–1),(–1,1),(1,–1),(1,–1))∗(1,1)

Când receptorul încearcă să decodeze semnalul folosind codul de *sender1* şi datele sunt toate zerouri, aceea corelaţia clarifică că *sender1* nu a făcut nici o transmisie a datelor.

3.4.5. ***EDGE*** **(*Enhanced Data Rates for GSM Evolution*)** se referă la o tehnică pentru creșterea ratelor de transmitere a datelor în rețelele de telefonie mobilă GSM prin introducerea unei metode modulare suplimentare. *EDGE* reprezintă o dezvoltare a tehnologiei GSM, care se integrează cu efort moderat în rețelele de telefonie

mobilă fără a modifica sau substitui infrastructura existentă. În esență, este necesar să se actualizeze *software*-ul de stație de bază *GSM* şi, după caz, eventual înlocuirea unor componente individuale.

3.4.6. *3G* (*Third Generation*). Telefonia mobilă analogică a format **prima generaţie de servicii mobile**. Sistemele digitale (prin dezvoltarea celor analogice) formează **generaţia** a **doua**. Serviciile mobile în viitor vor integra serviciile de digitizare a **transmisiilor de voce cu transmisiile de date**, pentru ca acelaşi dispozitiv să înlocuiască telefonul mobil, staţia de conectare la Internet, staţia de jocuri, playerul de *CD* şi *DVD* sau editorul de text.

Analizăm în scurt unele tehnologii din aşa numite „Proiecte *3G*".

- *3GPP* (*Generation Partnership Project*,) este o colaborare între asociaţii şi grupuri de telecomunicaţie în scopul definirii unui **standard comun** care să respecte recomandările *International Telecommunication Union* (*ITU*). *3GPP* **se bazează pe specificaţiile *GSM*** şi se referă la arhitecturile transmisiilor radio, reţelei centrale (*core network*) şi de servicii pentru standardul *UMTS* (*Universal Mobile Telecommuni-cations System*).

- *3GPP2* (*Generation Partnership Project* 2) este o colaborare între asociaţii şi grupuri de telecomunicaţie în scopul definirii unui **standard comun** care să respecte recomandările *ITU*. *3GPP2* **se referă la standardele *3G* bazate pe tehnologia *2G CDMA***, şi defineşte standardul *CDMA2000*.

- *3.9G* - este o tehnologie în telefonia mobilă **bazată pe standardul *3G,*** dar cu capabilităţi deja apropiate de *4G*. Va permite **transferul de date fără fir** la viteze aproape egale cu cele ale cablurilor de fibre optice, de ordinul a *100 Mbps* (faţă de maximum *7,2 Mbps* la tehnologia *3G*). Acest lucru este posibil datorită folosirii **noului sistem de telecomunicaţii** *LTE* (***Long Term Evolution***), care expandează gama frecvenţelor de aproape 10 ori în comparaţie cu cele folosite în anul 2009 în telefonia mobilă. Totodată, un telefon mobil

are nevoie de până la 4 antene, în comparaţie cu una singură la telefoanele *3G*.

Sistemul *3.9G* schimbă de asemenea felul în care este alocată transmisia datelor. **În tehnologia *3G*** transmisia datelor de către utilizatori diferiţi este alocată **ori unei frecvenţe specifice, ori unui timp anume**, în timp ce sistemul *LTE* combină aceste două metode.

3.4.7. *4G* (*Fourth generation*) - este numele generației a patra de tehnologie telefonică mobilă. Un sistem *4G* oferă **internet mobil de mare viteză**. De acest sistem pot beneficia laptopurile cu o conexiune prin **modem *USB* fără-fir**, **smartphonurile** şi alte sisteme mobile. Aplicațiile compatibile includ **televiziunea mobilă *high-definition*, televizunea *3D*, sistemele pentru conferinte video**. Recent noile sisteme de operare mobile: ***Android, iOS, Windows*-mobil intră în categoria *4G*.**

În Statele Unite *Sprint Nextel* a introdus rețele mobile *WiMAX* din 2008, iar *MetroPCS* a fost primul operator care a oferit servicii *LTE* (*Long Term Evolution*) în 2010. Modemuri *USB* fără-fir au fost disponibile de la început, în timp ce *smartphone*-urile *WiMAX* au fost disponibile din 2010, iar cele *LTE* din 2011.

Echipamentele făcute pentru diferite continente nu au fost întodeauna compatibile din cauza diferențelor de frecvență între rețele. Rețelele mobile *WiMAX* sunt disponibile în decembrie 2012 pentru piața europeană, mai exact în România, Italia și Germania.

3.4.8. *5G* (*5th generation mobile networks or 5th generation wireless systems*) – generaţia a 5-a a reţelelor de telefonie mobilă sau sistemelor *wireless* - este un standard pentru următoarea generaţie de telecomunicaţii. În prezent, *5G* nu este un termen oficial utilizat pentru o specificaţie specială sau în orice document oficial publicat de companii de telecomunicaţii sau organizaţiile de standardizare precum *3GPP, WiMAX Forum* şi *ITU-R*. Se consideră că lansarea acestor reţele va fi disponibilă în anul 2020. Lider de dezvoltare a acestei

tehnologii devine Compania chineza *Huawei*, care investeşte puternic în această tehnologie.

Nouă generaţie celulară apară fiecare 10 ani: prima - *1G* (*NMT*) în anul 1981, *2G* (*GSM*) în anul 1992, *3G* (*W-CDMA/FOMA*) în anul 2001, *4G* (*3GPP Long Term Evolution*) în anul 2010, introducerea de standard *5G* poate fi aşteptat în anul 2020 [64]. Cel mai probabil, desfăşurarea tehnologiei va avea loc în diapazonul de frecvenţă *791-862 MHz* sau *2.5-2.69 GHz* (în anul 2012 - sunt alocate pentru *LTE*).

Verifică-ţi cunoştinţele:

1) Caracterizaţi sistemele mobile *AMPS* şi *D-AMPS*.
2) Analizaţi asemănările dintre sistemele *D-AMPS* şi *GSM*.
3) Explicaţi modul de tratare a coliziunilor în domeniul digital?
4) Principiile de funcţionalitate a sistemelor *3G, 4G, 5G*.

3.5. *Bluetooth*

Bluetooth - este un standard pentru o rețea personală (*personal area network - PAN*) fără fir (*wireless*), care foloseşte **legături radio cu rază mică** de acoperire (între 10 cm şi 10 m, extensibilă la 100 m dacă puterea de transmisie este mărită), în scopul înlocuirii diverselor tipuri de cabluri, care interconectează echipamente fixe sau mobile cu un unic tip de **legătură radio**.

La 20 mai 1998 a fost fondată gruparea *Bluetooth Special Interest Group* (*SIG*), care azi are rolul de a vinde firmelor tehnologia *Bluetooth* și de a urmări evoluția acestei tehnologii [65].

O legătură *Bluetooth* lucrează în banda nelicenţiată *ISM* (*Industrial, Scientific and Medical <radio>*) de *2.4 GHz* şi este proiectată să fie durabilă prin folosirea unei tehnici de tip **salt de**

frecvenţă[21] şi a *frame*-urilor mai scurte decât celelalte tehnologii fără fir care lucrează în aceeaşi bandă (cum sunt reţelele locale standardizate de *IEEE 802.11*).

Specificaţiile sale definesc o stivă de niveluri logice independentă şi incompatibilă cu orice model existent. În plus faţă de această incompatibilitate, *Bluetooth* fixează în stiva sa de protocoale toate aplicaţiile care sunt suportate (numite *profile*), spre deosebire de specificaţiile *802.11*, de exemplu, care nu stabilesc decât modul de comunicare între echipamente.

Nivelurile inferioare *Bluetooth* specifică faptul că legătura radio acoperă prin salturi de frecvenţă banda dintre *2.402 GHz* şi *2.480 GHz*. Frecvenţele de salt, în număr de 79, se află la *1MHz* distanţă una de cealaltă, iar secvenţa acoperirii lor este *pseudo-random*.

O mini-reţea *Bluetooth* se numeşte *piconet* şi este formată dintr-un master şi unul sau mai mulţi (până la 7) slave. Toate echipamentele dintr-un *piconet* folosesc aceeaşi secvenţă de salturi în frecvenţă, secvenţă determinată de adresa în sistemul *Bluetooth* a masterului. *Offsetul* în secvenţă este dictat de timpul său curent.

Un *piconet* este un sistem *TDM*, în care *slot*-urile sunt folosite alternativ de master şi de slave pentru transmisie, astfel încât un master îşi poate începe transmisia numai în sloturile pare, iar echipamentele slave îşi împart sloturile impare. Dacă mai multe

[21] Frecvenţa purtătoarei semnalului de date modulat nu este continuă ci se schimbă periodic (1600 de ori pe secundă). Astfel secvenţa de cod nu mai modulează în mod direct semnalul de date ci este folosită în **scopul de controla un aşa numit sintetizor de frecvenţă, care este cel care alege secvenţa purtătoare,** care se va utiliza în următorul interval de salt. Secvenţa de frecvenţe pentru fiecare conexiune de comutare este pseudo-aleatoare şi este ştiută numai de către emiţător şi receptor, şi care fiecare 625μs (*un slot*) sunt schimbate. Astfel, în cazul în care există mai multe perechi de receptor-transmiţător, acestea nu interferează. Acest algoritm este o parte integrantă a sistemului de protecţie a confidenţialităţii informaţiilor transmise.

piconet-uri împart aceeaşi arie fizică, precum fiecare *piconet* are propriul său master care dictează secvenţa de salturi, *piconet*-urile vor folosi secvenţe diferite. Dacă densitatea de *piconet*-uri în aceeaşi arie fizică creşte, atunci creşte şi probabilitatea de coliziune pe o frecvenţă, iar conectivitatea se degradează. În cazul când un echipament *Bluetooth* comunică prin multiplexare cu slavele în mai multe *piconet*-uri, acestea vor fi conectate într-o reţea extinsă (numită *scatternet*).

Într-un sistem *Bluetooth* există două tipuri de **legături fizice**:

- ***SCO* (*Synchronous Connection Oriented*)** - o conexiune punct-la-punct cu bandă rezervată de *64 Kbps* între un master şi un slave din acelaşi *piconet*, fără retransmisie de pachete folosită în special pentru legături de voce. Pentru menţinerea unui *link SCO*, un master are *slot*-uri de timp rezervate (deci conexiunea este de tip *circuit-switched*) şi poate menţine până la 3 astfel de legături.

- ***ACL* (*Asynchronous Connectionless*)** - o conexiune de tip *best-effort multipunct* între master şi toate echipamentele slave din *piconet* la care se foloseşte retransmisia de pachete. Pentru menţinerea acestui tip de legătură masterul foloseşte *slot*-urile neocupate de conexiuni *SCO* (deci conexiunea este de tip *packet-switched*).

Verifică-ţi cunoştinţele:

1) Ce reprezintă *Bluetooth*?
2) Principiile de lucru a unei mini-reţea *Bluetooth.*
3) Tipurile lelegăturilor fizice ale sistemului *Bluetooth*.

3.6. *Frame Relay*

Pe lângă serviciile de acces la Internet, **există o cerere mare pentru interconectarea printr-o legătură de date a două sau mai multe puncte aflate la distanţă geografică mare**. ***Frame Relay*** este o tehnologie de nivelul doi care asigură astfel de servicii, oferind

circuite virtuale punct-la-punct [53, 66]. Putem privi o reţea *Frame Relay* ca pe **un nor[22] privat al furnizorului de servicii** la care se conectează utilizatorii. Între doi astfel de clienţi pentru care se doreşte interconectarea punct-la-punct se va crea o cale de date comutată între aceste două puncte. În nor vor exista *switch*-uri specializate care vor ajuta la crearea acestor legături virtuale.

Putem privi acest nor ca o **colecţie de „conducte" virtuale peste o reţea în care au loc comutări de cadre**, similar cu o reţea locală *Ethernet*. Marea diferenţă faţă de aceasta este că *switch*-ul de cadre nu se va face dinamic ci static, adică orice *frame* care este transmis de o staţie către o altă staţie va presupune existenţa unei căi pe care o vor urma toate *frame*-urile între aceste două staţii. Aceste căi sunt **circuitele virtuale**, care au o lăţime de bandă controlată şi garantată.

Un alt element important al *Frame Relay* este modalitatea de **comunicare în vederea stabilirii parametrilor de comunicaţie şi a sincronizării între elementele din reţeaua *frame relay*** (staţii şi *switch*-uri). Această comunicare este realizată printr-un aşa numit proces de semnalizare, produs cu ajutorul unui protocol de nivel 2 numit *LMI* (*Local Management Interface*).

Schema de adresare folosită de *frame relay* este una **plată**, în care **adresele nu sunt prestabilite**, fiind alocate sub forma unor **etichete pe fiecare nod** (staţie sau *switch*) pentru fiecare circuit virtual; aceste adrese au în consecinţă o **semnificaţie locală**, în sensul ca vom putea avea o aceeaşi adresă (etichetă) pe două noduri diferite din reţea.

---

[22] *Cloud computing* (literal: „calculare în nor", concret „calcul în Internet") este un concept modern, reprezentând un ansamblu distribuit de servicii de calcul, aplicaţii, acces la informaţii şi stocare de date, fără ca utilizatorul să aibă nevoie să cunoască amplasarea şi configuraţia fizică a sistemelor care furnizează aceste servicii. Conceptul şi termenul englez au apărut în practică prin anii 2006-2007.

Numele adreselor de nivelul doi folosite este *DLCI* (*DataLink Connection Identifier*).

Având în vedere **comportamentul predictibil** al *Frame Relay*-ului, acest tip de conexiuni sunt alese de către multe companii, mai ales pentru cazurile în care este nevoie de o lăţime de bandă fixă şi de o latenţă mică. *Frame Relay*-ul se preţează foarte bine în cazul în care este nevoie de o legătură de date sigură, ce nu va prezenta atacuri, congestii, pentru aplicaţii critice. S-au dezvoltat standarde cum ar fi *VoFR* (*Voice over Frame Relay*) ca o alternativă pentru transmis vocea în format digital, cu performanţe mult mai bune decât în cazul *VoIP*[23] (însă şi la preţuri mai mari).

Verifică-ţi cunoştinţele:

1) Ce reprezintă o reţea *Frame Relay*?
2) Avantajele tehnologiei *Frame Relay.*

3.7. *GPS* (*Global Positioning System*)

Localizarea cu acurateţe în orice poziţie de pe glob a devenit imediată, odată cu lansarea proiectului *GPS* de către Departamentul de apărare al *SUA*. Iniţial a fost utilizat pentru navigaţia militară, ulterior sistemul *GPS* a fost deschis pentru uz public. Serviciul oferit de sistem unui receptor constă în calculul poziţiei geografice în cele **trei dimensiuni ale spaţiului**, al **timpului universal** şi al **vitezei**, prin decodificarea semnalelor radio primite de la **sateliţii *GPS*.**

Sistemul *GPS* constă din trei segmente (Fig. 3.8):

1) Un segment **aflat în spaţiu**, **în fapt o „constelaţie" de 24 de sateliţi *GPS*** care orbitează Pământul. Fiecare dintre sateliţi parcurge lungimea unei orbite terestre o dată la fiecare 12 ore, la o înălţime de aproape 20.000 km. Configuraţia celor 24 de sateliţi în jurul

[23] *Voice over Internet Protocol*, numită şi **Telefonie *IP*** sau **Telefonie Internet** - este procesul de transmitere a conversaţiilor vocale umane prin legături de date de tip *IP* sau prin reţele în care este folosit acest protocol.

Pământului este proiectată astfel încât, în orice moment de timp şi în orice poziţie pe glob, între 5 şi 8 sateliţi sunt în raza de vizibilitate pe cercetări;

2) Un segment de **control al sistemului, format din staţii de monitorizare dispersate pe glob**. Aceste staţii calculează, pe baza semnalelor recepţionate de la sateliţi, câte un model al orbitelor acestora, şi corecţia de care are nevoie ceasul fiecărei staţii pentru a fi complet sincrone. Datele sunt transmise către sateliţi, care vor trimite receptoarelor *GPS* setul de efemeride;

3) **Segmentul de utilizatori, care dispun de receptoare *GPS***. Acestea calculează pe baza semnalelor primite de la cel puţin 4 sateliţi aflaţi în raza de vizibilitate poziţia geografică şi timpul universal.

4) Pentru comunicaţie sateliţii folosesc două unde purtătoare din domeniul microundelor: purtătoarea de *1575.42 MHz* este modulată de semnalul de navigaţie şi de codul de poziţionare, iar purtătoarea de *1227.60 MHz* este folosită pentru măsurarea întârzierilor în ionosferă ale codului de poziţionare, pentru un calcul precis al localizării.

5) Codul de poziţionare este o secvenţă predeterminată şi *pseudo-random* de date care modulează purtătoarea cu o frecvenţă de *1MHz* şi care se repetă la fiecare *1023 de biţi* (sau o milisecundă). Acest cod este diferit pentru fiecare satelit şi este baza de calcul a poziţiei geografice. Mesajul de navigaţie este un semnal de *50Hz* care modulează purtătoarea, semnalizând datele orbitelor sateliţilor şi datele de sincronizare a ceasului. Un receptor foloseşte codul de poziţionare, iar fiecare satelit *GPS* îşi modulează purtătoarea după acest cod. Fiecare receptor rulează intern acelaşi cod. Pentru decodificarea semnalului de la un satelit, codul intern trebuie aliniat cu cel modulat de satelit. Diferenţa de timp care rezultă din aliniere este egală cu timpul de călătorie al undei radio de la satelit la receptor, în ipoteza în care ceasurile lor sunt sincronizate. În acest fel, distanţa până la satelit poate fi calculată, luând în calcul viteza constantă a

undei radio.

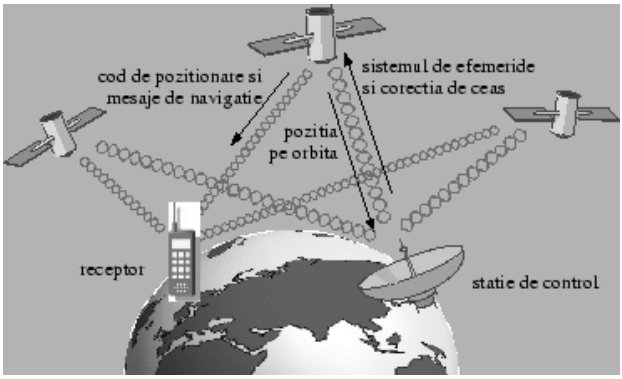


Fig. 3.8. Sistemul *GPS* [11]

Dacă un receptor măsoară în acest mod distanţa până la cel puţin patru sateliţi, poziţia sa geografică poate fi determinată ca fiind intersecţia sferelor cu centrul în poziţia exactă a satelitului (aflată de la staţiile de control, prin mesajele de navigaţie primite de la satelit), şi raza determinată (printr-un proces numit triangulaţie *3D*). Trei astfel de sfere, împreună cu sfera Pământului, sunt de ajuns pentru calculul coordonatelor geografice (*X, Y, Z*); a patra sferă calculează timpul exact, în mod sincron cu ceasul satelitului; corecţia adusă timpului local receptorului asigură şi corectitudinea calculului coordonatelor geografice.

Cum ceasul receptorului este în permanenţă sincronizat, acesta poate folosi un simplu ceas cu cuarţ în locul ceasului atomic precis cu care este echipat un satelit. Bineînţeles, deşi sistemul e bine pus la punct, **pot să apară erori** dacă: staţiile de control nu sincronizează precis ceasurile sateliţilor; nu se pot calcula corect întârzierile semnalului radio în staturile atmosferice; în cazul când undele reflectate sunt confundate cu undele directe este aproape imposibil de detectat erorile şi le corectat. Soluţia acestor probleme este sistemul *GPS* diferenţial (*DGPS*), care foloseşte principiul simplu al calculului

corecţiilor necesare într-un alt punct de referinţă, pentru fiecare satelit, şi aplicarea lor pentru receptor.

Verifică-ţi cunoştinţele:

1) Expicaţi funcţiile sistemului *GPS*.
2) Enumeraţi segmentele *GPS* şi particularităţile lor.

Întrebările pentru autoevaluare:

1. Nivelul Legătura de date. Structura generică a unui cadru.
2. Tehnici de acces la mediul de transfer in LAN
3. Scheme de adresare folosite în telecomunicaţii
4. Protocoale de comunicaţie la nivelul Legătură de date
5. Tehnici de acces pentru reţele largi (WAN)
6. Sistemele de telefonie mobilă.
7. Bluetooth. *Frame-Relay*
8. GPS (*Global Positioning System*)

Capitolul 4. Funcţii şi protocoale la niveluri Reţea şi Transport

4.1. Nivelul reţea
4.1.1. Determinarea căii optime
4.1.2. Clasificarea protocoalelor de rutare
a) Protocoale *distance-vector*
b) Protocoale *link state*
4.1.3. Sisteme autonome
4.2. Protocolul *IP*
4.2.1. Structura antetului *IP*
4.2.2. Adresa *IP* şi clasele de adrese
4.2.3. Masca de reţea
4.2.4. Subreţele (*host*-uri)
4.2.5. Prima şi ultima subreţea
4.2.6. *Supernetting*
4.3. Protocoalele de nivel Reţea
4.3.1. *ARP* (*Address Resolution Protocol*)
4.3.2. Alte protocoale în suita de protocoale Internet
4.4. Nivelul Transport. Protocoalele la nivel Transport
4.4.1. *UDP* (*User Datagram Protocol*)
4.4.2. *TCP* (*Transmission Control Protocol*)

Internetul a crescut cu mult peste scopul iniţial, evoluând de la o simplă reţea tip „coloană vertebrală" (*backbone*), spre o structură cu o ierarhie pe trei nivele (*three-tired hierarchical structure*).

Protocoalele fundamentale ale Internetului, care asigură inter-operabilitatea între orice două calculatoare sau aparate inteligente care le implementează, sunt *Internet Protocol* (*IP*), *Transmission Control Protocol* (*TCP*) şi *User Datagram Protocol* (*UDP*) [67].

Aceste trei protocoale reprezintă însă doar o parte din nivelul de bază al sistemului de protocoale Internet, care mai include şi protocoale de control şi aplicative, cum ar fi: *DNS* (*Domain Name*

System), *PPP* (*Point-to-Point Protocol*), *SLIP* (*Serial Line IP*), *ICMP* (*Internet Control Message Protocol*), *IMAP* (*Interactive Mail Access Protocol*), *SMTP* (*Simple Mail Transfer Protocol*), *HTTP* (*Hyper Text Transfer Protocol*), *HTTPS* (*HyperText Transfer Protocol/Secure*), *SSH* (*Secure Shell*), *Telnet*, *FTP* (*File Transfer Protocol*), *LDAP* (*Lightweight Directory Access Protocol*), *SSL* (*Secure Sockets Layer*), *SIP* (*Session Initiation Protocol*) (vezi Anexa 1).

4.1 Nivelul Reţea

Nivel reţea - joacă un rol important în transmisia datelor: **foloseşte o schemă de adresare** pe care se bazează echipamentele pentru a determina care este **destinaţia datelor** pe care le transmit.

Pentru a lega calculatoarele între ele avem nevoie de diverse dispozitive de interconectare. Echipamentele de nivel 3 folosite la interconectarea reţelelor sunt ***routerele***. Acestea sunt capabile să ia **decizii logice cu privire la traseul cel mai bun** pe care trebuie să-l urmeze un pachet prin reţea. Ruterul este dispozitivul ce face posibilă **scalabilitate a Internetului**, şi astfel chiar existenţa sa, astfel încât orice definiţie relevantă a Internetului trebuie să pornească mai degrabă de la rutere decât de la staţii [68].

Din punctul de vedere al modului cum funcţionează, Internetul este definit de simbioza a două tipuri de protocoale de nivel Reţea: **protocoale de rutare (***routing protocols***)** şi **protocoale rutate** (*routed protocols*):

Dintre protocoalele de rutare face parte **protocolul *IP,*** acesta fiind singurul **protocol rutabil** folosit peste Internet începând cu anii 2000.

Protocoalele rutate sunt acele protocoale responsabile pentru **asigurarea unui mod de identificare a entităţilor ce participă în Internet** prin stabilirea unei **scheme de adresare** ce trebuie să asigure unicitatea, dar şi **ierarhizarea adreselor**. Protocoalele rutate asigura adresarea (identificarea nodurilor).

Exemplu: *IP, IPX* (*Internet-work Packet eXchange/ Sequenced Packet eXchange*), *Appletalk.*

Protocoalele de rutare - determina **regulile prin care ruterele schimba informatii** despre accesibilitatea reţelelor. In functie de informaţiile furnizate de aceste protocoale se construieşte **tabela de rutare**, iar pe baza tabelei de rutare este **determinat traseul** pe care trebuie trimis fiecare pachet.

Exemplu: *EIGRP* (*Enhanced Interior Gateway Routing Protocol*), *BGP* (*Border Gateway Protocol*).

4.1.1. Sisteme autonome. Clasificarea protocoalelor de rutare. După cum **staţiile sunt grupate** în reţele pentru a oferi o ierarhizare a spaţiului de adrese şi **reţelele sunt grupate** în colecţii de reţele aflate sub o administraţie comună numite **sisteme autonome**. Un sistem autonom sau *AS* (*autonomous system*) este identificat printr-un număr, numit **număr** sau **adresă *AS***. Acest număr poate fi cuprins între 1 şi 65.535 (cu toate că ultimul segment al acestui spaţiu de adrese, şi anume numerele între 64.512 şi 65.535, sunt rezervate pentru uz privat, similar cu clasele de adrese *IP* private).

Odată cu gruparea reţelelor în sisteme autonome a apărut şi problema dezvoltării de **protocoale** care să facă faţă cerinţelor **rutării între *AS*-uri**. Astfel a fost creată distincţia între *IGP* (*Interior Gateway Protocol*) şi *EGP* (*Exterior Gateway Protocol*), adică între protocoale de rutare interne unui sistem autonom şi protocoalele de rutare exterioare unui *AS* (*inter-AS).*

Prima şi cea mai importantă cerinţă e aceea de a face faţă unor **tabele de rutare mai mari.** O tabelă de internet actuală care este schimbată între două rutere de graniţă din sisteme autonome diferite cuprinde aproximativ 180.000 de rute. A doua cerinţă este cea de flexibilitate.

O altă schimbare faţă de rutarea internă o reprezintă schimbarea paradigmei de **securitate**. Spre deosebire de rutarea internă, rutarea inter-*AS* nu lasă loc pentru existenţa mai multor protocoale diferite, datorită cantităţii importante de resurse, dar şi numărului relativ redus de sisteme autonome.

Există numeroase clasificări ale **protocoalelor de rutare**. P**rima clasificare** a protocoalelor de rutare se face în protocoale de rutare inter-*AS* (*autonomous system*), folosite pentru schimbul informaţiilor de rutare **între sisteme autonome diferite şi protocoale de rutare interne,** adică protocoale folosite în cadrul aceluiaşi sistem autonom.

A **doua clasificare** a protocoalelor de rutare se referă la clasificarea protocoalelor de rutare internă, în funcţie de **modulul de schimbare a informaţiei de rutare**. Cele trei clase în care sunt **împărţite protocoalele de rutare internă** sunt: a) protocoale bazate **pe vectori de distanţă** (*distance-vector protocols*), b) protocoale bazate **pe starea conexiunii** (*link-state protocols*) şi protocoale **hibride.**

a) Protocoale *distance-vector*. Calculul **drumului optim** se face pe bază de direcţie şi distanţa până la destinaţie, folosind direcţia respectivă. Informaţiile de rutare se schimbă numai **între ruterele învecinate**.

Cele mai populare protocoale de rutare bazate pe vectorii de distanţă sunt *RIP* (*Router IP*) şi *IGRP* (*Interior Gateway Router Protocol*) - aceste protocoale fiind uşor de configurat, iar resursele de lăţime de bandă şi timp de procesor sunt extrem de reduse.

RIP a apărut ca un efort de standardizare a unui prim protocol

de rutare la mijlocul anilor `80. *RIP* foloseşte drept metrică[24] numărul de *hopuri* sau de rutere până la reţeaua destinaţie. Pentru a **evita efectele negative ale buclelor logice a fost stabilită o metrică maximă, astfel încât orice informaţie despre o rută cu o metrică mai mare de 15 este ignorată**.

Actualizările se fac transmiţând toate informaţiile de rutare şi nu doar cele ce s-au modificat de la ultima actualizare, dar sunt trimise folosindu-se adrese de difuzare, adică pachetele de actualizare vor ajunge doar la ruterele adiacente, deoarece în mod implicit ruterele filtrează pachetele de *broadcast*.

Fiecare ruter ce primeşte un pachet de actualizare va incrementa metrica fiecărei rute conţinute în pachet cu 1, iar apoi pentru fiecare dintre rute va încerca să determine dacă nu există deja o rută cu o metrică mai bună către aceeaşi destinaţie în tabela de rutare.

b) Protocoale *link state*. Protocoalele de tip *link-state* (starea conexiunii) construiesc **o bază de date cu întreaga topologie a reţelei şi calculează drumul cel mai scurt** pe baza unui algoritm *SPF - Shortest Path First*.

