

REȚELE DE CALCULATOARE

Bibliografie:

- Computer Networks** (5th Ed.), Tanenbaum, A. S., Wetherall, D. J., *Prentice Hall*, 2010
- Data Communications and Networking** (5th Ed.), Forouzan, B.A., McGraw-Hill, 2012
- Alcatel-Lucent Scalable IP Networks Self-Study Guide**, Hundley K., Wiley, 2009
- Computer Networks: A Systems Approach** (5th Ed.), Peterson, L. L., Davie, B. S., *Elsevier*, 2011
- Computer Networking A Top-Down Approach** (6th Ed.), Kurose, J. F., Ross, K. W., Pearson, 2012
- TCP/IP Illustrated, Vol. 1: The Protocols**, Stevens, W. R., Addison-Wesley, 1993
- Unix Network Programming**, Vol. 1: The Sockets Networking API (3rd Ed.), Stevens, W. R., Fenner, B., Rudoff, A. M., Addison-Wesley, 2003

1. Introducere în rețele de calculatoare

Definiție

Vechiul model al unui singur calculator (mainframe) care servește problemele de calcul ale unui sistem a fost înlocuit cu un model în care munca este făcută de un număr mare de calculatoare, care sunt utilizate separat, dar interconectate.

Def. 1: rețeaua de calculatoare reprezintă un ansamblu de echipamente de calcul răspândite geografic, interconectate prin intermediul unor medii de comunicație, asigurându-se în acest fel utilizarea în comun de către un număr mare de utilizatori a tuturor resurselor fizice (hardware), logice (software și aplicații de bază) și informaționale (baze de date) de care dispune ansamblul de calculatoare conectate.

Def. 2: prin **rețea de calculatoare** înțelegem o colecție de calculatoare autonome interconectate între ele. Se spune despre două calculatoare că sunt interconectate dacă sunt capabile să schimbe informații între ele.

Avantajele rețelelor de calculatoare:

- asigurarea comunicării între sistemele de calcul;
- creșterea eficienței în transferurile de date;
- scăderea costurilor în funcționare;
- protejarea simplă și eficientă a datelor;
- asigurarea disponibilității;
- optimizarea supraîncărcării sistemelor de calcul;
- optimizarea întreținerii.

După tehnologia de transmisie:

- rețele cu difuzare (broadcast);
- rețele punct – la – punct.

După scara la care operează rețeaua (distanța):

- rețele personale (PAN);
- rețele locale (LAN);
- rețele metropolitane (MAN);
- rețele de arie înglobată (WAN);
- internet (GAN).

După topologie:

- rețele tip magistrală (bus);
- rețele tip stea (star);
- rețele tip inel (ring);
- rețele combinate.

După tipul sistemului de operare utilizat:

- rețele peer-to-peer;
- rețele bazate pe server.

După tipul mediului de transmisie al semnalelor:

- rețele prin medii ghidate (cablu coaxial, perechi de fire răsucite, fibra optică);
- rețele prin medii neghidate (transmitere în infraroșu, unde radio, microunde).

După tipul utilizatorilor:

- private (de uz industrial, militar, civil);
- publice (internet).

După tipul accesului la mediu

- ethernet;
- token ring;
- token bus.

Rețelele cu difuzare (broadcast) sunt acele rețele care au un singur canal de comunicație care este partajat de toate (este accesibil tuturor) calculatoarele din rețea. Mesajul (numit pachet) poate fi adresat unui singur calculator, tuturor calculatoarelor din rețea (acest mod de operare se numește difuzare) sau la un subset de calculatoare (acest mod de operare se numește trimitere multiplă).

Rețelele punct - la - punct sunt acele rețele care dispun de numeroase conexiuni între perechi de calculatoare individuale. Pentru a ajunge de la calculatorul sursă la calculatorul destinație, un pachet s-ar putea să fie nevoie să treacă prin unul sau mai multe calculatoare intermediare. Deseori sunt posibile trasee multiple, de diferite lungimi, etc.

In general rețelele mai mici (locale) tind să utilizeze difuzarea, în timp ce rețelele mai mari sunt de obicei punct - la - punct.

Clasificare după scara la care operează

Rețelele LAN - Local Area Network - sunt în general rețele private localizate într-o singură cameră, clădire sau într-un campus de cel mult câțiva kilometri.

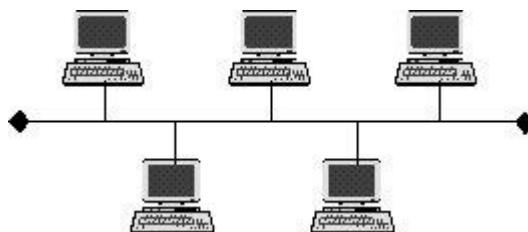
Rețelele MAN - Metropolitan Area Network - reprezintă o extensie a rețelelor LAN și utilizează în mod normal tehnologii similare cu acestea. Aceste rețele pot fi atât private cât și publice. Un aspect important al acestui tip de rețea este prezența unui mediu de difuzare la care sunt atașate toate calculatoarele. Aceste rețele funcționează, în general, la nivel de oraș.

Retele WAN - Wide Area Network - sunt acele rețele care acoperă o arie geografică întinsă - deseori o țară sau un continent întreg.

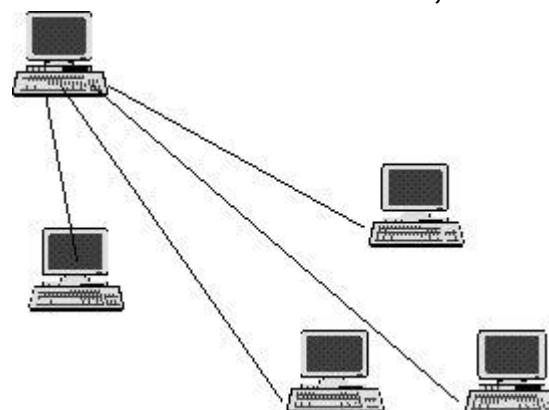
Rețeaua GAN - Global Area Network – descrie întinderea WAN-urilor la o dimensiune globală.

Clasificare după topologie

Topologia magistrală (bus) sau **liniară** - este cea mai simplă și mai uzuală metodă de conectare a calculatoarelor în rețea. Ea constă dintr-un singur cablu, numit *trunchi* care conectează toate calculatoarele din rețea pe o singura linie;

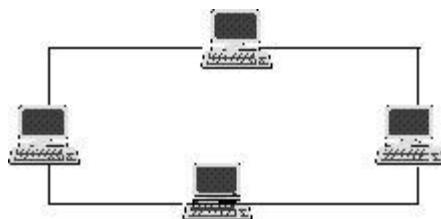


Topologia stea (star) - atunci când se utilizează aceasta topologie toate calculatoarele sunt conectate la un nod central care joacă un rol particular în funcționarea rețelei. Orice comunicație între două calculatoare va trece prin acest nod central, care se comportă ca un comutator față de ansamblul rețelei.



Clasificare după topologie

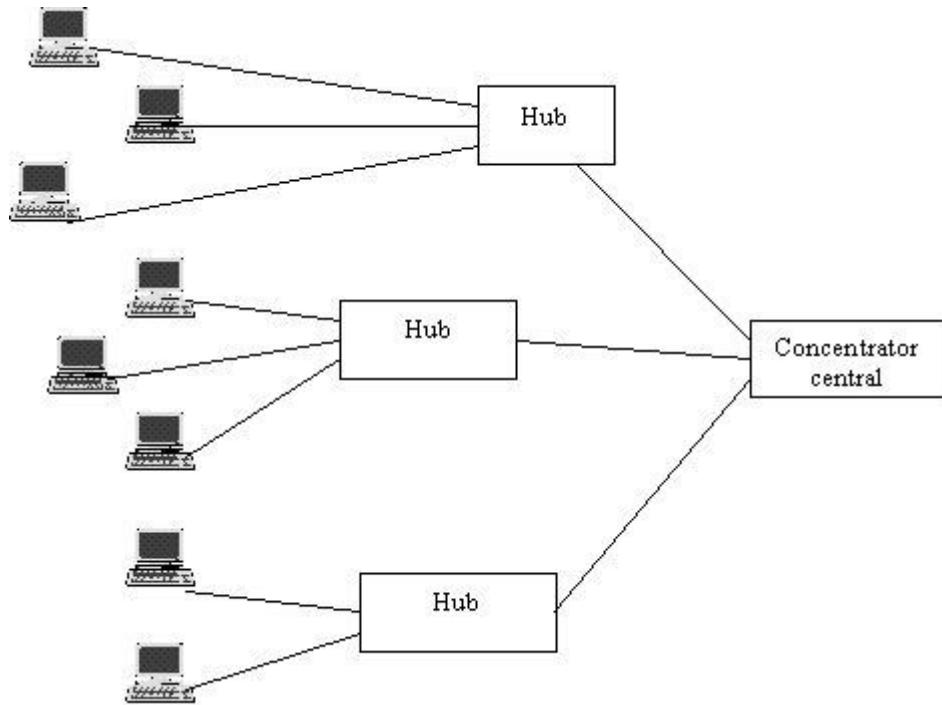
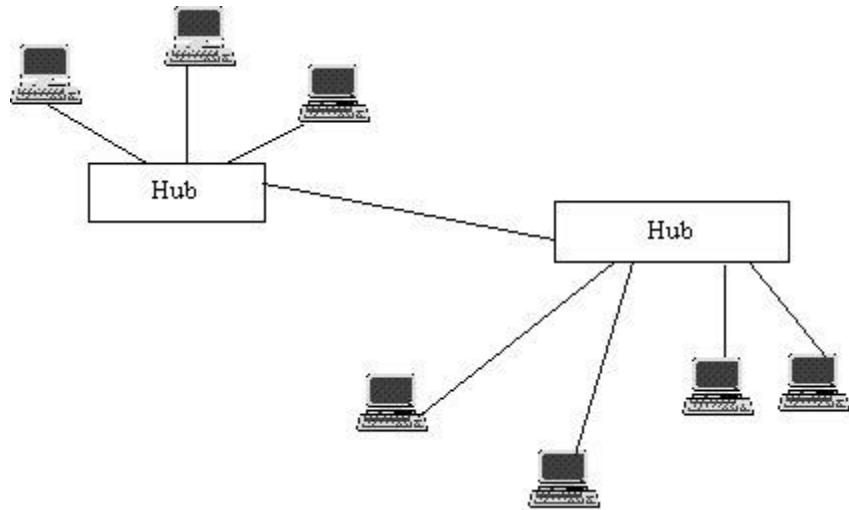
Topologia inel (ring) - într-o astfel de configurație toate calculatoarele sunt legate succesiv între ele, două câte două, ultimul calculator fiind conectat cu primul.



În afara acestor topologii standard există și alte variante combinate, dintre care cele mai uzuale sunt:

- **topologia magistrala-stea** leagă mai multe rețele cu topologie stea prin intermediul unor trunchiuri liniare de tip magistrală;
- **topologia inel-stea** este asemănătoare topologiei magistrală - stea. Deosebirea constă în modul de conectare a concentratoarelor: în topologia magistrală - stea ele sunt conectate prin trunchiuri lineare de magistrală, iar în topologia inel - stea sunt conectate printr-un concentrator principal.

Clasificare după topologie



Rețelele peer-to-peer sunt acele rețele în care partajarea resurselor nu este făcută de către un singur calculator ci toate aceste resurse sunt puse la comun de către calculatoarele din rețea.

Rețele bazate pe server (client / server) sunt acele rețele care au în componență un server specializat: (ex. de fișiere, de tipărire, de aplicații, de postă, de fax, de comunicatii).

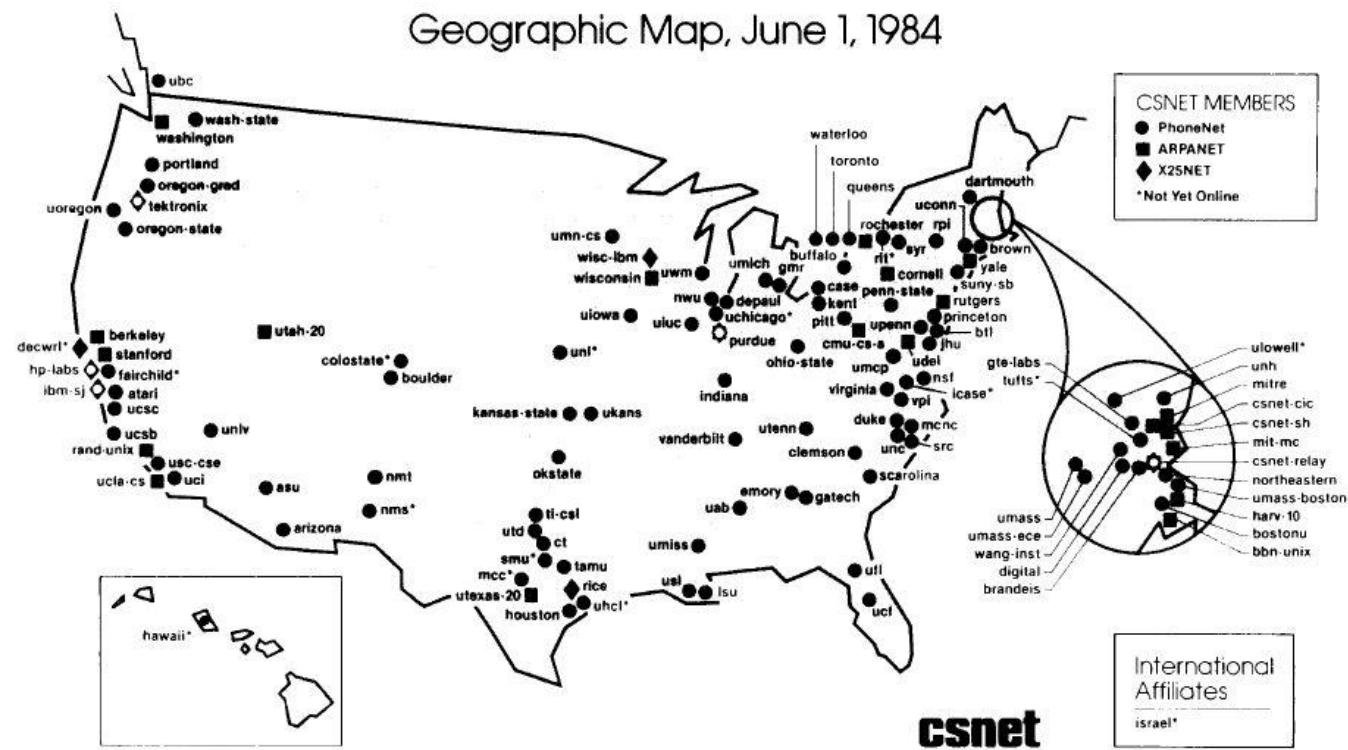


The ARPANET in December 1969

1969 ARPANET (<https://www.vox.com/a/internet-maps>)

1969 – telnet

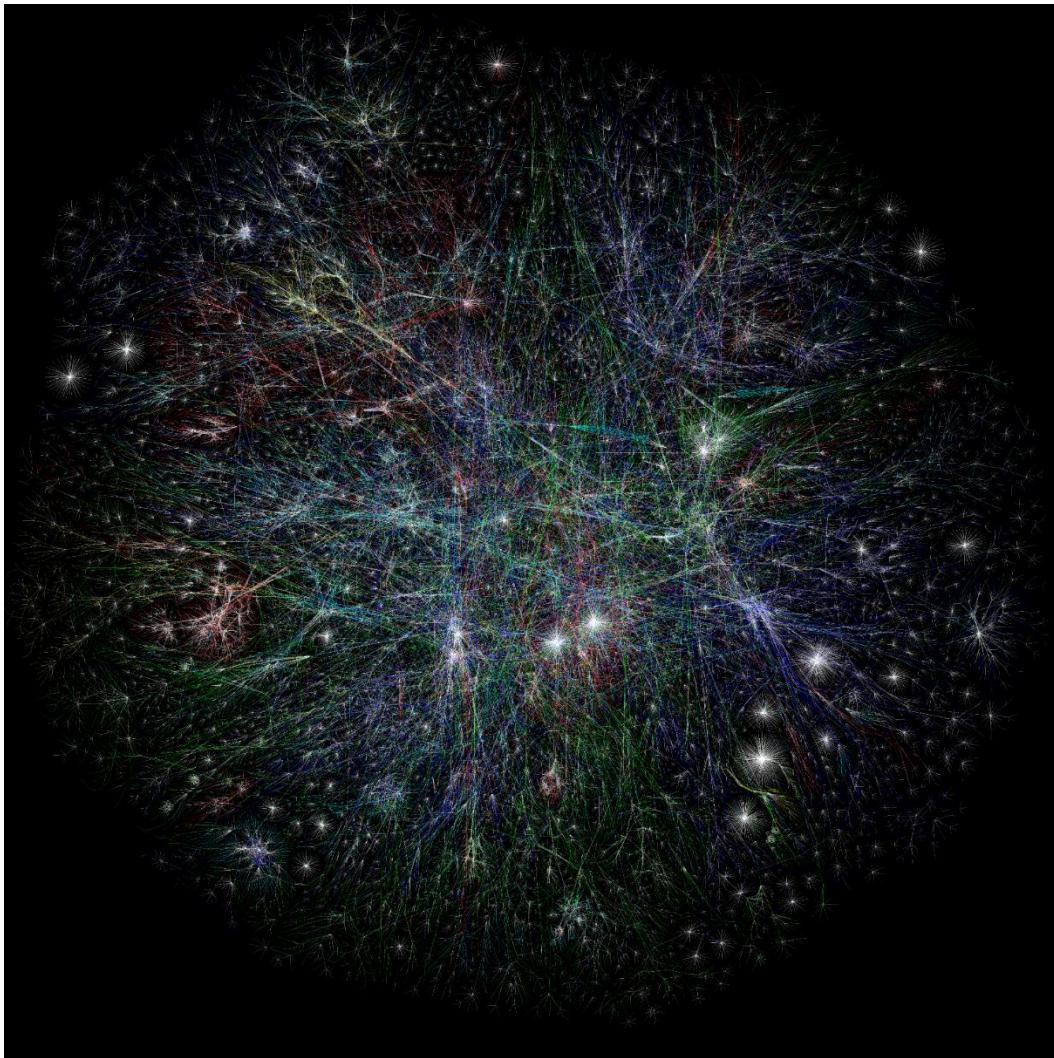
1971 – email (Ray Tomlinson)



1984 Internet (<https://www.vox.com/a/internet-maps>)

1983 - TCP/IP (Robert Kahn și Vint Cerf)

Internetul



2003 (<http://www.opte.org>)

Asia Pacific - Red

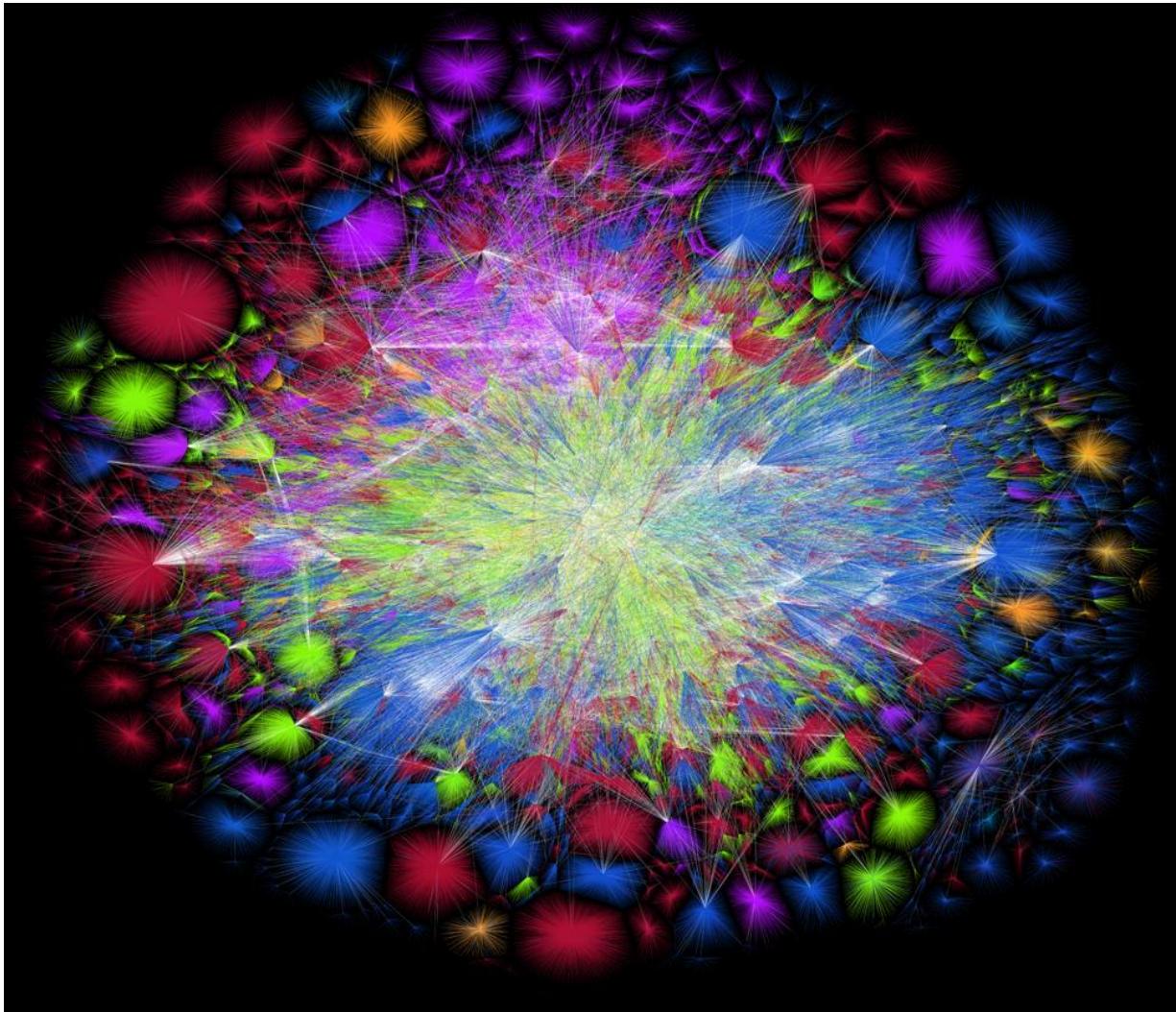
Europe/Middle East/Central Asia/Africa -
Green

North America - Blue

Latin American and Caribbean - Yellow

RFC1918 IP Addresses - Cyan

Internetul



North America (ARIN)
Europe (RIPE)
Latin America (LACNIC)
Asia Pacific (APNIC)
Africa (AFRINIC)
"Backbone" (highly connected networks)

2015 (<http://www.opte.org>)

Standardizare. Modele de rețea

Telecomunicații

ITU – International Telecommunication Union

(peste 200 membrii guvernamentali)

ITU-T – telecomunicații

ITU-R – radiocomunicații

ITU-D – dezvoltare

Standarde internaționale

ISO – International Standards Organization (157 țări membre)

IEEE – Institute of Electrical and Electronics Engineers

Internet

IAB – Internet Architecture Board (guvernat de Internet Society)

IETF – Internet Engineering Task Force,

IERF – Internet Engineering Research Force

(RFC-> Draft Standard -> Internet Standard)

W3C – World Wide Web Consortium

Protocol

- limbaj, set de reguli;
- defineste ce se comunica, cum se comunica si cand se comunica.

Elementele unui protocol:

1. **Sintaxa** – defineste structura sau formatul datelor si ordinea in care sunt prezentate.
2. **Semantica** – defineste rolul fiecarei sectiuni de date.
3. **Sincronizare temporală** – precizeaza cand se trimit datele si cu ce viteza.

Marea majoritate a retelelor de calculatoare sunt organizate pe mai multe niveluri (straturi), permitandu-se astfel impartirea unor sarcini complexe in unele mai mici de o complexitate mai redusa. Numarul nivelurilor, numele si continutul acestora difera de la o retea la alta. Rolul fiecarui nivel este de a oferi anumite servicii nivelurilor superioare, fara a furniza detalii legate de modul de implementare a acestor servicii. Colaboararea intre serviciile diverselor straturi se realizeaza cu ajutorul unor interfete (logice si/sau fizice) standardizate. Se poate face o analogie aici cu dezvoltarea software, unde serviciile oferite de o functie dintr-o biblioteca pot fi utilizate atata timp cat cunoastem interfata functiei, fara a avea detalii legate de modul de implementare.

Hierarhia de protocoale

Unul dintre avantajele impartirii pe straturi este separarea serviciilor oferite de implementarea lor, asigurandu-se in acest fel modularitatea retelei. Astfel, se faciliteaza interconectarea echipamentelor avand producatori diferiti. Daca interfetele de intrare/iesire (fizice si logice) ale unor echipamente, ce au o implementare diferita, sunt identice (sau macar compatibile), aceste echipamente pot fi interconectate sau interschimbate.

In general, pentru transmisia datelor intre o sursa si o destinatie sunt necesare o serie de echipamente intermediare de interconectare a retelelor ce compun traseul. Impartirea pe straturi permite echipamentelor intermediare sa implementeze doar anumite servicii strict necesare pentru transportul datelor. Astfel, echipamentele intermediare implementeaza doar anumite straturi, scazand costurile de productie. Aceasta facilitate reprezinta un alt avantaj al impartirii pe straturi.

Nota:

Cand nivelul n al unui echipament de retea comunica cu nivelul n al altui echipament, totalitatea regulilor si convintiilor folosite in conversatie alcatuiesc protocoalele respectivului nivel (denumite protocoale de nivel n).

Model de rețea – colecție de standarde, cu o structură modulară, ce definește unul sau mai multe tipuri de rețele.

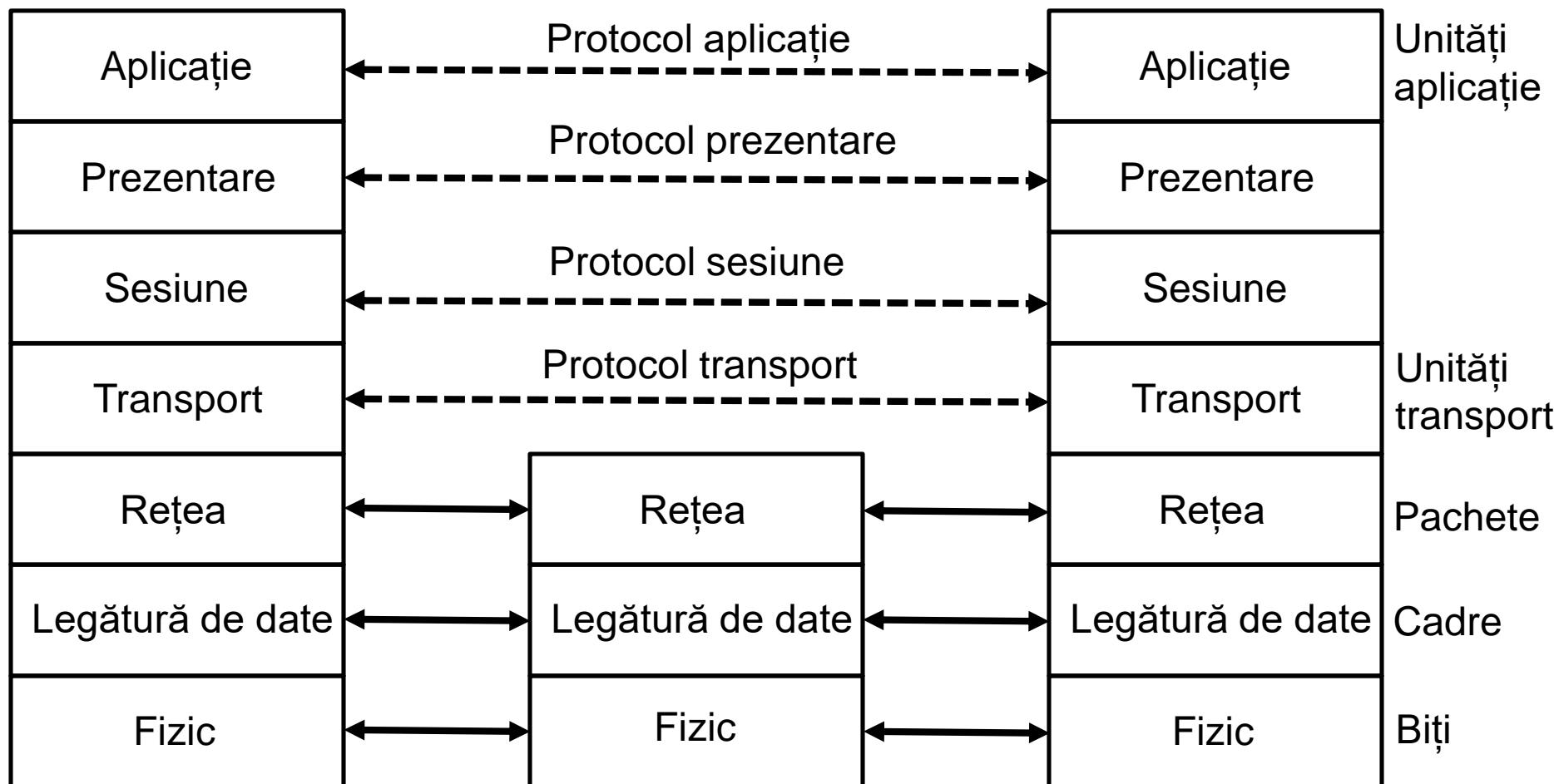


Modelul OSI - Open Systems Interconnection:

- propus de ISO ca prim pas în standardizarea protocolelor;
- împărțit pe 7 straturi;
- model didactic, de referință;
- protocole neutilizate în practică.

Modelul OSI

Modelul OSI. Conexiuni logice intre straturi



Modelul OSI. Conexiuni logice intre straturi

Pentru o mai buna intelegerere a functionalitatilor fiecarui nivel, ne putem imagina existenta unor legaturi logice intre niveluri. Fiecare nivel al unui echipament comunica cu acelasi nivel al unui interlocutor. Dupa cum se poate observa din figura de pe diapozitivul anterior, nivelurile aplicatie, prezentare, sesiune si transport implementeaza servicii de tip capat-la-capat (end-to-end), comunicand strict intre sursa si destinatie. In schimb, in cazul celorlalte trei niveluri, comunicatia este de tip nod-la-nod (hop-to-hop), unde un nod poate fi un echipament intermediar sau unul final.

Nivelul fizic

Coordoneaza functiile necesare transmisiei unui flux de biti pe mediul fizic, intre echipamentele de calcul.

La acest nivel sunt definite:

- specificatiile mecanice si electrice ale interfetelor si ale mediilor de transmisie, precum si functiile pe care acestea le au.
- modalitatile de reprezentare a bitilor (tipul si proprietatile semnalelor, modalitati de codificare)
- ratele de transmisie (numarul de biti pe secunda; durata fiecarui bit)
- modalitatile de sincronizare a bitilor
- tipurile de topologii fizice utilizate (bus, star, mesh etc)
- modurile de transmisie (simplex, half-duplex, full duplex)

Nivelul legatura de date

Abstractizeaza mediul fizic, definindu-l ca si un canal de comunicare fiabil, capabil sa transmita date fara erori.

Principalele responsabilitati ale acestui nivel sunt:

- controlul erorilor (detectia si corectia);
- controlul fluxului (pentru evitarea suprasaturarii receptorului);
- impartirea datelor in cadre (incapsularea si decapsularea);
- adresarea fizica a echipamentelor;
- controlul accesului la mediu.

Nivelul retea

Este responsabil de transmisia datelor de la sursa la destinatie, indiferent de numarul de retele intermediare.

Principalele responsabilitati ale acestui nivel sunt:

- interconectarea retelelor (internetworking);
- rutarea datelor (identificarea caii optime catre destinatie/comutarea pachetelor de date);
- impartirea datelor in pachete (incapsularea si decapsularea);
- fragmentarea pachetelor de date;
- adresarea logica a echipamentelor de retea;
- asigurarea calitatii serviciului.

Nivelul transport

Este responsabil de transmisia datelor intre procese (aplicatiile implicate in schimbul de mesaje, ce ruleaza pe sistemele de operare ale sursei si destinatiei).

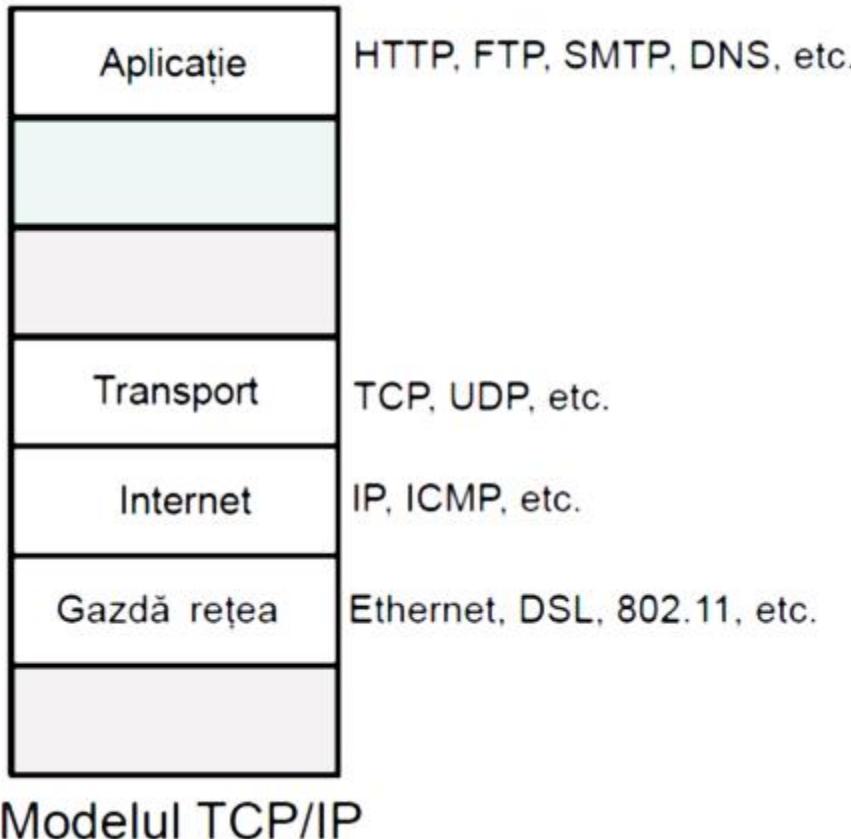
Principalele responsabilitati ale acestui nivel sunt:

- implementarea de servicii orientate/neorientate pentru nivelul aplicatie;
- controlul fluxului de date;
- controlul congestiei;
- adresarea proceselor (socket-urilor) ce comunica in retea;
- impartirea datelor in segmente/datagrame (incapsularea si decapsularea)

Nivelul sesiune – gestioneaza crearea si distrugerea de sesiuni de date.

Nivelul prezentare – defineste modul de formatare a datelor: encoding, codificare, etc.

Nivelul aplicatie – faciliteaza accesul utilizatorilor si al aplicatiilor la retea.

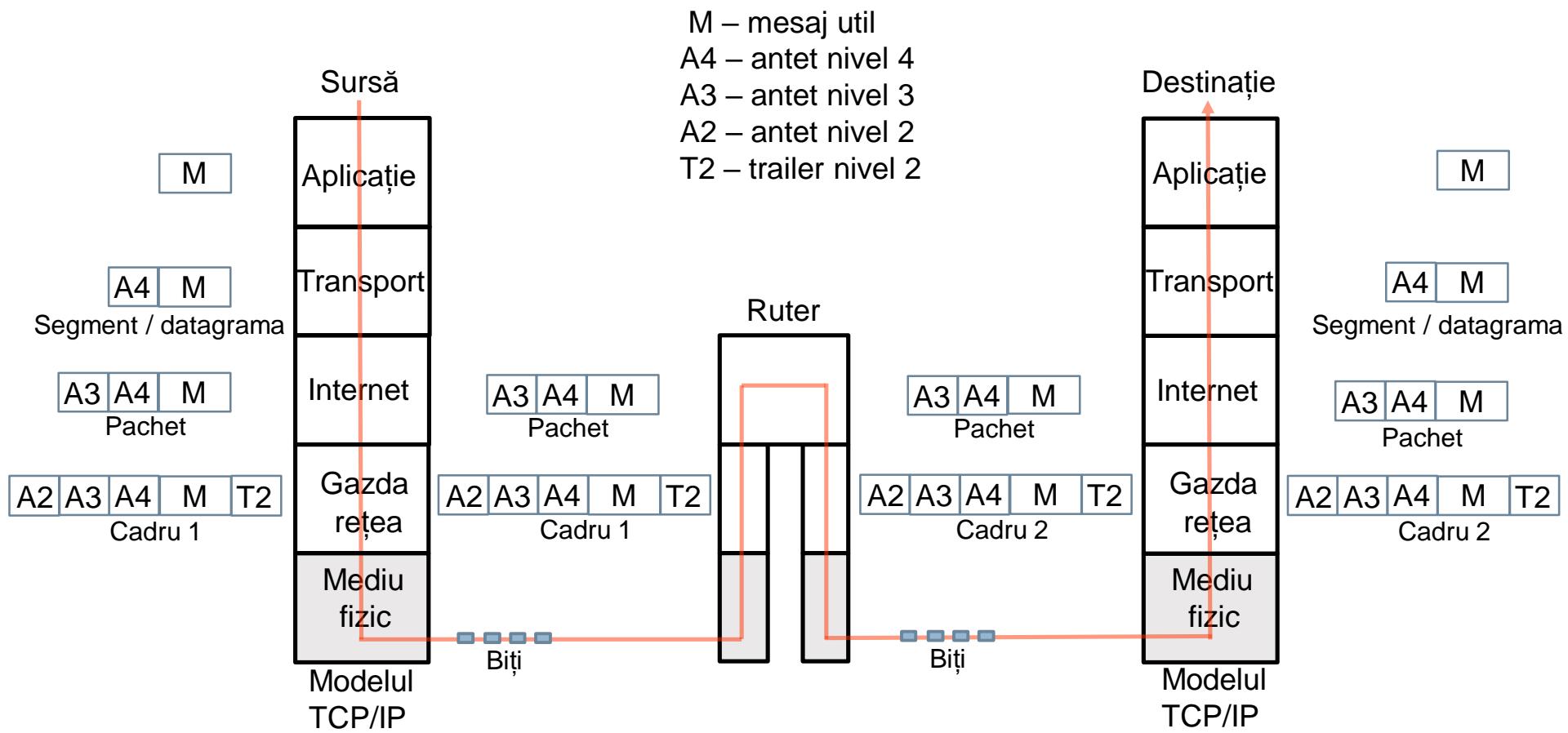


Modelul TCP/IP:

- model suport pentru internet;
- definit în urma dezvoltării protocolelor, în vederea documentării acestora.

Incapsularea / decapsularea datelor

Un concept important, introdus odată cu împărțirea pe niveluri a serviciilor de rețea, este cel de incapsulare / decapsulare date.



Incapsularea / decapsularea datelor

Echipamentul sursa realizeaza incapsularea datelor.

1. La nivelul aplicatie, protocolul specific imparte datele in mesaje, de diverse dimensiuni. Aceste mesaje se transmit la nivelul transport. Protocolul de nivel aplicatie s-ar putea sa introduca la randul sau un antet, dar nefiind obligatoriu il vom considera ca parte integranta din mesaj.
2. Nivelul transport preia mesajul de la nivelul aplicatie si il imparte in segmente sau datagrame (in functie de protocolul de nivel transport utilizat). Pe langa informatia utila din mesaj, segmentul/datagrama va contine un antet specific protocolului de nivel 4. Acest antet include, in anumite cazuri, pe langa adresele (porturi) ce definesc aplicatiile sursa si destinatie si informatii necesare controlului erorilor, fluxului, al congestiei etc. Segmentele/datagramele sunt transmise mai departe nivelului internet.
3. Nivelul internet adauga antetul propriu informatiei primita de la nivelul 4 si construieste pachetul de date. Antetul contine adresa echipamentului sursa si a celui destinatie, precum si alte informatii legate de o posibila fragmentare, calitatea serviciului etc. Pachetul de date este transmis mai departe nivelului gazda la retea.
4. Nivelul gazda-retea preia pachetul de date si ii adauga un antet specific si posibil un trailer, rezultand cadrul de date. Antetul introdus va contine adresele fizice ale sursei si nodului urmator de retea prin care trece pachetul catre destinatie.

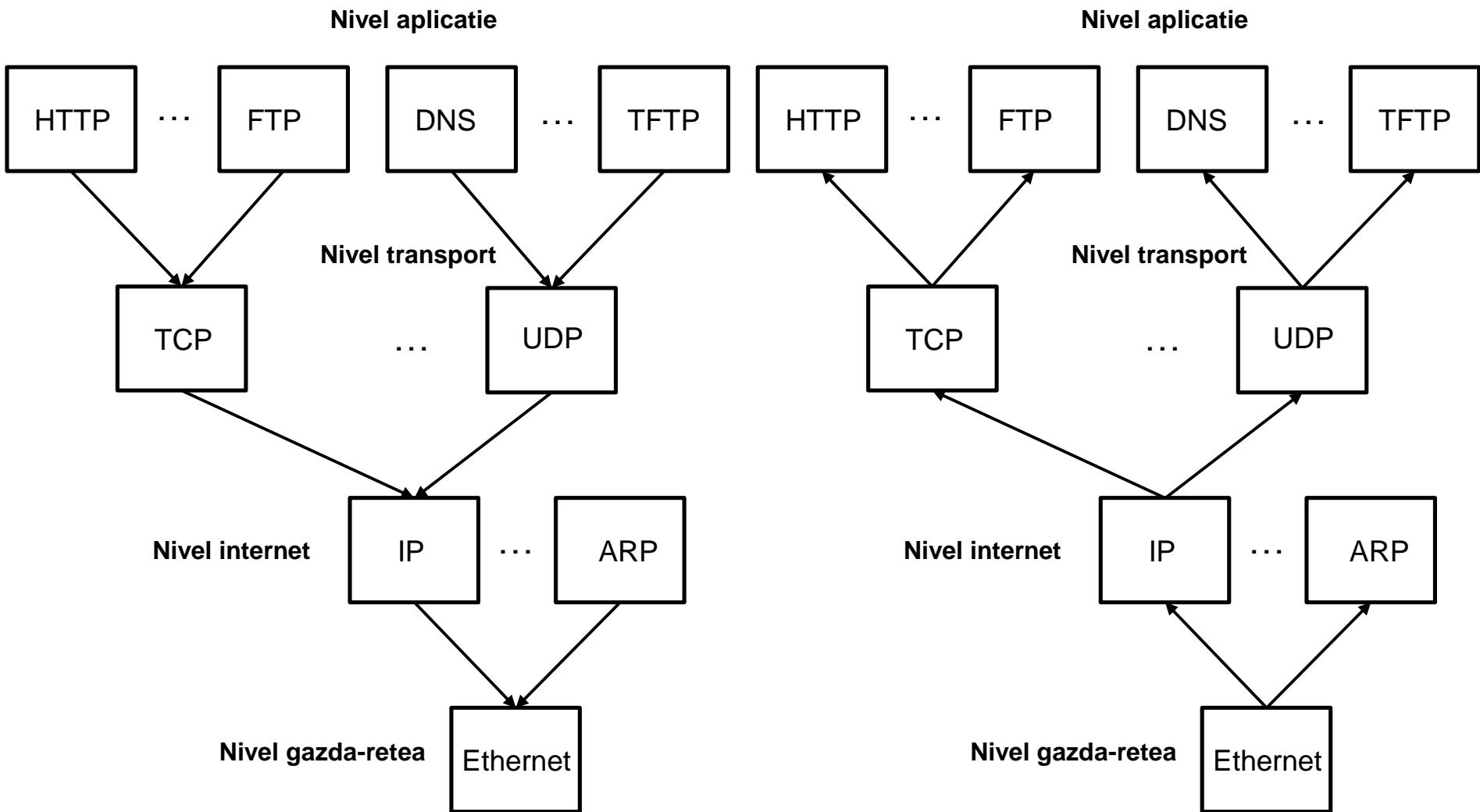
Incapsularea / decapsularea datelor

Echipamentul de tip ruter realizeaza ambele operatii, deoarece acesta este conectat la mai multe retele si retransmite mesajul primit dintr-o retea in alta.

1. In urma primirii multimii de biti ce formeaza cadrul de date, nivelul gazda-retea reface cadrul de date, interpreteaza antetul si trailerul si, in cazul in care cadrul i se adreseaza, il decapsuleaza si transmite pachetul inclus nivelului internet.
2. Nivelul internet verifica adresele logice sursa si destinatie din antetul pachetului siincearca sa identifice urmatorul nod de retea la care trebuie sa ii transmita pachetul in drumul sau catre destinatie si reteaua prin care trebuie sa il transmita pentru a ajunge la acel nod intermediar.
3. Nivelul gazda-retea reincapsuleaza pachetul intr-un cadru de date specific protocolului de nivel 2 utilizat pentru comunicarea cu nodul intermediar si transmite mesajul pe mediul fizic.

La destinatie, fiecare nivel inferior decapsuleaza informatia, transmitand-o mai departe nivelului superior.

Multiplexarea / demultiplexarea protoalelor



Multiplexarea protoalelor la sursa

Demultiplexarea protoalelor la destinatie

Multiplexarea / demultiplexarea protoalelor

Cum modelul TCP/IP utilizeaza mai multe protocoale pe fiecare nivel, un protocol de pe un nivel poate la un moment dat incapsula date de la mai multe protocoale de pe nivelul ierarhic superior. Vorbim in aceasta situatie de o multiplexare a protoalelor la sursa si o demultiplexare a lor la destinatie. Pentru a permite multiplexarea si demultiplexarea protoalelor de pe nivelul ierarhic superior, un protocol are nevoie de un camp de date special in antetul sau care sa ii permita stabilirea protocolului caruia ii apartin datele incapsulate.

Principiul este valabil si in cazul altor modele de retea.

2. Nivelul fizic

Una din funcțiile importante ale nivelului fizic este cea de transfer al informației, prin mediile de transmisie, între echipamentele de calcul. Pentru a putea fi transmise, aceste date sunt convertite în semnale electromagnetice.

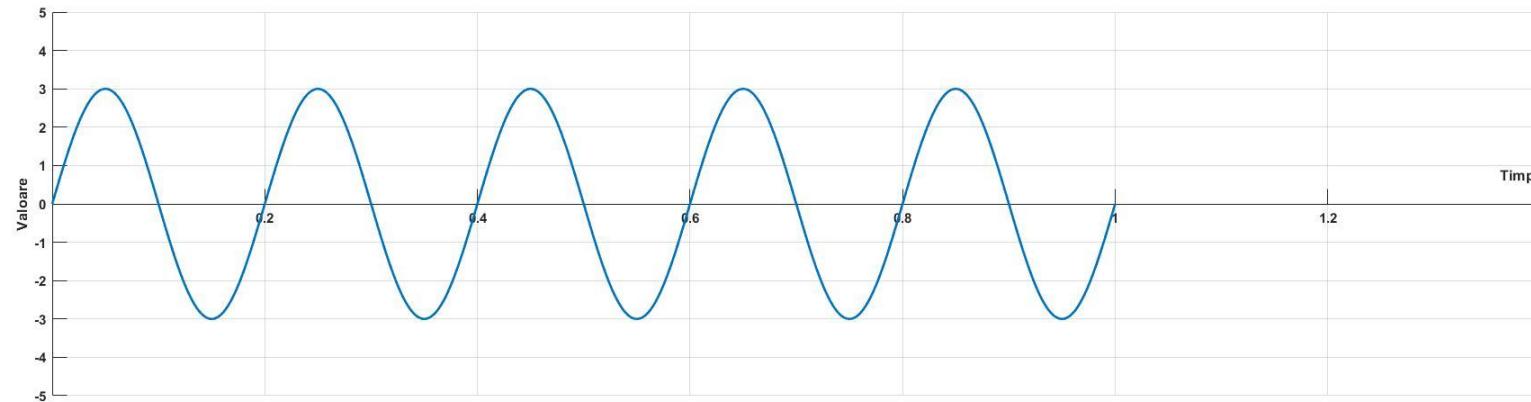
Analogic și digital

- **date analogice** = informație aflată sub o formă continuă;
- **date digitale** = informație reprezentată sub formă de stări discrete.

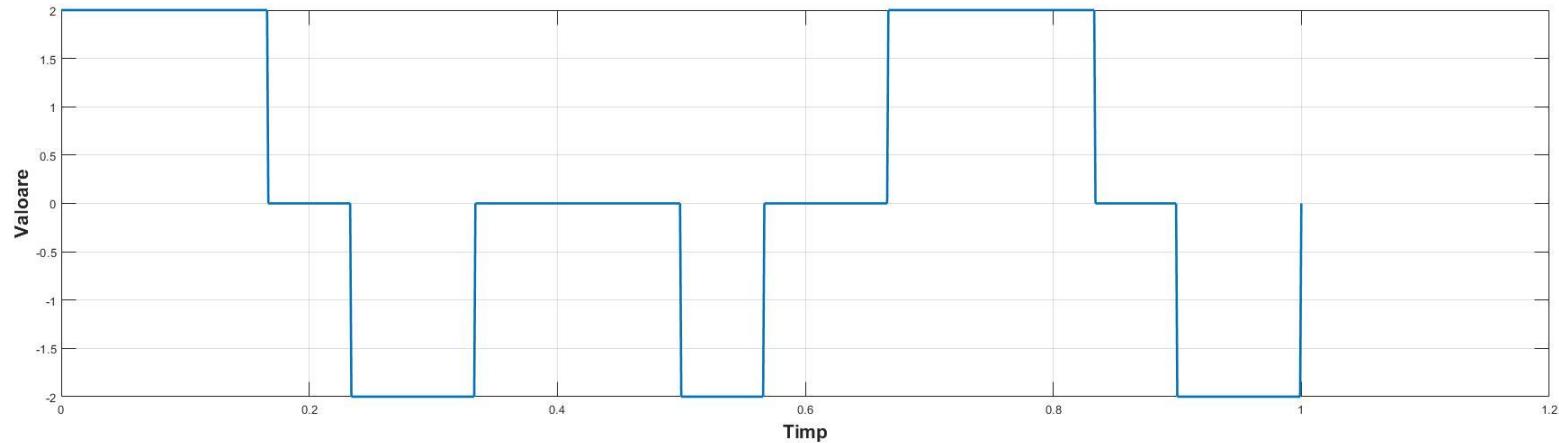
La fel ca și datele pe care le reprezintă, și semnalele pot fi analogice sau digitale.

- un **semnal analogic** are o infinitate de nivele de intensitate, într-o perioadă de timp;
- un **semnal digital** poate avea un număr limitat de valori definite.

Date și semnale



Semnal analogic



Semnal digital

Semnale periodice și aperiodice

Un semnal, definit de funcția $x(t)$, este periodic dacă există un număr real T , nenul, denumit perioadă, astfel încât să fie îndeplinită egalitatea: $x(t+T) = x(t)$. În caz contrar, semnalul este aperiodic.

Atât semnalele analogice, cât și cele digitale pot lua una din cele două forme: periodică sau aperiodică. În comunicațiile de date, în general, se folosesc semnale analogice periodice, deoarece au nevoie de o lățime de bandă mai mică și semnale digitale aperiodice, deoarece reusesc să descrie variația datelor.

Semnale analogice

Semnale analogice periodice simple

Semnalele periodice pot fi simple sau compuse. Un semnal periodic simplu nu poate fi descompus în semnale mai simple. Un semnal periodic compus este format din mai multe semnale simple.



Semnal analogic periodic simplu

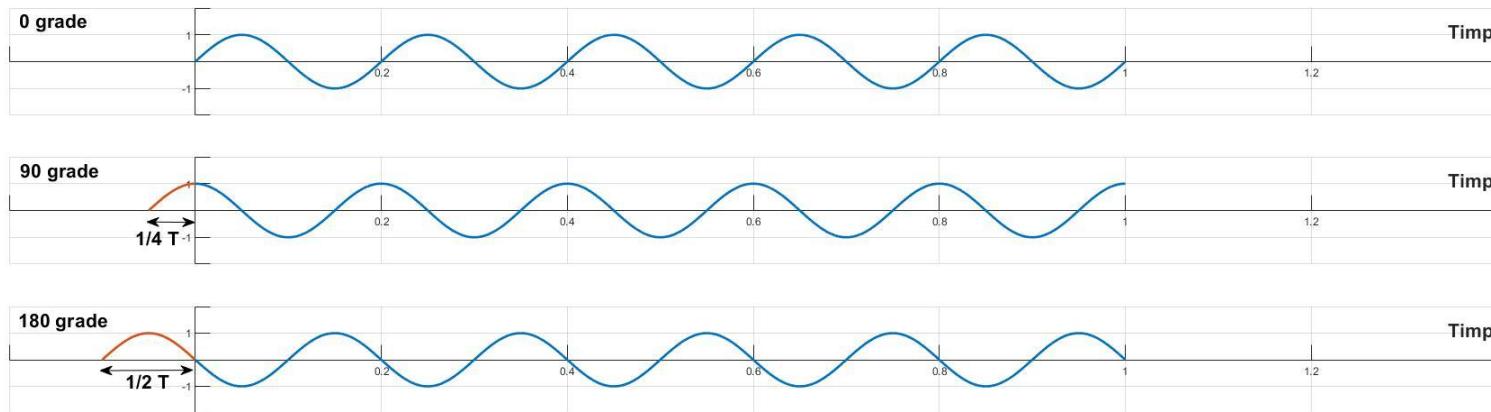
Amplitudinea maximă **A** este definită ca și valoarea absolută a intensității celei mai mari a semnalului, valoare proporțională cu energia sa. În cazul semnalelor electrice, amplitudinea se măsoară în mod normal în volți.

Frecvența **f** definește numărul de perioade dintr-o secundă și se măsoară în Hertz (Hz).

Frecvența se mai definește ca și rata de variație a semnalului în raport cu timpul. În cazul unui semnal constant, spunem că frecvența sa este zero. Dacă un semnal are o variație bruscă (instantanee) de la o valoare la alta, cum perioada este zero, frecvența este infinită.

Date și semnale

Faza semnalului ϕ descrie poziția formei de undă relativă la momentul zero. Dacă considerăm că unda poate fi translatată în față sau în spate pe axa timpului, faza descrie dimensiunea translației. Ea se măsoară în grade sau radieni. O fază de 360 grade corespunde la o translație completă a unei perioade.

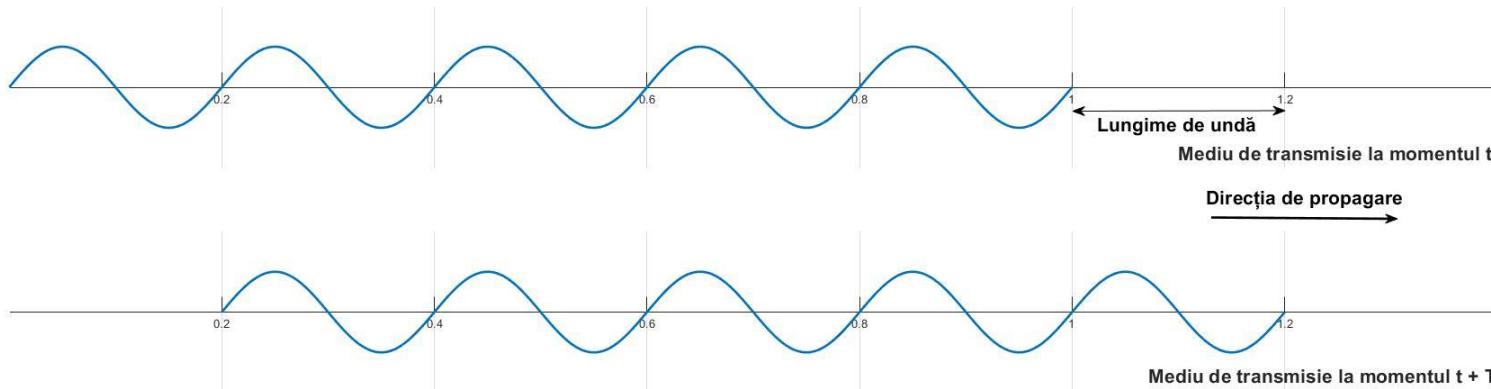


Defazajul semnalului

Lungimea de undă

Lungimea de undă este o altă caracteristică a semnalului ce traversează un mediu de transmisie. Ea leagă perioada unui semnal simplu de viteza de propagare prin mediu și se definește ca și distanța pe care un semnal o parcurge într-o perioadă. Dacă perioada sau frecvența unui semnal sunt independente de mediul de transmisie, lungimea de undă depinde atât de perioada (frecvența) semnalului, cât și de mediu de transmisie.

