# ACADEMIA DE STUDII ECONOMICE BUCUREȘTI

## FACULTATEA DE CIBERNETICĂ, STATISTICĂ ȘI INFORMATICĂ ECONOMICĂ


# Rețele de calculatoare

## Suport de curs pentru autoinstruire


**Titular disciplină:**
**Prof. univ. dr. Răzvan Daniel ZOTA**

# CUPRINS:

## Precizări privind alcătuirea manualului de studiu individual

Principalele elementele constitutive, care apar în alcătuirea acestui manualul de studiu individual, corespunzătoare unei unități de studiu proiectată/dezvoltată sunt următoarele:

| | |
|---|---|
|  | **Titlul unității –** corespunde titlului unuia sau mai multor capitole/subcapitole din programa de studiu (conform fișei disciplinei) |
|  | **Cuprinsul unității** - specifică secțiunile principale, subsecțiunile și numărul paginii unde acestea pot fi localizate |

| | |
|---|---|
|  | **Introducere -** secțiune (capitol) care va furniza informații în legătură cu: locul unității de studiu (US) în cadrul disciplinei, obiectivele US formulate în termeni de competențe generale şi specifice US |
|  | ***Durata medie de studiu individual - 2-4 ore*** |
|  | **Obiectivele unităților de studiu** – enunţă competenţele ce urmează a fi dobândite pe parcursul unității de studiu. Dacă, la nivelul programei, competenţele sunt prea general formulate (în consecinţă neevaluabile), se redefinesc competenţele programei prin raportare la conţinuturile unității de învăţare. Aceasta se face astfel: fiind dată competenţa Q din programă şi conţinuturile C ale unității de învăţare, se formulează competenţe specifice pentru unitatea de învăţare astfel încât acestea să fie evaluabile (pe parcurs şi la sfârşitul unității de învăţare).<br>Întrebarea la care trebuie răspuns este: *La ce foloseşte cursantului conţinutul unității de învăţare*?<br>Răspunsul se dă în termeni operaţionali/procedurali/contextuali, după caz şi în funcţie de disciplină. |
|  | **Conţinutul *unității de învăţare*** *–(sinteze teoretice, exemple)* redactarea textului propriu-zis va ţine cont de interacţiunea competenţe-conţinuturi. Textul va fi structurat astfel încât cantitatea de informaţie nouă pe unitatea de învăţare să fie raţională, echilibrat distribuită şi asimilabilă. |
|  | **Sinteza unității de studiu -** Rezumatul sau Sinteza ideilor, noţiunilor şi conceptelor dezbătute în cadrul unității de învăţare. |
|  | **Concepte și termeni de reținut -** Definiţii şi terminologie |
|  | ***Întrebări de control și teme de dezbatere*** |
|  | **Îndrumar pentru autoevaluare sau *Testele de autoevaluare*** reprezintă ***exercițiile*** sau rezolvarea unor ***probleme***. Acestea solicită studentul să efectueze o activitate mai complexă decât simpla rezolvare a unui test de autoevaluare. De exemplu, studentului i se poate solicita să scrie un paragraf prin care descrie opinia personală asupra unui subiect studiat şi analizat. De asemenea, exercițiile pot solicita practicarea unor deprinderi necesare formării studentului ca viitor specialist |

| | |
|---|---|
|  | **Bibliografie obligatorie -** va enunța o listă minimală pe care cursantul trebuie să o parcurgă pentru studiul unității de învățare.<br>Bibliografia va fi prezentată la sfârșitul fiecărei unități de studiu şi va constitui un decupaj din bibliografia generală de la sfârșitul manualului. |

# INTRODUCERE

*O rețea de calculatoare reprezintă o colecție de diverse echipamente ce comunică între ele. Aceste echipamente pot fi: computere, telefoane inteligente, rutere, switch-uri, imprimante de rețea, etc.*

## Obiectivele manualului de studiu

Obiectivele principale ale manualului de studiu individual constau în:

- Însușirea limbajului din domeniul rețelelor de calculatoare;
- Însușirea noțiunilor principale pentru a putea proiecta, analiza, implementa și depana o rețea de calculatoare;
- Formarea abilității de a folosi sursele de informații existente pe web cu privire la dezvoltarea domeniului rețelelor de calculatoare;
- Folosirea cunoștințelor din manualul de studiu individual, bibliografie și seminarii pentru elaborarea unui proiect în domeniul rețelelor de calculatoare.