Pentru **actualizarea tabelelor de rutare** se trimite într-o primă etapă **întreaga tabelă de rutare către toate ruterele** ce rulează acelaşi protocol de rutare, aceasta realizându-se prin folosirea în câmpul destinaţie a unei **adrese logice de *multicast*[25]** specifice fiecărui protocol în parte.

După această etapă de trimitere a tuturor informaţiilor, numită şi *flooding* (inundaţii), **actualizările se vor efectua doar la apariţia unei schimbări în topologie**, iar pachetele de actualizare vor conţine doar informaţii despre rutele modificate, această metodă de actualizare numindu-se **actualizare incrementală**.

Principala problemă a acestor protocoale este că fiecare dintre

[24] **Metrica** unei rute este un număr care apreciază cât de bun este un drum spre o anumită destinaţie în raport cu un set de factori specifici.
[25] Difuzare multiplă a datelor.

rutere va trebui **să construiască arborele topologic**, şi apoi să extragă rutele cu drumuri optime în acest arbore, iar acest proces necesită resurse de memorie. Cu toate acestea, datorită iniţierii procesului de actualizare odată cu apariţia modificărilor în topologie, precum şi datorită folosirii adresării *multicast*, cât şi a propagării informaţiilor de actualizare în întreaga reţea, **timpul de convergenţă pentru protocoalele *link-state* este semnificativ mai redus** decât pentru cele *distance-vector*.

4.1.2. Determinarea căii optime. Protocoalele de rutare, uneori denumite şi **protocoale de rutare dinamică**, au drept obiectiv **schimbarea informaţiilor despre reţelele cunoscute între ruterele ce rulează acelaşi protocol de rutare**. Pe baza acestor informaţii se construiesc **rute dinamice** [69].

Ruterele au o singură tabelă de rutare pentru fiecare protocol rutat, astfel încât aceeaşi tabelă de rutare va conţine atât **rutele direct conectate, rutele statice, cât şi rutele dinamice**. Un ruter poate rula unul sau mai multe protocoale de rutare.

Numărul protocoalelor de rutare ce pot fi rulate fiind limitat în general de sistemul de operare, sau de modelul ruterului[26]. Problema care apare este: că (a) acelaşi **protocol de rutare poate să furnizeze două sau mai multe rute către aceeaşi destinaţie**; (b) pot exista **două rute dinamice către aceeaşi reţea** provenite din protocoale de rutare diferite; (c) este posibil chiar să avem **o rută dinamică către o reţea direct conectată**. Astfel, deşi avem trei tipuri de rute trebuie să avem un **mecanism de comparare a rutelor** între ele sau este necesară **ierarhizarea tuturor rutelor**.

Mecanismele de ierarhizare a rutelor se numesc **distanţă administrativă**[27] şi **metrică**. Atât **metrica**, precum şi setul de

[26] Un ruter *Cisco* spre exemplu rulează în general până la 30 de instanţe de protocoale de rutare.

[27] **Distanţa administrativă** este un număr între 0 şi 255, asociat cu un tip de rută sau cu un protocol de rutare, ce permite ierarhizarea protocoalelor de rutare.

factori, sunt relevanţi pentru un anumit protocol de rutare - adică nu are sens să comparăm metricile unor rute obţinute prin protocoale de rutare diferite.

Verifică-ţi cunoştinţele:

1) Enumeraţi protocoalele fundamentale ale Internetului.
2) Rolul nivelului Reţea.
3) Care echipamentele de nivel 3 se folosesc la interconectarea reţelelor?
4) Diferenţa între protocoalele rutate şi protocoalele de rutare.
5) Clasificaţi protocoalele de rutare.
6) Precizaţi mecanismul de determinare a căii optime.

4.2. Protocolul *IP*

Elementul central al Internetului este protocolul de nivel reţea numit *IP* (*Internet Protocol*), utilizat pentru interconectarea reţelelor din Internet.

Este un protocol **fără conexiune** care permite transmiterea unor blocuri de date (*datagrame*) între surse şi destinaţii identificate prin adrese cu lungime fixă.

În cazul datagramelor foarte mari, protocolul IP realizează fragmentarea şi reasamblarea în vederea **transmiterii prin orice reţea**.

***IP* nu dispune de mecanisme care să asigure securitatea serviciului sau controlul fluxului de informaţii**.

Este apelat de protocoalele superioare pentru transferul prin reţea al datelor, apelând la rândul lui la protocoalele reţelei locale pentru transportul datelor către un echipament local. Acest echipament local (adiacent) poate fi destinaţia finală a pachetelor de date sau poate fi un nod intermediar al sistemului de comunicaţii (*router*), care trebuie să redirecţioneze datele [70, 71].

În cazul când pachetele *IP* ajung la staţia de destinaţie cu viteză mai mare decât viteza de livrare, modulul *IP* emite un *ICMP* mesaj (*Internet Control Message Protocol*) la sursa originară de informaţii arătând că datele ajung **cu viteză prea mare faţă de procesul de recepţionare**. În cazul necesităţii, semnalul sursă *ICMP* de moderare a transmisiei (la o viteză rezonabilă) este transmis la modulul *TCP* (*Transmission Control Protocol*) presupunând că acest mesaj *TCP* va pondera staţia sursă, care va reduce cantitatea de date ce se va transmite pe acea conexiune.

Modul de funcţionare a protocolului *IP* este următorul [72]:

a) **aplicaţia pregăteşte datele** şi le transmite nivelului Internet al *software*-ului de reţea;

b) nivelul Internet **adaugă acestor date un antet** (*header*), conţinând adresa de destinaţie;

c) *datagrama* este transmisă interfeţei de reţea, care **adaugă la rândul ei un antet şi transmite întreg cadrul către primul nod intermediar** al reţelei de comunicaţii, care va efectua rutarea pachetului;

d) la recepţie, un nod intermediar va decide după adresa de destinaţie prezentă în antet **care este subreţeaua** şi, implicit, **următorul nod intermediar,** către care trebuie redirecţionat pachetul;

e) în cadrul destinaţiei finale, **antetul este înlăturat şi *datagrama* se transmite nivelului Internet**, de unde este transmis nivelului Aplicaţie.

4.2.1. Structura antetului *IP*- este prezentată în Tabelul 4.1.

Câmpul „*Vers*” memorează **versiunea protocolului** căruia aparţine *datagrama* transmisă. Astfel devine posibilă tranziţia dintre versiunile aceluiaşi protocol (de exemplu: de la *IPv4* la *IPv6*).

Tabelul 4.1. **Structura antetului IP**

Vers **(0…3) 4b**	***Hlen*** **(4…7) 4b**	***TOS*** **(8…15) 8b**	**Lungime totală (16…19) 4b**	**(20…24…31) 12b**
Identificare			**Semnale**	***Fragment Offset***
Timp de viaţă		**Protocol**	**Suma de control a antetului**	
Adresa *IP* a sursei				
Adresa *IP* a destinaţiei				
Opţiuni *IP* (dacă este cazul)				
Date				
…				

Câmpul „*HLen”* (*Header Length*) specifică **cât de lung este antetul** (lungimea sa nu este constantă) în cuvinte de ***32 biti***. Aceasta este lungimea totală a informaţiei din antet. **Valoarea minimă este de *5 biţi*** şi se aplică atunci cand nu sunt prezente alte opţiuni.

Câmpul *„* ***TOS”*** (Tip serviciu) - este câmpul care permite sursei să comunice ce tip de serviciu doreşte: **fiabil**, **rapid** sau o **combinaţie**.[28]

Câmpul „Lungime totală” se referă la întregul conţinut al *datagramei*: antetul şi datele. **Lungimea maximă este de 65.535 octeţi**. La ora actuală pot fi transmise *datagrame* mai mari de această dimensiune doar în măsură în care destinatarul este capabil să le accepte.

Câmpul „**Identificare”** - prin intermediul acestui câmp, destinatarul unei *datagrame* determină cărei *datagrame* aparţine un

[28] La rândul său acest câmp conţine un subcâmp numit **precedenţă** şi 3 indicatori: ***D, T, R***. Subcâmpul precedenţă are o lungime de 3 biti şi stabileşte **priorităţile de la 0 la 7**. Cei trei indicatori (flaguri) permit sursei să stabilească care **factori o afectează cel mai mult**: *Delay* (întârzierea), *Throughput*-ul (productivitatea) sau *Reliability* (fiabilitatea). Aceste câmpuri au fost introduse pentru a sprijini deciziile pe care le au de luat ruterele.

anumit pachet. Toate fragmentele unei *datagrame* conţin aceeaşi valoare de identificare. Pentru a obţine **lungimea încărcăturii de date** se scade *HLEN* din lungimea totală.

Câmpul „Deplasamentul fragmentului” (*Fragment Offset*) este precedat de două indicatoare: *DF* şi *MF*.

- *DF* (*Don't Fragment*) - indică ruterelor să **nu fragmenteze o *datagramă*** deoarece calculatorul destinaţie nu este capabil să le asambleze la loc. Toate calculatoarele trebuie să accepte fragmente de **576 octeţi** sau mai mici.

- *MF* (*More Fragments*) este indicatorul care arată dacă **toate fragmentele unei *datagrame* au ajuns la destinaţie**. Toate fragmentele, cu excepţia ultimului au acest indicator activat.

Câmpul „Timpul de viaţă” - este **un contor** folosit pentru a **limita durata de viaţă a pachetelor**. Acest timp este măsurat în secunde, având o valoare maximă de **255 secunde.** Prin intermediul său să preîntâmpine ca un pachet să circule la infinit prin reţea. În practică ***TTL*** (*Time To Live*) contorizează doar *hop*-urile (salturile, ruterele) dintr-o reţea în alta. Când contorul ajunge la zero, pachetul este eliminat.

După ce reasamblează *datagramele*, nivelul Reţea trebuie să ştie ce să facă mai departe cu aceasta. În acest moment intervine **câmpul** „**Protocol”** care spune nivelului Reţea cărui proces de transport trebuie pasată *datagrama*.

Câmpul „Suma de control a antetului” - ajută la asigurarea integrităţii antetului *IP,* trebuie să fie recalculată de fiecare dată când antetul unei *datagrame* se modifică (de obicei la trecerea dintr-o reţea în alta) şi detectează erorile generate. Câmpurile adresă sursă şi adresă destinaţie indică cine este la originea *datagramei* şi cine este destinatarul acesteia.

Câmpul „Opţiuni IP” - permite *IP* să suporte diferite opţiuni cum ar fi securitate şi lungime variabilă ce permit dezvoltarea versiunilor viitoare ale protocolului.

Unele din cele mai importante **opţiuni** vezi Tabelul 4.2.

Tabelul 4.2. Unele opţiuni ale protocolului *IP*

Opţiune	Lungime	Descriere
2 – Securitate	11b	Cât de secretă este *datagrama*
7 – Înregistrează calea	variabilă	Fiecare ruter îşi adaugă adresa
9 – Dirijare strictă pe baza sursei	variabilă	Indică calea completă pe parcurs

Câmpul „Date" - conţine informaţii de nivelul superior şi are o lungime variabilă de până la 64 Kocteţi.

4.2.2. Adresa *IP* şi clasele de adrese. O adresă *IP* conţine informaţiile necesare pentru a transporta un pachet cu date prin reţea şi este un şir de **32 de biţi ce identifică două lucruri: o reţea** (*network*) şi **o staţie** (*host*) în cadrul acelei reţele. Forma în care sunt folosite adresele *IP* nu este cea binară, mai degrabă se reprezintă în **forma decimală a patru octeţi, separaţi prin puncte**. Valoarea maximă a fiecărui octet (în zecimal) este $\mathbf{2^8 = 256\text{-}1(zero) = 255}$.

Pentru o adresă *IP* dată: 10110001 00000100 00010110 00001000 vom separa mai întâi biţii în grupuri de **câte 8 biţi**: **10110001.00000100.00010110.00001000** şi în final vom converti fiecare grup în decimal: **177.4.22.8** [73, 74].

Porţiunea „*host*" din cadrul unei adrese *IP* se numeşte **Identificator *host*** şi reprezintă **zona prin intermediul căreia se identifică un dispozitiv dintr-o reţea**.

Încercarea de a păstra reprezentarea decimală ca model de referinţă pentru *IP* şi de a clarifica distincţia între cele două componente ale adresei *IP* a dus la **definirea claselor de adrese *IP*.** Se cunoaşte că fiecare clasă de adrese *IP* permite un număr fix de *host*-uri şi că **prima adresă din fiecare reţea este rezervată pentru a identifica reţeaua, iar ultima adresă este rezervată pentru *broadcast*.**

În Tabelul 4.3. sunt prezentate cele 5 clase definite pentru spaţiul de adrese IP.

Tabelul 4.3. Clasele definite pentru spaţiul de adrese *IP*

Clasa	Primul octet în *binary*	Prima adresă	Ultima adresă	Observaţii
A	0xxxxxxx	0.0.0.1	127.255.255.255	foloseşte 8 biţi pentru reţea şi 24 pentru staţia de lucru
B	10xxxxxx	128.0.0.0	191.255.255.255	foloseşte 16 biţi pentru reţea şi 16 pentru staţie
C	110xxxxx	192.0.0.0	223.255.255.255	foloseşte 24 biţi pentru reţea şi 8 pentru staţie
D	1110xxxx	224.0.0.0	239.255.255.255	folosită pentru adresarea de tip *multicast*
E	11110xxx	240.0.0.0	255.255.255.255	utilizată în scopuri experimentale

Adresele reţelelor au toţi biţii de staţie 0 şi nu pot fi folosite pentru o staţie. O astfel de adresă este folosită pentru **identificarea întregii reţele**, această fiind în fapt forma relevantă a oricărei adrese ce călătoreşte peste Internet. Porţiunea „*network*" din cadrul unei adrese *IP* se numeşte **identificatorul reţelei** (*network ID*).

În plus, mai există şi **adrese de difuzare, care au toţi biţii de staţie 1**. Un pachet destinat unei astfel de adrese va ajunge la toate staţiile din aceeaşi reţea.

Într-o reţea, *host*-urile **pot comunica între ele doar dacă au acelaşi identificator de reţea**. Acestea pot să partajeze acelaşi segment fizic de reţea, dar **dacă au identificatori de reţea diferiţi**, nu pot comunica decât dacă există un alt dispozitiv care să realizeze conexiunea între segmentele logice ale reţelei (sau identificatorii acestora).

Pentru identificarea staţiilor se folosesc numai adresele de clasă A până la C. În plus, există două intervale de adrese de clasă A nefolosite în Internet:

Intervalul 0.0.0.0 - 0.255.255.255 nu se foloseşte, pentru a nu fi **confundat cu ruta implicită**;

Intervalul 127.0.0.0 - 127.255.255.255 este folosit numai pentru

diagnosticarea nodului local (întotdeauna acesta va fi cel care va răspunde la apelul unei adrese din aceasta clasă).

Analizăm mai detaliat clasele definite pentru spaţiul de adrese *IP*:

1) Clasa A a fost proiectată pentru a satisface **cerinţele ridicate de reţele de mari dimensiuni**. Ele sunt definite de **valoarea zero a primului bit** din adresa *IP*.

Astfel, **pentru definirea reţelei** va fi folosit doar primul octet (fără primul zero fixat: 2^{8-1}=128), rămânând pentru **identificarea staţiei** 24 de biţi, adică mai mult de 2^{24}=16,7 milioane de posibilităţi.

2) O clasă de adrese B este definită de valorile primilor doi biţi din adresa *IP*, aceşti primi doi biţi fiind 10. Respectând această constrângere rezultă că toate adresele *IP* ale căror prim octet se află între 10000000 şi 10111111, adică între 128 şi 191, aparţin unei clase B. Câmpul de reţea pentru o clasă B va cuprinde primii doi octeţi, dar cum primii doi biţi ai primului octet sunt fixaţi, ne rămân doar 14 biţi pentru a crea clasa B. Pentru definirea staţiilor vom avea la dispoziţie ultimii doi octeţi, adică 16 biţi. Astfel vom obţine 2^{16-2}=16.384 reţele, fiecare având un număr maxim de staţii de 2^{16}=65.536.

3) Clasele C se definesc prin alocarea primilor 3 octeţi pentru definirea reţelei şi doar a ultimilor 8 biţi pentru distingerea între staţiile aceleiaşi reţele. Primii trei biţi din primul octet trebuie să fie 110, adică valoarea acestui prim octet trebuie să se afle între 192 şi 223 pentru ca o adresă să aparţină unei clase C. Deşi numărul reţelei clasei C depăşeşte 2^{24-3}=2,… milioane, numărul de staţii pentru fiecare dintre aceste reţele este de doar 2^{8}=256.

Clasa A este rezervată organizaţiilor guvernamentale din lumea întreagă; **clasa B** este rezervată organizaţiilor medii-mari, iar **clasa C** este rezervată oricărui alt tip de organizaţie. Clasele A şi B la un loc

reprezintă 75% din spaţiul de adrese disponibile, aceste clase **fiind epuizate în primii ani de expansiune a Internetului (`92 -`94).**

Tabelul 4.4. Numărului de octeţi alocaţi pentru câmpul staţie

Clasa A/ octeţi	**Reţea**	**Staţie**		
	1	**2**	**3**	**4**
Clasa B/ octeţi	**Reţea**		**Staţie**	
	1	**2**	**3**	**4**
Clasa C/ octeţi	**Reţea**			**Staţie**
	1	**2**	**3**	**4**
Clasa D/ octeţi	**Staţie**			
	1	**2**	**3**	**4**
Adresarea IP				

4) **Clasa de adrese D** este folosită pentru **reţele *multicast***. Pentru adresa *multicast* spaţiul de adrese este plat, toţi cei 4 octeţi fiind folosiţi pentru **definirea adresei de staţie**. Deoarece primii 4 biţi ai primului octet sunt fixaţi, şi anume 1110, numărul adreselor de *multicast* este de 2^{32-4}=268,… milioane.

5) Clasa de adrese E este rezervată şi nu poate fi folosită în reţelele publice, sau în soluţii de *multicast*.

Tabelul 4.4. sumarizează tipurile de adrese IP, prezentând ponderea numărului de octeţi alocaţi pentru câmpul staţie din totalul celor patru octeţi ce formează o adresă IP. Un ruter va folosi aceste adrese pentru a transmite datele în Internet.

Când se transmit date către toate echipamentele dintr-o reţea trebuie creată **o adresă de *broadcast* (difuzare).** *Broadcast*-ul apare când staţia sursă transmite date către toate celelalte dispozitive din reţea. Dar pentru a fi sigură că toate aceste dispozitive sunt „atente”

la mesajul *broadcast*, staţia sursă trebuie să folosească o adresă *IP* pe care să o recunoască toate celelalte echipamente din reţea. De obicei, într-o astfel de adresă, biţii din porţiunea *host* au toţi valoarea 1.

4.2.3. Masca de reţea. Blocarea creşterii Internetului au apărut începând cu mijlocul anilor `80. Astfel în 1985 a fost introdus încă un nivel ierarhic în formatul de adresare *IP*.

Adresele *IP* vor avea în continuare 32 de biţi, dar aceştia nu vor mai fi grupaţi doar în două câmpuri: reţea şi staţie, ci vor putea aparţine unui **nou câmp - subreţea**.

Odată cu subreţelele a apărut distincţia între adresarea ce ţine cont doar de cele trei clase: A, B şi C, aceasta fiind numită **adresare *classful***, şi noul tip de adresare, ce oferă suport pentru câmpul de subreţea, aceasta din urmă fiind numită **adresare *classless*** [75, 76].

Masca de reţea este un şir de 32 de biţi care, în conjuncţie logică cu o adresă *IP*, va separa adresa de reţea, anulând biţii de staţie. Fiecare bit din masca de reţea ce corespunde (se află pe aceaşi poziţie) cu **un bit din câmpul de reţea va avea valoare 1, în timp ce toţi biţii corespunzători câmpului de staţie vor avea valoarea zero.**

De exemplu:

(1) Adresa IP: 11000000.10101000.00000001.00000010
(192.168.1.2) - Clasa C

(2) Masca de reţea: 11111111.11111111.11111111.00000000
(255.255.255.0) /24

(3) Adresa reţelei: 11000000.10101000.00000001.00000000
(192.168.1.0)

***Remarcă:* (3)=(1) AND (2)**

Măştile de reţea **sunt inutile** într-un mediu ce oferă adresare *classful*, deoarece simplă testare a valorii primului octet faţă de 128 şi 192 ne-ar oferi toate informaţiile necesare despre numărul biţilor ce aparţin **câmpului reţea** dintr-o adresă *IP* dată. În schimb, odată cu

apariţia adresării *classless*, masca de reţea a devenit „piatra de temelie” în deciziile de rutare.

Tabelul 4.5. Reprezentarea măştilor de reţea

Clasa	Reprezentarea deximală	Reprezentarea ca prefix (numărul de biţi de 1)
A	**255.0.0.0**	**/8**
B	**255.255.0.0**	**/16**
C	**255.255.255.0**	**/24**
Măştile de reţea pentru clasele rutate		

Reprezentarea măştilor de reţea folosită cel mai des este cea decimală, datorită similitudinii cu forma de exprimare a adreselor *IP*. Această formă de reprezentare a măştilor de reţea este sub forma unui număr ce reprezintă **numărul de biţi de 1** din masca de reţea (vezi Tabelul 4.5.), această formă de reprezentare fiind **referită ca prefix de reţea.**

Reprezentarea decimală a măştilor de reţea este mai populară decât reprezentarea ca prefix.

4.2.4. Subreţele (*host*-uri). În **adresarea *classful*** aveam trei dimensiuni de reţele, ducând la o utilizare extrem de ineficientă a spaţiului de adrese. Întrebarea este dacă în loc de o întreagă clasă B nu am fi putut aloca doar jumătate de clasă B, dublând astfel eficienţa alocării de adrese.

În cazul înjumătăţirii unui spaţiu de adrese, va trebui să **înjumătăţim numărul de staţii**, adică să reducem cu unu numărul de biţi de staţie. Bitul astfel obţinut va intra în componenţa unui nou câmp, pe care îl vom numi câmp de subreţea [77].

Masca de reţea va avea **valoarea 1** atât în câmpurile corespunzătoare **biţilor de reţea**, cât şi în câmpurile corespunzătoare **biţilor de subreţea**. În concluzie, pentru a înjumătăţi un spaţiu de adrese, trebuie să extindem masca de reţea cu un bit (cel corespunzător câmpului de subreţea), iar cele două jumătăţi vor fi

obţinute făcând **acest bit o dată 0, o dată 1**.

Tabelul 4.6. Rezultatul operaţiei de înjumătăţire a unui spaţiu de adrese (130.170.0.0)

130.	**170.**	**0.**	**0**	**/16**	**Spaţiul iniţial**
10000010	**10101010**	**00000000**	**00000000**	**/16**	
130.	**170.**	**0.**	**0**	**/17**	**Prima jumătate**
10000010	**10101010**	**0 0000000**	**00000000**		
130.	**170.**	**128**	**0.**	**/17**	**A doua jumătate**
10000010	**10101010**	**1 0000000**	**00000000**		
Înjumătăţirea unei clase B					

Tabelul 4.7. Formarea subreţelelor

130.	**170.**	**132.**	**0**		**Adresa *IP* iniţială nr. 1 (clasa B)**
10000010	**10101010**	**10000100**	**00000000**	**AND**	
11111111	**11111111**	**10000000**	**00000000**		**Masca de reţea (/17)**
130.	**170.**	**128.**	**0**	**/17**	**Prima reţea rezultantă**
10000010	**10101010**	**10000000**	**00000000**		
130.	**170.**	**32.**	**0**		**Adresa *IP* iniţială nr. 2 (clasa B)**
10000010	**10101010**	**00100000**	**00000000**	**AND**	
11111111	**11111111**	**10000000**	**00000000**		**Masca de reţea (/17)**
130.	**170.**	**0.**	**0**	**/17**	**A doua reţea rezultantă**
10000010	**10101010**	**00000000**	**00000000**		

Având de înjumătăţit o clasă B, cele două jumătăţi vor avea masca de reţea /17, bitul de subreţea fiind chiar al (16+1=) 17-lea bit din adresa *IP*. Rezultatul operaţiei de înjumătăţire este prezentat în Tabelul 4.6.

Modul de utilizare a unei măşti de reţea reiese direct din definiţia acesteia. De exemplu, fie că avem unele adrese ce se aflau **în spaţiile iniţiale de adrese**, dar după înjumătăţire au ajuns în reţele diferite. Fie 130.170.132.0 şi 130.170.32.0 aceste adrese (vezi Tabelul 4.7.).

Astfel, subreţelele au apărut în scopul eficientizării modului de alocare a adreselor *IP*. Pentru a împărţi în subreţele un spaţiu de adrese dat, o parte din biţii de staţie sunt trecuţi într-un nou câmp, cel de subreţea, acesta având rolul de a oferi un al treilea nivel de ierarhizare a adreselor *IP*.

Din punctul de vedere al unui ruter, orice adresă *IP* are doar două niveluri de ierarhizare, şi anume **reţea şi staţie.** Astfel procesul de creare de subreţele se traduce în transferarea unui număr de biţi din câmpul staţie în reţea, extinderea măştii de reţea cu un număr egal cu numărul de biţi transferaţi. Într-un mediu *classless*, nu există nici o diferenţă în modul cum ruterele sau calculatoarele tratează adrese aparţinând unei reţele sau a unei subreţele. De fapt, prin reţele se înţelege totalitatea subreţelelor, clasele de adrese fiind privite ca un caz particular al acestora.

4.2.5. Prima şi ultima subreţea. În momentul când creăm subreţele, este uşor de observat posibila confuzie ce se poate face între adresa de reţea a spaţiului de adrese iniţial şi adresa de reţea a primei subreţele create, dar totodată şi între adresa de difuzare pentru spaţiul de adrese iniţial şi adresa de difuzare a ultimei subreţele.

În exemplul de mai înainte (vezi Tabelul 4.7.) singura diferenţă între clasa B şi prima ei jumătate era **masca de reţea** folosită, şi tot masca de reţea este singura diferenţă între adresa de difuzare a clasei B şi adresa de difuzare a celei de a doua jumătăţi (vezi Tabelul 4.8.). Datorită acestei ambiguităţi**, odată cu apariţia subreţelelor a apărut şi restricţia de a folosi prima şi ultima subreţea**. Astfel, **numărul maxim de subreţele** ce poate fi folosit devine $\mathbf{2^n - 2}$, unde **n este număr de biţi de subreţea**.

Primă consecinţă este imposibilitatea împrumutării unui singur bit pentru crearea de subreţele, adică imposibilitatea înjumătăţirii unui spaţiu de adrese. Numărul minim de biţi ce trebuie împrumutaţi este 2.

A doua consecinţă - este pierderea unui procent din spaţiul de adrese în urma procesului de creare de subreţele.

Tabelul 4.8. Obţinerea primelor două subreţele

130.	**170.**	**0.**	**0**		**Adresa de reţea pentru spaţiul iniţial**
10000010	**10101010**	**00000000**	**00000000**	**/16**	
130.	**170.**	**128.**	**0**		**Adresa de reţea pentru prima jumătate**
10000010	**10101010**	**1 0000000**	**00000000**	**/17**	
130.	**170.**	**255.**	**255**		**Adresa de difuzare pentru spaţiul iniţial**
10000010	**10101010**	**11111111**	**11111111**	**/16**	
130.	**170.**	**255.**	**255**		**Adresa de difuzare pentru prima jumătate**
10000010	**10101010**	**1 1111111**	**11111111**	**/17**	

Singurele dispozitive din reţea ce ar fi putut comite erori în urma acestei ambiguităţi sunt ruterele, iar ruterele, odată cu implementarea *CIDR*[29] şi dezvoltarea protocoalelor de rutare *classless*, au avut la dispoziţie măştile de reţea, pentru fiecare dintre rute: începând cu 1996 majoritatea ruterelor au fost fabricate cu **capacitatea evitării confuziei cauzate de folosirea primei şi ultimei subreţele**.

Deşi la ora actuală multe cărţi de calculatoare recomandă evitarea folosirii primei şi ultimei subreţele, Internetul conţine echipamente capabile să evite eventualele confuzii.

---

[29] *CIDR* (*Classless Inter Domain Routing*) - se referă la modul de reprezentare a adreselor *IP* în tabela de rutare şi la modul de trimitere a mesajelor de actualizare. În notaţia *CIDR*, adresa *IP* este reţinută întotdeauna împreună cu masca de reţea.

4.2.6. *Supernetting*. O a doua componentă a *CIDR*, este procesul invers - **posibilitatea agregării mai multor spaţii de adrese într-un singur spaţiu** [53]

Faptul că în tabela de rutare este precizată şi masca de reţea permite agregarea reţelelor vecine, reducând dimensiunea tabelei de rutare. De exemplu, reţelele 192.0.2.0/24 şi 192.0.3.0/24 vor fi reţinute ca 192.0.2.0/23:

192.0.2.0/24 = 11000000.00000000.00000010 / 00000000
192.0.3.0/24 = 11000000.00000000.00000011 / 00000000

192.0.2.0/23 = 11000000.00000000.0000001 / 0.00000000

Procesul de creare de super-reţele, numit şi proces de agregare de adrese, este extrem de important mai ales pentru optimizarea funcţionării ruterelor. Agregarea adreselor are drept consecinţă reducerea dimensiunii tabelei de rutare, care în final se traduce în latenţă mai scăzută a rutării.