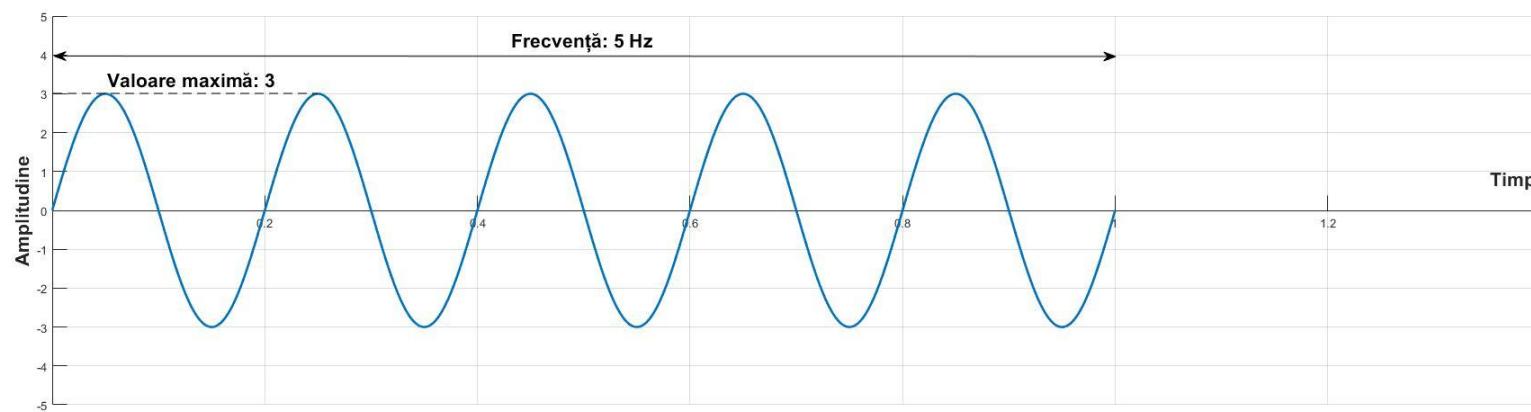
$$\text{Lungimea de undă} = \text{viteza de propagare} \times \text{perioada} = \text{viteza de propagare} / \text{frecvență}$$



Lungimea de undă

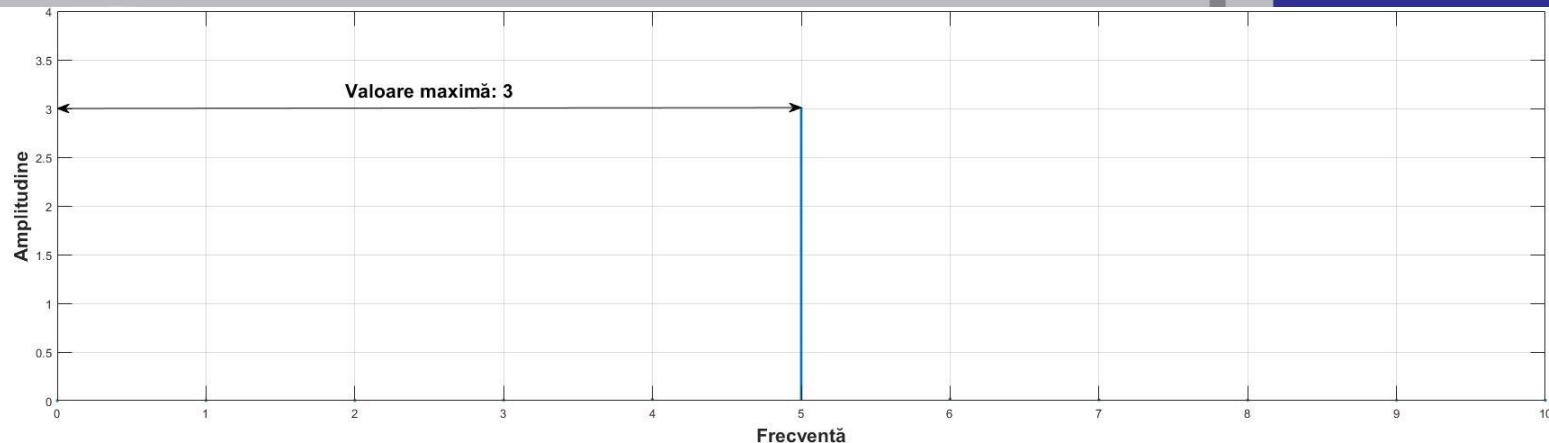
Domenii de reprezentare

Un semnal periodic este definit de amplitudine, frecvență și fază. Un grafic ce reprezintă schimbările de amplitudine în raport cu timpul se numește grafic în **domeniul timp**. Pentru a vedea relația dintre amplitudine și frecvență putem utiliza un grafic în **domeniul frecvență**. Acesta definește valorile maxime ale amplitudinii rapportate la frecvență.



Domeniul timp

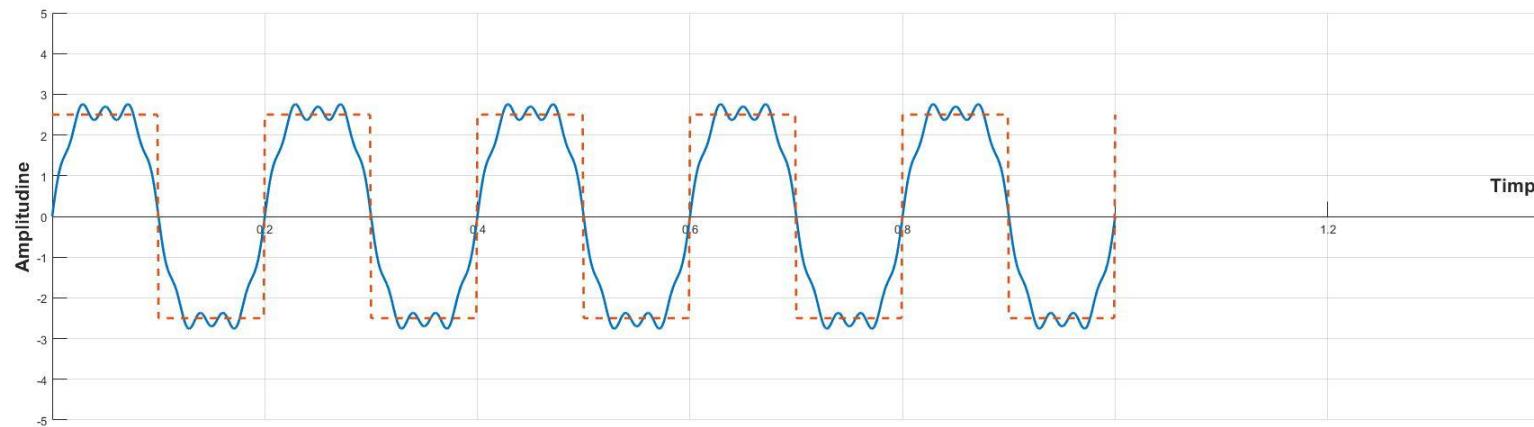
Date și semnale



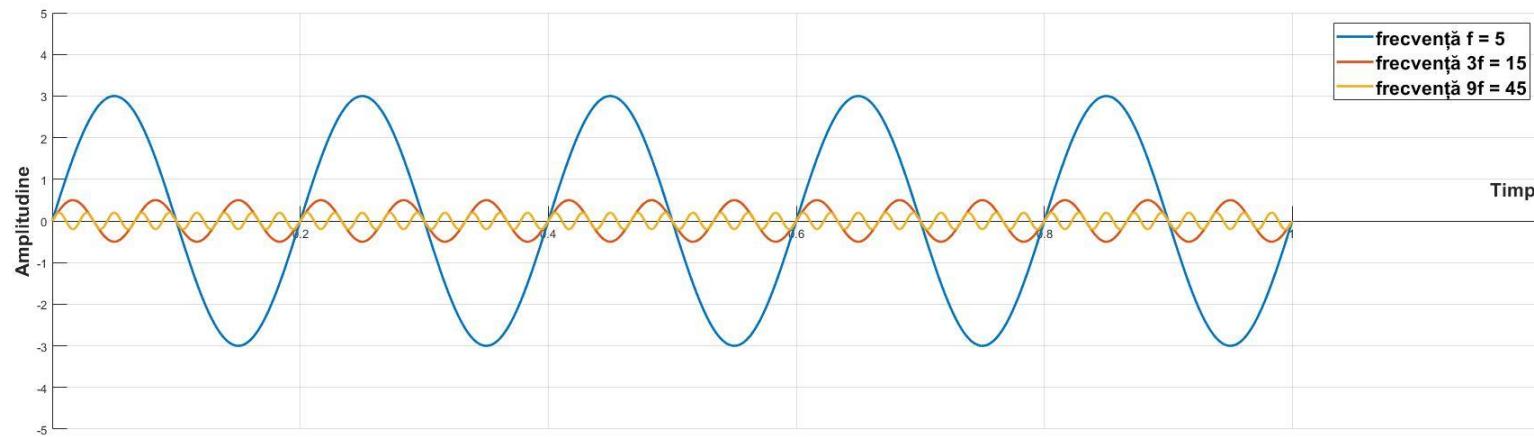
Domeniul frecvență

Semnale analogice compuse

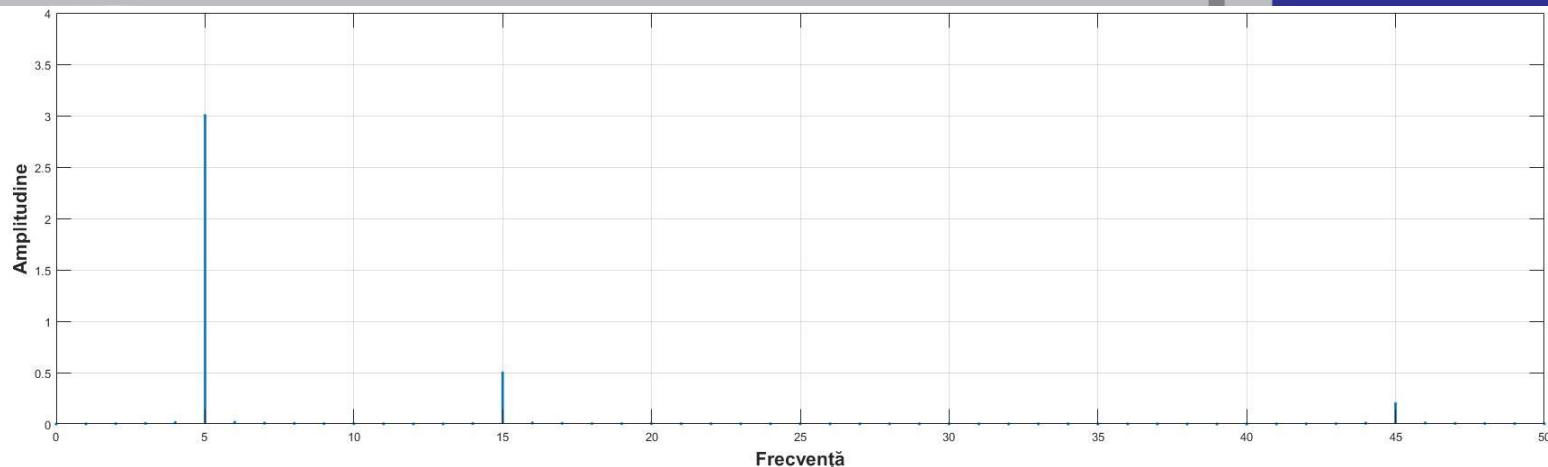
Un semnal periodic simplu, având o singură frecvență, nu este util în comunicațiile de date. Pentru a putea transmite informația avem nevoie de un semnal compus, format din mai multe semnale sinusoidale simple. Conform analizei Fourier, orice semnal compus este o combinație de semnale sinusoidale simple având amplitudini, frecvențe și faze diferite. Si semnalul compus poate fi periodic sau aperiodic. Un semnal compus periodic poate fi descompus într-o serie de semnale sinusoidale simple cu frecvențe discrete, având valori intregi. Un semnal aperiodic poate fi descompus într-o combinație infinită de semnale sinusoidale simple cu frecvențe continue, ce au valori reale.



Semnal periodic compus în domeniul timp

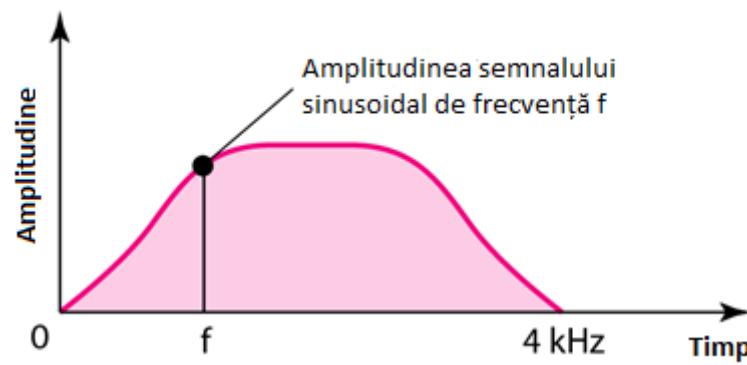


Descompunerea semnalului periodic compus în domeniul timp



Semnal periodic compus în domeniul frecvență

Frecvența cea mai joasă, $f = 5$ Hz, se numește frecvență fundamentală, sau prima armonică. Celelalte două, $3f$ și $9f$, sunt definite ca și a treia, respectiv a nouă armonică.



Semnal aperiodic în domeniul timp și în domeniu frecvență

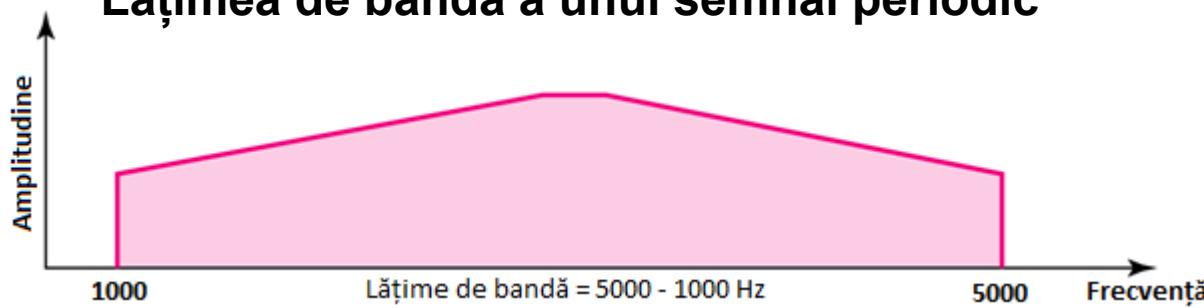
Lățimea de bandă (se utilizează și termenul lărgime de bandă)

Lățimea de bandă a unui canal de comunicație se definește ca și diferența dintre cea mai mare, respectiv cea mai mică dintre frecvențele semnalelor pe care canalul le poate transmite în mod satisfăcător.

Lățimea de bandă a unui semnal reprezintă lățimea spectrului său de frecvențe.



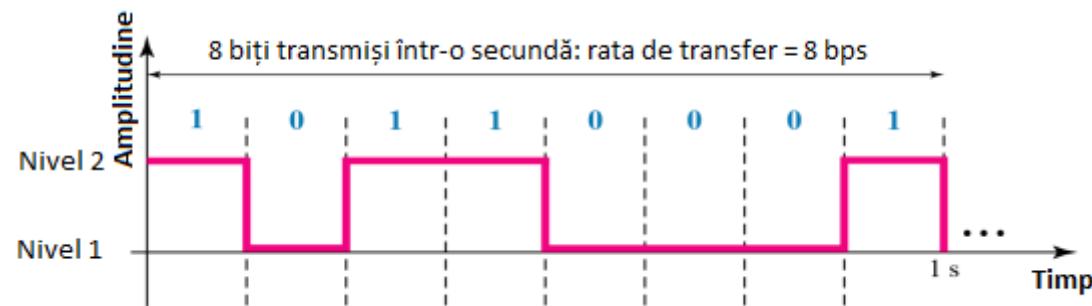
Lățimea de bandă a unui semnal periodic



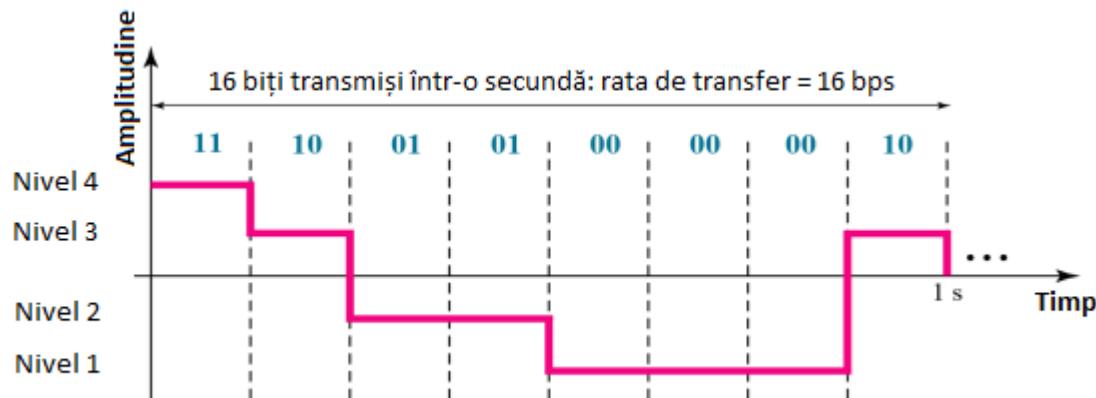
Lățimea de bandă a unui semnal aperiodic

Semnale digitale

Informația poate fi reprezentată și prin semnale digitale. Spre exemplu, un unu logic poate fi reprezentat printr-o tensiune pozitivă și un zero prin lipsa tensiunii (zero). Un semnal digital poate avea două sau mai multe nivele.



Semnal digital cu două nivele



Semnal digital cu patru nivele

Rată de transfer (bit rate)

Majoritatea semnalelor digitale utilizate în transmisia de date sunt aperiodice, asa că perioada, respectiv frecvența, nu sunt caracteristici potrivite pentru a descrie un astfel de semnal. În locul frecvenței, se folosește termenul de rată de transfer pentru descrierea semnalelor digitale.

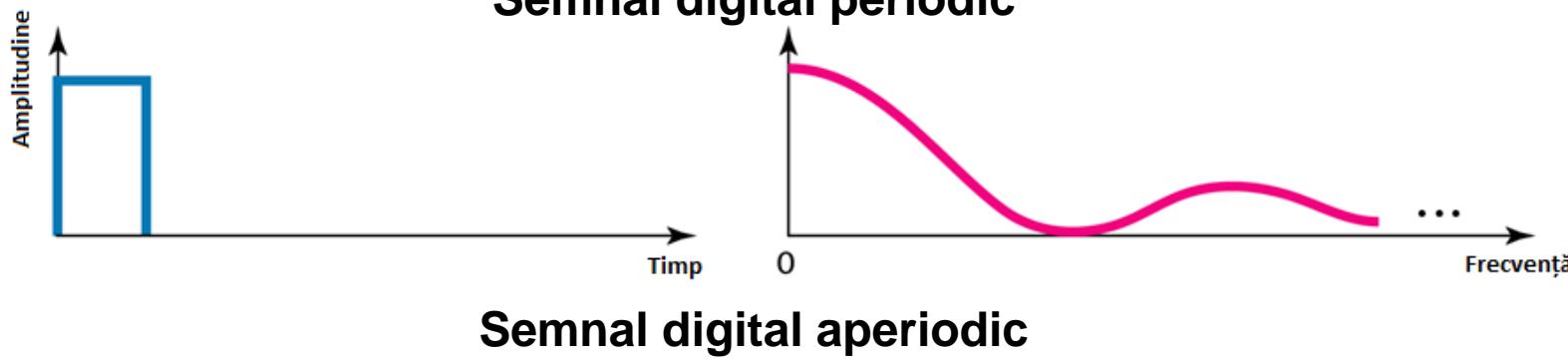
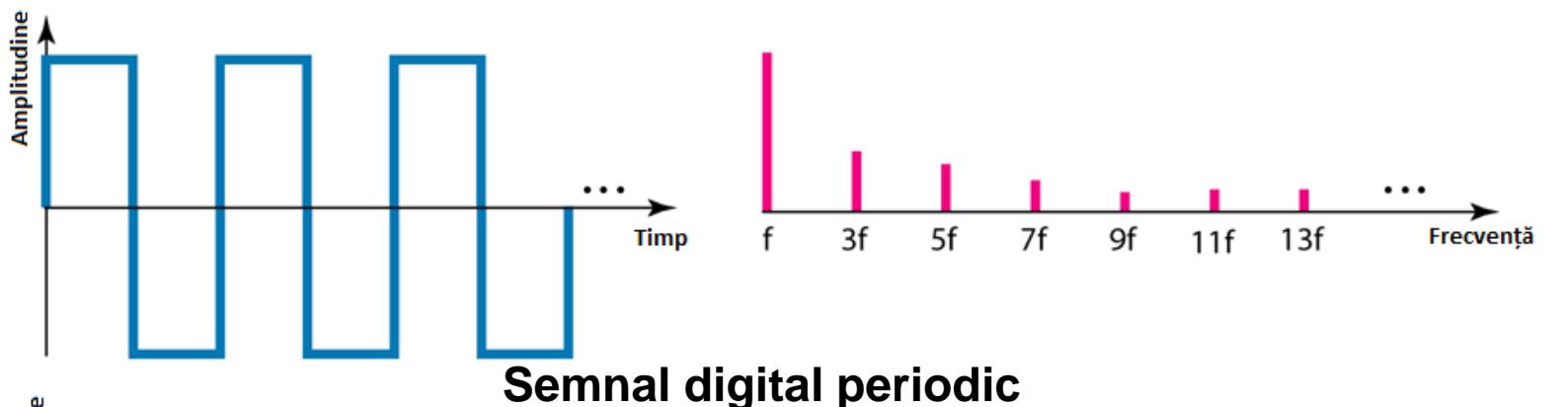
Rata de transfer reprezintă numărul de biți transmiși într-o secundă.

Lungimea bitului

Lungimea bitului este un concept similar cu lungimea de undă definită în cazul semnalelor analogice. Lungimea bitului reprezintă distanța ocupată de un bit pe mediul de transmisie.

$$\text{Lungime bit} = \text{viteza de propagare} \times \text{durată bit}$$

Conform analizei Fourier, un semnal digital este un semnal analogic compus, cu lățime de bandă infinită. Un semnal digital, în domeniul timp, este definit printr-o serie de segmente de dreaptă interconectate asezate orizontal sau vertical, făcându-se trecerea bruscă de la frecvență 0, la frecvență infinită și invers, ceea ce înseamnă că spectrul de frecvențe este infinit. În cazul unui semnal digital periodic spectrul este discret, iar în cazul unuia aperiodic, el este continuu.



Transmisia semnalelor digitale

Sunt utilizate două abordări: **baseband** și **broadband**.

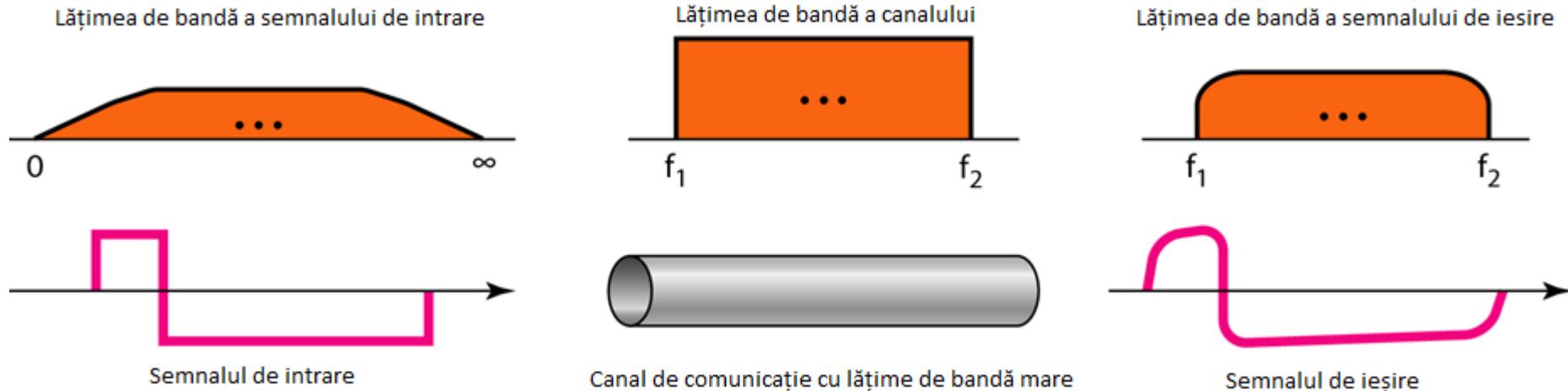
În cazul transmisiei baseband se trimit direct semnalul digital pe mediu, fără a fi convertit într-un semnal analogic. Această transmisie necesită un canal trece-jos, cu o lățime de bandă disponibilă ce începe de la 0 (canal dedicat pentru transmisia între două stații sau canal partajat între mai multe stații, unde doar două stații pot comunica la un moment dat). Pentru a putea trasmite semnalul nedistorsionat, avem nevoie de o lățime de bandă infinită a canalului. În practică, un astfel de canal nu există. Luăm în considerare două scenarii: transmisia baseband în cazul unui canal cu lățime de bandă mare și în cazul unui canal cu lățime de bandă limitată (îngustă).



Canal de transmisie trece-jos, cu o lățime de bandă mare



Canal de transmisie trece-jos, cu o lățime de bandă limitată



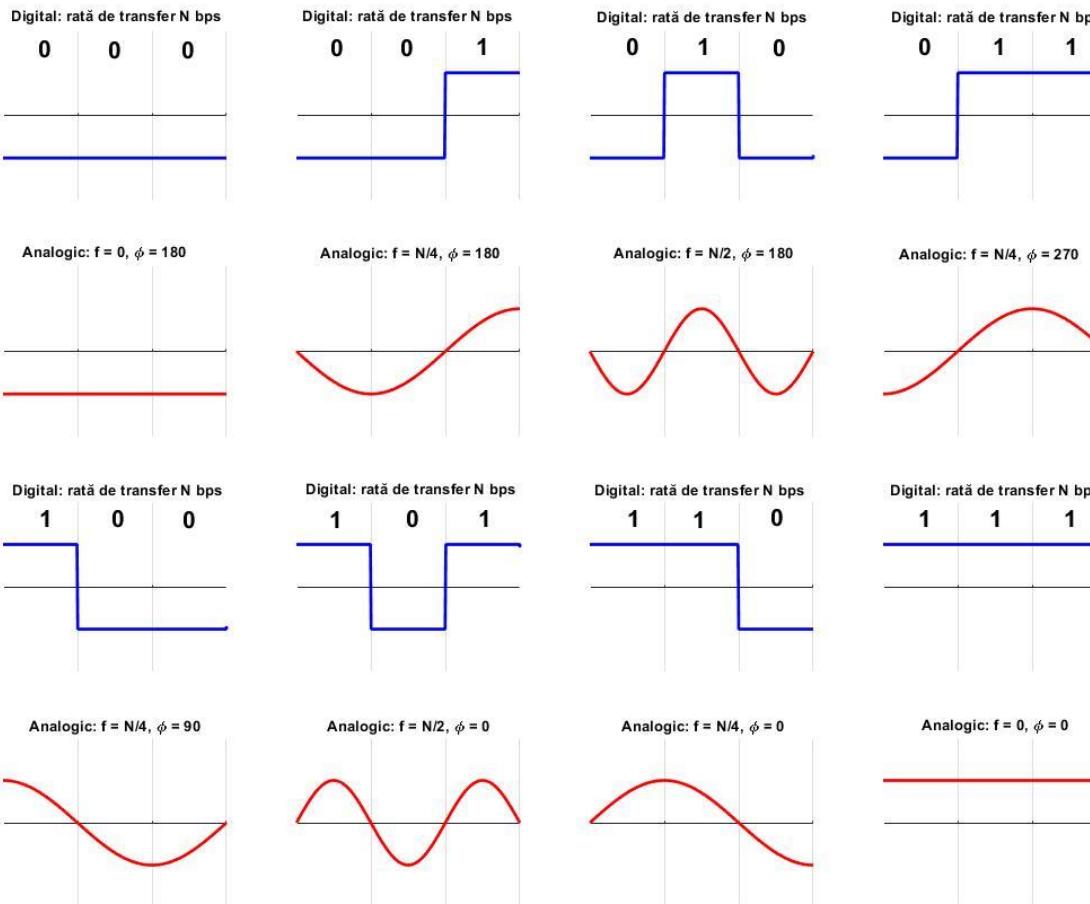
Transmisie baseband printr-un canal cu lățime de bandă mare

În cazul în care avem la dispoziție un canal cu lățime de bandă limitată, nu putem transmite semnalul digital, ci doar un semnal analogic ce aproximează semnalul digital.

O aproximare brută a semnalului digital presupune utilizarea primei armonici a semnalului, în cazul cel mai defavorabil. În consecință vom avea nevoie de o lățime de bandă egală cu $N/2$, unde N reprezintă rata de transfer a semnalului.

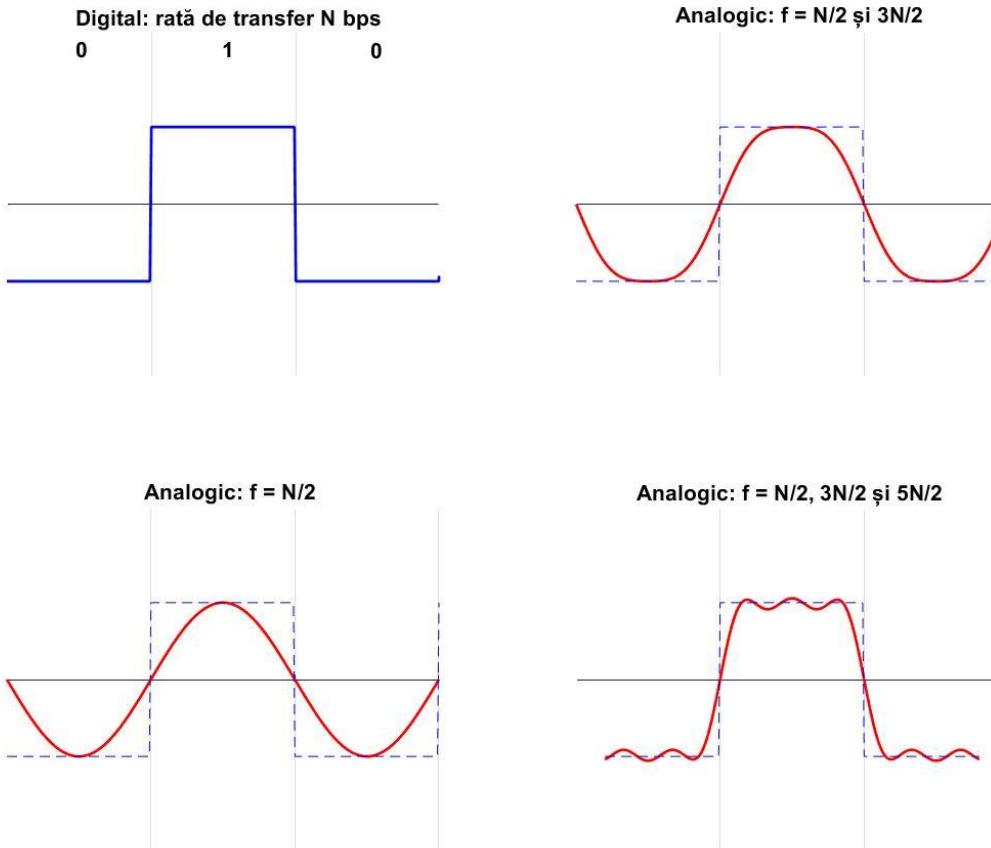
Aproximări mai bune, utilizează și a treia, respectiv și a cincea armonică, cazuri în care lățimile de bandă ale canalului trebuie să fie cel puțin $3N/2$, respectiv $5N/2$.

Date și semnale



Aproximație brută a semnalului digital utilizând prima armonică
în cazul cel mai defavorabil - lățimea de bandă necesară a canalului = $N/2$

Date și semnale



**Aproximarea semnalului digital utilizând până la 3 armonici –
lățimile de bandă necesare ale canalului: $N/2, 3N/2, 5N/2$**

Rată de transfer	Armonica: 1	Armonice: 1, 3	Armonice: 1,3,5
$N = 1 \text{ kbps}$	$B = 500 \text{ Hz}$	$B = 1,5 \text{ kHz}$	$B = 2,5 \text{ kHz}$
$N = 10 \text{ kbps}$	$B = 5 \text{ kHz}$	$B = 15 \text{ kHz}$	$B = 25 \text{ kHz}$
$N = 100 \text{ kbps}$	$B = 50 \text{ kHz}$	$B = 150 \text{ kHz}$	$B = 250 \text{ kHz}$

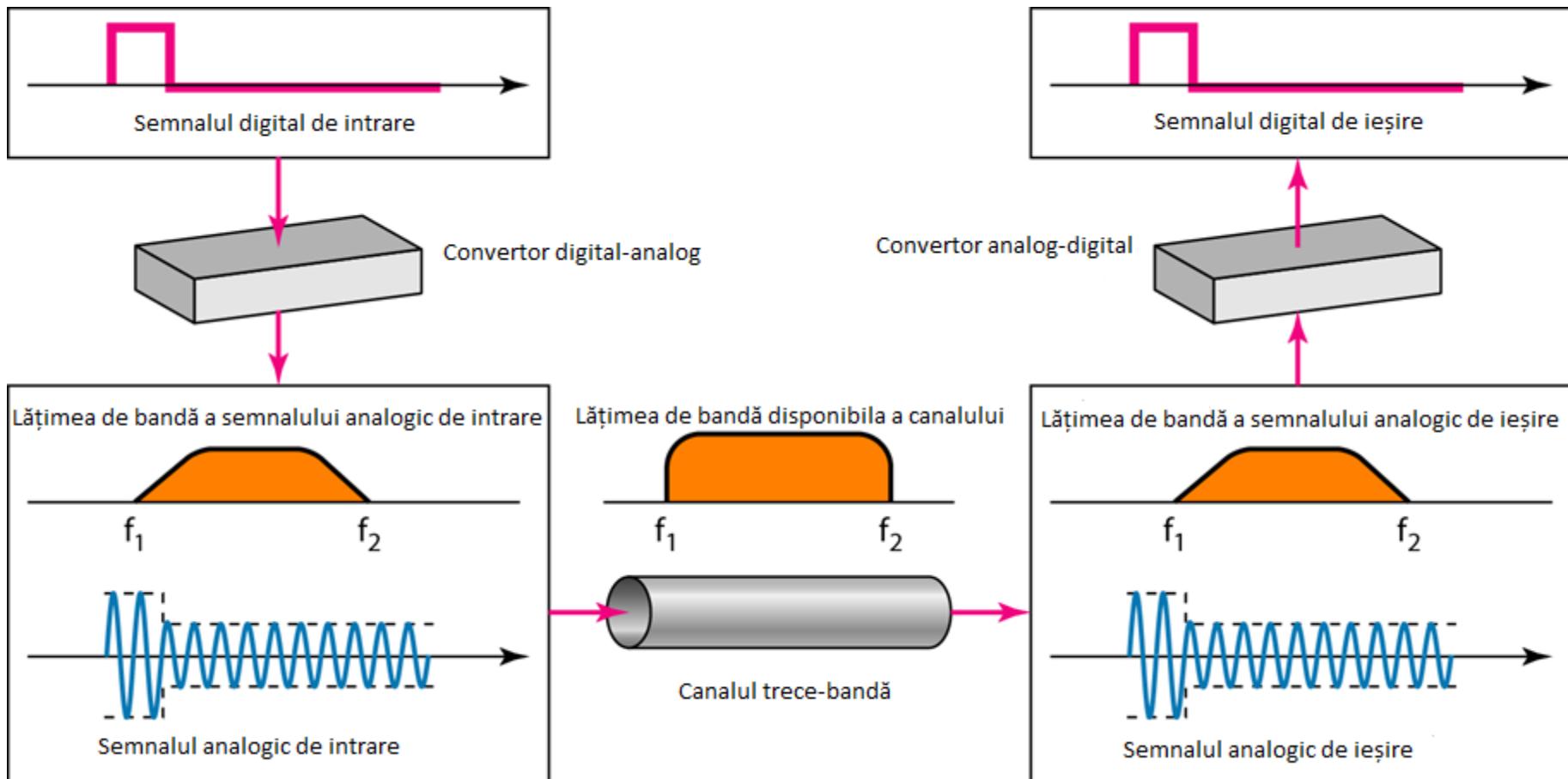
Necesarul de bandă

Transmisia broadband (modulația) presupune transformarea semnalului digital într-un semnal analogic. Aceasta permite utilizarea unui canal trece-bandă, a cărui lățime de bandă nu începe de la 0.



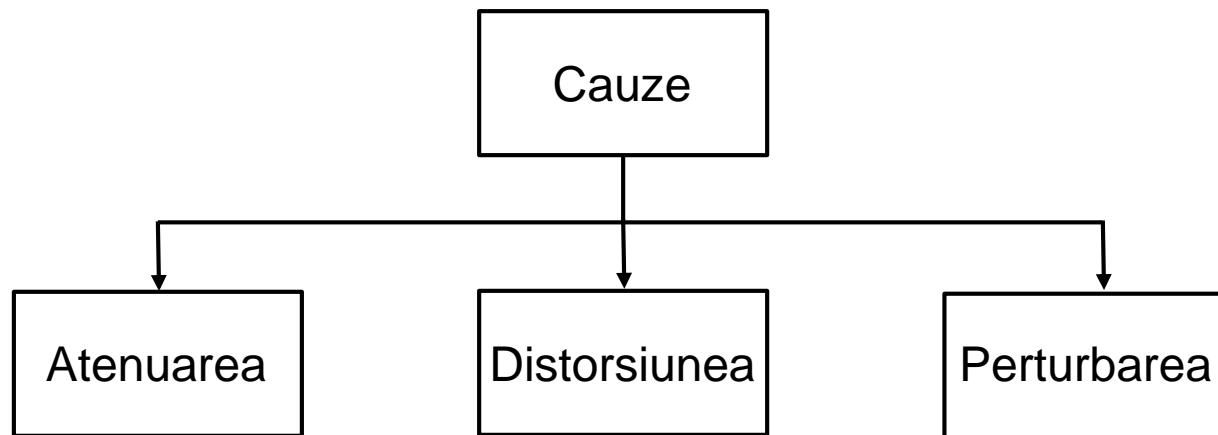
Canal de transmisie trece-bandă

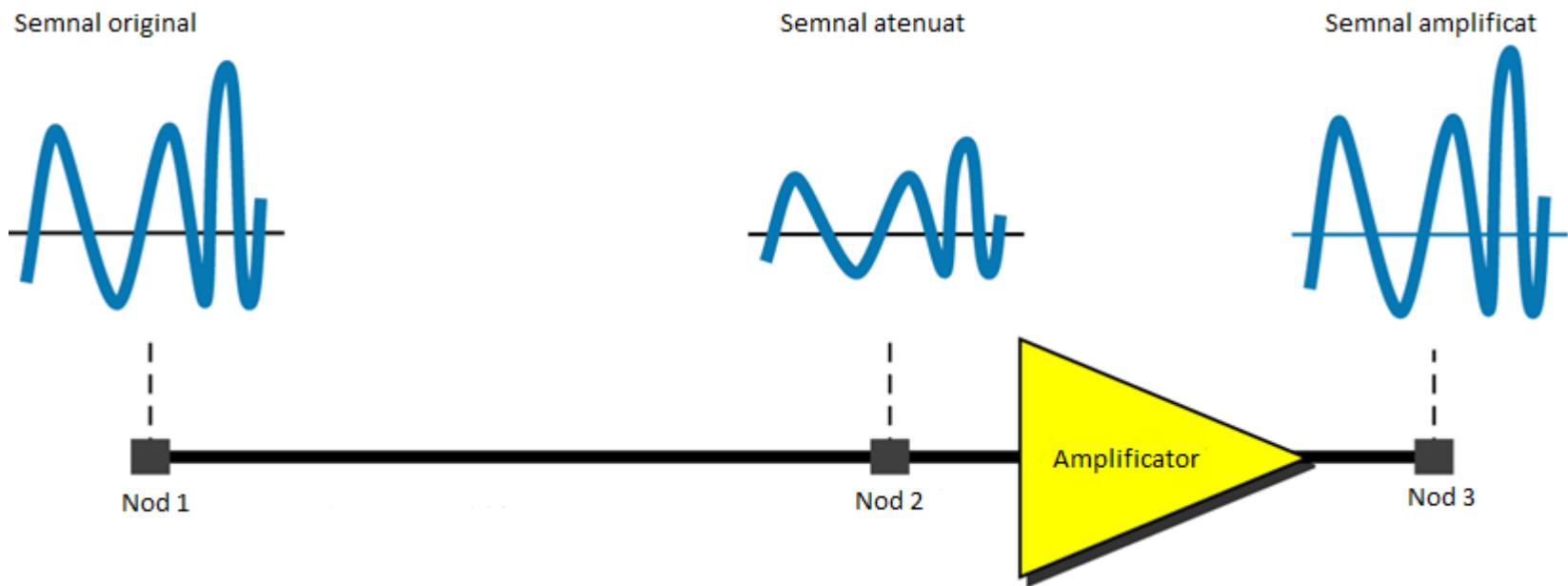
Date și semnale



Modulația semnalului pentru transmisia pe un canal trece-bandă

Deteriorarea semnalului la transmisie





Atenuarea semnalului

Date și semnale

Atenuarea semnalului

Atenuarea semnalului presupune pierderea energiei, datorată rezistenței mediului.

În anumite situații este necesară utilizarea unor amplificatoare pentru refacerea proprietăților semnalului.

Pierderea sau câștigul de energie se măsoară în decibeli:

$$\text{dB} = 10 \log_{10} P_2/P_1,$$

unde P_2 este puterea semnalului de ieșire, iar P_1 a celui de intrare

Uneori decibelii se folosesc și pentru măsurarea puterii semnalului în mW.

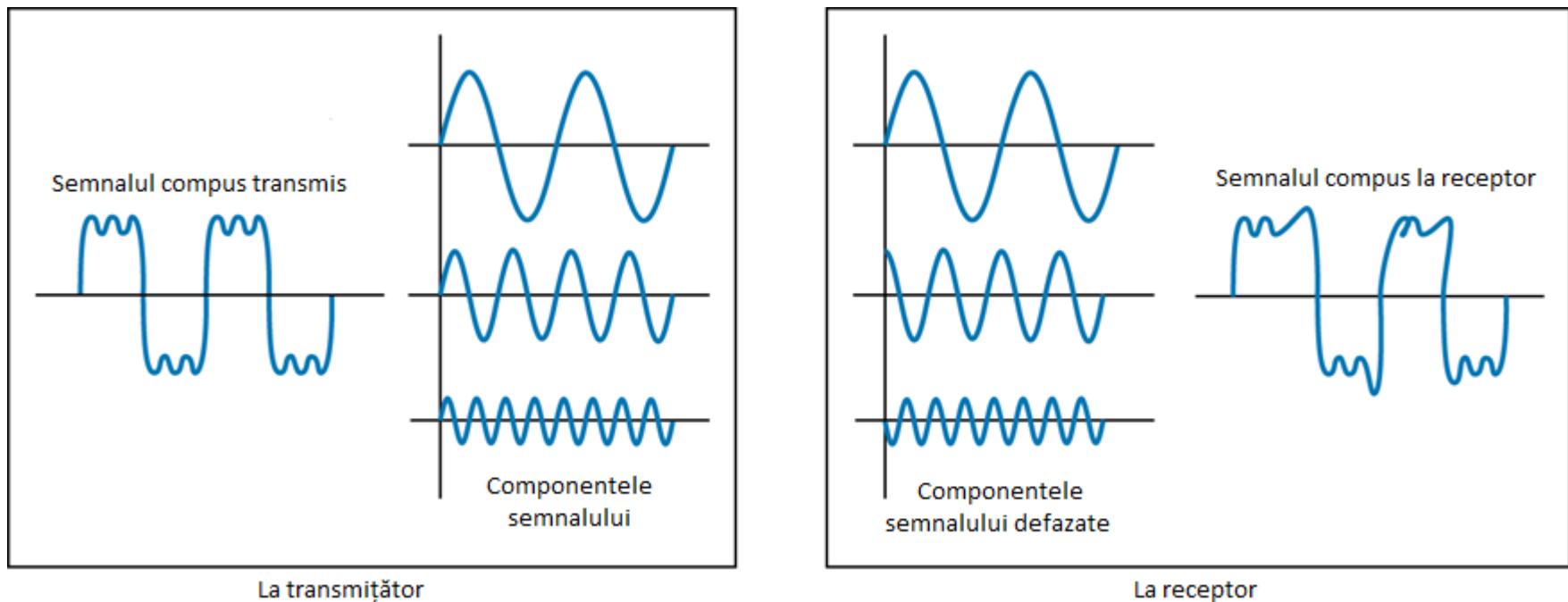
$$\text{dB}_m = 10 \log_{10} P_m,$$

unde P_m reprezintă puterea semnalului în mW.

Atenuarea semnalului pe cablu se definește în dB/km.

Distorsiunea semnalului

Distorsiunea presupune schimbarea formei semnalului. Ea apare în semnale compuse. Cum fiecare de frecvență diferită are o viteză de propagare diferită în mediu, componentele semnalului vor ajunge cu întârzieri diferite la receptor. Asta înseamnă că fazele semnalelor pot fi diferite la receptor, față de transmitemtor.



Distorsiunea semnalului

Perturbarea semnalului

Perturbarea presupune schimbarea formei semnalului, datorate suprapunerii acestuia cu un alt semnal de tip zgomot.

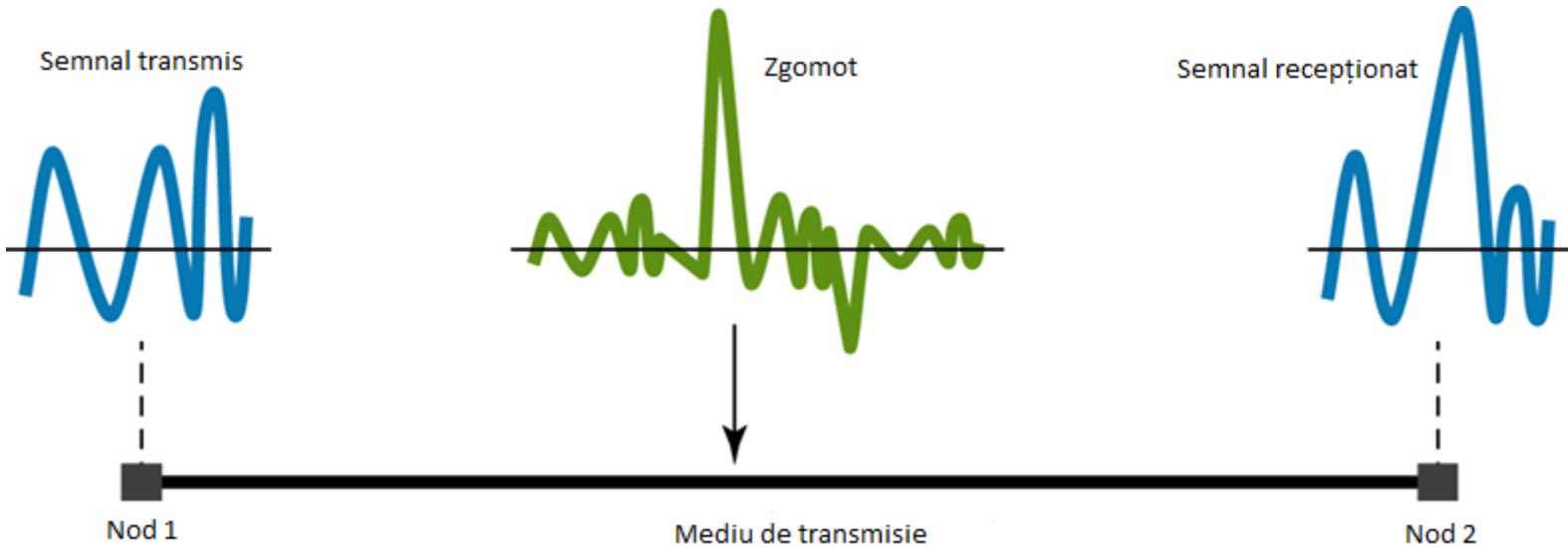
Există mai multe tipuri de semnale perturbatoare de tip zgomot:

- Zgomot termic – mișcarea aleatoare a electronilor în mediu de transmisie crează un semnal de tip zgomot;
- Zgomot inducție externă – mediul de transmisie acționează ca o antenă pentru diversele semnale externe;
- Crosstalk – zgomot inducție de la alte fire ale mediului de transmisie în altele.

Raportul semnal / zgomot (signal-to-noise ratio)

SNR - este un indicator de calitate a unui sistem ce descrie puterea semnalului util raportată la puterea zgomotului. În general se măsoară în decibeli (SNR_{dB}).

$$\begin{aligned}\text{SNR} &= \text{puterea medie a semnalului} / \text{puterea medie a zgomotului} \\ \text{SNR}_{\text{dB}} &= 10 \log_{10} \text{SNR}\end{aligned}$$



Perturbarea semnalului

Limita ratei de transfer a mediului

Rata de transfer a unui canal de comunicație depinde de:

- lățimea de bandă
- nivelele de semnal
- calitatea canalului (nivelul de zgomot)

Rata de transfer Nyquist (canal fără zgomot):

Rata de transfer = $2 \times$ lățimea de bandă $\times \log_2 L$,
unde L reprezintă numărul de nivele de semnal utilizate

Capacitatea Shannon (canal cu zgomot):

Capacitatea (rata de transfer) = lățimea de bandă $\times \log_2(1+SNR)$

Exerciții

1. Care este capacitatea de transmisie a unui canal ideal cu banda de frecvență de 3000 Hz, ce folosește 8 niveluri de semnalizare?
2. Trebuie să transmitem date cu o rată de transfer de 240 Kbps utilizând un canal cu o lățime de bandă de 20 KHz. De câte nivele de semnal avem nevoie? Cu cât ar crește rata de transfer, dacă am dubla numărul de nivele de semnal?
3. Care este capacitatea de transfer a unei linii telefonice, având frecvență de transmisie între 300 Hz și 3400 Hz și $\text{SNR}_{\text{dB}} = 30 \text{ dB}$?
4. O imagine TV se transmite de la o sursă care folosește o matrice de 480x500 de pixeli (elemente de imagine). Fiecare pixel poate avea una din cele 32 de intensități posibile. Se transmit 30 de imagini pe secundă. Care este debitul sau rata sursei? Dacă se folosește un canal cu banda de 4,5 Mhz și raportul $\text{SNR}_{\text{dB}} = 35 \text{ dB}$, poate fi făcută transmisia în timp real?
5. Având la dispoziție un canal cu o lățime de bandă de 1 MHz și SNR de 15, care este capacitatea maximă a canalului și care ar fi numărul de nivele de semnal pentru a asigura respectiva capacitate?

Indicatori de performanță

Bandwidth (capacitate/lățime de bandă digitală) = numărul de biți pe secundă ce pot fi transmiși pe un canal de comunicație (valoare teoretică). Lățimea de bandă digitală exprimată în biți pe secundă este direct proporțională cu lățimea de bandă (analogică) exprimată în hertz.

Throughput = valoarea reală (măsurată) a capacitatii de bandă (lățimea de bandă digitală definește o valoare ideală).

Goodput = cantitatea de date utile (pentru aplicație și implicit utilizator) din throughput.

Latență = timpul necesar unui mesaj să ajungă în întregime la destinație, de la momentul la care este transmis primul bit de către sursă. Latență are multiple componente, unele constante, altele variabile.

Latență = timpul de propagare + timpul de transmisie + timpul de procesare + timpul de așteptare în memoriile tampon + ...

Timpul de propagare = distanță până la destinație / viteza de transmisie

Timpul de transmisie = dimensiunea mesajului / lățimea de bandă

Jitter = viteza de variație a latenței

Un alt indicator de performanță utilizat este produsul **lățime de bandă x latență** ce definește numărul de biți ce umple un canal de date.

Conversia digital-digital

Este utilizată pentru reprezentarea datelor digitale sub formă de semnale digitale. Aceasta se realizează prin procesul de codare a liniei. Suplimentar, în anumite cazuri, înainte de operația de codare a liniei, datele sunt prelucrate, utilizându-se tehnici de codare a blocurilor, respectiv amestecare.

Codarea liniei

Presupune conversia unei secvențe de biți într-un semnal digital. La sursă, datele sunt codate în semnale digitale, la destinație semnalele sunt decodate, rezultând secvența de biți inițială.

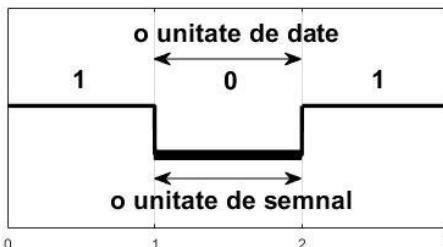
Aspecte de luat în calcul la definirea schemei de codificare

1. Maximizarea vitezei de transmisie (utilizarea eficientă a benzii de frecvență)

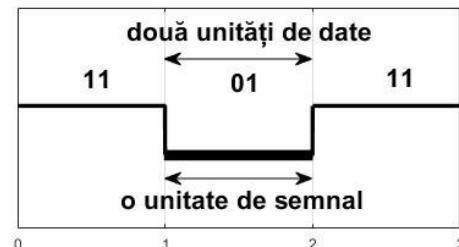
Unitatea de date este definită ca și cea mai mică entitate capabilă să reprezinte informația. În cazul datelor digitale, aceasta este bitul.

Unitatea de semnal (simbolul) reprezintă elementul cel mai mic de semnal (raportat la timp), purtător de informație.

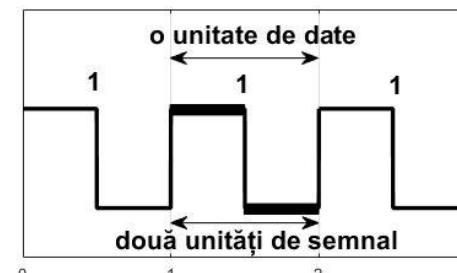
În comunicațiile de date, unitățile de date sunt trasferate pe baza unităților de semnal. Se definește rația r ca fiind numărul de elemente de date transportate de fiecare element de semnal.



o unitate de date / o unitate de semnal ($r=1$)



două unități de date / o unitate de semnal ($r=2$)



o unitate de date / două unități de semnal ($r=0.5$)

Exemple unități de date / unități de semnal

Rata de transfer a datelor se definește ca fiind numărul de unități de date transmise într-o secundă și se măsoară în biți pe secundă (bps).

Rata de transfer a semnalelor (rată de modulație) se definește ca și numărul de unități de semnal transmise într-o secundă și se măsoară în baud(s).

Scopul final în transmisiile de date este să creștem rata de transfer a datelor (creșterea vitezei de transmisie), scazând rata de transfer a semnalelor (scădere la latimii de bandă necesară).

2. Evitarea devierii valorii de referință

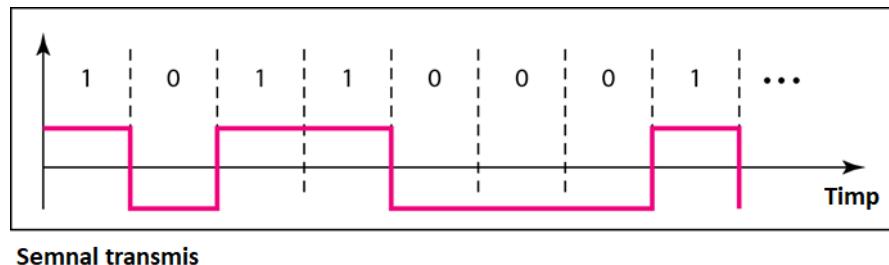
Pentru decodarea semnalului, receptorul calculează continuu valoarea medie a puterii semnalului primit. Această medie se numește valoare de referință (baseline). Puterea semnalului la un moment dat este comparată cu cea de referință pentru determinarea valorii unității de date (bitului). Un sir lung de 0 sau 1, transmis pe canal, determină o deviere a valorii medii, ceea ce duce la îngreunarea decodării.

3. Eliminarea componentelor de curent continuu

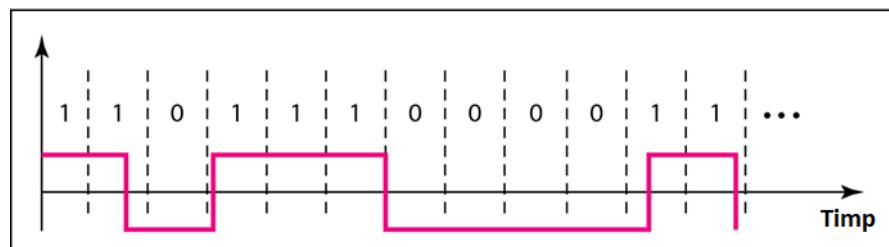
Când nivelul tensiunii semnalului rămâne constant pentru o perioadă de timp, în spectrul de frecvențe al acestuia apar componente de frecvență joasă. Aceste componente de frecvență apropiată de 0, denumite componente de curent continuu, determină apariția unor probleme de recepție și interpretare incorectă a semnalelor, atunci când sistemul de transmisie nu permite trecerea lor (filtre trece-jos, cuplare prin transformatoare, etc).

4. Evitarea apariției problemelor de sincronizare

În cazul transmisiilor sincrone, pentru interpretarea corectă a semnalului recepționat, tactul receptorului trebuie sincronizat în mod constant cu tactul transmițătorului. Sincronizarea este facilitată de transmisia de secvențe de date ce determină tranziții cât mai numeroase ale valorii semnalului. Este de evitat menținerea unei valori constante a semnalului pentru perioade de timp suficient de lungi, astfel încât să apară desincronizări.



Semnal transmis



Semnal recepționat

Efectele desincronizării tactului

4. Implementarea unui mecanism de detectie a erorilor

Un avantaj al schemei de codare l-ar reprezenta existența unui mecanism de detectie a erorilor de transmisie.

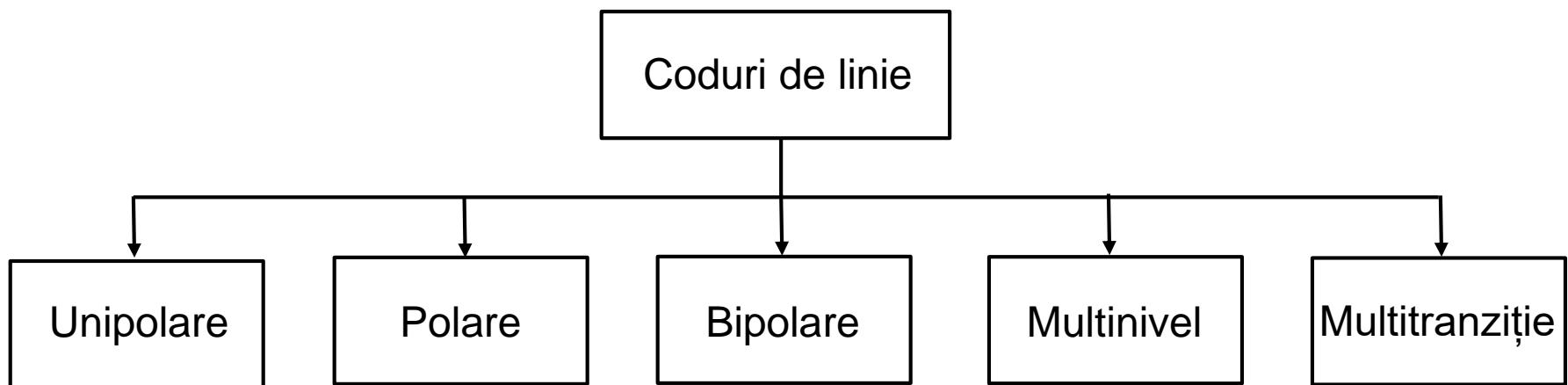
5. Imunitatea la zgomote

Robustetea la anumite tipuri de zgomote este un alt avantaj al unei scheme de codare.