## Competențe conferite

1. **Cunoaștere și înțelegere** ( *cunoașterea și înțelegerea adecvată a noțiunilor specifice disciplinei)*
   - ✓ *Cunoașterea și înțelegerea noțiunilor și conceptelor cu care operează domeniul rețelelor de calculatoare;*
   - ✓ *Folosirea corectă a termenilor de specialitate din domeniul rețelelor;*
   - ✓ *Folosirea competentă a informațiile cu privire la caracteristicile unei rețele de calculatoare.*

2. **Explicare și interpretare** *(explicarea și interpretarea unor idei, proiecte, procese, precum și a conținuturilor teoretice și practice ale disciplinei)*
   - ✓ *Organizarea și funcționarea rețelelor de calculatoare;*
   - ✓ *Organizarea procesului de învățare în domeniul rețelelor de calculatoare într-o viziune sistemică;*
   - ✓ *Realizarea unui studiu de caz cu privire la proiectarea unei rețele de calculatoare.*

3. **Instrumental aplicative** *( proiectarea, conducerea și evaluarea activităților practice specifice; utilizarea unor metode, tehnici și instrumente de investigare și de aplicare)*
   - ✓ *Capacitatea de a transpune în practică a cunoștințelor obținute din bibliografie, seminarii, proiecte și referate ;*
   - ✓ *Abilități de cercetare, creativitate, competențe în rezolvarea studiilor de caz;*
   - ✓ *Cunoașterea modului de planificare a unei rețele locale de calculatoare.*

4. **Atitudinale** *( manifestarea unei atitudini pozitive și responsabile față de domeniul științific, cultivarea unui mediu științific centrat pe valori și relații democratice, promovarea unui sistem de valori culturale, morale și civice, valorificarea optimă și creativă a propriului potențial în activitățile științifice, implicarea în dezvoltarea instituțională și în promovarea inovațiilor științifice, angajarea în relații de parteneriat cu alte persoane, instituții cu responsabilități similare, participarea la propria dezvoltare personală)*

- ✓ *Reacții pozitive la disciplina universitară în general și față de exigențele disciplinei Rețele de calculatoare în particular*
- ✓ *Implicarea studenților în activități științifice în legătură cu disciplina studiată pentru participarea la sesiunile științifice ale universității;*
- ✓ *Capacitatea de a avea un comportament etic în relațiile cu colegii și cadrele didactice;*
- ✓ *Abilitatea de a colabora cu specialiștii din alte domenii.*

## Resurse și mijloace de lucru

Disciplina ***Rețele de calculatoare*** dispune de un Suport de curs pentru autoinstruire pentru studenți, precum și de materiale prezentate pe **online.ase.ro** sub formă de sinteze, lecții și unități de studiu, studii de caz și aplicați, necesare întregirii cunoștințelor practice și teoretice în domeniul rețelelor de calculatoare.
La această disciplină, în timpul *activităților tutoriale* sunt folosite echipamente audio-vizuale, metode interactive și participative de antrenare a studenților pentru conceptualizarea și vizualizarea practică a disciplinei.