Verifică-ţi cunoştinţele:

1) Funcţia protocolului *IP*.
2) Modul de funcţionare a protocolului *IP*.
3) Structura antetului *IP*.
4) Adresa *IP* şi clasele de adrese.
5) Masca de reţea.
6) Subreţele (*host*-uri).
7) Prima şi ultima subreţea.
8) *Supernetting.*

4.3. Protocoalele de nivel Reţea: *ARP* şi *RARP*

Stiva de protocoale *TCP/IP* conţine **două protocoale de nivel Reţea**: ***ARP*** (*Address Resolution Protocol*) şi ***RARP*** (*Reverse Address Resolution Protocol)* [53, 78]. Aceste protocoale nu sunt întâlnite la toate reţelele, ci numai la o parte din ele.

Pentru ca două dispozitive de reţea să poată comunica, este necesară cunoaşterea atât a adresei *MAC*, cât şi a celei logice. În cazul în care numai una dintre adrese este disponibilă se apelează la un protocol, care pe baza acesteia va determina cealaltă adresă.

4.3.1. *ARP* (*Address Resolution Protocol*) - se bazează pe construirea şi menţinerea unei **tabele *ARP***. O tabelă *ARP* are rolul de a oferi o **corespondenţă între adresele *IP* şi cele *MAC***. Astfel, ***ARP*** - este un protocol care traduce adresa Internet în adresă *hardware,* oferă adresa *MAC* a unui dispozitiv de reţea dată fiind adresa sa *IP*. Acestea sunt construite dinamic şi sunt stocate în memoria ***RAM***. Deşi există mecanisme pentru adăugarea sau eliminarea unei intrări într-o tabelă *ARP,* acestea sunt rareori folosite.

Fiecare computer sau dispozitiv de reţea îşi păstrează propria sa tabelă *ARP*.

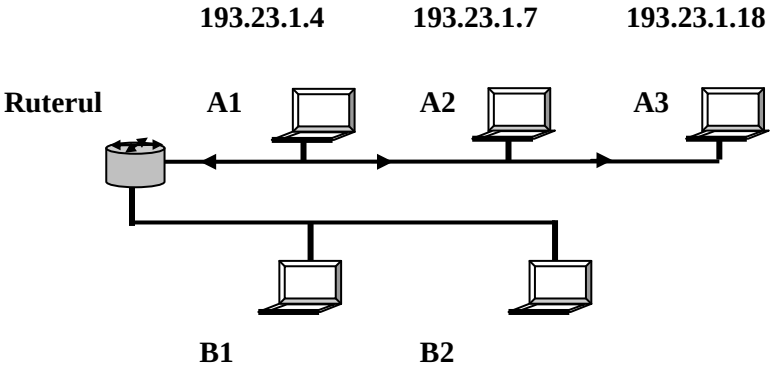


Fig. 4.1. Crearea tabelei *ARP* [5]

Fie reţeaua din figura 4.1. Toate staţiile sunt tocmai pornite, astfel tabelele *ARP* sunt vide. Presupunem că staţia A1 vrea să comunice cu staţia A2, cunoscând doar adresa *IP* a acesteia. **La nivelul Reţea** datele venite de la nivelurile superioare **vor fi încapsulate** şi vor primi un **antet** ce va conţine în câmpul **Adresă destinaţie** 193.23.1.7, iar ca **Adresă sursă** 193.23.1.4.

Înainte de trecerea **la nivelul Legătură de date** adresa *IP* destinaţie va fi căutată în tabela *ARP* şi nefiind găsită se va crea un **cadru special** (*ARP request*) ce va avea în câmpul **Adresă destinaţie** din antet adresa de difuzare: *FF.FF.FF.FF.FF.FF,* iar în câmpul **Adresă sursă** adresa *MAC* a staţiei A1.

Dacă vom considera că reţeaua din figură foloseşte *Ethernet* drept protocol de nivel *MAC,* datele vor fi difuzate şi vor ajunge la A2 şi la interfaţa ruterului conectată la segmentul A.

La nivelul Legătură de date va fi analizat antetul cadrului. Câmpul „**Destinaţie"** fiind o adresă de difuzare, cadrul va fi trimis la nivelul superior. Totodată, pe baza conţinutului câmpului „**Sursă"** de nivel 2 şi 3, va fi creată prima intrare în tabela *ARP* a staţiei A2. Ajuns la nivelul 3 cadrul este identificat drept o cerere *ARP*, şi se iniţiază un răspuns transmis atât la nivel Reţea cât şi la nivel Legătură de date.

După primirea răspunsului, A1 va putea insera în tabela sa *ARP* adresa *MAC* a lui A2, iar comunicaţia din acest moment va avea loc fără probleme.

Fiind pe un segment *Ethernet* toate cadrele schimbate de A1 şi A2 vor ajunge la toate staţiile de pe segment, astfel că, deşi nu au emis nici un cadru, atât A3 cât şi ruterul vor avea câte o tabelă *ARP* cu 2 intrări. Aceste intrări expiră după o perioadă de timp, fiind înlăturate din tabela *ARP*.

Comunicaţia între staţii aflate în reţele diferite are loc în modul următor: protocolul de rezoluţie a adresei se bazează pe difuzări la nivel Legătură de date. Ruterele, în schimb, nu propagă pachetele de difuzare de nivel Legătură de date în afara reţelei din care provin.

Există două modalităţi prin care staţii aflate în reţele diferite pot comunica: (a) *proxy ARP* şi (b) *default gateway*.

a) Proxy ARP - este o **extensie a protocolului de rezoluţie a adresei**. Pornind de la faptul că ruterul nu va transfera pachetele de difuzare, *Proxy ARP* va determina ruterul să răspundă la toate

cererile *ARP* destinate unor adrese în afara reţelei cu propria sa adresă *MAC*.

În cazul reţelei de mai sus, să considerăm că staţia A1 vrea să comunice cu B1. După ce nu va găsi adresa *MAC* a staţiei B1 în tabela *ARP* va **trimite o cerere *ARP***. Cadrul va fi recepţionat de către toate dispozitivele de reţea aflate pe acest segment.

Staţiile A2 şi A3 deja au în tabela *ARP* informaţii despre A1, astfel încât vor **reseta timpul de viaţă** al acestei intrări. Ruterul va reseta şi el acest timp, iar apoi, analizând adresa *IP* destinaţie, va concluziona că destinaţia nu se află în acelaşi segment. Dacă acesta ar fi fost un **cadru obişnuit**, ruterul ar fi luat o decizie pe baza tabelei sale de rutare. Fiind totuşi o **cerere *ARP*,** ruterul va genera un **răspuns *ARP*,** ce va conţine propria sa adresă *MAC*.

Răspunsul *ARP* va fi încapsulat, iar antetul va avea atât la nivelul **Legătură de date** cât şi la nivelul **Reţea** în câmpul „**Adresă sursă"** adresa interfeţei ruterului ce se află conectată la reţea. Ruterul va determina pe ce interfaţă trebuie să trimită pachetele destinate pentru 24.8.17.2 şi va trimite pe această interfaţă o nouă cerere *ARP*. B1 va răspunde la aceasta.

În final toate staţiile din reţeaua A îşi vor adăuga o nouă intrare în tabela *ARP* ce va face corespondenţa între 193.23.1.18 şi adresa *MAC* a interfeţei ruterului. În plus staţia A1 va mai adăuga o intrare ce va mapa 24.8.17.2 cu adresa *MAC* a interfeţei ruterului. Staţiile din reţeaua B vor insera două intrări în tabelele *ARP* proprii.

Din acest moment staţia A1 va încapsula transmisia destinată staţiei B1, folosind adresa *IP* a lui B1 şi adresa *MAC* a ruterului. Ruterul va primi cadrele, va înlocui **Adresa sursă** din antetul de nivel **Legătură de date** cu **Adresa sa** şi le va trimite mai departe către B1.

b) Pentru o **staţie dată *default gateway*** este adresa *IP* a interfeţei de pe ruter, ce conectează reţeaua din care face parte respectiva staţie. Odată precizat un *default gateway*, **nivelul Reţea** va mai căpăta o nouă atribuţie, trebuind să determine dacă destinaţia este sau nu în aceeaşi reţea.

Dacă nu este, atunci se va folosi adresa *IP* a destinaţiei finale şi adresa *MAC* a *default gateway*. Astfel, în tabela *ARP* va fi căutată adresa interfeţei ruterului.

4.3.2. Alte protocoale în suita de protocoale Internet. Trebuie menţionat că mai există şi alte protocoale în suita de protocoale Internet, cum ar fi: ***RARP* (*Reverse Addres Resolution Protocol*), *GGP* (*Gateway-to-Gateway Protocol*)** etc. Dar aceste protocoalele au modul de utilizare mai ingust şi specific:

a) ***RARP* (*Reverse Addres Resolution Protocol*) -** este protocolul care realizează aplicaţia inversă protocolului ***ARP***, traducând adresa *hardware* în adresă Internet.

b) **Porţile (*gateway*)** - sunt combinaţii de *hard* şi *soft* care realizează o legătură între două tipuri diferite de reţele. Poarta are rolul de a **transfera informaţiile şi de a le converti într-un format compatibil** cu protocoalele utilizate de reţeaua destinaţie. De exemplu, prin porţi se pot stabili legături **între diferite sisteme *e-mail***, astfel încât utilizatorii să poată realiza schimburi de mesaje fără probleme de incompatibilitate a echipamentelor folosite.

Protocolul *Gateway-la-Gateway* (*GGP*) este un protocol definit pentru rutarea *datagramelor* între *gateway*-urile în Internet. Principiul de funcţionare a acestui protocol constă în aceea că **fiecare mesaj are un antet *GGP*** - câmp care identifică **tipul de mesaj şi formatul câmpurilor rămase**.

Protocolul *Gateway-la-Gateway* a fost conceput ca un *Internet Protocol* similar protocolului *TCP* şi *UDP*.

c) **Protocolul *VMTP* (*Versatile Message Transaction Protocol*) –** este destinat suportului de tranziţie a informaţiei prin *Remote Procedure Call* (*RPC*). Funcţia de bază a acestui protocol implică **securitate, schimburi de mesaje asincrone, *multicast*** etc.

Verifică-ţi cunoştinţele:

1) Descrieţi funcţiile protocoalelor de nivel Reţea.
2) Explicaţi principiile de lucru a protocolului *ARP* (*Address Resolution Protocol*).
3) Analizaţi destinaţia protocolului *RARP* (*Reverse Addres Resolution Protocol*).
4) Protocoalele *GGP* (*Gateway-to-Gateway Protocol*), *VMTP* (*Versatile Transaction Protocol*).

4.4. Nivelul Transport. Protocoalele la nivel Transport

Principala sarcină a nivelului 4 - o reprezintă **transportarea şi controlarea fluxului informaţional de la sursă către destinaţie**. Informaţiile trebuie să ajungă în mod sigur „curate" la destinaţie.

Este important de specificat faptul că reţelele conectate la internet folosesc protocoalele *TCP/IP*, iar foarte multe organizaţii (neafiliate guvernamental) au realizat propriul internet utilizând aceleaşi protocoale *TCP/IP*. Tot ce se referă la suita de protocoale *TCP/IP*, se foloseşte cuvântul Internet scris cu literă mare, pentru a evita confuzia cu interneturile care folosesc alte protocoale diferite de *TCP/IP* [79, 80].

La nivelul Transport *TCP/IP* foloseşte două protocoale: *UDP* (protocol fără conexiuni) şi *TCP* (protocol orientat pe conexiune).

4.4.1. *UDP* (*User Datagram Protocol*)

1) Caracteristici *UDP*

UDP - este un protocol de nivel Transport construit special pentru a oferi un **serviciu de comunicare cât mai simplu peste *IP***. *UDP* este proiectat pentru aplicaţiile care nu trebuie să recompună segmentele cu date: protocoalele de la nivelul aplicaţii sunt direct răspunzătoare de siguranţa datelor transmise. Acest protocolul **nu este orientat la conexiune** şi este folosit pentru a transmite *datagrame* fără a fi nevoie de confirmarea recepţiei sau de garantarea

transmiterii acestora. Specificaţiile protocolului au fost publicate în 1980 [81]. Retransmiterea datelor în caz de erori trebuie „ordonată" de alte protocoale.

Unele caracteristici ale UDP-ului sunt:

- *UDP* este potrivit pentru scopuri, unde **verificarea şi corectarea eroarelor nu este necesară**, evitând de o astfel de prelucrare la nivel de reţea;
- *UDP* este un serviciu de tip *datagramă*: **cererile** de trimitere de date primite de la nivelul superior sunt **tratate independent**;
- *UDP* trimite *datagrame UDP* corespunzătoare cu dimensiunea datelor primite **de la nivelul Aplicaţie**;
- comunicarea are loc fără stabilirea unei legături (*conection-less*): **nu există mecanisme de stabilire şi terminare a unei conexiuni** deoarece toate datele sunt trimise în cadrul unui singur pachet *IP*, care potenţial va fi supus fragmentării;
- *UDP*-ul produce pentru fiecare transfer cerut de nivelul Aplicaţie, un **pachet *IP*** (care ulterior poate fi supus fragmentării) care, dacă ajunge la destinaţie corect, va fi livrat direct nivelului Aplicaţie;
- **nu se garantează ajungerea la destinaţie** a datelor (*best effort*): ajungerea la destinaţie nu este anunţată sursei;
- datele transportate sunt protejate de o **sumă de control**.

Utilizarea *UDP*-ului:

- servicii de rezolvare a numelor (*DNS - Domain Name System*) - întrebările şi răspunsurile scurte pot fi mai eficient implementate peste *UDP*;
- fluxuri multimedia - mecanismele complicate de control al fluxului ale *TCP*-ului ar deprecia interactivitatea;
- server de fişiere (*NFS - Network File System*) - acest tip de aplicaţii sunt în general rulate în reţele locale cu performanţe ridicate care nu necesită mecanismele *TCP*;

- managementul reţelei (*SNMP - Simple Network Management Protocol*);

- protocoale de rutare (*RIP - Router IP*).

2) Formatul pachetelor *UDP* (*User Datagram Protocol***) (**vezi Tabelul 4.9.**)**

Datagramele UDP sunt formate dintr-un antet urmat de datele care se doresc transmise.

Tabelul 4.9. Structura unei *datagrame UDP*

Port sursa (16 biţi)	**Port destinaţie (16 biţi)**
Lungime pachet *UDP* (16 biţi)	**Suma de control (16 biţi)**
Date	

Antetul cuprinde:

- Port sursă - împreună cu adresa *IP* a sursei, acest număr identifică în mod unic locul de unde a fost trimis *datagrama UDP*;

- Port destinaţie - împreună cu adresa *IP* a destinaţiei, acest număr identifică în mod unic destinaţia dorită pentru *datagrama UDP*;

- Lungimea pachetului *UDP* - lungimea minimă măsoară *datagrama UDP* cu antet şi prin urmare are o valoare minimă de 8 octeţi;

- Sumă de control - acoperă întreg pachetul *UDP* cât şi un pseudo-antet de **12 octeţi** (**96 biţi**) format din: adresa *IP* a sursei - **32 biţi**; adresa *IP* a destinaţiei - **32 biţi**; **8 biţi de zero** pentru aliniere; numărul protocolului *UDP* reprezentat pe **8 biţi**; lungimea pachetului *UDP* - **16 biţi**. Deoarece suma de control necesită în calculul ei un număr multiplu de 16 octeţi, la sfârşitul datelor se adaugă un număr potrivit de **octeţi de zero**. Dacă suma este 0 atunci ea va fi stocată ca 2^{16}=65.536 (toţi biţii pe 1). Un 0 în câmpul sumei de control indică faptul că suma de control nu a fost calculată.

4.4.2. *TCP* (*Transmission Control Protocol*) - reprezintă protocolul **orientat la conexiune** care oferă transmisia datelor în modul ***full-duplex*** [82].

Deoarece protocolul *IP* este de tip *datagramă,* utilizarea lui direct în aplicaţii care în general au nevoie de conexiuni sigure, este dificilă. Din aceste motive peste *IP* a fost construit un alt protocol, *TCP* (*Transimission Control Protocol*), care corectează aceste probleme.

Prin intermediul *TCP*, calculatoarele schimbă informaţii între ele sub **formă de segmente**. Un segment este format dintr-un **antet de 20 de octeţi urmat de zero** sau mai mulţi octeţi de date. Programul decide singur care este mărimea acestor segmente ţinând cont de două situaţii:

(1) Fiecare segment **nu trebuie să depăşească cei 65.536 octeţi** de informaţie utilă *IP*.

(2) Fiecare reţea are o unitate maximă de transfer (*MTU – Maximum Transfer Unit*) în care trebuie să încapă segmentul *TCP*.

Un segment care ajunge într-o reţea cu o ***MTU* mai mică decât dimensiunea sa**, va fi fragmentat de către ruterul respectivei reţele. Fiecare nou segment format prin fragmentare are propriile antete *TCP* şi *IP,* această situaţie conducând la creşterea gradului de încărcare al reţelei (fiecare segment primeşte un antet de **40 octeţi de informaţie suplimentară**).

a) Caracteristici *TCP*:

- Alături de protocolul *UDP*, *TCP* se situează **pe nivelul transport** în ierarhia de protocoale. Dar, cu toate că, se bazează pe acelaşi protocol (*IP*) ca şi *UDP*, *TCP* furnizează către nivelul **Aplicaţie** cu totul alt tip de servicii, **servicii orientate-conexiune**, sigure, de tip **flux de octeţi**.

– **Termenul de orientat-conexiune** presupune că, între cele două aplicaţii care comunică utilizând *TCP*, trebuie să se stabilească o conexiune *TCP* înainte ca transferul de date să aibă loc. Această **conexiune este virtuală**, asemănător cum se întâmplă în sistemul de

telefonie clasică: cineva formează un număr şi abia în momentul în care cealaltă persoană răspunde se poate începe conversaţia.

Fiind o **conexiune *host-la-host*, nu există noţiunile de *broadcast* sau *multicast***.

– **Transferul sigur de date** este asigurat în următorul mod: datele sunt împărţite în bucăţi a căror dimensiune optimă e determinată de *TCP*. Unitatea de date trimisă de *TCP* către nivelul reţea poartă **numele de segment**.

Când *TCP* trimite un segment, porneşte un *timer* şi dacă nu se primeşte confirmarea segmentului respectiv într-un anumit timp, îl **retransmite**. În momentul în care se primeşte un segment *TCP*, trimite o confirmare (din motive de eficienţă aceasta poate fi amânată).

– În *header*-ul *TCP* este menţinută o **sumă de control** pentru detectarea modificărilor în date. Dacă se recepţionează un segment corupt, *TCP* îl ignoră, urmând să fie retransmis datorită neprimirii confirmării.

– Deoarece segmentele *TCP* sunt transmise mai departe încapsulate în *datagrame IP*, iar acestea pot ajunge în orice ordine, segmentele *TCP* **pot ajunge în altă ordine decât cea în care au fost trimise**. De aceea, la destinaţie *TCP*-ul trebuie să se folosească de numere de secvenţă pentru a **reordona eventual segmentele** înainte de a le livra către nivelul Aplicaţie. De asemenea, *TCP* trebuie să asigure **ignorarea duplicatelor**.

– *TCP* asigură **controlul fluxului** în condiţiile în care **viteza de trimitere a datelor de la sursă poate fi mult mai mare decât capacitatea de prelucrare de la destinaţie**.

– Serviciul oferit de *TCP* este de tip **flux de octeţi,** deoarece oferă **garanţii** că fluxul de date trimis de sursă va fi livrat fără modificări la destinaţie.

b) Formatul pachetelor *TC:*

Segmentele sunt formate dintr-un **antet de 20 octeţi** urmat de datele primite de la nivelul Aplicaţie. Antetul poate uneori conţine şi o **serie de opţiuni**, caz în care poate ajunge la **60 octeţi**.

Antetul unui segment *TCP* cuprinde (vezi Tabelul 4.10):

Tabelul 4.10. Antetul unui segment *TCP*

<table>
<tr><td colspan="8">Lungimea antetului</td></tr>
<tr><td colspan="7">Port sursă (16 biţi)</td><td>Port destinaţie (16 biţi)</td></tr>
<tr><td colspan="8">Număr de secvenţă (32 biţi)</td></tr>
<tr><td colspan="8">Număr de confirmare (32 biţi)</td></tr>
<tr><td>N (4</td><td>Rezer-vat (6 biţi)</td><td>URG (1 bit)</td><td>ACK (1 bit)</td><td>PSH (1 bit)</td><td>RST (1 bit)</td><td>SYN (1 bit)</td><td>FIN (1 bit)</td><td>Dimensiunea ferestrei (16 biţi)</td></tr>
<tr><td colspan="7">Sumă control (16 biţi)</td><td>Indicator urgent (16 biţi)</td></tr>
<tr><td colspan="8">Opţiuni (cuvinte de 32 biţi) (până la 40 octeţi)</td></tr>
<tr><td colspan="8">Date (opţional)</td></tr>
</table>

- **Port sursă** - pentru specificarea portului aplicaţiei transmiţătoare. **Port destinaţie** - pentru specificarea portului aplicaţiei ce recepţionează segmentul *TCP*. Aceste două porturi, **împreună cu adresele ale sursei şi destinaţiei**, conţinute în antetul *IP* identifică în mod unic conexiunea. Pentru o pereche formată dintr-o adresă *IP* şi un port se foloseşte adesea denumirea de *socket*.

- **Număr de secvenţă** - este folosit pentru a **asigura la destinaţie ordonarea corectă a informaţiilor**. Pentru a asigura un serviciu de tip ***flux* de octeţi,** *TCP* numerotează fiecare octet de date utilizând un număr de secvenţă. Numărul de secvenţă din cadrul antetului *TCP* specifică primul octet din *fluxul* trimis.

La iniţierea unei conexiuni, se setează *flag*-ul *SYN* (sincronizare) şi se alege aleator un număr de secvenţă de început, utilizând un generator de numere de secvenţă, *ISN* (*Initial Sequence Number*). Numărul de secvenţă al primului octet de date va fi numărul generat *ISN* plus 1, deoarece pachetul de iniţiere (cu *SYN* setat) consumă şi el un număr de secvenţă. *Flag*-ul de terminare a conexiunii, *FIN* (Sfârşit), consumă şi el un număr de secvenţă.

- **Confirmare** - reprezintă numărul de secvenţă al octetului de

date pe care transmiţătorul confirmării aşteaptă să-l primească. Astfel, dacă s-a recepţionat octetul cu numărul de secvenţă *x* (în numerotaţia sursei) pachetul de confirmare va avea *flag*-ul *ACK* (confirmare) setat şi va conţine numărul de confirmare *x*+1.

- **Lungimea antetului** - indică numărul de cuvinte de 32 de biti (4 octeţi), care sunt conţinute în antetul *TCP*, deoarece câmpul „**Opţiuni"** este de lungime variabilă. Acest câmp este urmat de 6 biţi nefolosiţi (rezervaţi pentru dezvoltările ulterioare).

- 6 biţi **de control** reprezentând următoarele *flag*-uri ce pot fi setate simultan:

  - ***URG*** (Urgent) - câmpul urgent *pointer* este valid, indică deplasamentul în octeţi faţă de numărul curent de secvenţă la care se află informaţia urgentă şi se foloseşte la înlocuirea mesajelor de întrerupere.

  - ***ACK*** (Confirmare) - câmpul de confirmare este valid, indică validitatea numărului de confirmare. Dacă acest bit are valoarea zero, este ignorat câmpul „N**umăr de confirmare"** al segmentului respectiv.

  - ***PSH*** (*Push* - Forţare) – destinaţia: trebuie să trimită datele către nivelul **Aplicaţie** cât mai devreme, acest bit indică destinatarului să livreze datele.

  - ***RST*** (*Reset*) – desfiinţează o conexiune care a devenit inutilizabilă, se doreşte resetarea conexiunii.

  - ***SYN*** (Sincronizare) – stabileşte conexiunea între calculatoare, iniţierea conexiunii.

  - ***FIN*** (Sfârşit) – termină o conexiune.

- **Dimensiunea ferestrei** - reprezintă numărul de octeţi (începând cu numărul de secvenţă conţinut în câmpul de confirmare) pe care destinaţia e dispusă să-l primească. Este limitat la 65.536 octeţi dar, cu toate că pare o limită destul de mare, există situaţii când se doresc valori mult mai mari. În acest scop se folosesc nişte opţiuni speciale.

- **Suma de control** - este calculată pentru antetul şi informaţia

propriu zisă. Iniţial acest câmp are valoarea zero, iar câmpul de date este completat cu un octet suplimentar nul, dacă lungimea sa este un număr impar. Această sumă de control acoperă întregul segment *TCP* (atât datele cât şi antetul *TCP*) şi este obligatoriu să fie **completat de transmiţător şi verificat de receptor**. Dar la calculul sumei de control se ia in considerare şi o zona specială de 12 octeţi ce cuprinde pentru re-verificare câmpuri din antetul *IP* (adresa *IP* a sursei şi a destinaţiei). După adăugarea acestui pseudo-antet **datele se împart în cuvinte de 16 biţi** în vederea calculării sumei de control, adăugându-se eventual şi un octet de aliniere.

- **Urgent *pointer*** - este utilizat pentru specificarea deplasamentului ultimului octet de date trimis în regim urgent. Astfel, ultimul octet din secvenţa urgentă se obţine adunând acest câmp la numărul de secvenţă. Ce înseamnă acest mod urgent de transmitere: există situaţii în care aplicaţiile trimit date neordonate. Să presupunem că aplicaţia a trimis către nivelul transport o cantitate mare de date, dar la un moment dat se observă ceva în neregulă şi doreşte să anuleze operaţia. Dacă trimite o comandă de anulare, aceasta va fi adăugată la sfârşitul şirului de octeţi, deci va ajunge la nivelul Aplicaţie de la receptor. Activând **Modul urgent,** datele vor fi plasate la începutul pachetului. Aplicaţiile mai cunoscute care folosesc acest mod sunt *Telnet, Rlogin* şi *FTP*.

- **Opţiuni** - până la 40 de octeţi – este proiectat pentru a suporta facilităţi ulterioare; câmpul **„Opţiune”** este folosit mai ales pentru a indica **lungimea maximă** a unui segment *TCP*.

c) Iniţierea şi terminarea unei conexiuni *TCP:*

TCP este un protocol orientat la conexiune, deci presupune stabilirea unei căi virtuale între sursă şi destinaţie. Modul de transmisie în cazul protocolului *TCP* este *full-duplex,* deci sunt transmise date simultan în ambele direcţii. Aceasta presupune că cel care iniţiază conexiunea trebuie să primească aprobarea celuilalt capăt înainte de începerea transferului de date: sursa îşi anunţă

intenţia de a iniţia o conexiune, destinaţia trimite un pachet cu confirmarea cererii şi un pachet de iniţiere a conexiunii de la el la sursă, iar apoi sursa confirmă cererea primită de la destinaţie.

În cadrul **protocolului de iniţiere există 2 tipuri de cereri de iniţiere**: cerere activă (*active open*), iniţiată de clientul ce doreşte să stabilească conexiunea cu serverul şi cerere pasivă (*passive open*), din partea serverului.

Etapele procesului de stabilire a conexiunii sunt:

1. **Clientul trimite** un segment cu *flag*-ul *SYN* (sincronizare – stabileşte conexiunea între calculatoare, iniţierea conexiunii) setat, care conţine informaţii despre portul sursă, portul de destinaţie şi numărul de secvenţă generat iniţial (*ISN*). Opţional se pot stabili parametrii conexiunii prin setarea lungimii maxime a segmentelor (*MSS*) dispus să le primească de la server.

2. Al doilea segment e **trimis de server** şi are un rol dublu: (a) confirmă segmentul primit de la client prin setarea *flag*-ului *ACK* (câmpul de confirmare - indică validitatea numărului de confirmare: dacă acest bit are valoarea zero, este ignorat câmpul „număr de confirmare" al segmentului respectiv) şi (b) completează numărul de secvenţă cu numărul de secvenţă primit plus 1.

3. **Clientul trimite** un segment cu confirmarea cererii din partea serverului (*ACK* setat este completat cu numărul confirmat şi cu numărul de secvenţa primit plus 1). Acest pachet poate conţine date. Se poate întâmpla că uneori cele două capete care vor să comunice, încearcă să stabilească conexiunea simultan. În acest caz, după ce fiecare a trimis segmentul de iniţiere (iniţiere activă), ambele vor trimite un segment cu *SYN* plus *ACK* şi se va stabili o singură conexiune, nu două.

Protocolul de terminare. Oricare dintre cele două părţi poate solicita închiderea conexiunii, dar conexiunea fiind *full-duplex*, transferul de date în celălalt sens poate continua (această situaţie e denumită *half-close*). O închidere completă a unei conexiuni *TCP* presupune **următorul algoritm:**

1. **Clientul trimite** un segment cu *flag*-ul *FIN* (sfârşit – termină o conexiune) activat, solicitând închiderea conexiunii.