6. Menținerea unui nivel acceptabil de complexitate

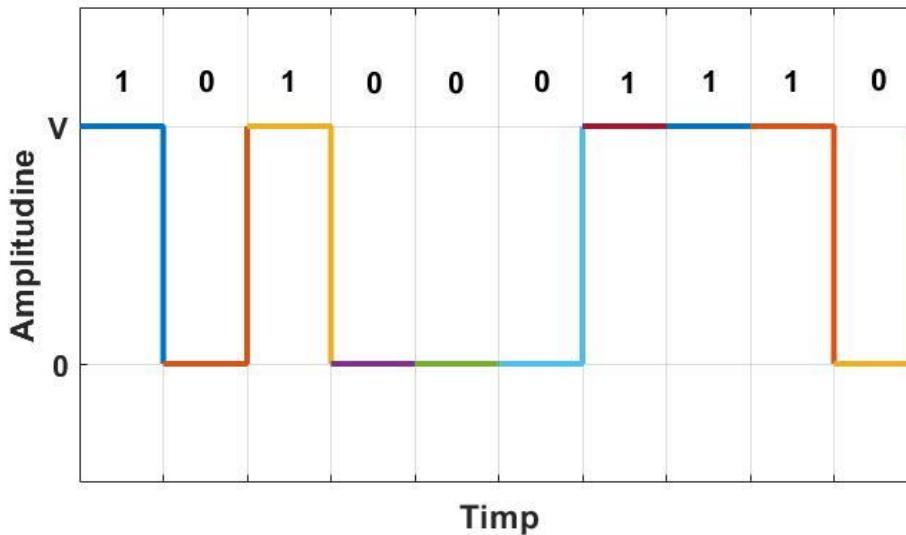
Creșterea complexității unei scheme de codare determină o creștere automată a costului de implementare.

Clasificarea schemelor de codare de linie



Scheme de cod unipolare

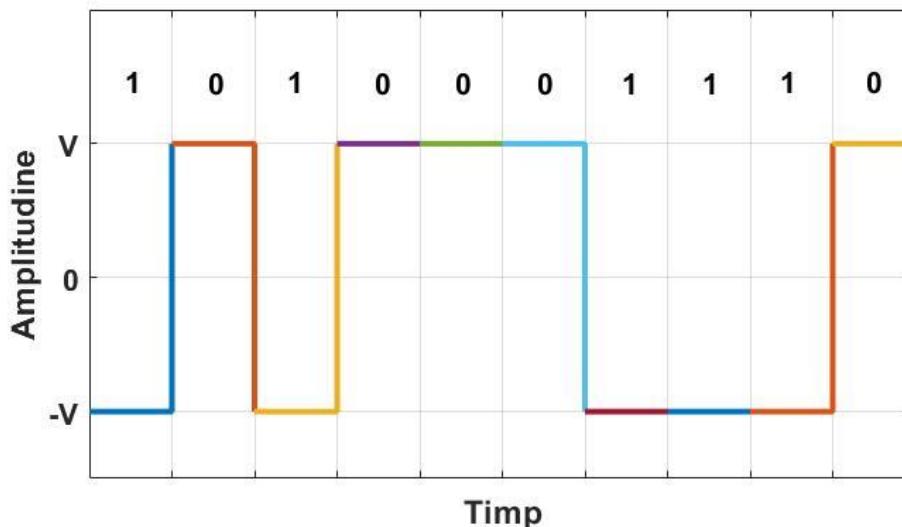
Într-o schemă unipolară, toate nivelele de semnal se află de o parte a axei timpului, fie deasupra, fie dedesubtul acesteia. Un exemplu de schemă unipolară este una din variantele NRZ (non-return-to-zero), în care valoarea pozitivă a tensiunii definește bitul 1, iar valoarea zero a tensiunii definește bitul 0. Schema se numește NRZ deoarece semnalul nu revine la 0 la mijlocul bitului.



Principalul avantaj al acestei scheme îl reprezintă simplitatea, însă are o serie de dezavantaje: devierea valorii de referință, desincronizarea tactului în cazul unor sevențe mai lungi de 0 sau 1, componente de curent continuu, puterea necesară mare, ce o fac neutilizabilă în practică.

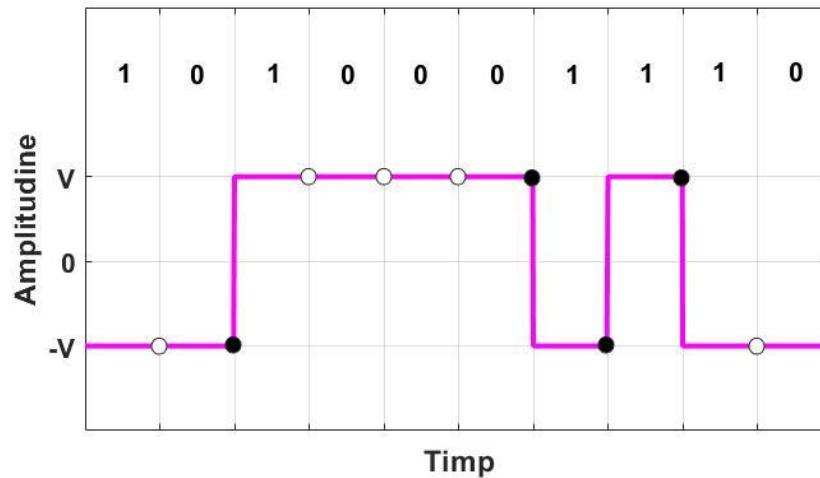
Scheme de cod polare

Într-o schemă polară, nivelele de semnal se află de o parte și de alta a axei timpului. Figura de mai jos prezintă o variantă polară a schemei NRZ, în care nivelul applitudinii este pozitiv pentru 0 și negativ pentru 1.



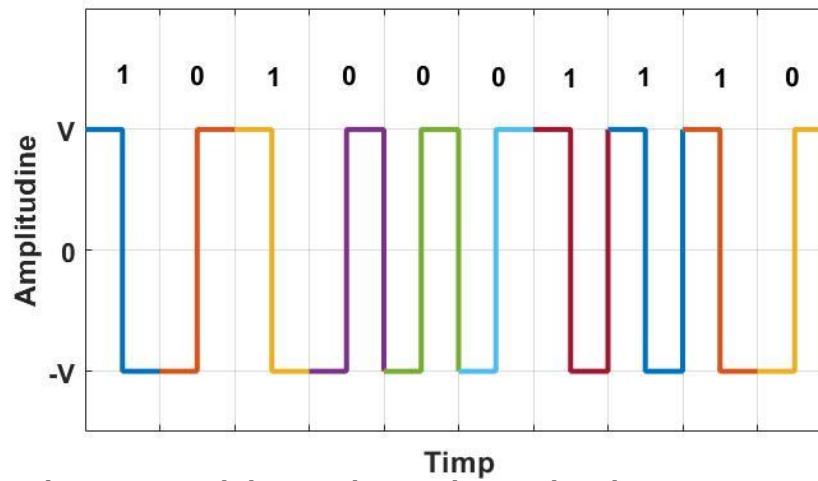
Avantajul variantei polare a NRZ, față de cea unipolară este dat de puterea necesară mai mică pentru transmisie. Estimarea se face pe baza puterii normalize (puterea necesară pentru transmisia unui bit / unitatea de rezistență a liniei). În primul caz, considerând nivelele de tensiune 0 V și 2 V, avem $P_N = \frac{1}{2}(2)^2 + \frac{1}{2}(0)^2 = 2$. În cel de-al doilea caz, considerând nivelele de tensiune -1 V și 1 V, avem $P_N = \frac{1}{2}(1)^2 + \frac{1}{2}(-1)^2 = 1$.

O altă variantă polară a schemei NRZ, denumită NRZ-I (inversată), este prezentată în figura de mai jos.



În acest caz, o variație a nivelului amplitudinii denotă transmisia unui bit de 1, în timp ce menținerea valorii anterioare a nivelului denotă transmisia unui bit de 0. Față de varianta anterioară, NRZ-I poate ajunge la o desincronizare de tact doar în cazul unui sir lung de biți de 0. Similar devierea valorii de referință se întamplă doar în cazul sirului lung de biți de 0.

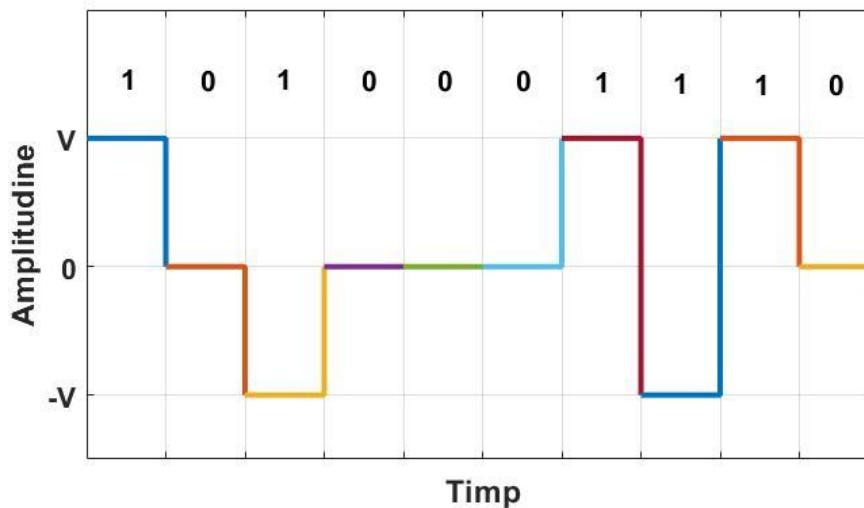
Ultima schemă polară prezentată este Manchester. Aici durata unui bit este împărțită în două, transmițându-se două unități de semnal pentru fiecare bit. Amplitudinea rămâne la un nivel pentru prima jumătate și sare pe al doilea nivel în a doua jumătate. Tranzitia de la mijlocul bitului asigură sincronizarea și determină valoarea acestuia. În exemplul prezentat în figura de mai jos transmisia unui bit de 0 este marcată de frontul crescător al impulsului, în timp ce transmisia unui bit de 1 este marcată de frontul descrescător al impulsului.



Această schemă rezolvă problemele de deviere a valorii de referință, de desincronizare a tactului în cazul unor secvențe mai lungi de 0 sau 1 și elimină componente de curent continuu. Dezavantajul său este dat de dublarea ratei semnalului.

Scheme de cod bipolar

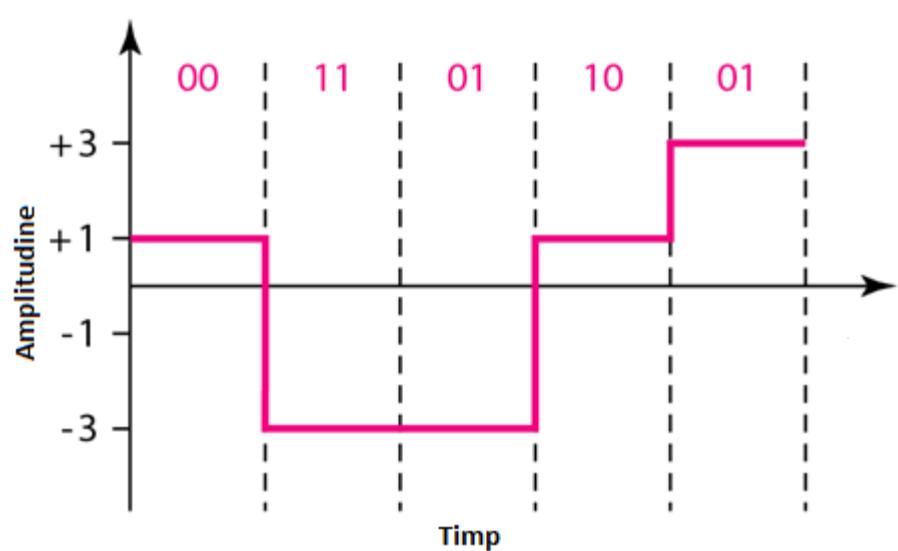
Schemele bipolare, denumite și binare multinivel utilizează 3 nivele de amplitudine: pozitiv, negativ și zero. Nivelul zero al semnalului definește o valoare a bitului, în timp ce cealaltă valoare este codată folosind celelalte două nivele de semnal, ce vor fi utilizate alternativ. Exemplul de mai jos prezintă schema de codare AMI (alternate mark inversion). Termenul mark este preluat din telegrafie și definește valoarea 1. În această schemă transmisia unui bit de 0 este semnalizată printr-un nivel 0 a semnalului, în timp ce pentru transmisia unui bit de 1 valoarea semnalului alternează între nivelul pozitiv și cel negativ.



Dezavantajul acestei scheme rămâne posibilitatea desincronizării în cazul transmisiiei unui sir lung de biți de 0.

Scheme de cod multinivel

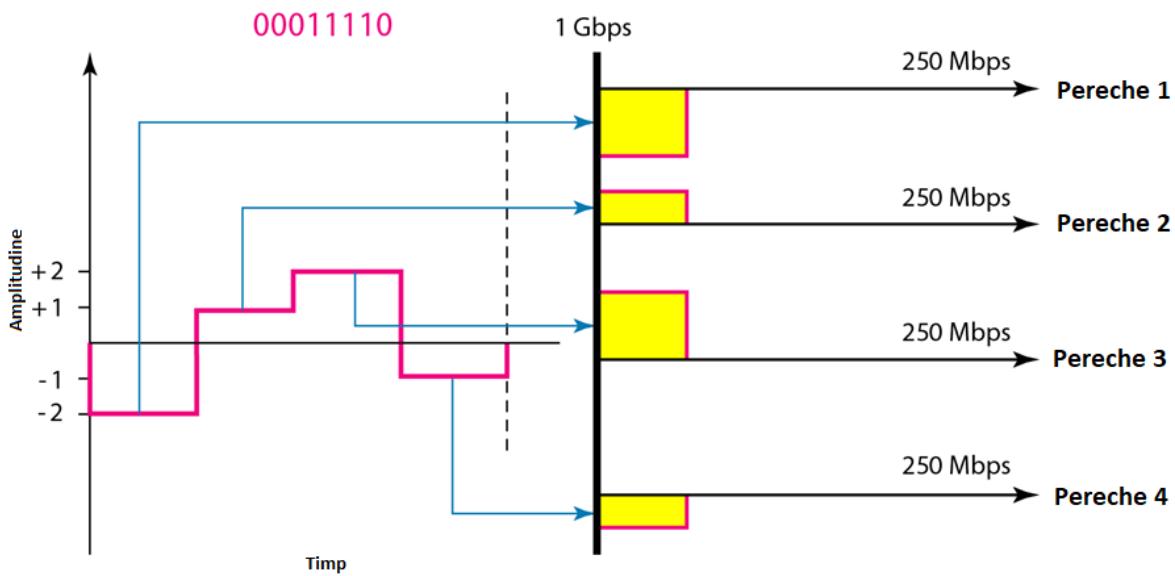
Dorința de creștere a ratei de transfer și/sau de scădere a lățimii de bandă utilizată a dus la apariția schemelor multinivel. Scopul este de creștere a numărului de biți / baud prin codarea unei secvențe de m biți într-o secvență de n unități de semnal. Un exemplu de astfel de schemă este 2B1Q (2 binary, 1 quaternary), prezentat mai jos.



	Valoare anterioară pozitivă	Valoare anterioară negativă
Biții următori	Următorul nivel	Următorul nivel
00	+1	-1
01	+3	-3
10	-1	+1
11	-3	+3

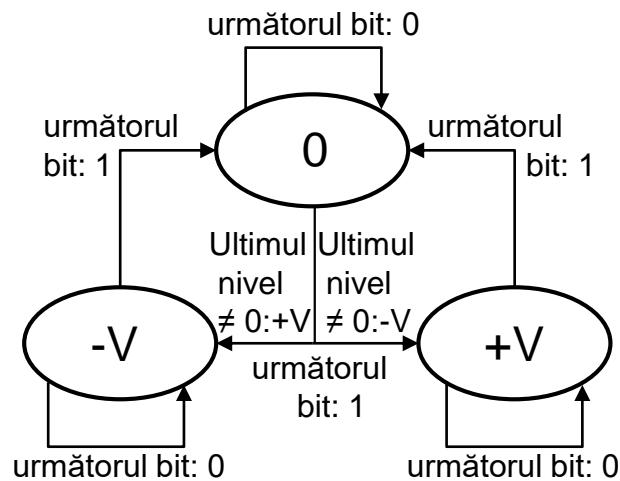
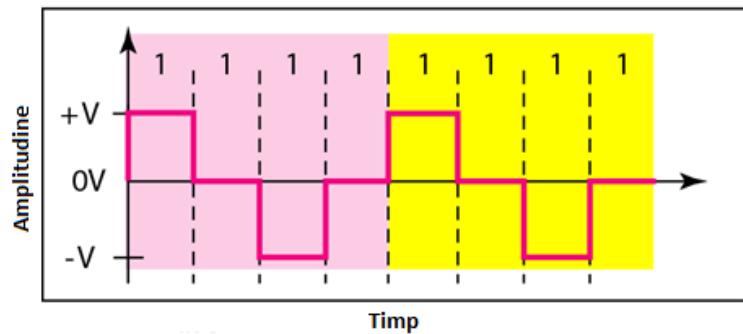
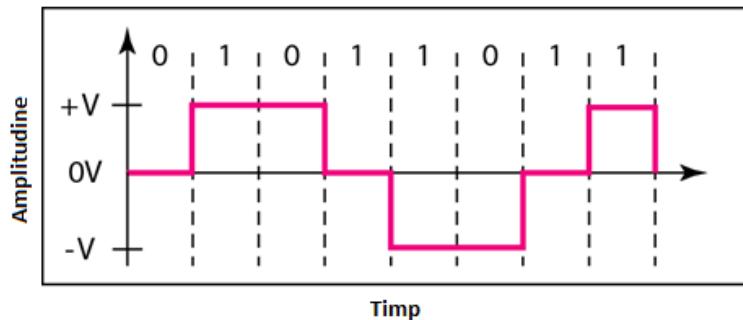
2B1Q folosește secvențe de doi biți, codate sub forma unui singur element de semnal având patru nivele de amplitudine. Tabelul de tranziții este prezentat alăturat schemei. Dezavantajul este unul comun acestei categorii: creșterea numărului de nivele duce la creșterea complexității, echipamentele trebuind să fie capabile să discearnă între aceste nivele.

O altă variantă multnivel este 4D-PAM5 (4 dimensional five level pulse amplitude modulation). 4D se referă la faptul că semnalul se transmite pe 4 canale de comunicație în paralel. Sunt utilizate 5 nivele de semnal, dar 0 este doar pentru transmisia detecției unei erori. Dacă ignorăm faptul că semnalele se transmit pe 4 canale în paralel, schema codifică o secvență de 8 biți folosind un semnal cu 4 nivele. Fără a intra în amănunte un exemplu de codare este prezentat în figura de mai jos.



Scheme de cod multitranzitie

Astfel de scheme definesc diverse reguli de tranziție pentru semnal, atunci când avem mai mult de două nivele de amplitudine. Un exemplu il reprezintă MLT-3 prezentat mai jos, împreună cu mașina de stări utilizată pentru generarea semnalului. Această schemă, deși utilizează o unitate de semnal pentru fiecare unitate de date, reduce nevoia de bandă, generând semnal periodic pe portiuni.

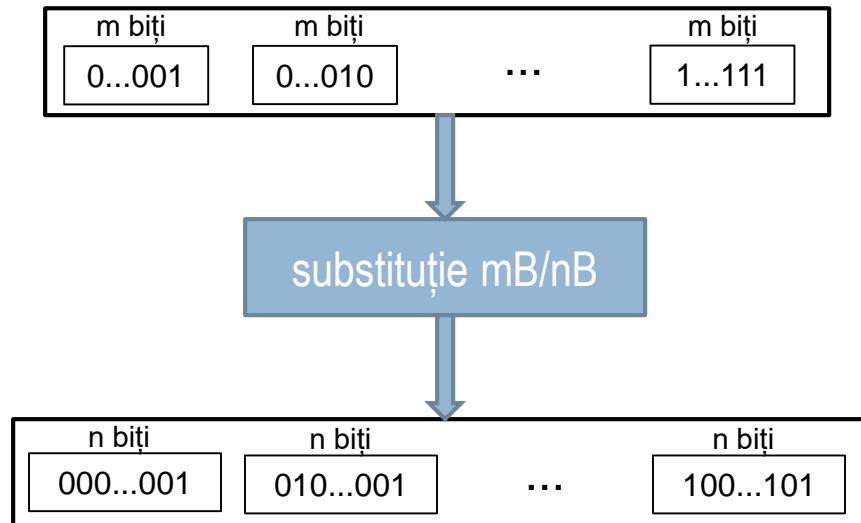


Sumar scheme de codare a liniiei

Categorie	Schemă	Utilizare	Dezavantaje
Unipolare	NRZ		putere necesară mare, desincronizare de tact la şiruri lungi de 1 și 0, deviere valoare referință, componente de CC
Polare	NRZ		desincronizare de tact la şiruri lungi de 1 și 0, deviere valoare referință, componente de CC
	NRZ-I	100 BaseFx, FDDI	desincronizare de tact la şiruri lungi de 0, deviere valoare referință, componente de CC
	Manchester	10Base5, 10Base2, 10BaseT, 10BaseFL	lătime de bandă ridicată
Bipolare	AMI	ADSL	desincronizare de tact la şiruri lungi de 0
Multinivel	2B1Q	ISDN, HDSL	creșterea complexității
	4D-PAM5	1000BaseT	creșterea complexității
Multitranziție	MLT-3	100BaseTx, 100BaseT4	creșterea complexității, desincronizare de tact la şiruri lungi de 0

Codarea blocurilor

Codarea blocurilor a fost introdusă ca o metodă de compensare a neajunsurilor codurilor de linie. Aceste coduri ne oferă o oarecare redundanță ce ne permite asigurarea sincronizării tactului și ne oferă o metodă rudimentară și neexhaustivă de detecție a erorilor de transmisie. În general, schemele de codare a blocurilor schimbă un bloc de m biți într-unul de n biți, unde $n > m$. Schemele de codare a blocurilor sunt denumite sub forma mB/nB . Procedeul de codare implică 3 pași: diviziune, substituție și combinare. În primul pas fluxul de biți este împărțit în grupuri de câte m biți. În pasul al doilea, aceste grupuri sunt substituite cu grupurile de n biți corespunzătoare, iar în ultimul pas grupurile de n biți sunt combinate pentru a forma un nou flux de biți.



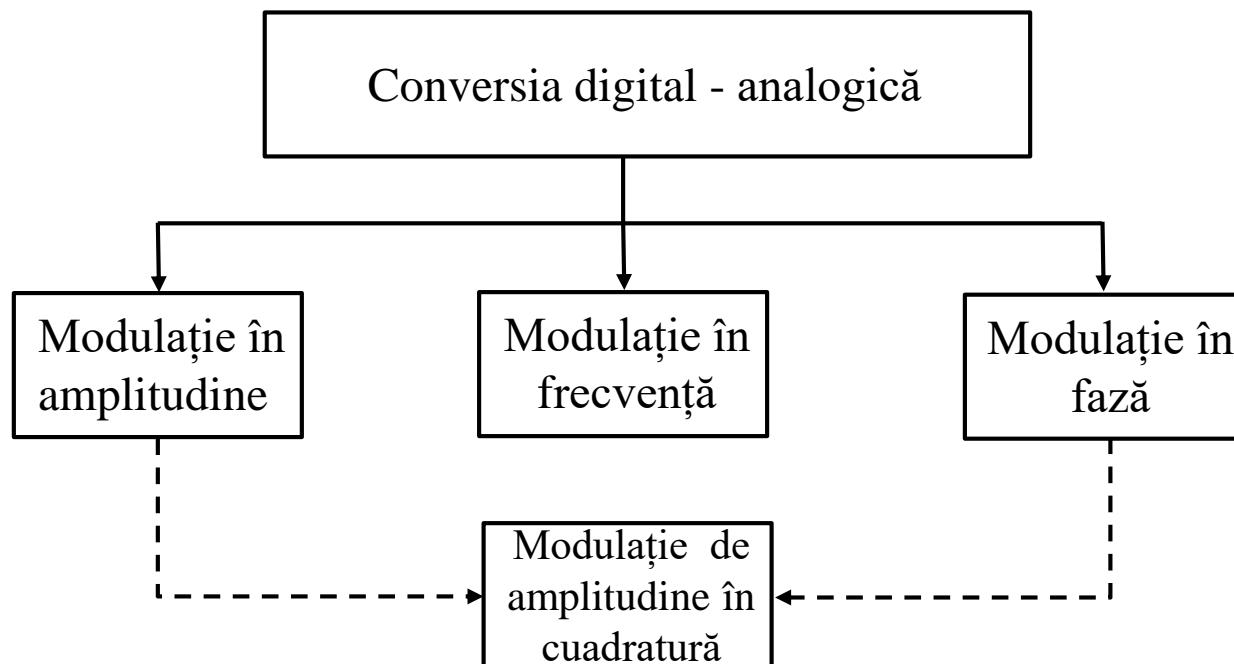
Schema de codare blocuri 4B/5B

A fost dezvoltată pentru a se utiliza în combinație cu NRZ-I. NRZ-I are o problemă de desincronizare atunci când este transmisă o secvență lungă de biți de 0. O soluție la această problemă constă în schimbarea secvenței de biți astfel încât să nu mai existe grupuri mari de biți de 0. 4B/5B rezolvă această problemă înlocuind fluxul de date initial cu unul ce conține diverse alternanțe de biți de 0 și 1. Orice secvență de 5 biți nu va avea mai mult de un 0 la început și doi la final. Cum sunt înlocuite 16 combinații de biți, având la dispoziție o plajă de 32 de variante de substituție, 16 variante vor rămâne de rezervă. Din acestea unele sunt folosite pentru cuvinte de control, iar altele nu sunt utilizate deloc. Aceste variante neutilizate asigură o oarecare detectie de erori, atunci când ele sunt identificate la destinație. Tabelul de mai jos prezintă variantele de substituție utilizate pentru datele utile.

4B	5B	4B	5B	4B	5B	4B	5B
0000	11110	0100	01010	1000	10010	1100	11010
0001	01001	0101	01011	1001	10011	1101	11011
0010	10100	0110	01110	1010	10110	1110	11100
0011	10101	0111	01111	1011	10111	1111	11101

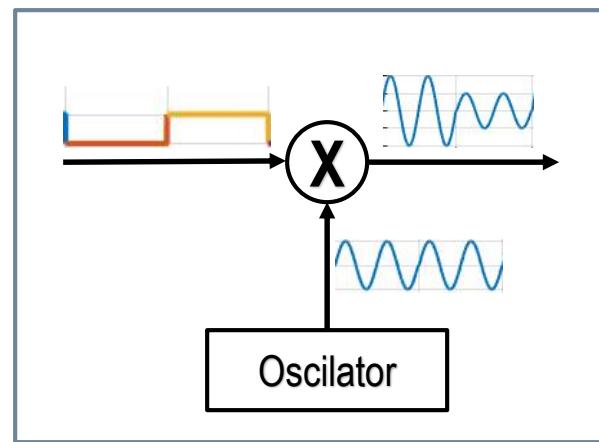
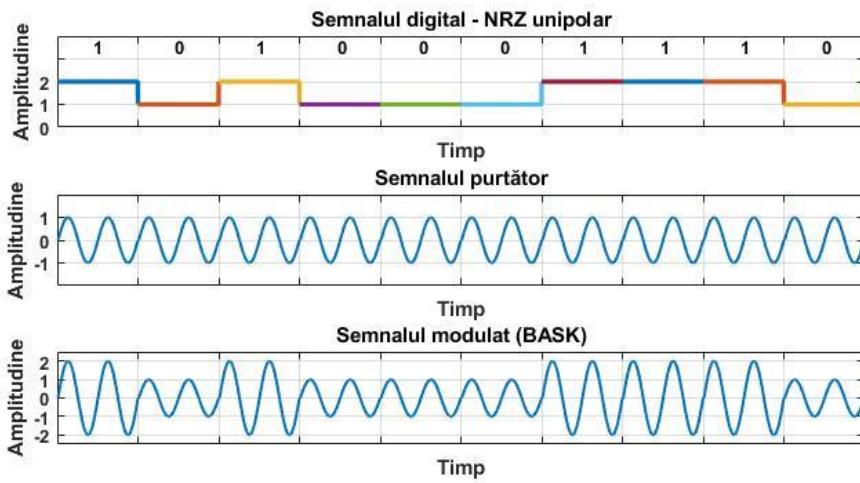
Conversia digital-analogică (modulația)

Este procesul de schimbare al caracteristicilor semnalului analogic pe baza informațiilor din datele sau semnalele digitale. Un semnal analogic periodic este definit de trei caracteristici: amplitudine, frecvență și fază. Când variem una sau mai multe dintre aceste caracteristici, se crează o nouă versiune a semnalului. Astfel, schimarea acestor caracteristici poate fi utilizată pentru reprezentarea informației digitale. Tipurile de conversii digital-analogice sunt prezentate mai jos:



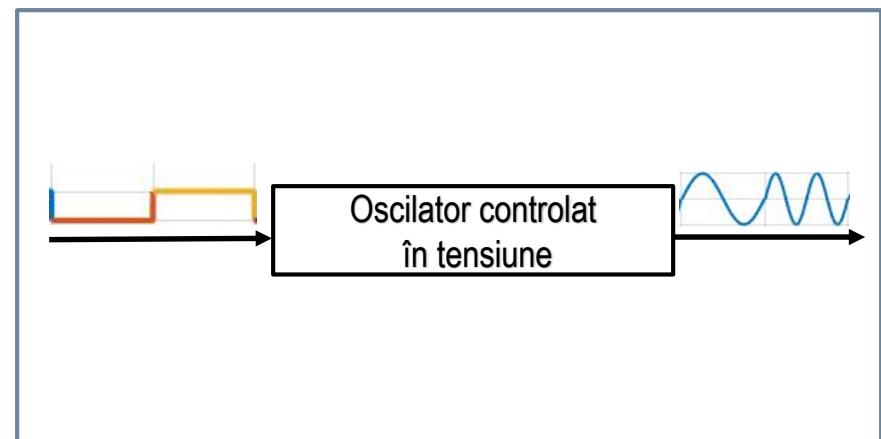
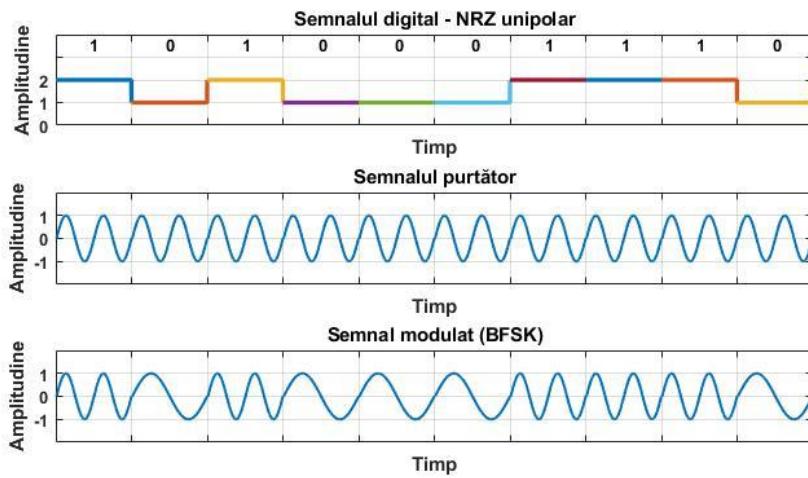
Modulația în amplitudine binară (BASK- binary amplitude shift keying)

Cu toate că putem avea mai multe tipuri de unități de semnal, fiecare având amplitudine maximă diferită, de obicei modulația în amplitudine folosește doar două tipuri (binară). Amplitudinea maximă a unuia dintre tipuri reprezintă 0 logic, iar amplitudinea maximă a celui de-al doilea reprezintă 1 logic. Figura din dreapta prezintă ideea din spatele implementării BASK.



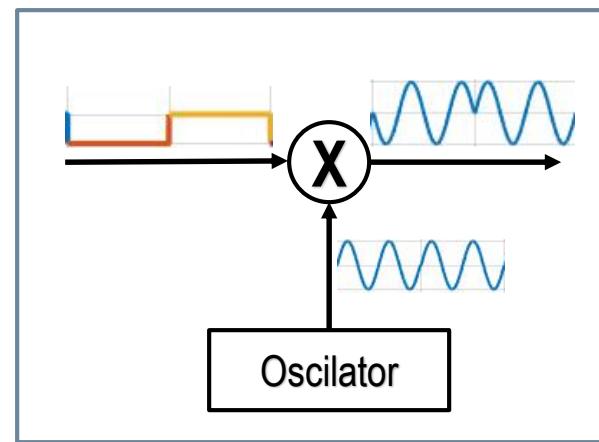
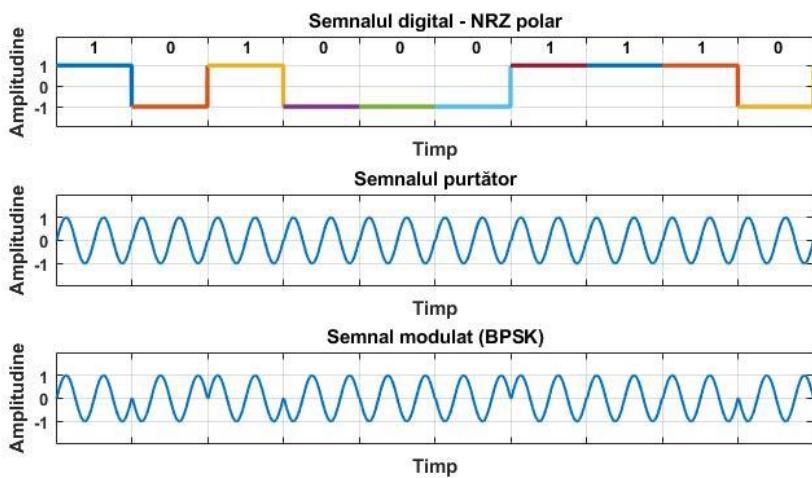
Modulația în frecvență binară (BFSK – binary frequency shift keying)

Atunci când se modifică frecvența semnalului analogic (purtător) pentru reprezentarea datelor, vorbim despre modulația în frecvență. BFSK utilizează doar două frecvențe diferite pentru a reprezenta valorile binare. Există mai multe modalități de implementare a BFSK, una din acestea utilizând un oscilator controlat în tensiune pentru generarea semnalului modulat.



Modulația în fază binară (BPSK - binary phase shift keying)

Cea de-a treia caracteristică a semnalului care se poate modifica pentru reprezentarea datelor digitale este faza. O primă variantă, simplă, de modulație în fază este cea binară, ce folosește un defazaj de 180 grade între semnalele ce reprezintă cele două valori binare.



Modulația în fază cuaternară (QPSK - quadrature phase shift keying)

O schemă de modulație în fază ce folosește mai eficient lățimea de bandă a canalului este cea cuaternară, ce folosește patru unități de semnal defazate (de ex. 45, 135, -45 și -135 grade) pentru a transmite câte doi biți pe simbol.

Pentru generarea semnalului, QPSK utilizează două modulații BPSK separate, având semnalele purtătoare în cuadratură. Semnalele în cuadratură, denumite și semnale IQ sunt des utilizate în scheme de modulație complexe. O pereche de semnale periodice sunt în cuadratură atunci când sunt defazate cu 90 de grade. Semnalul "în fază", sau de referință, se notează cu I, iar semnalul defazat, sau "în cuadratură" se notează cu Q. În primul pas, biții sunt trecuți prin un convertor serial-paralel (SP), ce trimit un bit unui modulator BPSK și următorul bit celuilalt modulator. Implementarea modulatorului QPSK este prezentată pe diapozitivul următor.

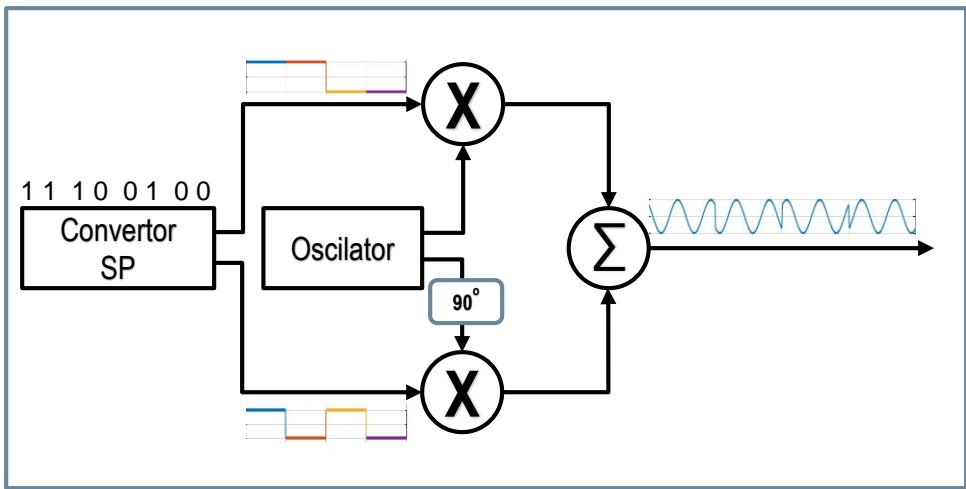
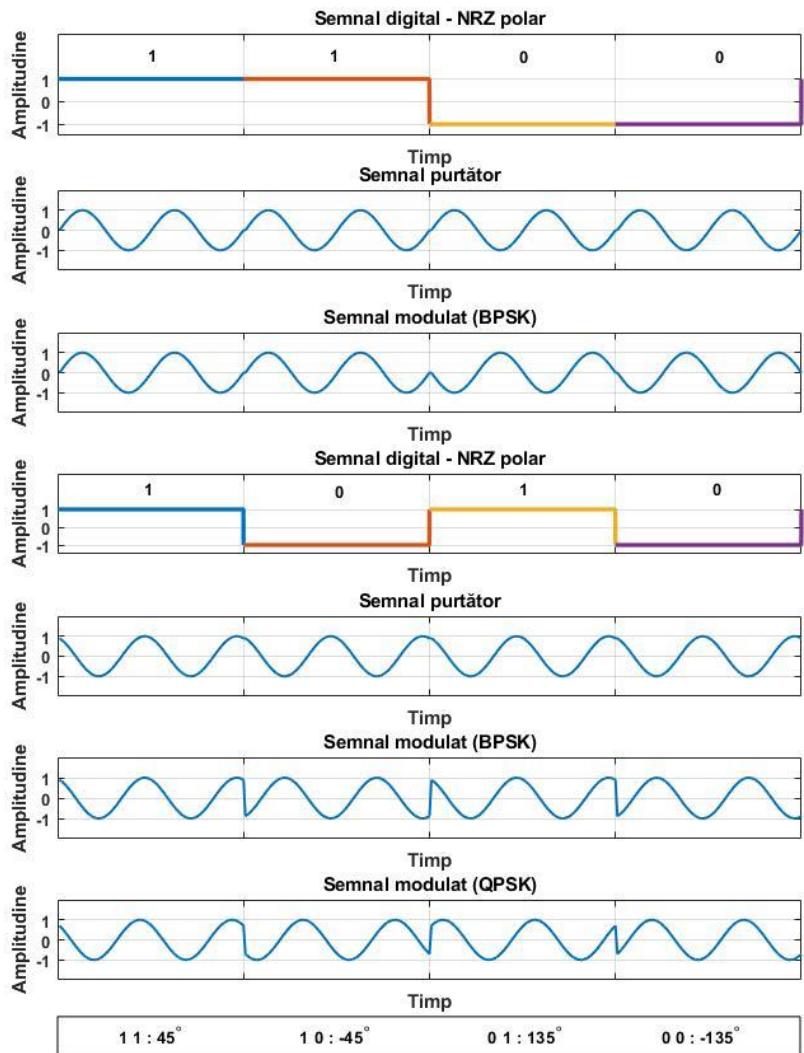
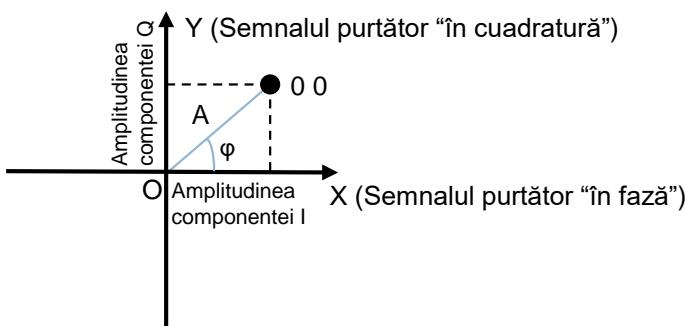
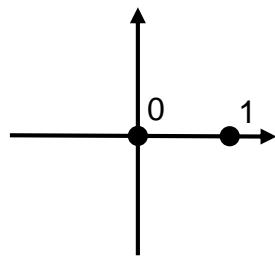


Diagrama constelație

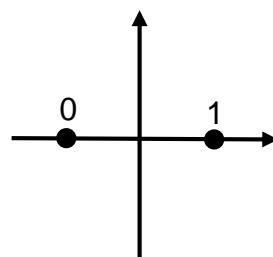
O diagramă constelație ne poate ajuta să definim amplitudinea și fază unei unități de semnal, mai ales atunci când utilizăm două semnale purtătoare: unul “în fază” și celălalt “în cuadratură”. Într-o astfel de diagramă un tip de unitate de semnal este reprezentat printr-un punct. Bitul sau combinația de biți transportați de unitatea de semnal sunt de obicei prezențați alături punctului.



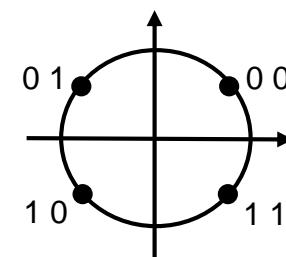
Exemple de constelații



a) BASK



b) BPSK

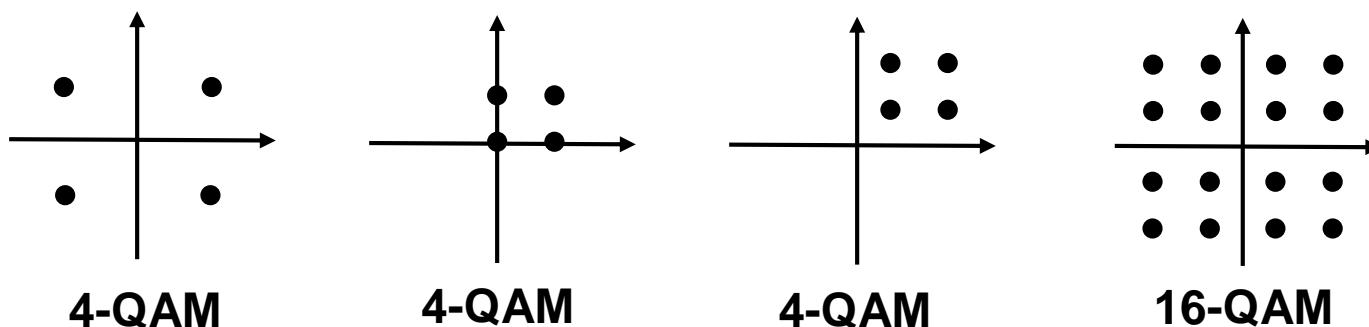


c) QPSK

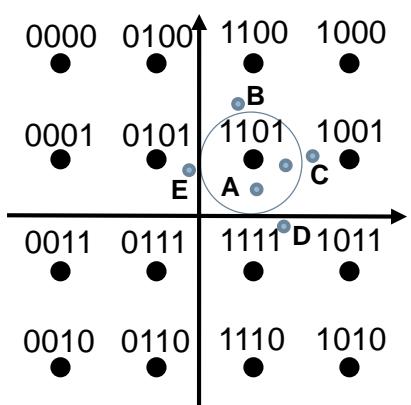
- a) În cazul BASK, avem doar semnal purtător “în fază”. În consecință punctele se vor afla pe axa OX. 0 logic corespunde unei unități de semnal cu amplitudine 0, 1 logic corespunde unei unități de semnal cu o amplitudine pozitivă.
- b) Și BPSK utilizează doar un semnal purtător “în fază”, însă cum semnalul modulator este polar, vom obține un semnal modulat având două tipuri de unități de semnal, una cu amplitudine pozitivă și una cu amplitudine negativă, adică defazate cu 180 de grade.
- c) QPSK utilizează două semnale purtător, unul “în fază” și celălalt “în quadratură”. Spre exemplu, unitatea de semnal asociată bițiilor 1 1 este obținută prin combinarea celor două semnale, având ambele amplitudine pozitivă egală, astfel că punctul asociat acestuia se află în primul cadran, având un defazaj ϕ de 45 de grade și o amplitudine A pozitivă.

Modulația de amplitudine în cuadratură(QAM - quadrature amplitude modulation)

Creșterea numărului de unități de semnal distincte este limitată la modulația în fază de capacitatea echipamentelor de a distinge diferențele mici de fază. Pentru a crește în continuare capacitatea de transfer, prin creșterea numărului de unități de semnal distincte, putem altera mai mult o caracteristică a semnalului în același timp. Cum faza și frecvența semnalului sunt interdependente (frecvența reprezintă viteza de variație a fazei – prima derivată), rămâne posibilitatea combinării ASK cu PSK. Ideea utilizării a două semnale purtătoare, unul în fază și celălalt în cuadratură, având nivele de amplitudine diferite pentru fiecare semnal purtător stă la baza QAM. Există multe variațiuni de QAM posibile, câteva dintre acestea fiind prezentate mai jos.



Constelațiile QAM date ca exemplu nu prezintă modul de atribuire a bițiilor la nivel de unități de semnal. Când facem această asociere, trebuie avut în vedere ca o variație relativ mică a semnalului, datorată zgomotului, să nu ducă la erori de detectie de mai mulți biți. Acest lucru se poate întâmpla dacă asociem valori consecutive de biți, unităților de semnal adiacente. Spre exemplu, pentru 16-QAM, dacă unei unități de semnal ii asociem 0111 și unei unități învecinate ii asociem 1000, în cazul în care receptorul alege din greșală unitatea învecinată, în locul celei corecte, toți biții identificați vor fi gresiți. O soluție mai bună este să alegem biții, în aşa fel încât unitățile adiacente să difere doar printr-un bit. Acest tip de asociere se numește cod Gray. Un astfel de exemplu este prezentat în continuare.



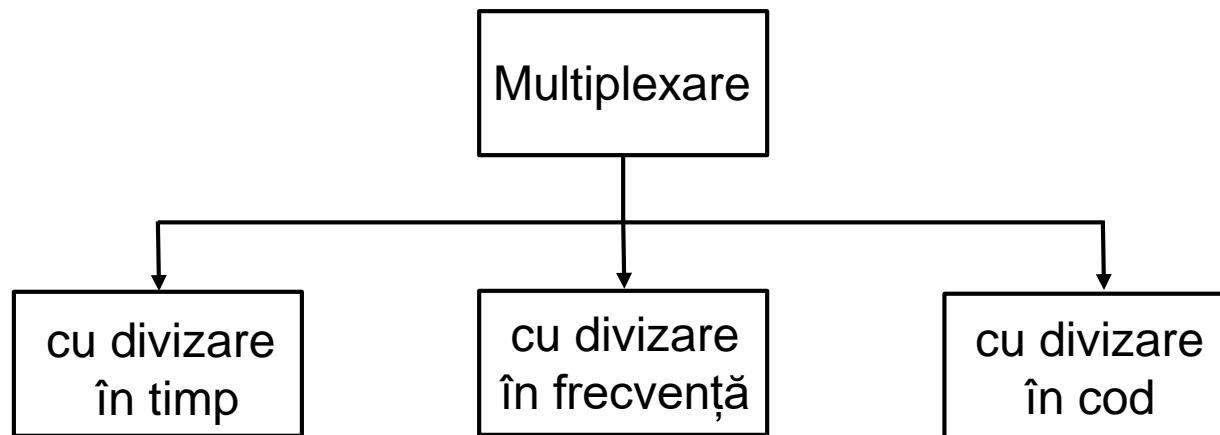
Punct	Valoare	Erori biți
A	1101	0
B	110 <u>0</u>	1
C	<u>1</u> 001	1
D	1 <u>1</u> 11	1
E	<u>0</u> 101	1

Utilizarea eficientă a lățimii de bandă disponibilă prin multiplexare

Canalele de comunicație reale au capacitați de bandă limitate. Utilizarea eficientă a capacitații de bandă reprezintă un obiectiv important în transmisiile de date. Multiplexarea presupune combinarea mai multor canale logice (transmisii de date) ce necesită o lățime de bandă îngustă în vederea utilizării unui procent cât mai mare din lățimea de bandă disponibilă a unui canal fizic. În acest fel, canalul este partajat și utilizat în același timp de mai multe echipamente.

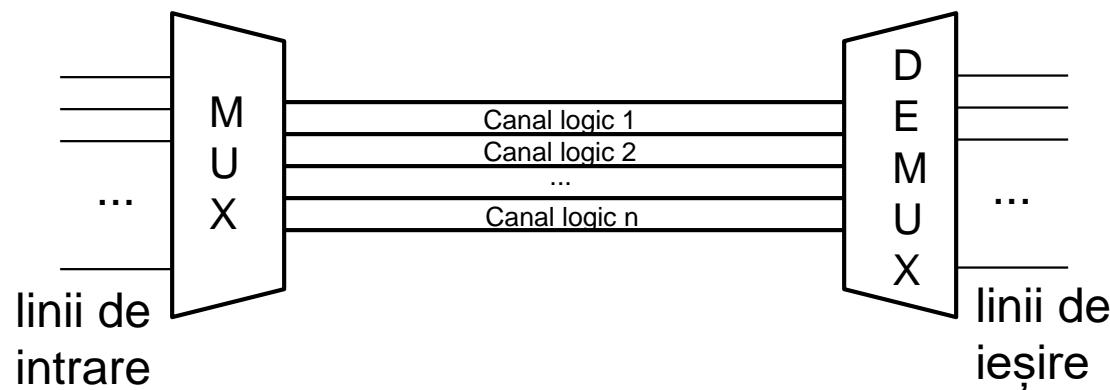


Tipuri de multiplexare



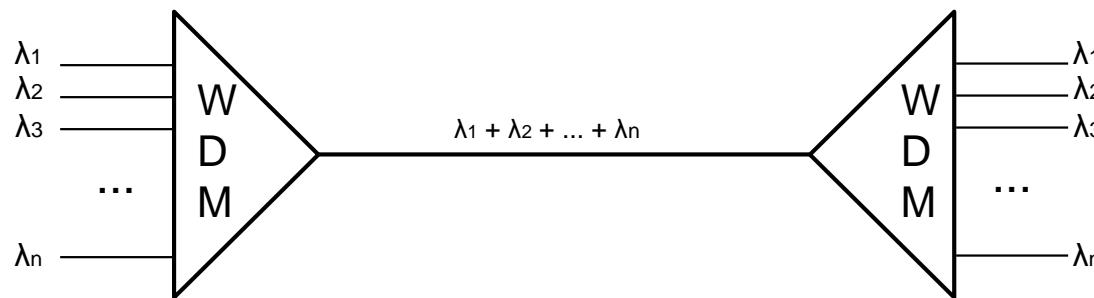
Multiplexarea cu divizare în frecvență (FDM – frequency division multiplexing)

FDM este o tehnică analogică ce poate fi utilizată atunci când lățimea de bandă a unui canal fizic este mai mare decât lățimile de bandă combinate ale semnalelor transmise. În FDM, semnalele generate de fiecare transmițător modulează semnale purtătoare de frecvențe diferite. Semnalele modulate rezultate sunt apoi combinate într-un singur semnal compus ce este transportat de canalul fizic. Frecvențele semnalelor purtătoare sunt separate de o lățime de bandă suficientă astfel încât să se permită modularea lor. Aceste intervale de bandă definesc canalele logice prin care circulă diversele semnale modulate. Canalele logice pot fi separate de porțiuni înguste de bandă neutilizată (denumite benzi de gardă), pentru prevenirea suprapunerii semnalelor. În plus frecvențele semnalelor purtătoare nu pot să interfereze cu frecvențele semnalelor originale (modulatoare).



Multiplexarea cu divizare a lungimii de undă (WDM – wavelength division multiplexing)

Conceptul din spatele WDM este același ca și în cazul FDM, doar că WDM este utilizat pentru multiplexarea semnalelor optice ce sunt transmise pe fibra optică.

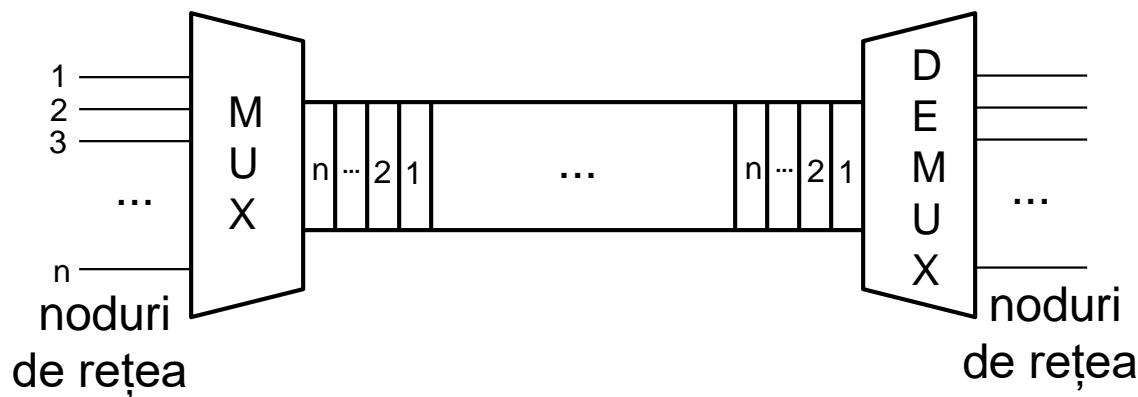


Scopul WDM este de a combina mai multe surse de lumină, rezultând o singură undă de lumină, care la recepție este despărțită înapoi, rezultând elementele componente originale. Combinarea și despărțirea surselor de lumină se realizează cu ajutorul unor prisme, ce curbează unda de lumină în funcție de unghiul de incidentă și frecvența sa. Folosind această tehnică, un multiplexor poate combina mai multe unde de lumină, fiecare având o bandă de frecvențe îngustă, într-una cu o bandă de frecvențe mai mare. Un demultiplexor poate, folosindu-se de aceeași tehnică, inversa procesul.

O metodă mai nouă de WDM, denumită DWDM (dense WDM), poate multiplexa un număr foarte mare de canale logice, lăsând un spațiu îngust între ele.

Multiplexarea cu divizare în timp (TDM – time division multiplexing)

TDM este un proces digital, ce permite mai multor conexiuni de date să partajeze lățimea de bandă largă a unui canal fizic, dar în loc să se împartă câte o porțiune din această lățime de bandă pentru fiecare conexiune, întregă capacitate de bandă poate fi utilizată periodic de o conexiune pentru un interval scurt de timp (slot).



Multiplexarea cu divizare în cod (CDM – code division multiplexing)

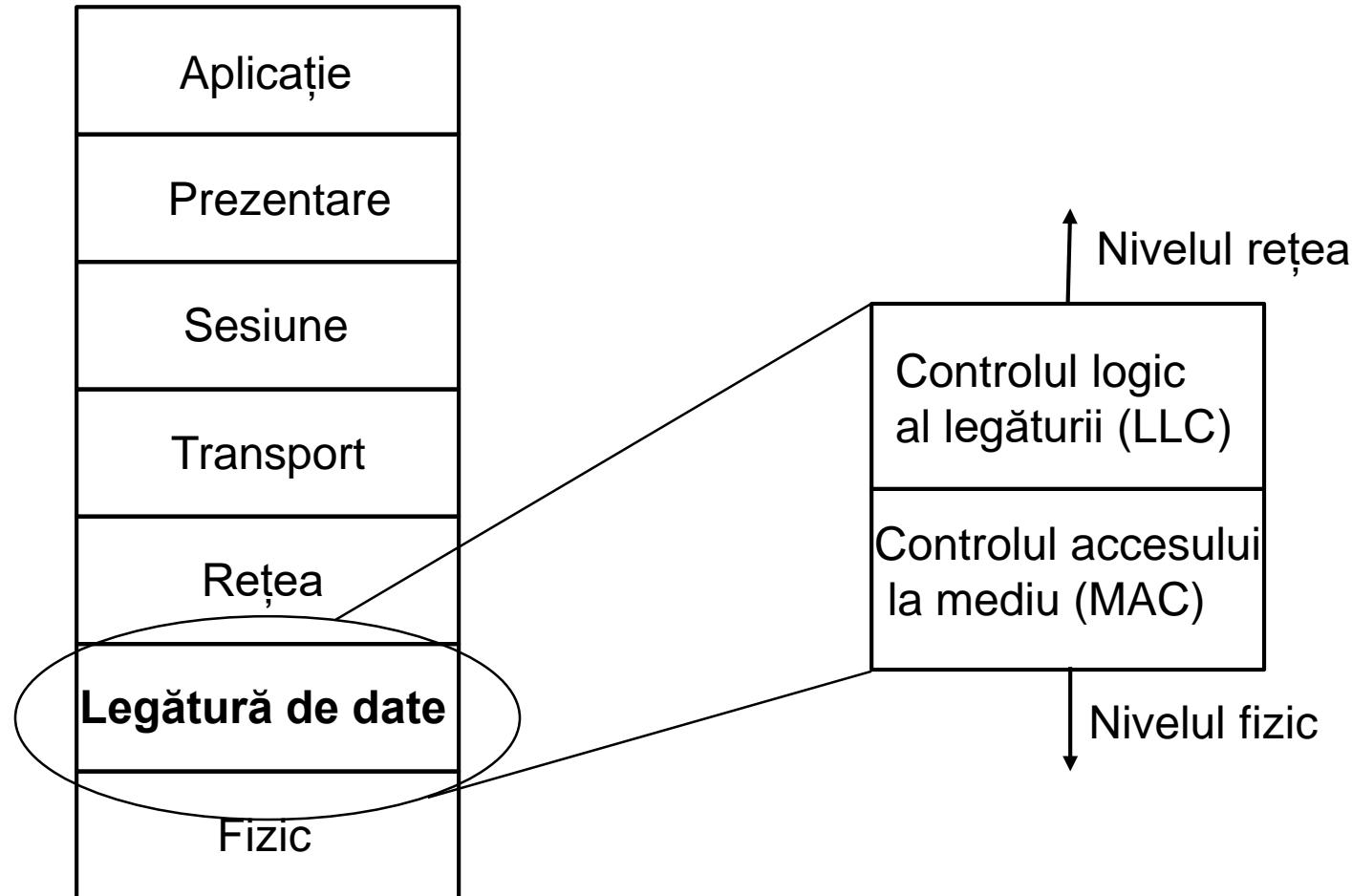
CDM este o metodă de împrăștiere a spectrului (spread spectrum), prin care un semnal cu o lățime de bandă îngustă, este împrăștiat pe o bandă de frecvență mai largă, asigurându-se o toleranță mai mare la interferențe și permitând mai multor semnale să folosească aceeași lățime de bandă disponibilă a canalului fizic.