## Structura manualului de studiu individual

Unitățile de studiu individual sunt proiectate corespunzător obiectivelor prevăzute în Fișa disciplinei de ***Rețele de calculatoare,*** fiind compuse din 7 unități de studiu, astfel:

| Unitatea de studiu | Tematica | Nr. ore |
|---|---|---|
| *Unitatea de studiu 1.* | Internetul și rețelele de calculatoare | 4 ore |
| *Unitatea de studiu 2.* | Modele ierarhice pentru studiul rețelelor de calculatoare | 4 ore |
| *Unitatea de studiu 3.* | Nivelul legătură de date | 4 ore |
| *Unitatea de studiu 4.* | Nivelul rețea | 4 ore |
| *Unitatea de studiu 5.* | Caracteristicile protocolului IPv6 | 4 ore |
| *Unitatea de studiu 6.* | Nivelul transport | 4 ore |
| *Unitatea de studiu 7.* | Nivelul aplicație | 4 ore |

## Teste de control

Desfășurarea testelor de control se va derula conform Calendarului Disciplinei. Subiectele punctuale vor fi prezentate studenților la momentul activităților tutoriale.

## Bibliografie obligatorie:


1. Răzvan Zota – Rețele de calculatoare în era Internet, Editura Economică, 2002
2. Răzvan Daniel Zota – Rețele de calculatoare, Editura ASE, 2013


### Bibliografie suplimentară:


1. J. Kurose, K. Ross, *Computer Networking*, Ed. Addison Wesley, USA, 2001.


## Metoda de evaluare:

Examenul final la disciplină *Rețele de calculatoare* este un examen scris.
Subiectele de examinare conțin întrebări de tip grilă și întrebări cu răspuns scurt.

# UNITATEA DE STUDIU 1. Internetul şi reţelele de calculatoare



## 1.1 Ce este Internetul?

În zilele noastre, termenul „Internet" nu mai reprezintă o noutate, ci, mai degrabă, un termen omniprezent în vocabularul tuturor. Într-o lume în care mobilitatea, instrumentele de lucru colaborativ şi reţelele sociale sunt lucruri obişnuite, reţeaua Internet reprezintă o adevărată „coloană vertebrală" pentru multe dintre activităţile zilnice. În cele ce urmează vom încerca să clarificăm definiţia Internetului, având în vedere că până şi mulţi utilizatori împătimiţi ai săi nu ştiu să facă deosebirea între Internet şi Web sau între serviciul de poştă electronică şi cel de transfer de fişiere.

Trebuie să lămurim de la început că nu se poate da o definiţie complexă a termenului de Internet în câteva rânduri. Având însă câteva noţiuni de bază şi o serie de caracteristici cunoscute, ne putem face o privire de ansamblu asupra concepţiei de Internet.

În primul rând, Internetul este o reţea de calculatoare (este, de fapt, o reţea de reţele) la nivel mondial prin intermediul cărora sunt interconectate milioane de echipamente de calcul (aici sunt incluse şi calculatoarele personale) din întreaga lume. În cel mai simplu sens, o reţea de calculatoare reprezintă o colecţie de calculatoare interconectate, ce sunt capabile să schimbe informaţie între ele [Tanenbaum, 1996].

Pe de altă parte, Internetul este denumirea celei mai vaste grupări de surse de informaţie din lume. Reţeaua de care vorbeam mai înainte are o dimensiune extinsă la mărimea planetei noastre şi cuprinde o cantitate inimaginabilă de resurse fizice, logice, informaţionale.

Printre echipamentele interconectate se găsesc: calculatoare personale, staţii de lucru Unix, servere de Web sau de e-mail, laptop-uri, pagere, telefoane mobile, tablete, etc. De curând au fost conectate la Internet şi dispozitive electrocasnice, cum ar fi frigiderul sau cuptorul cu microunde. Se prevede că în viitor multe dintre echipamentele electrocasnice vor dispune de conexiune Internet. Toate aceste echipamente sunt denumite sisteme gazdă (hosts sau end systems). Aplicaţiile Internet care ne sunt tuturor foarte familiare (poşta electronică, web-ul, Facebook sau Twitter) sunt de fapt, aplicaţii de reţea ce rulează pe aceste sisteme gazdă.