2. **Serverul trimite** un segment *ACK* confirmând primirea cererii. Numărul de confirmare se completează normal, ca numărul de secvenţă primit plus 1

3. **Serverul continuă să trimită date către client**. Când doreşte să închidă conexiunea, el trimite un segment cu *FIN* activat.

4. **Clientul trimite** un pachet *ACK*, confirmând închiderea conexiunii.

5. Cel care iniţiază procedura de închidere (trimite primul *FIN*), realizează o închidere activă (*active close*), iar la celălalt capăt are loc o închidere pasivă (*passive close*).

În mod similar cu deschiderea simultană există posibilitatea ca **ambele capete ale conexiunii *TCP* să iniţieze simultan procedura de închidere a conexiunii**. În acest caz ambele părţi vor trimite *FIN* şi vor aştepta primirea unui *ACK*. În continuare fiecare va primi cererea de terminare şi va trimite *ACK*. Se observă că numărul de segmente transferate pentru realizarea închiderii conexiunii (4 segmente) este acelaşi ca la terminarea normală.

Verifică-ţi cunoştinţele:

1) Analizaţi caracteristicile protocoalelor *UDP şi TCP*
2) Explicaţi formatul pachetelor *UDP şi TCP*
3) Descrieţi etapele procesului de stabilire a conexiunii *TCP*

Întrebările pentru autoevaluare:

1. Nivelul Reţea. Sisteme autonome. Clasificarea protocoalelor de rutare
2. Caracterizaţi protocolul *IP, TCP*. Clasele de adrese
3. Cum se utilizează masca de reţea? Subreţele (*host*-uri)
4. Protocoalele de nivel Reţea. Protocoalele *ARP* şi *RARP*
5. Nivelul Transport. Protocoalele la nivel Transport

Capitolul 5. Descrierea nivelelor: Sesiune, Prezentare, Aplicaţie

5.1. Nivelul Sesiune
5.2. Nivelul Prezentare
5.3. Nivelul Aplicaţie
    5.3.1. *Telnet*
    5.3.2. *File Transfer Protocol* (*FTP*)
    5.3.3. *World Wide Web*
    5.3.4. Poşta electronică

Menţionăm că nivelele utilizator ale modelului *OSI* se consider: nivelul 5 - Sesiune, nivelul 6 - Prezentare, nivelul 7 - Aplicaţie [11].

Nivelul Sesiune - furnizează controlul comunicaţiei între aplicaţii. Stabileşte, menţine, gestionează şi închide conexiuni (sesiuni) între aplicaţii.

Nivelul Prezentare - transformă datele în formate înţelese de fiecare aplicaţie şi de calculatoarele respective, asigură compresia datelor şi criptarea.

Nivelul Aplicaţie - realizează interfaţa cu utilizatorul şi interfaţa cu aplicaţiile, specifică interfaţa de lucru cu utilizatorul şi gestionează comunicaţia între aplicaţii. Acest nivel nu reprezintă o aplicaţie de sine stătătoare, ci doar interfaţa între aplicaţii şi componentele sistemului de calcul.

În continuare studiem particularităţile acestor nivele mai detaliat.

5.1. Nivelul Sesiune

Nivelul Sesiune este cel care coordonează aplicaţiile care interacţionează când două calculatoare comunică între ele.

Nivelul Sesiune stabileşte, gestionează şi încheie sesiunile de lucru între aplicaţii. Comunicarea între două calculatoare implică derularea unor mini-conversaţii pentru a se asigura că cele două calculatoare pot efectiv comunica. În timpul acestor mini-conversaţii fiecare din **participanţi joacă un rol dublu**: ca şi în cazul unui

client, pot să ceară la un moment dat un serviciu, dar ca şi în cazul unui **server** pot să ofere un serviciu. Procesul prin care se determină ce rol joacă la un moment dat unul din calculatoare se numeşte **controlul dialogului** [83]. De exemplu, nivelul Sesiune decide când are loc o comunicare în ambele sensuri simultan sau când are loc o comunicare în ambele sensuri alternativ (controlul dialogului).

Dacă se permite **o comunicare în ambele sensuri simultan**, nivelul Sesiune devine mai puţin activ în ceea ce priveşte gestionarea conversaţiei şi permite celorlalte niveluri ale celor două calculatoare să controleze întregul proces. În acest caz este posibil să apară coliziuni în cadrul acestui nivel.

Coliziunile de la nivelul Sesiune se manifestă doar sub forma a două mesaje transmise unul către celălalt şi care **creează confuzie fie la nivelul unui calculator, fie în ambele**. Dacă aceste coliziuni nu sunt tolerate, controlul dialogului apelează la o **comunicare în ambele sensuri alternativ**. În acest caz se foloseşte un **jeton specific** nivelului Sesiune, prin care cele două calculatoare stabilesc ordinea în comunicare (similar cu jetonul de la nivelul 2).

Protocoalele nivelului 5 pot fi identificate în timpul *login*-ului sau în cadrul unei aplicaţii: *NFS* (*Network File System*), *SQL* (*Structured Query Language*), *RPC* (*Remote Procedure Call*), *X-Window System*, *ASP* (*Apple Talk Session Protocol*), *DNA* (*Digital Network Architecture*), *SCP* (*Session Control Protocol*) (vezi Anexa 1).

5.2. Nivelul Prezentare

Nivelul Prezentare este cel care răspunde de prezentarea datelor într-o formă pe care calculatorul sursă să o poată „înţelege”.

Acest nivel acţionează ca un traducător pentru echipamentele care comunică într-o reţea şi îndeplineşte trei funcţii principale:

- prezentarea datelor;
- criptarea datelor;
- compresia datelor.

După ce primeşte datele de la nivelul Aplicaţie, dar înainte de a le transmite nivelului Sesiune, nivelul Prezentare execută una sau mai multe din funcţiile prezentate anterior. La destinaţie, nivelul Prezentare preia datele de la nivelul Sesiune, execută funcţiile necesare şi apoi transferă datele nivelului Aplicaţie.

Presupunem, că o staţie vrea să comunice cu un *maincadru*. Staţia foloseşte codurile *ASCII* pentru reprezentarea caracterelor, în timp ce *maincadru*-ul foloseşte codurile *EBCDIC*. Traducerea informaţiilor dintr-un cod în altul este realizată cu ajutorul nivelului 6.

Afară de reprezentarea caracterelor, standardele nivelului 6 vizează şi **modalităţile de prezentare a imaginilor grafice**:

- *PICT* – format pentru imagini, utilizat pentru transferul imaginilor grafice *QuicDraw* între programele sistemelor *MAC*;
- *TIFF* – format pentru imagini *bit-map* cu rezoluţie mare;
- *JPEG* – formatul *joint photographic experts group*;

Alte cerinţe se referă la **formatul de prezentare a sunetelor şi filmelor:**

- *MIDI* – pentru sunet digital (*Musical Instrument Digital Interface*);
- *MPEG* – standard pentru compresia şi codificarea filmelor video pe suport *CD* etc. (*Motion Picture Experts Group*)
- *QuickTime* – standardul pentru lucrul cu fişiere audio-video pe maşini *MAC* (diferenţă fată de *QuickTime for Windows*)

Folosind un soft specializat, la nivelul 6 se poate realiza şi **criptarea datelor**. Prin criptarea datelor se înţelege protejarea informaţiei în timpul transmiterii ei prin reţea. Majoritatea tranzacţiilor financiare ce se derulează prin Internet fac apel la criptare. De cele mai multe ori, o astfel de aplicaţie foloseşte o cheie de criptare pentru a codifica datele într-o nouă formă şi o cheie de decriptare pentru a le aduce în forma iniţială.

Tot nivelul Prezentare este cel care răspunde şi de **compresia fişierelor** - o tehnică prin care se reduce mărimea lor folosind algoritmi destul de complex.

5.3. Nivelul Aplicaţie

Nivelul Aplicaţie este cel mai apropiat de utilizatorul calculatorului.

Nivelul Aplicaţie este responsabil cu **identificarea partenerilor** disponibili să comunice, sincronizează aplicaţiile, stabileşte proceduri pentru recuperarea datelor şi controlează integritatea acestora.

Aplicaţiile Internetului sunt numeroase: în primul rând afişarea de informaţii mai mult sau mai puţin statice cu formă de text, imagini şi sunete (pagini *web*), poşta electronică (*e-mail)*, transferul de fişiere de date şi informaţii, *chat*, video, telefonie şi telefonie cu imagine prin Internet, televiziune prin Internet, *e-commerce*, învăţământul la distanţă (*e-learning*), transmisiea vocii prin Internet (*VoIP –Voice over Internet Protocol*), conversaţii în timp real (*IRC - Internet Relay Chat*), transmisii multimedia în timp real, sondări de opinie, mediu pentru răspândirea ştirilor, mediu pentru toate genurile de grafică şi muzică, deschiderea unei sesiuni de lucru de la distanţă, grupuri de discuţii pe teme prestabilite, jocuri interactive prin reţea, operaţii bancare (*Internet banking*) şi multe altele [53]. Printre ele, *World Wide Web*, prescurtat *WWW*[30], este la loc de vârf, deoarece este o aplicaţie multimedială şi integrativă, cu o interfaţă de utilizator (*Graphic User Interface - GUI*) foarte atrăgătoare din punct de vedere grafic, practic şi simplu de folosit.

Toate aceste servicii se bazează pe diverse aplicaţii Internet, dezvoltate în ultimii ani, precum pagini şi *site*-uri *web* editate cu

[30] *WWW* a fost inventat de către *Tim Berners-Lee* în anul 1993.

diverse limbaje (*HTML – HyperText Markup Language*, *XML – Extendable Markup Language*, *PHP – Personal Home Page* sau *Hypertext PreProcessor*), baze de date care pot fi accesate numai pentru preluare de informaţii şi/sau pentru înscriere de date prin intermediul formularelor electronice etc.

Numărul furnizorilor de servicii Internet (*ISP – Internet Service Provider*) este în creştere. De asemenea, vitezele oferite pentru trafic sunt mai mari, întârzierile de transmisie şi pierderile de pachete mai mici datorită dezvoltării echipamentelor de comunicaţie pentru reţelele de calculatoare (modem – *Modulator DEModulator*; *hub, switch, bridge, route*r), a diversificării mediilor fizice de transmisie (cablu torsadat, cablu coaxial, fibră optică, în eter „fără fir" sau „*wireless*"), dar şi a tehnologiilor Internet: (*Ethernet, FastEthernet, GigaEthernet, FDDI – Fiber Distributed Data Interface, WLAN – Wireless LAN, FR – Frame Relay, ATM – Asynchronous Transfer Mode, ISDN – Integrated Services Digital Network, ADSL – Asymmetric Digital Subscriber Line* etc.).

Pentru folosirea tuturor aplicaţiilor este nevoie în general doar de un singur program multifuncţional numit ***browser*.** Exemple: *MS Internet Explorer, Mozilla Firefox* (provenit din *Netscape Navigator*), *Google Chrome, Opera, Apple Safari* etc. De asemenea, accesul la Internet poate fi asigurat şi din afara unei reţele proprii-zise de calculatoare, din alte reţele de comunicaţii cum sunt cele de telefonie mobilă. Transportul pachetelor Internet poate fi realizat nu numai de reţelele de calculatoare dedicate acestui scop ci şi de reţele de comunicaţii cu alt profil, precum cele de televiziune prin cablu.

În 1999, a apărut conceptul de INTERNET 2, administrat de *UCAID* (*University Corporation for Advanced Internet Development*), ca parteneriat între universităţi, corporaţii şi agenţii guvernamentale din întreaga lume, având ca scop dezvoltarea de noi aplicaţii Internet şi a infrastructurii în care se vor utiliza acestea.

Pentru dezvoltarea Internetului, se au în vedere noi standarde şi protocoale pentru reţelele de comunicaţii, asigurarea securităţii comunicaţiilor prin operaţii de autentificare a mesajelor, criptare a datelor şi folosirea semnăturilor digitale, implementarea de noi servicii la cererea clienţilor în special pentru dezvoltarea aplicaţiilor de tip „realitate virtuală", cum sunt jocurile interactive, magazinele „virtuale" pentru cumpărături *on-line* sau spitalele „virtuale" cu accesare de la distanţă, în care pot colabora doctori din diferite ţări.

Companiile multinaţionale îşi creează reţele de calculatoare private, securizate (*intranet*) pentru comunicaţii între diverse locaţii de pe glob, securizate faţă de utilizatorii din afara reţelei.

Analizăm unele protocoale care oferă facilităţi la nivelul Aplicaţie:

5.3.1. *Telnet* (*Terminal Emulation Protocol*)

TELNET (*Virtual Terminal Connection Protocol*) este un protocol de terminal virtual care permite **conectarea unui utilizator de la distanţă la anumite calculatoare-gazd**ă, rulând programul *telnet* al serverului. Se utilizează algoritmi de negociere cu terminalul respectiv, pentru a-i cunoaşte caracteristicile. Acesta este văzut ca un **terminal virtual** cu care se poate comunica de la distanţă, indiferent de caracteristicile lui fizice.

Protocolul *Telnet* transmite apăsările de taste (*keystrokes*) către *remote host* şi afişează rezultatul acestor *keystrokes* pe terminalul local; permite utilizatorilor să se logheze pe computerele aflate în altă locaţie şi să acceseze resursele acestora de pe computerul local; emulează un terminal pentru a fi folosit peste o conexiune *TCP* [84].

5.3.2. *File Transfer Protocol* (*FTP* - Protocol pentru transferul fişierelor) - este protocolul care oferă facilităţi pentru **transferul fişierelor** pe/de pe un calculator din reţea. De multe ori pentru această acţiune utilizatorul este nevoit să se autentifice pe calculatorul de pe care doreşte să încarce/descarce fişiere.

Facilitatea cunoscută sub numele de *anonymous FTP* lucrează cu un **cont public implementat pe calculatorul gazdă, numit *guest*** [85].

Când se iniţiază un transfer prin *FTP* trebuie precizate următoarele aspecte:

1) **Tipul fişierului** - se specifică maniera în care datele conţinute de un fişier vor fi aduse într-un format transportabil prin reţea:

- fişiere *ASCII* (*American Standard Code for Information Interchange*) şi *EBCDIC* (*Extended Binary Coded Decimal Interchange Code*) – calculatorul care transmite fişierul îl converteşte **din formatul local text în format *ASCII***;
- fişiere binare (*binary*) – fişierul este transmis exact cum este memorat pe calculatorul sursă şi **memorat la fel** pe calculatorul destinaţie;
- fişiere locale – folosite în mediile în care cel care transmite **precizează numărul de biţi/*byte***.

2) **Controlul formatului** – se referă la fişierele text care sunt transferate direct către o imprimantă:

- *No printing controls (default);*
- *Telnet printing controls;*
- *Fortran printing controls.*

3) **Structura** – fişierele pot să-şi păstreze structura internă în timpul transmisiei. Există trei posibilităţi:

- Structura fişierului – fişierul este văzut ca un flux continuu de *bytes*, fără o structură internă;
- Structura înregistrării – fişierul reprezintă o serie de înregistrări (valabil în cazul *fişierelor-text*);
- Structura paginii (structură-bloc) – fiecare pagină este numerotată pentru a putea fi transmisă în orice ordine.

4) **Modul de transmitere**. Sunt trei posibilităţi:

- *Stream* – fişierul este transferat într-o serie de *bytes*;
- *Bloc* – fişierul este transferat bloc cu bloc, fiecare cu un *header*;
- Compresat – se foloseşte o schemă de comprimare a secvenţelor de *bytes* identici.

În timpul unui transfer prin *FTP* nu există **nici un mecanism de negociere a transmisiei.**

5.3.3. *World Wide Web*. Conceptul care a stat la baza WWW este conceptul de *hypertext*.[31]

HTTP este acronimul pentru *HyperText Transfer Protocol* sau **protocolul ce stabileşte regulile de transfer** a documentelor *hypermedia*. Aplicatiile care folosesc acest protocol sunt considerate entităţi abstracte din punctul de vedere al protocolului. Ele trebuie să poată formula cereri şi/sau recepţiona răspunsuri (modelul *client-server*). Pentru referirea unei resurse în Internet, se foloseşte termenul generic *URI - Uniform Resource Identifier*. Dacă se face referire la o locaţie, spunem că avem de a face cu un *URL -Universal Resource Locator*. Dacă se face referire la un nume, avem de-a face cu un *URN - Universal Resource Name* [86, 87, 88].

Protocolul *HTTP* se bazează pe paradigma cerere/răspuns. Clientul cere accesul la o resursă, aceasta fiind identificată prin *URI*, iar serverul răspunde printr-o linie de stare ce conţine un cod de succes sau eroare şi urmează datele cerute.

Cel mai simplu caz este acela când conexiunea *client-server* se realizează prin intermediul unei singure conexiuni. În general, există mai mulţi intermediari de-a lungul conexiunii:

- ***proxy* serverul** - primeşte cereri adresate unei resurse identificate prin *URI*, rescrie anumite părţi ale mesajului, după care retrimite cererea către calculatorul adresat iniţial. El se substituie,

[31] Prin *hypertext* se înţelege o colecţie de documente unite între ele prin legături (*link*) ce permit parcurgerea acestora bidirectional.

practic, clientului iniţial, mesajul de răspuns fiind primit tot de el;

- ***gateway*** - este similar unui *proxy,* dar pe partea de server. Este un fel de cameră de primire pusă în faţa unui server sau a unui grup de servere. Serverele de „după *gateway*" nu sunt vizibile, ele fiind reprezentate de *gateway*. Cererile sosite la *gateway* sunt dirijate spre serverul care poate răspunde cererii, sau celui mai liber dintre serverele ce pot răspunde. *Gateway* realizează şi o conversie de protocol, serverul nefiind obligat să „cunoască" protocolul *HTTP;*

- ***tunnel*** - transportă date pe care nu le „înţelege". De obicei, la un capăt al tunelului se află un *server gateway*, iar la capătul celălalt - un *proxy*.

Adresarea unei resurse în Internet se face prin construcţii de forma: *protocol://[servciu].nume_dns[.nume_local/cale/subcale/ nume_document]*.

Serverul care răspunde cererilor privitoare la documente *hypermedia* se numeşte server *WWW*, acest server „cunoaşte" protocolul *HTTP* şi oferă serviciul *WWW*.

5.3.4. Poşta electronică – este astăzi una din patru aplicaţii principale ale Internetului: poşta electronică, ştiri, conectarea la distanţă, transferul de fişiere [89, 90].

Pentru **a putea transmite un mesaj** prin intermediul poştei electronice este nevoie de câteva ingrediente: un calculator, o conexiune la reţea (*modem,* de exemplu), un program care permite utilizarea acestui serviciu de Internet, o conexiune la Internet (oferită de un *provider* sau de un serviciu *on-line*) şi o adresă de *e-mail*.

Mesajul pe care îl transmiteţi este preluat în reţeaua Internet de către un server şi apoi livrat calculatorului menţionat în adresa de *e-mail*. Adresa de poştă electronică este o adresă Internet formată din două părţi, despărţite de caracterul *@*:

- prima parte a adresei reprezintă **numele de conectare a persoanei** căreia îi este destinat mesajul (*ID_user*);

- a doua parte reprezintă **denumirea domeniului** din care face

parte persoana (identifică nodul destinaţie - adresa_nod)

Dacă este instalat un *browser* ca *Microsoft Internet Explorer, Pine* (pentru *Unix), EudoraPro, America Online (AOL), HotCast, Calypso, Messenger, Mozilla Firefox*, etc., sunt instalate şi aplicaţiile necesare pentru *e-mail*.

Pentru a primi sau a trimite un mesaj, calculatorul trebuie însă să comunice cu un server de *e-mail* folosind un anumit protocol de livrare. **Acest protocol se stabileşte, de obicei, în momentul configurării softului de *e-mail*:**

- *POP* (*Post Office Protocol*) - este un protocol simplu utilizat pentru **aducerea mesajelor dintr-o cutie poştală aflată la distanţă** şi de a le depozita pe calculatorul local al utilizatorului. Este cel mai vechi protocol (1984), ajungându-se în prezent la *POP3*;

- *IMAP* (*Interactive Mail Access Protocol*) - este un protocol care a fost proiectat pentru a ajuta utilizatorilor care **folosesc mai multe calculatoare** (un calculator la birou, un calculator acasă sau un *notebook*). În acest caz, *server*-ul de *e-mail* păstrează un depozit central de mesaje, la care accesul poate fi realizat de pe orice calculator. În comparaţie cu protocolul *POP, IMAP* nu copiază poşta electronică pe calculatorul personal al utilizatorului;

- *DMSP* (*Distributed Mail System Protocol)* - este un protocol care permite utilizatorilor să **aducă poşta electronică de pe serverul** de *e-mail* pe un calculator şi după aceasta să se **deconecteze de la server**.

Când se alege un client de *e-mail,* trebuie să avem în vedere următoarele: ce standarde suportă - *IMAP, POP* etc.; capabilitatea de lucru cu conturi de *e-mail* multiple; posibilitatea de a aduce de pe server doar mesajele dorite, celelalte fiind eliminate prin filtre; posibilitatea de arhivare a *mail*-urilor, precum şi importul şi exportul textelor; ergonomia (interfaţa cu utilizatorul, modul de explicitare a erorilor intervenite, documentaţia); funcţionalitatea (în ce măsură

clientul de *e-mail* îndeplineşte şi atinge cerinţele utilizatorului, prin opţiunile puse la dispoziţie); resurse necesare sistemului pentru fiecare aplicaţie în parte pentru a rula optim şi fără întreruperi; suportul formatului *HTML*.

Verifică-ţi cunoştinţele:

1) Enumeraţi nivelele de utilizator.
2) Caracterizaţi funcţia nivelului Sesiune.
3) Cum acţionează nivelul Prezentare?
4) Numiţi nivelul cel mai apropiat de utilizatorul calculatorului.
5) Analizaţi protocoalele care oferă facilităţi la nivelul Aplicaţie.
6) Poşta electronică

Întrebările pentru autoevaluare:

1. Descrieţi succint Nivelul Sesiune, Nivelul Prezentare, Nivelul Aplicaţie
2. Explicaţi noţiunea de *FTP*
3. Lămuriţi modul de utilizare a poştei electronice

Capitolul 6. Dispozitivele retelelor de calculatoare. Modul de interconectare

6.1. Repetoare. *Hub*-uri
6.2. Punţile (poduri, *bridge*-uri)
  6.2.1. Principiile de funcţionare a punţilor
  6.2.2. Rolul punţii în comunicaţia din interiorul aceluiaşi segment
  6.2.3. Rolul punţii în comunicaţia dintre segmente
  6.2.4. Cum îşi construieşte puntea tabela de comutare
6.3. Comutator (*Switch*)
  6.3.1. Tipurile de comutare folosite de un comutator
  6.3.2. Rolul comutatoarelor în implementarea conexiunilor *Ethernet half-duplex*
  6.3.3. Rolul comutatoarelor în implementarea conexiunilor *Ethernet full-duplex*
6.4. Ruterele.
  6.4.1. Tabele de rutare
  6.4.2. Clasificări ale rutelor
  6.4.3. Efectul ruterelor asupra domeniilor de difuzare şi a domeniilor de coliziune
  6.4.4. Tipurile ruterelor
6.5. Protocolul *STP* (*Spanning Tree Protocol*)
  6.5.1. Prevenirea apariţiei avalanşelor de difuzări
  6.5.2. Modul de funcţionare a *STP*
6.6. Placa de reţea (adaptor *LAN*)
6.7. Modemul
6.8. Intranetul

Pentru situaţiile în care se exploatează mai multe reţele locale sunt necesare atât echipamente ce facilitează realizarea conexiunii fizice, cât şi un *soft* de interconectare.

Dispozitivele de interconectare pot fi clasificate în trei categorii: **repetoare, punţi şi rutere**. Această tipologie prezintă avantajul de a urmări de aproape **stiva de protocoale OSI**, prezentând astfel acţiunile specifice fiecărui nivel.

Pentru interconectarea dispozitivelor se folosesc: **cabluri coaxiale, cabluri torsadate, fibre optice şi unde radio** [91]. La ora actuală în locul unui cablu fizic pot fi utilizate legături radio. Acestea pot fi folosite pentru interconectarea segmentelor de cablu ale reţelelor locale sau pentru conectarea sistemelor individuale la *LAN*. Este permisă astfel deplasarea sistemelor de calcul şi a altor echipamente ale reţelei dintr-un loc în altul fără a fi nevoie de modificarea unui cablaj fizic.

6.1. Repetoare. *Hub*-uri

Repetoarele se folosesc pentru prelungirea liniilor de date [92].

Repetorul este dispozitivul de interconectare ce funcţionează **la nivel Fizic**. Repetoarele conectează segmentele media ale reţelei şi asigură amplificarea şi retransmiterea semnalelor digitale, fără a putea interveni asupra conţinutului de informaţii (filtrare, corectare).

Deoarece la nivelul Fizic nu există date ci doar biţi, repetorul nu este preocupat de identificarea destinaţiei sau de verificarea unui cod de corecţie, ci doar de semnalul electric pe care-l primeşte şi de regenerarea acestuia. **Principala sa funcţie** este aceea de a extinde suprafaţa acoperită de o reţea locală cu un cost şi o latenţă foarte scăzute. Aceste echipamente permit **amplificarea semnalelor** prin retransmiterea acestora pe mai multe segmente de cablu care alcătuiesc o structură de tip arbore. În felul acesta **se măreşte distanţa fizică** pe care poate acţiona o reţea locală. Totodată repetorul poate fi utilizat pentru a face **legătura între medii de transmisiune** diferite (cablu coaxial - fibră optică, cablu coaxial - cablu torsadat). Există repetoare pentru toate mediile de transmisie pe cupru - de la cablul coaxial de diferite impedanţe până la cel torsadat.[32]

[32] La reţelele de topologia „inel" repetoarele nu sunt folosite, în aceste reţele fiecare sistem acţionând ca un repetor.

Principiul de funcţionare a repetorului constă în următoare: şirul de biţi generat iniţial de o placă de reţea respectă strict nivelurile de tensiune standardizate. Cu cât şirul de biţi călătoreşte mai mult prin cablu, semnalul electric se deteriorează şi devine din ce în ce mai slab. Repetoarele nu interpretează cadrele pe care le recepţionează, ci doar le **repetă bit cu bit** pe celelalte segmente. Pentru a opri deteriorarea semnalului peste o limită ce l-ar face de nerecunoscut pentru destinaţie, repetorul ia şirul de biţi, îl aduce **la treptele de semnalizare standardizate şi îl amplifică**.

Deprecierea semnalului nu apare doar când acesta călătoreşte prin mediul de cupru, dar şi când ataşăm prea multe dispozitive la mediul de transmisie, deoarece fiecare nou dispozitiv ataşat la mediu va provoca o mică degradare a semnalului.

Una din componentele esenţiale ale protocolului *Ethernet* este **detecţia coliziunilor**. Un domeniu de coliziune reprezintă acea secţiune dintr-o reţea în care se va propaga o coliziune, iar un domeniu de difuzare (domeniu de *broadcast*) reprezintă acea secţiune dintr-o reţea în care se va propaga un pachet de difuzare. Deoarece pentru un repetor nu există noţiunea de coliziune şi de pachet de date, repetoarele extind atât domeniile de coliziune, cât şi pe cele de difuzare.

Repetoarele împart reţeaua în microsegmente. Există o regulă foarte importantă pentru proiectarea reţelelor *Ethernet*: regula 5-4-3.

Regula 5-4-3: Comunicaţia dintre oricare două calculatoare sau dispozitive dintr-o reţea nu trebuie să treacă prin mai mult de: **5 microsegmente; 4 repetoare consecutive; 3 microsegmente populate (la care pot fi conectate staţiile)** [93].

Analizăm funcţionarea acestei reguli.

Există **o fereastră de timp pentru transmiterea unui bit**. Pentru *Ethernet*, ce oferă o viteză de *10 Mbps*, durata transmiterii unui singur bit este de **100 de nanosecunde**. Dimensiunea minimă a

cadrului *Ethernet* este de **64 de octeţi=512 biţi**. Rezultă că timpul necesar **transmiterii cadrului** de dimensiune minimă este de **51,2 microsecunde.**

Ne interesează acest timp din cauza că **apariţia unei coliziuni trebuie detectată înainte de expirarea acestui interval de timp**. În caz contrar, apariţia unei coliziuni va fi interpretată ca o coliziune la cel de-al doilea cadru şi nu pentru primul.

Latenţa introdusă de mediul de transmisie va fi dată de viteza de propagare a semnalului electric, aceasta fiind aproximativ **două treimi din viteza luminii**[33]. Rezultă că propagarea pe un **segment de 100 de metri** va dura aproximativ **0,5 microsecunde**. Comparativ cu **latenţa introdusă de un repetor *Ethernet* de aproximativ 5,6 microsecunde**, latenţa introdusă de mediul de conectare poate fi neglijabilă.

Cel mai defavorabil caz se obţine când sursa şi destinaţia se află la distanţa maximă, iar **coliziunea apare lângă destinaţie**, astfel încât coliziunea ce trebuie detectată şi de sursă, trebuie să parcurgă de două ori distanţa maximă. Dacă vom considera acum că între sursă şi destinaţie se află **cinci repetoare**, vom determina că în cel mai defavorabil caz detecţia coliziunii va fi posibilă doar după cel puţin **(5,6*5)*2=56 de microsecunde**, asta însemnând că un alt doilea pachet deja a fost trimis.