CDM permite fiecărui nod din rețea să transmită, utilizând întreg spectrul de frecvențe, tot timpul. Mai multe transmisii simultane sunt separate folosind teoria codării.

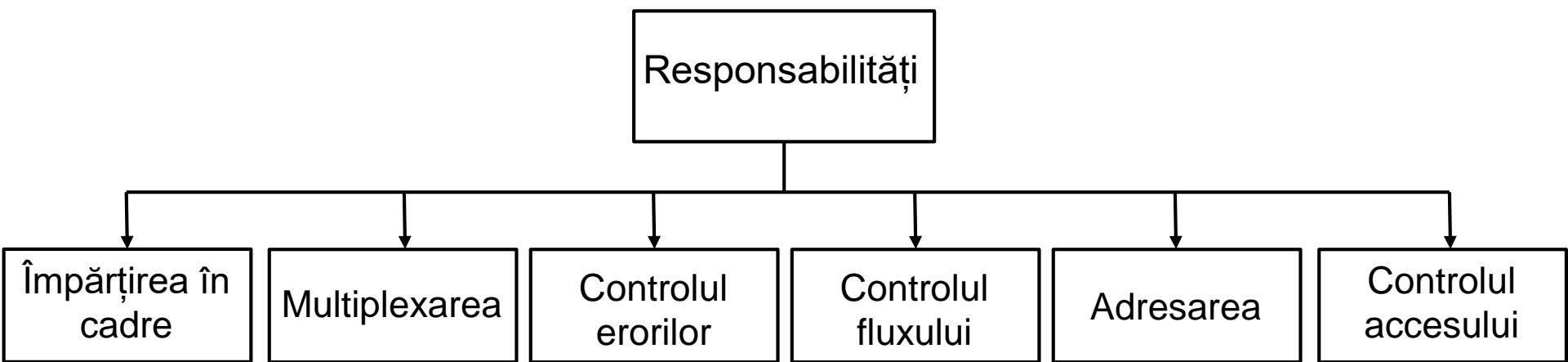
În CDM, fiecare interval de timp necesar transmisiei unui bit, este împărțit în m sloturi (intervale scurte) denumite "chips". De obicei se folosesc 64 sau 128 de chips pe bit. Fiecărui nod de rețea i se asociază un cod unic de m biți, denumit secvență chip. Pentru a transmite un bit de 1, un nod va transmite secvența sa de cod, iar pentru a transmite un bit de 0, va utiliza secvența de cod negată. Nicio altă secvență nu poate fi transmisă de respectivul nod. Toate secvențele chip ale nodurilor sunt ortogonale în pereche.

Nivelul legătură de date

Modelul OSI



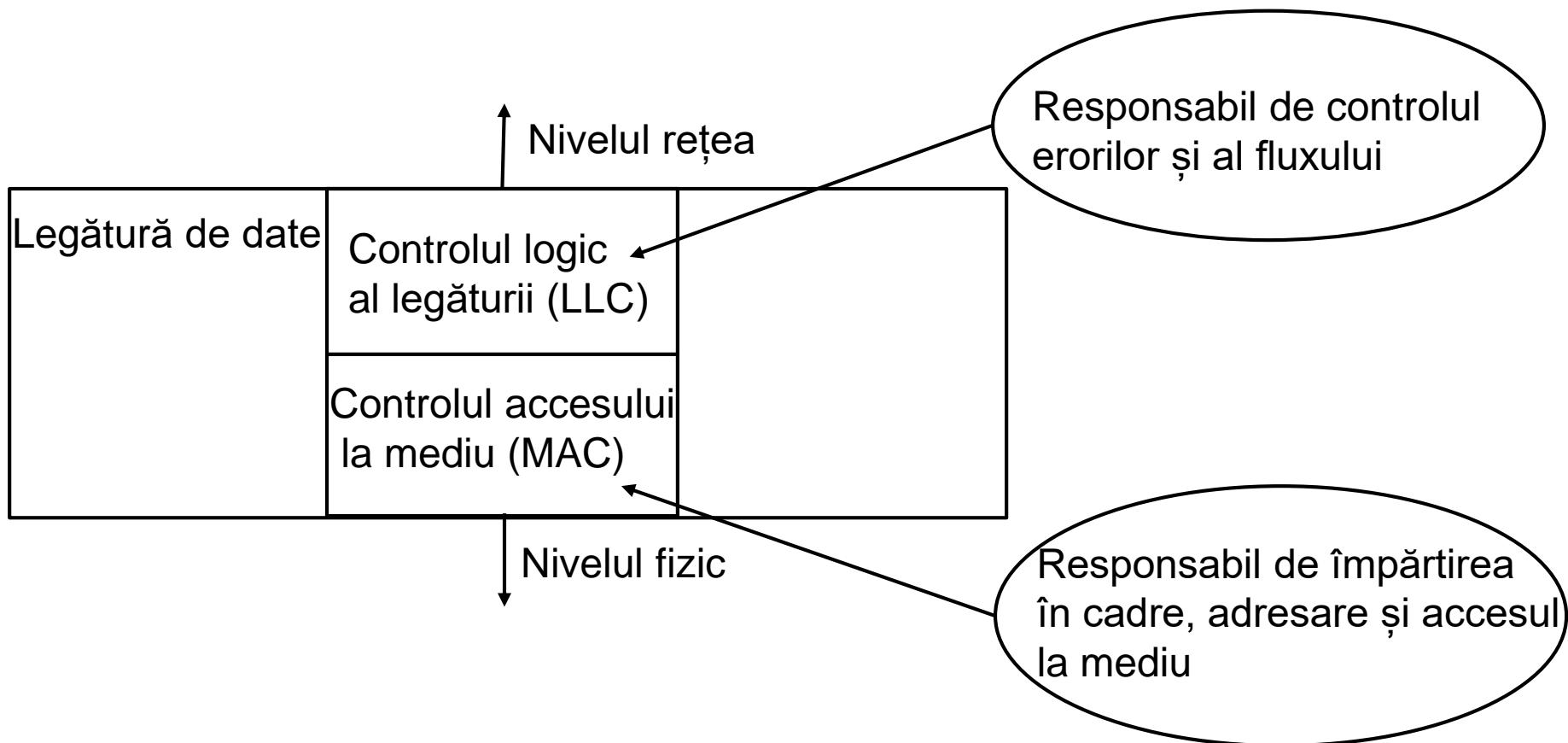
Responsabilitățile nivelului legătură de date



Responsabilitățile nivelului legătură de date

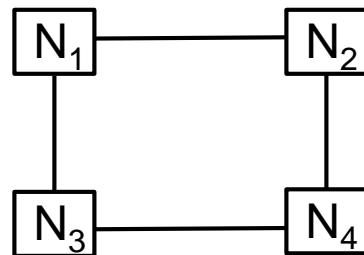
- împărțirea în cadre
 - împarte pachetele la expeditor în cadre adaugând un header și un trailer;
 - decapsulează cadrele la receptor;
 - formatul cadrelor diferă în funcție de protocol.
- adresarea
 - echipamentele de rețea sunt adresate utilizând o adresă fizică (MAC);
 - adresele fizice sunt introduse în header-ul cadrelor folosindu-se pentru identificarea sursei respectiv a destinației unui cadru;
- controlul erorilor
 - se utilizează diverse tehnici de detectie și corecție a erorilor;
- controlul fluxului
 - previne transmisia de cadre la o rată de transfer care să depășească capacitatea de recepție a destinatarului;
- controlul accesului
 - controlul accesului în cazul unui mediu partajat.

Subnivele LLC și MAC

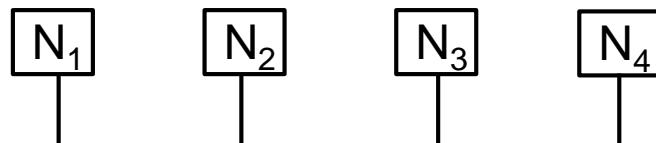


Tipuri de rețele

- Rețele **punct-la-punct** (point-to-point): nodurile sunt conectate în rețea în mod direct două câte două;



- Rețele cu **difuzare** (broadcast): toate nodurile din rețea partajează un singur canal de comunicație;



Controlul accesului multiplu la mediu

Într-o rețea cu difuzare când două sau mai multe noduri transmit date în același timp, utilizând același spectru de frecvență, cadrele transmise vor intra în coliziune.

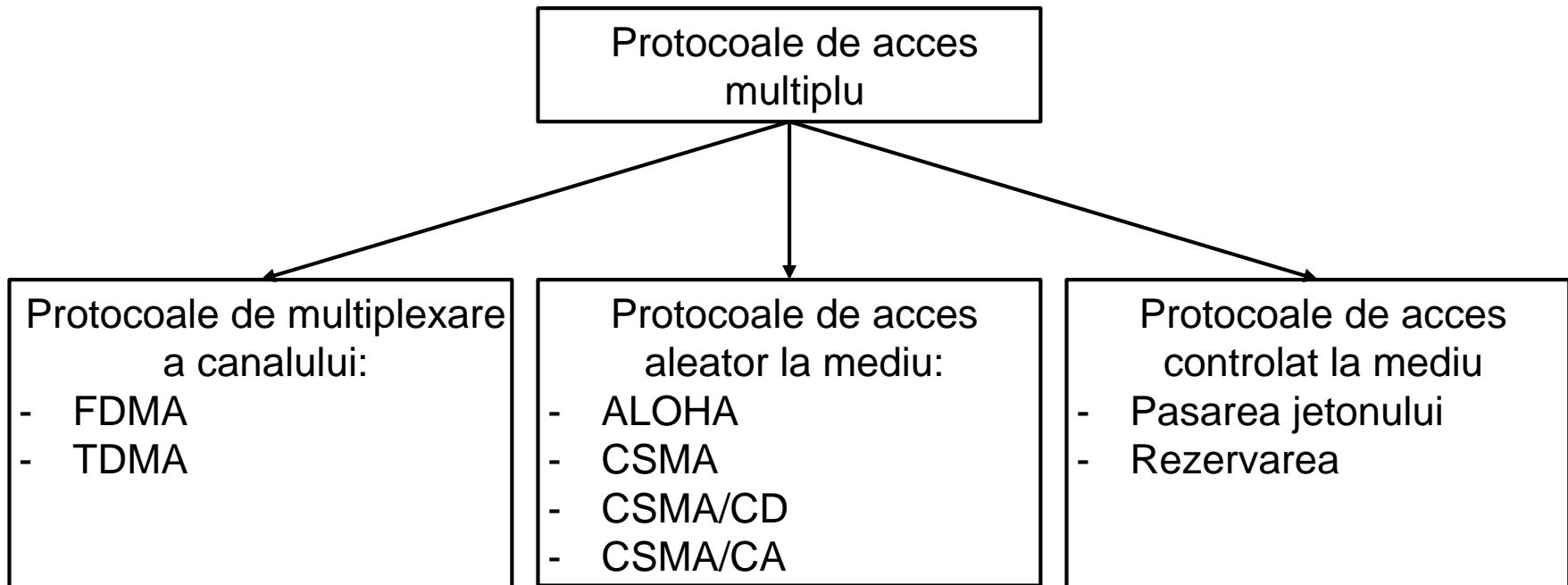
Pentru preîntâmpinarea coliziunilor este nevoie de un protocol care să arbitreze / coordoneze accesul la mediu.

Acstea protocole se numesc protocole de control al accesului la mediu (Multiple Access Control Protocols – MAC Protocols)

Obiectivul principal al unui protocol MAC este de a minimiza coliziunile, crescând rata de transfer a conexiunii (throughput). Astfel, un protocol MAC stabilește:

- când un nod poate transmite date pe mediu;
- cum reacționează nodul în cazul în care dorește să transmită date, dar mediul este ocupat;
- cum reacționează nodul în cazul în care cadrul transmis de acesta este implicat într-o coliziune.

Protocole de acces multiplu



Alocarea canalului

- **Alocarea statică** a canalului presupune împărțirea capacității acestuia la mai mulți utilizatori utilizând o metodă de multiplexare a datelor.

Exemple:

- Frequency Division Multiple Access (FDMA)
 - Lățimea de bandă este divizată în N părți egale, câte una pentru fiecare din cei N utilizatori. Fiecare utilizator va avea la dispoziție o bandă de frecvență pentru transmisie, transmisia sa neinterferând cu cea a altor utilizatori.
- Time Division Multiple Access (TDMA)
 - Întreaga capacitate de bandă este utilizată într-un singur canal care este partajat în timp între cei N utilizatori. Fiecare utilizator își va aștepta rândul pentru a putea transmite pe mediu.
- **Alocarea dinamică** a canalului presupune alocarea capacității de bandă a canalului în funcție de nevoia de transmisie a datelor.

Protocolul ALOHA

A fost proiectat la începutul anilor 70 de catre Norman Abramson pentru a deservi o rețea wireless, însă poate fi utilizat pentru orice tip de mediu de transmisie partajat.

Există două variante ale protocolului

- Pure ALOHA în timp continuu
- Slotted ALOHA în timp discret

Pure ALOHA

- Toate nodurile transmit cadre de date egale ca și dimensiune => timpul de transmisie este același;
- Un nod poate transmite date în orice moment;
- După transmisia unui cadru, expeditorul asteaptă o confirmare de primire din partea destinatarului;
- Dacă nu se primește confirmarea, expeditorul retransmite cadrul după un interval de timp aleator;
- Dacă după mai multe încercări de transmisie a unui cadru nu se primește nici o confirmare, expeditorul renunță la transmisia cadrului.

Protocolul ALOHA

- Eficiența canalului (throughput) reprezintă procentul de cadre transmise care ajung cu succes la destinație sau procentul din lățimea benzii de transmisie a canalului utilizată pentru transmisia cadrelor care nu suferă coliziuni.
- Eficiența canalului în cazul protocolului Pure ALOHA este de 18%.

Slotted ALOHA

- Timpul este divizat în eșanțioane egale cu durata de transmisie a unui cadru;
- O stație poate transmite doar la începutul unui eșantion de timp;
- Dacă o stație pierde începutul unui eșantion de timp va aștepta următorul eșantion pentru a transmite;
- Un ceas central va informa stațiile asupra începutului fiecărui eșantion de timp;
- Eficiența canalului în cazul protocolului Slotted ALOHA este de 37%.

Protocolul ALOHA

Avantaje:

- Dacă mediul este liber, un nod poate transmite cadre în mod continuu utilizând întreaga capacitate de bandă disponibilă;
- Simplu de implementat;
- Nu este nevoie de un nod central care să gestioneze accesul la mediul.

Dezavantaje:

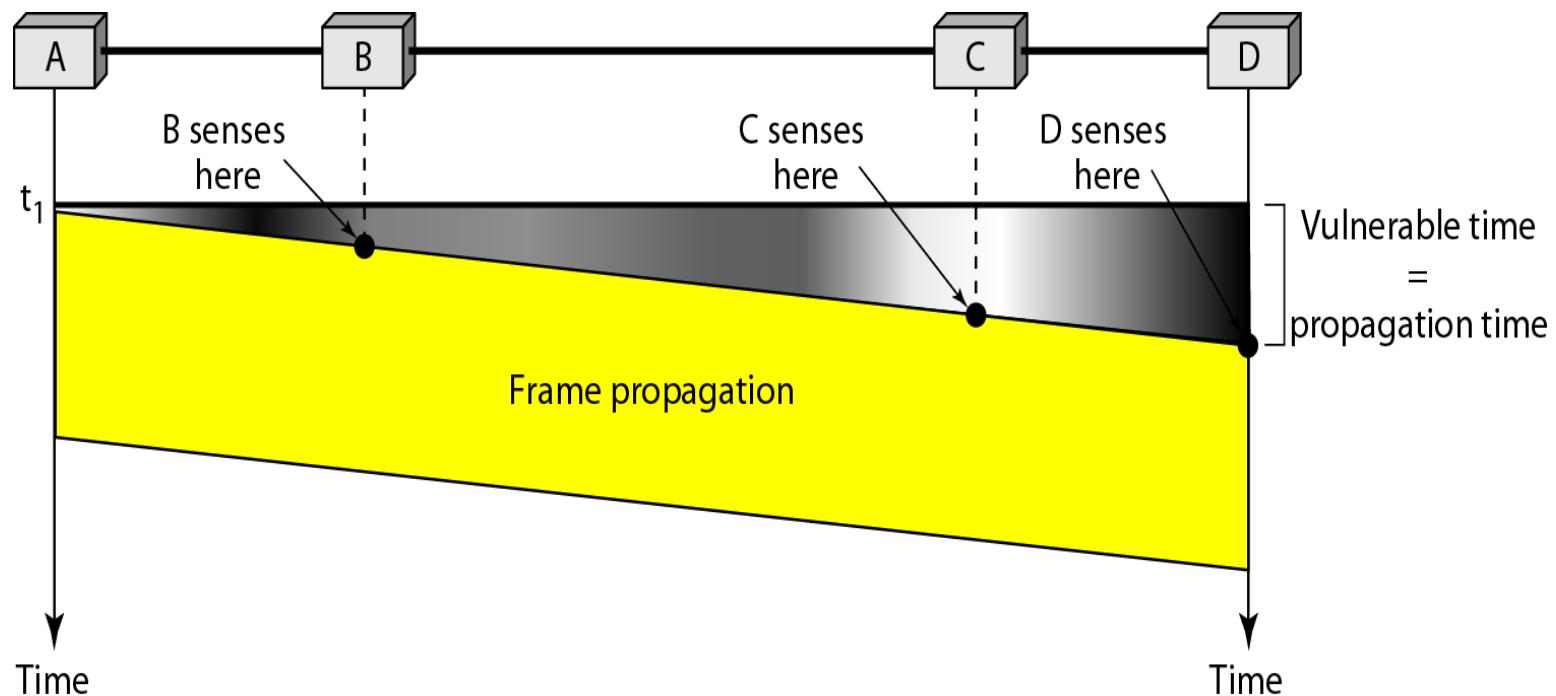
- Dacă mai multe noduri vor să transmită date în același timp vor apărea coliziuni multiple ce vor duce la o eficiență scazută a canalului.

Protocolul CSMA (Carrier Sense Multiple Access)

- Este utilizat în rețele LAN cu timp de propagare redus;
- Pentru îmbunătățirea performanțelor protocolul nu permite transmisii de cadre care conduc în mod cert la coliziuni;
- Dacă o stație are de transmis date verifică disponibilitatea mediului de transmisie înainte de a transmite;
- Pot apărea coliziuni în momentul în care mai multe stații încep transmisia într-un interval de timp scurt.
- În funcție de modul în care se comportă un nod atunci când mediul de transmisie este disponibil, respectiv când este ocupat identificăm următoarele versiuni ale protocolului CSMA:
 - Non-Persistent CSMA
 - 1-Persistent CSMA
 - p-Persistent CSMA

Protocolul CSMA (Carrier Sense Multiple Access)

Intervalul de timp în care pot apărea coliziuni este egal cu timpul de propagare
Cu cât timpul de propagare este mai mare cu atât performanțele rețelei scad



Nonpersistent CSMA

Verifică mediul de transmisie:

1. Dacă mediu de transmisie este liber, transmite, dacă nu sari la pasul 2;
2. Dacă mediul de transmisie este ocupat, așteaptă un interval de timp aleator și sari la pasul 1.

1-persistent CSMA

Verifică mediul de transmisie:

1. Dacă mediu de transmisie este liber, transmite;
2. Dacă mediul de transmisie este ocupat, verifică continuu până acesta devine liber apoi transmite datele imediat cu probabilitate 1.

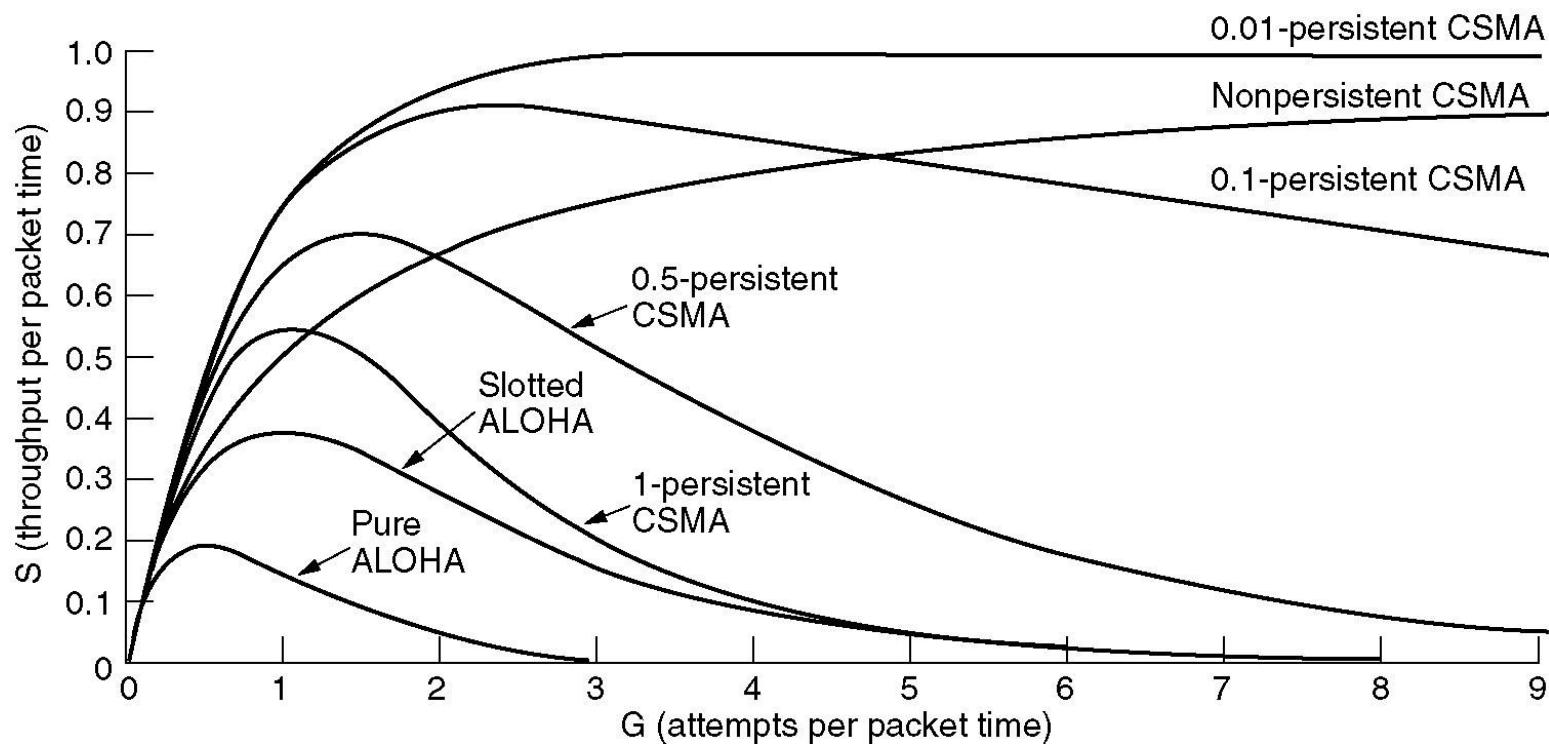
p-persistent CSMA

Timp discret – eșantionul de timp egal cu timpul maxim de propagare

Verifică mediul de transmisie:

1. Dacă mediul de transmisie este liber
 - transmite cu probabilitate p sau
 - așteaptă un eșantion de timp cu probabilitatea $(1-p)$ apoi repetă 1
2. Dacă mediul de transmisie este ocupat continuă verificarea până se eliberează apoi sari la pasul 1.

Performanțele teoretice ale protocolelor de acces la mediu

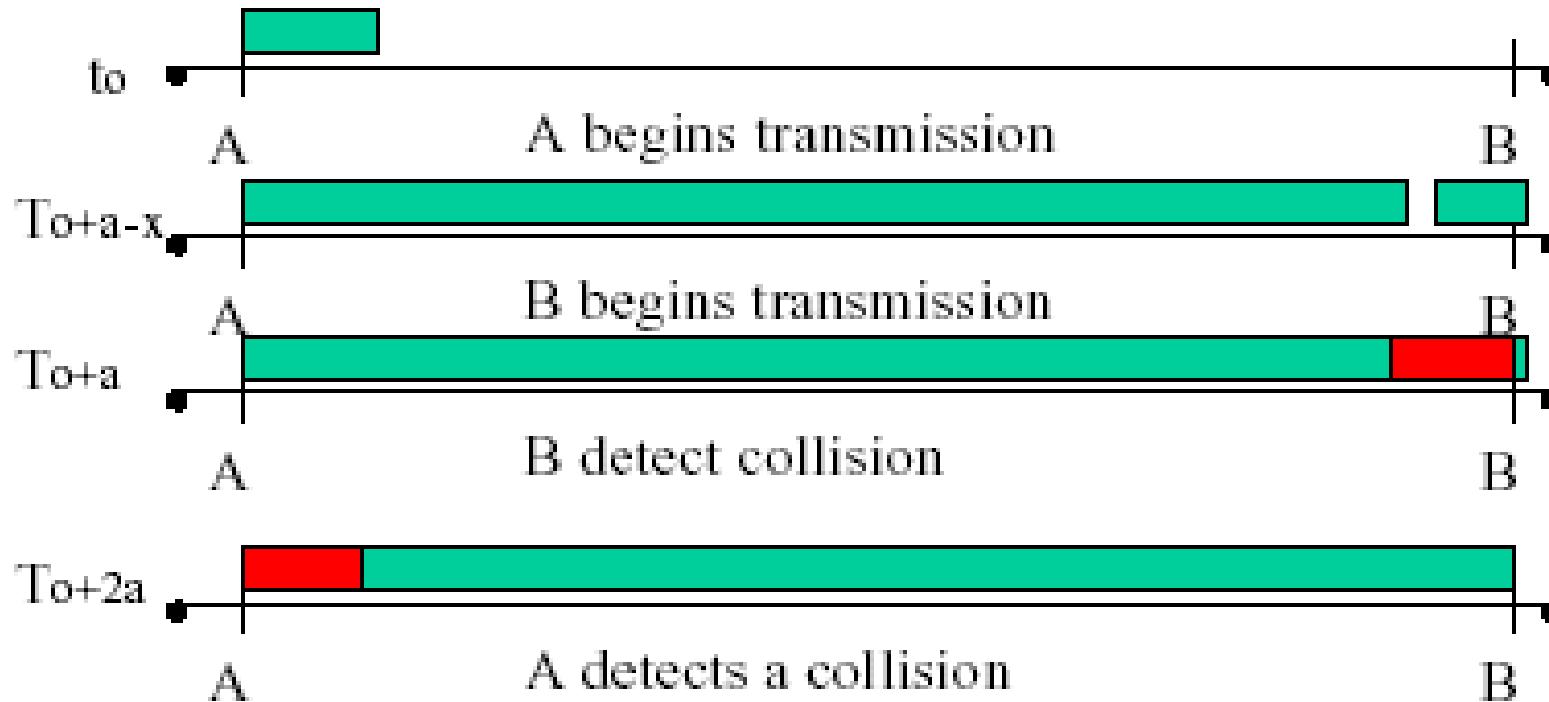


S – eficiența canalului; G – încărcarea canalului

Protocolul CSMA / CD

- Utilizează unul din protocolele CSMA pentru transmisie
- Dacă un nod detectează o coliziune în timp ce transmite:
 - Întrerupe transmisia și
 - Transmite un semnal de bruiaj (48 biți) pentru a notifica toate nodurile de faptul că s-a produs o coliziune; toate nodurile vor distruge cadrul transmis;
 - După transmisia mesajului de bruiaj, așteaptă un interval de timp aleator și
 - Retransmite cadrul.

Protocolul CSMA / CD – timpul necesar detecției unei coliziuni



a – timpul maxim de propagare pe mediu;

Restricție CSMA/CD: timpul de transmisie a unui cadru trebuie să fie mai mare sau egal dublului timpului maxim de propagare.

Protocolul CSMA / CD - algoritmul exponențial de așteptare

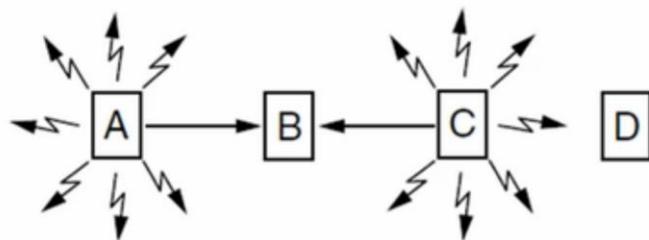
Utilizat de standardul Ethernet pentru a determina perioada optimă de așteptare aleatoare după o coliziune.

- Stabilește eșantionul de timp egal cu 2^* timpul maxim de propagare + timpul de transmisie al secvenței de bruiaj;
- După K coliziuni, selectează un număr aleator R cuprins între 0 și 2^k-1 , așteaptă o perioadă de timp egală cu R eșantioane de timp și apoi transmite cadrul când mediul este liber;
- Limitează intervalul în care R poate lua valori pentru maxim K=10 ($\{0 - 1023\}$);
- Renunță la transmisia cadrului după 16 încercări nereușite.

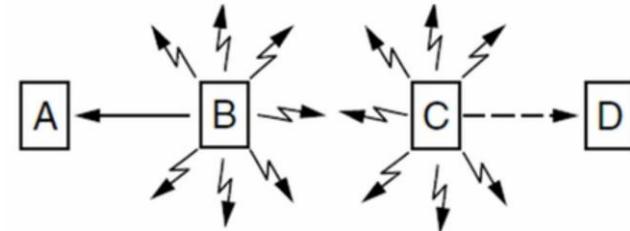
Acces multiplu – transmisii fără fir

Probleme:

Nodurile pot avea suprafete de acoperire diferite.



Problema statiei ascunse

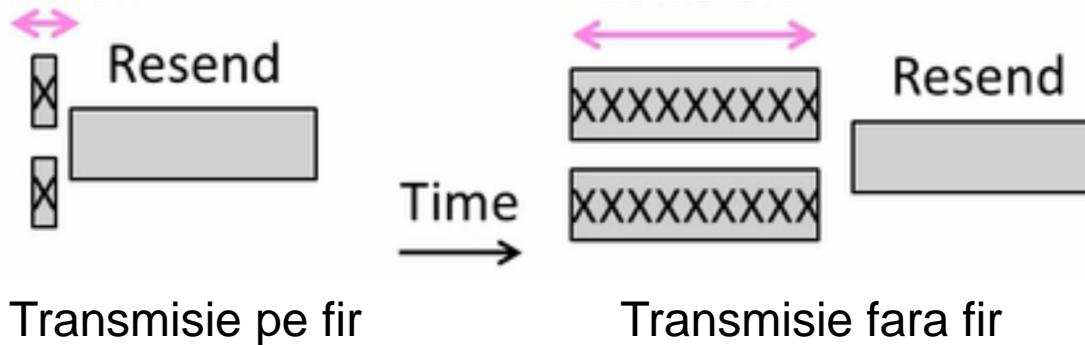


Problema statiei expuse

Acces multiplu – transmisii fara fir

Probleme:

Nodurile nu pot asculta mediul in timp ce transmit.



Acces multiplu – transmisii fara fir

Protocolul MACA

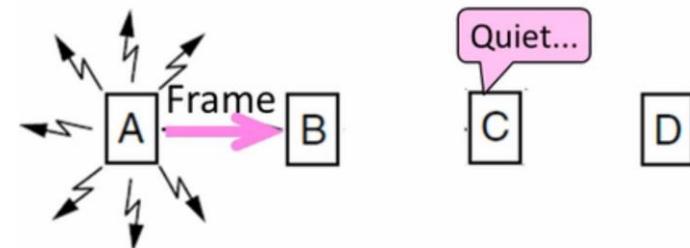
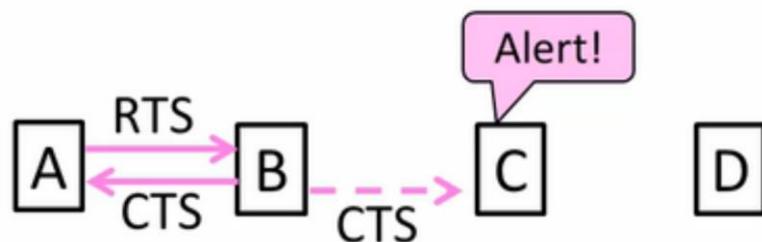
Algoritm:

1. Sursa unui cadru de date trimite initial un cadru scurt: RTS (Request-To-Send) ce contine lungimea cadrului de date ce urmeaza a fi transmis.
2. Destinatia raspunde printr-un CTS (Clear-To-Send) ce contine in copie lungimea cadrului de date.
3. Sursa transmite cadrul de date, in timp ce restul statilor ce au primit mesajul CTS se abtin sa emita pe perioada necesara transmisiei.

Pot aparea coliziuni de mesaje RTS/CTS, insa cadrele sunt de dimensiune redusa, timpul consumat pentru transmisia lor fiind mic.

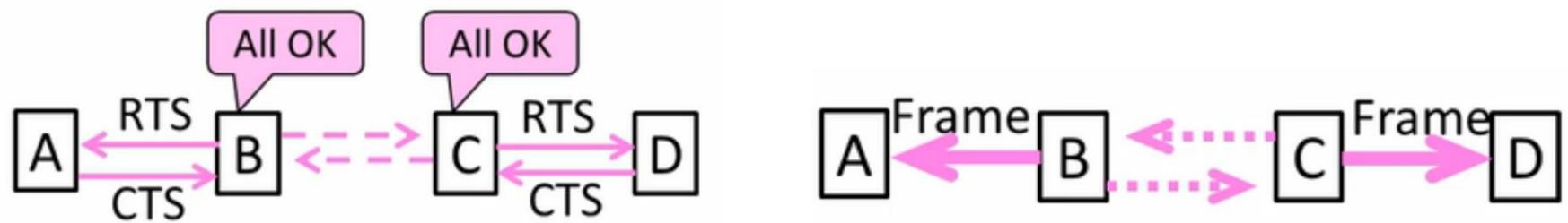
Acces multiplu – transmisii fără fir

Protocolul MACA – cazul stației ascunse



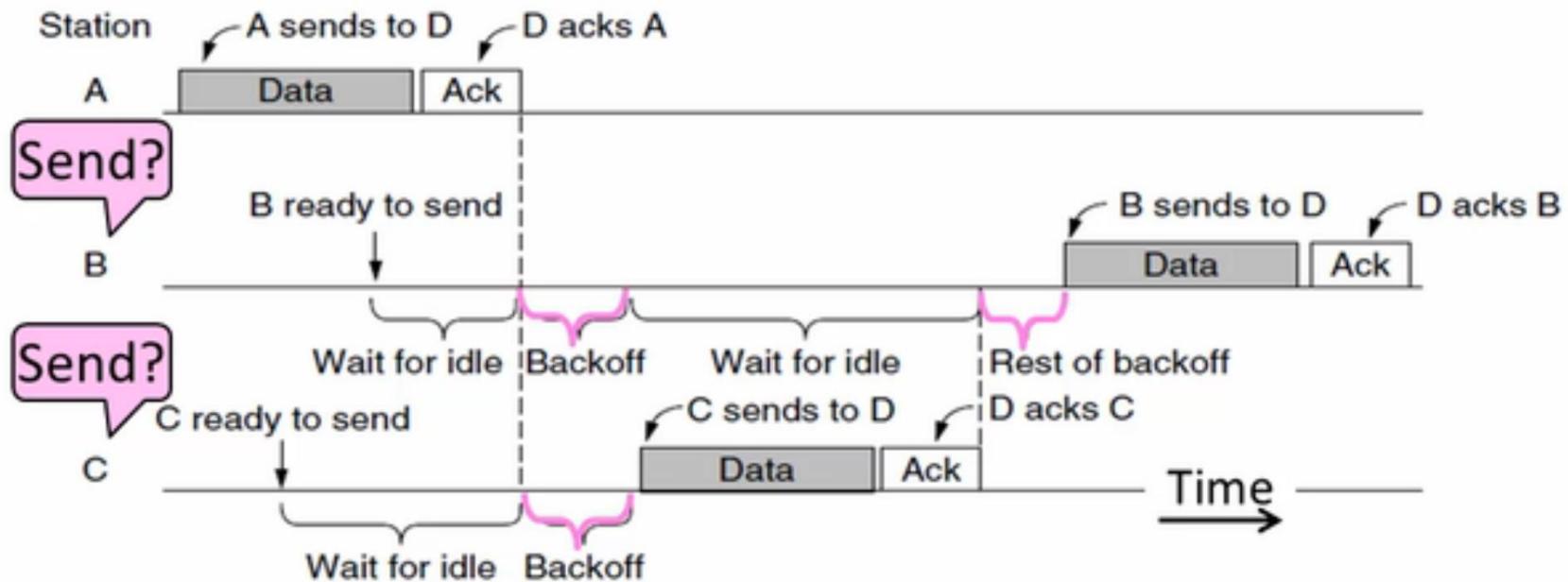
Acces multiplu – transmisii fără fir

Protocolul MACA – cazul statiei expuse



Acces multiplu – transmisii fără fir

Protocolul CSMA/CA



Detectia și corectia erorilor

Rețelele de calculatoare trebuie să asigure un anumit nivel de acuratețe pentru a putea fi utilizate pentru transferul informației. În cazul multor aplicații, integritatea datelor trebuie garantată. Însă, datorită diverselor tipuri de perturbații, mesajele transmise pot fi interpretate eronat la destinație, devenind corupte. În consecință, diversele protocoale utilizate în transmisia datelor necesită anumite mecanisme de detectie și corecție a erorilor.

Tipuri de erori

Eroare singulară – presupune alterarea unui singur bit, din 1 în 0 sau invers, din unitatea de date (byte, pachet, etc).

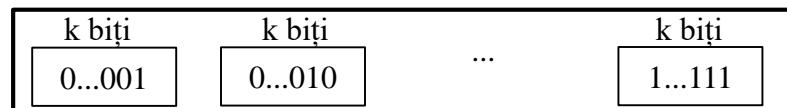
Eroare în rafală – descrie modificarea nedorită, în tipul transmisiei, a mai multor biți din unitatea de date. Biții eronați nu trebuie să fie neapărat consecutivi. Dimensiunea rafalei se măsoară de la primul la ultimul bit modificat.

Pentru a detecta sau corecta o eroare, mecanismele specialize trebuie să transmită un număr suplimentar de biți, pe lângă informația utilă. Acești biți redundanti sunt adăugați de către sursa mesajului și sunt eliminați la destinație. Redundanța este asigurată prin diverse scheme de codare, ce crează o relație între biții de date și cei utilizati în operațiunile de detectie, respectiv corecție. Rația dintre biții de date și cei redundanti, precum și robustețea procesului sunt două elemente importante în orice schemă de codare. Schemele de codare se împart în două categorii: coduri de bloc și coduri convoluționale. La rândul lor codurile de bloc se împart în două categorii: liniare și neliniare. În cazul liniar, cei r biți redundanti sunt calculați pe baza celor k biți de date, cu ajutorul unei funcții liniare. În continuare, vom face referire doar la codurile de bloc liniare.

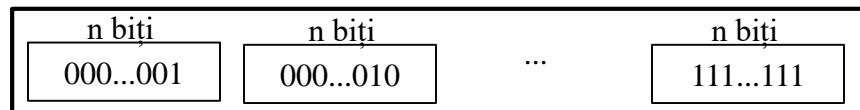
Codarea blocurilor

Codurile de bloc împart mesajele în blocuri, de câte k biți fiecare, denumite **cuvinte de date (datawords)**. Fiecarui bloc îi sunt adăugați r biți redundanti, rezultând un nou bloc, definit **cuvânt de cod (codeword)**, cu dimensiunea $n = k + r$. Numărul de biți adăugați și modul de alegere a acestor biți depind de schema de codare. În urma operației de codare, putem obține un număr maxim de 2^k cuvinte de date și 2^n cuvinte de cod distincte. Cum $n > k$, mulțimea cuvintelor de cod este mai mare decât mulțimea

cuvintelor de date. Procesul de codare este de tip unu-la-unu, un cuvânt de date având întotdeauna același cuvânt de cod asociat. În consecință, vom avea $2^n - 2^k$ cuvinte de cod neutilizate, considerate invalide.



2^k cuvinte de date



2^n cuvinte de cod

Un prim exemplu de codare de bloc, 4B/5B, a fost studiat în capitolul anterior. În acest caz avem $2^4 = 16$ cuvinte de date și $2^5 = 32$ cuvinte de cod.

Procesul de detectie a erorilor

Codurile de bloc permit detectia aparitiei unei erori dacă:

1. Receptorul detine (sau poate determina) o lista ce contine cuvintele de cod valide.
2. Cuvantul de cod original a fost modificat intr-unul invalid.

Fără a specifica modul în care asociem cuvintele de cod la cuvintele de date, considerăm următorul exemplu ($k=2$, $n=3$):

Cuvânt de date	Cuvânt de cod
00	000
01	011
10	101
11	110

Presupunem că sursa dorește să transmită biții 11, astfel că ea va trimite cuvântul de cod 110. La destinație se pot distinge următoarele cazuri:

1. Receptorul primește mesajul 110. Acesta reprezintă un cuvânt de cod valid, astfel că el este acceptat și se extrage cuvântul de date 11 pentru utilizare.
2. Receptorul primește mesajul 100, apărut în urma unei erori singulare. Cuvântul de cod nefiind valid este distrus.
3. Receptorul primește mesajul 000, apărut în urma unei erori în rafală. Cuvântul de cod, fiind unul valid, este acceptat astfel că se extrage cuvantul de date eronat 00 pentru a fi utilizat.

Observăm că schema de codare din exemplu permite identificarea unei erori singulare, însă nu și a unei erori în rafală. În general codurile de detecție și corecție sunt dezvoltate pentru un anumit tip de erori, neputând asigura robustețea necesară în celelalte cazuri.

Procesul de corecție a erorilor

Corecția erorilor este mai dificilă decât detecția. În cazul detecției, receptorul trebuie doar să știe că cuvântul de cod recepționat este invalid. În cazul corecției, el trebuie să identifice (sau să ghicească) cuvântul de cod original. Pentru ca identificarea să se poată realiza cu succes, sunt necesari mai mulți biți redundanți.

Considerând aceleași cuvinte de date ca și în exemplu anterior, un cod ce asigură corecția în cazul unei erori singulare ar putea genera cuvintele de cod prezentate în tabelul următor:

Cuvânt de date	Cuvânt de cod
00	00000
01	10101
10	11010
11	01111

Transmitând cuvântul de cod 00000 asociat cuvântului de date 00, presupunem că acesta este corupt de procesul de transmisie, în urma unei erori singulare, rezultând noul cuvânt de cod 00100. Receptorul, presupunând aparitia unei erori singulare folosește următoarea strategie pentru a identifica cuvântul de date corect:

1. Comparând cuvântul de cod recepționat cu primul cuvânt de cod din tabel, observă că cele două diferă printr-un singur bit.
2. Comparând cuvântul de cod recepționat cu oricare alt cuvânt de cod din tabel, observă că acestea diferă prin mai mulți biți.
3. Receptorul selectează primul cuvânt de cod ca fiind, cel mai probabil, cuvântul transmis și extrage cuvantul de date asociat 00 pentru utilizare.

Distanța Hamming

Reprezintă un concept important pentru domeniul detecției și corecției erorilor. Distanța Hamming $d(x,y)$ dintre două cuvinte de cod, x și y , având aceeași dimensiune, este definită ca fiind numărul de diferențe dintre biții similari (ce au aceeași poziție în fiecare cuvânt). Această distanță se poate identifica ușor cu ajutorul operației binare \oplus . Astfel:

$$d(\vec{x}, \vec{y}) = \sum_{i=1}^n (x_i \oplus y_i)$$

Distanța Hamming minimă

Parametrul luat în calcul, în momentul dezvoltării unei scheme de cod, este distanța Hamming minimă, d_{\min} . Aceasta este definită ca și valoarea minimă a distanțelor asociate tuturor perechilor de cuvinte de cod posibile dintr-un set.

În cazul codurilor de bloc liniare, distanța Hamming minimă reprezintă numărul de biți de unu din cuvântul de cod diferit de 0, ce conține numărul minim de biți de 1.

Orice schemă de codare C, definește cel puțin 3 parametrii: n , k și d_{\min} .

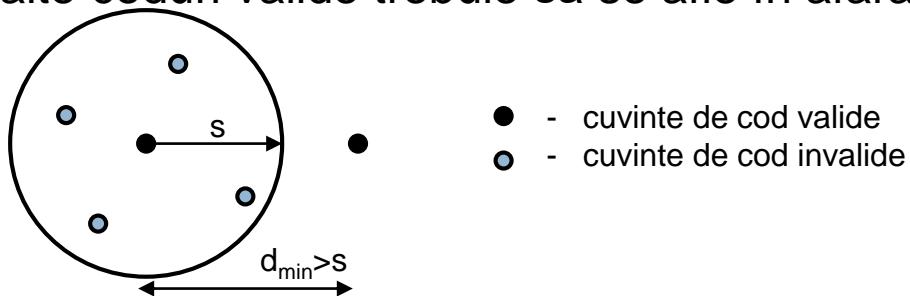
Distanța minimă și eroarea

Când un cuvânt de cod este corupt în timpul transmisiei, distanța Hamming dintre cuvântul transmis și cel recepționat reprezintă numărul de biți afectați de eroare.

Distanța minimă necesară pentru detectia erorilor

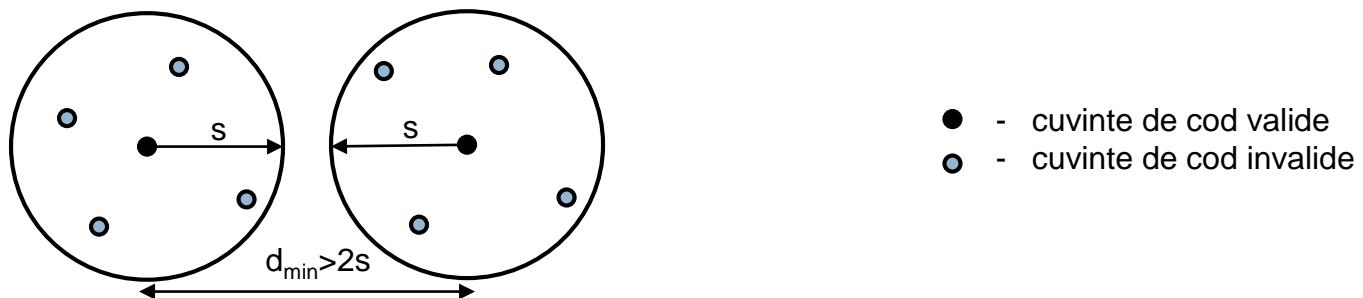
Se pune problema identificării distanței Hamming minime a unei scheme de cod necesare pentru garantarea detectiei unor erori ce afectează s biți. Dacă s biți sunt afectați de erorile de transmisie, distanța Hamming dintre cuvântul de cod transmis și cel recepționat este s. Pentru a putea detecta cu succes aceste erori ce modifică s biți, distanța minimă necesară este s+1 (pentru a împiedica o transformare, în urma erorilor, a cuvântului de cod transmis, într-un alt cuvânt de cod valid recepționat). Notă: chiar dacă o schemă de codare având $d_{min} = s + 1$, s-ar putea să detecteze apariția erorilor, în cazul în care sunt afectați mai mult de s biți, o astfel de schemă garantează identificarea acestora, doar atunci când sunt afectați maxim s biți.

În figura de mai jos, cercul de rază s, având în centru un cuvântul de cod oarecare transmis, definește locul geometric ce conține toate posibilele coduri invalide recepționate în urma unor erori, ce modifică maxim s biți. Pentru detectia erorilor, toate celelalte coduri valide trebuie să se afle în afara cercului.



Distanță minimă necesară pentru corecția erorilor

Corecția erorilor presupune luarea unei decizii mai complicate: stabilirea unui cuvânt de cod valid ce să înlocuiască cuvântul invalid recepționat. Pentru a se putea lua această decezie, a fost introdus conceptul de teritoriu al cuvântului de cod valid, ce definește un loc geometric ce aparține exclusiv cuvântului de cod. Orice cuvânt de cod invalid, ce se găsește în interiorul locului geometric respectiv, este înlocuit cu cuvântul de cod de care aparține locul geometric.



Dacă analizăm figura de mai sus, ajungem la concluzia că avem nevoie de o distanță minimă $d_{min} = 2s+1$ între cuvintele de cod din setul definit de schema de codare pentru a putea garanta corecția erorilor.

Coduri corectoare de erori

Identificarea numărului minim de biți redundanți ce garantează corectarea erorilor singulare

Presupunem că dorim să proiectăm o schemă de cod ce utilizează cuvinte de date de k biți, r biți redundanți și ce permite corecția oricărei erori singulare apărute la transmisie. Codul, $C(n,m)$, având $d_{\min} = 3$, definește în total 2^n cuvinte de cod ($n=k+r$), din care doar 2^k vor fi valide (cuvintele de cod asociate unu-la-unu celor 2^k cuvinte de date), celelalte fiind invalide. Fiecare din cele 2^k cuvinte de cod valide vor avea în teritoriul lor n cuvinte de cod invalide, obținute sistematic prin modificarea a câte un bit din cei n biți ai cuvântului de cod valid. În consecință, fiecare din cele 2^k cuvinte de cod valide va avea în teritoriul său $n+1$ cuvinte diferite (unul valid și n invalide), ceea ce înseamnă că pentru a putea corecta orice eroare singulară, condiția $(n+1) \cdot 2^k \leq 2^n$ trebuie indeplinită. Cum $n = k+r$, rezultă condiția $(k+r+1) \leq 2^r$, ce identifică pentru un k dat, numărul minim de biți redundanți r necesari.

Codul Hamming

Codurile Hamming sunt proiectate pentru identificarea unei erori singulare, având $d_{\min} = 3$. Într-un cod Hamming, bițiii unui cuvânt de cod sunt numerotați consecutiv de la stânga la dreapta, începând cu valoarea 1. Pozițiile a căror valoare reprezintă o putere a lui 2, conțin biți redundanti, restul fiind rezervate bițiilor cuvântului de date. Fiecare bit redundant, forțează suma modulo 2, denumită paritate, a unei colecții de biți ce il include și pe el, să fie pară (sau impară). Un bit de date poate fi utilizat de mai multe ori, pentru calculul bițiilor redundanti. Pentru a vedea la calculul căror biți redundanti contribuie un bit de date, poziția sa se scrie ca și sumă de puteri de doi. Astfel, acel bit va contribui la calculul bițiilor redundanti de pe pozițiile definite de sumă.

Când cuvântul de cod, construit după legile definite mai sus, ajunge la destinație, receptorul recalculează bițiii de paritate, utilizând inclusiv valorile bițiilor redundanti primiți. Dacă mesajul nu este eronat, pentru paritate pară, toți bițiii recalculați ar trebui să fie 0. În caz contrar, este clar că a apărut o eroare. Setul de biți de paritate definește sindromul erorii, a cărui valoare zecimală indică poziția bitului eronat.

Vom considera ca și exemplu transmisia mesajului 1010, utilizând pentru corecția erorilor codul Hamming.

Cuvântul de date fiind compus din 4 biți ($k=4$), avem nevoie de un număr de biți redundanți care să satisfacă inegalitatea $(4+r+1) \leq 2^r$, adică $r = 3$. Codul Hamming utilizat va fi $C(7,3)$ cu $d_{\min} = 3$. Vom considera paritate pară.

Cuvântul de cod, format din 7 biți, va avea biții redundanți pe pozițiile 1,2,4, restul fiind ocupate de biții de date.

r_1	r_2	1	r_3	0	1	0
1	2	3	4	5	6	7

$$\text{Poziția } 3 = 1 + 2$$

$$\text{Poziția } 5 = 1 + 4$$

$$\text{Poziția } 6 = 2 + 4$$

$$\text{Poziția } 7 = 1 + 2 + 4$$

Bitul r_1 setează paritatea pentru biții de pe pozițiile 1,3,5,7 $\Rightarrow r_1 \oplus 1 \oplus 0 \oplus 0 = 0 \Rightarrow r_1 = 1$

Bitul r_2 setează paritatea pentru biții de pe pozițiile 2,3,6,7 $\Rightarrow r_2 \oplus 1 \oplus 1 \oplus 0 = 0 \Rightarrow r_2 = 0$

Bitul r_3 setează paritatea pentru biții de pe pozițiile 4,5,6,7 $\Rightarrow r_3 \oplus 1 \oplus 0 \oplus 0 = 0 \Rightarrow r_3 = 1$

Rezultă cuvântul de cod transmis: 1011010

La recepție considerăm două scenarii:

- a) cuvântul de cod recepționat este 1011010

Calculăm biții de paritate:

$$p_1 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$p_2 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$$

$$p_3 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

Deoarece valoarea zecimală a sindromului erorii ($p_3 \ p_2 \ p_1$) este 0, cuvântul de cod este corect.

- b) cuvântul de cod recepționat este 1001010

Calculăm biții de paritate:

$$p_1 = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$

$$p_2 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$p_3 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

Valoarea zecimală a sindromului erorii ($p_3 \ p_2 \ p_1$) este 3, rezultă că bitul de pe poziția 3 este eronat, mesajul corect fiind 1011010.

După cum am precizat, codurile Hamming corectează doar erori singulare. Totuși, printr-un artificiu, se pot corecta și erori în rafală. O secvență de x cuvinte de cod consecutive este aranjată sub forma unei matrice, având câte un cuvânt de cod pe fiecare linie. În mod normal, datele ar trebui transmise linie cu linie, de sus în jos. Pentru a corecta erorile în rafală, datele vor fi transmise pe coloane, de la stânga la dreapta. Când au fost transmiși cei x biți ai primei coloane, se transmite coloana a doua și asa mai departe. La receptor matricea este reconstruită. Dacă a apărut o eroare în rafală ce a afectat maxim x biți, va fi afectat maxim câte un bit de pe fiecare linie, codul Hamming putând recupera cuvintele de cod corecte. Această metodă utilizează $x \cdot r$ biți redundantă pentru a asigura imunitatea unor blocuri de $x \cdot m$ biți la erori în rafală ce modifică maxim x biți.

Coduri detectoare de erori

Bitul de paritate

Codurile corectoare de erori se folosesc pe scară largă atunci când canalul de transmisie este predispus erorilor (ex.: medii de transmisie fără fir). Acolo unde rata de apariție a erorilor este mult mai mică se justifică mai degrabă utilizarea unui cod detector de erori, împreună cu o procedură ce asigură retransmisia mesajului eronat. Spre exemplu, dacă considerăm un canal de transmisie în care erorile sunt izolate, având o rată de apariție de 10^{-6} per bit și un bloc de date de 1000 biți, pentru a asigura corecția erorilor sunt necesari 10 biți redundanți. Un Mbit de date, va necesita 10000 de biți redundanți.

Pentru a detecta apariția unei erori singulare într-un bloc de 1000 de biți este suficientă adăugarea ca informație redundantă a unui bit de paritate. Cum rata de apariție a erorii este de 10^{-6} , la fiecare 1000 de blocuri este foarte probabilă apariția unei erori, caz în care este necesară retransmisia blocului de 1001 biți. Astfel, încărcarea suplimentară în cazul unui Mbit de date, este de 2001 biți, adică mult mai puțini decât în cazul codului Hamming.

Dacă unui bloc de date i se adaugă un singur bit de paritate și blocul este puternic alterat de o eroare în rafală, probabilitatea ca eroarea să fie detectată este de 50%. O varintă îmbunătățită consideră fiecare bloc transmis ca o matrice dreptunghiulară, de l biți lățime și i biți înălțime și calculează pentru fiecare coloană un bit de paritate care este adăugat într-o nouă linie la sfârșitul matricei. Matricea este apoi transmisă linie cu linie. La sosire se verifică toți biții de paritate. Dacă oricare din ei este greșit se va cere retransmisia blocului. Această strategie poate detecta o singură rafală de lungime l, aceasta modificând maxim un bit pe coloană. Astfel, metoda utilizează l biți de paritate pentru a asigura imunitatea unor blocuri de l-i biți la erori în rafală ce modifică maxim l biți.

Suma de control (checksum)

Termenul sumă de control este utilizat de obicei pentru a defini un grup de biți redundanți asociați mesajului, indiferent de modul în care sunt calculați. Astfel, un grup de biți de paritate reprezintă o sumă de control. Totuși, există și alte sume de control, mai puternice ce presupun adunarea datelor organizate în cuvinte de câte n biți, rezultând o sumă pe n biți. Pentru ca suma să nu depășească cei n biți, se utilizează aritmetică în complement de 1.

Spre exemplu, unele protocole de nivel 3, respectiv 4 (IP, TCP, UDP) utilizează o sumă de control pe 16 biți, denumită Internet Checksum.

Transmițătorul calculează suma de control utilizând următorii pași:

1. Mesajul este împărțit în cuvinte pe 16 biți.
2. Valoarea sumei de control este setată pe 0.
3. Toate cuvintele, inclusiv sumă de control, sunt adunate în complement de 1.
4. Suma obținută este complementată și devine suma de control.
5. Suma de control este transmisă împreună cu datele.

Receptorul utilizează următorii pași pentru identificarea erorilor:

1. Mesajul (inclusiv suma de control) este împărțit în cuvinte pe 16 biți.
2. Toate cuvintele sunt adunate în complement de 1.
3. Suma este complementată și devine noua sumă de control.
4. Dacă valoarea sumei de control este 0, mesajul este acceptat.