Pentru a comunica între ele, sistemele gazdă folosesc aşa numitele protocoale pentru controlul transmiterii, recepţiei şi corecţiei informaţiilor care circulă prin Internet. Dintre aceste protocoale, TCP (Transmission Control Protocol) şi IP (Internet Protocol) sunt cele mai importante protocoale folosite în Internet. Aşa numita stivă de protocoale TCP/IP nu conţine doar aceste două protocoale (TCP şi IP) ci şi alte protocoale, dar acestea două sunt cele de bază. De asemenea, pentru asigurarea conexiunii între ele, sistemele gazdă folosesc legături de comunicaţie ce constau din diverse tipuri de cabluri, printre care cablu coaxial, torsadat, fibră optică sau pot fi conexiuni fără fir, prin unde radio, de exemplu. Una dintre caracteristicile importante ale acestor legături este viteza teoretică de transfer a datelor care

este denumită lăţime de bandă (bandwidth) şi care se exprimă în biţi sau multipli ai acestora pe secundă (1 Mb/s = 1.000 biţi/s, 1 Gb/s = 1.000.000 biţi/s, 10 Gb/s = 10.000.000 biţi/s etc.).

Sistemele gazdă nu sunt interconectate direct între ele, ci prin intermediul unor dispozitive intermediare denumite rutere. Pe scurt, un ruter este un dispozitiv care preia informaţia ce ajunge la el prin intermediul uneia dintre legăturile (de intrare) de comunicaţie şi o trimite mai departe pe o altă legătură (de ieşire) de comunicaţie. Formatul informaţiilor care sunt recepţionate şi transmise mai departe între rutere şi sistemele gazdă sunt precizate de protocolul IP. Acest protocol reprezintă "limbajul universal" al Internetului şi de aceea se mai numeşte şi "Internet dial tone". Drumul pe care îl parcurg informaţiile de la transmiţător la receptor poartă numele de rută (route / path) în reţea.

Modalitatea de stabilire a unei conexiuni în Internet (pentru a putea transmite informaţii de la un transmiţător la un receptor) se bazează pe o tehnică denumită comutare de pachete, care permite mai multor sisteme să comunice pe o rută (sau o porţiune dintr-o rută) Internet, în acelaşi timp. Topologia Internetului (structura sistemelor conectate la Internet) este ierarhizată în modul următor: la bază sunt sistemele gazdă conectate la un ISP (Internet Service Provider - Furnizor de Servicii Internet) local prin intermediul unor reţele de acces, furnizorii locali sunt conectaţi la nişte furnizori naţionali sau internaţionali, iar aceştia din urmă sunt conectaţi împreună la cel mai înalt nivel din această ierarhie.

## 1.2 Obiectivele și competențele unității de studiu

*Obiectivele unității de studiu*:

- identificarea principalelor etape al apariției și dezvoltării Internetului;
- definirea conceptelor de baza cu care se operează în studiul rețelelor de calculatoare;
- prezentarea generală a rețelelor de calculatoare și a Internetului.

*Competențele unității de studiu*:

- studenții vor putea să definească concepte de bază cu care operează domeniul rețelelor de calculatoare;
- studenții vor cunoaște detalii legate de proiectarea, analiza, implementarea și depanarea rețelelor de calculatoare.

***Durata medie de studiu individual alocat unității:*** 4 ore

## 1.3 Conținutul unității de studiu

### 1.3.1 Definiția unui protocol de rețea

Unul din termenele cele mai folosite atunci când este vorba de o reţea de calculatoare sau de Internet este termenul de "protocol". Vom prezenta în continuare o definiţie şi câteva exemple pentru a putea identifica un protocol.

Probabil că cea mai bună modalitate de a înţelege noţiunea de protocol este aceea de a considera pentru început o serie de analogii cu intercomunicarea din lumea umană. Să considerăm exemplul în care întrebăm pe cineva unde se află o anume stradă (Figura 1.1).