În cazul **nerespectării regulii 5-4-3**, în primul rând, se va cere retransmisia unui cadru corect, în vreme ce cel pierdut în urma coliziunii va fi considerat ca ajuns la destinaţie intact. Astfel responsabilitatea integrităţii datelor va fi pasată nivelului superior şi anume **nivelului Reţea**. Deoarece acest nivel **nu are posibilitatea manipulării de cadre**, şi va determina că întregul **pachet din care face parte şi cadrul eronat este incorect,** cerând **retransmiterea pachetului**. Această practică, deşi va asigura integritatea datelor, introduce o **latenţă semnificativă**.

[33] 299 792 458 м/s

Frecvent sunt utilizate şi **repetoare *multipunct***, numite ***Hub*-uri** (*Host Unit Broadcast*). Practic aceste dispozitive au rolul de a uni liniile de comunicaţie **într-o locaţie centrală, oferind o conexiune comună tuturor dispozitivelor din reţea**.

Deoarece toate *host*-urile împart **aceeaşi lărgime de bandă şi acelaşi domeniu de coliziune,** *hub*-urile vor **transmite datele** primite pe unul dintre porturi **pe toate celelalte porturi**.

Iniţial au existat două tipuri de *hub*-uri: pasive şi active.

***Hub*-urile pasive** oferă posibilitatea interconectării la acelaşi mediu de transmisie a mai multor dispozitive **fără a regenera semnalul** la trecerea prin ele.

***Hub*-urile active** vor oferi în plus faţă de primele - **regenerarea semnalului**. Datorită scăderii extrem de rapide a preţurilor şi avantajelor ce le oferă această regenerare de semnal *hub*-urile pasive au dispărut de pe piaţă încă de la sfârşitul anilor '80 [11].

Verifică-ţi cunoştinţele:

1) Unde se folosesc repetoarele şi *hub*-urile?
2) Explicaţi necesitatea respectării Regulii 5-4-3.

6.2. Punţile (poduri, *bridge*-uri)

Dispozitivele folosite în **reţele locale la nivelul Legătură de date** sunt: punţi (*bridge*-uri) şi comutatoare (*switch*-uri).

Punţile interconectează două sau mai multe segmente de reţea; sunt folosite la interconectarea a grupurilor de calculatoare ce diferă prin protocolul folosit (de exemplu, *Ethernet* şi *Token Ring*) la nivelul Legătură de date sau a mediului de transmisie.

Puntea este primul dispozitiv de interconectare ce poate lua **decizii logice**. Există două mecanisme ce fac din punte **un dispozitiv de interconectare „inteligent”**: (1) **încapsularea datelor** la nivel Legătură de date şi (2) folosirea unei **scheme de adresare** pentru livrarea acestora.

Punţile filtrează traficul de reţea bazat pe adresa de control al mediului fizic (*MAC address*), elimină erorile din reţea. Adresele dispozitivelor conectate în ambele părţi ale punţii sunt memorate **în tabela de comutare**.

O punte ce primeşte cadre de date le retransmite reţelelor interconectate pe baza unor **algoritmi de expediere** (*forwarding*), selectaţi de producător (dirijare explicită, filtrare de adrese etc.).

Spre deosebire de repetor, o punte este capabilă **să decodeze cadrul** pe care-l primeşte pentru a face prelucrările necesare **transmiterii pe reţeaua vecină**.

Pentru transmiterea unui cadru, puntea trebuie să aştepte disponibilitatea reţelei. Aceasta înseamnă că **mesajele recepţionate sunt temporar memorate** de către punte şi apoi emise către sistemul destinatar. Deoarece reface electric semnalele, la apariţia unor **coliziuni sau zgomote, puntea nu le propagă mai departe în reţea**. În plus, puntea permite administratorului de reţea divizarea acesteia **în segmente logice mai mici**, pentru a fi administrată mai uşor.

6.2.1. Principiile de funcţionare a punţilor. Faţă de un simplu calculator, care la nivelul Legătură de date se preocupă doar de încapsularea datelor în cadre, o punte trebuie să ia **decizia spre ce segment să trimită cadrul primit**. În cazul în care pe una dintre interfeţe puntea primeşte un şir de biţi ale căror valori nu sunt 0,85V sau -0,85V (în cazul *Ethernetului*), va încerca să-şi dea seama care au fost valorile iniţiale a acestor biţi pentru a putea înţelege cadrul primit.

Odată obţinut un cadru valid, adică după corectarea biţilor ce nu mai aveau niveluri de tensiune corectă, puntea va desface antetul cadrului şi va **analiza informaţiile legate de adresa destinaţie**. După determinarea interfeţei pe care trebuie trimis cadrul, **placa de reţea îl va transforma în biţi**, trecându-l la nivelul fizic. Placa de reţea poate genera doar câteva niveluri de tensiune, astfel încât nici nu ar fi posibilă trimiterea şirului de biţi depreciat.

Principala funcţie a unei punţi este filtrarea traficului pe baza adresei fizice.

Pentru a putea lua astfel de decizii punţile folosesc o tabelă, **numită tabelă de comutare** (*bridging/switching table*). **Tabela de comutare** (Tabelul 6.1) este o **listă de reguli**, fiecare cuprinzând o parte de identificare (*matching*) şi una de acţiune. În **partea de identificare se afla o adresă *MAC* destinaţie**, iar pentru partea de acţiune era precizată una din interfeţele. În tabela de comutare fiecărei adrese fizice îi este asociată una dintre interfeţele sale. În tabelul 6.1 se reprezintă o astfel de **tabelă de comutare cu 3 întrări**.

Tabelul 6.1. Tabela de comutare

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

De exemplu, prima întrare are următoarea semnificaţie: destinaţia 00.48.C2.01.78.12 se află pe segmentul conectat pe interfaţa E0 a punţii (E0 este prescurtarea de la *Ethernet 0*, prima interfaţă *Ethernet*).

6.2.2. Rolul punţii în comunicaţia din interiorul aceluiaşi segment. Protocolul *Ethernet* oferă un mediu de comunicaţie distribuit, adică comunicaţia dintre două staţii va fi accesibilă nivelului Legătură de date a oricărei alte staţii conectate pe acelaşi segment. Pentru fiecare cadru primit de o staţie, nivelul Legătură de date va verifica **dacă această staţie este sau nu destinaţia**. În cazul afirmativ, cadrul va fi pasat nivelului Reţea, în caz contrar - va fi ignorat.

Pentru cazul comunicaţiei în interiorul aceluiaşi segment (de exemplu, a reţelei *Ethernet*) considerăm reţeaua din figura 6.1.

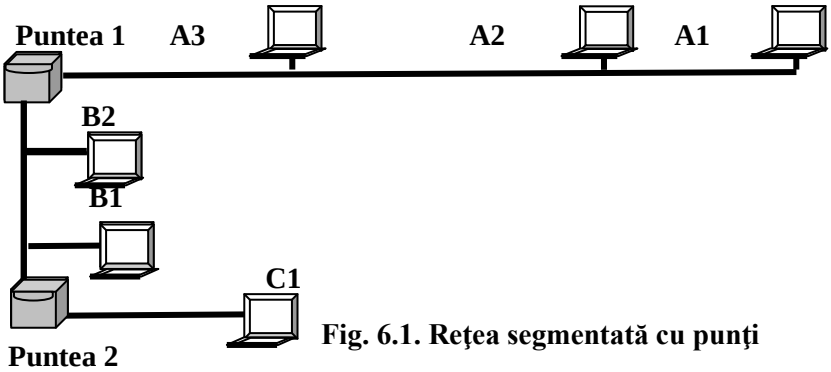


Fig. 6.1. Reţea segmentată cu punţi

Presupunem că staţia A1 vrea să transmită date staţiei A2. Primul lucru pe care-l va face staţia A1 va fi **ascultarea mediului.** Dacă mediul este liber, va începe transmisia datelor. Cadrul emis de A1 se va propaga către toate staţiile conectate pe acest segment, inclusiv către punte. Staţia A2 va trece cadrul către nivelul reţea, staţia A3 îl va ignora.

Odată ajuns la punte **cadrul este despachetat** şi **adresa destinaţie este căutată în tabela de comutare** a punţii. Puntea va stabili că destinaţia se află chiar pe interfaţa pe care a primit cadrul. În acest caz puntea ia decizia că acest cadru nu mai trebuie transmis, deoarece retransmiterea cadrului ar duce la o duplicare a acestuia la destinaţie.

Cum va acţiona puntea 1 în cazul **comunicaţiei între B1 şi B2**? Ambele punţi (deşi vor recepţiona cadrele) vor lua decizia de a nu le mai retransmite. Să presupunem, că cele două comunicaţii apar simultan: atât A1 transmite către A2, cât şi B1 către B2. Va apărea în acest caz o coliziune? Dacă în loc de puntea 1 ar fi folosit un repetor, cu siguranţă ar fi avut o coliziune. În cazul nostru, nici un cadru din comunicaţia dintre A1 şi A2 nu va ajunge pe segmentul B, şi nici un cadru din comunicaţia dintre B1 şi B2 nu va ajunge pe segmentul A, este **imposibil să apară o coliziune.**

Puntea izolează comunicaţia între staţii aflate în acelaşi segment la nivelul segmentului.

Consecinţele acestui fapt sunt extrem de importante. În primul rând, **puntea va mărgini domeniile de coliziune**. Totodată ea va oferi **mai multă bandă disponibilă**, deoarece comunicaţia în interiorul aceluiaşi segment nu va consuma din banda disponibilă a întregii reţele. O altă consecinţă o reprezintă **minimizarea riscurilor de securitate** legate de atacurile din interiorul reţelei locale[34]. Prin folosirea punţilor putem izola de restul reţelei staţiile ce prezintă un risc de securitate.

6.2.3. Rolul punţii în comunicaţia dintre segmente. Pentru acest caz vom considera aceeaşi reţea din figura 6.1. şi un trafic între **staţia A1 şi B1**. Staţia A1 va asculta mediul şi când acesta va fi liber va transmite un cadru. Cadrul se va propaga spre staţiile A2, A3 şi spre puntea 1. Staţiile vor ignora cadrul, acesta nefiind adresat lor, în schimb puntea va căuta adresa destinaţie în tabela sa de comutare. Va determina interfaţa pe care trebuie trimis cadrul şi apoi va decide că această **interfaţă este diferită de cea pe care cadrul a fost primit**. Astfel puntea va transmite cadrul primit din segmentul A pe segmentul B. Cadrul va fi recepţionat atât de B1, cât şi de B2, dar doar B1 îl va prelucra.

Faţă de avantajele prezentate mai sus, puntea aduce şi o serie de **dezavantaje**: **costul** unei punţi este cu cel puţin un ordin de mărime mai mare decât cel al unui repetor. Înlocuirea repetoarelor cu punţi duce o **creştere a latenţei** în reţea cu 10-30%, datorită timpului necesar prelucrării informaţiei de nivel Legătură de date. În cazul unui trafic intens între staţii aflate în segmente diferite puntea **poate duce la o „gâtuire" a traficului**.

[34] Unul dintre cele mai populare atacuri este ascultarea liniei (*sniffing attack*), prin care pe una dintre staţiile conectate la mediul distribuit se forţează nivelul Legătură de date să trimită spre nivelurile superioare toate cadrele, inclusiv cele ce nu sunt destinate acestei staţii.

Pentru a putea funcţiona eficient o punte trebuie să aibă la dispoziţie o tabelă de comutare ce conţine câte **o intrare pentru fiecare dintre staţiile** din acea reţea locală. Căutarea în această tabelă este o **căutare secvenţială**, deci extrem de ineficientă pentru o dimensiune prea mare a tabelei. Astfel dimensionarea optimă a reţelei ce foloseşte doar punţi, deşi nu se supune nici unei restricţii de lungime, va fi puternic **influenţată de numărul de staţii**, precum şi de latenţa admisibilă de tipul traficului.

6.2.4. Cum îşi construieşte puntea tabela de comutare. În exemplele anterioare se presupune că tabela de comutare era deja construită. Această **tabelă este păstrată în memoria *RAM* a punţii**, prin urmare se va pierde, dacă reiniţializăm puntea. În plus, o punte **trebuie să fie în stare să includă dinamic în tabela de comutare informaţii** despre o nouă staţie conectată în reţea.

Să considerăm reţeaua din figura 6.2, unde puntea 1 a fost reiniţializată, şi staţia A1 vrea să comunice cu staţia B1.

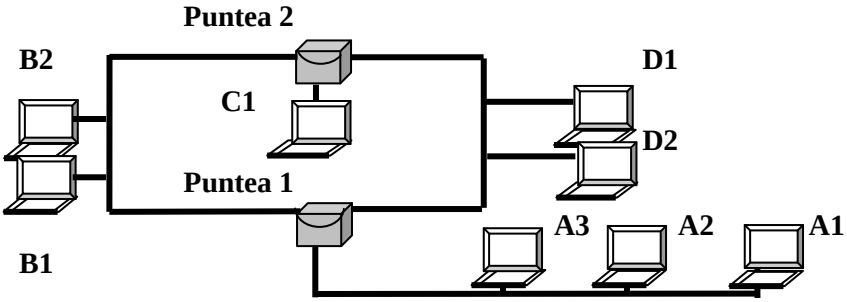


Fig. 6.2. Construirea tabelei pentru Puntea 1

Staţia A1 ascultă mediul, iar când acesta este liber, trimite un cadru ce are ca destinaţie staţia B1. Staţiile A2 şi A3 vor ignora cadrul. Puntea 1 va primi cadrul şi va încerca să găsească adresa destinaţie în tabela sa de comutare. Puntea nu va reuşi să găsească destinaţia, deoarece tabela sa de comutare era goală, astfel încât va

retransmite cadrul pe toate segmentele la care este ea conectată, în afară de segmentul de pe care a fost primit cadrul.

Înainte de a retransmite cadrul puntea va verifica dacă adresa sursă este prezentă în tabela sa de comutare. În cazul nostru ea nu este, astfel încât puntea va crea prima intrare în tabela de comutare ce va conţine adresa fizică a staţiei A1 şi interfaţa ce conectează segmentul A.

Cadrul va ajunge atât pe segmentul D, unde staţiile D1 şi D2 vor determina că acesta nu le este adresat lor, deci îl vor ignora; cât şi pe segmentul B, la staţiile B1, B2 şi puntea 2. Puntea 2 va determina că destinaţia este în acelaşi segment din care a primit cadrul şi va decide să nu-l mai retransmită, iar staţia B1 va determina că ea este destinatarul cadrului.

Chiar şi comunicaţia între două staţii aflate în acelaşi segment poate afecta lăţimea de bandă din întreaga reţea, dacă puntea nu a apucat să-şi construiască tabela de comutare.

După cadrul trimis către staţia B1, să considerăm că staţia **A1 va trimite un cadru pentru A2**. Cadrul va ajunge la destinaţie fără ajutorul punţii, dar puntea, neidentificând destinaţia în tabela sa de comutare, va retransmite cadrul atât pe segmentul B, cât şi pe segmentul D.

Datorită dificultăţii căutării într-o mulţime neordonată, în tabela de comutare **nu se vor păstra toate adresele staţiilor din reţeaua locală**, **ci doar a celor ce au o probabilitate mare să transmită în viitorul apropiat**, mai exact a ultimilor staţii ce au transmis. Pentru implementarea acestui concept, o întrare într-o tabelă de comutare va avea, **pe lângă adresa *MAC* şi interfaţa, şi o etichetă de timp**. Această etichetă de timp este actualizată la o nouă primire a unui cadru cu aceeaşi adresă sursă. Acest mecanism permite înlăturarea intrărilor învechite şi deci restrângerea dimensiunii tabelei de comutare. Preţul plătit pentru aceasta este consumul din lăţimea de bandă a tuturor segmentelor din reţea în cazul în care o staţie nu transmite nici un cadru într-un interval de timp.

Verifică-ţi cunoştinţele:

1) Enumeraţi dispozitivele folosite în *LAN* la nivelul 2.
2) Explicaţi principiile de funcţionare a punţilor.
3) Rolul punţii în comunicaţia din interiorul aceluiaşi segment.
4) Rolul punţii în comunicaţia dintre segmente.
5) Cum îşi construieşte puntea tabela de comutare.

6.3. Comutator (*Switch*)

Un comutator de reţea (*switch*) - este un dispozitiv care realizează interconectarea diferitelor segmente de reţea pe baza adreselor *MAC*. Uneori comutatorul este privit ca un dispozitiv de interconectare ce acţionează atât la nivel Fizic, cât şi la nivel Legătură de date [94, 95].

***Switch*-ul** – este un dispozitiv cu mai multe porturi (*bridge* multiport).

Informaţiile folosite pentru comutarea pachetelor sunt ţinute într-o **tabelă denumită *Content Adressable Memory* (*CAM*)** care trebuie construită automat de către *switch* în cadrul procesului de memorizare a adreselor. Procesul este relativ simplu: după ce *switch*-ul efectuează nişte verificări de bază despre integritatea cadrului, se trece la căutarea în *CAM* a adresei; dacă cadrul trece de verificări impuse de utilizator, se scrie în tabelă o nouă intrare. Altfel, se modifică una deja existentă sau nu se executa nici o acţiune.

Dacă într-o reţea sunt prezente doar *switch*-uri şi nu există *hub*-uri, atunci **domeniile de coliziune sunt fie reduse**, fie eliminate.

Pentru comutarea cadrelor *switch*-ul analizează cadrele primite şi decide dacă ele trebuie trimise, şi dacă da, pe ce port. Acest proces este destul de complex şi nu poate fi realizat eficient decât cu procesoare specializate.

Analiza diferenţelor dintre *switch* şi alte dispozitive multiport:

a) Diferenţa *hub – switch*.

Există două paradigme în reţelele de calculatoare: arhitecturi **bazate pe magistrală şi indirect pe difuzare** şi **arhitecturi bazate pe comutare**. Optarea pentru una dintre cele două paradigme se traduce în decizia **de a folosi un comutator sau un *hub*.**

Un *hub* este cel mai simplu dispozitiv *multiport*. Totuşi, tehnologia folosită este considerată depaşită din moment ce un *hub* retrimite orice pachet de date primit la toate porturile sale cu excepţia celui de la care l-a primit. Prin folosirea *switch*-ului acest neajuns a fost rezolvat.

O reţea *Ethernet* poate fi constituită:

• doar dintr-un singur cablu (coaxial) legând un număr de calculatoare;

• dintr-un repetor, conectând: fiecare calculator printr-un segment de cablu (torsadat); câte un segment conţinând mai multe calculatoare. Fiecare port al unui repetor leagă împreună segmentele de cablu *Ethernet* individuale pentru a crea o nouă reţea ce functionează ca un *Ethernet* independent şi singular. Segmentele şi repetoarele din această nouă reţea trebuie să respecte limitările timpului de întoarcere.

Mai multe reţele *Ethernet* pot forma **o reţea extinsă** prin utilizarea unui **comutator** de pachete. În timp ce o reţea *Ethernet* simplă poate suporta un număr de câteva zeci de staţii, o reţea extinsă poate lega câteva sute sau mii de staţii.

Spre deosebire de un repetor ale cărui porturi combină segmentele de cablu pentru a forma un singur *LAN*, **un comutator face posibilă divizarea unei reţele *Ethernet* de dimensiuni mari, în mai multe reţele *Ethernet* independente**, ce sunt legate printr-un mecanism de comutare a pachetelor. **Comutatoarele examinează fiecare pachet** recepţionat pe fiecare port, îl procesează şi îl transmite (dacă este cazul), pe baza unei baze de date iniţiale sau create dinamic, către portul ce corespunde staţiei destinaţie. Pe când repetorul retransmite fiecare cadru primit pe toate porturile, fără nici

un fel de prelucrare a pachetului. În comutator se păstrează o bază de date cu adresele *Ethernet* ale staţiilor şi portul din comutator corespunzător fiecărei staţii.

Principalul **avantaj al înlocuirii *hub*-urilor cu comutatoare** nu îl reprezintă înlăturarea restricţiilor impuse de regula 5-4-3, ci **reducerea numărului de utilizatori ce partajează aceeaşi lăţime de bandă**.

Concluzionăm, că *switch*-ul este o alternativă de mai înaltă performanţă la un *hub*. *Hub*-urile operează utilizând un model *broadcast,* iar *switch*-ul operează utilizând un model de circuit virtual. Comutatoarele vor oferi protecţie împotriva atacurilor prin ascultare a liniei. Astfel, „războiul" *hub versus comutator* opune costul şi latenţa mai scăzute, pe de o parte, cu cerinţele crescânde de lăţime de bandă disponibilă şi de securitate, pe de altă parte [11]. Aceasta nu se datorează unei latenţe mai mici sau unui cost mai scăzut, ci datorită faptului că **în reţelele *Ethernet* ce folosesc mediul torsadat,** comutatorul preia funcţia principală a *hub*-ului, şi anume aceea de a asigura **conectarea tuturor nodurilor la un mediu de transmisie.**

b) **Diferenţele dintre un *switch* şi un *bridge***. Definiţia cea mai răspândită a *switch*-urilor identifică orice punte *multiport* cu un *switch*. În realitate, deşi această definiţie acoperă majoritatea cazurilor, **există punţi *multiport* ce nu sunt *switch*-uri.**

Numărul de interfeţe sau porturi este fără îndoială cea mai importantă diferenţă. Cerinţele de latenţă pentru o punte cu două interfeţe sunt mult mai relaxate decât pentru un comutator. Din această cauză **punţile comută pachete folosind componente *software,*** în vreme ce **comutatoarele vor lua toate deciziile la nivel *hardware*.**

Puntea reface semnalul la nivel de bit, pentru a obţine un cadru, apoi despachetează cadrul, foloseşte informaţiile din câmpul „adresă destinaţie" pentru a filtra sau nu cadrul, iar „adresa sursă" va fi

folosită pentru construirea tabelei de comutare. Dar una dintre funcţiile nivelului Legătură de date este acela de a oferi mecanisme de corecţie a datelor la nivel de cadru.

Cele mai importante **două diferenţe dintre un comutator şi o punte** se referă la metodele de comutare. Faţă de punţi, **comutatoarele implementează metode de comutare mai rapide.** În general, **punţile**, deşi nu sunt interesate de detecţia unui număr cât mai mare de erori, implementează doar **comutarea după stocare**[35], aceasta există mai degrabă din raţiuni istorice decât ca rezultat al unei decizii de optimizare a traficului în reţea. Cea de a doua diferenţă se referă la capacitatea **comutatoarelor de a permite mai multe comunicaţii simultane fără a scădea lăţimea de bandă alocată fiecăreia dintre conexiuni.**

Cele două diferenţe dintre *switch*-uri şi *bridge*-uri sunt în fapt **avantaje ale *switch*-urilor**, iar preţul unui *switch* este foarte apropiat de cel al unui *bridge*. Cu toate acestea încă se mai produc *bridge*-uri şi în ziua de azi.

Există un caz în care cele două avantaje ale comutatoarelor nu mai sunt relevante. Este vorba de interconectarea a două reţele ce folosesc **protocoale de nivel 2 diferite**. În acest caz singura metodă de comutare posibilă este comutare după stocare (*store-and-forward*), deoarece cadrele trebuie reîmpachetate.

6.3.1. Tipurile de comutare folosite de un comutator. Există două metode de comutare a pachetelor: (a) comutare după stocare (*store-and-forward*) şi (b) comutare directă (*cut through*).

a) **Metoda de comutare după stocare.** - se bazează pe recepţionarea întregului cadru înainte de a începe retransmisia acestuia. Latenţa acestei metode creşte odată cu dimensiunea câmpului de date. Cu toate acestea, performanţele metodei de comutare după stocare pot fi superioare celor oferite de comutarea

[35] Metoda de comutare după stocare - se bazează pe recepţionarea întregului cadru înainte de a începe retransmisia acestuia.

directă, mai ales în cazul liniilor expuse unor interferenţe puternice. Mecanismele de detecţie a erorilor pe care le oferă această metodă de comutare permite asigurarea unei conexiuni sigure la nivelul Legătură de date.

Metoda de comutare după stocare ridică şi **problema asigurării memoriei pentru stocarea cadrelor**. Să luăm exemplul unui comutator cu 24 de porturi. Acesta va trebui să poată gestiona 12 comunicaţii simultane, care în cel mai defavorabil caz posibil vor transfera cadre de lungime maximă. Deşi dimensionarea memoriei ***RAM* folosite pentru stocarea cadrelor** nu este principalul factor de stabilire a preţului unui *switch*, nu trebuie omis faptul că preţurile pentru **memoriile dispozitivelor dedicate este de câteva ori mai ridicat** decât cel pentru memoriile folosite în calculatoarele personale.

b) Comutarea directă - presupune ca puntea să înceapă transmiterea cadrului pe portul destinaţie imediat ce adresa destinaţie a fost trecută prin tabela de comutare şi interfaţa de plecare a fost determinată. Pentru comutarea directă nu este necesară nici măcar recepţionarea integrală a antetului cadrului, adresa destinaţie fiind suficientă. Această metodă se numeşte comutare directă rapidă (*fast forward*) şi oferă o latenţă de aproximativ 21 de microsecunde.

Datorită faptului că retransmisia cadrului începe imediat după citirea adresei destinaţie, **cadrele eronate vor fi transmise cu erori**. Deşi aceste cadre sunt respinse la nivelul Legătură de date al destinaţiei (de către placa de reţea), traficul generat de retransmisia lor poate să ducă la o depreciere severă a performanţelor reţelei.

Al doilea tip de comutare directă este **comutarea fără fragmente** (*fragment free*). Pentru această metodă de comutare vor fi **filtrate fragmentele de cadre rezultate în urma unei coliziuni**. Într-o reţea ce respectă specificaţiile standardului *Ethernet*, **dimensiunea fragmentelor de coliziuni nu poate depăşi 64 de octeţi**. Pentru comutarea fără fragmente, comutatorul va determina că şirul de octeţi recepţionaţi nu fac parte dintr-un fragment de

coliziune şi abia apoi va începe retransmisia pe portul destinaţie. Latenţa în acest caz este de **minim 51,2 microsecunde**, ceea ce reprezintă timpul necesar recepţionării a 64 de octeţi.

6.3.2. Rolul comutatoarelor în implementarea conexiunilor *Ethernet half-duplex*. Comunicaţia *semi-duplex* (*half-duplex*) permite doar unui singur nod să transmită date. În *Ethernet* aceasta este controlată cu ajutorul coliziunilor. Dacă două sau mai multe staţii încearcă să comunice simultan, rezultatul va fi o coliziune.

Pe interfeţele unui comutator putem conecta o staţie sau un segment întreg. Reţelele comutate vor folosi câte un port pentru fiecare staţie, **reducând dimensiunea domeniilor de coliziune la doar două noduri** (unul fiind placa de reţea din respectiva staţie, iar cel de-al doilea - portul din comutator ce o conectează pe aceasta).

Altfel spus, **comutatoarele oferă suportul pentru implementarea reţelelor comutate**, reţele în care domeniile de coliziune nu depăşesc două noduri.

6.3.3. Rolul comutatoarelor în implementarea conexiunilor *Ethernet full-duplex*. Ethernetul *full-duplex* permite trimiterea şi recepţionarea simultană. Pentru implementarea sa este suficientă folosirea a două perechi de fire, la fel ca şi în cazul comunicaţiei *semi-duplex*. Diferenţa faţă de *semi-duplex* apare în numărul nodurilor (a staţiilor) ce pot participa într-o astfel de conexiune. Astfel, conexiunea pentru o legătură *full-duplex* este considerată punct-la-punct, adică poate fi folosită de două şi numai două noduri. Acesta este şi motivul pentru care **doar comutatoarele şi nu şi *hub*-urile pot comunica *full-duplex*.**

Ethernetul, datorită coliziunilor, foloseşte în medie 50-60% din bandă, în vreme ce *Ethernetul full-duplex* oferă 100% din bandă în ambele sensuri, adică o bandă potenţială de *20 Mbps* (câte *10 Mbps* pe sens).

Eliminarea coliziunilor duce şi la eliminarea circuitelor de detecţie a coliziunilor de la nivelul plăcilor de reţea şi a

comutatorului, deci şi a latenţei introdusă de detecţia acestora.

Verifică-ţi cunoştinţele:

1) Caracterizaţi funcţiile comutatotorului de reţea.
2) Analizaţi diferenţele dintre ***switch*** şi alte dispozitive multiport.
3) Tipurile de comutare folosite de un comutator.
4) Rolul comutatoarelor în implementarea conexiunilor *Ethernet half-duplex* şi *full-duplex*.

6.4. Ruterele

Ruterele sunt folosite pentru interconectarea **la nivelul Reţea** mai multor reţele locale şi are **rolul de a determina calea ce trebuie urmată de un pachet pentru a ajunge la destinaţie**. Ruterul poate fi întâlnit mai ales la nivel *WAN*, dar şi la nivelul reţelei locale, una din funcţiile sale principale, fiind şi acea de a oferi posibilitatea **conectării *LAN*-urilor la *WAN*** [96, 97].