CRC (Cyclic Redundancy Check)

Bitul de paritate și suma de control sunt utilizate, în general, la nivelele superioare. Nivelul doi utilizează, de obicei, pentru detecția erorilor CRC, denumit și cod polinomial. Codurile polinomiale tratează o însiruire de biți ca fiind o reprezentare a unui polinom având doar coeficienți de 0 sau 1. Un cadru de date de dimensiune k , este privit ca și lista de coeficienți a unui polinom format din k termeni ($x^{k-1} \dots x^0$), având gradul $k-1$. Spre exemplu 110101 reprezintă un polinom cu 6 termeni, având coeficienții 110101, adică: $1x^5 + 1x^4 + 0x^3 + 1x^2 + 0x^1 + 1x^0$. Algebra polinomială este realizată modulo 2, după regulile teoriei câmpurilor. Adică toate calculele sunt realizate în binar, fără transport. În acest caz, atât operațiile de adunare, cât și cele de scădere sunt identice cu XOR. Împărțirea se realizează la fel ca și în binar, doar că scăderea este din nou modulo 2.

Când utilizăm un cod polinomial, transmițătorul și receptorul trebuie să cadă de acord asupra unui generator polinomial $G(x)$, în care bitul cel mai semnificativ și bitul cel mai puțin semnificativ sunt 1. Pentru a putea calcula CRC-ul unui cadru de date, asociat polinomului $M(x)$, având k biți, acesta trebuie să fie mai mare decât polinomul generator. Ideea este să calculăm CRC-ul în aşa fel încât, în urma atașării sale la cadru, polinomul reprezentat de cadru + CRC să fie divizibil cu $G(x)$. Când receptorul

primește cadrul ce conține CRC, va încerca să îl împartă la $G(x)$. Dacă restul împărțirii este 0 cadrul este corect, în caz contrar a apărut o eroare.

Algoritmul de calcul al CRC-ului este următorul:

1. Fie r gradul lui $G(x)$. Atașează r biți de zero la capătul ce conține biții cei mai puțin semnificativi ai cadrului de date. În acest fel, cadrul va conține $k + r$ biți și va fi asociat polinomului $x^r M(x)$.
2. Divide modulo 2 însiruirea de biți asociată lui $x^r M(x)$ cu cea asociată lui $G(x)$.
3. Scade modulo 2 restul împărțirii (ce va fi mai mic sau egal cu r) din $x^r M(x)$, iar rezultatul obținut va fi cadrul de date ce conține CRC-ul și este pregătit pentru transmisie.

În continuare vom calcula, ca și exemplu, CRC-ul pentru cadrul 1101011111, folosind generatorul $G(x) = x^4 + x + 1$.

1. Generatorul având gradul 4 vom atașa 4 biți de zero cadrului rezultând 11010111110000.

2.

11010111110000 / 10011

10011

010011

10011

00000

11110

10011

011010

10011

010010

10011

000010

3. Scăzând restul, vom obține cadrul ce conține CRC-ul: 11010111110010

Standardul Ethernet

Scurt istoric

Ethernet a fost dezvoltat între anii 1972-1974 de Bob Metcalfe și David Boggs, în cadrul centrului de cercetare al companiei Xerox PARC (Palo Alto Research Center), cu scopul de a interconecta stațiile de lucru de tip PC, Xerox Altos și imprimantele laser de mare viteză dezvoltate de Xerox.

În 1980, un consorțiu format din companiile Digital, Xerox și Intel publică primul standard Ethernet sub denumirea de DIX Standard. Ultima revizie a acestui standard DIX V2.0 a fost publicată în 1982.

În paralel, în 1980, IEEE începe proiectul 802, în vederea standardizării unor tehnologii LAN/MAN. Ethernet, cu unele modificări, este aprobat de către IEEE ca și standard în 1982, iar în 1985 este publicat sub denumirea de IEEE 802.3 CSMA/CD. Ulterior denumirea a fost modificată în 802.3 Ethernet.

În anii care au urmat, standardul Ethernet a fost modificat în mod continuu, adăugându-se alte medii de transmisie și alte capabilități, în vederea creșterii flexibilității și a performanței. Modificările aduse standardului, sunt publicate inițial de către IEEE ca și suplimente ale acestuia. Odată finalizat procesul de standardizare, suplimentele devin parte integrantă din standard, nemaifiind publicate separat. Denumirile cătorva suplimente, împreună cu anii când au fost integrate în standard sunt prezentate în tabelul ce urmează.

Denumire	An	Descriere
802.3a	1988	10BASE2 thin Ethernet
802.3c	1985	10 Mb/s repeater specifications
802.3i	1990	10BASE-T twisted-pair
802.3j	1993	10BASE-F fiber optic
802.3u	1995	100BASE-TX, 100BASE-T4, 100BASE-FX Fast Ethernet with Auto-Negotiation
802.3x	1997	Full-duplex standard and Flow Control
802.3z	1998	1000BASE-X Gigabit Ethernet
802.3ab	1999	1000BASE-T Gigabit Ethernet over twisted-pair
802.3ac	1998	Frame size extension to 1,522 bytes for VLAN tag
802.3ad	2000	Link aggregation for parallel links
802.3ae	2002	10 Gb/s Ethernet
802.3af	2003	Power over Ethernet
802.3ak	2004	10GBASE-CX4 10 Gigabit Ethernet over short-range coaxial cable
802.3an	2006	10GBASE-T 10 Gigabit Ethernet over twisted-pair
802.3as	2006	Frame expansion to 2,000 bytes for all tagging
802.3az	2010	Energy-efficient Ethernet
802.3ba	2010	40 Gb/s and 100 Gb/s Ethernet

Cateva suplimente IEEE 802.3

Componentele sistemului Ethernet

O rețea Ethernet este compusă dintr-o serie de componente hardware și software ce conlucrează pentru a permite transferul datelor între echipamentele de calcul. În cei aproape 50 de ani de existență, standardul a fost modificat continuu, definindu-se o serie de variante de implementare, ce utilizează echipamente și medii de transmisie diferite.

Principalele elemente ce descriu o rețea Ethernet, la care facem referire atunci când prezentăm o anumită variațiune, sunt:

- **Cadrul de date:** un set standard de biți, utilizați pentru transmisia datelor prin sistem.
- **Protocolul de control al accesului la mediu:** set de reguli înglobat în interfața Ethernet, care definește modul de acces al stațiilor la canalul de transmisie.
- **Componentele de semnalizare, tipuri de semnale și codare:** echipamentele electronice standardizate care transmit și recepționează semnalele, precum și tipurile de semnale, respectiv tehniciile de codare, utilizate.
- **Mediul fizic:** cablurile, conectorii și celelalte echipamente hardware pasive utilizate pentru transmisia datelor între noduri.

Structura cadrului de date

8 bytes	6 bytes	6 bytes	2 bytes	46 – 1500 bytes	4 bytes
Preambul	Adresa destinație	Adresa sursă	Tip	Date	Suma de control

Standardul DIX

7 bytes	1 byte	6 bytes	6 bytes	2 bytes	46 – 1500 bytes	4 bytes
Preambul	S F D	Adresa destinație	Adresa sursă	Lungime / Tip	Date / LLC	Suma de control

Standardul IEEE 802.3

unde:

- **preambulul** reprezintă o secvență de 7 bytes, fiecare având structura 10101010, utilizată pentru initializarea și sincronizarea interfețelor de rețea;
- **SFD** (start of frame delimiter) – are structura 10101011 și marchează terminarea preambulului;

In cazul standardului DIX, SFD este integrat în preambul, însă are aceeași structură.

- **adresa destinație** definește adresa fizică a echipamentului ce recepționează cadrul;
- **adresa sursă** definește adresa fizică a echipamentului ce transmite cadrul;
- **tipul** indică protocolul de nivel rețea transportat. Codurile încep de la $0x600 = 1536$ și sunt gestionate de IEEE. Spre exemplu, $0x800$ este codul asociat protocolului IP;
- **Lungimea** reprezintă numărul de octeți din câmpul de date (minim 46, maxim 1500). Inițial, în standardul IEEE 802.3, câmpul tip a fost înlocuit cu câmpul lungime. Ulterior, standardul a fost actualizat, astfel încât câmpul să permită fie definirea tipului, fie definirea lungimii cadrului. Diferențierea între cele două se face în funcție de valoarea stocată.
- câmpul **date** conține informația primită de la nivelul ierarhic superior, pentru a fi încapsulată în cadrul de date;
- **suma de control** este utilizată pentru identificarea apariției unei erori, codul implementat fiind CRC.

Cadre de date extinse

7 bytes	1 byte	6 bytes	6 bytes	4 bytes	2 bytes	46 – 1500 bytes	4 bytes
Preambul	S F D	Adresa destinație	Adresa sursă	Q-Tag	Lungime / Tip	Date / LLC	Suma de control

Cadru extins, definit în IEEE 802.3ac

7 bytes	1 byte	6 bytes	6 bytes	2 bytes – 482 bytes	2 bytes	46 – 1500 bytes	0 - 482 bytes	4 bytes
Preambul	S F D	Adresa destinație	Adresa sursă	Prefix plic	Lungime / Tip	Date / LLC	Sufix plic	Suma de control

Cadru extins de tip plic, definit în IEEE 802.3as

unde:

- Q-Tag este utilizat pentru definirea de priorități, respectiv de rețele virtuale locale (VLANs). Primii 2 octeți conțin identificatorul 0x8100 pentru compatibilitatea cu cadrele standard;
- prefixul și / sau sufixul, în cazul cadrelor de tip plic introduse în suplimentul 802.3as, extind cadrul standard până la maxim 2000 bytes fără preambul, pentru a permite adăugarea unor parametri proprietari (definiți de producător);

Adresarea

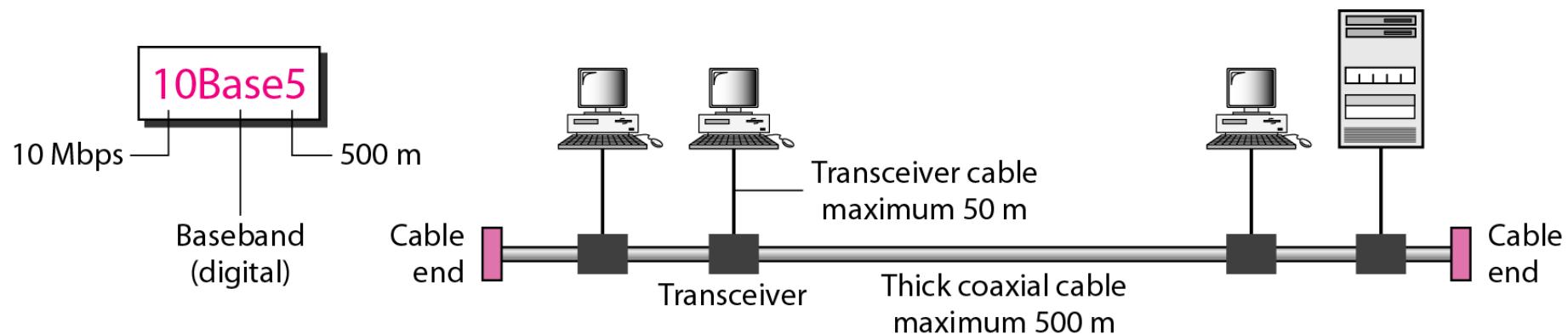
Toate echipamentele conectate într-o rețea Ethernet (stații de calcul, imprimante, etc.) conțin cel puțin o interfață de rețea specifică (NIC – network interface card). Orice interfață de rețea Ethernet are definită o adresă pe 6 octeți, denumită adresă fizică sau MAC. Primii 3 octeți reprezintă identificatorul producătorului de echipamente Ethernet. Fiecare producător va avea cel puțin un astfel de identificator unic. Gestionația acestor identificatori este realizată de către IEEE. Ultimii 3 octeți din adresa fizică selectează un singur echipament din lotul producătorului, asigurându-se astfel unicitatea adreselor MAC la nivel global.

Adresele fizice pot fi unicast, multicast sau broadcast. Diferențierea se face în funcție de ultimul bit din primul octet al unei adrese (octetul cel mai semnificativ). O adresă unicast va avea acest bit pe valoarea 0, în timp ce o adresă unicast va avea bitul pe valoarea 1. Dacă toți bitii din adresă au valoarea 1, atunci adresa este de tip broadcast. În antetul unui cadru de date, adresa fizică sursă va fi întotdeauna de tip unicast. Administratorul unei rețele Ethernet poate alege să folosească adresele fizice definite de producători, ce le asigură o unicitate la nivel global, sau poate să modifice aceste adrese, asigurându-se de o unicitate a lor la nivel local. Diferențierea dintre adresele administrate global sau local se face în funcție de valoarea celui de-al doilea bit din octetul cel mai semnificativ.

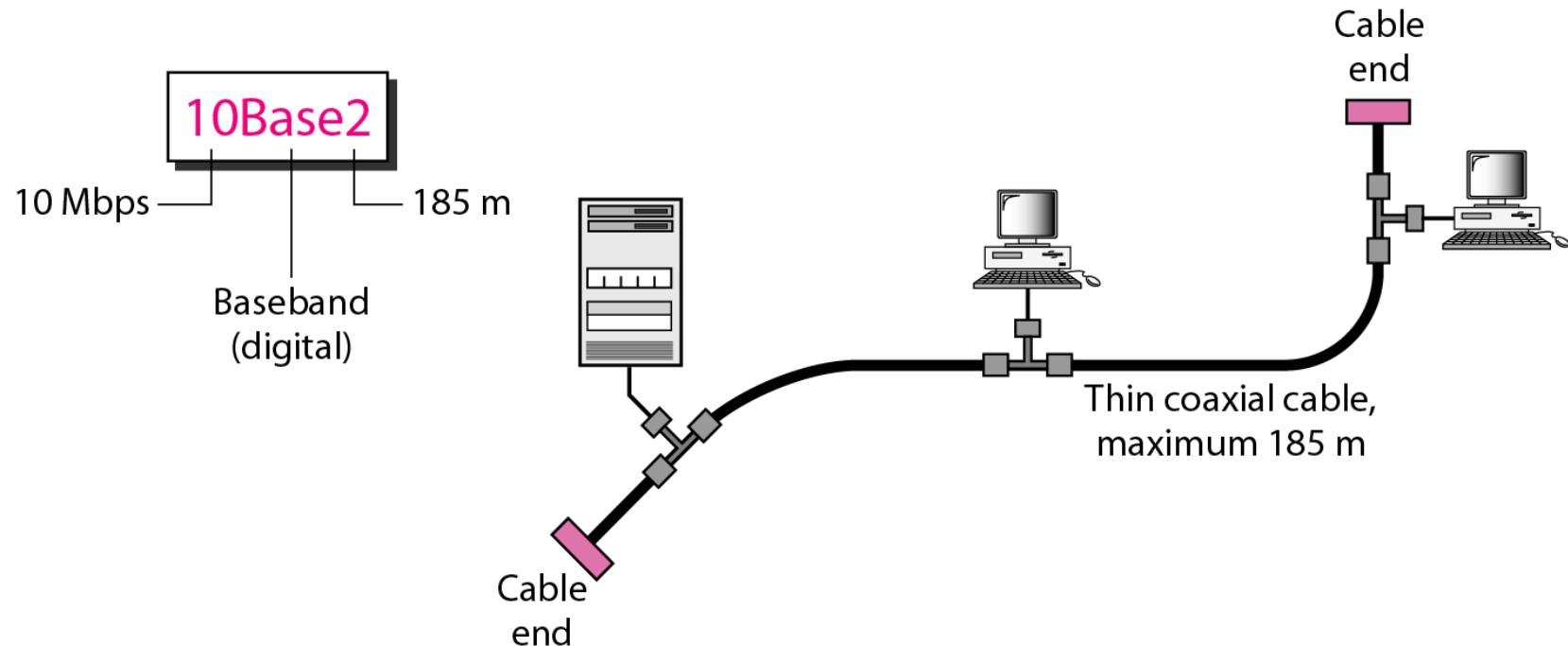
Tipuri de implementări

Denumire	Categorie	Viteză	Topologie	Mediu de transmisie	Perechi utilizate	Tip duplex	Protocol MAC	Codare	Lungime maximă
10Base5	Classic Ethernet	10 Mb/s	magistrală	cablu coaxial gros	-	half	CSMA/CD	Manchester	500m
10Base2	Classic Ethernet	10 Mb/s	magistrală	cablu coaxial subțire	-	half	CSMA/CD	Manchester	185m
10Base-T	Switched Ethernet	10 Mb/s	stea	cablu UTP cat. 3+	2	half full	CSMA/CD -	Manchester	100m
10Base-F	Switched Ethernet	10 Mb/s	stea	fibră optică	1	half full	CSMA/CD -	Manchester	2000m
100Base-TX	Switched Ethernet	100 Mb/s	stea	cablu UTP cat. 5+	2	half full	CSMA/CD -	MLT-3 4B/5B	100m
100Base-FX	Switched Ethernet	100 Mb/s	stea	fibră optică	1	half full	CSMA/CD -	NRZ-I 4B/5B	100m
100Base-T4	Switched Ethernet	100 Mb/s	stea	cablu UTP cat. 3+	4	half full	CSMA/CD -	8B/6T	100m
1000Base-T	Switched Ethernet	1000 Mb/s	stea	cablu UTP cat. 5+	4	full	-	4D-PAM-5	100m
1000Base-C	Switched Ethernet	1000 Mb/s	stea	cablu UTP cat. 6+	2	full	-	NRZ 8B/10B	30m
1000Base-LX	Switched Ethernet	1000 Mb/s	stea	fibră optică	1	full	-	NRZ 8B/10B	5000m

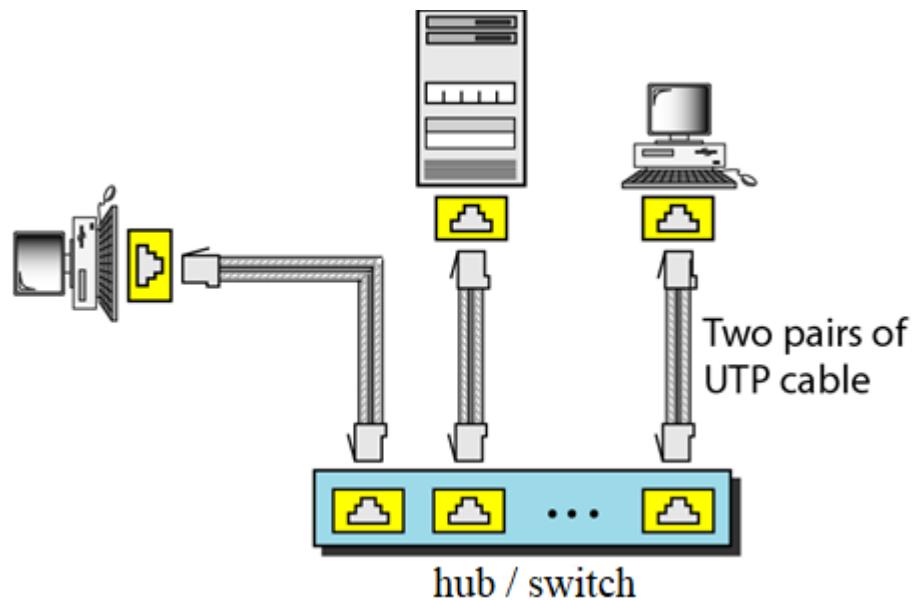
Mediu fizic -10Base5



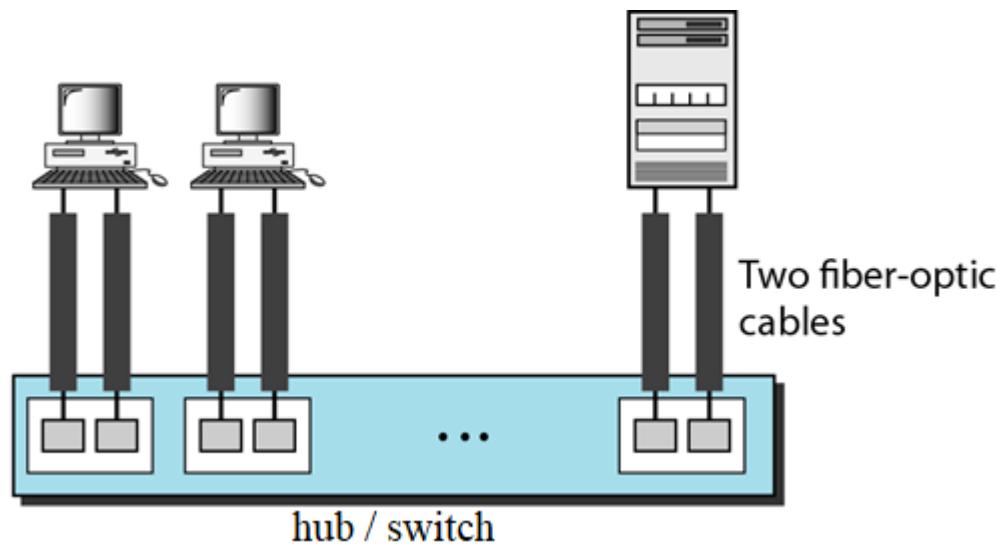
Mediul fizic -10Base2



Mediul fizic -10Base-T, 100Base-TX,1000Base-T



Mediul fizic -10Base - F, 100Base – FX



Protocole suport utilizate în cazul transmisiei cu mediu partajat

Protocolul ARP

ARP (Address Resolution Protocol)

- Asociază în mod dinamic o adresă fizică (hardware) la o adresă logică (IP);
- Se utilizează în rețele cu acces multiplu la mediu și se bazează pe difuzare.

Funcționare:

- Difuzează un pachet conținând adresa IP a nodului destinație;
- Nodul având adresa IP destinație trimite un pachet răspuns care conține adresa sa fizică;
- Adresa fizică primită este utilizată pentru transmisia pachetului original către destinație.

ARP Cache

- Pentru utilizarea eficientă a protocolului se mențin într-o memorie tampon sub forma unui tabel (tabelă ARP) asocierile de tip adrese fizice – adrese IP identificate recent;
- Fiecare înregistrare din tabelă are asociat și un timp de viață fortându-se reactualizarea periodică a informațiilor din tabelă;
- La primirea unui alt pachet ARP răspuns conținând o înregistrare existentă, acea înregistrare este actualizată automat.

Gratuitous ARP

Definim Gratuitous ARP cazul în care un nod transmite o cerere ARP căutând adresa fizică asociată adresei sale IP.

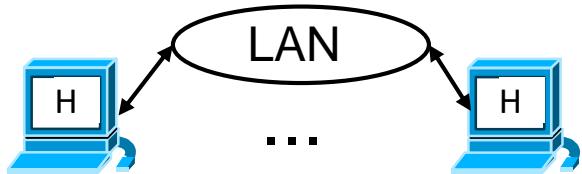
Utilitate:

- Identifică un posibil conflict de adrese IP;
- În cazul unei modificări recente a adresei fizice, mesajul actualizează tabelele ARP ale nodurilor vecine.

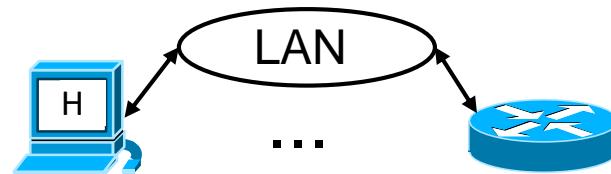
Antetul pachetului ARP

Adresa destinație (6 bytes)	Adresa sursă (6 bytes)	Tip protocol (2 bytes) 0x8060 ARP 0x8035 RARP	ARP Cerere/Răspuns	CRC
Tip hardware (2 bytes) (Ethernet - 1)		Tip protocol (2 bytes) (IP – 0800h)		
Lungime adresă hardware(1byte)	Lungime adresă protocol (1byte)	Codul operației (1,2 – Cerere/Răspuns ARP 3,4 – Cerere/Răspuns RARP)		
Adresă sursă hardware (6 bytes pentru Ethernet)				
Adresă sursă protocol (4 bytes pentru IP)				
Adresă destinație hardware (6 bytes pentru Ethernet)				
Adresă destinație protocol (4 bytes pentru IP)				

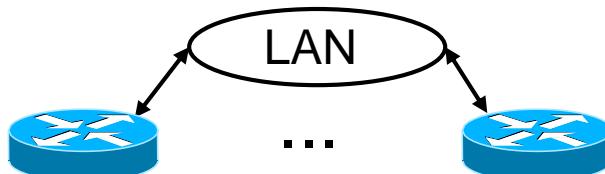
Utilizare ARP



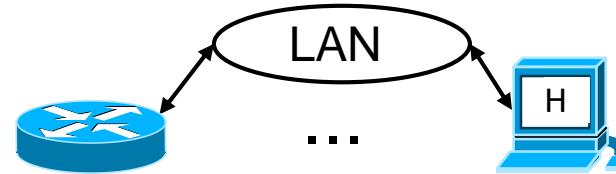
Cazul 1: Un nod transmite date la un alt nod în aceeași rețea



Cazul 2: Un nod are de transmis date la un alt nod în altă rețea. Inițial transmite datele la ruter.



Cazul 3: Un ruter primește un pachet destinat unui nod în altă rețea.
Transmite pachetul către un alt ruter.



Cazul 4: Un ruter primește un pachet care trebuie transmis unui nod în aceeași rețea.

Protocolul RARP

- Utilizat pentru obținerea adresei IP asociată unei adrese fizice.
- Se folosește pentru pornirea stațiilor fără disk (fără sistem de operare local).

Algoritmi de dirijare

Algoritmul de dirijare este acea parte a software-ului nivelului rețea care determină calea optimă dintre noduri și care răspunde de alegerea liniei de ieșire pe care un pachet recepționat trebuie transmis mai departe.

Un algoritm trebuie să fie: corect, simplu, stabil, robust, eficient, să prevină defavorizarea nodurilor.

Algoritmii pot fi:

neadaptivi – alegerea căii se calculează în avans (offline) și se transmite ruterului la inițializarea rețelei;

adaptivi – își modifică deciziile de dirijare pentru a reflecta modificările de topologie și pe cele de trafic.

A doua categorie este utilizată în protocolele de rutare.

Algoritmii de dirijare adaptivi diferă prin:

- locul de unde își iau informația:
 - de la ruterele vecine;
 - de la toate ruterele.
- momentul în care schimbă rutele:
 - când se schimbă încărcarea;
 - când se schimbă topologia.
- metrică:
 - distanță;
 - timp de transmisie;
 - număr de hopuri;
 - etc.

Două categorii de algoritmi adaptivi, foarte utilizate în protocoalele de rutare:

- ce se bazează pe starea legăturilor;
- ce utilizează vectori distanță.

Dirijarea pe calea cea mai scurtă

Atunci când se poate afla arhitectura rețelei, identificarea rutei optime se poate reduce la aflarea căii celei mai scurte. Metoda necesită construcția unui graf al rețelei, fiecare ruter fiind un nod al grafului, iar fiecare linie de comunicație fiind un arc, având un anumit cost (distanță) în funcție de metrică folosită în aprecierea liniei respective. Metrica poate fi: distanța în km, numărul de salturi, rata de transfer, traficul mediu, costul comunicației, lungimea minimă a cozilor de așteptare, întârzierile măsurate, sau o combinație a acestora.

Algoritmul găsește calea cea mai scurtă (valoarea minimă a metricii) între două rutere. Cel mai cunoscut și utilizat algoritm de determinare a căii celei mai scurte este Dijkstra (1959) și se bazează pe principiul optimalității.

Principiul optimalității:

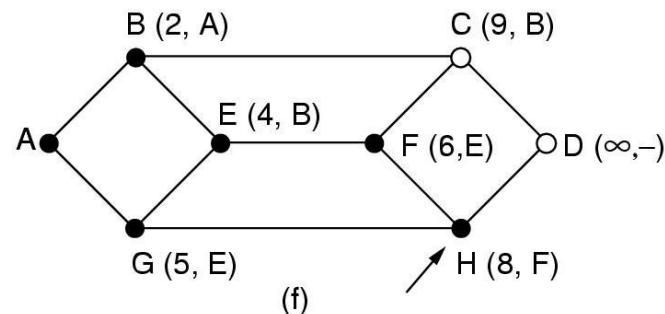
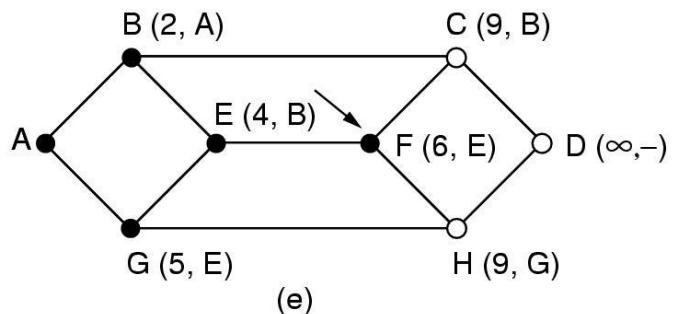
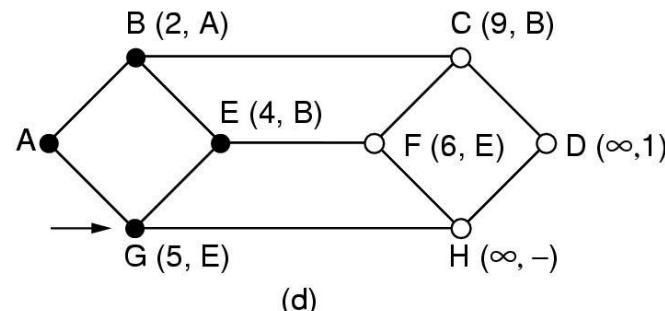
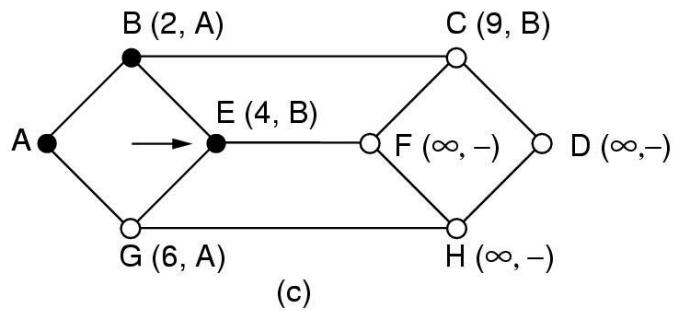
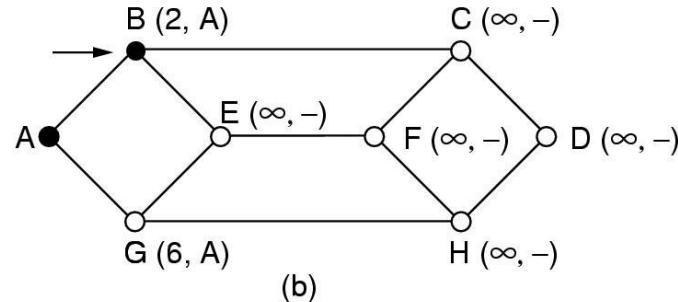
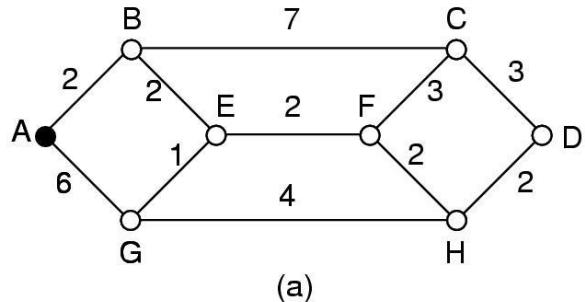
Dacă ruterul J este pe calea optimă de la ruterul I la ruterul K, atunci calea optimă de la J la K este pe aceeași rută (se poate demonstra prin metoda reducerii la absurd).

Algoritmul Dijkstra:

1. Sunt marcate toate nodurile vizate, stabilindu-se distanța de la sursă la ea însăși ca fiind 0 și la toate celelalte noduri ca fiind ∞ .
2. Pornind de la sursă, atât timp cât mai există noduri vizate:
 - extrage nodul cu distanța cea mai mică;
 - adaugă arcul la subgraful ce definește calea cea mai scurtă;
 - relaxează distanțele vecinilor nodului.

Noțiuni introductive

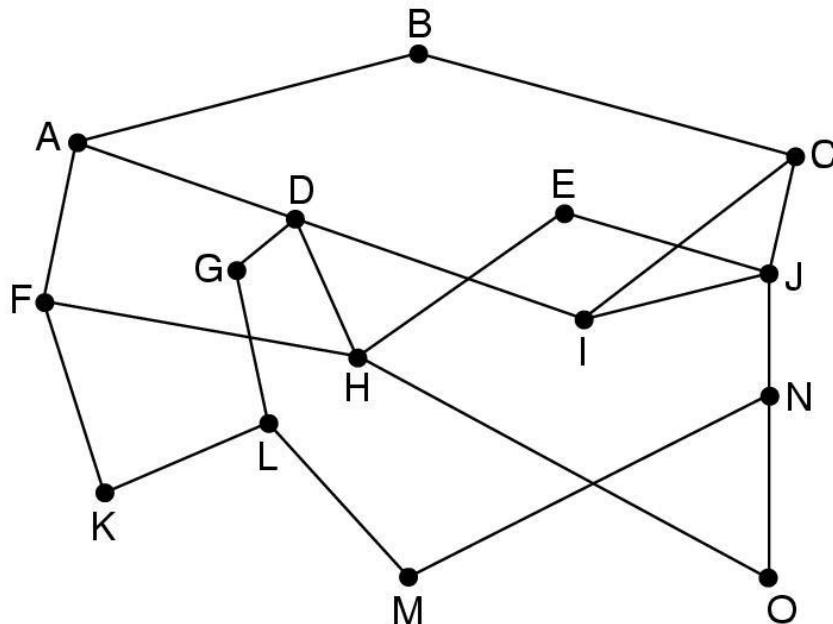
Exemplu: calea cea mai scurtă de la A la H utilizând algoritmul Dijkstra



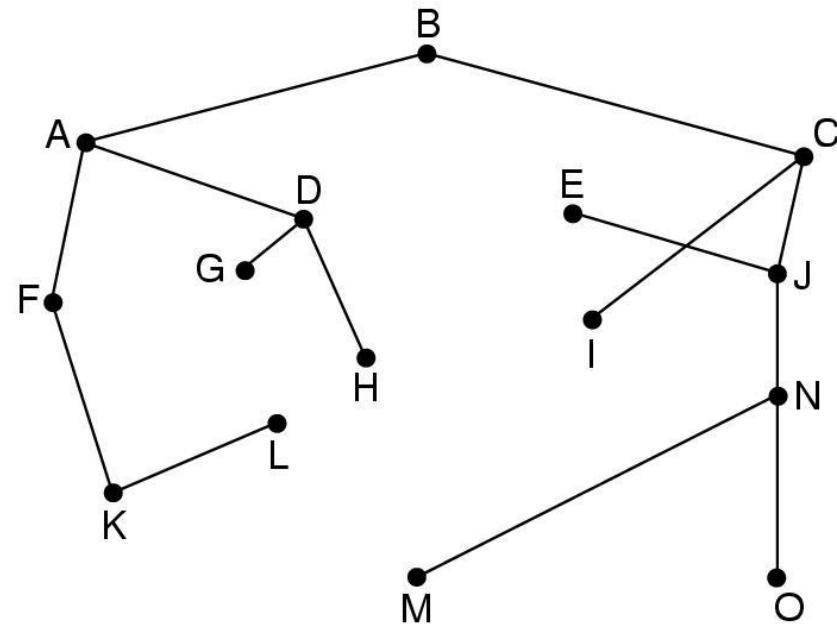
Arborele de dirijare

Mulțimea rutelor optime către o anumită destinație formează un arbore, având destinația ca rădăcină, care se cheamă arbore de scufundare. Similar, mulțimea tuturor rutelor optime ce pornesc de la o anumită sursă se numește arbore sursă.

Pot exista mai mulți arbori de scufundare sau sursă pentru o arhitectură de rețea dată.



(a)



(b)

Algoritmul de inundare

- orice pachet primit este retransmis pe toate căile mai puțin cea de pe care a venit.
- decizie bazată pe cunoștințe locale, fară o cunoaștere a topologiei.
- generează foarte multe pachete.
- pentru a limita impactul asupra rețelei:

1. antetul fiecărui pachet contine un contor de salturi decrementat la fiecare salt și care face ca pachetul să fie distrus atunci când contorul atinge valoarea 0.

2. ruterul sursă plasează un număr de secvență în pachetul transmis.

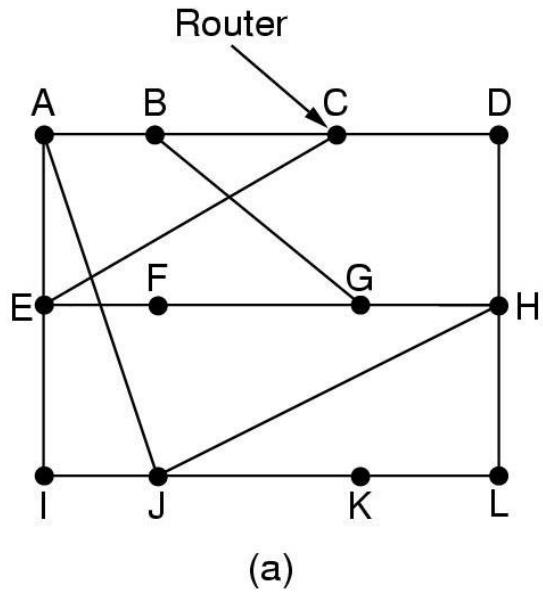
Fiecare ruter menține o listă cu numerele de secvență a pachetelor deja transmise pentru a evita retransmisia.

Algoritmul de dirijare cu vectori distanță (Bellman – Ford)

Fiecare ruter menține o tabelă (un vector), care păstrează cea mai bună distanță cunoscută spre fiecare destinație și linia care trebuie urmată pentru a ajunge acolo. Aceste tabele sunt actualizate prin schimbul de informații între nodurile vecine.

Se pot folosi diferite metriki. Dacă se folosește metrica salturilor, distanța este de doar un salt. Dacă metrica folosită este lungimea cozilor de aşteptare, ruterul examinează lungimile acestor cozi. Dacă metrica este cea a întârzierilor ruterul o poate măsura direct prin pachetele speciale numite ECHO în care receptorul va marca doar timpul curent și le va trimite înapoi cât mai repede.

Algoritmul cu vectori distanță



New estimated delay from J

Line

To	A	I	H	K	Line
A	0	24	20	21	8 A
B	12	36	31	28	20 A
C	25	18	19	36	28 I
D	40	27	8	24	20 H
E	14	7	30	22	17 I
F	23	20	19	40	30 I
G	18	31	6	31	18 H
H	17	20	0	19	12 H
I	21	0	14	22	10 I
J	9	11	7	10	0 -
K	24	22	22	0	6 K
L	29	33	9	9	15 K

JA delay is 8 JI delay is 10 JH delay is 12 JK delay is 6

Vectors received from J's four neighbors

New routing table for J

(b)

Algoritmul cu vectori distanță

Problema incrementării la infinit

A	B	C	D	E	
•	•	•	•	•	Initially
1	•	•	•	•	After 1 exchange
1	2	•	•	•	After 2 exchanges
1	2	3	•	•	After 3 exchanges
1	2	3	4	•	After 4 exchanges

(a)

A	B	C	D	E	
•	X	•	•	•	Initially
1	2	3	4	•	After 1 exchange
3	2	3	4	•	After 2 exchanges
3	4	3	4	•	After 3 exchanges
5	4	5	4	•	After 4 exchanges
5	6	5	6	•	After 5 exchanges
7	6	7	6	•	After 6 exchanges
7	8	7	8	•	
⋮	⋮	⋮	⋮	⋮	
•	•	•	•	•	

(b)

Algoritm de dirijare folosind starea legăturilor

Conform acestui algoritm fiecare ruter trebuie:

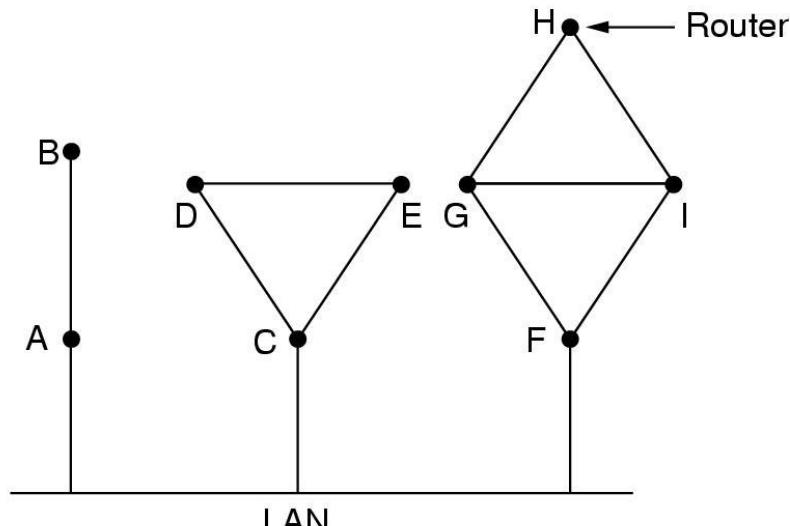
1. Să descopere care sunt vecinii săi și să afle adresele de rețea ale acestora.
2. Să determine costul până la vecinii săi.
3. Să pregătească un pachet cu datele culese despre vecini.
4. Să trimită acest pachet către toate celelalte rutere.
5. Să calculeze cea mai scurtă cale spre fiecare ruter.

Algoritmul bazat pe starea legăturilor

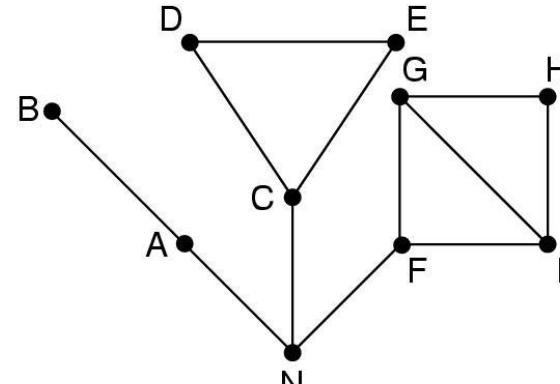
Identificarea vecinilor

Se rezolvă prin trimitera unor pachete speciale HELLO pe fiecare linie care face legătura cu un alt ruter. Partenerul este obligat să răspundă anunțându-și identitatea.

Dacă mai multe rutere sunt conectate printr-o rețea cu difuzare, rețeaua se modelează ca un nou nod (artificial).



(a)



(b)

Un exemplu de **măsurare a costului**, atunci când metrica este întârzierea în rețea:

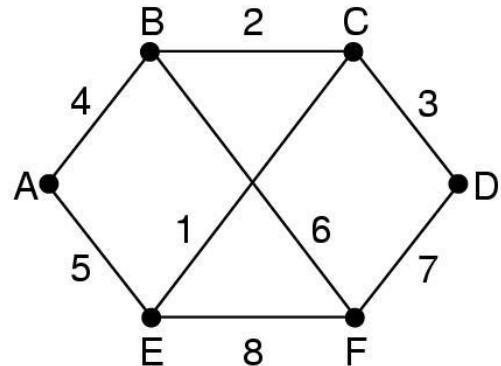
- se transmite un pachet ECHO la un ruter vecin, cerând partenerului să-l returneze imediat;
- contorizând timpul necesar propagării (RTT – round trip time) se obține o estimare rezonabilă a întârzierii.

Întârzierea poate ține cont de coadă sau nu; în primul caz timpul se măsoară din momentul așezării pachetului ECHO în coadă, iar în al 2-lea caz din momentul când pachetul ajunge pe prima poziție din coadă.

Algoritmul bazat pe starea legăturilor

Construcția pachetului constă în completarea de către fiecare ruter a unui pachet conținând: identitatea sa; un număr de secvență; vârsta (timpul de viață); lista vecinilor cu întârzierea asociată.

Pachetele pot fi reconstruite periodic sau la producerea unui eveniment: oprirea unui vecin; pornirea unui vecin; modificări de parcurs (rută).



(a)

Link	State			
A	B	C	D	E
Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5
E 5	C 2	D 3	F 7	B 6
	F 6	E 1		D 7
				E 8

(b)

Distribuția pachetelor

Pachetele se distribuie prin inundare. Pentru a avea controlul inundării, fiecare pachet conține un număr de secvență care este incrementat de ruter la fiecare nou pachet trimis. Ruterele păstrează evidența tuturor perechilor (ruter sursă, număr secvență) pe care le văd. La sosirea unui nou pachet cu starea legăturilor, el este căutat în lista pachetelor deja văzute. Dacă pachetul este nou, el este trimis pe toate liniile, cu excepția celei de pe care a sosit. Dacă este duplicat, pachetul este distrus. Dacă pachetul sosit are un număr de secvență mai mic decât cel mai mare număr de secvență detectat, atunci el este rejectat (distrus), ca fiind învecit. Vârsta se decrementează la fiecare hop, iar când ajunge la 0, pachetul este distrus.

Calcularea noilor rute o face fiecare ruter de îndată ce a acumulat un set complet de pachete cu starea legăturilor. Incepe prin a construi graful rețelei, iar apoi cu ajutorul unui algoritm de dirijare (ex. Dijkstra), găsește calea cea mai scurtă către toate destinațiile (arborele de scufundare).

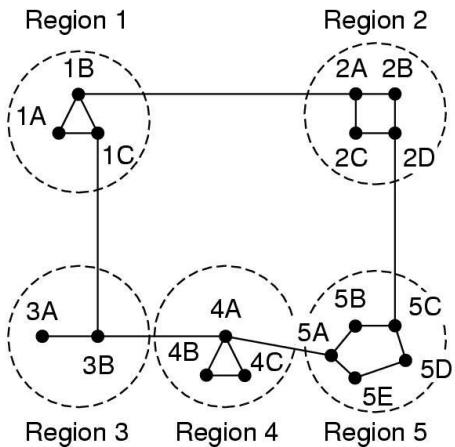
În rețelele foarte mari, poate fi critică memoria ocupată de aceste pachete, sau timpul de interpretare.

Comparație DV - LS

- **DV**
 - Transmit întreaga tabelă de rutare;
 - Actualizari periodice;
 - Convergența greoaie;
 - Puțin scalabile;
 - + Folosesc mai puține resurse;
 - + Transmit informații la vecini;
 - + Sunt mai ușor de configuraț.
- **LS**
 - Cerințe mai mari de hardware;
 - Transmit informații în întreaga rețea (porțiuni din tabela de rutare);
 - + Imagine de ansamblu a rețelei;
 - + Actualizări determinate de schimbări în topologie;
 - + Mai puțin predispuse la bucle;
 - + Convergență rapidă.

Algoritmi de dirijare

Rutarea ierarhică: rețeaua este împărțită în **regiuni**, fiecare ruter știind toate detaliile necesare pentru a dirija pachete spre orice ruter destinație în cadrul regiunii sale, respectiv spre regiune, în cazul în care ruterul destinație se află în altă regiune.



(a)

Full table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

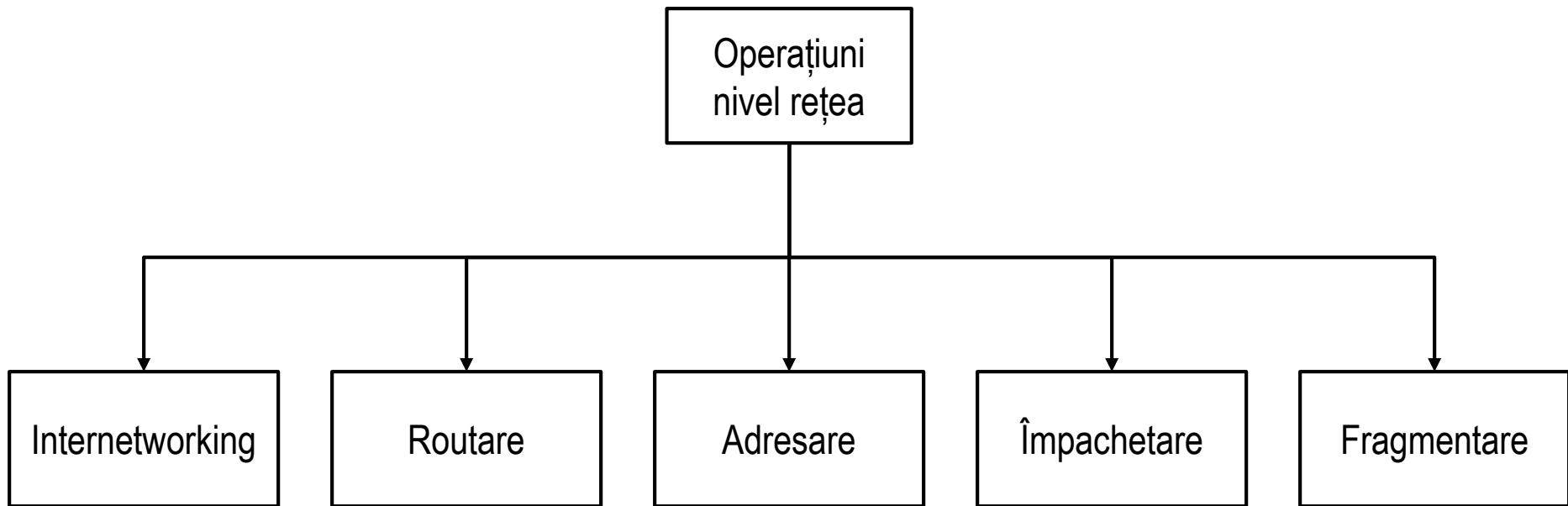
Hierarchical table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

Nivelul rețea

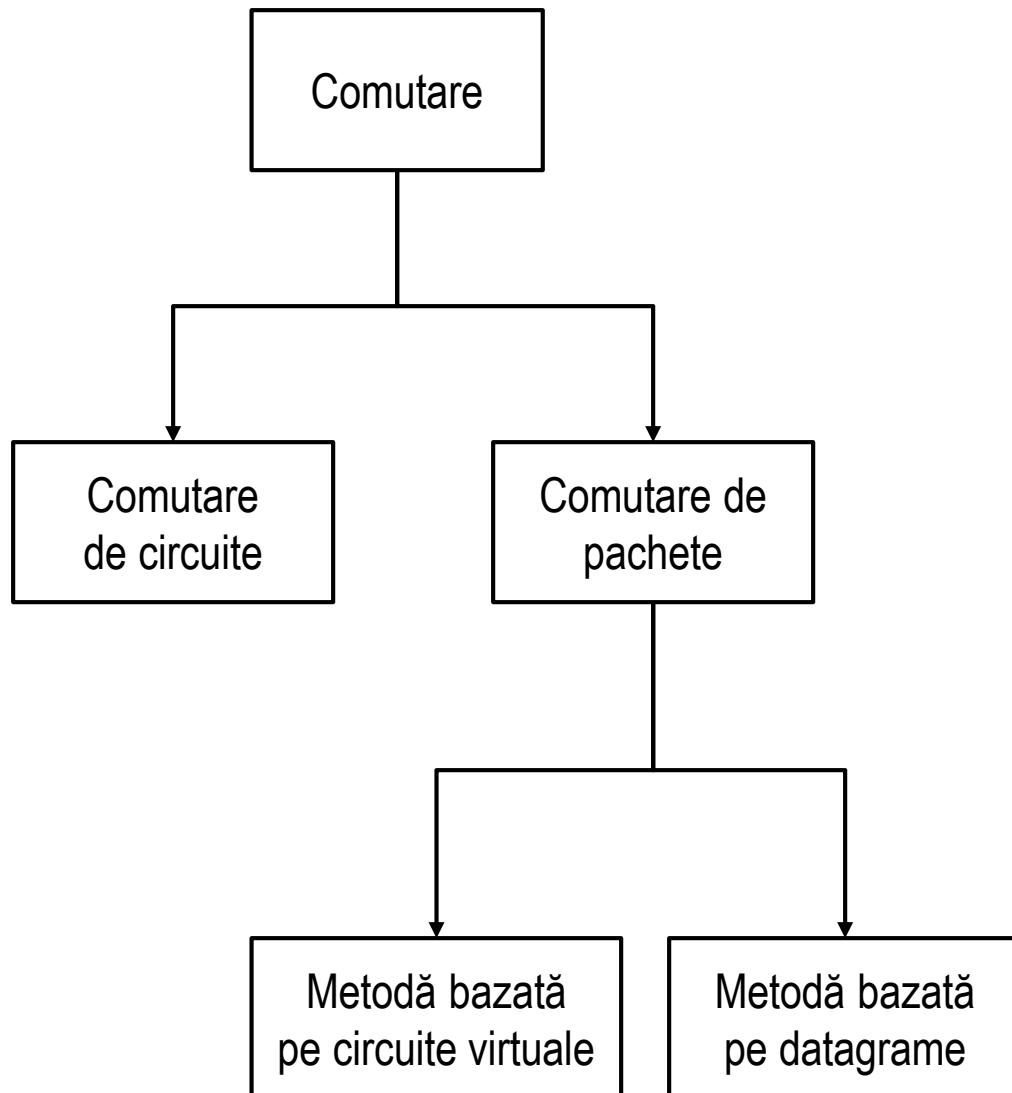
Atribuțiile nivelului rețea



Clase de protocoale

- Există două clase de protocoale utilizate la nivelul rețea:
 - protocoale rutate (routed protocols)
 - protocoale de rutare (routing protocols)
- **Protocolele de rutare** determină regulile prin care ruterele schimbă informații despre accesibilitatea rețelelor. Tabela de rutare este construită în mod dinamic în funcție de informațiile furnizate de protocolele de rutare. Pe baza tabelei de rutare este determinat traseul pe care trebuie trimis fiecare pachet.
 - Exemple de protocole de rutare: RIP, OSPF, BGP.
- **Protocolele rutate** sunt protocolele transportate între rețele.
 - Exemple: IP, IPX, Appletalk, DECnet.

Comutarea datelor



Tipuri de transmisie a datelor

Tipuri de transmisie a datelor:

1. **Orientat pe conexiune;**
2. **Neorientat pe conexiune.**

Transmisii de date **neorientate pe conexiune**:

- Destinatia nu este contactata înainte de trimiterea unui pachet.
- Pachetele pot urma căi diferite pentru a ajunge la destinație, și pot ajunge într-o ordine diferită față de cea în care au fost trimise.
- Aceste procese mai sunt denumite **packet-switched**.
- Asemănătoare sistemului poștal.

Transmisii de date **orientate pe conexiune**:

- Este stabilită o conexiune între sursă și destinație înainte de transmiterea oricăror date;
- Overhead mai mare față de protocolele neorientate pe conexiune;
- Toate pachetele circula pe același traseu fizic sau virtual, în ordinea în care sunt trimise;
- Aceste procese sunt denumite **virtual circuit-switched**;
- Asemănătoare sistemului telefonic.

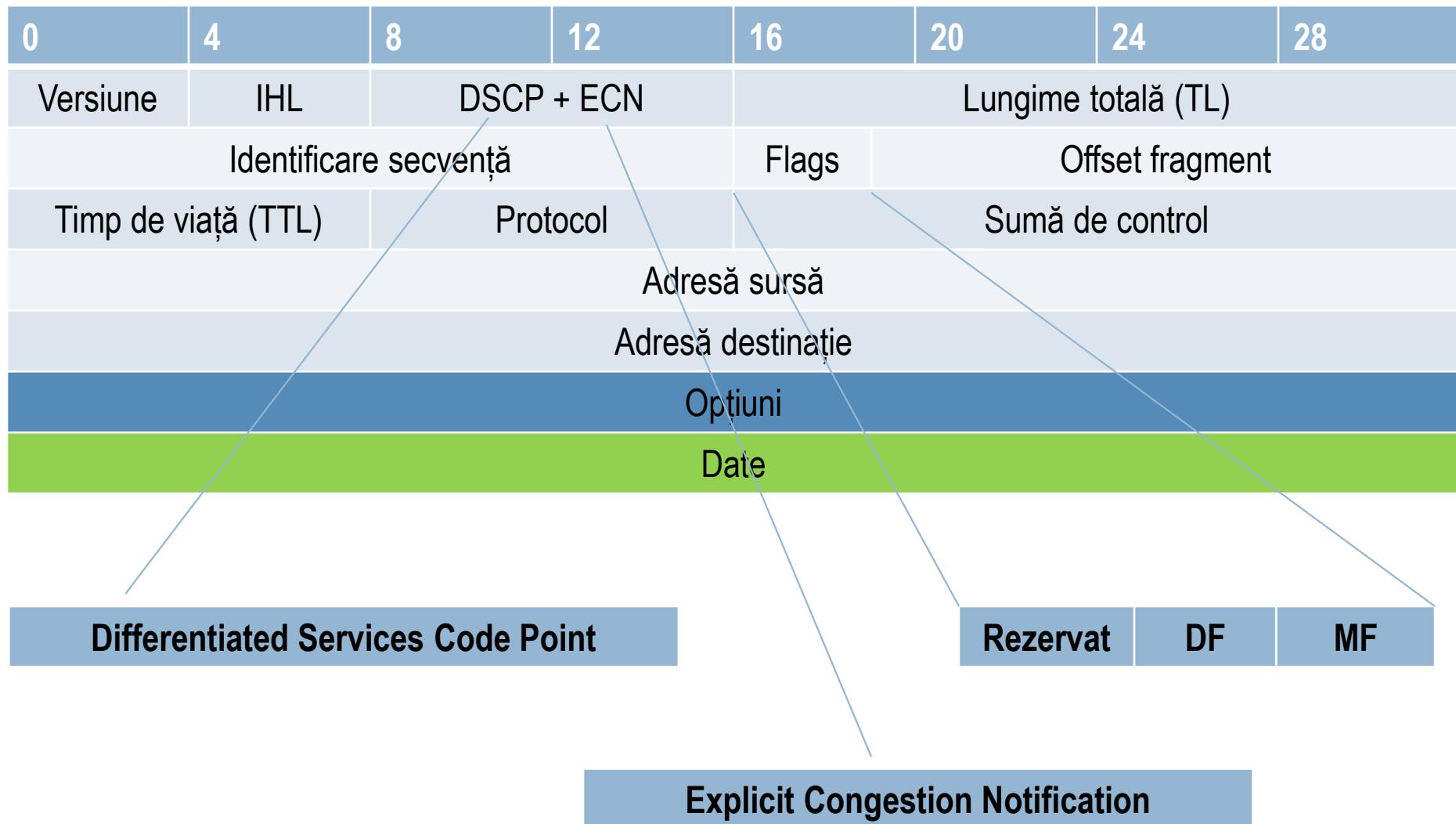
Protocolul IP

- IP este implementarea cea mai larg folosită pentru un plan de adresare ierarhizată a nivelului 3.
- Caracteristici:
 - **Neorientat pe conexiune;**
 - **Nesigur;**
 - **Transmisie best-effort.**

Integritatea datelor se asigură de către nivelele superioare.