Bunele maniere (protocolul uman) ne fac să spunem întâi "Bună ziua!" pentru a

începe comunicarea cu o altă persoană. Răspunsul ar trebui să fie, desigur, tot "Bună ziua!", ca o confirmare a faptului că este acceptată comunicarea. Interpretarea răspunsului ca un accept al comunicării ne permite acum să formulăm întrebarea care ne interesează. Dacă răspunsul iniţial al persoanei căreia îi adresăm "Bună ziua!" ar fi fost "Lasă-mă în pace, am treabă!" sau ceva asemănător, atunci ar fi însemnat că nu există posibilitatea comunicării. În acest caz, nu mai are rost să formulăm întrebarea al cărei răspuns dorim să-l aflăm. Uneori este posibil să nu primim nici un răspuns la o întrebare, caz în care de regulă renunţăm a mai repeta întrebarea.

Regulile intercomunicării umane (protocolul uman) sunt astfel reprezentate de mesajele pe care le trimitem şi de acţiunile specifice pe care le întreprindem corespunzătoare răspunsului primit de la interlocutor sau producerii altor evenimente. Mesajele transmise şi cele recepţionate joacă un rol fundamental în cazul protocoalelor umane; dacă o persoană are obiceiuri diferite sau foloseşte un limbaj străin altei persoane, atunci protocoalele diferite nu vor permite intercomunicarea între respectivele persoane. Acelaşi lucru este valabil şi în cazul comunicării între entităţile dintr-o reţea de calculatoare. Pentru a putea comunica, respectivele entităţi trebuie să folosească (să ruleze) acelaşi protocol de reţea.

Un protocol de reţea este asemănător unui protocol uman, excepţie făcând obiectele comunicării: în loc să avem de-a face cu oameni, avem de-a face cu componente hardware sau software ale reţelei. Toate activităţile dintr-o reţea de calculatoare (deci şi din Internet) sunt bazate pe funcţionarea unui anumit set de protocoale. De exemplu, comunicarea dintre două calculatoare în reţea se face prin protocoale implementate în hardware la nivelul plăcii de reţea pentru controlul fluxurilor de biţi transmişi prin intermediul suportului fizic; protocoalele de control al congestiilor au grijă să controleze viteza de transmitere a datelor între un transmiţător şi un receptor iar protocoalele de poştă electronică guvernează modalitatea de transmitere şi de recepţie a mesajelor de tip e-mail.

În figura 1.1 este prezentat cazul în care un calculator face o cerere unui server Web (asta se întâmplă în momentul în care scriem adresa web în fereastra browser-ului), se primeşte un răspuns afirmativ de conexiune din partea serverului şi apoi calculatorul foloseşte un mesaj de tip "GET" pentru a recepţiona pagina respectivă. În cele din urmă, serverul returnează conţinutul fişierului calculatorului care a făcut cererea.

Ca urmare a analogiei cu comportamentul uman, putem da următoarea definiţie a protocolului: *un protocol defineşte formatul şi ordinea mesajelor schimbate între două sau mai multe entităţi ce comunică între ele, precum şi acţiunile ce sunt întreprinse odată cu transmiterea sau recepţia unui mesaj sau a unui alt eveniment*.

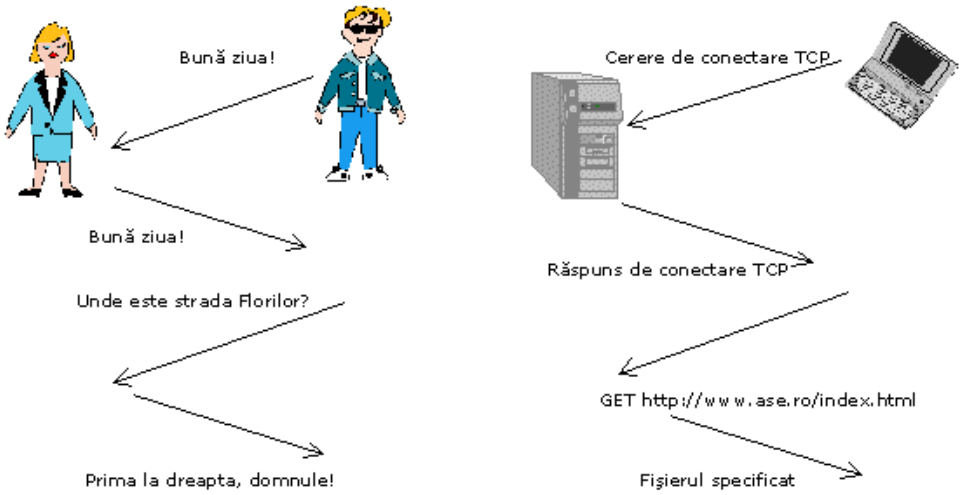


Figura 1.1 Analogie între un protocol uman şi un protocol de reţea.