Procesul de rutare sau de determinare a căii optime se bazează pe construirea şi menţinerea unei **tabele de rutare**. O intrare într-o **tabelă de rutare** se numeşte **rută şi este compusă din minim 3 elemente**: adresă de reţea, mască de reţea, adresa următorului ruter şi/sau interfaţă de plecare.

6.4.1. Tabele de rutare. O tabela de rutare este **o listă de rute cu acces secvenţial**. Folosirea tabelei de rutare se face analizând secvenţial rutele începând cu prima. Construcţia tabelei se face prin **inserarea oricărei noi rute în faţa primei rute**. Ruterul verifică mai întâi dacă adresa destinaţie nu este cumva una dintre adresele sale. Dacă este printre adresele sale, atunci cadrul va fi trecut la nivelul superior, dacă nu - ruterul va verifica dacă adresa destinaţie nu este în aceeaşi reţea cu interfaţa de pe care a primit pachetul. Dacă este, atunci va abandona prelucrările asupra respectivului pachet şi va lua următorul pachet [98].

În cazul, în care destinaţia nu este nici el şi nici nu se află pe aceeaşi interfaţă de unde a primit pachetul, atunci va începe procesarea tabelei de rutare. Va extrage prima rută din tabelă şi va **aplica masca de reţea adresei destinaţie conţinută în antetul pachetului.** Rezultatul îl va compara cu adresa de reţea a respectivei rute. Dacă cele două coincid, pachetul va fi trimis pe interfaţa specificată de rută. Daca nu, este extrasă o nouă rută din tabelă.

Procesul se repetă până la ultima rută din tabelă sau până la găsirea primei potriviri. Dacă pachetul nu corespunde nici ultimei rute atunci acesta este abandonat şi se trece la pachetul următor. Înainte de a trimite pachetul sau de a îl abandona, tabela *ARP* (*Address Resolution Protocol*) a interfeţei pe care a sosit pachetul va fi actualizată folosindu-se adresa *MAC* şi cea *IP* a sursei.

Astfel, deşi adresa următorului *hop* este întotdeauna de ajuns pentru specificarea completă a unei rute, informaţia despre interfaţa de ieşire se dovedeşte uneori insuficientă şi anume în cazul în care această interfaţă este conectată la un mediu *multiacces*.

6.4.2. Clasificări ale rutelor. Există numeroase criterii de clasificare a rutelor.

1) O primă clasificare a rutelor a fost în funcţie de **tipul procesului de rutare**, şi anume *classfull* sau *classless*. Odată cu creşterea în popularitate a adresării *classless,* tabelele de rutare au devenit *classfull,* chiar dacă sunt alimentate uneori de protocoale de rutare *classless* (adică protocoale ce nu transmit informaţii despre masca de reţea), ruterele urmând să precizeze explicit masca de reţea înainte de a introduce informaţiile în tabela de rutare.

În rutarea cu rute *classfull* adresa destinaţie extrasă din antetul unui pachet ajuns la ruter va fi mai întâi **comparată cu 192**, şi în cazul în care e mai mică de 192 va fi comparată **cu 128,** determinându-se astfel clasa de adrese şi implicit masca de reţea. Din acest punct procesul este similar cu cel din rutarea *classfull*, adică se va efectua **o operaţie de „şi" logic între adresa destinaţie şi masca**

reţelei, rezultatul urmând a fi comparat cu adresa de reţea conţinută în rută. Odată cu răspândirea rutării *classless* a apărut clasificarea rutelor în funcţie de **tipul destinaţiei**. Astfel vom avea: **rute de tip nod (sau rute *host*) şi rute de tip reţea**.

Rutele de tip *host* conţin informaţii doar despre o singură staţie, adică **masca de reţea este /32**. Odată cu creşterea Internetului, şi a dimensiunii tabelelor de rutare, a apărut tendinţa de a agrega cât mai mult de rute, precum şi a **se renunţa la rutele de tip nod**. Cu toate acestea, datorită promovării rutelor de nod la începutul tabelei de rutare, acestea având prefixul maxim, **rutele *host*** mai sunt încă folosite pentru unele optimizări de trafic, mai ales **pe ruterele de la periferia Internetului**.

2) Alt criteriu de clasificare a rutelor reprezintă **modul de conectare**, iar cele două tipuri de rute sunt: (a) **rutele direct conectate** şi (b) **rute *gateway***.

a) **Rutele direct conectate** sunt rute către reţele în care **ruterul are o interfaţă**, şi în majoritatea cazurilor aceste rute sunt **automat introduse în tabela de rutare** de către sistemul de operare odată cu configurarea şi activarea interfeţei respective. Rutele direct conectate în general nu conţin adresa următorului *hop*, având specificată doar interfaţa de ieşire din ruter. Astfel rutele direct conectate sunt singurele rute valide ce pot avea specificată **ca interfaţă de ieşire o interfaţă *multiaccess* (gen *Ethernet*), fără a necesita precizarea adresei următorului *hop*.**

b) **Un *gateway*** este, în genere, un nod de reţea care conectează diferite segmente de reţea, frecvent - reţelele interne cu Internet. *Gateway*-ul este asociat cu un ruter, care utilizează antetele şi tabele de rutare pentru a determina calea reală a pachetului şi poarta de acces. Cu alte cuvinte, un *gateway* oferă un punct de intrare şi un punct de ieşire într-o reţea.

3) O altă clasificare a rutelor se face în **funcţie de mediul de acces** către o reţea, având astfel **rute pe medii punct la punct şi rute pe medii *multiacces***. Diferenţa între cele două tipuri de rute constă în faptul că rutele către o reţea conectată pe o legătură punct

la punct pot fi specificate ori prin interfaţa de ieşiere din ruter, ori prin adresa următorului *hop*, ori prin ambele, în vreme ce rutele pe medii *multiacces* sunt specificate doar prin adresa următorului *hop*, interfaţa de ieşire nefiind suficientă.

Ar fi important de precizat că din punctul de vedere al unui ruter, două medii de transmisie acoperă marea majoritatea a rutelor: interfeţele *Ethernet* şi cele seriale, prima - permiţând transmisia peste un mediu *multiacces*, în vreme ce cea de a doua - este o interfaţă punct la punct. Alte interfeţe punct la punct destul de populare sunt cele de fibră optică şi cele *ISDN* (*Integrated Services Digital Network*).

4) Există o clasificare ce împarte rutele în **rute generice şi rute specifice**. Această clasificare este artificială deoarece rutele agregate au trecut în tabăra rutelor specifice, sigura rută generică fiind ruta implicită sau ruta *default*. **Ruta implicită sau ruta *default*** este ruta spre care se trimit toate pachetele pentru care nu se cunoaşte o destinaţie specifică. Altfel spus, ruta *default* este ruta care se potriveşte cu toate destinaţiile, având în partea de adresă de reţea din rută un spaţiu de adrese ce cuprinde toate adresele *IP*. Acest **spaţiu de adrese este 0.0.0.0/0** şi deşi deseori ruta *defaul*t este denumită ca ruta cu 4 de zero sau *quad-zero route*, esenţa acestei rute se află în **masca de lungime zero**.

Analizăm, în ce măsură putem avea într-o tabelă de rutare mai mult de o rută *default*. Să luăm tabela de rutare (Tabelul 6.2.), ale cărei ultime două rute sunt două rute *default*.

Tabelul 6.2. Tabela de rutare

Adresă reţea	Mască	*Next hop*	Interfaţă
…..			
0.0.0.0	**/0**	**194.230.5.65**	**S1**
0.0.0.0	**/0**	**-**	**S1**

În mod evident **nici un pachet nu va ajunge să prelucreze ce-a de a doua rută implicită**, toate pachetele fiind acceptate de prima.

Dezactivarea unei interfeţe, ce poate avea loc ori ca o consecinţă a unei închideri administrative sau a întreruperii legăturii de nivel Fizic sau a celei de nivel Legătură de date, are drept consecinţă directă **înlăturarea tuturor rutelor ce folosesc respectiva interfaţă**, ca interfaţă de ieşire din ruter. Astfel, în cazul în care nu am avea cea de a doua rută *default* şi interfaţa S0 ar fi dezactivată, toate pachetele care ar fi fost rutate prin prima ruta implicită ar urma să fie ignorate.

În concluzie, într-o tabelă de rutare **există o singură rută *default* activă**, dar pot fi precizate mai multe rute *default* în scopuri de *backup*.

5) Ultima clasificare a rutelor este cea mai semnificativă. Această clasificare se face în funcţie de modul în care informaţia pe baza căreia sunt construite rutele şi se împart în **rutele statice şi rutele dinamice**.

Rutele statice – sunt introduse manual de către administratorul ruterului, spre deosebire de rutele dinamice ce necesită doar configurarea unui protocol de rutare, rutele urmând a fi învăţate schimbând informaţii despre rutele direct conectate cu celelalte rutere.

Clasificarea rutelor în rute statice şi rute dinamice se referă doar la rutele *gateway,* deoarece rutele direct conectate sunt introduse automat în tabela de rutare.

6.4.3. Efectul ruterelor asupra domeniilor de difuzare şi a domeniilor de coliziune. Ruterul va face atât **regenerarea semnalului** cât şi **detecţia coliziunilor**. În plus ruterele, spre deosebire de punţi au acces şi la informaţiile de nivel Reţea, permiţându-le **controlul difuzărilor** şi a **pachetelor de *multicast***. În mod implicit ruterele nu transferă pachetele de difuzare sau de *multicast*. Astfel, **ruterele mărginesc atât domeniile de coliziune, cât şi pe cele de difuzare**.

Deoarece **rolul unui ruter este de a direcţiona pachetele între diversele reţele** pe care le interconectează, **principalele acţiuni** pe care le realizează ruterul în procesul de comutare a pachetelor sunt:

- examinarea **pachetului** şi determinarea tipului acestuia precum şi a adresei destinaţie;
- determinarea **adresei următorului ruter** (sau *host*) către care trebuie trimis pachetul prin examinarea tabelei de rutare;
- determinarea **interfeţei** pe care urmează să fie transmis pachetul;
- determinarea **adresei de nivel 2 a următorului ruter** (sau *host*);
- **reîncapsularea pachetului cu informaţiile de nivel 2** necesare şi transmiterea sa pe interfaţa determinată anterior.

În afară de aceste operaţii de bază, ruterul mai poate efectua şi **operaţii de filtrare de pachete**.

6.4.4. Tipurile ruterelor. Pot fi identificate următoarele tipuri ale ruterelor: (a) rutere cu memorie partajată; (b) rutere cu procesoare de comutare; (c) rutere cu procesoare multiple.

a) **Rutere cu memorie partajată**. Primele generaţii de rutere au avut la bază arhitectura unui calculator de uz general şi foloseau pentru comutare de pachete de arhitectura cu memorie partajată (*shared memory routers*). Componentele principale ale unui ruter din prima generaţie sunt: **procesorul** (asigură, prin implementarea *soft*-urilor pe parcursul comutării pachetelor, interfaţa cu utilizatorul fără a utiliza *hardware* specializat; construieşte tabelele de rutare şi le menţine); **memoria** (este gestionată de către sistemul de operare) şi **interfeţele** (elementele discrete ce asigură recepţionarea şi transmiterea pachetelor pentru diversele medii de transmisie suportate de ruter).

Exemple clasice de **rutere din prima generaţie** sunt ruterele ***Cisco*** *1600* şi *2500*.

b) **Rutere cu procesoare de comutare**. Ruterele de tip *shared memory routers* au o serie de **dezavantaje** datorită faptului că există o **singură unitate de prelucrare** – procesorul, care realizează mai multe operaţii în acelaşi timp. Pentru creşterea performanţelor s-a încercat dezvoltarea unor altfel de arhitecturi în care procesul de *switching* să se facă în *hardware*, şi nu *software*. În aceste tipuri de arhitecturi există două tipuri de procesoare: **un procesor de rutare** (*route processor* - care, în afară de menţinerea *cache*-lui de rute, rulează procesele ce asigură interfaţa cu utilizatorul, procesele de menţinere a tabelei de rutare, translatarea de adresă, etc.) şi **un procesor de comutare** (*switch processor* - specializat pe comutarea de pachete).

c) **Rutere cu procesoare multiple.** Deşi ruterele cu procesoare de comutare au ridicat performanţele ruterelor destul de mult, pentru un ruter cu multe interfeţe, ce suportă medii cu lărgimi de bandă de ordinul sutelor de *Mbps*, **un sigur procesor de comutare nu mai face faţă**. Arhitectura ruterelor cu procesoare multiple a fost direcţionată către o arhitectură paralelă pentru a crea rutere scalabile.

Ruterele folosesc o abordare *ASMP* (*ASymmetric Multi-Processing*) în care există **un procesor principal** care coordonează unul sau mai multe procesoare secundare. Datorită faptului că procesoarele secundare sunt procesoare puternice toate facilităţile pot fi implementate distribuit.

Verifică-ţi cunoştinţele:

1) Cu ce scop sunt folosite ruterele?
2) Ce reprezintă Tabela de rutare?
3) Enumeraţi criteriile de clasificare a rutelor.
4) Efectul ruterelor asupra domeniilor de difuzare şi a domeniilor de coliziune.
5) Descrieţi avantajele şi dezavantajele diferitor tipuri ale ruterelor**.**

6.5. Protocolul *STP* (*Spanning Tree Protocol*) [52]

O buclă de nivel Legătură de date apare într-o reţea când **între două dispozitive ale acesteia există două sau mai multe legături active**, fiecare conexiune folosind doar dispozitive de interconectare ce pot analiza cel mult informaţii de nivel Legătură de date.

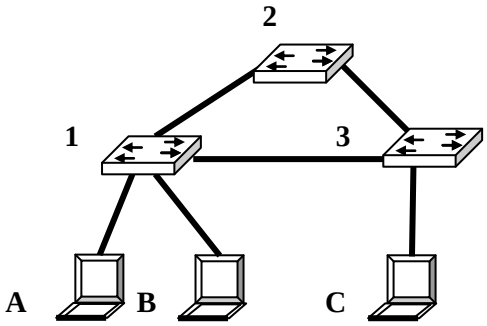


Fig. 6.3. Reţea în care s-a creat o buclă [11]

Apariţia buclelor de nivel Legătură de date este corelată cu faptul că **punţile şi comutatoarele nu filtrează pachetele de difuzare** şi duc la o depreciere semnificativă a performanţelor reţelei prin determinarea unor **avalanşe de difuzări** (*broadcast storm*).

Să considerăm reţeaua din figura 6.3. Presupunem că staţia A trimite un cadru de difuzare. Comutatorul 1 nu va găsi adresa destinaţie în tabela sa de comutare, astfel încât va transmite cadrul pe celelalte segmente: segmentul ce conţine staţia B, segmentul dintre comutatoarele 1 şi 2, şi segmentul dintre comutatoarele 1 şi 3. Staţia B va examina cadrul, va decide că îi este adresat şi îl va trece spre nivelul Legătură de date.

Comutatorul 2 va lua decizia de a transmite cadrul pe toate interfeţele sale, cu excepţia celei de pe care a primit cadrul. Am ajuns să avem în reţea două cadre destinate staţiei FF.FF.FF.FF.FF.FF,

adică două cadre de difuzare. Indiferent de ordinea în care acestea ajung la comutatorul 3, acesta va determina că nu cunoaşte adresa destinaţie şi le va retransmite către staţia C, dar şi către celelalte comutatoare.

Avalanşa de difuzări **consumă din banda utilă a reţelei**, ducând la o micşorare a bandei efective disponibile. O avalanşă de difuzări se **va opri doar în cazul întreruperii buclei**.

6.5.1. Prevenirea apariţiei avalanşelor de difuzări. Soluţia trivială ar fi să instruim **punţile şi comutatoarele să nu retransmită cadrele de difuzare**. Din păcate acest lucru nu este posibil, deoarece o serie de protocoale folosesc cadre de difuzare pentru a funcţiona corect, unul dintre acestea fiind chiar *ARP - Address Resolution Protocol*. Altfel spus, filtrarea cadrelor de difuzare de către punţi ar presupune **rescrierea protocoalelor fundamentale** ce asigură suportul de comunicaţie.

Soluţia validă presupune identificarea buclelor şi întreruperea lor. Protocolul ce realizează aceasta se numeşte *STP - Spanning Tree Protocol*, şi presupune **construirea unui arbore de acoperire pe graful determinat** de dispozitivele de interconectare şi de conexiunile dintre acestea.

6.5.2. Modul de funcţionare a *STP*. Funcţionarea acestui protocol se bazează pe crearea topologiei reţelei folosind nişte cadre speciale numite cadre *BPDU* (*Bridge Protocol Data Unit*). Aceste cadre speciale sunt folosite intens la **iniţializarea comutatoarelor**; ulterior, la **fiecare două secunde** vor fi schimbate cadre *BDPU*, pentru a verifica dacă nu au apărut modificări.

Totodată sunt **definite cinci stări** în care se poate afla o interfaţă a comutatorului: starea **blocat, de ascultare, de învăţare, de comutare de cadre şi nefuncţional** (*blocking, listening, learning, forwarding, disabled*).

- în starea „blocat” nu se acceptă decât cadre *BPDU,* în cea de ascultare se primesc şi cadre, dar acestea nu sunt retransmise;
- în starea „de învăţare”, în plus faţă de starea „de ascultare”, este inspectată adresa sursă a cadrelor primite, permiţând astfel construirea tabelei de comutare;
- în starea „de comutare” cadrele primite sunt retransmise, iar tabela de comutare este actualizată;
- în starea „nefuncţional” nu se vor accepta nici cadre *BPDU*.

Pentru **construirea arborelui de acoperire**[36] sunt necesare aproximativ **30 de secunde**, timp în care toate porturile comutatoarelor sunt în starea „blocat”. Există trei paşi ce trebuie urmaţi pentru construirea arborelui de acoperire: (1) mai întâi trebuie aleasă **rădăcina arborelui** (*root bridge*), (2) apoi trebuie alese **porturile rădăcină**, pentru ca în final (3) să fie determinate **porturile active**.

Prioritatea punţii are o valoare numerică implicită **atribuită de producător**, o valoare păstrată în memoria fiecărei punţi, ce poate fi modificată ulterior. Pe baza comparării priorităţilor tuturor punţilor din reţea se va determina puntea cu **prioritatea cea mai scăzută, aceasta devenind rădăcina arborelui de acoperire**.

În cazul folosirii mai multor echipamente produse de aceeaşi firmă, se întâmplă adesea să existe mai multe punţi ce vor avea aceeaşi prioritate. Dintre două sau mai multe **punţi cu aceeaşi prioritate,** rădăcina arborelui să devină pe baza **celei mai mici** adresei fizice.

Pasul al doilea presupune identificarea **căilor redundante dintre fiecare punte şi puntea rădăcină**, apoi selectarea unei sigure căi între respectiva punte şi rădăcină şi, în final, dezactivarea celorlalte.

[36] Tabelul Costului portului, prin care trece calea în dependenţă de Lăţimea de bandă, şi algoritmul de bază a STP vezi Anexa 3.

Pentru evaluarea unei căi vom determina **costul căii**, care va fi definit ca **suma costurilor porturilor prin care trece calea**.

Costul unui port este determinat pe lăţimea de bandă pe care o oferă portul sau uneori chiar pe mediul de transmisie folosit pentru conectarea la port.

De exemplu, pentru comutatoarele *Cisco* costul portului este determinat prin împărţirea lui 1000 la **lăţimea de bandă pe care o oferă portul (*10 Mbps*)**, astfel încât un port *Ethernet* va avea costul 100.

Pentru alegerea **porturilor rădăcină** vor avea **prioritate porturile conectate direct la rădăcina arborelui de acoperire**. În cazul în care nu există nici un port cu o conexiune directă spre puntea rădăcină, sau când avem mai mult de un singur port cu conexiune directă spre rădăcină, va fi ales portul ce are cel **mai scăzut cost al căii spre rădăcină**.

Fie reţeaua din figura 6.4. Vom urmări pentru această reţea etapele construirii arborelui de acoperire. Prima întrebare pe care trebuie să ne-o punem este: **care este prioritatea fiecărui comutator?** Să considerăm că toate cele trei comutatoare sunt produse de acelaşi fabricant. Asta înseamnă că toate comutatoarele vor avea aceeaşi prioritate. În acest caz va trebui să aflăm adresele fizice.

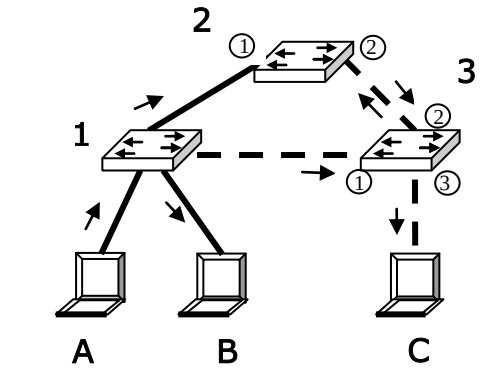
Tabelul 6.3. Adresele fizice ale comutatorilor

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

Din analiza tabelului 6.3. rezultă că **rădăcina arborelui de acoperire va fi comutatorul 1**.

În continuare vom determina pentru restul comutatoarelor **costurile porturilor ce oferă căi spre comutatorul rădăcină**.

Pentru comutatorul 2 costul portului 1 va fi 100 (=*1000/10Mbps*), iar pentru portul 2 va fi 200 (100 + costul portului 2 din comutatorul 3). Pentru comutatorul 3, portul 1 va avea costul 100, iar portul 3 costul 200.



Conexiune *Ethernet*

Conexiune *Fast Ethernet*

Fig. 6.4. Construirea arborelui de acoperire [5]

Pentru comutatorul 2 **portul rădăcină va fi portul 1**, astfel încât portul 1 trece în starea de comutare, în vreme ce portul 2 va rămâne în starea de blocat.

Pentru comutatorul 3 **portul rădăcină va fi portul 1**, deoarece, este direct conectat la rădăcină, astfel încât portul 1 va trece în starea de comutare, porturile 2 şi 3 vor rămâne în starea de blocat.

Verifică-ţi cunoştinţele:

1) Descrieţi procesul de apariţie a buclelor de nivelul 2.
2) Cum se poate preveni apariţia avalanşelor de difuzări?
3) Explicaţi modul de funcţionare a *STP*.

6.6. Placa de reţea (adaptor *LAN*)

Fiecare computer se conectează la reţea printr-o **placă de reţea** (denumită şi ***adaptor LAN*** - *Network Interface Card* (*NIC*)), care transmite prin mediile din reţea impulsuri electrice, semnale luminoase, unde electromagnetice [99, 100].

Fiecare *NIC* are **o adresa unică** scrisă într-un *ROM*. Menţionăm că aceasta adresă se numeşte adresa *MAC* (*Media Access Control*).

Orice placă de reţea îndeplineşte **următoarele funcţii**:

a) **Pregăteşte datele** pentru a putea fi transmise printr-un mediu; **transmite datele**; **controlează fluxul** datelor de la PC la mediul de transmisie. Prin reţea **datele circulă în serie** (un bit odată) în timp ce **în interiorul calculatorului circulă în paralel** (16, 32 sau 64 biţi odată, în funcţie de *bus*-ul sistemului). Prin urmare, **cartela de reţea** trebuie să convertească datele care circulă în interiorul PC-ului **în format serial**.

b) Plăcile de reţea şi suportul *soft* **recunosc erorile**, **coliziunile** sau **echipamentele defectate**, **provoacă alterarea unor porţiuni din pachetul de date**. Erorile sunt în general detectate cu ajutorul unei sume de verificare ciclică (*CRC*). Câmpul *CRC* este verificat de receptor; dacă valoarea calculată de acesta nu se potriveşte cu cea din pachetul de date, receptorul anunţă emiţătorul despre eroare şi îi cere retransmisia pachetului de date care a sosit eronat.

c) Tipul adaptorului *LAN* îl leagă de unul dintre nivelurile de protocol *Ethernet*, *Token Ring* sau alt protocol. Adaptoarele cu **detecţie a coliziunilor** şi cu **vehicularea mesajului *Token*** conţin suficientă logică pe placă pentru a şti când este permisă trimiterea unui pachet de date şi pentru a recunoaşte pachetele care le sunt destinate. Cu suportul *soft*-ului al adaptoarelor, ambele tipuri de plăci îndeplinesc **şapte paşi** importanţi în procesul de transmisie sau recepţie a unui pachet de date. Descrierea adaptoarelor pentru tehnologiile *Ethernet*, *ARCnet*, *LANtastic*, *TokenRing* este prezentată în Anexa 2.

Algoritmul de trimitere/recepţionare a datelor:

1. **Transfer de date** - datele sunt transferate din memoria calculatorului (*RAM*) la placă adaptoare sau de la aceasta către memoria calculatorului prin ***DMA*** (*Direct Memory Access*).

2. ***Buffering*** - sunt procesate de placă adaptoare de reţea, datele sunt reţinute într-un *buffer*. Această memorie-tampon oferă plăcii accesul la un pachet întreg de date deodată şi îi dă posibilitatea să gestioneze diferenţa dintre rata de transmisie a datelor în reţea şi rata cu care calculatorul procesează datele.

3. **Trimiterea/recepţionarea impulsurilor** - impulsurile codificate, care compun pachetul de date, sunt amplificate şi transmise pe linie. La recepţie, impulsurile trec prin etapa de decodificare.

4. **Codificarea/decodificarea** - datele transmise sau recepţionate sunt modelate.

5. **Conversia paralelă/serială** - datele din *buffer* sunt trimişi sau recepţionaţi prin cabluri în mod serial.

6. **Structura pachetului de date** - adaptorul pentru reţea trebuie să spargă datele în porţiuni care pot fi procesate sau la recepţie să le reasambleze[37].

7. **Acces la cablu** - într-o reţea *CSMA/CD*, înainte de a-şi trimite datele (sau de a-şi retransmite datele dacă apare o coliziune), adaptorul pentru reţea se **asigură că linia este liberă**. Într-o reţea de vehiculare a mesajelor *Token*, adaptorul aşteaptă până la primirea unui *token* pe care îl poate reclama.

---

[37] Într-o reţea *Ethernet* aceste porţiuni sunt de circa 1500 de octeţi. Reţelele *Token Ring* folosesc pachete de date de 4000 de octeţi. Adaptorul pune un preambul (*header*) în faţa pachetului de date şi îi adaugă la sfîrşit un postambul (*trailer*). *Header*-ul şi *trailer*-ul reprezintă anvelopa nivelului Fizic. În acest moment există un pachet de date gata pentru transmisie.

Verifică-ţi cunoştinţele:

1) Enumeraţi echipamentele de conectare la reţea.
2) Funcţiile plăcii de reţea.
3) Descrieţi algoritmul de trimitere/recepţionare a datelor.

6.7. Modemul

Accesul unui utilizator la Internet prin intermediul reţelei de telefonie analogice se efectuează în modul următor: calculatoarele implicate în reţea, comunică folosind fiecare un modem, acestea însă trebuie să fie conectate prin intermediul reţelei telefonice. **Linia telefonică** utilizată poate fi de două tipuri [101]:

- **linie cu comutare** (*dial-up*), la care realizarea legăturii se face **manual de către utilizator**, asigurând un trafic cu viteza destul de scăzută (*56 Kbps*) şi cu probleme privind erorile din cauza centralelor telefonice prin care trece semnalul;

- **linie dedicată** (închiriată), mai scumpe decât cele cu comutare, dar la care legătura este stabilită *non-stop* şi nu este întreruptă de centrala telefonică, asigurând astfel viteze până la *45 Mbps*.

Modemul este o componentă a calculatorului care **converteşte semnalele digitale (de transmis) în semnale analogice**, care pot circula în reţeaua telefonică. Apoi modemul „formează" numărul de telefon al unui furnizor de servicii Internet - *ISP*.

Semnalele modulate (de fapt datele) sunt transferate la punctul de livrare (*Point Of Presence*, *POP*) al *ISP*-ului, unde sunt preluate din sistemul telefonic şi transferate în reţeaua regională de Internet a *ISP*-ului. Din acest punct **sistemul este în întregime digital** şi se bazează pe **comutarea de pachete** (*packet switching*); în acest sistem de transmisie informaţia care trebuie transmisă este „mărunţită" în multe pachete mici, care sunt apoi transmise la destinaţie în mod **independent unele de altele şi chiar pe căi**

diferite; sigur că la destinaţie pachetele trebuiesc reasamblate în ordinea corectă.

Odată ajunse la postul telefonic, semnalul analogic este **demodulat în semnal digital şi folosit de serverul de *dial-up*** la care este ataşat modemul. Toate modemurile moderne permit traficul din ambele direcţii in acelaşi timp (folosind frecvenţe diferite pentru direcţii diferite).

Modemurile pot fi:

- **interne**, se prezintă ca o placă ce se poate instala în calculator;
- **externe**, sub forma unui dispozitiv în afara calculatorului, cu alimentare separată.

În ambele situaţii, modemul este legat atât la magistralele calculatorului (cel intern direct, cel extern printr-un cablu serial) cât şi la priza de perete ce asigură legătura cu reţeaua telefonică printr-un cablu cu conector.