- Versiuni:
 - IPv4
 - versiune utilizată cu preponderență în Internet;
 - adresare pe 32 biți.
 - IPv6
 - versiunea viitoare;
 - adresare pe 128 biți;
 - versiune optimizată și simplificată;
 - asigură prin construcție mobilitatea și securitatea datelor.

Antet protocol IPv4



Antet protocol IPv4

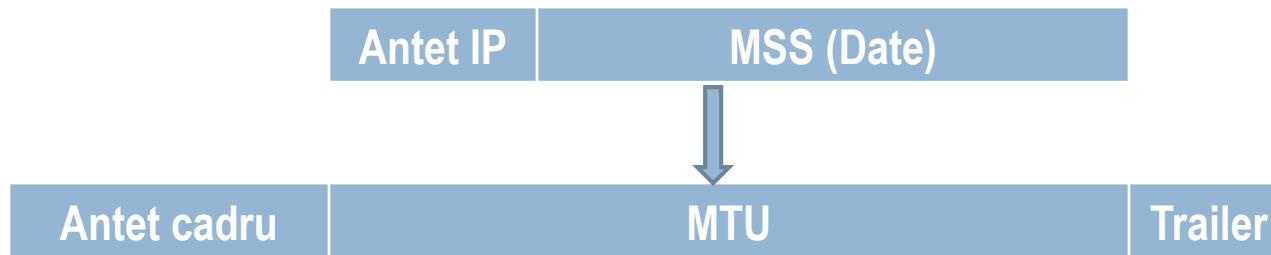
- **versiune** – versiunea protocolului IP.
- **lungime antet (IHL)** – lungimea antetului;
 - poate avea între 20 și 60 de octeți (reprezentare: multiplii de 4 octeți);
 - în mare majoritate a cazurilor antul IP va fi doar de 20 octeți.
- **DSCP** (Differentiated Services Code Point) – folosit pentru implementarea QoS.
- **ECN** (Explicit Congestion Notification) – utilizat pentru notificarea apariției unei congestii.
- **TTL** (Time To Live) – folosit pentru diminuarea efectului buclelor;
 - fiecare ruter va decrementa valoarea acestui câmp;
 - un pachet cu TTL 1 nu va părăsi rețeua locală;
 - este singurul câmp din antetul de nivel 3 ce este modificat la trecerea printr-un ruter.
- **CRC** – suma de control a antetului;
 - va fi recalculată de fiecare ruter, doar datorită operației de decrementare a TTL.

Antet protocol IPv4

- **protocol**
 - indică tipul protocolului incorporat.

1 – ICMP pentru IPv4	4 – IPv4 în IPv4
2 – IGMP pentru IPv4	6 – TCP
17 – UDP	41 – IPv6 în IPv4
58 – ICMP pentru IPv6	59 – nu mai există alt antet
- **lungime pachet**
 - definește lungimea pachetului;
 - limitează la maxim 64 Kocteți dimensiunea unei datagrame.
- **identificator secvență**
 - identifică datagrama.
- **fanoane (flags) [3]**
 - bitul 49 – rezervat (are valoare 0);
 - bitul 50 – DF (“do not fragment”);
 - bitul 51 – MF (“more fragments”).
- **offset [13]**
 - definește poziția fragmentului în cadrul datagramei IP.

Dimensiunea pachetelor



- MTU (Maximum Transfer Unit) reprezinta dimensiunea maximă a datelor înainte de încapsularea de nivel 2;
- MSS (Maximum Segment Size) reprezinta dimensiunea maximă a datelor înainte de încapsularea de nivel 4;
- Tehnologia dominantă de transfer este TCP/IP/Ethernet;
 - TCP nu limitează dimensiunea datelor dintr-un segment;
 - IP definește dimensiunea maximă a datelor după encapsulare 64 Kocteți;
 - Ethernet definește dimensiunea maximă a datelor înainte de encapsulare 1500 octeți;
- Fragmentarea este o operațiune costisitoare;
- În implementările actuale se încearcă evitarea dublei fragmentării prin stabilirea dimensiunii maxime a datelor de la nivelul 4;
- $MTU = MSS + 40$.

Adresare IPv4

Studiată la laborator !

Echipamente de tip ruter

- Pot fi folosite în LAN-uri sau WAN-uri:
 - Interconectează rețelele diferite;
 - Oferă interfețe pentru WAN.
- Operează la nivelul 3 al modelului OSI.
- Două funcții principale care definesc rutarea:
 - determinarea căii;
 - comutarea pachetelor.
- Mențin tabelele de rutare:
 - Static (rutele sunt definite de administratorul de rețea);
 - Dinamic (rute sunt adăugate cu ajutorul protocolelor de rutare).
- Realizează conexiunea dintre diferite standarde de nivel fizic, legatură de date sau rețea.

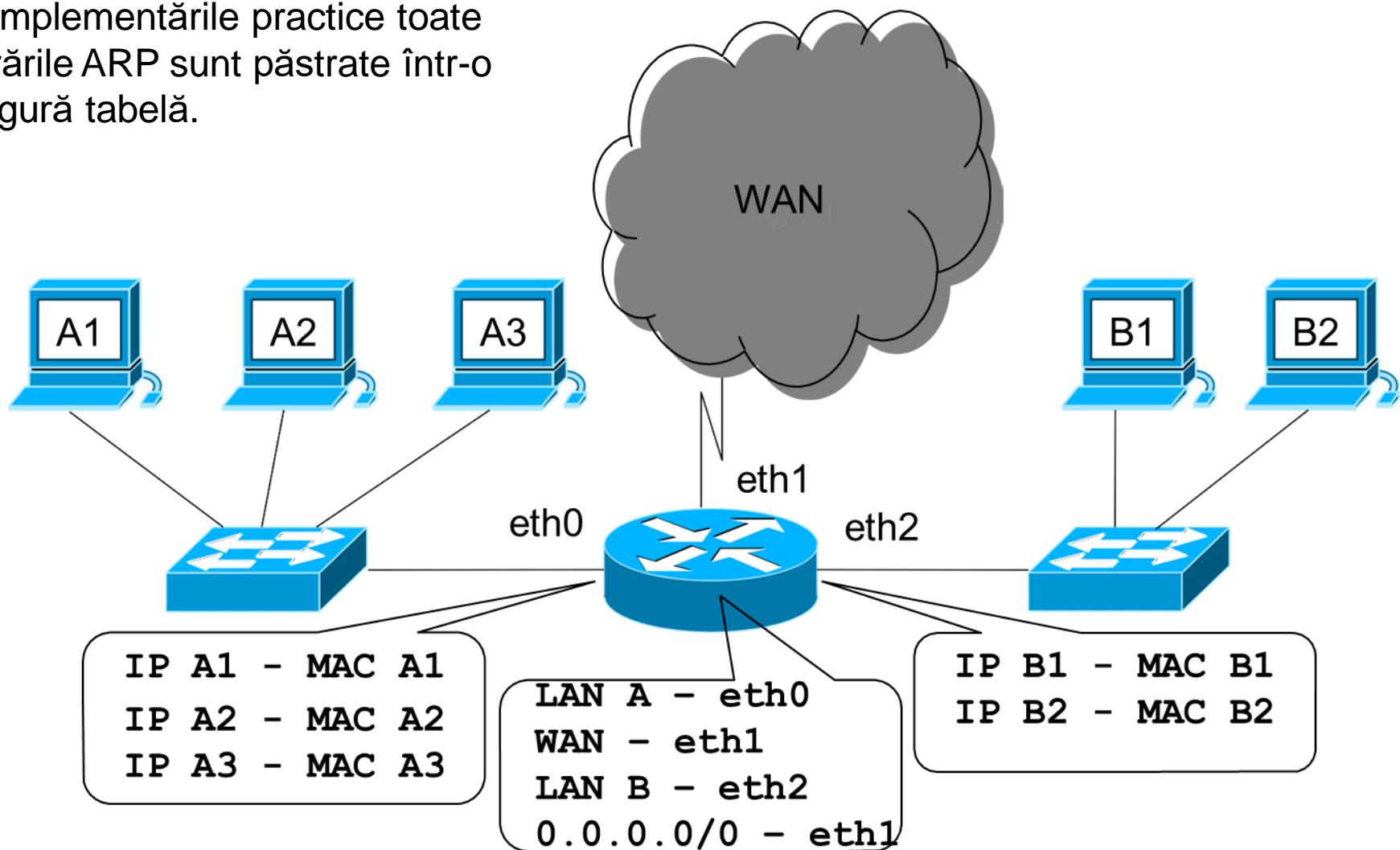
Comparație ruter - switch

	Ruter	Switch
nivel OSI	rețea	legătură de date
latență	mare	mica
adresă	IP	MAC
domenii de coliziune	limitează	limitează
domenii de broadcast	limitează	extinde
securitate	ridicată	scăzută

- Echipamentele de nivel 2+ mențin o **tabelă ARP** pentru comunicația la nivelul legătură de date. Tabela ARP este folositoare doar pentru domeniul de broadcast la care este conectată interfața.
- Echipamentele care implementează și nivelul 3 al modelului OSI mențin o **tabelă de rutare** pentru determinarea căii unui pachet atunci când destinația este în afara domeniilor de broadcast direct conectate.

Tabele ruter

În implementările practice toate intrările ARP sunt păstrate într-o singură tabelă.



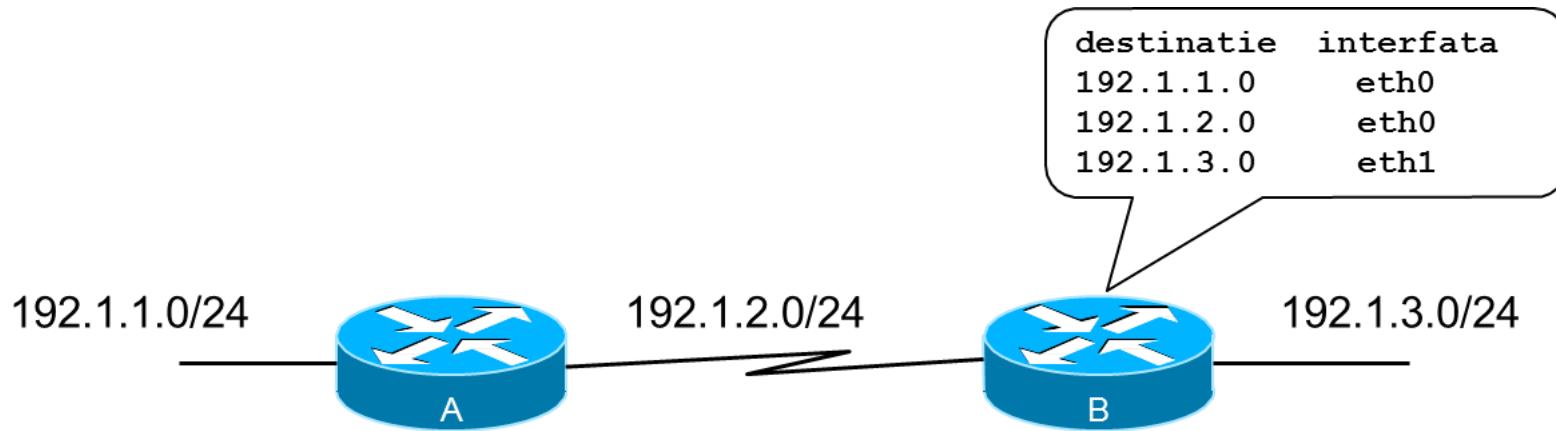
- O **rută** este o regulă ce cuprinde o parte de identificare și una de acțiune. Partea de identificare este compusă din două elemente: adresa rețelei destinație și masca acesteia, în vreme ce partea de acțiune poate fi exprimată prin ambele sau doar unul dintre următoarele elemente: adresa următorului nod (denumită next hop address) și interfața de ieșire din echipament.
- O **tabelă de rutare** este o listă de rute cu acces secvențial.

Tabele de rutare

- Pentru ca ruterele să poată comuta pachete, ele trebuie să învețe unde se află celelalte rețele, această informație fiind organizată sub forma unei tabele de rutare;
- Prima sursă de informații pentru tabela de rutare o reprezintă configurația propriilor interfețe de rețea, sistemul de operare generând pe baza ei **rutele rețelelor direct conectate**;
- Construirea unei tabele de rutare se poate face în două moduri:
 - **Static**: adăugarea rutelor se face manual, de către administratorul rețelei. Avantaje: securitate ridicată și overhead redus în rețea;
 - **Dinamic**: adăugarea rutelor se face cu ajutorul protocolelor de rutare (pe baza informațiilor primite de la alte rutere).
- Într-o tabelă de rutare putem avea simultan:
 - rute direct conectate;
 - rute dinamice;
 - rute statice.

Procesul de rutare

- Acest proces este alcătuit din două mecanisme:
 - **Determinarea căii optime**: este folosita tabela de rutare;
 - **Comutarea pachetelor** (forwarding): transmisia unui pachet de pe o interfață de rețea pe alta.
- Ruterele creeaza tabele de rutare.



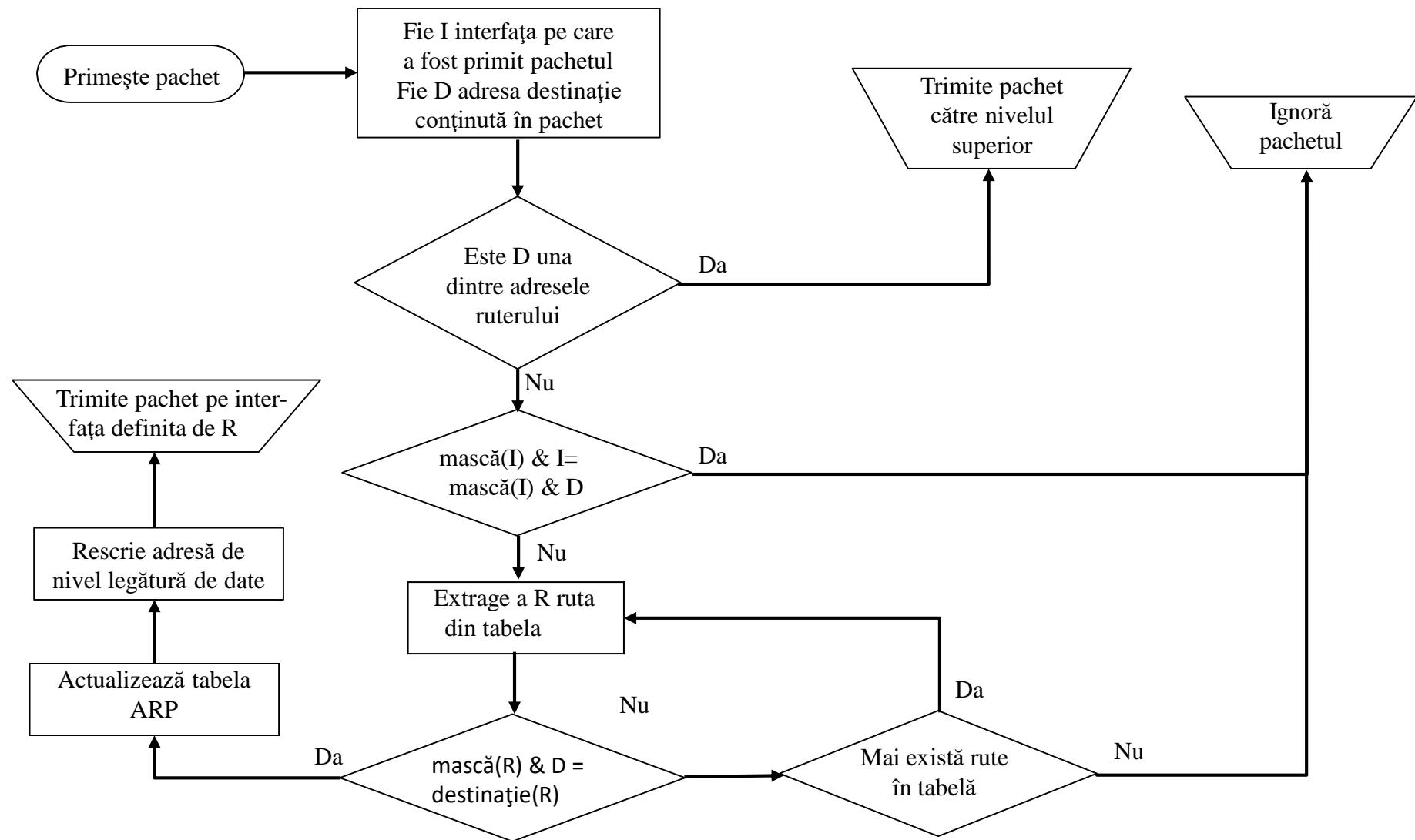
Rute statice

- Nu este necesara configurarea de rute statice pentru retele direct conectate;
- In cazul rutelor statice via legaturi punct la punct se indica specificarea numai a interfetei de iesire, deoarece adresa urmatorului hop nu este folosita in acest caz;
- In cazul rutelor statice via retele multiacces (Ethernet) este important sa se precizeze adresa urmatorului hop, doar interfata de iesire nefiind suficient
 - acest inconvenient poate fi compensant prin rularea Proxy ARP, la nivelul urmatorului hop.

Rute implicate

- Toate pachetele care nu au destinația verificată de alte rute în tabela de rutare vor fi trimise pe această rută;
- O ruta implicită este caracterizată de prefixul /0 (orice destinație se verifică pe această rută);
- Mai poartă denumirea de “quad zero route”;
- O astfel de rută poate fi definită ca și o rută statică, sau poate fi generată de un protocol de rutare.

Funcționare ruter



Construirea tabelei de rutare

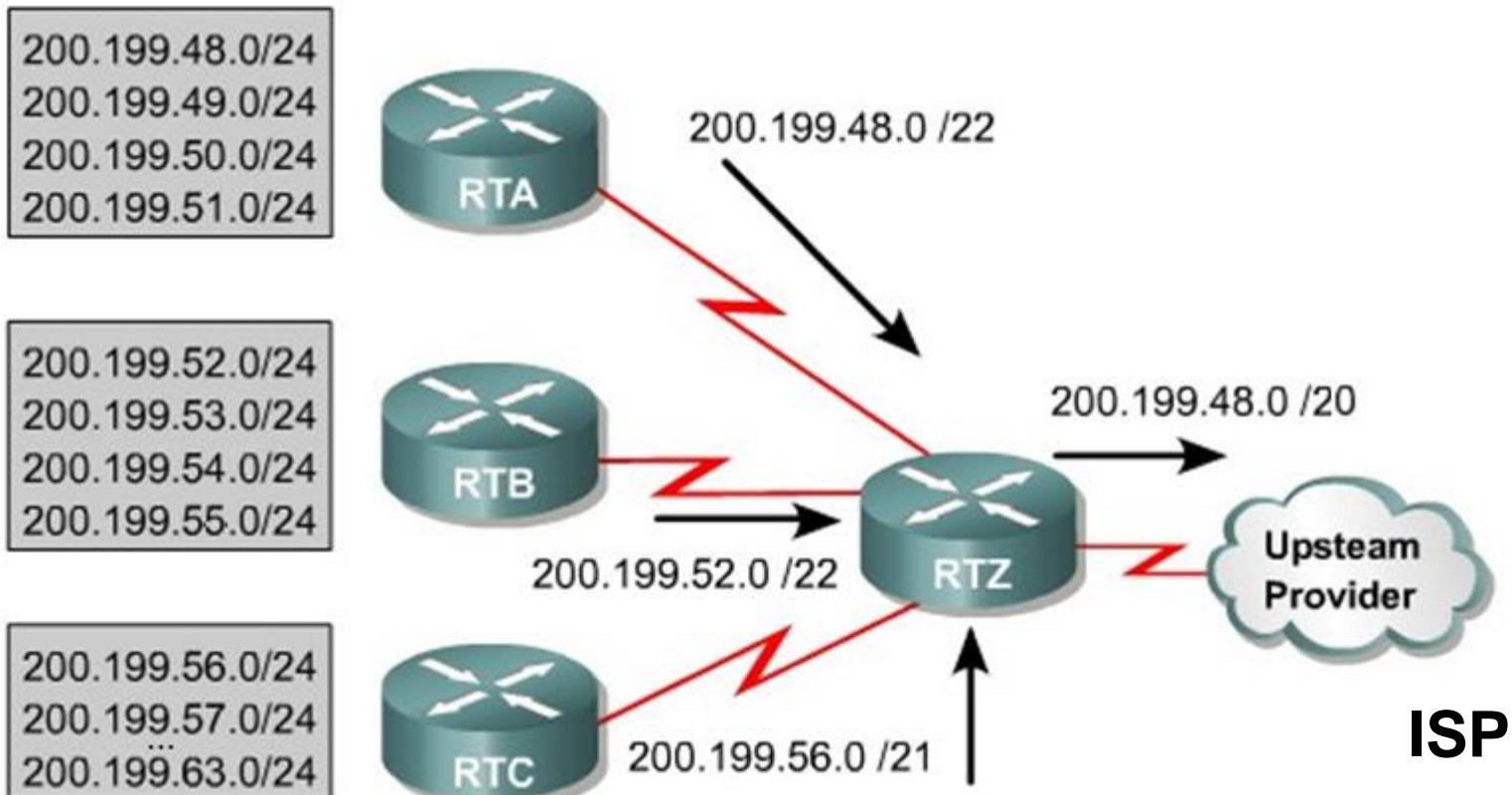
- Rutele sunt ordonate la introducerea lor în tabela de rutare în funcție de lungimea măștii;
- Pentru măști de lungime egală rutele sunt adăugate în tabela de rutare în ordinea cunoașterii lor;
- Parcurgerea tabelei de rutare se va face secvential.

- procesul de rutare poate să fie folosit și în scopul filtrării pachetelor în funcție de adresa destinație;
- o rută ce are precizată drept interfață de ieșire interfața logică de null;
- În Unix aceasta este interfața /dev/null.

Ex: 10.10.4.0/24 /dev/null va filtra toate pachetele cu destinația 10.10.4.0/24.

- Rutarea dinamica se bazeaza pe folosirea unui protocol de rutare;
- Există două clase de protocoale de rutare:
 - Distance Vector;
 - Link State.

Agregarea rutelor



- Sumarizarea rutelor reduce dimensiunea tabelei de rutare prin agregarea mai multor subnet-uri într-un singur “supernet”;

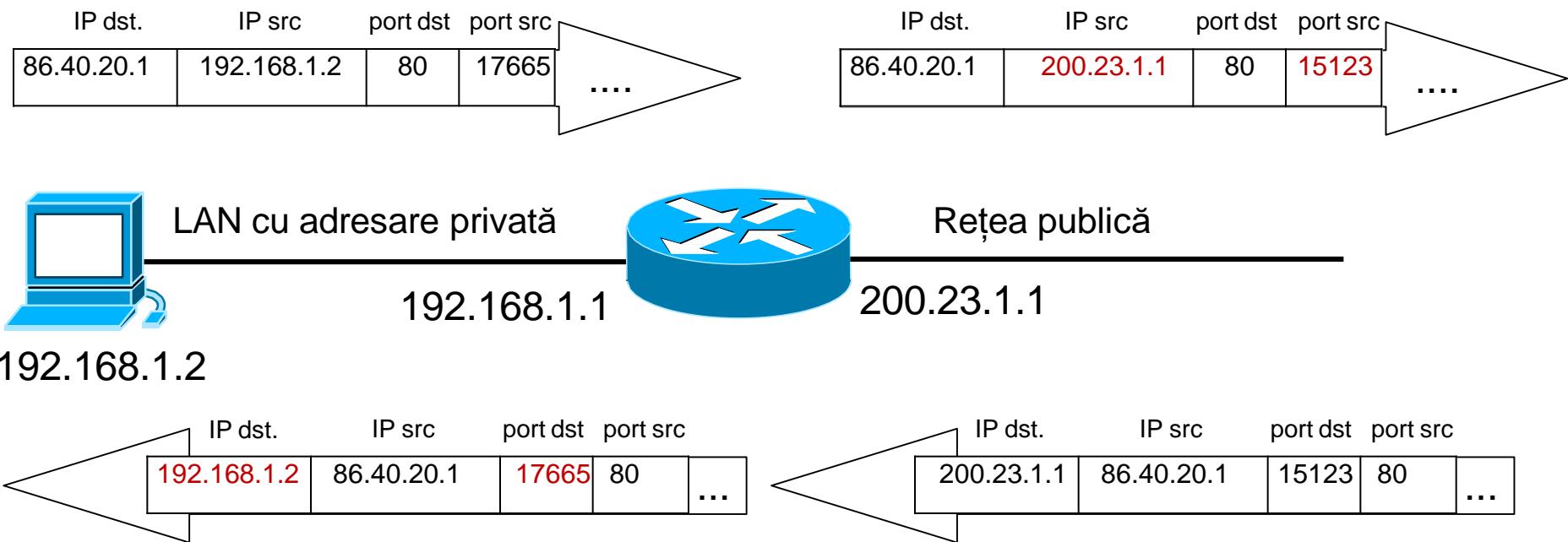
Adrese IP private

Clasa	Intervalul de adrese	Prefix CIDR
A	10.0.0.0 - 10.255.255.255	10.0.0.0/8
B	172.16.0.0 - 172.31.255.255	172.16.0.0/12
C	192.168.0.0 - 192.168.255.255	192.168.0.0/16

- Pot fi folosite de oricine, fără restricții;
- Un astfel de spațiu de adrese nu trebuie rutat în Internet.

- **NAT** (Network Address Translation)
 - atribuie o adresa privată unei adrese publice.
- **PAT** (Port Address Translation)
 - asociază mai multe adrese private aceeași adrese publice, folosind numerele de porturi pentru a diferenția între surse (stațiile din rețeua locală);
 - translatare de adrese cu supraîncărcare;
 - denumită și overloading, port forwarding, masquerading.

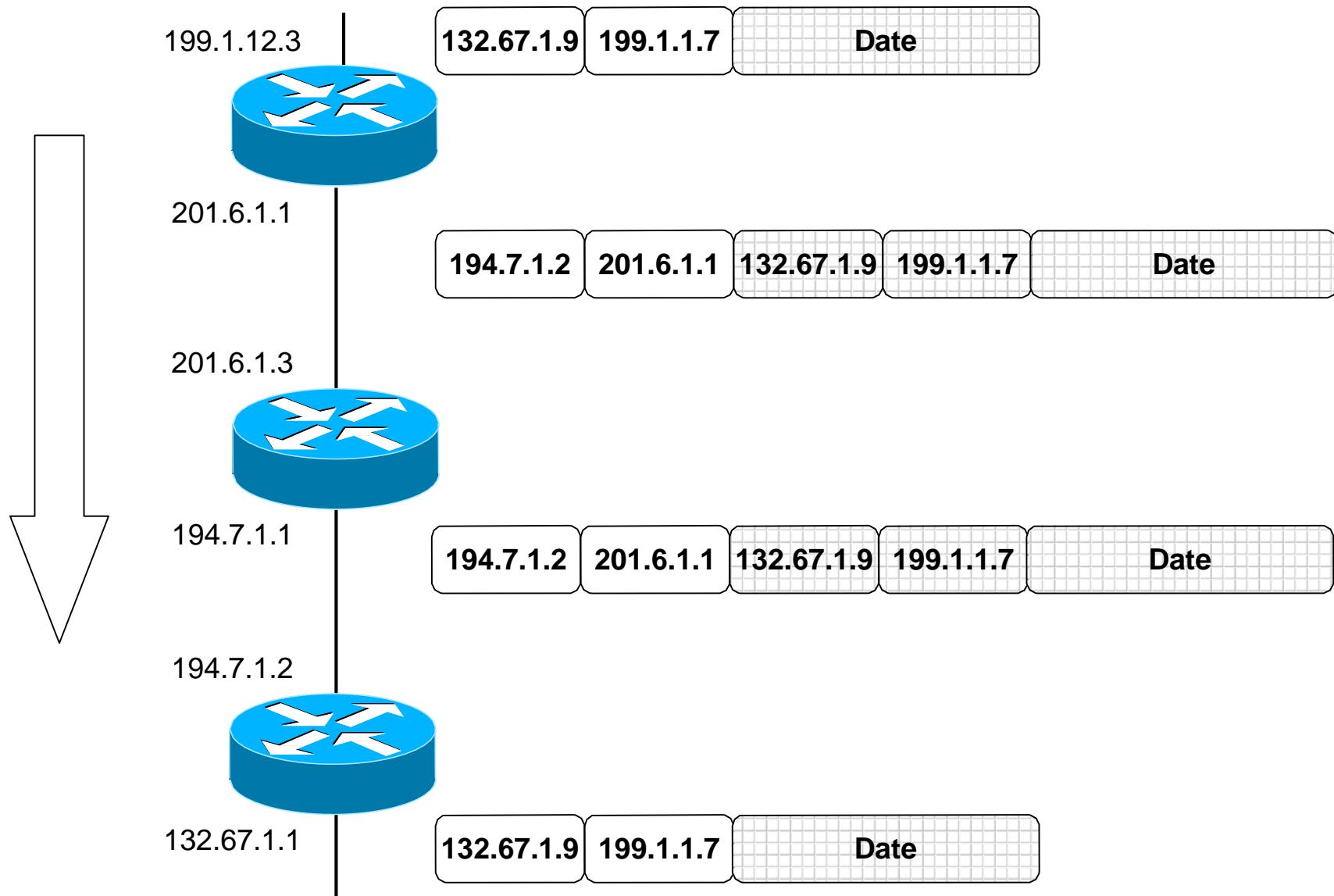
Exemplu PAT



Tunelarea IP

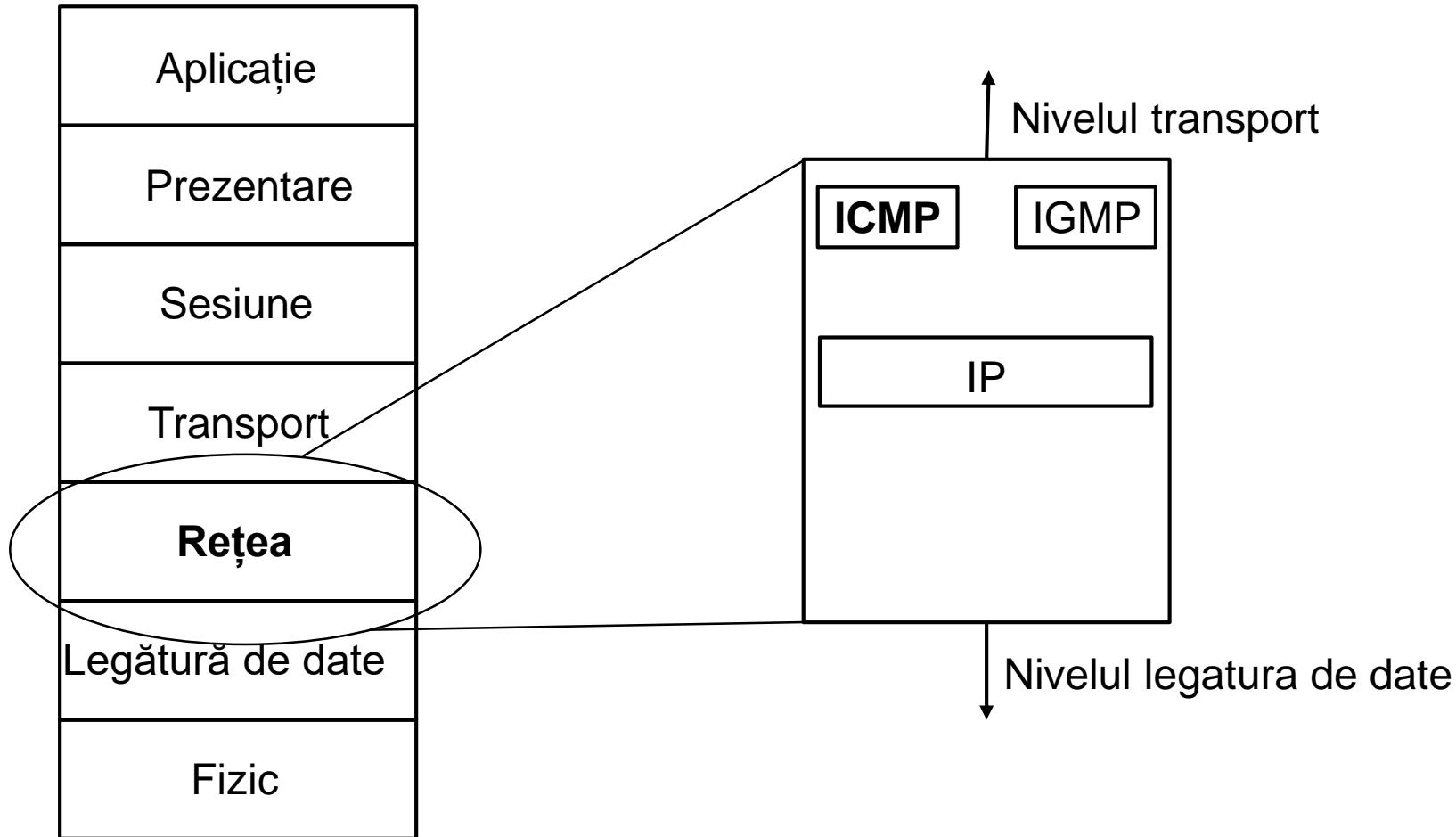
- reprezintă operația prin care pachetele IP sunt încapsulate încă o dată;
- principalul scop al tunelării este transportarea informațiilor din antetul IP original sub formă de date;
- antetul inițial al pachetului este păstrat nealterat, atașându-se un nou antet ce va avea ca adresă sursă adresa capătului local al tunelului, iar ca adresă destinație adresa celuilalt capăt al tunelului.

Tunelarea IP



ICMP – Internet Control Message Protocol

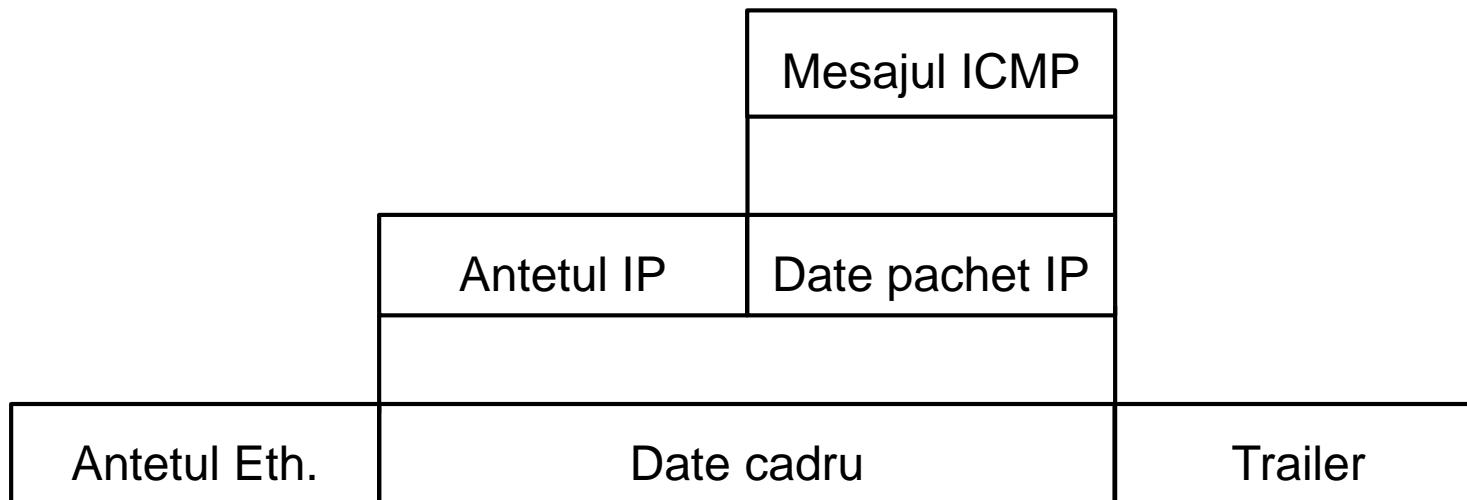
Modelul OSI



Introducere

- IP este un protocol de tip “best effort” astfel ca datele pot sa nu ajunga la destinatie: eroare hardware, configurare incorecta, congestie, rutare incorecta.
 - IP nu detine un mecanism pentru raportarea si corectia erorilor.
 - IP nu permite interogarea statiilor in vederea managementului lor.
- ICMP este proiectat pentru a rezolva aceste neajunsuri.
 - Mesajele ICMP se impart in doua mari categorii:
 - Mesaje pentru raportarea de erori.
 - Mesaje de interogare.
 - Mesajele pentru raportarea erorilor transmit posibile problemele aparute in procesarea unui pachet IP la nivelul unui nod.
 - Mesajele de interogare (transmise in pereche) ajuta la obtinerea unor informatii specifice despre un ruter sau o statie din retea.

Incapsulare mesaj ICMP



Structura pachet ICMP

Tip (8 biți)	Cod (8 biți)	Suma de control (16 biți)
Parametrii		
Date		

Structura pachet ICMP

Type	Nume	Type	Nume
---	-----	---	-----
0	Echo Reply	17	Address Mask Request
1	Unassigned	18	Address Mask Reply
2	Unassigned	19	Reserved (for Security)
3	Destination Unreachable	20-29	Reserved (for Robustness Experiment)
4	Source Quench	30	Traceroute
5	Redirect	31	Datagram Conversion Error
6	Alternate Host Address	32	Mobile Host Redirect
7	Unassigned	33	IPv6 Where-Are-You
8	Echo	34	IPv6 I-Am-Here
9	Router Advertisement	35	Mobile Registration Request
10	Router Solicitation	36	Mobile Registration Reply
11	Time Exceeded	37	Domain Name Request
12	Parameter Problem	38	Domain Name Reply
13	Timestamp	39	SKIP
14	Timestamp Reply	40	Photuris
15	Information Request	41-255	Reserved
16	Information Reply		

Structura pachet ICMP

Type 3: Destination Unreachable

Codes

- 0 Net Unreachable
- 1 Host Unreachable
- 2 Protocol Unreachable
- 3 Port Unreachable
- 4 Fragmentation Needed and Don't Fragment was Set
- 5 Source Route Failed
- 6 Destination Network Unknown
- 7 Destination Host Unknown
- 8 Source Host Isolated
- 9 Communication with Destination Network is Administratively Prohibited
- 10 Communication with Destination Host is Administratively Prohibited
- 11 Destination Network Unreachable for Type of Service
- 12 Destination Host Unreachable for Type of Service
- 13 Communication Administratively Prohibited
- 14 Host Precedence Violation
- 15 Precedence cutoff in effect

Destination unreachable - Tip 3

0	8	16	24
Tip	Cod	Sumă de control	
Neutilizat			
Antet IP + primii 64 de biți de date din pachetul original			

Coduri:

0 = network unreachable

- mesaj ICMP generat de un router care nu identifică o rută către destinație

1 = host unreachable

- ultimul router nu poate contacta destinația

2 = protocol unreachable

- protocol necunoscut

3 = port unreachable

- portul nu este asociat niciunui socket

4 = fragmentation needed and don't fragment was set

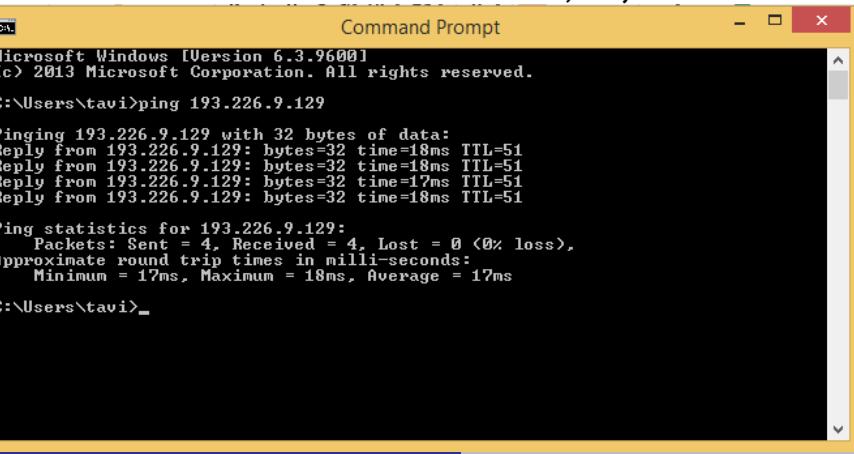
- pachetul necesită fragmentare însă aceasta nu este permisă

ICMP Echo (Request) and Echo Reply - Tip 8 și 0

0	8	16	24
Tip	Cod	Sumă de control	
	Identifier	Număr de secvență	
Date ...			

Cod 0

La receptia unui mesaj echo un nod inversează IP-ul sursă cu cel destinație, stabilește câmpul tip pe valoarea 0 și recalculează suma de control. Identifierul, secvența și datele sunt trimise înapoi nemodificate.



```

C:\> ping 193.226.9.129

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\>ping 193.226.9.129

Pinging 193.226.9.129 with 32 bytes of data:
Reply from 193.226.9.129: bytes=32 time=18ms TTL=51
Reply from 193.226.9.129: bytes=32 time=18ms TTL=51
Reply from 193.226.9.129: bytes=32 time=17ms TTL=51
Reply from 193.226.9.129: bytes=32 time=18ms TTL=51

Ping statistics for 193.226.9.129:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 17ms, Maximum = 18ms, Average = 17ms

C:\>
  
```

ICMP Time Exceeded - Tip 11

Format identic cu tipul 3

0	4	8	12	16	20	24	28												
Versiune	IHL	Tipul serviciului (TOS)			Lungime totală (TL)														
Identificare secvență			Flags	Offset fragment															
Timp de viață (TTL)			Protocol	Sumă de control															
Adresă sursă																			
Adresă destinație																			
Optiuni																			
Date																			

Coduri:

0 = timp de viață expirat în tranzit;

1 = timpul de viață expirat la reasamblarea mesajului.

Se transmite dacă: un router identifică un pachet cu valoare TTL = 0 sau un nod nu primește toate fragmentele unui pachet până la expirarea contorului de reasamblare.

ICMP Parameter problem - Tip 12

0	8	16	24
Tip	Cod	Sumă de control	
Pointer		Neutilizat	
Antet IP + primii 64 de biți de date din pachetul original			

Coduri:

0 = eroare oarecare de parametrii;

1 = lipsă opțiune necesară.

Pointer-ul identifică octetul unde apare eroarea.

Se transmite unui nod atunci când la procesarea pachetului său se identifică o eroare, iar pachetul este distrus.

ICMP Redirect - Tip 5

Format identic cu tipul 3

Coduri:

- 0 = redirectare pachete destinate unei rețele;
- 1 = redirectare pachete destinate unui nod;
- 2 = redirectare pachete destinate unui tip de serviciu și unei rețele;
- 3 = redirectare pachete destinate unui tip de serviciu și unui nod.

Este transmis de către un router atunci când determină că un pachet trebuie retransmis unui alt router care poate fi accesat direct de către nodul sursă.

ICMP Source Quench - Tip 4

Format identic cu tipul 3

Cod: 0

Este transmis de un router intermediar sau de către destinație atunci când acestea nu pot ține pasul cu rata de transmisie a datelor de către sursă.

Se transmite câte un mesaj ICMP pentru fiecare pachet distrus.

La primirea unui astfel de mesaj sursa diminuează rata de transfer.

ICMP Timestamp / Timestamp Reply - Tip 13 și 14

0	8	16	24
Tip	Cod	Sumă de control	
Identifier		Număr de secvență	
Timestamp sursă			
Timestamp destinație - la recepție			
Timestamp destinație - la transmisie			

Format - unix time.

Utilizat pentru identificarea latenței dintre receptor și transmițător.

ICMP Information Request / Information Reply - Tip 15 și 16

0	8	16	24
Tip	Cod	Sumă de control	
Identifier		Număr de secvență	

Cod: 0

Mesajul se transmite într-un pachet cu adrese IP sursă și destinație 0.

Utilizat de un nod pentru a identifica rețeaua din care face parte.

ICMP Addr. Mask Req. / Addr. Mask Reply - Tip 17 și 18

0	8	16	24
Tip	Cod	Sumă de control	
Identifier		Număr de secvență	
Mască			

Utilizat de un nod pentru a identifica masca rețelei.

ICMP Router Solicitation și Advertisement - Tip 10 și 9

La pornire un nod poate identifica routerele disponibile în rețea utilizând ICMP Router Solicitation utilizând multicast sau broadcast. Routerele raspund cu un pachet de tip ICMP Advertisement.

Routerele pot transmite automat în mod periodic mesaje de "ICMP Router Advertisement".

Adresarea IPv4

Adresa IP este un identificator logic al unui nod intr-o retea. In cazul variantei 4 de protocol, adresa IP are o rezolutie pe 32 de biti si se reprezinta, pentru o utilizare mai usoara, sub forma a 4 numere zecimale, corespunzatoare fiecarui octet, separate prin puncte. Exemplu: 192.168.1.100

Spatiul de adresare IPv4 contine 2^{32} adrese, cu valori cuprinse intre 0.0.0.0-255.255.255.255. Acest bloc de adrese este impartit in retele logice, dupa anumite reguli. Orice adresa IP este formata din doua parti, una care identifica radicalul retelei din care face parte adresa (**network id**), iar cealalta care identifica nodul de retea caruia ii este atribuita adresa (**host id**). Delimitarea celor doua parti se face cu ajutorul unei **masti de retea (network mask)**.

Masca este un alt identificator pe 32 de biti, asociat retelei, avand o structura specifica, ce respecta urmatoarele reguli:

1. In masca nu pot exista biti de unu intercalati cu biti de zero.
2. Cei mai semnificativi biti din masca au valoarea unu si definesc rezolutia partii de retea dintr-o adresa IP, identificand practic radicalul retelei (partea fixa din adresa).
3. Cei mai putin semnificativi biti au valoarea zero si definesc rezolutia partii de host, adica acea parte variabila din retea, ce defineste fiecare adresa IP ce face parte din retea.

Initial, blocul de adrese IP a fost impartit, prin conventie, in 5 clase, in functie de primii 4 biti cei mai semnificativi din adresa, astfel:

Clasa	Primii 4 biti	Interval de adrese
A	0xxx	0.0.0.0 - 127.255.255.255
B	10xx	128.0.0.0 - 191.255.255.255
C	110x	192.0.0.0 - 223.255.255.255
D	1110	224.0.0.0 - 239.255.255.255
E	1111	240.0.0.0 - 255.255.255.255

De asemenea, s-a hotarat ca primele trei clase de adrese sa fie utilizate pentru transmisii de date unicast, a patra pentru transmisii de date multicast si ultima a fost rezervata pentru o utilizare ulterioara.

Clasele A,B si C au fost impartite in retele, stabilindu-se cate o masca specifica fiecarei clase in asa fel incat impartirea sa se faca foarte simplu. Astfel, retelelor de clasa A li s-a asociat masca 255.0.0.0, celor de clasa B masca 255.255.0.0, iar celor de clasa C masca 255.255.255.0.

În continuare, tinand cont de masile specifice, partea de rețea (N) și partea de host (n) pentru o anumită adresă din cadrul unei clase de IP-uri este:

Class A – NNNNNNNN.nnnnnnnn.nnnnnnnn.nnnnnnnn

Class B – NNNNNNNN.NNNNNNNN.nnnnnnnn.nnnnnnnn

Clasa C – NNNNNNNN.NNNNNNNN.NNNNNNNN.nnnnnnnn

Astfel, orice retea de clasa A, va avea primul octet fixat (radicalul retelei), avand în componenta sa 2^{24} adrese IP (rezolutia partii de host fiind pe 24 de biti, deoarece avem 24 de biti de 0 în masca). O retea de clasa B va avea primii doi octeti fixati, continand 2^{16} adrese IP, iar o retea de clasa C va avea primii trei octeti fixati, continand 2^8 adrese IP.

Spre exemplu, reteaua de clasa C 192.168.100.0, avand masca 255.255.255.0, va avea radicalul format din primii trei octeti: 192.168.100, întinzându-se de la adresa 192.168.100.0 pana la adresa 192.168.100.255. Prima adresa IP din retea (192.168.100.0) se numeste adresa retelei, iar ultima adresa din retea (192.168.100.255) se numeste adresa de broadcast (de difuzare) a retelei, adica adresa utilizata de o sursa atunci cand doreste sa transmita un mesaj tuturor nodurilor din retea. In orice retea, cele doua adrese sunt rezervate, neputand fi alocate nodurilor.

Pornind de la adresa IP a unui nod de retea si de la masca retelei, se poate identifica adresa retelei, respectiv adresa de broadcast a retelei cu ajutorul urmatoarelor operatii binare:

ADRESA IP **SI** NETMASK = ADRESA RETEA

ADRESA RETEA **XOR** NETMASK = ADRESA BROADCAST

De asemenea, numarul de adrese IP alocabile nodurilor, sau numarul de hosturi poate fi calculat astfel:

$Nr_H = 2^n - 2$, unde n reprezinta numarul de zerouri din masca.

Cele doua adrese IP care se scad sunt cele rezervate pentru adresa retelei, respectiv pentru adresa de broadcast.

Exemplu:

Se consideră adresa IP 192.168.100.125. Se cere:

- a) Adresa rețelei din care face parte adresa IP;
- b) Adresa de broadcast a rețelei din care face parte adresa IP;
- c) Numarul de hosturi din rețea.

Adresa IP fiind de clasa C (primii 4 biti sunt 1100), masca rețelei este 255.255.255.0.

Transformând în binar și efectuând calculele vom obține:

IP : 11000000.10101000.01100100.01111101	SI
NM : 11111111.11111111.11111111.00000000	
NA : 11000000.10101000.01100100.00000000	XOR
BA : 11000000.10101000.01100100.11111111	

Revenind înapoi în zecimal, obținem adresa rețelei (NA) 192.168.100.0, respectiv adresa de broadcast a rețelei (BA) 192.168.100.255.

Numarul de hosturi din rețea: $Nr_H = 2^8 - 2 = 254$

Impartirea in retele de trei dimensiuni, cu cate 2^{24} , 2^{16} , 2^8 adrese IP s-a dovedit, o data cu cresterea numarului de noduri conectate in internet, a fi una ineficienta. Astfel, spre exemplu, daca se urmarea adresarea unei retele fizice, ce continea 1000 de noduri, folosind o singura retea logica, era nevoie de o retea de clasa B compusa din 2^{16} adrese IP, diferența irosindu-se.

In vederea optimizarii operatiei de adresare, pentru a evita irosirea adreselor nefolosite, precum si pentru a imbunatatii performantele retelei, scazand numarul de hosturi dintr-un domeniu de broadcast, a fost introdus conceptul de impartire in subretele (subnetting).

Urmărindu-se scaderea numarului de hosturi dintr-o retea logica, solutia a fost scaderea rezolutiei partii de host, prin “imprumutul” de biti din masca de la partea de host, la partea de retea. Cum regulile de constructie a mastii trebuie pastrate, pentru a nu permite intercalarea bitilor de unu cu cei de zero, “imprumutul” bitilor se face de la stanga la dreapta.

Exemplu:

Se considera reteaua 192.168.100.0, avand masca 255.255.255.0. Se cere impartirea retelei in subretele, necesare adresarii unor laboratoare de cate 18 statii fiecare, urmarindu-se irosirea unui numar minim de adrese IP.

Numarul de hosturi dintr-o retea este dat de formula $Nr_H = 2^n - 2$, unde n reprezinta numarul de biti de zero din masca. Cum urmarim ca fiecare subretea sa aiba cel putin 18 hosturi si, in acelasi timp, ne dorim irosirea unui numar minim de adrese IP, vom identifica valoarea minima a lui n astfel incat $2^n - 2 \geq 18$. Rezulta n = 5, iar masca subretelelor, avand in componenta 27 de biti de 1 si 5 biti de zero, este 255.255.255.224. Practic, dintr-o retea ce contine 2⁸ adrese IP am obtinut 8 subretele de cate 2⁵ adrese IP ($2^8 = 2^3 \times 2^5$). Cele 8 subretele sunt:

- | | |
|-------------------------------------|--------------------------------------|
| 1. 192.168.100.0 – 192.168.100.31 | 5. 192.168.100.128 – 192.168.100.159 |
| 2. 192.168.100.32 – 192.168.100.63 | 6. 192.168.100.160 – 192.168.100.191 |
| 3. 192.168.100.64 – 192.168.100.95 | 7. 192.168.100.192 – 192.168.100.223 |
| 4. 192.168.100.96 – 192.168.100.127 | 8. 192.168.100.224 – 192.168.100.255 |

Exemplu:

Se pune problema identificarii subretelei, cu masca 255.255.255.224, din care face parte adresa IP 192.168.100.175.

Operatiile de calcul pentru aflarea adresei de subretea, respectiv a adresei de broadcast a subretelei, sunt identice cu cele de la retea, diferenta facand-o masca utilizata.

Transformand in binar obtinem si efectuand calculele vom obtine:

IP : 11000000.10101000.01100100.10101111	SI
NM : 11111111.11111111.11111111.11100000	
NA : 11000000.10101000.01100100.10100000	XOR
BA : 11000000.10101000.01100100.10111111	

Revenind inapoi in zecimal, adresa subretelei din care face parte IP-ul este 192.168.100.160, iar adresa de broadcast a acesteia este 192.168.100.191 (adica subreteleaua 6 din exemplul anterior).

Numarul de hosturi din retea se calculeaza similar $Nr_H = 2^n - 2 = 2^5 - 2 = 30$.

Numarul de subretele din retea se calculeaza cu formula $Nr_{SR} = 2^m - 2$, unde m reprezinta numarul de biti imprumutati din masca (transformati din 0 in 1). In cazul exemplului nostru $Nr_{SR} = 2^3 - 2 = 6$. Observam ca acest numar difera fata de numarul identificat de noi, adica 8. Aceasta diferența apare dintr-o limitare impusa la momentul la care s-a introdus conceptul de subnetting, rezervandu-se prima si ultima subretea obtinute in urma operatiei de impartire in subretele. Limitarea a fost introdusa deoarece adresa primei subretele este identica cu cea a retelei originale si adresa de broadcast a ultimei subretele este identica cu cea a retelei originale, iar unele echipamente de routare de la vremea respectiva nu puteau face diferența intre ele.

Exemplu:

Se considera adresa IP 129.180.231.220 avand masca 255.255.240.0. Se cere:

- a) Adresa retelei, respectiv adresa de broadcast a retelei din care face parte IP-ul
- b) Adresa subretelei, respective adresa de broadcast a subretelei din care face parte IP-ul
- c) Numarul de hosturi din retea, respectiv din subretea
- d) Numarul de subretele din retea

IP-ul fiind de clasa B, masca retelei este 255.255.0.0, masca din enunt fiind masca subretelei.

Transformand in binar si efectuand calculele, obtinem:

IP: 10000001.10110100.11100111.11011100	SI
NM: 11111111.11111111.00000000.00000000	
NA: 10000001.10110100.00000000.00000000	XOR
BA: 10000001.10110100.11111111.11111111	

Rezulta adresa retelei: 129.180.0.0, iar adresa de broadcast a retelei: 129.180.255.255

IP: 10000001.10110100.11100111.11011100	SI
NM: 11111111.11111111.11110000.00000000	
NA: 10000001.10110100.11100000.00000000	XOR
BA: 10000001.10110100.11101111.11111111	

Rezulta adresa subretelei: 129.180.224.0, iar adresa de broadcast a subretelei: 129.180.239.255

Numarul de hosturi din retea: $Nr_H = 2^{16}-2$, iar din subretea $Nr_H = 2^{12}-2$.

Numarul de subretele din retea $Nr_{SR} = 2^4-2$

Desi impartirea in subretele a optimizat procesul de adresare, totusi utilizarea claselor, cu masti predefinite mentinea anumite limitari. Spre exemplu, avand la dispozitie doua retele de clasa C consecutive 192.168.8.0, respectiv 192.168.9.0, fiecare avand in componenta cate 256 de adrese IP, nu putem adresa o retea fizica ce contine 500 de noduri, intr-o singura retea logica, deoarece nu avem cum sa unim cele doua retele de clasa C, restrictiile impuse de clase nepermitandu-ne acest lucru. Situatia ingreuna agregarea rutelor la nivelul tabelelor de rutare, ducand la o supradimensionare a acestora. Pentru a elmina si aceste restrictii si pentru a permite cresterea dimensiunii retelei peste limitarile stabilite de clase (supernetting), s-a hotarat renuntarea la clase si introducerea conceptului de Classless Inter-Domain Routing (CIDR) si o data cu el s-a introdus si notiunea de prefix ca si alternativa pentru reprezentarea mastii. Prefixul reprezinta numarul de biti de 1 dintr-o masca. Astfel masca 255.255.255.0 se poate reprezenta /24.

Revenind la exemplul anterior, al celor doua retele de clasa C: 192.168.8.0/24 si 192.168.9.0/24, CIDR ne permite unirea lor intr-o “super-retea” 192.168.8.0/23 (adica masca 255.255.254.0), prin imprumutarea, in sens invers, a unui bit de 1 din masca si transformarea lui intr-un bit de 0, crescand astfel rezolutia partii de host si creand astfel o retea de 512 adrese IP. Practic prin CIDR se renunta la impartirea blocului de adrese in clase de retele cu dimensiuni fixe, noua impartire facandu-se in functie de nevoie, stabilind corespunzator masca pentru fiecare retea in parte.

Odata cu introducerea CIDR, a aparut si mecanismul de subnetare dinamica (VLSM – Variable Length Subnet Mask) prin care o retea se poate imparti in subretele de dimensiuni diferite, avand masuri diferite.

Exemplu:

Avand la dispozitie reteaua 10.10.4.0/23, se cere impartirea ei in 3 subretele, astfel incat sa fie utilizat tot spatiul de adresare disponibil.