### 1.3.2 Naşterea "limbajului" TCP/IP

Naşterea Internetului a dus în 1973 la începutul dezvoltării stivei de protocoale TCP/IP, care se dorea a fi o colecţie de protocoale de reţea bazate pe software care să permită *oricărui* sistem să se conecteze cu *orice alt sistem*, folosind *orice topologie* de reţea. Cinci ani mai târziu, în 1978, era gata versiunea 4 IP, adică aceeaşi versiune care încă se mai foloseşte şi astăzi, dar care, treptat, este înlocuită de noua versiune IPv6 (în care adresele de reţea sunt stocate pe 128 de biţi). Imediat după aceea au început să apară o serie de semnale pozitive în legătură cu recunoaşterea TCP/IP: Universitatea Berkeley din California a încorporat stiva de protocoale TCP/IP în versiunea proprie de UNIX - FreeBSD (distribuită gratis) ce avea să devină cel mai folosit sistem de operare în comunităţile academice şi de cercetare.

Introducerea la scară largă a suitei de protocoale TCP/IP a produs o serie de schimbări majore în lumea reţelelor de calculatoare. În primul rând, topologia de bază a unei reţele era concentrată pe un nod central, în care fiecare sistem ataşat trimitea datele unui nod central (pe post de dispecer) pentru a fi procesate. Cu alte cuvinte, utilizatorii din reţea nu aveau independenţă în lucru, orice procesare, tipărire la imprimantă, etc. trebuind să treacă pe la nodul central.

Odată cu introducerea TCP/IP, lucrurile s-au schimbat: s-a introdus "descentralizarea", astfel încât fiecare echipament din reţea era tratat independent şi complet funcţional, fără a mai depinde de un nod central. Comunicarea cu alt echipament din reţea se putea face acum direct, fără să se comunice mai întâi cu nodul central. Reţelele bazate pe protocolul IP sunt oarecum anarhice, fiecare echipament acţionând pe cont propriu ca o unitate autonomă, responsabilă pentru serviciile de reţea proprii [Hall 2000]. Această concepţie arhitecturală a permis partajarea aplicaţiilor şi a resurselor la scară largă, având în vedere că un model centralizat top-down nu era viabil în cazul existenţei a milioane de echipamente larg răspândite. În plus, acest model oferea siguranţă în exploatare în cazul "căderii" unei componente din reţea, în contrast cu modelul centralizat în care toată funcţionarea se oprea în cazul "căderii" nodului central.

## 1.4 Îndrumar pentru autoverificare

### 1.4.1 Sinteza unității de studiu 1

În lumea de astăzi, Internetul face parte integrantă din viața noastră de zi cu zi. În zilele noastre, termenul „Internet" nu mai reprezintă o noutate, ci, mai degrabă, un termen omniprezent în vocabularul tuturor. Într-o lume în care mobilitatea, instrumentele de lucru colaborativ şi reţelele sociale sunt lucruri obişnuite, reţeaua Internet reprezintă o adevărată „coloană vertebrală" pentru multe dintre activităţile zilnice.

### 1.4.2 Concepte și termeni de reținut

| | |
|---|---|
| *Internet* | *Rețea de calculatoare* |
| *Protocol de rețea* | *Clienți de rețea* |

| | |
|---|---|
| *Transmițător* | *Receptor* |
| *Servere* | *Calculatoare gazdă* |
| *Structura de bază a Internetului* | *Rețea fiabilă* |

### 1.4.3 Întrebări pentru autoverificare

***Întrebarea 1.*** Ce este Internetul?

a) Oferă acces la rețea pentru echipamente mobile
b) Oferă conexiuni prin intermediul unor rețele interconectate la nivel global.
c) Este o rețea privată a unei organizații ce are conexiuni locale și globale.
d) Este o rețea bazată pe tehnologia Ethernet.