Modemurile mai pot fi clasificate în:

- **asincrone**, dacă datele sunt transmise serial, fără o coordonare a transmisiei. Sunt relativ ieftine şi asigură viteze bune de transmitere, dar o parte a traficului (cea 25%) nu conţine date ci informaţii de control;
- **sincrone**, cu transmitere tot serială dar cu împărţirea grupurilor de biţi în cadre separate prin caractere de sincronizare pentru coordonarea transmisiei. Sunt mai scumpe şi mai complexe, de aceea nu sunt foarte accesibile utilizatorului obişnuit, în schimb au tehnici mai uşoare de detectare şi corectare a erorilor, care de multe ori se fac prin simpla retransmitere a cadrelor respective.

Principalul parametru prin care se identifică nivelul tehnologic al unui modem este viteza de transmitere a datelor, măsurată în *bps* (biţi într-o secundă), iar prin metode avansate de comprimare a datelor se pot asigura viteze de *76.800 bps*.

Verifică-ţi cunoştinţele:

1) Descrieţi funcţia modemului.
2) Clasificaţi tipurile modemurilor.
3) Care este principalul parametru de identificare a unui modem?

6.8. Intranetul

Prin **Intranet** se înţelege utilizarea **tehnologiilor Internet în vederea legării într-un tot unitar a resurselor informaţionale ale unei organizaţii**. Pe de altă parte, intranetul nu trebuie privit ca o entitate separată, ci integrat în cadru mai vast al realităţilor lumii actuale, a modelului pe care un organism comercial, non-profit, etc. îl urmează în vederea propriei sale organizări.

Intranet este o reţea proiectată pentru **prelucrarea informaţiilor în cadrul unei organizaţii sau al unei firme**. Printre serviciile pe care le oferă se numără distribuirea documentelor, a *soft*-ului, accesul la baze de date şi instruirea personalului. Intranetul implică de obicei aplicaţii asociate cu Internetul (paginile *Web*, *browser Web*, *site*-urile *FTP*, poşta electronică, grupurile de informare şi listele poştale), dar care sunt accesibile numai celor care activează în cadrul firmei sau organizaţiei respective [102].

În acest context intranetul propune un model, un mod tehnocratic de organizare a comunicaţiilor interumane.

Intranetul permite: sporirea eficacităţii în interiorul organizaţiei, reducerea activităţilor transpuse pe hârtie, reducerea costurilor, creşterea transparenţei în interiorul organizaţiei, oferirea, pentru orice activitate, a unui instrument ce conduce la competitivitate.

Caracteristicile intranetului sunt:

- este uşor de învăţat, utilizatorii beneficiind de experienţa acumulată prin folosirea reţelei Internet;
- este independent de localizarea utilizatorului;

- are un cost scăzut de menţinere şi întreţinere;
- facilitează comunicarea întregului personal al organizaţiei.

Intranetul este un concept flexibil, care nu are o reţetă de implementare prestabilită ce trebuie respectată pas cu pas. Din acest punct de vedere, intranetul poate integra oricare din serviciile Internet:

- *e-mail* (poşta electronică) - servicii de mesagerie internă;
- *web* - metodă de publicare a informaţiei în formă electronică;
- *FTP* (*File Transfer Protocol*) - transfer de fişiere;
- *IRC* (*Internet Relay Chat*) sau altă variantă de serviciu colaborativ - comunicarea sincronă (în direct, *on-line*) între două sau mai multe persoane;
- *mailing-lists* (liste de discuţii) - bazate pe serviciul de poştă electronică, facilitează trimiterea simultană a unui mesaj către un grup de destinatari;
- *newsgroups* (grupuri de ştiri) - difuzarea de ştiri pe diferite teme de interes etc.

Verifică-ţi cunoştinţele:

1) **Ce** se înţelege prin Intranet?
2) Enumeraţi caracteristicile Intranetului.

Întrebările pentru autoevaluare:

1. Descrieţi succint dispozitivele de interconectare a reţelelor.
2. Caracterizaţi repetoare, *hub*-uri, punţi, rutere.
3. Clasificaţi tipurile de comutare utilizate de un comutator.
4. Explicaţi modul de funcţionare a protocolului *STP*.
5. Enumeraţi funcţiile adaptorului *LAN* şi modemului.
6. Ce servicii propune Intranet?

Capitolul 7. Administrarea retelelor

7.1. Caracteristici ale ale retelelor client-server
7.2. Securitatea reţelei
7.2.1. Modelul de securitate
7.2.2. Modalităţi de protecţie a informaţiei
7.2.3. Categorii principale de atacuri asupra informaţii
7.2.4. Programe distructive

7.1. Caracteristici ale retelelor client-server

Reţelele client-server cu server de fişier (*file-server*) sunt formate din: un calculator ce **controleaza intreaga activitate a reţelei; calculatoarele** pe care lucrează utilizatorii, numite **staţii de lucru** (*workstation*).

Reţeaua de tip „client-server” foloseşte un calculator performant separat (*server*) - calculator „central”, care efectuează serviciile pentru mai mulţi utilizatori [103].

Calculatorul central oferă **răspunsuri rapide clienţilor, asigură cea mai bună protecţie a datelor din reţea şi foloseşte un sistem de operare avansat**. ***File server***-ul are rolul: de a gestiona intreaga activitate a reţelei; de a controla toate accesele la resursele comune (fişiere, imprimante, etc); de a asigura securitatea sistemului; de a realiza comunicaţia între staţii [104]. Pe *file-server* este **instalat sistemul de operare al reţelei**, care, de exemplu, la reţelele de tip *Novell* se numeşte *NetWare*, iar la reţelele *Microsoft* - *WindowsNT*. In principiu pot exista ***file-servere* dedicate**, care lucrează exclusiv pentru controlul reţelei şi ***file-servere* nededicate**, care pot lucra şi ca staţii de lucru. **Clienţii din reţea** sunt calculatoare conectate la server, puternice sau cu putere redusă la viteză de lucru, capacitate de memorie, etc. O reţea poate avea mai multe servere [105,106].

Staţiile de lucru sunt, de regulă, calculatoare personale obişnuite ce folosesc sistemele de operare: *Windows, Macintosh,*

UNIX, OS/2 sau *DOS* (*Disk Operating System*). Pentru utilizator modul de lucru al acestor staţii este asemănător celui din situaţia neconectării calculatorului la reţea. Periferia este constituită din **totalitatea echipamentelor comune pentru utilizatori** şi care sunt accesibile prin *file server* (imprimante, unităţi de disc, *plottere*). Elementele de conectare asigură comunicaţia între elementele reţelei.

Deoarece funcţia principală a unei reţele este partajarea de *hardware*, de programe şi diverse servicii de reţea, pentru a conecta un calculator la o reţea se utilizează **o placă specială de interfaţă**. La majoritatea reţelelor conectarea se face **prin cablu coaxial şi conectoare în forma literei „T”**. Tipurile de servicii partajate sunt: conectivitatea; stocarea şi regăsirea fişierelor; prelucrarea distribuită şi centralizată a datelor, imprimarea; securitatea de reţea; arhivarea, precum şi alte metode de protecţie a datelor; comunicarea între diverse birouri, departamente, etc.

Tipuri de utilizatori:

- **Utilizatorul obişnuit** - este persoana ce lucrează la o staţie de lucru in cadrul reţelei;

- **Managerul de grup** - este un utilizator cu drepturi suplimentare de gestionare a resurselor reţelei şi de control al utilizatorilor din subordine;

- **Administratorul** (Supervizorul) - este cel ce asigură funcţionarea a întregii reţele. El este responsabil de: menţinerea în parametri a reţelei; actualizările impuse de condiţiile noi apărute in timpul exploatării; asigurarea drepturilor de acces şi de lucru pentru fiecare utilizator şi păstrarea securităţii informaţiilor.

Verifică-ţi cunoştinţele:

1) Din ce sunt formate reţelele client-server cu server de fişier (*file-server*)?
2) Enumeraţi tipurile de utilizatori.

7.2. Securitatea reţelei

Securitatea reţelei nu se referă doar la securitatea pe care o asigură softul (drepturile utilizatorilor, parolele, etc.) ci şi securitatea fizică. Un astfel de document poate fi unul de bază pentru munca administratorului, pentru că descrie modul în care utilizatorii interacţionează cu reţeaua. Ameninţările la adresa securităţii unei reţele de calculatoare pot avea următoarele origini: dezastre sau calamităţi naturale, defectări ale echipamentelor, greşeli umane de operare sau manipulare, fraude [107, 108, 109].

7.2.1. Modelul de securitate pentru un calculator seamănă cu o „varză". Niveluri de securitate înconjoară subiectul ce trebuie protejat. Fiecare nivel izolează subiectul şi îl face mai greu de accesat în alt mod decât în cel în care a fost planificat.

1) **Securitatea fizică** reprezintă nivelul exterior al modelului de securitate. Problema cea mai mare o constituie **salvările pentru copii de rezervă ale datelor, programelor şi siguranţa păstrării suporţilor de salvare**. În aceste situaţii, reţelele locale sunt de mare ajutor: dacă toate fişierele schimbate frecvent rezidă pe un server, aceleaşi persoane (sigure şi de încredere), care lansează salvările pentru *mainframe*-uri, pot face acelaşi lucru şi la server.

Într-un sistem în care **prelucrarea este distribuită**, prima măsură de securitate fizică, care trebuie avută în vedere, este **prevenirea accesului la echipamente**. Pentru a învinge orice alte măsuri de securitate, trebuie să se dispună de acces fizic la echipamente. Acest lucru este comun tuturor sistemelor de calcul, distribuite sau nu.

2) **Securitatea logică** constă din acele metode care asigură controlul **accesului la resursele şi serviciile sistemului**. Ea are, la rândul ei, mai multe niveluri, împărţite în două grupe mari: niveluri de securitate a accesului (*SA*) şi niveluri de securitate a serviciilor (*SS*).

a) Securitatea accesului cuprinde:

– accesul la sistem (*AS*), care este răspunzător de a determina

dacă şi când reţeaua este accesibilă utilizatorilor. EI poate fi, de asemenea, răspunzător pentru decuplarea unei staţii, ca şi de gestiunea evidenţei accesului. *AS* execută, de asemenea, deconectarea forţată, dictată de supervizor. *AS* poate, de exemplu, să prevină conectarea în afara orelor de serviciu şi să întrerupă toate sesiunile, după un anumit timp;

– accesul la cont (*AC*), care verifică dacă utilizatorul care se conectează cu un anumit nume şi cu o parolă există şi are un profil utilizator valid;

– drepturile de acces (*DA*), care determină ce privilegii de conectare are utilizatorul (de exemplu, contul poate avea sesiuni care totalizează 4 ore pe zi sau contul poate utiliza doar staţia 27).

c) Securitatea serviciilor, care se află sub *SA*, controlează accesul la serviciile sistem, cum ar fi fire de aşteptare, *I/O* la disc şi gestiunea serverului.

Din acest nivel fac parte:

– controlul serviciilor (*CS*), care este responsabil cu funcţiile de avertizare şi de raportare a stării serviciilor; de asemenea, el activează şi dezactivează diferitele servicii;

– drepturile la servicii (*DS*), care determină exact cum foloseşte un anumit cont un serviciu dat; de exemplu, un cont poate avea numai dreptul de a adăuga fişiere la *spooler*-ul unei imprimante, dar are drepturi depline, de a adăuga şi şterge fişiere, pentru o altă imprimantă.

O dată stabilită conexiunea *SA* validează şi defineşte contul. Operaţiile ce trebuie executate sunt controlate de SS, care împiedică cererile ce nu sunt specificate în profilul utilizatorului.

Accesul intr-un sistem de securitate perfect trebuie să se facă prin aceste niveluri de securitate, de sus în jos. Orice sistem care vă lasă să evitaţi unul sau mai multe niveluri ale modelului de securitate implică riscul de a fi nesigur.

7.2.2. Modalităţi de protecţie a informaţiei. Exista **patru modalităţi de protecţie** a informaţiei din reţea:

- **securitatea la conectare** - aceasta metoda de protecţie se aplica tuturor utilizatorilor. La conectare utilizatorul îşi declară numele care i s-a asignat şi pe care îl recunoaşte file *server*-ul. Daca se introduce un nume incorect sau care nu există în evidenţa file *server*-ului se interzice accesul la reţea. De asemenea dacă există parola, numai introducerea ei corectă va permite conectarea; obligativitatea utilizării parolei este opţională;

- **securitatea prin drepturi ale utilizatorilor** - acest gen de securitate controlează posibilităţile de lucru ale utilizatorilor într-un directoriu dat.

- **securitatea prin drepturi permise în (sub)directoare** - acest tip de securitate se referă la drepturile permise tuturor utilizatorilor intr-un directoriu dat;

- **securitatea prin atributele fişierelor şi directoarelor** - se aplică la controlul modului în care un fişier poate fi modificat sau partajat între mai mulţi utilizatori.

Drepturile efective sunt acele drepturi pe care un utilizator le are într-un directoriu specificat. Ele sunt o intersecţie a drepturilor utilizatorului cu ale (sub)directorului respectiv. Atributele fişierelor şi directoarelor sunt mai puternice decât drepturile efective.

7.2.3. Categorii principale de atacuri asupra informaţii. Se disting două **categorii principale de atacuri** asupra informaţii: pasive şi active.

1) **Atacuri pasive** - sunt acelea în cadrul cărora intrusul observă informaţia ce trece prin „canal", fără să interfereze cu fluxul sau conţinutul mesajelor. Ca urmare, se face doar analiza traficului, prin citirea identităţii părţilor care comunică şi „analizând" lungimea şi frecvenţa mesajelor vehiculate pe un anumit canal logic, chiar dacă conţinutul acestora este neinteligibil. Atacurile pasive au următoarele caracteristici comune:

– Nu cauzează pagube (nu se şterg sau se modifică date);

– Încalcă regulile de confidenţialitate;

– Obiectivul este de a „asculta” datele schimbate prin reţea;

– Pot fi realizate printr-o varietate de metode, cum ar fi supravegherea legăturilor telefonice sau radio, exploatarea radiaţiilor electromagnetice emise, rutarea datelor prin noduri adiţionale mai puţin protejate.

2) **Atacuri active** - sunt acelea în care intrusul se angajează fie în furtul mesajelor, fie în modificarea, reluarea sau inserarea de mesaje false. Aceasta înseamnă că el poate şterge, întârzia sau modifica mesaje, poate să facă înserarea unor mesaje false sau vechi, poate schimba ordinea mesajelor, fie pe o anumită direcţie, fie pe ambele direcţii ale unui canal logic. Aceste atacuri sunt serioase deoarece modifică starea sistemelor de calcul, a datelor sau a sistemelor de comunicaţii. Există următoarele tipuri de ameninţări active:

– **Mascarada** - este un tip de atac în care o entitate pretinde a fi o altă entitate. De exemplu, un utilizator încearcă să se substitue altuia sau un serviciu pretinde a fi un alt serviciu, în intenţia de a lua date secrete (numărul cărţii de credit, parola sau cheia algoritmului de criptare). O „mascaradă” este însoţită, de regulă, de o altă ameninţare activă, cum ar fi înlocuirea sau modificarea mesajelor;

– **Reluarea** - se produce atunci când un mesaj sau o parte a acestuia este reluată (repetată), în intenţia de a produce un efect neautorizat. De exemplu, este posibilă reutilizarea informaţiei de autentificare a unui mesaj anterior. În conturile bancare, reluarea unităţilor de date implică dublări şi/sau alte modificări nereale ale valorii conturilor;

– **Modificarea mesajelor** - face ca datele mesajului să fie alterate prin modificare, inserare sau ştergere. Poate fi folosită pentru a se schimba beneficiarul unui credit în transferul electronic de fonduri sau pentru a modifica valoarea acelui credit. O altă utilizare

poate fi modificarea câmpului destinatar/expeditor al poştei electronice;

– **Refuzul serviciului** - se produce când o entitate nu izbuteşte să îndeplinească propria funcţie sau când face acţiuni care împiedică o altă entitate de la îndeplinirea propriei funcţii;

– **Repudierea serviciului** - se produce când o entitate refuză să recunoască un serviciu executat. Este evident că în aplicaţiile de transfer electronic de fonduri este important să se evite repudierea serviciului atât de către emiţător, cât şi de către destinatar.

În cazul atacurilor active se înscriu şi unele programe create cu scop distructiv şi care afectează, uneori esenţial, securitatea calculatoarelor. Există o terminologie care poate fi folosită pentru a prezenta diferitele posibilităţi de atac asupra unui sistem. Acest vocabular este bine popularizat de „poveştile" despre „hackeri". Atacurile presupun, în general, fie citirea informaţiilor neautorizate, fie (în cel mai frecvent caz) distrugerea parţială sau totală a datelor sau chiar a calculatoarelor.

7.2.4. Programe distructive. Dintre aceste **programe** amintim următoarele:

– **Viruşii** - reprezintă programe inserate în aplicaţii, care se multiplică singure în alte programe din spaţiul rezident de memorie sau de pe discuri; apoi, fie saturează complet spaţiul de memorie/disc şi blochează sistemul, fie, după un număr fixat de multiplicări, devin activi şi intră într-o fază distructivă (care este de regulă exponenţială);

– **Bomba *software*** - este o procedură sau parte de cod inclusă intr-o aplicaţie „normală", care este activată de un eveniment predefinit. Autorul bombei anunţă evenimentul, lăsând-o să „explodeze", adică să facă acţiunile distructive programate;

– **Viermii** - au efecte similare cu cele ale bombelor şi viruşilor. Principala diferenţă este aceea că nu rezidă la o locaţie fixă sau nu se duplică singuri. Se mută în permanenţă, ceea ce îi face dificil de

detectat. Cel mai renumit exemplu este Viermele INTERNET-ului, care a scos din funcţiune o parte din INTERNET în noiembrie 1988;

– **Trapele** - reprezintă accese speciale la sistem, care sunt rezervate în mod normal pentru proceduri de încărcare de la distanţă, întreţinere sau pentru dezvoltătorii unor aplicaţii. Ele permit însă accesul la sistem, eludând procedurile de identificare uzuale;

– **Calul Troian** - este o aplicaţie care are o funcţie de utilizare foarte cunoscută şi care, intr-un mod ascuns, îndeplineşte şi o altă funcţie. Nu creează copii. De exemplu, un *hacker* poate înlocui codul unui program normal de control „login" prin alt cod, care face acelaşi lucru, dar, adiţional, copiază într-un fişier numele şi parola pe care utilizatorul le tastează în procesul de autentificare. Ulterior, folosind acest fişier, *hacker*-ul va penetra foarte uşor sistemul.

Verifică-ţi cunoştinţele:

1) Ce se înţelege sub securitatea reţelei?
2) Cu ce seamănă modelul de securitate pentru un calculator?
3) În ce constă modalităţi de protecţie a informaţiei?
4) Enumeraţi categorii principale de atacuri asupra informaţii.

Întrebările pentru autoevaluare:

1. Descrieţi caracteristicile sistemelor de operare în LAN
2. Care modalităţi de protecţie a informaţiei cunoaşteţi?
3. Enumeraţi categoriile principale de atacuri asupra informaţiei
4. Cu ce scop se efectuează depanarea reţelei?

Referinţe bibliografice

1. Bolun, I.; Covalenco, I. Bazele informaticii aplicate (ediţia a II-a). Ch.: ASEM, 2001. 615 p.
2. Plohotniuc, E. Informatica generală. Bălţi: US „A.Russo”, 2001, 304 p.
3. Олифер, В. Г.; Олифер, Н. А. Компьютерные сети. Принципы, технологии, протоколы. Учебник для вузов (4-е изд.). СПб.: Питер, 2010, 944 с.
4. Таненбаум, Э.; Уэзеролл, Д. Компьютерные сети (5-е изд.). СПб.: Питер, 2012, 960 с.
5. Gremalschi, A.; Gremalschi, L.; Mocanu, Iu. Informatică. Manual pentru clasa a 10-a. Ch: Ştiinţa, 2007, 188 p.
6. Gremalschi, A.; Vasilache, Gr.; Gremalschi, L. Informatica. Manual pentru clasa a 7-a. Ch: Ştiinţa, 2001, 134 p.

Referinţe din Internet

7. http://facultate.regielive.ro/cursuri/retele/retele-15108.html
8. http://shannon.etc.upt.ro/laboratoare/pc/luc5/iso-osi.html
9. http://www.quietroot.org/servicii/retele-de calculatoare/ clasificare -retele-calculatoare.html
10. http://ro.wikipedia.org/wiki/Re%C8%9Bea_de_calculatoare
11. Cătălina Lehănceanu, Cristian Orban, Octavian Purdilă, Răzvan Rughiniş. Reţele locale de calculatoare. Ghid de laborator. http://www.cs.jhu.edu/~ralucam/ralucam-cv-july-2009.pdf
12. http://www.infoiasi.ro/~busaco/
13. http://ro.wikipedia.org/wiki/Protocol_de_re%C8%9Bea
14. http://www.uab.ro/cursuri.../curs_modul2.doc
15. http://www.scientia.ro/tehnologie/34-cum-functioneaza-calculatorul/419-cum-functioneaza-osi-modelul-de-interconectare-a-sistemelor-deschise.html
16. http://facultate.regielive.ro/laboratoare/retele/modelul-de-referinta-osi-290276.html
17. http://www.et.upt.ro/admin/tmpfile/fileN1288535592file4ccd7e28852cc.ppt

18. http://www.gsastl.ro/teste/modeltcp/tcpip.files/frame.htm
19. http://www.gsastl.ro/teste/osi/osi.html
20. http://lucicap.3x.ro/nivfizic.htm
21. http://ro.wikipedia.org/wiki/Nivelul_Fizic
22. http://www.cs.ucv.ro/staff/dmancas/HSN.doc
23. http://www.et.upt.ro/admin/tmpfile/fileQ1288535374file4ccd7d4ebcdb5.ppt
24. http://ro.wikipedia.org/wiki/Sistem_digital
25. http://ro.wikipedia.org/wiki/Transmisiune_digitală
26. http://telecom.etc.tuiasi.ro/telecom/staff/vlcehan/discipline predate/rrcs/RRCS_cap 3.pdf
27. http://www.petrerau.inforapart.eu/Dictionar de acronime in informatica.pdf
28. http://facultate.regielive.ro/cursuri/calculatoare/ retele-de-calculatoare-nivel-fizic-244925.html
29. http://www.slideshare.net/mrrbiz/global-broadband-aggregation-equipment-market-20112015industry-sizetrendsanalysis-and-forecast-20112015
30. http://ro.wikipedia.org/wiki/Bandă_largă
31. http://en.wikipedia.org/wiki/Baseband
32. http://cndiptfsetic.tvet.ro/materiale/Materiale_de_predare/ML/06_Medii de transmisie II.doc
33. http://rf-opto.etc.tuiasi.ro/docs/rc/RETELE DE CALCULATOARE_2.pdfrast. orgfree.com/retele/retele_2001_05.html
34. http://facultate.regielive.ro/referate/electronica/ medii-de-transmisie-in-retelele-de-comunicatii-212298.html
35. http://facultate.regielive.ro/referate/calculatoare/cabluri-de-retea-coaxial-torsadat-in-pereche-optic-239720.html
36. http:/www.scribd.com/doc/4532428
37. http:/ro.wikipedia.org/wiki/Cablu_coaxial
38. http:/facultate.regielive.ro/referate/electrotehnica/cablul-coaxial-203536.html

39. http://www.etc.ugal.ro/imunteanu/Csemnale_bmk.pdf
40. http:/ro.wikipedia.org/wiki/Fibră_optică
41. http:/ro.wikipedia.org/wiki/Cablu_de_fibră_optică
42. http:/ro.wikipedia.org/wiki/Rețea_fără_fir
43. http:/www.megasound.ro/Sisteme-Fara-Fir-wireless.html
44. ro.wikipedia.org/wiki/Modelul_OSI
45. http://www.competentedigitale.ro/internet/internet_TCP_IP.html
46. www.cs.ucv.ro/staff/dmancas/rmtw.doc
47. http://www.scritube.com/stiinta/informatica/Retele-de-calculatoare626181015.php
48. http://ind3c3ntul.3x.ro/
49. http://stuffyspace.blogspot.com/2012/11/retele-locale-de-calculatoare-local.html
50. http://shannon.etc.upt.ro/laboratoare/pc/luc5/adrese_MAC.html
51. http://www.angelfire.com/mi4/sim_brd0/ro.wikipedia.org/wiki/Ethernet
52. http://upgradeteam.myforum.ro/nivelul-legatura-de-date-vp67.html?highlight=
53. http://www.kulituranauka.com/pedagogie-2/material-didactic-retele-de-calculatoare
54. http://ru.scribd.com/doc/49267977/retele
55. ro.wikipedia.org/wiki/Wi-Fi
56. http://ciobanu.freehosting.md/articole_view.php?id=1
57. http://www.etc.ugal.ro/imunteanu/Csemnale_bmk.pdf
58. ro.wikipedia.org/wiki/Comutație_de_circuite
59. ro.wikipedia.org/wiki/Comutație_de_pachete
60. http://www.sursa.md/product_info.php/info/p901_Tehnologia-regimului-asincron-de-transmitere-a-informa%C5%A3iei-ATM.html
61. www.cs.ucv.ro/staff/dmancas/HSN.doc
62. www.uab.ro/cursuri/curs_modul3.doc
63. http://key.md/blog/22-definitia-de-functionare-a-unei-retele-client-server.html
64. http://ru.wikipedia.org/wiki/5G
65. www.bluetooth.com/

66. ro.wikipedia.org/wiki/Releu_de_cadre
67. ro.wikipedia.org/wiki/Internet
68. ro.wikipedia.org/wiki/TCP/IP
69. http://shannon.etc.upt.ro/laboratoare/pc/luc6/3_Nivelul_3.htm
70. http://shannon.etc.upt.ro/laboratoare/pc/luc6/3_Nivelul_3.htm
71. ro.wikipedia.org/wiki/Ruter
72. profs.info.uaic.ro/~busaco/teach/.../net4.pdf
73. ro.wikipedia.org/wiki/Protocol_Internet
74. www.sursa.md › Informatica
75. http://www.scritube.com/stiinta/informatica/Retele-de-calculatoare21423.php
76. ro.wikipedia.org/wiki/Adresă_IP
77. http://www.scritube.com/stiinta/informatica/retele/Clase-de-adrese-IP34477.php
78. http://facultate.regielive.ro/laboratoare/calculatoare/adresarea-ip-automatica-inteligenta-artificiala-184.html
79. ro.wikipedia.org/wiki/Adresă_IP
80. stud.usv.ro/TMC/lab4/
81. http://shannon.etc.upt.ro/laboratoare/pc/luc7/teorie.html
82. ro.wikipedia.org/wiki/TCP/IP
83. lucicap.3x.ro/nivtransport.htm
84. http://ro.wikipedia.org/wiki/Telnet
85. http://ru.scribd.com/doc/57507533/silabus-retele-tomai-2008
86. ro.wikipedia.org/wiki/World_Wide_Web
87. ro.wikipedia.org/wiki/Hipertext
88. http://windows.microsoft.com/ro-RO/windows-vista/File-Transfer-Protocol-FTP-frequently-asked-questions
89. http://www.afaceri.net/articole/diverse/outlook_expres/posta_electronica.htm
90. ro.wikipedia.org/wiki/E-mail
91. realpunk.3x.ro/html/net.html
92. emsc.ub.ro/docs/1109_RC.pdf
93. www.inforetele.com/tag/tabela-rutare/

94. http://www.scrigroup.com/calculatoare/retele-calculatoare/Componentele-unei-retele-LAN-w74786.php
95. ro.wikipedia.org/wiki/Comutator
96. http://www.scritube.com/stiinta/informatica/ARHITECTURI-DE-REEA101232075.php
97. ro.wikipedia.org/wiki/Ruter
98. http://windows.microsoft.com/ro-RO/windows7/How-do-hubs-switches-routers-and-access-points-differ
99. ro.wikipedia.org/wiki/Placă_de_reţea
100. http://www.unibuc.ro/prof/niculae_c_m/telecom/transm_ctrl_prot_tcp.htm
101. ro.wikipedia.org/wiki/Modem
102. ro.saferpedia.eu/wiki/Intranet
103. http://www.marksistem.md/index.php/ro/servicii/administrare-retele.html
104. http://platniucnik.blogspot.com/p/cele-mai-cunoscute-sisteme-de-operare.html
105. http://www.electricats.ro/electricats/relatii/racordare/racordare.html
106. http://ro.wikipedia.org/wiki/Server_web
107. http://ro.wikipedia.org/wiki/Securitatea_re%C8%9Belelor_de_calculatoare
108. http://ru.scribd.com/doc/25154918/Securitatea-Retelei
109. http://www.securitatea-informatica.ro/securitatea-informatica/securitatea-retelelor/securitatea-retelelor/
110. http://accesinform.ucoz.ru/_ld/0/32_Informatica_Gim.pdf
111. http://www.ctice.md/downloads/Curriculum%20pentru%20disciplina%20INFORMATICA%20%28X-XII%29.pdf

Anexa 1

Descrierea succintă a unor protocoale folosite în reţele de calculatoare

Nr	Protocol		Nivel *OSI*	Descriere
	Abre-viere	**Denumire (limba engleză)**		
1.	*ADSL*	*Asymmetric Digital Subscriber Line*	2	O tehnologie care permite transmiterea asimetrică de date digitale, pe linie telefonică de cupru, mai rapid (1,5-8 Mbps) decât un modem *voiceband* convenţional. Acest lucru este posibil prin folosirea frecvenţelor care nu sunt utilizate de semnalul vocal digitalizat. Un microfiltru permite ca o conexiune de telefon unică să poată fi utilizată atât pentru transmisii de date cât şi pentru apeluri vocale, în acelaşi timp. *ADSL* funcţionează doar pe distanţe scurte, de obicei mai puţin de 4 km, în funcţie de grosimea firului de cupru.
2.	*AFP*	*Apple Filing Protocol*	6+7	Protocolul traduce comenzile de manipulare de fişiere trimise de la un computer local la un server. Acest lucru permite computerului local de a rula fişiere cu comenzile proprii, mai degrabă decât achiziţionarea de comenzi de server.
3.	*ARP*	*Address Resolution Protocol*	2+3	Protocolul este proiectat pentru a determina MAC-adresa dacă este cunoscută adresa *IP*.
4.	*ASN.1*	*Abstract Syntax Notation 1*	6	Este un limbaj pentru a descrie structura de obiecte care nu depind de maşină de date şi pentru a reprezenta, a codifica, a transmite şi a decodifica datele în telecomunicaţii şi reţele de calculator.
5.	*ASP*	*Apple talk*	4+5	Protocolul asigură livrarea sigură a

		Session Protocol		datelor şi oferă acces la serviciile de transport prin protocolul *ATP*
6.	*ATM*	*Asynchronous Transfer Mode*	2+3 +4+ 5+6 +7	Este un protocol de bază folosit prin magistrala *SONET/SDH* din standardul *ISDN* (*Reţea de servicii digitale integrate*). Mod de transfer asincron este o tehnică (un protocol) de comutaţie temporală asincronă, de mare viteză, orientată pe conexiune şi bazată pe circuite virtuale, folosită în transportul traficului de reţea. Deoarece reţelele *ATM* sunt orientate pe conexiune, transmisiunea datelor necesită mai întâi transmisia unui pachet de la un capăt la altul pentru iniţializarea conexiunii – aşa numitul circuit virtual. Însă majoritatea reţelelor *ATM* suportă şi circuite virtuale permanente, de genul liniilor închiriate din domeniul telecomuncaţiilor.
7.	*BGP*	*Border Gateway Protocol*	3	Este protocolul de rutare folosit în nucleul Internetului. El menține o tabelă cu rețele *IP* (sau «prefixe») care arată calea folosită pentru a ajunge la rețeaua respectivă prin diferitele sisteme autonome (*AS*). *BGP* este considerat din acest motiv un protocol de rutare vector-cale (spre deosebire de protocoalele vector-distanță, care nu păstrează toată calea).