Imprumutand doi biti de 0 din masca si transformandu-i in biti de 1, vom obtine 4 subretele /25:

1. 10.10.4.0 - 10.10.4.127 /25
2. 10.10.4.128 - 10.10.4.255 /25
3. 10.10.5.0 - 10.10.5.127 /25
4. 10.10.5.128 - 10.10.5.255 /25

Prin unirea ultimelor doua rezulta o noua retea:

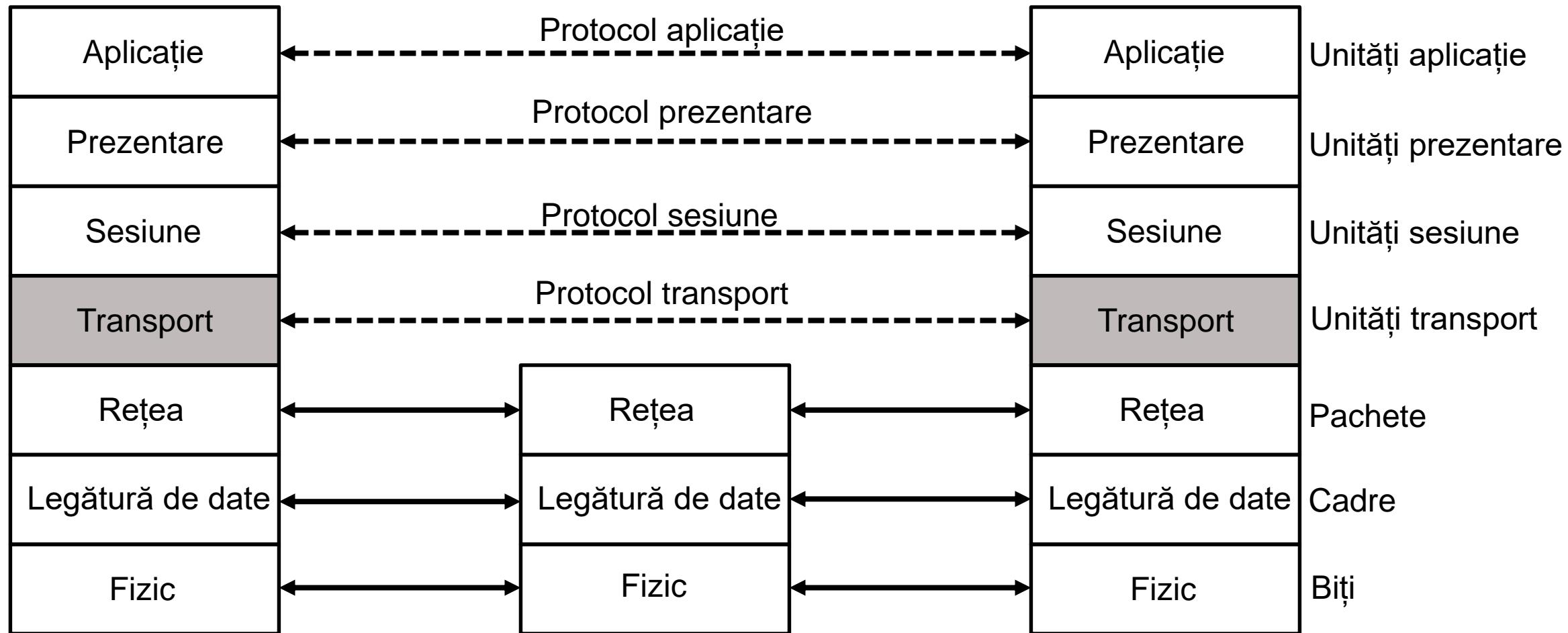
5. 10.10.5.0 – 10.10.5.255 /24

In final, am obtinut doua retele /25 (1 si 2) si o retea /24 (5), utilizand tot spatiul de adrese disponibil.

Odata cu introducerea CIDR s-a renuntat la rezervarea primei si ultimei subretele, numarul de subretele dintr-o retea fiind egal cu $Nr_{SR} = 2^m$, unde m reprezinta numarul de biti imprumutati din masca.

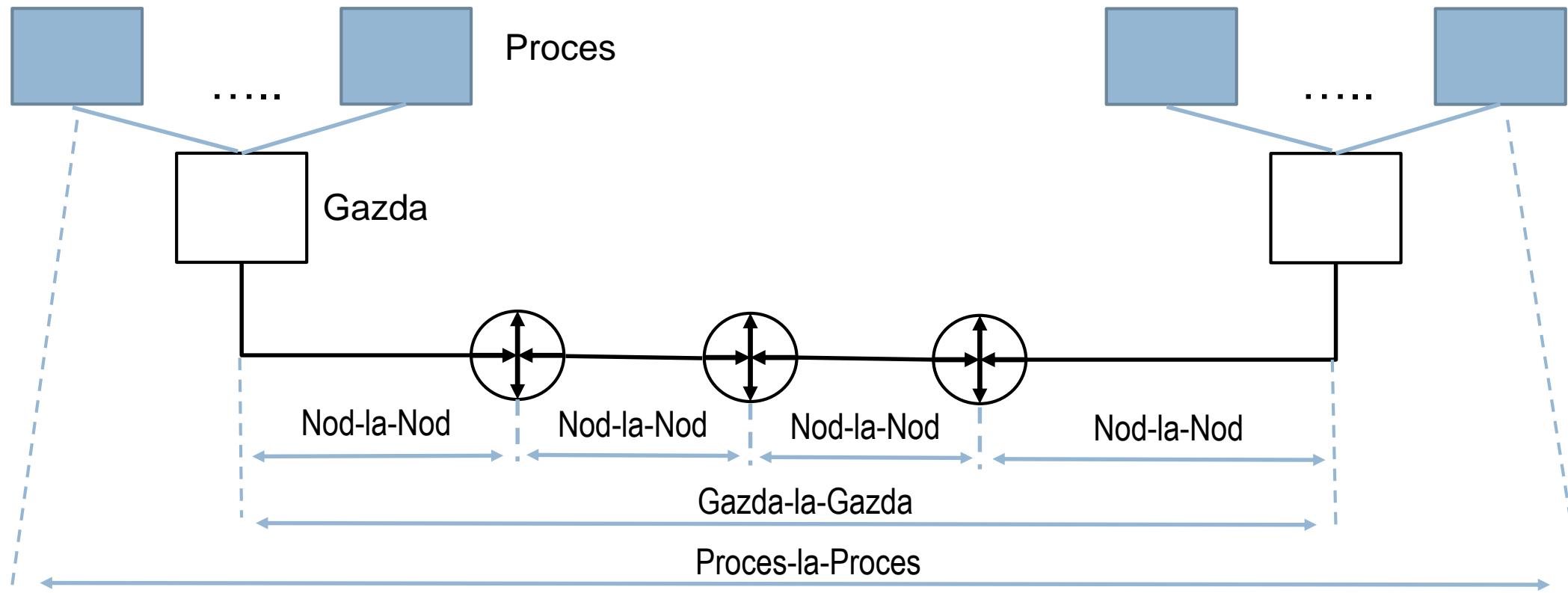
Nivelul transport

Modelul OSI

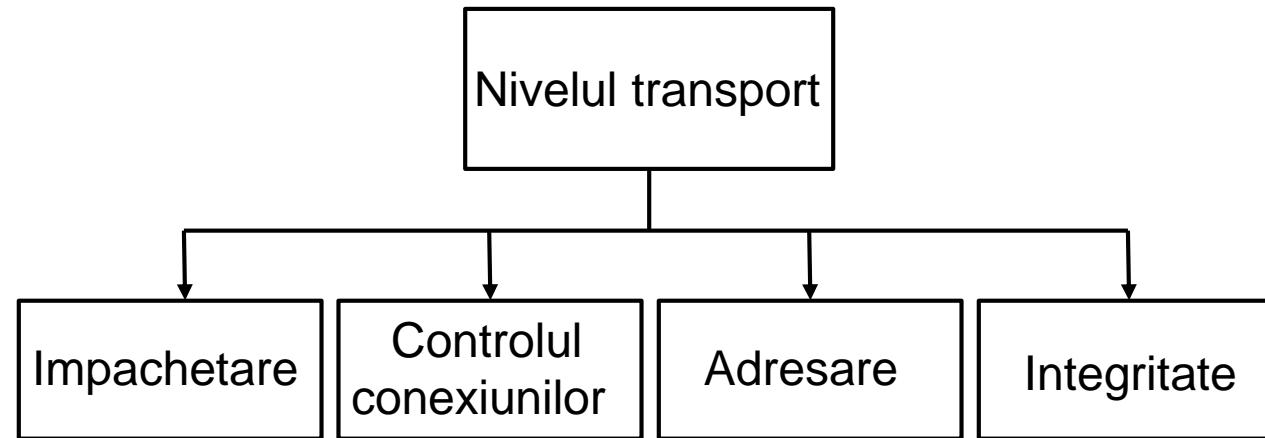


Nivelul transport

Nivelul transport este responsabil de transportul datelor intre procese, adica de transportul pachetului ca parte a unui mesaj, de la un proces la altul.



Ofera servicii nivelului sesiune si primeste servicii de la nivelul retea.



Responsabilitati:

- impărtirea datelor in **segmente / datagrame**;
- crearea de **conexiuni**;
- un nou mecanism de adresare (**porturi**);
- **controlul fluxului** (controlul congestiei);
- siguranta transmisiei (**reliability**).

Port = sistemul de adresare folosit de nivelul transport.

Existenta mai multor procese pe aceeasi statie necesita o **multiplexare prin porturi**.

16 biti → valori de la 0 la 65535.

Intervale de porturi (IANA):

- *Porturi rezervate (well-known)*: între 0 și 1023 (ex. SSH –22, FTP – 21, Telnet –23, HTTP –80);
- *Porturi înregistrate* între 1024 și 49151 (ex. Kazaa, RMI Registry, MySQL, etc.);
- *Porturi dinamice (efemere)*: de la 49152 la 65535.

Adresa unui socket reprezinta o pereche formata dintr-o adresa IP si un port.

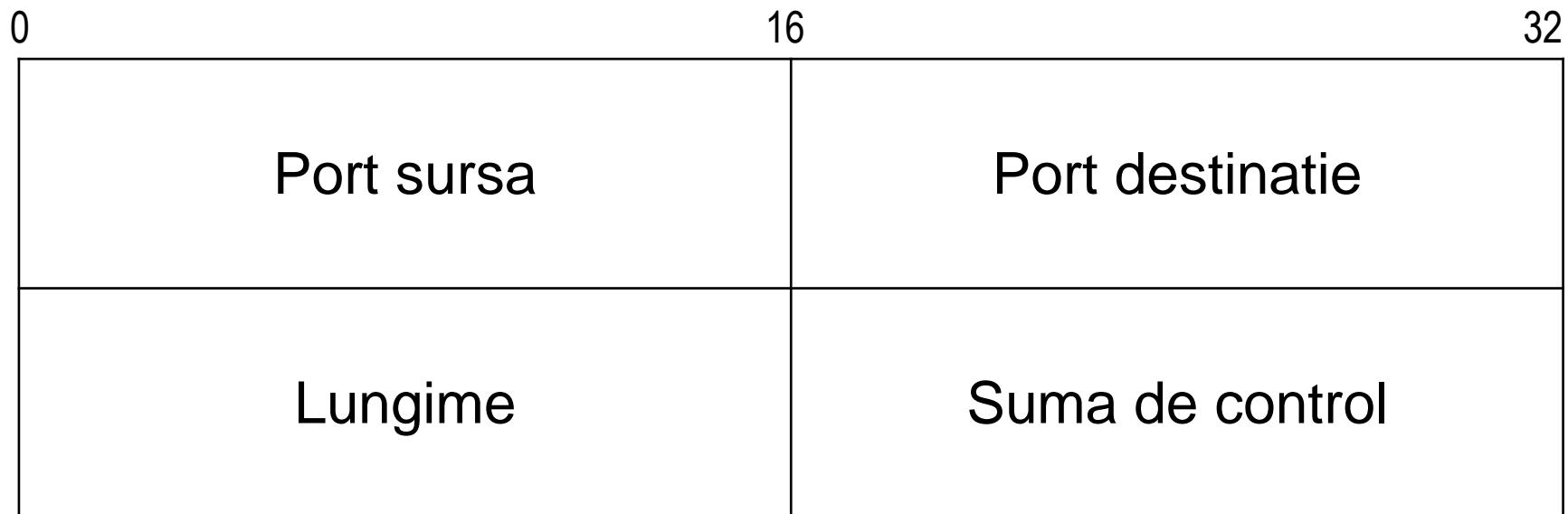
User Datagram Protocol (UDP)

- Neorientat conexiune;
- Nesigur (unreliable) (segmente pierdute);
- Fara controlul fluxului (segmente fara ordine).

Utilizare:

- atunci cand conexiunile orientate ofera un randament scazut : ex. DNS, SNMP;
- transmisii de date in timp real: ex. Skype;
- aplicatia asigura controlul integritatii: ex. TFTP.

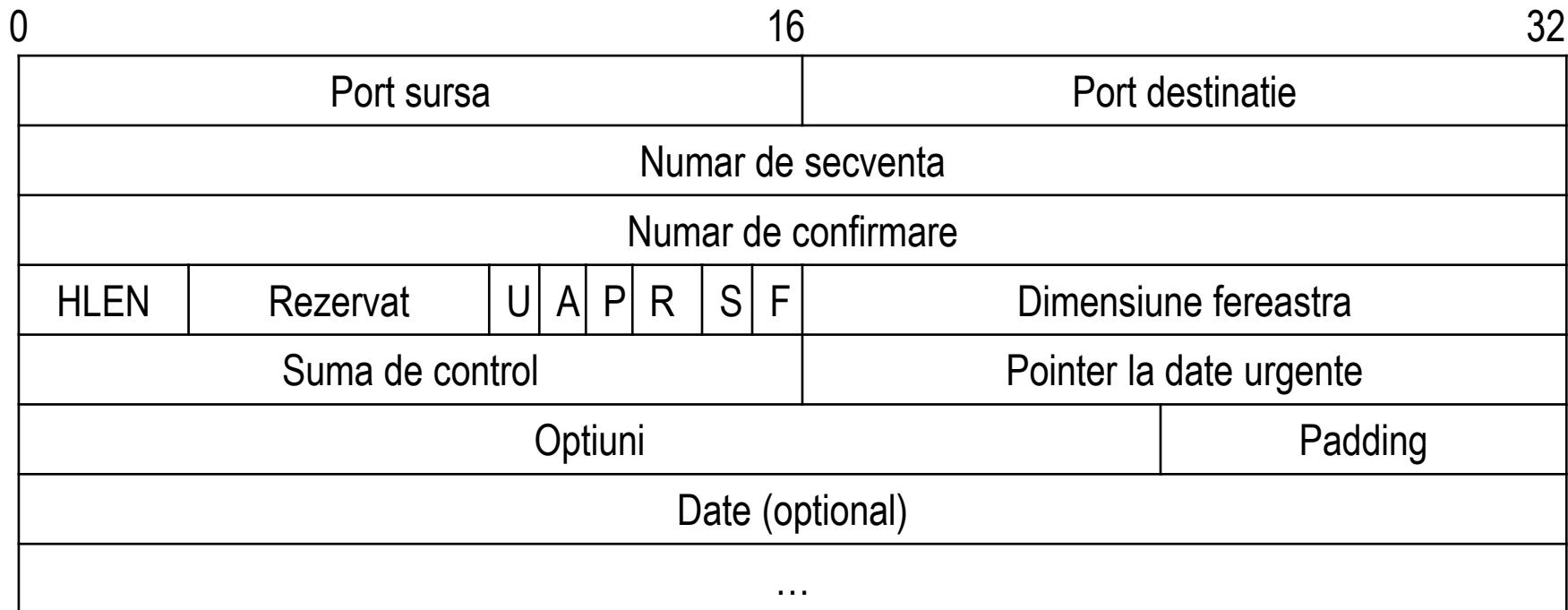
Antet UDP



Transmission Control Protocol (TCP)

- orientat pe conexiune;
- circuit virtual in care are loc comunicatia;
- protocol sigur (reliable) (datele ajung garantat la destinatie);
- datele ajung in ordine la destinatie;
- numere de sevenanta si numere de confirmare;
- controlul fluxului: corelare sender si receiver prin fereastra glisanta;
- controlul congestiei;
- controlul erorii: suma de control.

Antet TCP



Numar de secenta:

- indexul primului octet din segmentul TCP;
- in cadrul fiecarui segment;

Exemplu:

- primul octet are numarul de secenta 1000;
- cel de-al 100-lea octet are numarul de secenta 1099.

Numar de confirmare:

- indexul urmatorului octet pe care receptorul se asteapta sa-l primeasca de la transmitator;
- confirmarea primirii datelor de pana la acest numar;
- nu este present in toate segmentele;
- activat de prezenta campului (fanionului) ACK.

Grup de 8 biti din antetul TCP.

Identifica diverse stari ale protocolului.

Mai multi biti pot fi activi simultan.

URG

- semnalizeaza transmisia unor date urgente;
- activeaza campul “pointer la date urgente”.

PSH

- fanionul determina golirea imediata a bufferelor: livrare imediata (pentru eficienta TCP foloseste buffere de intrare si iesire).

Exemplu:

- transmiterea sechetei “login:” in retea.

RST

- utilizat pentru resetarea conexiunii;
- invalideaza numerele de secventa.

ACK

- utilizat pentru confirmarea pachetelor transmise;
- activeaza campul “numar de confirmare”.

SYN

- utilizat in protocolul de initiere a conexiunii (handshake).

FIN

- utilizat in protocolul de inchidere a conexiunii.

HLEN (Header Length)

- lungimea antetului TCP în cuvinte de 32 de octeți.

Dimensiune fereastra

- spațiu pentru stocare date neconfirmate (receptor);
- valoarea maxima este 65535;
- opțiune de scalare a ferestrei.

Suma de control

- antet + date.

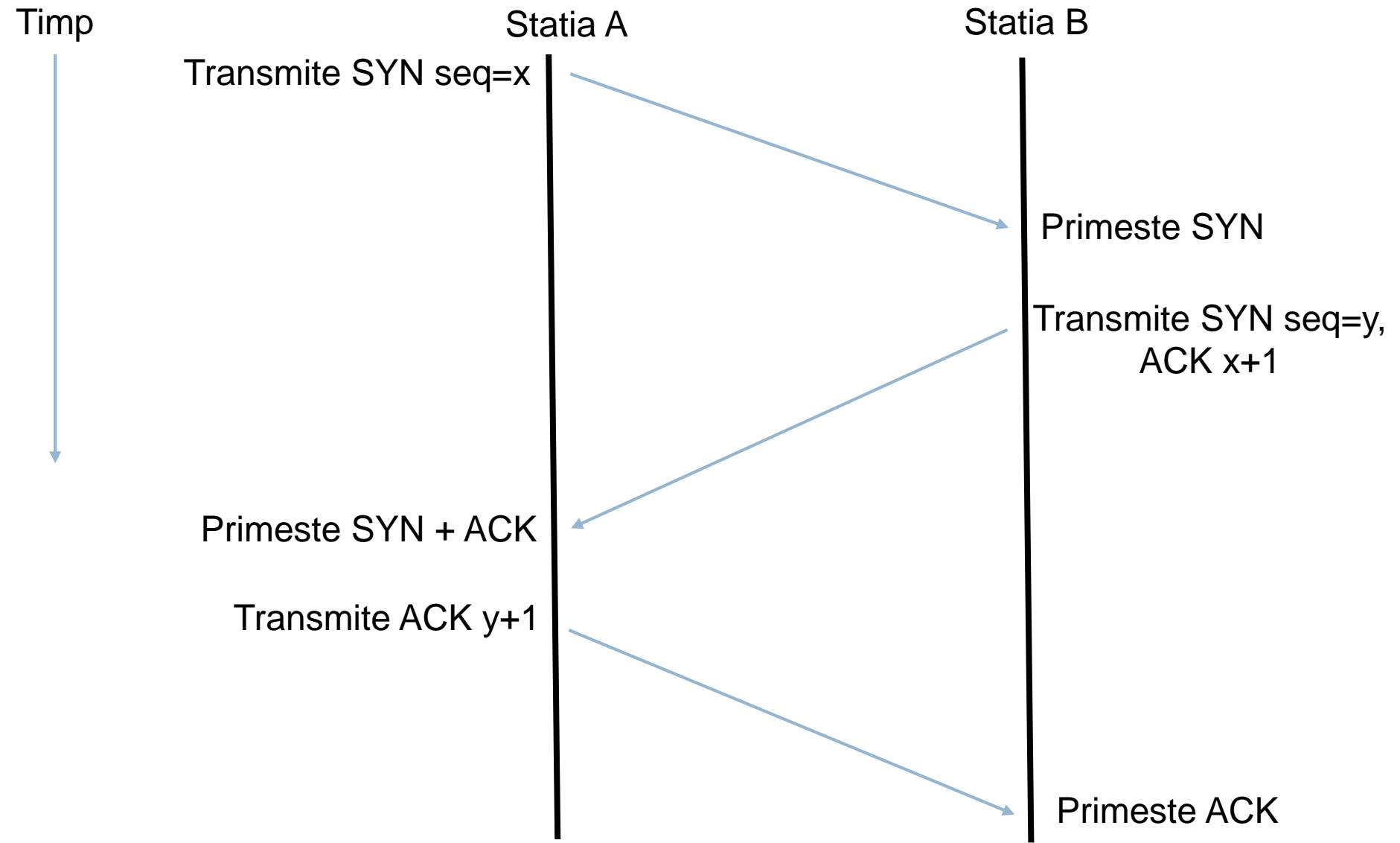
Optiuni

- diverse opțiuni / extensii definite în RFC.

Exemplu:

- specificarea MSS (Maximum Segment Size).

TCP - Initierea conexiunii



Clientul este entitatea activă – inițiază conexiunea.

Campul SYN este activat.

ISN (Initial Sequence Number) = numărul de secvență dintr-un segment cu SYN activat.

Protocolul de inițiere de conexiune - **3-way handshake**.

- primul pachet (SYN)
 - stabilirea ISN pentru comunicatia de la client la server.
- al doilea pachet (SYN+ACK)
 - confirmarea primului pachet;
 - stabilirea ISN pentru comunicatia de la server la client.
- al treilea pachet (ACK)
 - confirmarea celui de-al doilea pachet.

Cele două ISN sunt generate aleator.

De ce sunt necesare două numere de sevență ?

- comunicația este full duplex.

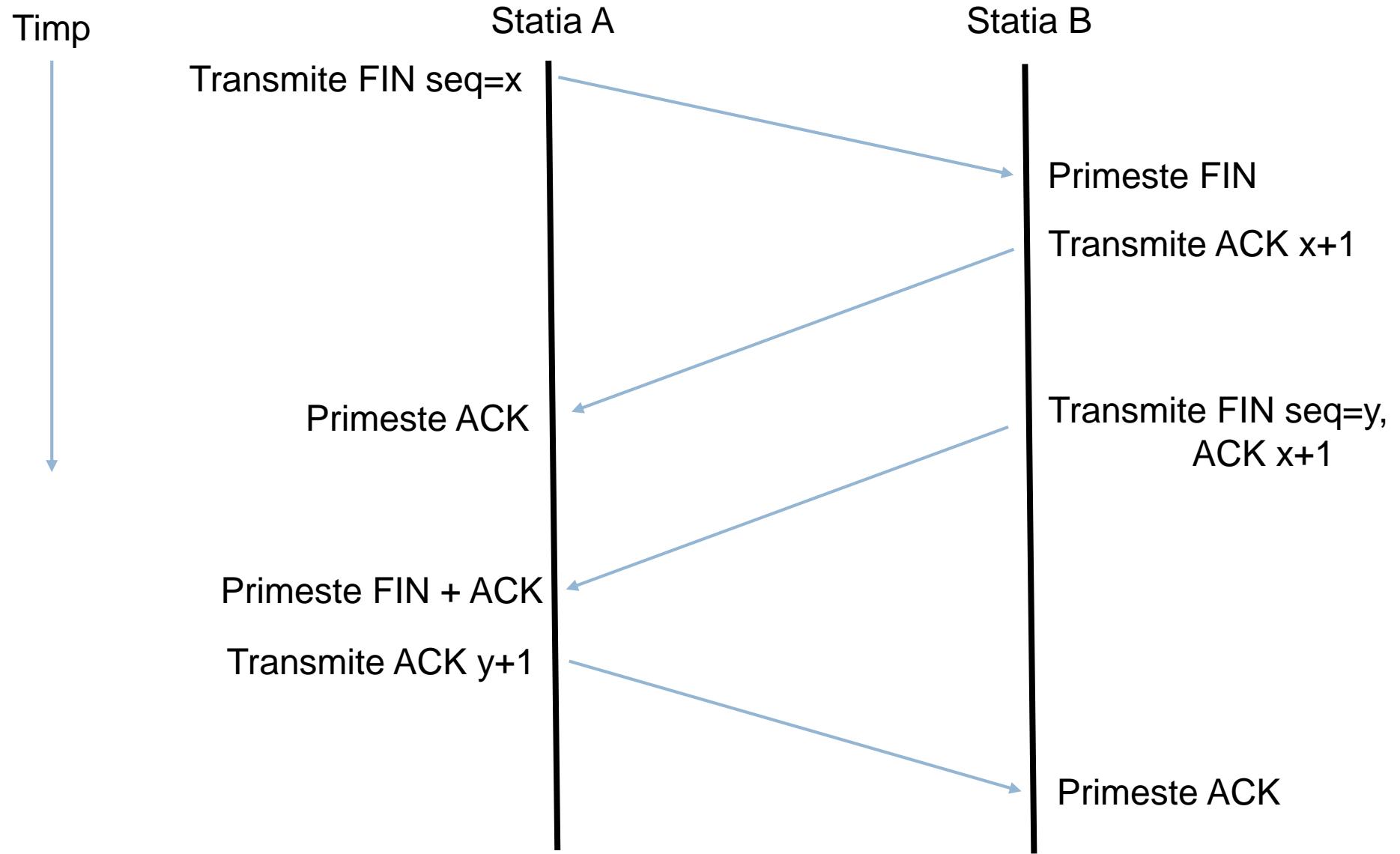
O conexiune TCP

- două canale virtuale de comunicatie:
client → server;
server → client.

Un socket = două buffere (citire / scriere).

Este posibila comunicatia half-duplex prin inchiderea unui capat al conexiunii.

TCP - Încheierea conexiunii



Initiată de oricare capat al transmisiei.

Campul FIN activat.

Protocol de tipul **4-way handshake**:

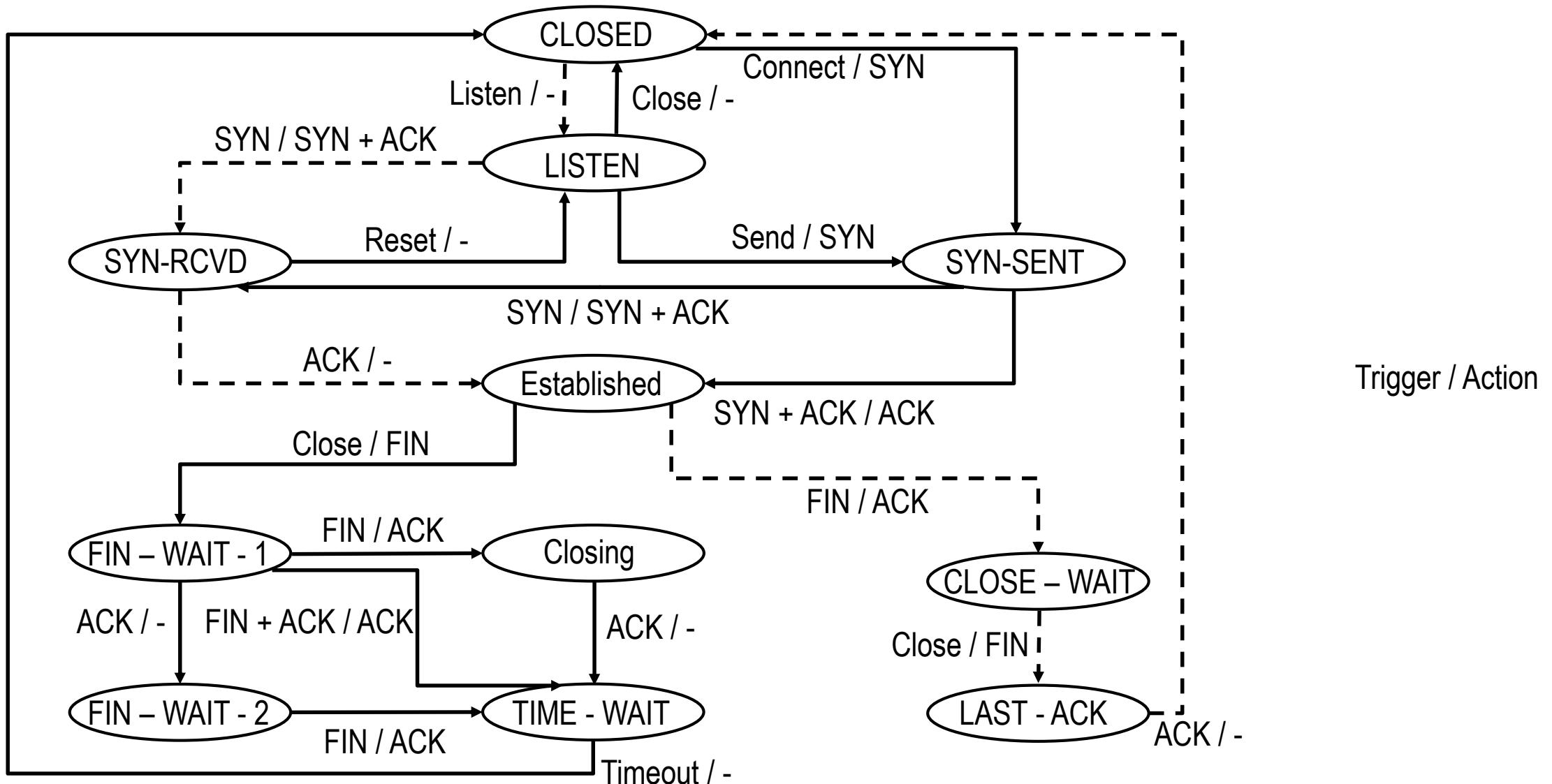
- primul segment are campul FIN activ;
- al doilea segment este o confirmare a primului;

Dupa al doilea pachet transmis conexiunea este pe jumătate închisa (HALF CLOSED): comunicatia este intr-un singur sens.

- urmatoarele două segmente inchid conexiunea în celalalt sens.

Este posibil și un protocol de tipul 3-way handshake atunci când cele două entități inchid conexiunea în același timp. În acest caz al doilea și al treilea segment sunt "unite".

TCP- Diagrama de stari



Automatic Repeat reQuest (ARQ) – metodă de control al integritatii/erorilor studiată la nivelul legătură de date.

- folosește mesaje de confirmare (ACK) și timpi de expirare pentru a asigura fiabilitatea transmisiei.

Dacă transmițătorul nu primește un mesaj de confirmare din partea receptorului într-un interval de timp predefinit, acesta va retransmite mesajul.

Tipuri de ARQ:

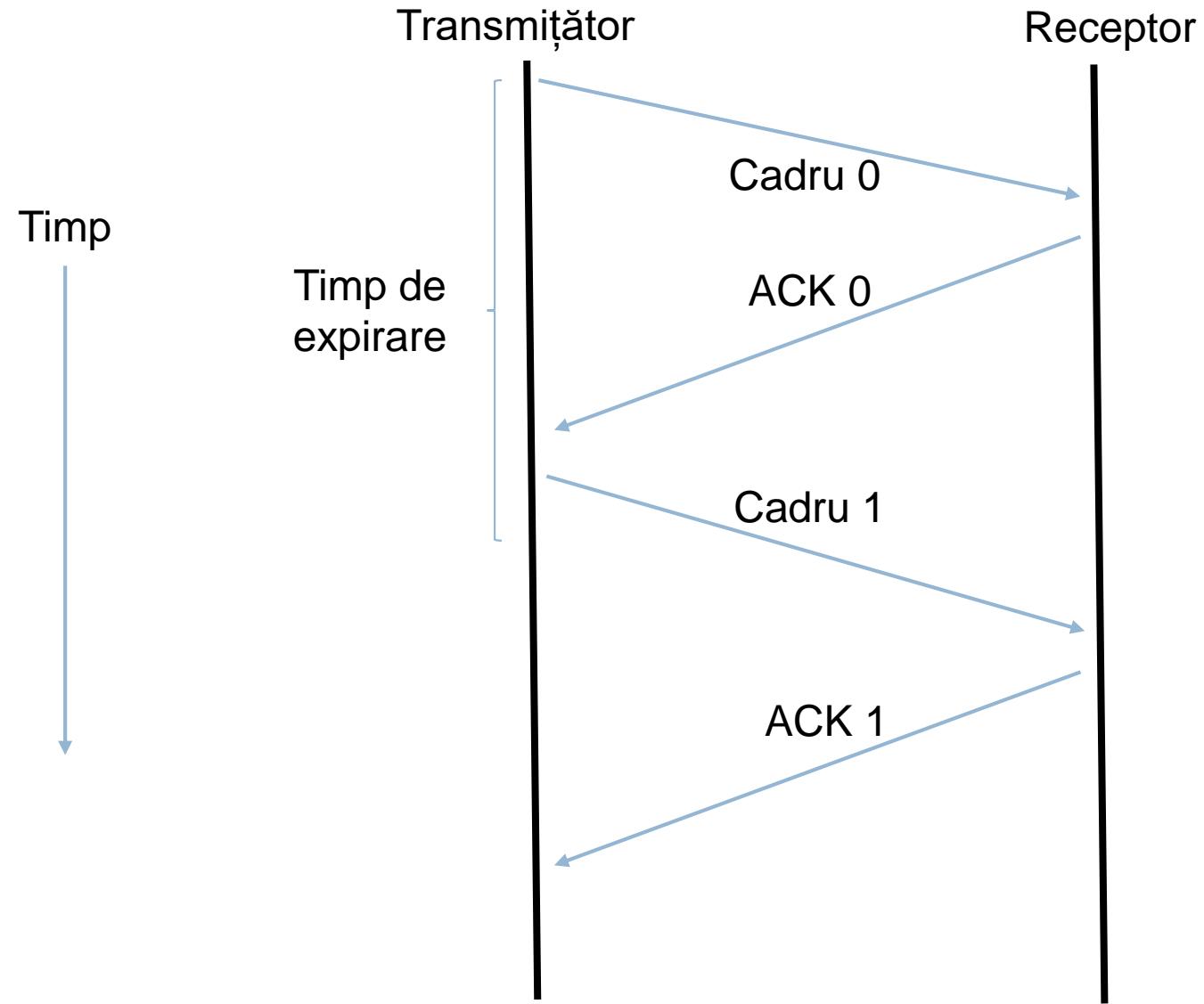
Stop-and-wait

Go-back-N

Selective Repeat

TCP - Fereastra glisanta

Stop-and-wait ARQ



Sliding Window (SW) – generalizare Stop-and-Wait

Permite transmisia a W pachete / RTT.

Exemplu limitare Stop-and-wait ARQ clasic:

Lățime bandă (B) canal = 1 Mbps

Latență (L) = 50 ms => RTT = 100 ms

Dimensiune Pachet = 10000 biți = 1250 bytes

Rată de transfer = $(1s / 100ms)$ Pachete * Dimensiune Pachet = 100 Kbps << B

Eficiență maximă pentru $W = 2BL / \text{Dimensiune Pachet}$.

Exemple de implementare:

Go-Back-N – simplu, ineficient;

Selective Repeat – complex, eficiență sporită.

Sliding Window – Transmițător

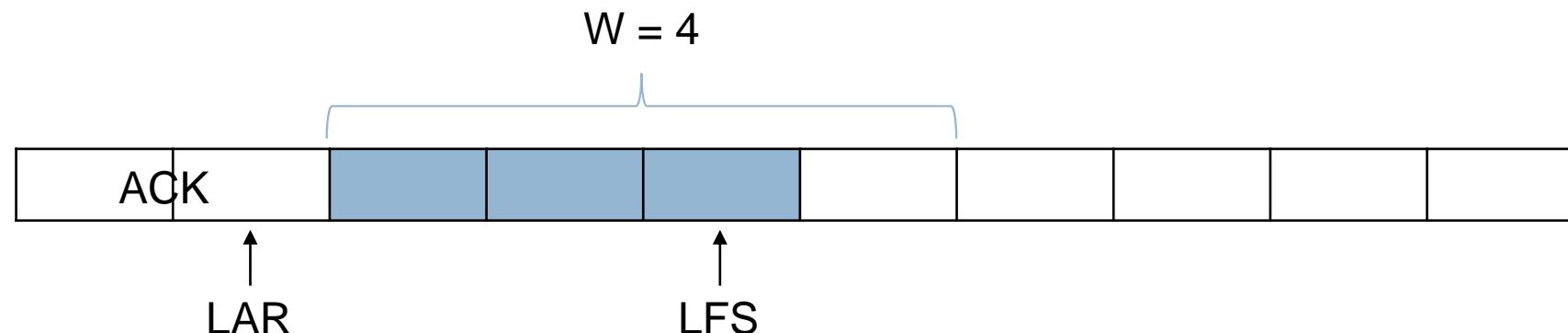
Transmițătorul menține în memoria tampon până la W segmente până când acestea sunt confirmate.

Variabile de stare:

LFS – Last Frame Sent

LAR – Last ACK Received

Funcționare: trimit segmente atât timp cât $LFS - LAR \leq W$



Sliding Window – Receptor

Go-Back-N

Receptorul menține o memorie tampon pentru următorul segment

Variabilă de stare:

LAS – Last ACK Sent

Funcționare:

La recepție:

- Dacă numărul de secvență este LAS + 1 acceptă segmentul și transmite-l aplicației
- Dacă numărul de secvență diferă de LAS + 1 distrugе pachetul (out of order)

Sliding Window – Receptor

Selective Repeat

Receptorul:

- transmite datele către aplicație în ordine;
- menține o memorie tampon de dimensiune W pentru segmentele ajunse în ordine incorectă

Se transmit mesaje de confirmare pentru ultimul mesaj ajuns în ordine

Variabilă de stare:

LAS – Last ACK Sent

Funcționare:

La recepție:

- Menține în memoria tampon segmentele cu numărul de secvență între $[LAS+1, LAS+W]$
- Trimit mesajul având numărul de secvență $LAS+1$ la aplicație și actualizează LAS
- Trimit ACK pentru LAS

Sliding Window – Retransmisia

Go-Back-N – folosește un singur timp de expirare

- La expirare retransmite pachetele salvate în memoria tampon începând cu LAR+1

Selective Repeat – folosește câte un timer pentru fiecare segment neconfirmat

- La expirare retransmite doar segmentul asociat

Timpul de expirare:

$$\text{SRTT}_{N+1} = 0.9 * \text{SRTT}_N + 0.1 * \text{RTT}_{N+1}$$

$$\text{Svar}_{N+1} = 0.9 * \text{Svar}_N + 0.1 * (\text{RTT}_{N+1} - \text{SRTT}_{N+1})$$

$$\text{Timeout}_N = \text{SRTT}_N + 4 * \text{Svar}_N$$

SRTT = Smoothed RTT

Svar = Smoothed Variance

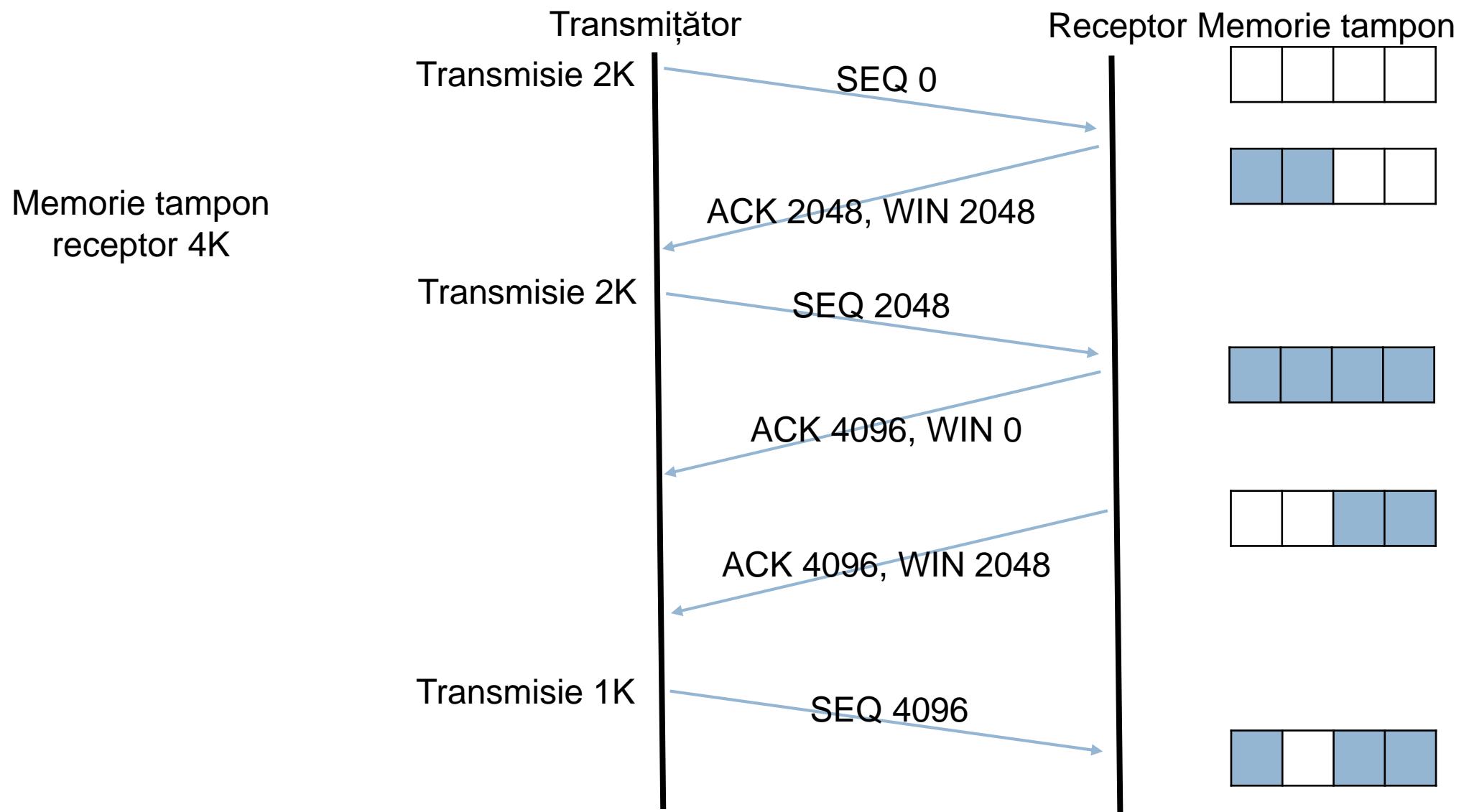
Controlul fluxului

Dimensiunea ferestrei transmitatorului este controlată de cea a receptorului.

Receptorul definește o fereastră de control al fluxului WIN pe care o trimite transmițătorului.

Transmițătorul folosește fereastra cu dimensiunea cea mai mică dintre SW și WIN.

TCP - Controlul fluxului

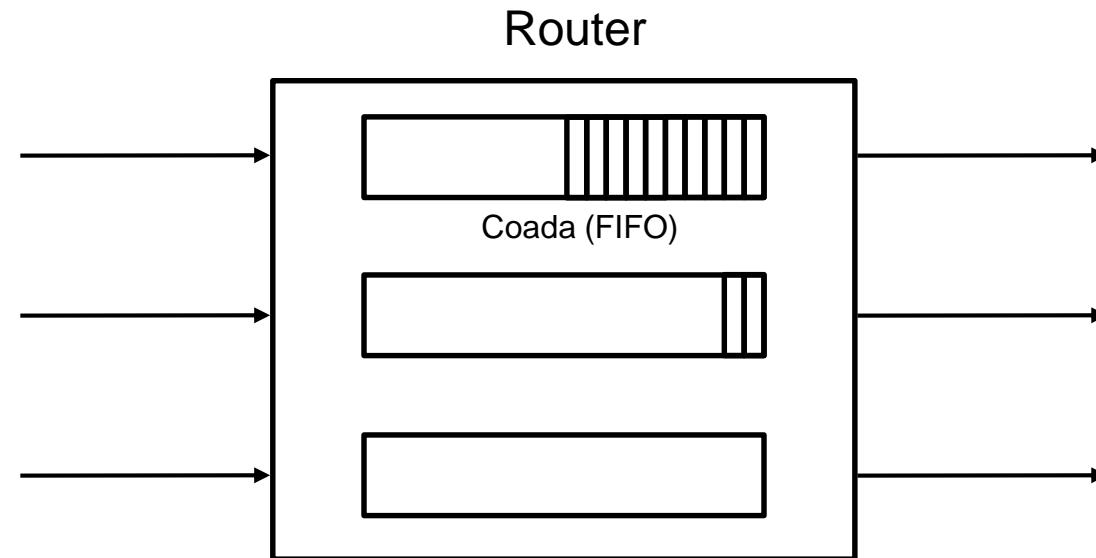


Controlul congestiei

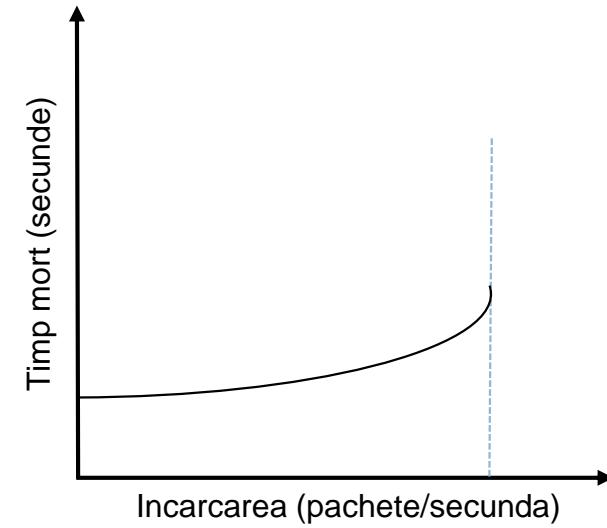
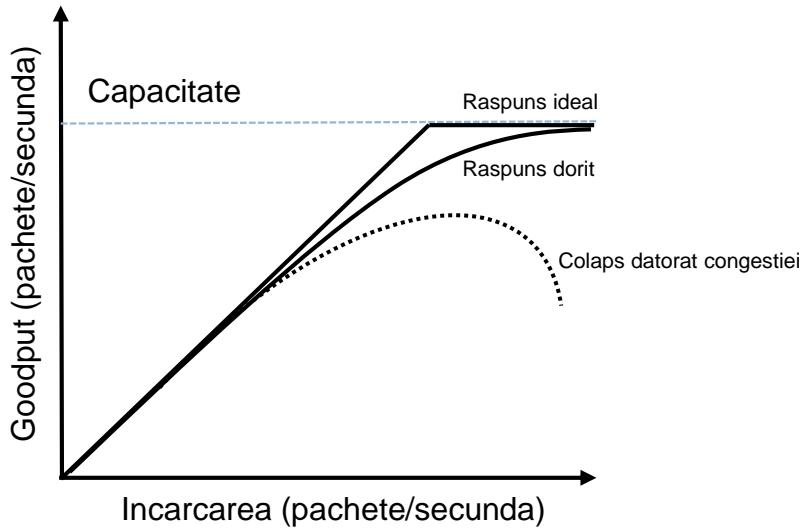
Controlul congestiei



Aparitia congestiei



Efectele congestiei



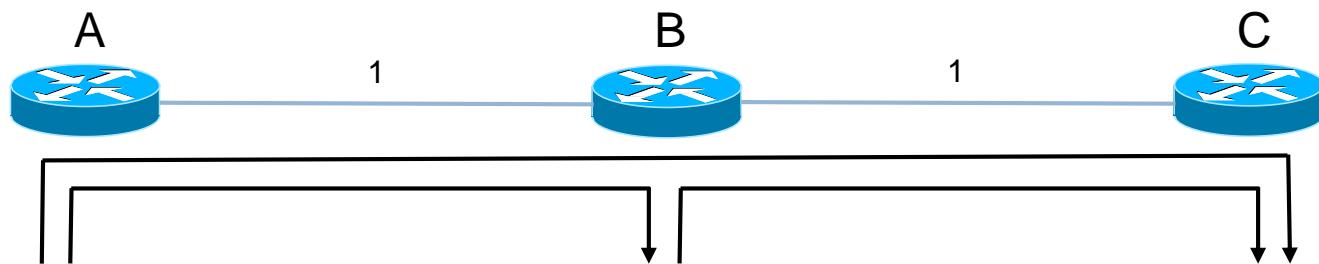
Alocarea de banda potrivita in retea presupune:

- eficienta = se foloseste toata capacitatea de banda fara a se ajunge la congestie
- corectitudine = fiecare transmitator primeste o capacitate de banda rezonabila

Eficienta \leftrightarrow Corectitudine

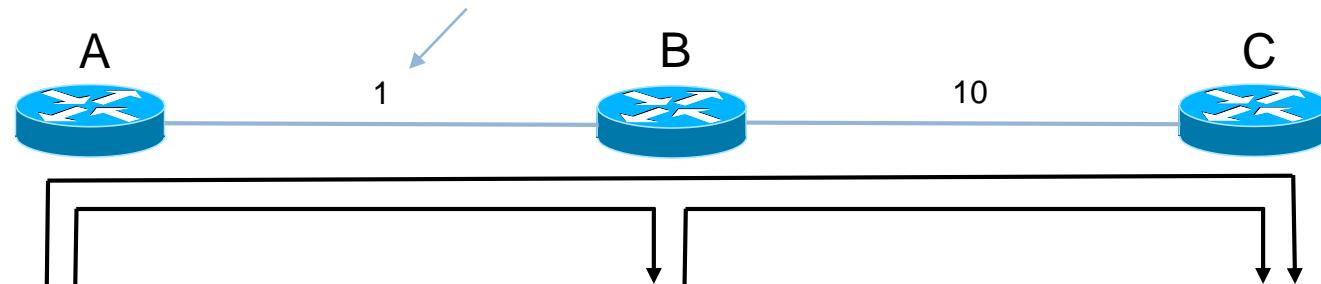
Ce inseamna corectitudine ?

- egalitate / flux;
- este importanta evitarea izolarii.



Corectitudinea alocarii de banda

Bottleneck = segmentul de retea care limiteaza latimea de banda a unui flux de date



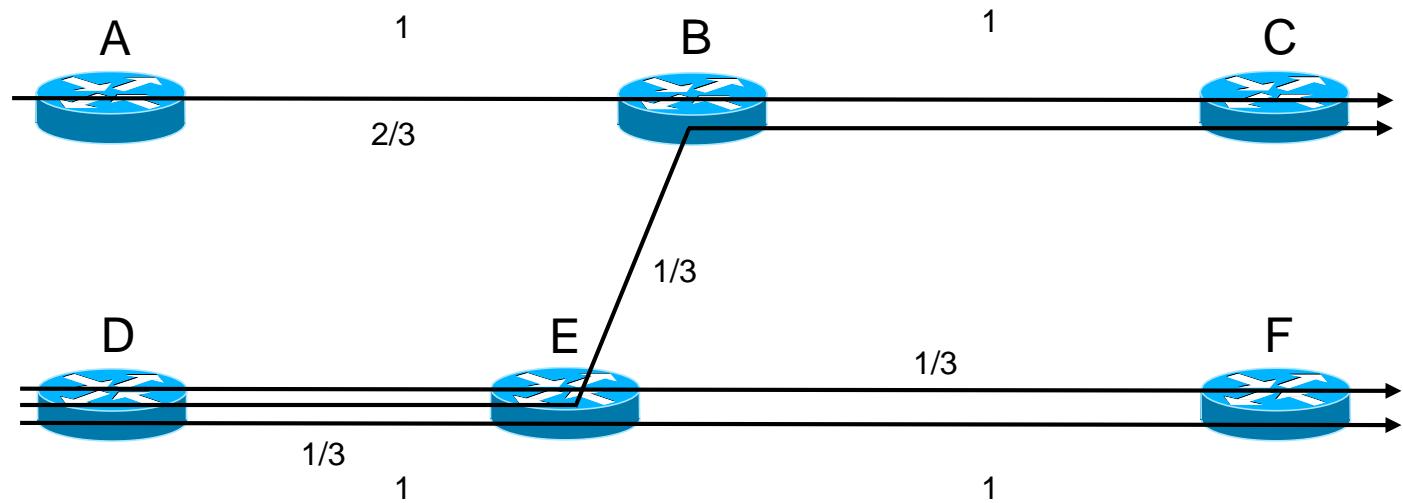
Corectitudinea $\text{Max} - \text{Min} = \text{fluxurile de date limitate de un segment de retea}$ vor avea alocate parti egale din capacitatea de banda a respectivului segment.

Algoritmul de alocare:

1. Initializarea tuturor fluxurilor de date la valoarea 0.
2. Incrementarea valorii fluxului de date pana la aparitia unei limitari in retea.
3. Pastrarea unei rate fixe pentru fluxurile limitate.
4. Repetarea de la pasul 2 pentru fluxurile ramase.

Corectitudinea alocarii de banda

Exemplu:



Adaptarea in timp



Modele de alocare de banda

Alocarea de banda se face in bucla:

- deschisa = rezervare de banda inainte de utilizare;
- Inchisa = ajustarea ratei de transfer in functie de feedback.

Atribuirea alocarii este efectuata de:

- de retea = alocarea se face de retea;
- de nod = alocarea se face de nod.

Alocarea este exprimata:

- in dimensiunea ferestrei glisante;
- rata absoluta de transfer.

TCP - alocare de banda

TCP = alocare in bucla inchisa, atribuire efectuata de nod si exprimata in dimensiunea ferestrei.

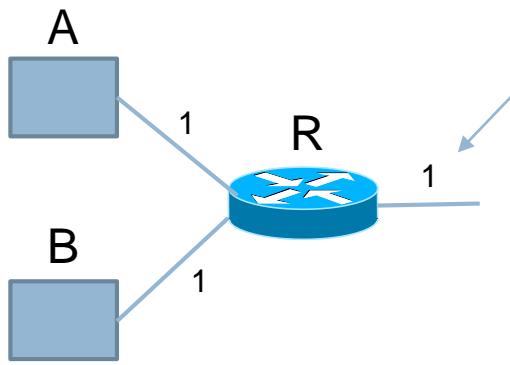
Nivelul retea furnizeaza feedback-ul (informeaza asupra aparitiei congestiei).

Nivelul transport ajusteaza comportamentul transmitatorului, modificand fereastra glisanta cu ajutorul unei legi de control.

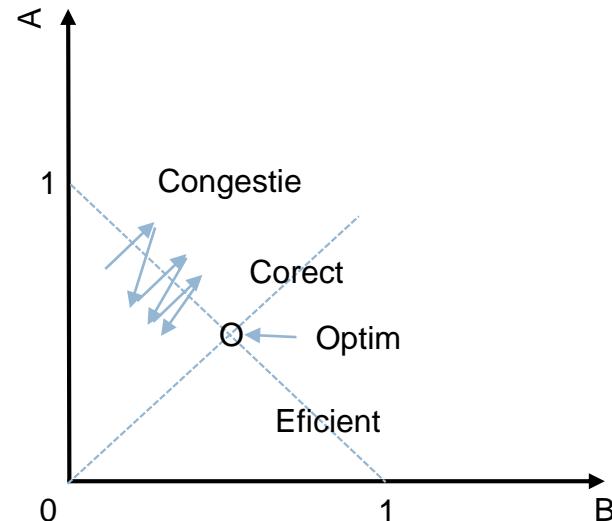
Additive Increase Multiplicative Decrease – o lege de control ce poate fi utilizata de noduri pentru atingerea unei bune alocari de banda ca si corectitudine si eficienta.

- se incrementeaza aditiv rata de transfer cat timp nu exista congestie;
- se decrementeaza multiplicativ rata de transfer la aparitia congestiei.

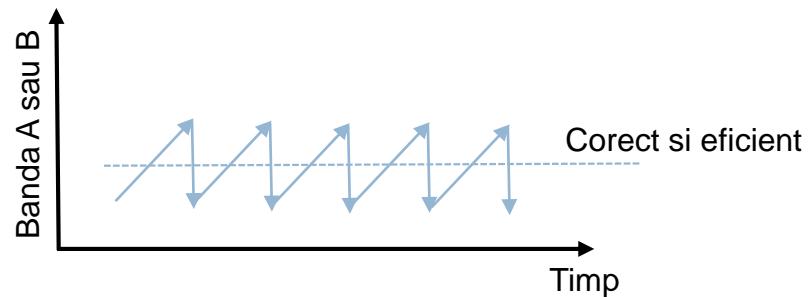
Additive Increase Multiplicative Decrease (AIMD)



Alocare banda:



Proprietati AIMD



Converge catre o alocare corecta si eficienta.
Necesa feedback minim de la retea (binar).

Semnale de feedback

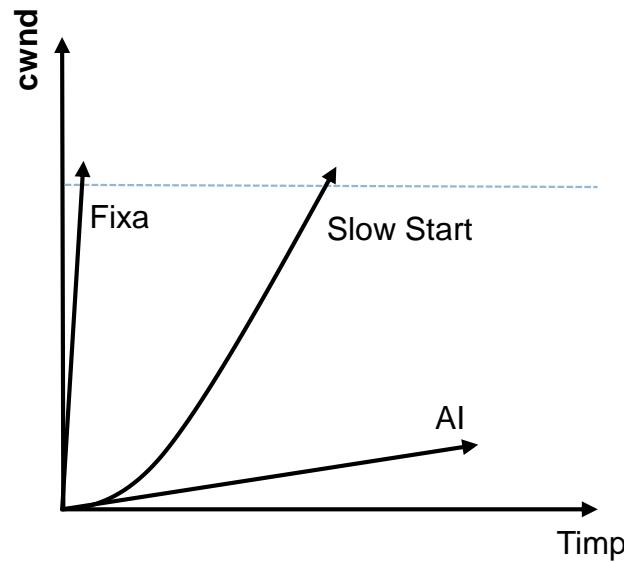
Semnal	Exemplu protocol
Pierdere de pachete	TCP NewReno
Intarzierea pachetelor	Compound TCP
Atentionare router	TCP cu ECN

Slow start – componenta a AI in TCP

- TCP defineste o fereastra de congestie **cwnd** pentru stabilirea ratei de transfer.
- TCP trebuie sa functioneze in conditii de latime de banda foarte diferite.
- O fereastra fixa nu se adapteaza la conditiile din retea.
- AI clasic necesita un timp mare de ajustare pentru **cwnd**.
- Se doreste ajungerea la o rata de transfer ideală **cwnd_{ideal}** cat de repede posibil.