*Răspuns:b*

***Întrebarea 2.*** Un utilizator dorește să acceseze rețeaua organizației de la distanță, în mod securizat. Ce tehnologie specifică de rețea permite acest lucru?

a) VPN
b) ACL
c) BYOD
d) IPS

*Răspuns:a*

***Întrebarea 3.*** Ce caracteristică a unei rețele permite să crească pentru a oferi suport pentru noi utilizatori și aplicații, fără a avea un impact negativ asupra performanțelor serviciilor oferite utilizatorilor existenți?

a) Calitatea serviciilor (QoS)
b) Scalabilitatea
c) Integritatea
d) Toleranța la erori

*Răspuns:b*

### Întrebări de control și teme de dezbatere

1. Cum definiți o rețea de calculatoare?
2. Care sunt asemănările și deosebirile între modelele arhitecturale ISO-OSI și TCP/IP?
3. Cum se face împărțirea în sub-rețele în cazul protocolului IPv4?
4. Care sunt beneficiile aduse de introducerea protocolului IPv6?
5. Explicați funcționarea serviciului DNS.
6. Ce topologii de rețele locale (LAN) de calculatoare cunoașteți?
7. Ce topologii de rețele de arie largă (WAN) de calculatoare cunoașteți?
8. Detaliați funcționarea protocolului DHCP. Care sunt principalele beneficii ale acestuia?
9. Dați exemple de protocoale și servicii de tip P2P.
10. Care sunt caracteristicile de bază ale protocolului TCP?

11. Care sunt caracteristicile de bază ale protocolului UDP?
12. Care sunt principalele caracteristici/probleme ale rețelelor de calculatoare din zilele noastre?

### 1.4.4 Bibliografie obligatorie


- Răzvan Daniel Zota, Rețele de calculatoare, capitolul 1, Ed. ASE, 2013.

# UNITATEA DE STUDIU 2. Modele ierarhice pentru studiul rețelelor de calculatoare



## 2.1 Introducere

Primele reţele de calculatoare din lume erau formate, de regulă, din calculatoare ce proveneau de la acelaşi producător, neexistând posibilitatea de a face să coopereze computere şi echipamente de reţea produse de firme diferite. Pe măsură ce numărul de calculatoare a crescut şi complexitatea reţelelor s-a mărit, a apărut necesitatea de a putea fi integrate împreună soluţii provenite de la mai mulţi fabricanţi de computere şi tehnologii de reţea. La sfârşitul anilor 1970, Organizaţia Internaţională pentru Standardizare (ISO – International Organization for Standardization) a început dezvoltarea primului model arhitectural – denumit OSI (Open Systems Interconnection) pe baza căruia să se rezolve această necesitate. ISO este cea mai mare organizaţie din lume ce dezvoltă standarde pentru diverse produse şi servicii.

ISO nu este un acronim al numelui întreg al organizaţiei, ci mai degrabă este bazat pe cuvântul grecesc „isos" care înseamnă egal. Organizaţia Internaţională pentru Standardizare a ales acest termen pentru a-şi afirma poziţia de egalitate pentru toate ţările din lume. În lumea IT există numeroase standarde ISO foarte cunoscute. Spre exemplu, extensia de fişier ISO este folosită pentru imaginile de CD pentru a semnifica faptul că se foloseşte standardul ISO 9660 pentru sistemul de fişiere de pe CD.