8.	*CHAP*	*Challenge Handshake Authentication Protocol*	2	Protocolul este folosit atât la stabilirea conexiunii cât şi după aceea, periodic, după un timp aleator, pentru a verifica identitatea clientului. Serverul trimite clientului un mesaj de „încercare” numit „*challange*”. Clientul preia acest mesaj şi parola configurată, trimite un răspuns serverului. Serverul calculează un răspuns pe baza mesajului trimis şi a parolei pe care o are configurată şi compară rezultatul cu răspunsul primit de la client. Dacă mesajele coincid, înseamnă că parola pe care a folosit-o clientul pentru a genera răspunsul este identică cu parola folosită de server pentru verificare, deci identitatea clientului este verificată şi se stabileşte conexiunea. Pentru a fi sigur că la celălalt capăt se află mereu clientul autentificat iniţial, serverul trimite din când în când astfel de mesaje de *challenge* şi procedura explicată mai sus se repetă.
9.	*DSL*	*Digital subscriber line*	2	Familia de tehnologii de transmisie digitală a datelor
10.	*EIGRP*	*Enhanced Interior Gateway Routing Protocol*	3	Este un protocol de rutare bazat pe principiul distanţei vectoriale, si constă dintr-un schimb de informaţii cu celelalte rutere din reţea, legat cu un proces intern de stocare a datelor primite de la acestea, incluzând detaliile bazate pe caracteristicile calitative ale rutelor raportate, pe baza căror informaţii se va lua decizia de alegere a rutei spre o anumită destinație.

11.	FDDI	Fiber Distributed Data Interface	2	Este un tip de reţea *LAN* (sau *MAN* cu mai multe reţele *LAN*), cu viteza de *100 Mbit/s* pe fibră optică (care îi permite să ajungă la o distanţă maximă de 200 km). Aceasta este de fapt o pereche de inele (unul este numit primar, celălalt pentru a prinde greşelile de la prima, se numeşte secundar). *FDDI* este un protocol care utilizează un *token ring* de detectare şi corectare a erorilor (în cazul în care acest lucru este inelul secundar devine importantă). *Token*-ul circula între maşine cu o viteză foarte mare. Dacă nu reuşeşte, după un anumit timp aparatul constată că a existat o eroare la reţea.
12.	FR	Frame Relay	2	Comutarea cu relee de cadre este o tehnică eficientă de transmisie a datelor folosită pentru a trimite informaţie digitală în mod rapid şi ieftin de la una sau mai multe surse la unul sau mai multe puncte finale (destinaţii). Este implementată în mod obişnuit drept o tehnică de încapsulare pentru voce şi date şi este folosită între reţele locale (*LAN*), peste o reţea de arie largă (*WAN*). Fiecare utilizator primeşte o linie privată (sau linie închiriată) către un *releu de cadre*. Reţeaua *releu de cadre* se ocupă de transmisia datelor pe o cale care se schimbă în mod frecvent şi care este transparentă (invizibilă) pentru utilizatorii finali.

13.	*FTP*	*File Transfer Protocol*	7	Este un protocol utilizat pentru a transfera fişiere aflate pe un host la alt host printr-o reţea bazată pe TCP, cum este Internetul. Protocolul este orientat pe conexiune şi construit pe o arhitectură client-server.
14.	*HDLC*	*High-level Data Link Control*	2	Este un protocol de nivel 2 Modelului *OSI*. Scopul său este de a defini un mecanism de a delimita diferite tipuri de cadre, adăugând verificarea erorilor.
15.	*HTTP*	*Hyper Text Transfer Protocol*	7	Este metoda cea mai des utilizată pentru accesarea informaţiilor în Internet care sunt păstrate pe servere *World Wide Web* (*WWW*). Protocolul *HTTP* este un protocol de tip text, fiind protocolul «implicit» al *WWW*. Adică, dacă un *URL* nu conţine partea de protocol, aceasta se consideră ca fiind *http*. *HTTP* presupune că pe calculatorul destinaţie rulează un program care înţelege protocolul. Fişierul trimis la destinaţie poate fi un document *HTML* (abreviaţie de la ***H****yper* ***T****ext* ***M****arkup* ***L****anguage*), un fişier grafic, de sunet, animaţie sau video, de asemenea un program executabil pe *server*-ul respectiv sau şi un editor de text. După clasificarea după modelul de referinţă *OSI*, protocolul *HTTP* este un protocol de nivel aplicaţie.
16.	*GSM*	*Global System for Mobile Communica-tions*	2	Reprezintă standardul de telefonie mobilă (celulară) cel mai răspândit din lume. Atributul „mobil” al multor aparate şi dispozitive actuale se referă în primul rând la conectivitatea lor (fără fir, prin semnale radio) la sistemul *GSM*, practic din orice punct de pe glob. Mai este cunoscut şi sub denumirea de *2G*. Este sistemul dominant în Europa.

17.	*ICMP*	*Internet Control Message Protocol*	3	Este un protocol din suita *TCP/IP* care foloseşte la semnalizarea şi diagnosticarea problemelor din reţea. Mesajele *ICMP* sunt încapsulate în interiorul pachetelor *IP*.
18.	*IGMP*	*Internet Group Management Protocol*	3	IGMP este un protocol asimetric, în sensul că un comportament specificat pentru gazde diferă de cea de *routere multicast*. *IGMP* este un protocol de executat între maşinile gazdă pe aceeaşi subreţea şi routere multicast în această subreţea. Routere menţine o listă a grupurilor de multicast pentru care maşinile lor de gazdă raportate a fi subscris. Ceea ce permite routerul pentru a determina pachete IP multicast la releul de pe aceste subrețele.
19.	*IGRP*	*Interior Gateway Router Protocol*	3	Este un tip de protocolul de rutare *classful*, se utilizează pentru a face schimb de tabelele de rutare într-un sistem autonom. *IGRP* permite valori multiple pentru fiecare rută, inclusiv de lăţime de bandă, de încărcare, întârziere şi fiabilitatea. Pentru a compara două rute aceste valori sunt combinate într-o singură, folosind o formulă reglabil. Numărul maxim de „*hop*” pentru pachetele de a fi rutate în *IGRP* este de 255.
20.	*IP*	*Internet Protocol*	3	Este o metodă sau un protocol prin care datele sunt trimise de la un calculator la altul prin intermediu Internetului.
21.	*IPX*	*Internetwork Packet eXchange*	3	Este un protocol de reţea bazat pe datagrame şi lipsit de conexiuni. Pachetele *IPX* sunt trimise către destinaţiile lor, dar nu se garantează şi nici nu se verifică dacă acestea ajung sau nu la destinație.

22.	*ISDN*	*Integrated Services Digital Network*	1	*ISDN* este un set de protocoale folosite atât pentru stabilirea şi întreruperea conexiunilor telefonice, cât şi pentru alte funcţii complexe cum ar fi videoconferinţe, *Telex* sau *Teletex*.
23.	*ISM*	*Industrial, Scientific and Medical radio bands*	2	Benzi de radio industrială, ştiinţifică şi medicală
24.	*LCP*	*Link Control Protocol*	2+5 +6	Protocol care se utilizează cu protocolul PPP, pentru stabilirea conexiunii punct la punct.
25.	*NCP*	*Network Control Protocol*	2+5 +6	Protocol de control al reţelei care se utilizează cu protocolul PPP
26.	*NFS*	*Network File System*	5	Un protocol de reţea de gestionare a sistemului de fişiere.
27.	*OSPF*	*Open Shortest Path First*	3	Este un protocol *IP* dinamic destinat rutării în interiorul unui reţele mari (guvernate de un singur gestionar). *OSPF* este bazat pe caracteristicile conexiunilor dintre interfeţe. Caracteristic pentru *OSPF* este baza de date cuprinzând link-urile spre ruterele adiacente. Aceasta cuprinde o listă a tuturor ruterelor conectate direct – constituind „miezul topologiei reţelei". Pentru a menține actuală baza de date corespunzatoare topologiei este necesar un schimb permanent de informație între routere.
28.	*PAP*	*Password Authenti-cation Protocol*	2	Clientul (*dial-up* sau ruter) trimite combinaţia user/parolă, necriptate, până când serverul îl acceptă (dacă combinaţia e corectă) sau până când conexiunea se închide (dacă combinaţia nu e bună). Este suportat de *PPP*.

29.	*POP*	*Post Office Protocol*	7	Protocolul *POP* este utilizat pentru a permite unei staţii de lucru să primească poşta electronică pe care serverul o stochează.
30.	*PPP*	*Point-to-Point Protocol*	2	Este un protocol folosit pentru a încapsula date pe interfeţele seriale sincrone. Prezintă numeroase avantaje faţa de alte încapsulări existente, dintre care menţionăm: - Este standardizată şi implementată la fel de toţi producătorii de echipamente; - Permite folosirea pe acelaşi ruter a mai multor protocoale de nivel 3; - Poate fi folosită pe interfeţele seriale sincrone, pe cele asincrone (atunci când facem *dial-up* folosind un modem), şi pe interfeţe *ISDN*; - Este posibilă autentificarea.
31.	*RARP*	*Reverse Address Resolution Protocol*	2+3	*RARP* permite de a determina adresa *IP* a unei masini dacă este cunoscută adresa de *hardware*-ul (adresa *MAC*). În rezumat, *RARP* este opus *ARP*. Protocolul *RARP* este folosit pentru determinarea adreselor logice a staţii de lucru.
32.	*RIP*	*Router IP*	3	Acest protocol este utilizat pentru rutele de reţea relativ mici şi relativ omogene şi se caracterizează printr-un vector distanţa până la destinaţie. Se presupune că fiecare *router*-ul este punctul de plecare al mai multor trasee la reţelele la care se referă. Descrierile acestor rute sunt stocate într-un tabel de rutare. Tabelul de rutare *RIP* conţine intrări pentru fiecare maşina de service. Intrare ar trebui să includă: a) *IP*-adresa de destinaţie; b) metrica traseului (de la 1 până la 15, numărul de paşi până la locul de destinaţie); c) adresa *IP* a *router*-ului

				cel mai apropiat (poarta de acces), pe drumul spre destinaţie; d) *timer*-ul rutei.
33.	*RPC*	*Remote Procedure Call*	5	Este o specificaţie simplă şi un set de coduri care permite a efectua apeluri printr-o reţea în medii diferite, *RPC* se utilizează pentru a invoca procedare de pe un server de la distanţă în orice sistem (*Windows, Mac OS X, GNU / Linux*), şi cu orice limbaj de programare. Acest lucru oferă un serviciu de web fără restricţii de sistem sau de limbă. Procesul de invocare de la distanţă foloseşte protocolul *HTTP* pentru a transfera date şi standard de *XML* pentru structurarea datelor.
34.	*RTP*	*Real-time Transport Protocol*	7	Este un protocol prin intermediul căruia se pot transmite informaţii de tip mulitimedia (sunete, imagini) printr-o reţea de telecomunicaţii. Aplicaţiile multimedia pun condiţii foarte dure asupra ambianţei de transmitere. Aplicaţiile de obicei folosesc *RTP* implementat peste *UDP*, pentru ca să se poată folosi de posibilitatea sa de multiplexare şi controlul checksum. Dar *RTP* se poate folosi de asemenea şi deasupra oricărui protocol de nivel 4 *OSI*. *RTP* permite transmiterea concomitentă pe adrese diferite, dacă multicastul este posibil la nivel de reţea.Trebuie de luat în consideraţie că *RTP* nu garantează transmiterea la timp a pachetelor şi nu oferă garanţia integrităţii transmiterii datelor.
35.	*RTSP*	*Real Time Streaming Protocol,*	7	Este un protocol de comunicare pentru recepţia datelor sistemelor de mass-media. Se va controla un server mass-media de la distanţă, oferind caracteristici tipice ale unui *player* video, cum ar fi „citire” şi „pauză” şi să

				permită accesul pe poziţia temporală. *RTSP* nu transportă datele şi trebuie să fie ataşat la un protocol de transport, cum ar fi *RTP* pentru această sarcină.
36.	*SCTP*	*Stream Control Transmissio n Protocol*	4	Ca un protocol de transport, *SCTP* este într-un sens echivalent cu *TCP* sau *UDP* oferind servicii similare pentru *TCP*, asigurând fiabilitate, secvenţe reordonate, şi controlul congestiei. În timp ce *TCP* este orientat pe octeţi, *SCTP* gestionează „cadre”. Iniţial, *SCTP* a fost conceput pentru a transporta protocoale de voce peste *IP* (*ISDN* partea utilizatorului, *SMS*-uri). Un avantaj reprezintă abilitate a *SCTP* de comunicare multi-ţintă, în cazul în care un capăt (sau ambele) de conectare constă din mai multe adrese *IP*.
37.	*SDH*	*Synchronous Digital Hierarchy*	2	Ierarhia digitală sincronă - este o interfață de transmisiune optică la debit înalt, folosită de operatorii telecom din Europa pentru multiplexarea unor fluxuri de ordin imediat inferior. Debitul digital este transferat folosind lasere şi diode electroluminiscente (*LED*-uri). În SDH cadrul bazic este *STM*-1 (modul de transport sincron 1), cu o viteză de *155 Mbps*.
38.	*SIP*	*Session Initiation Protocol*	7	Protocolul de iniţializare a sesiunii. Astfel de sesiuni includ apeluri telefonice prin Internet, sesiuni multimedia, conferinţe multimedia. *SIP* are următoarele caracteristici: a) Independența de nivelul de transport, putând fi folosit cu *UDP*, *TCP*, *ATM;* b) Bazat pe mesaje de tip text.
39.	*SMB*	*Server Message Block*	5+6	Este un protocol pentru partajarea de resurse (fişiere şi imprimante), pe *LAN*-uri cu *PC*-urile *Windows*. *SMB* funcţionează printr-o structură de client

				/ server, clientul va trimite întrebări specifice şi server de fişiere va răspunde. *SMB* server poate oferi clienţilor acces la reţea pentru sisteme de fişiere şi alte resurse. Clientul poate avea evidenţele sale proprii, care nu sunt comune şi pot accesa simultan discuri partajate şi imprimante de pe server.
40.	*SMTP*	*Simple Mail Transfer Protocol*	7	Protocolul simplu de transfer al corespondenţei este un protocol, folosit pentru transmiterea mesajelor în format electronic pe Internet. Protocolul *SMTP* specifică modul în care mesajele de poştă electronică sunt transferate între procese *SMTP* aflate pe sisteme diferite. Procesul *SMTP* care are de transmis un mesaj este numit client *SMTP*, iar procesul *SMTP* care primeşte mesajul este serverul *SMTP*. Protocolul nu se referă la modul în care mesajul ce trebuie transmis este trecut de la utilizator către clientul, sau cum mesajul recepţionat de serverul este livrat utilizatorului destinatar şi nici cum este memorat mesajul sau de câte ori clientul încearcă să transmită mesajul.
41.	*SNMP*	*Simple Network Management Protocol*	7	Este un protocol de comunicare care permite administratorilor de reţea să gestioneze dispozitivele de reţea, monitorizarea şi diagnosticarea problemelor de reţea şi *hardware* de la distanţă. Comutatoare, *routere*, *hub*-uri, staţii de lucru şi servere (fizic sau virtual) sunt exemple de echipamente care conţin obiecte de gestionat. Aceste obiecte pot fi informaţii despre *hardware* de gestionat, setările de configurare, statisticile de performanţă şi alte obiecte care sunt direct legate de

				comportamentul de echipamente de curent.
42.	*SONET*	*Synchronous Optical NETwork*	2	Sistema (sistemul American) este proiectată pentru medii bazate pe fibră optică, poate fi implementată şi în cazul firelor de cupru. Oferă lăţimi de bandă de la *51,84Mbps* la *9952Mbps*. SONET oferă standarde pentru debite de linie de până la *39,808 Gbps.*
43.	*SPX*	*Sequenced Packet eXchange*	4	Este un protocol de reţea la sistemul de operare *Novell NetWare* folosit pentru a controla furnizarea de date într-o reţea locală (şi într-o măsură mai mică, *WAN*). Împreună cu *IPX* (*Novell*, de asemenea) stiva de *IPX / SPX* este similar cu *TCP / IP*. Protocolul este responsabil pentru asigurarea integrităţii de pachete trimise şi pachete de confirmare primite. Efectuează un control al fluxului de date, controlul vitezei pachetelor trimise şi primite, precum şi reducerea riscului de corupţie. Atunci când o eroare este descoperit într-un pachet care este trimis sau primit, toate pachetele trimise sau primite în această perioadă sunt marcate la fel de rău şi de *re-broadcast*.
44.	*SSH*	*Secure SHell*	7	Este un protocol ce permite ca datele să fie transferate folosind un canal securizat intre dispozitive de reţea. Folosit cu precădere în sistemele de operare multiutilizator *Linux* şi *Unix*, *SSH* a fost dezvoltat ca un înlocuitor al *Telnet*-ului şi al altor protocoale nesigure de acces de la distanţă, care trimit informaţia, în special parola, în clartext, făcând posibilă descoperirea ei prin analiza traficului. Criptarea folosita de *SSH* intenționează să asigure

				confidenţialitatea şi integritatea datelor transmise printr-o reţea nesigură cum este Internetul.
45.	*TCP*	*Transmission Control Protocol*	4	Reprezintă un protocol de comunicaţie de nivel înalt care oferă transmiterea sigură a unui şir de biţi de la un program rulând pe un calculator către un program rulând pe alt calculator.
46.	*Telnet*	*Terminale virtuale*	7	*Telnet* este un protocol de reţea care se foloseşte în Internet şi în reţele de calculatoare tip *LAN* la comunicaţia textuală, bidirecţională şi interactivă.
47.	*TLS*	*Transport Layer Security*	5	Eeste un acronim care reprezintă un protocol web pentru a transmite fără risc documente private prin Internet. Pentru a cripta datele *TLS* utilizează un sistem criptografic cu două chei: una publică, cunoscută de oricine, şi una privată, secretă, cunoscută numai de destinatarul mesajului.
48.	*UDP*	*User Datagram Protocol*	4	Este un protocol fără conexiune, semnalarea erorilor sau reluărilor fiind asigurată de nivelul superior, iar datele transmise nu sunt segmentate. *UDP* este folosit în situaţiile în care eficienţa şi viteza transmisiei sunt mai importante decât corectitudinea datelor.
49.	*Whois*	*Who is*	6	«Cine e» - este un protocol de căutare/răspuns, larg folosit pentru a căuta în baza de date oficială pentru a determina deţinătorul unui domeniu, adresă IP sau a unui număr de sistem autonom din internet. Căutările se făceau, tradiţional, printr-o interfaţă command line, dar acum există mai multe site-uri web ce oferă moduri mai simple de căutare prin bazele de date. *WHOIS* rulează pe portul 43, protocolul *TCP*.

50.	*X.25*	*Packet Switching*	2+3	Este un protocol pentru *WAN*, care defineşte felul în care conexiunile între dispozitivele utilizate şi dispozitivele de reţea sunt stabilite. *X.25* este conceput pentru a se putea opera indiferent de tipul sistemelor conectate la reţea.
51.	*XDR*	*eXternal Data Representation*	6	Este un standard de reprezentarea externă (codificare/decodare) a datelor; este implementat ca o bibliotecă de *software* de funcţii, care este portabilă între diferite sisteme de operare şi este, de asemenea, independentă de nivelul transport.
52.	*XMPP*	*eXtensible Messaging and Presence Protocol*	7	Este un protocol pentru a face schimb de informaţii structurate în timp real între oricare două noduri de reţea.

Anexa 2

Descrierea unor tipuri de adaptoare

Adaptoare *LANtastic*. *Artisoft* produce adaptoare *Ethernet*, cât şi propriile plăci adaptoare pentru reţele, modelul brevetat al firmei numindu-se adaptor *LANtastic*, produs confundat deseori cu sistemul de operare în reţea cu acelaşi nume, produs de aceeaşi firmă. Adaptorul *LANtastic* operează la rata de *2Mbps* şi foloseşte un cablu cu patru conductori ce leagă toate staţiile. Instalarea este uşoară dacă acest cablu nu trebuie să treacă prin pereţi sau tavan.

Adaptoare *ARCnet*. Unul dintre cele mai vechi tipuri de hard pentru *LAN* este *ARCnet*. Iniţial a fost o schemă brevetată a firmei *Datapoint Corporation*, însă acum mai multe companii produc plăci compatibile *ARCnet*. Adaptorul *ARCnet* este mai lent, dar ignoră micile erori de instalare. Oferă siguranţă în funcţionare, iar problemele cablurilor şi ale adaptorului *ARCnet* sunt uşor de diagnosticat. Totodată, este mai ieftin decăt *Ethernet*. Funcţionează oarecum ca Token Ring, dar la rata mai mică de *2,5Mbps*.

Adaptoare Ethernet. Oferă posibilitatea interconectării unei mari varietăţi de echipamente, inclusiv calculatoare *UNIX, Apple, IBM PC* şi clone *IBM*. Există foarte mulţi producători ai acestor plăci. *Ethernet* este livrat în trei variante (*ThinNet, UTP* şi *ThickNet*), în funcţie de lungimea cablurilor folosite. *Ethernet* operează cu o rată de transfer a datelor de *10Mbps*. Între transferurile de date (cereri şi răspunsuri la şi de la file server) reţelele *Ethernet* rămân tăcute. După ce o staţie de lucru trimite o cerere prin cablu *LAN*, cablul rămâne din nou tăcut. Când însă două staţii sau mai multe (şi/sau file servere) încearcă să folosească reţeaua în acelaşi timp, apare o coliziune, datorită faptului că numai două calculatoare pot comunica prin cablu la un moment dat. În asemenea caz, ambele calculatoare renunţă, după care încearcă din nou. Pentru a detecta o coliziune, adaptoarele de reţea *Ethernet* folosesc metoda *CSMA/CD* (*Carrier Sense, Multiple Access/Collision Detection*) şi fiecare renunţă o perioadă aleatoare de timp. Această metodă oferă efectiv posibilitatea unui calculator să fie primul. La un trafic mai mare, frecvenţa coliziunilor creşte, iar timpii de răspuns devin tot mai nesatisfăcători, reţeaua putând ajunge să consume mai mult timp pentru revenirea din coliziuni decât transmiţând date. Reţeaua *Token Ring*, proiectată de firmele *IBM* şi *Texas Instruments* rezolvă aceste limitări de trafic ale reţelei *Ethernet*.

Adaptoare *Token Ring*. Folosesc perechi de cabluri răsucite, ecranate sau neecranate. *Token Ring* este cel mai scump tip de reţea *LAN*. Poate fi întâlnită în clădirile marilor corporaţii cu reţele vaste, mai ales când reţelele sunt conectate la calculatoare mainframe. *Token Ring* operează la o rată de *4Mbps* sau *16Mbps*. Staţiile de pe o reţea locală *Token Ring* trec de la una la alta mesaje *token*. *Token* este un scurt mesaj indicând că reţeaua este neocupată. Dacă o staţie nu are nimic de trimis, îndată ce primeşte un token îl transmite staţiei de lucru imediat următoare. O staţie nu poate trimite un mesaj în reţeaua decât atunci când primeşte un *token*. Mesajul trimis circulă prin staţiile şi file serverele reţelei *LAN*, ajungând din nou la

emiţător, după care acesta trimite un token care indică faptul că reţeaua nu mai este ocupată. În timp ce mesajul circulă, o staţie sau un file server recunoaşte că acesta i se adresează şi începe procesarea lui. *Token Ring* nu risipeşte însă resursele reţelei, cum ar putea părea, circulaţia mesajului token prin reţea neconsumând timp, chiar dacă sunt foarte multe staţii de lucru. Anumitor staţii de lucru şi file server-e li se pot asigna priorităţi, pentru ca acestea să obţină mai des accesul la *LAN*. În plus, schema de trecere a mesajului *token* este mult mai tolerantă la niveluri mari de trafic în *LAN* decât percepţia *Ethernet* a coliziunilor. *ARCne*t şi *Token Ring* nu sunt compatibile între ele, dar şi *ARCnet* utilizează o schemă similară de trecere a mesajelor token pentru controlul accesului la reţea a staţiei de lucru şi al file serverului. Staţiile *LAN* se supraveghează reciproc şi folosesc o procedură complexă de regenerare a unui *token*, în cazul când una dintre ele l-a pierdut.

Anexa 3

Costul porţilor pentru unele lăţimi de bandă

Lăţimea de bandă	**Costul portului prin care trece calea (802.1D-1998)**	**Costul portului prin care trece calea (802.1W-2001)**
4 Mbps	250	5000000
10 Mbps	100	2000000
16 Mbps	62	1250000
100 Mbps	19	200000
1 Gbps	4	20000
2 Gbps	3	10000
10 Gbps	2	2000

Anexa 4

Rată de transmisie a datelor

Topologia *RING* (inel)			1..10 Mbps
Transmisia	asincronă		maxim 115 Kbps
	sincronă		2 Mbps
Ethernet 10G			10 Gbps
Gigabit Ethernet			1000 Mbps
FastEthernet			100 Mbps
Ethernet			10 Mbps
Reţele locale fără fir	802.11a, 802.11g		54 Mbps
	802.11b		11Mbps
Comutaţie	de circuite (*Circuit-switched*)	*ISDN* (Servicii Integrate *Digital Network*)	128 Kbps…3 Mbps
	de pachete (*Packet-switched*)	*X.25 (Packet Switching)*	2 Mbps
		FR (*Frame Relay*)	1,544 Mbps
	Cell-switched - ATM (*Asyncronous Transfer Mode*)		622 Mbps.
Vocea umană în format digital			64 Kbps
Telefonie digitală	*T1*		1,544 Mbps
	T2		6,312 Mbps
	T3		44,736 Mbps
	T4		274 Mbps
	E1		2,048 Mbps
	E2		8,488 Mbps
	E3		34,368 Mbps
	ADSL		1 Mbps…8 Mbps
	SONET		51,84 Mbps… 39,81Gbps
Analog services			maxim 10 Mbps
WiMAX			70…75 Mbps
Adaptor			10 Mbps
Linia telefonică	cu comutare (*dial-up*)		56 Kbps
	dedicată (închiriată)		45 Mbps

Telefonie mobilă	*D-AMPS* (*Digital Advanced Mobile Phone System*)	8 Kbps
	GSM (*Global System for Mobile Communications*)	13 Kbps
	EDGE (în regim de comutaţie de pachete)	maxim 474 Kbps
	3G (*Third Generation*)	1,5…2 Mbps
	3.9G (*Third Generation*)	100 Mbps
Bluetooth - *SCO* (*Synchronous Connection Oriented*)		64 Kbps
Ethernet - o fereastră de timp pentru transmiterea unui bit		10 Mbps