Slow Start

Solutia Slow Start – creste cwnd exponential



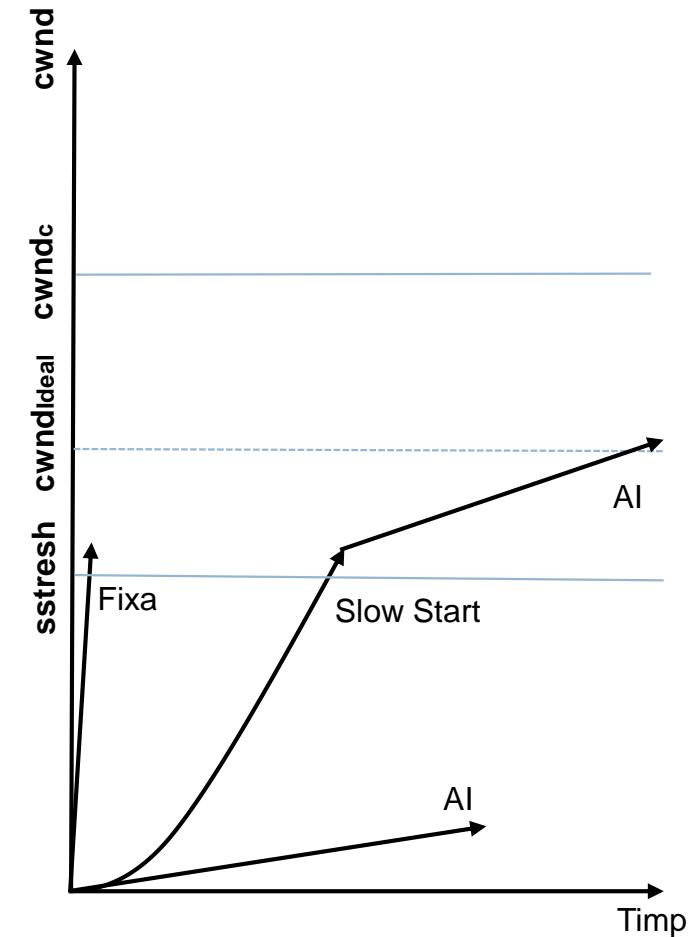
Solutia Slow Start – creste **cwnd** exponential

La aparitia congestiei apare pierderea pachetului

Expirarea timer-ului duce la definirea **cwnd_c**

Se stabileste **ssthresh** = **cwnd_c** / 2

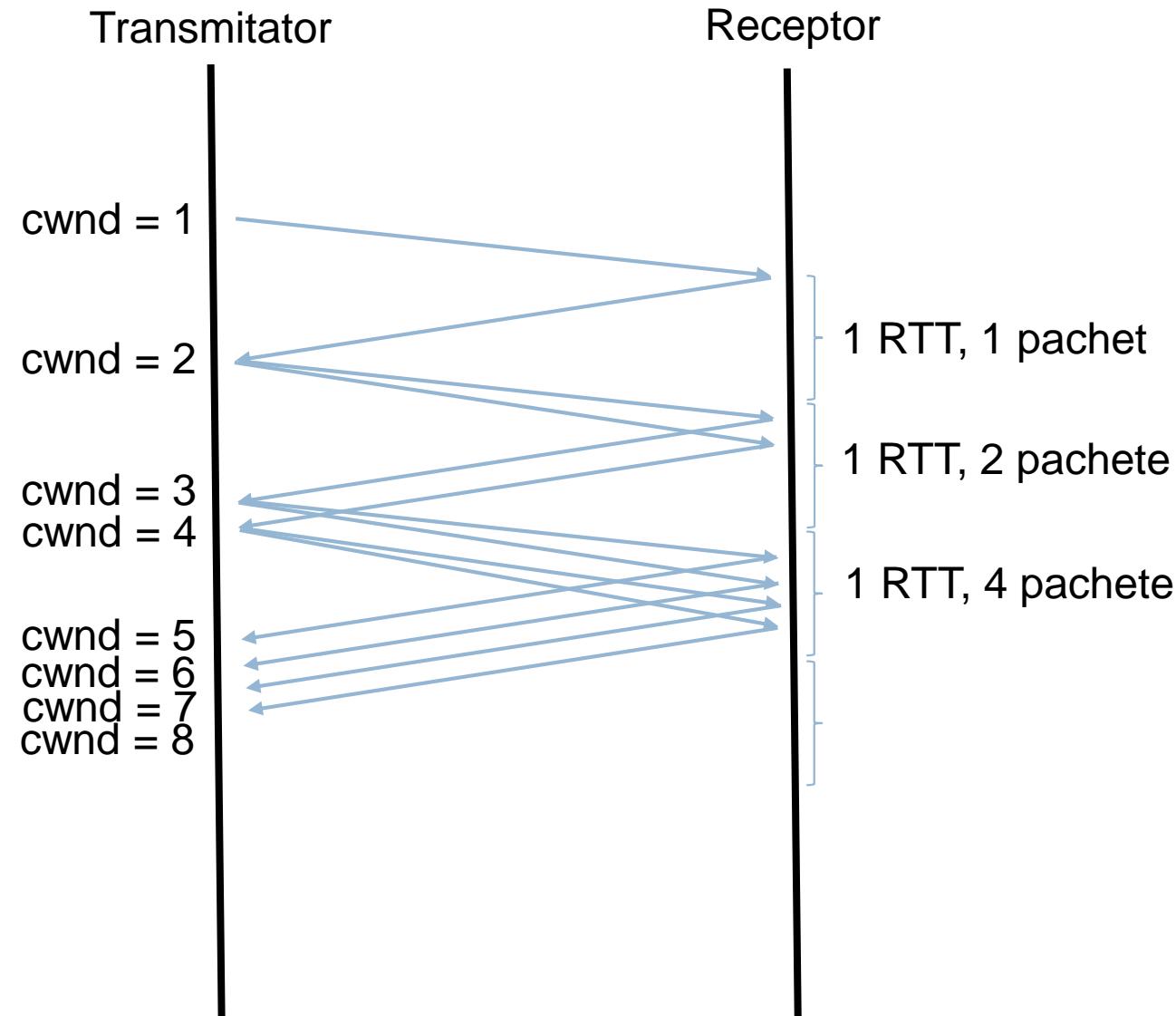
In pasul urmator se utilizeaza AI dupa atingerea **ssthresh**



Slow Start

Faza Slow Start

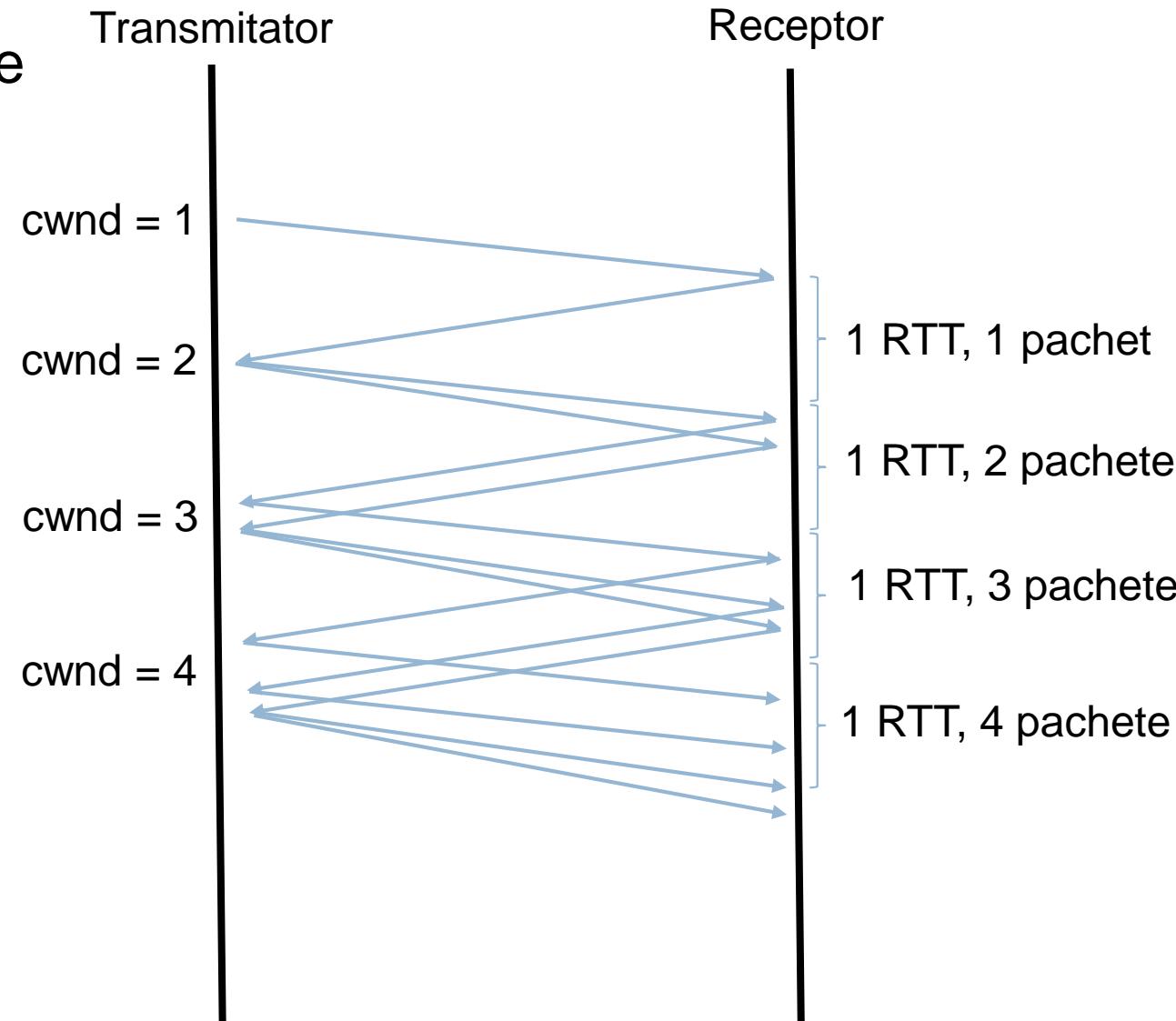
- incrementeaza **cwnd** cu un pachet pentru fiecare ACK primit



Slow Start

Faza Additive Increase

- incrementeaza **cwnd** cu un pachet pentru fiecare RTT



Implementare TCP Tahoe

Faza Slow – Start

- Initializeaza cwnd=1;
- cwnd +=1 pachet / ACK

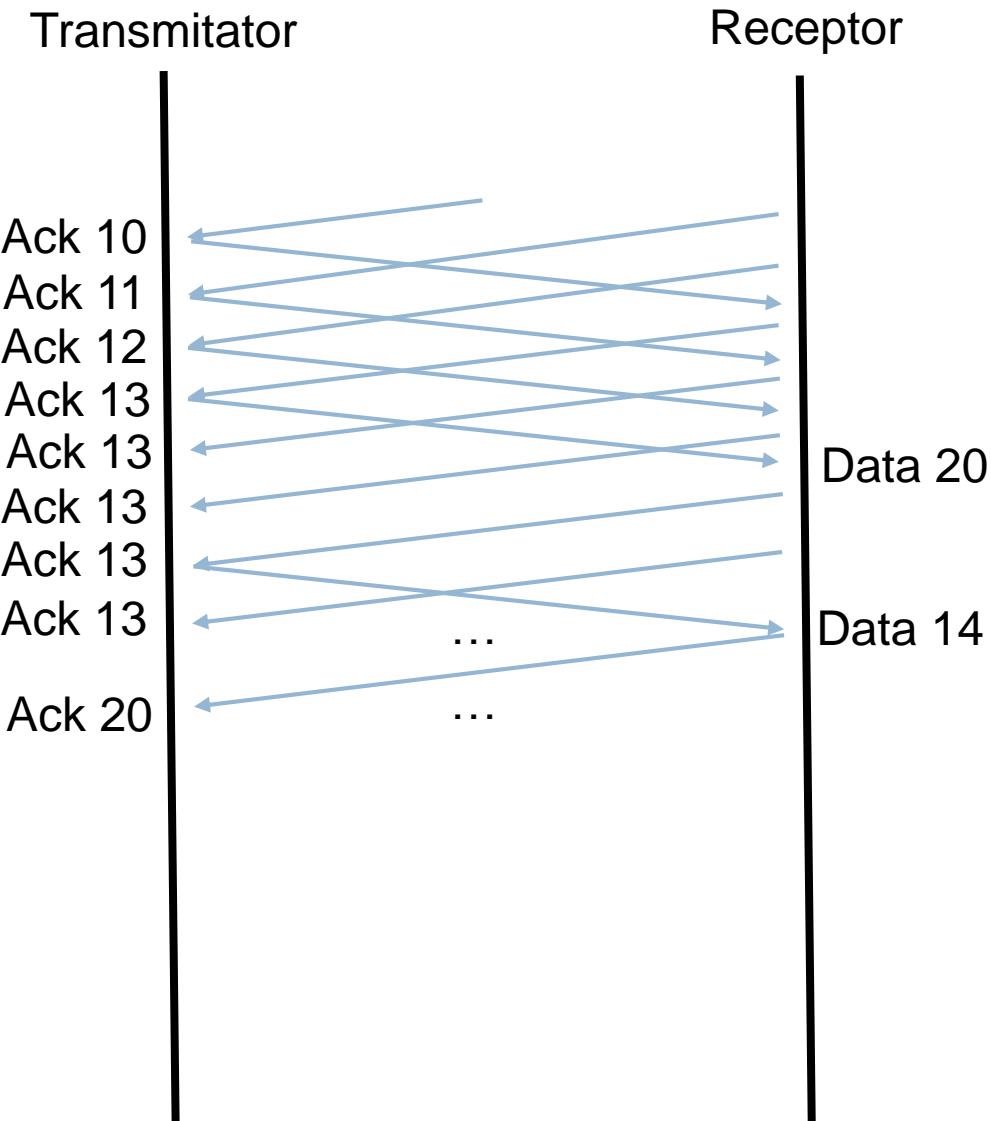
Faza AI

- cwnd = 1/cwnd pachete / ACK

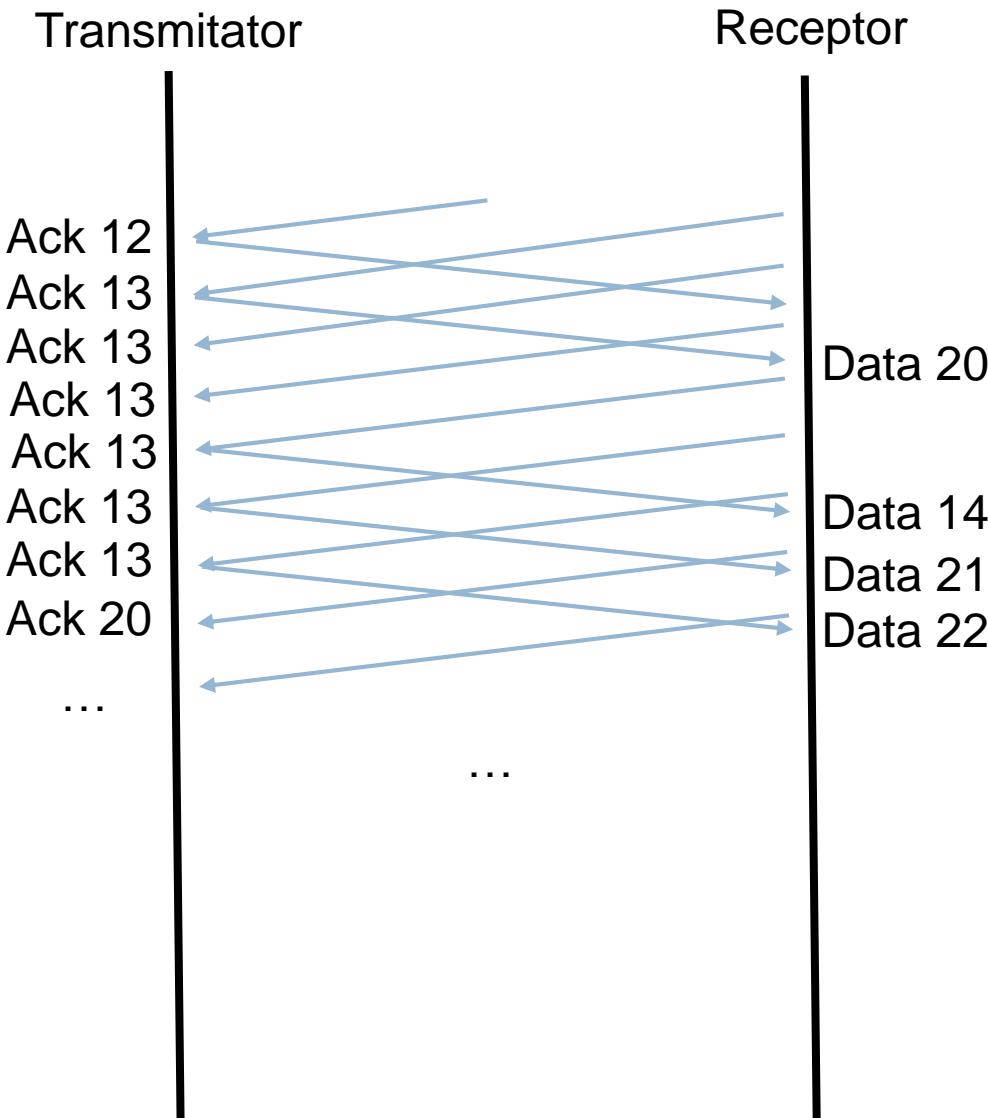
Modificare threshold (initial infinit)

- Comutare pe AI cand cwnd > ssthresh
- ssthresh = cwnd / 2 la pierderea pachetului
- Incepe Slow – Start dupa expirare timer

Fast retransmit



Fast recovery



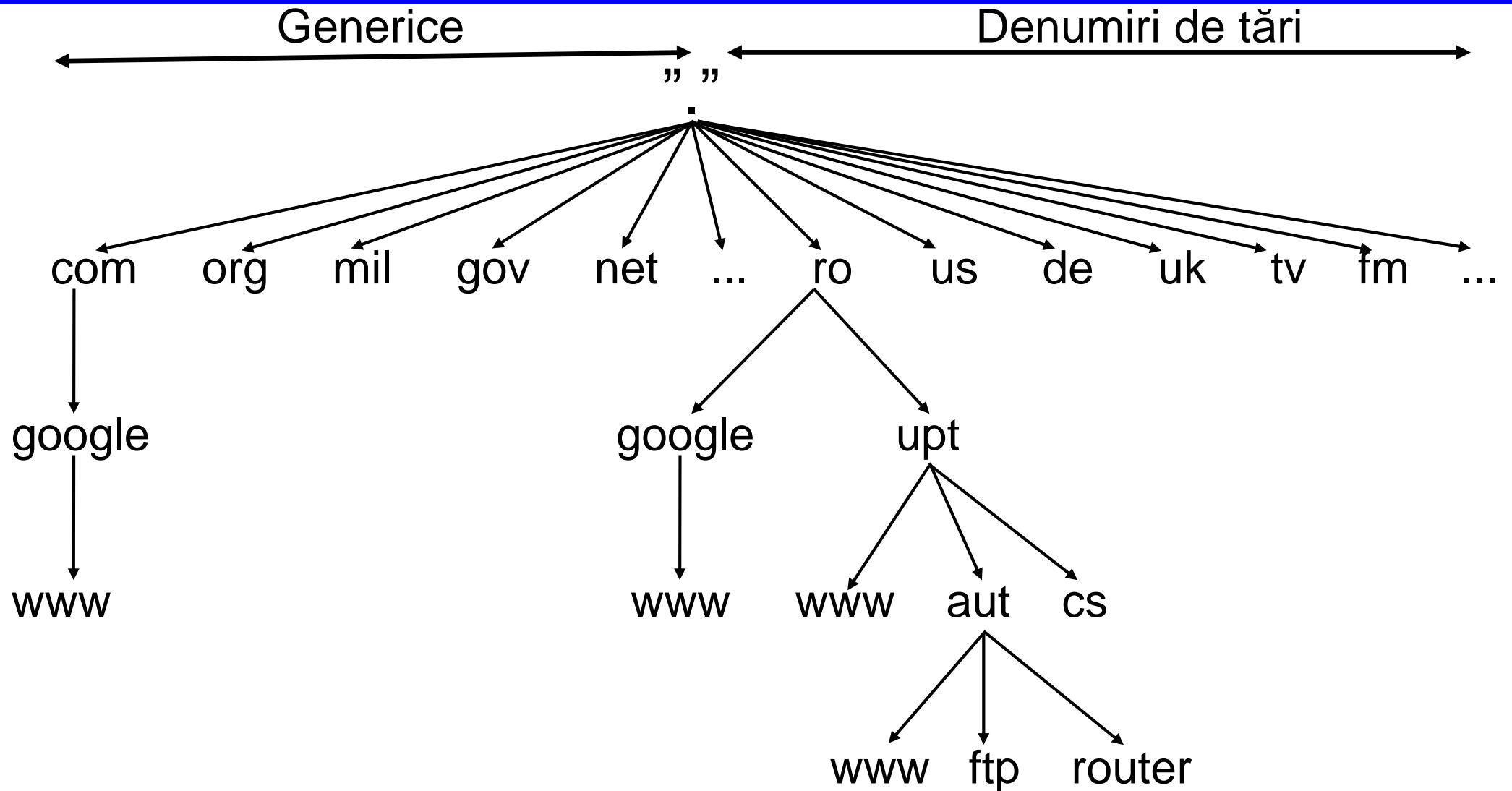
Nivelul aplicație - Sistemul DNS

1. Introducere
2. Istoria DNS
3. Spațiul de nume DNS
4. Zone DNS
5. Înregistrările de tip resursă DNS
6. Rezoluția de nume
7. Resolveri, servere de nume și memoria tampon
8. Protocolul DNS

- Nivelul aplicație al modelului de rețea TCP/IP descrie două categorii de aplicații: cele accesate direct de către utilizator și cele de suport pentru prima categorie de aplicații. Sistemul DNS face parte din categoria aplicațiilor de tip suport și definește un sistem de nume ierarhic, distribuit, utilizat pentru identificarea și accesarea într-o manieră cat mai facilă a resurselor de rețea.
- A apărut din nevoia asocierii unor nume ușor de reținut adreselor logice (valori numerice) ale nodurilor de rețea.
- **Obiective:**
 - ușurință în mențenanță și administrare;
 - eficiență ridicată (temp de răspuns cât mai mic, consum scăzut de resurse).
- **Soluția identificată:**
 - bază de date distribuită construită pe un spațiu de nume ierarhic;
 - un protocol automat și transparent pentru legătura între componentele distribuite.

- În rețeaua ARPANET asocierile dintre numele echipamentelor de calcul și adresele lor se păstrau într-un fișier text: hosts.txt;
 - Fișierul era menținut și actualizat de către NIC (Network Information Center);
 - O dată cu creșterea ARPANET acest mod de asociere a devenit inefficient și greu de utilizat.
-
- Paul Mockapetris a proiectat și implementat o prima versiune a sistemului DNS la Universitatea din California în 1983;
 - Specificațiile inițiale sunt publicate în RFC 882 și RFC 883. Ulterior sunt actualizate în RFC 1034 și RFC 1035;
 - În 1984 apare prima versiune de server Berkeley Internet Name Domain (BIND) pentru UNIX, iar în 1990 este portată pentru Windows NT.

Spațiu de nume DNS

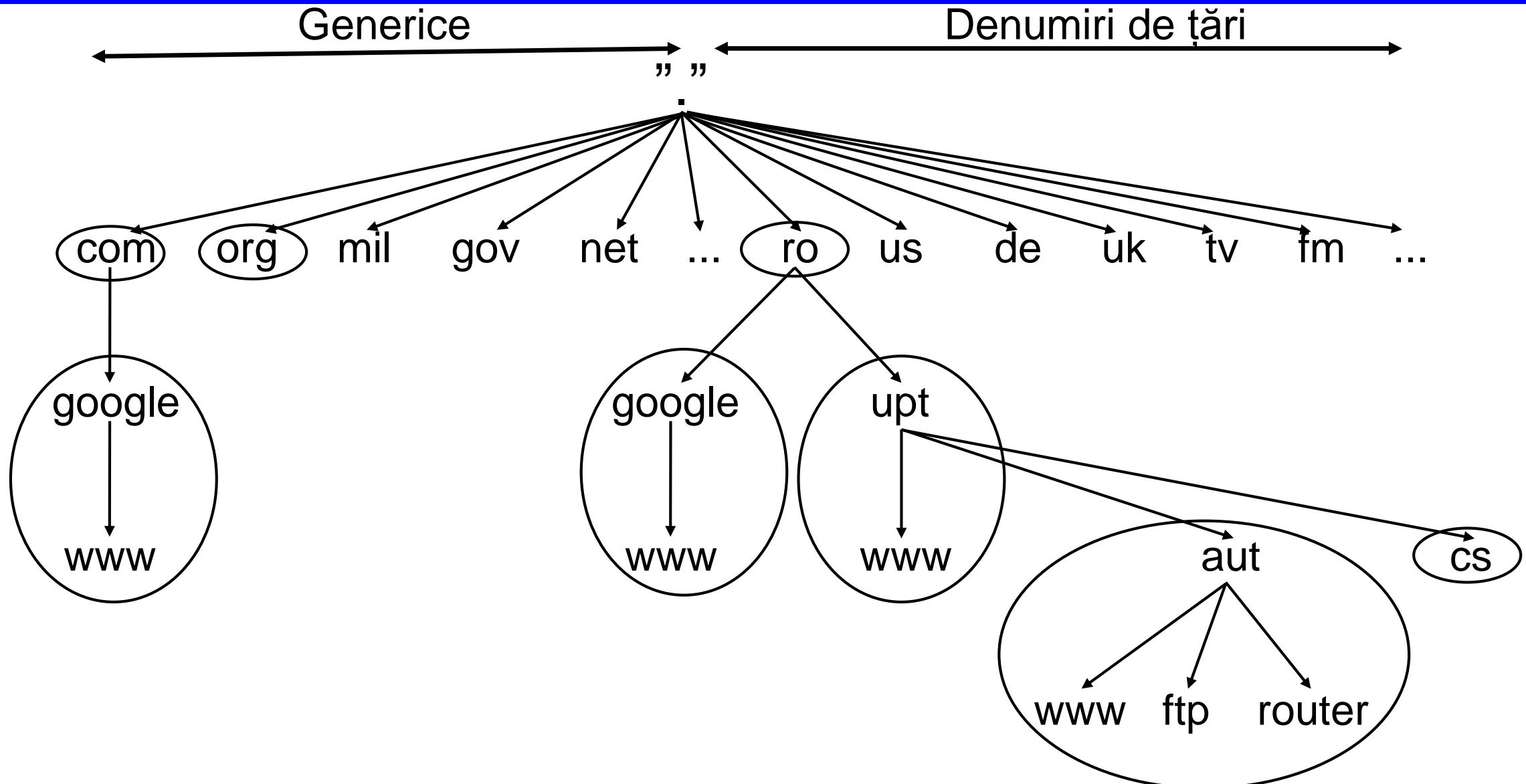


Spațiu de nume DNS

- reglementat de Internet Corporation for Assigned Names and Numbers (ICANN);
- caractere unicode din 2010.

TLD generic	Utilizare	Apariție	Restricții
com	comercial	1985	nu
edu	educațional	1985	da
gov	guvern	1985	da
int	internațional	1985	da
mil	militar	1985	da
net	furnizori servicii internet	1985	nu
org	organizații non-profit	1985	nu
biz	afaceri	2001	nu
aero	transport aerian	2001	da
info	informațional	2002	nu
name	persoane	2002	nu
pro	profesioniști	2002	da
mobi	echipamente mobile	2005	da
travel	călătorii	2005	da

Zone DNS



Înregistrări de tip resursă

- O zonă este compusă din înregistrări de tip resursă DNS care conțin informații asociate numelor de domenii aparținând zonei.

<<Nume domeniu>> <<Temp de viață>> <<Clasa>> <<Tip>> <<Valoare>>

Tip	Definiție	Valoare
SOA	Început de autoritate	Parametrii care definesc zona
A	Adresă IPv4	Întreg pe 32 de biți
AAAA	Adresă IPv6	Întreg pe 128 biți
NS	Server de nume	Nume server DNS
MX	Server de email	Nume server email și prioritate
CNAME	Nume canonic	Nume de domeniu
PTR	Pointer	Nume canonic pentru o adresă IP
TXT	Text	Text descriptiv

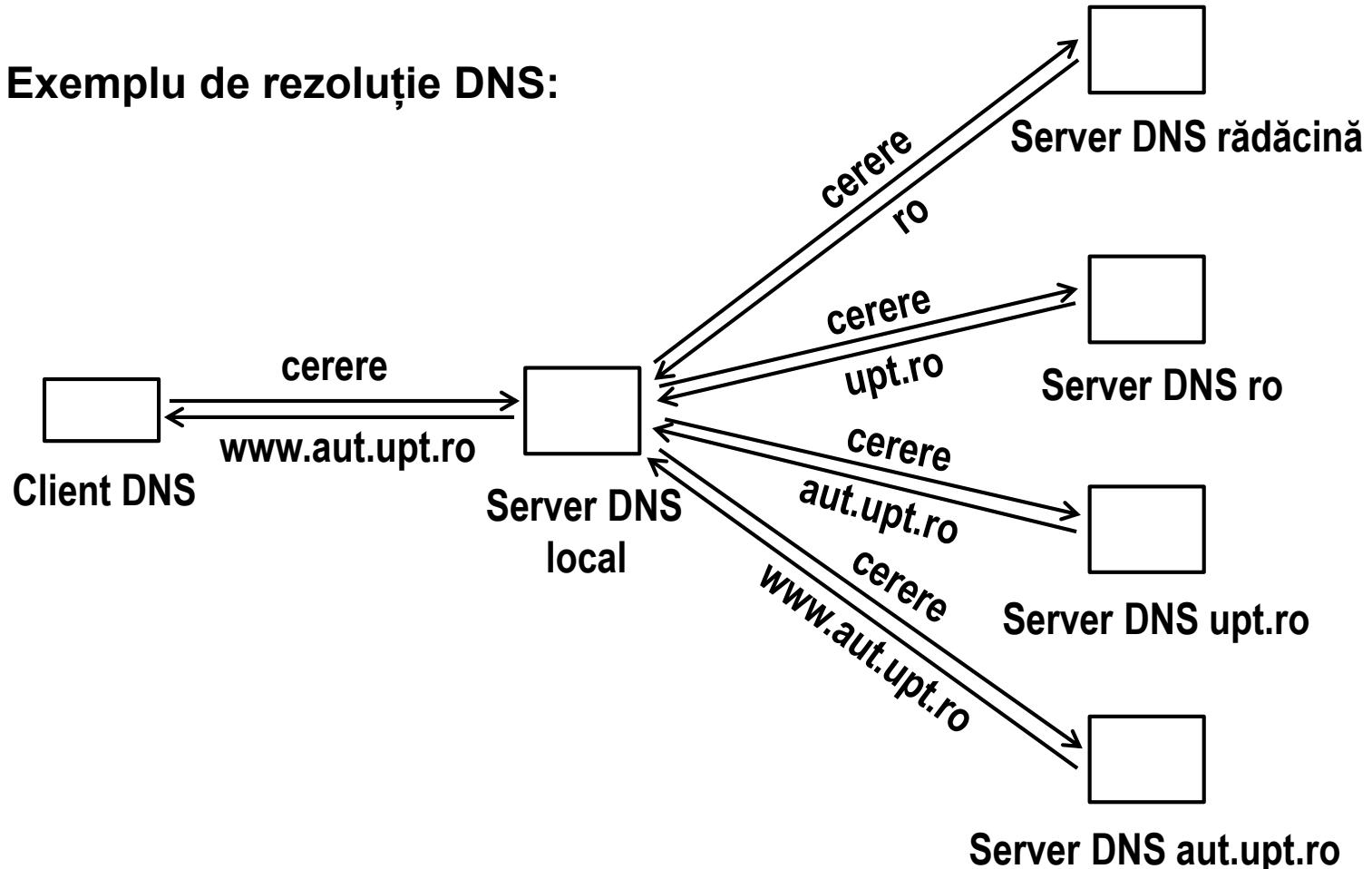
Exemple de înregistrări DNS

aut.upt.ro.	86400 IN SOA	(2015010101 3600 600 86400 86400)
aut.upt.ro.	86400 IN NS	ns1.aut.upt.ro.
aut.upt.ro.	86400 IN MX	1 mail1.aut.upt.ro.
aut.upt.ro.	86400 IN A	193.226.9.150
ns1.aut.upt.ro.	86400 IN A	193.226.9.1
www.aut.upt.ro.	86400 IN A	193.226.9.150
mail1.aut.upt.ro.	86400 IN A	193.226.9.10
router.aut.upt.ro.	86400 IN CNAME	ns1.aut.upt.ro

Rezoluția DNS

- **Rezoluția de nume** este procesul prin care sunt identificate înregistrările de tip resursă asociate unui nume de domeniu.

Exemplu de rezoluție DNS:



Abordare iterativă și recursivă

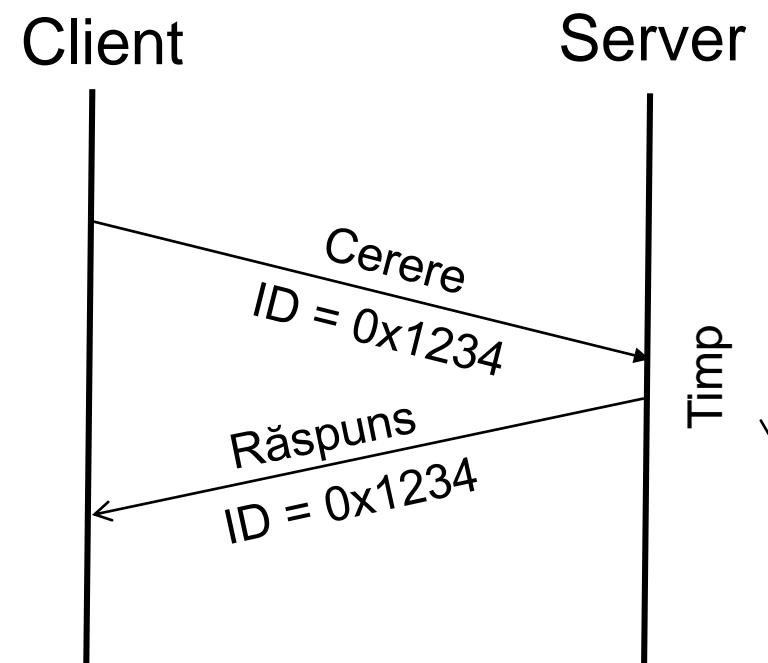
- **Abordare recursivă:**
 - Serverul DNS finalizează rezoluția de nume și returnează rezultatul final.
- **Abordare iterativă:**
 - Serverul DNS returnează fie răspunsul, fie următorul server DNS ce trebuie contactat în vederea identificării răspunsului.

Serverele de nume și memoria tampon (cache)

- Serverele DNS mentin răspunsurile primite în memoria tampon în vederea scăderii latenței rezoluției de nume;
- Perioada de memorare a unei informații depinde de valoarea parametrului TTL.

Protocolul DNS

- Mesaje de tip cerere -> răspuns
 - încapsulare UDP, port 53;
 - identificator pachet pe 16 biți.



Nivelul aplicație - Sistemul de poștă electronică

Cuprins

1. Introducere
2. Istoria sistemului de poștă electronică
3. Arhitectura sistemului
4. Servicii oferite
5. Structura mesajului
6. Formatarea mesajului
7. Extensia MIME
8. Protocole de transfer
9. Protocole de livrare

Introducere

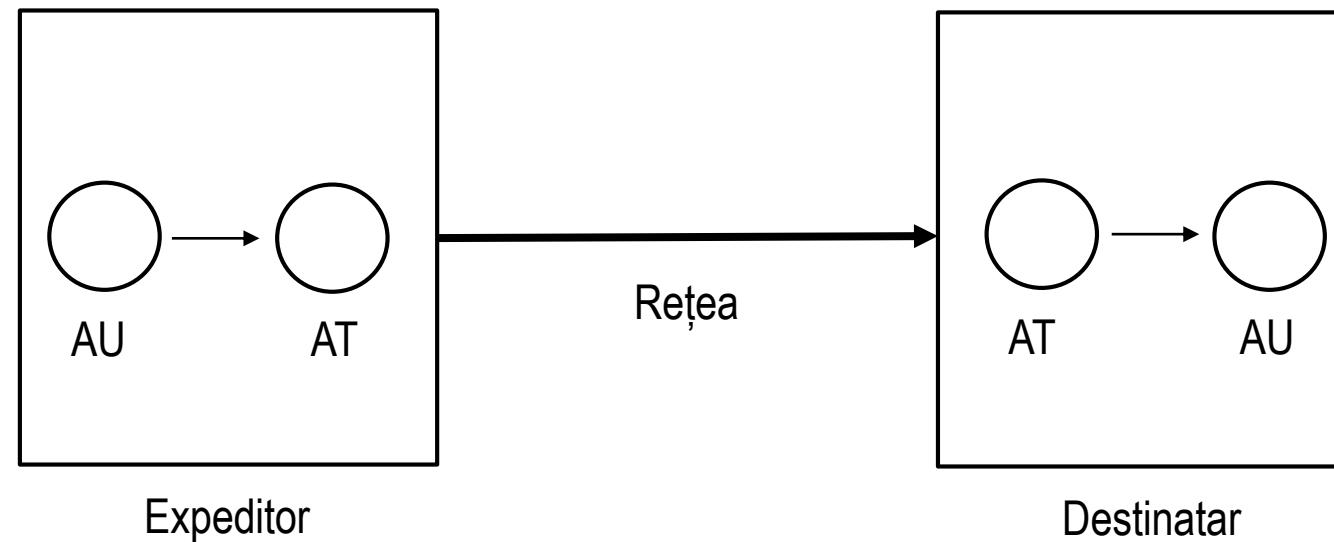
- Poșta electronică (e-mail) facilitează schimbul de mesaje electronice între utilizatorii sistemelor de calcul;
- Utilizat la început în mediul academic a devenit foarte popular în anii 90;
- Primul sistem de poștă electronică constă din protocoale de transfer de fișiere care conțineau pe prima linie adresa receptorului;
- Dezavantajele sistemului inițial de poștă electronică:
 - Trimiterea de mesaje către un grup de persoane era incomodă;
 - Mesajele nu aveau structură internă, fiind greu de procesat;
 - Expeditorul nu avea confirmarea primirii mesajului;
 - Redirectarea automată era greu de realizat;
 - Interfața cu utilizatorul nu se integra cu agentul de transfer;
 - Nu se puteau transmite mesaje combinate text – multimedia.

Istoria sistemului de poștă electronică

- În 1982 este propus protocolul Simple Mail Transfer Protocol (SMTP) (RFC 821) și structura unui mesaj e-mail (RFC 822);
- Revizii minore propuse în RFC 2821 și RFC 2822;
- International Telegraph Union (ITU) propune în 1984 recomandarea X.400.

Arhitectura sistemului

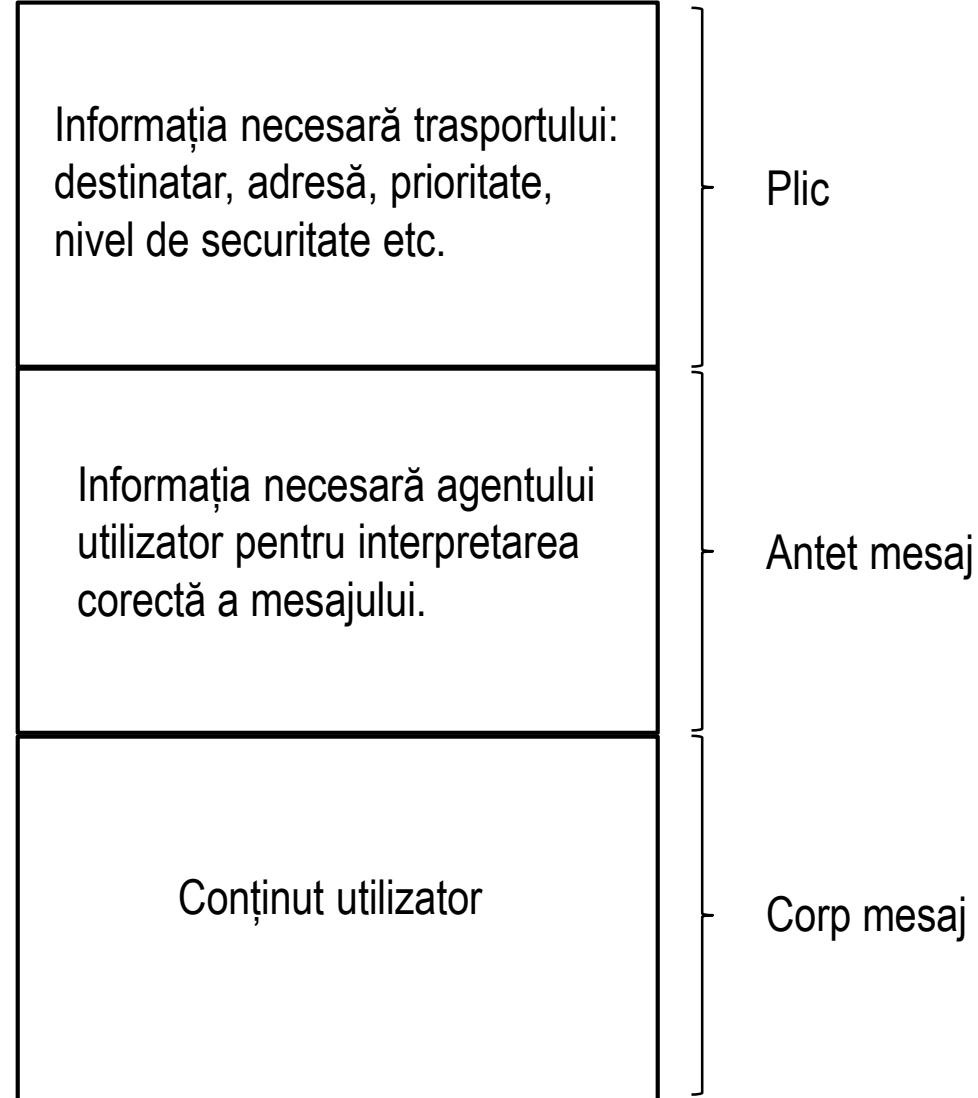
- Sistem format din două componente: **agenți utilizator și agenți de transfer de mesaje**;
- **Agenți utilizator** – aplicație software în format grafic sau text care permite interacțiunea cu sistemul de poștă electronică;
- **Agenți de transfer de mesaje** – aplicații software care rulează în fundal și transferă mesajele de poștă electronică;



Servicii oferite

- **Primare:**
 - Compunerea – procesul de creare a mesajelor și răspunsurilor;
 - Transferul – transmisia mesajului de la expeditor la destinatar;
 - Raportarea – confirmarea transmisiei;
 - Afisarea – interpretarea și afisarea mesajelor în vederea citirii de către utilizator;
 - Dispoziția – prelucrarea finală a mesajului (ștergere, salvare, arhivare etc.).
- **Avansate:**
 - Cutii poștale – recipiente care permit stocarea de mesaje electronice;
 - Liste de poștă – colecții de destinatari.

Structura mesajului



Formatarea mesajului

- RFC 822
 - caractere ASCII
 - plic simplu + câmpuri antet + linie goală + corpul mesajului
 - plicul nu se distinge clar de antet

Antet	Conținut
To:	Adresele de email ale destinatarilor primari
CC:	Adresele de email ale destinatarilor secundari
BCC:	Adresele de email ale destinatarilor privați
From:	Persoana care a creat mesajul
Sender:	Persoana care a transmis mesajul
Received:	Antet introdus de fiecare agent de transfer în drumul către destinație
Return-Path:	Calea de întoarcere la transmițător

Câmpurile care definesc transferul de mesaje în RFC 822

Formatarea mesajului

Antet	Conținut
Date:	Data și momentul de timp la care a fost transmis mesajul
Reply-To:	Adresa de email la care ar trebui transmise răspunsurile
Message-Id:	Identifier unic al mesajului
In-Reply-To:	Identifierul mesajului pentru care este formulat răspunsul
References:	Alți identifieri relevanți
Keywords:	Cuvinte cheie care definesc conținutul
Subject:	Scurtă descriere a mesajului pe un singur rând

Alte câmpuri utilizate în antetul RFC 822

Formatarea mesajului - MIME

MIME – Multipurpose Internet Mail Extensions (RFC 1341, RFC 2045-2049)

- Păstrează formatul definit în RFC 822 și îl extinde
- Restructurează corpul mesajului
- Definește reguli de codificare pentru caracterele non-ASCII

Antet	Conținut
MIME-Version:	Versiunea de MIME definită
Content-Description:	Descrierea mesajului
Content-Id:	Identifier unic
Content-Transfer-Encoding:	Codificarea textului din corpul mesajului
Content-Type:	Tipul (natura) mesajului

Câmpuri antet definite de extensia MIME

Tipuri de codificări MIME

1. Codificare pe 7 biți - ASCII

- dimensiunea maximă a unei linii 1000 caractere

2. Codificare pe 8 biți

3. Codificare binară

4. Codificare în baza 64

- grupurile de 24 de biți sunt împărțite în grupuri de 6 biți care sunt transmise ca și caractere ASCII a-z, A-Z, 0-9, +, /

- se ignoră caracterele pentru new line

5. Codificare afișabilă marcată

- caracterele cu cod mai mare de 127 sunt codificate printr-un = și două caractere hexazecimale

Tipuri și subtipuri MIME

Tip	Subtip	Descriere
Text	Plain	Text neformatat
	Enriched	Text formatat
Imagine	Gif	Imagini fixe în format gif
	Jpeg	Imagini fixe în format jpeg
Audio	Basic	Sunet
Video	Mpeg	Video în format mpeg
Aplicație	Octet-Stream	Fișier binar
	Poscript	Document PostScript
Mesaj	RFC 822	Mesaj RFC 822
	Partial	Mesaj fragmentat
	External-body	Conținut din exteriorul mesajului
Multipart	Mixed	Părți independente în ordine specificată
	Alternative	Același mesaj în formate diferite
	Parallel	Părți care necesită vizualizare simultană
	Digest	Fiecare parte este un mesaj RFC 822 complet

Protocoale de transfer - SMTP

220 mail.aut.upt.ro ESMTP mailer

EHLO intern.aut.upt.ro

250-mail.aut.upt.ro

250-PIPELINING

250-SIZE 31457280

250-ETRN

250-STARTTLS

250-ENHANCEDSTATUSCODES

250-8BITMIME

250 DSN

MAIL FROM: <postmaster@aut.upt.ro>

250 2.1.0 Ok

RCPT TO: <admin-aut@aut.upt.ro>

250 2.1.5 Ok

DATA

354 End data with <CR><LF>.<CR><LF>

From: postmaster@aut.upt.ro

To: admin-aut@aut.upt.ro

MIME-Version: 1.0

Message-Id: 550A68A6.9020605@aut.upt.ro

Subject: Mesaj de test

Content-Type: text/plain; charset=UTF-8; format=flowed

Content-Transfer-Encoding: 7bit

Mesaj de test

.

250 2.0.0 Ok: queued as 24CFD1DE99E

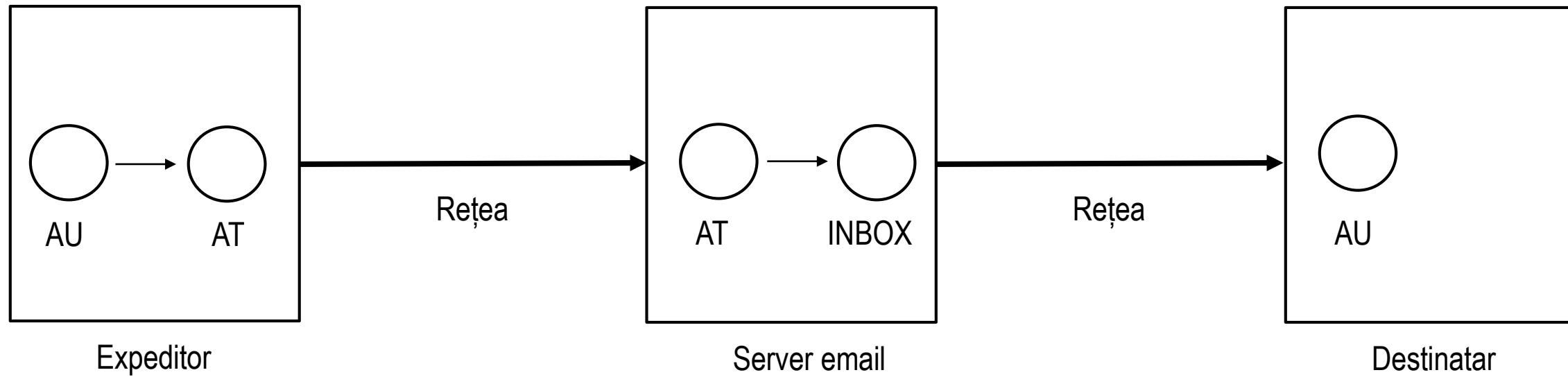
QUIT

Protocole de transfer - SMTP

Cod	Descriere
220	<domeniu> Serviciul pregatit
221	<domeniu> Serviciul inchide canalul
250	Cerere valida, finalizata cu success
251	Utilizator la distanta, retransmitere
252	Utilizatorul nu poate fi validat , dar se incearca transmisia mesajului
354	Incepere transmisie mesaj
421	<domeniu> Serviciul este indisponibil
450	Casuta de email indisponibila
500	Eroare de sintaxa. Comanda incorecta
502	Comanda neimplementata
503	Secventa incorecta de comenzi
504	Parametrul comenzii neimplementat
554	Transfer eronat

Coduri de raspuns

Livrare finală



Post Office Protocol v3

Port standard TCP 110

Protocol simplu în mod text

Implementează trei stări:

1. Autorizare
2. Tranzacționare
3. Actualizare

+OK Dovecot ready.

USER admin-aut

+OK

PASS *****

+OK Logged in.

LIST

1 ...

2 ...

3 ...

.....

RETR 1

.....

DELE 1

QUIT

Internet Message Access Protocol

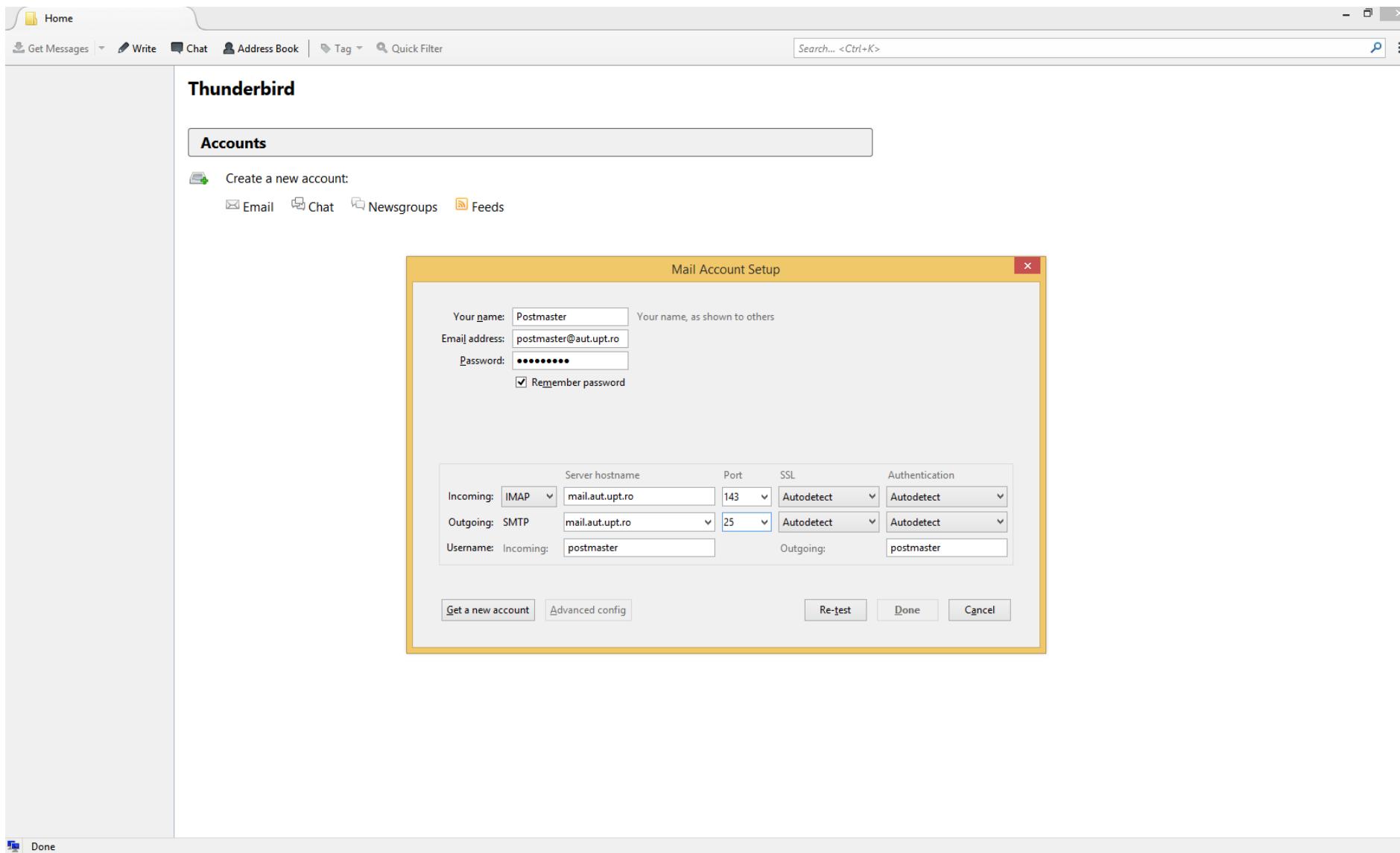
Port standard TCP 143

Protocol complex în mod text

Implementează căsuțe de email multiple

Facilități de indexare, filtrare și căutare

Configurare client email



Nivelul aplicație – World Wide Web

World Wide Web

- sistem distribuit construit peste Internet
- colecție de documente răspândită la nivel global
- arhitectura client – server
- transferul de documente utilizând text formatat (protocol http)
- pagini interconectate prin hiper-legături
- aplicații client cu interfață grafică sau text

World Wide Web

- propus în cadrul CERN în 1989 de către Tim Berners-Lee
- concept funcțional după 1 ½ ani
- Mosaic - primul program de navigare grafic dezvoltat de Marc Andreessen în 1993
- în 1994 este format Consorțiul World Wide Web (W3C) de către CERN și MIT
- Andreessen înfîntăză Netscape Communication Corp.
- în 1998 Netscape Communication Corp. este achiziționată de AOL

Clientul WEB – browser

- utilizează URL-uri (Uniform Resource Locators) pentru accesul la paginile web
- URL-ul este format din trei componente: protocol, nume resursă rețea, denumire pagină
- interpretează pagini web în format HTML
- utilizează extensii – plugin-uri sau aplicații auxiliare – pentru interpretarea MIME

Serverul WEB

- server concurent (fire de execuție multiple)
- memorie tampon

Operații implementate:

1. Rezolvarea numelui paginii de Web cerute.
2. Autentificarea clientului.
3. Verificarea drepturilor de acces ale clientului.
4. Verificarea drepturilor de acces asupra paginii de Web.
5. Verificarea memoriei ascunse.
6. Obținerea paginii cerute, de pe disc.
7. Determinarea tipului MIME ce va fi inclus în răspuns.
8. Rezolvarea altor probleme minore.
9. Transmiterea răspunsului către client.
10. Adăugarea unei înregistrări în jurnalul serverului.

World Wide Web:

- nu menține starea conexiunii
- utilizează cookies pentru transmiterea informației între sesiuni

Cookie – un fișier sau sir de caractere de dimensiune mică (max. 4 KB)

Domeniu	Cale	Conținut	Expiră	Sigur
---------	------	----------	--------	-------

Clasificare:

- persistent
- non-persistent

Hyper Text Transfer Protocol

HTTP

- utilizează protocol de transport TCP

HTTP 1.0

- conexiunea TCP transferă o singură cerere și un singur răspuns

HTTP 1.1

- conexiune TCP persistentă

Metodă	Descriere
GET	Cerere de citire a unei pagini Web
HEAD	Cerere de citire a antetului unei pagini Web
PUT	Cerere de memorare a unei pagini Web
POST	Adaugarea de informații la o resursă
DELETE	Ștergerea unei pagini Web
TRACE	Tipărirea cererii care a sosit
CONNECT	Rezervat
OPTIONS	Interrogarea de opțiuni

Metode de cerere standard pentru HTTP

Hyper Text Transfer Protocol

Cod	Semnificație	Exemple
1xx	Informație	100 = serverul acceptă tratarea cererii de la client
2xx	Succes	200 = cerere reușită; 204 = nu există conținut
3xx	Redirectare	301 = pagină mutată; 304 = pagina din memoria ascunsă este încă validă
4xx	Eroare la client	403 = pagină interzisă; 404 = pagina nu a fost găsită
5xx	Eroare la server	500 = eroare internă la server; 503 = încearcă mai târziu

Grupuri de răspunsuri ale codurilor de stare

Hyper Text Transfer Protocol

Antet	Tip	Descriere
User-Agent	Cerere	Informație asupra programului de navigare și a platformei
Accept	Cerere	Tipul de pagini pe care clientul le poate trata
Accept - Charset	Cerere	Seturile de caractere care sunt acceptabile la client
Accept - Encoding	Cerere	Codificările de pagini pe care clientul le poate trata
Accept - Language	Cerere	Limbajele naturale pe care clientul le poate trata
Host	Cerere	Numele DNS al serverului
Cookie	Cerere	Trimite un cookie setat anterior înapoi la server
Date	Ambele	Data și ora la care mesajul a fost trimis
Upgrade	Ambele	Protocolul la care transmițătorul vrea să comute
Content-Encoding	Răspuns	Cum este codat conținutului (de exemplu, gzip)
Content-Language	Răspuns	Limbajul natural utilizat în pagină
Content-Length	Răspuns	Lungimea paginii în octeți
Last-Modified	Răspuns	Ora și data la care pagina a fost ultima dată modificată

Antete de mesaje HTTP

Surse bibliografice:

Tanenbaum, A. S., Wetherall, D. J., **Computer Networks (5th Edition)**, Pearson, 2010