Modelul de referinţă OSI s-a impus ca un standard bine cunoscut în lumea reţelelor de calculatoare, iar pentru că organizaţia ISO a conceput acest model, numele complet al său este modelul ISO-OSI. ISO a făcut public acest model în 1984 în dorinţa de a oferi un cadru de referinţă (împărţit pe mai multe nivele) pentru protocoalele de reţea. Acest model se doreşte să fie un ajutor pentru ca producătorii de calculatoare şi de echipamente de reţea să aibă produse interoperabile cu cele similare ale altor producători. Modelul ISO-OSI este modelul arhitectural de bază al reţelelor de calculatoare, descriind modul în care aplicaţiile de pe un computer comunică prin intermediul mediilor de reţea cu aplicaţiilor de pe un alt computer aflat în reţea.

În literatura de specialitate există şi alte modele arhitecturale de reţea, precum modelul TCP/IP şi modelul ierarhic Cisco. Toate aceste modele au o caracteristică principală comună, în sensul că abordarea problematicii reţelelor se face pe nivele. Având în vedere că modelul ierarhic Cisco este un model particular elaborat de către compania respectivă, în continuare vom prezenta doar modelele OSI şi TCP/IP.

## 2.2 Obiectivele și competențele unității de studiu

*Obiectivele unității de studiu*:

- identificarea principalelor etape al apariției și dezvoltării Internetului;
- definirea conceptelor de baza cu care se operează în studiul rețelelor de calculatoare;
- prezentarea generală a rețelelor de calculatoare și a Internetului.

*Competențele unității de studiu*:

- studenții vor putea să definească concepte de bază cu care operează domeniul rețelelor de calculatoare;
- studenții vor cunoaște detalii legate de proiectarea, analiza, implementarea și depanarea rețelelor de calculatoare.

***Durata medie de studiu individual alocat unității:*** 4 ore

## 2.3 Conținutul unității de studiu

### 2.3.1 Avantajele unui model ierarhic

Pentru a înţelege mai bine de ce a apărut necesitatea existenţei unui model după care să fie proiectate, dezvoltate, analizate şi depanate reţelele de calculatoare trebuie să definim noţiunea de *flux informaţional*. Considerând exemplul a două calculatoare aflate într-o reţea (Figura 2.1), *comunicarea* dintre acestea se face pe baza unui schimb de date; această deplasare a datelor de la calculatorul sursă la cel destinaţie poartă numele de *flux de date* sau, pe scurt, *flux*.

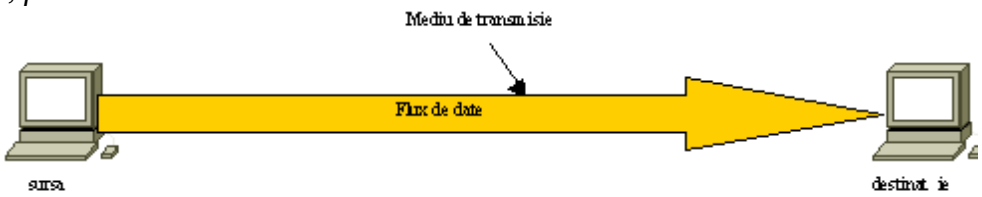



Figura 2.1 Fluxul de date între sursă şi destinaţie

Putem face analogii ale scurgerii fluxului de date cu numeroase exemple din viaţa de zi cu zi: traficul maşinilor pe stradă, scurgerea apei într-o conductă sau drumul parcurs de o scrisoare de la expeditor la destinatar. În toate aceste exemple este vorba de o mişcare a unor obiecte (fie că este vorba de maşini, apă sau scrisori) dintr-un loc în altul, iar această mişcare reprezintă un *flux*. În legătură cu fluxul de date dintr-o reţea de calculatoare, apar o serie de întrebări care trebuie lămurite:

- Care este fluxul?
- Care sunt diferitele forme de flux?
- Ce reguli guvernează acest flux?
- Unde apare acest flux?

Pentru a clarifica răspunsurile la aceste întrebări ce apar când este forma de fluxul de

date dintr-o reţea de calculatoare s-a recurs la divizarea problemei de comunicaţie pe mai multe nivele, lucru pe care îl fac şi modelele arhitecturale de reţea. Divizarea problematicii comunicaţiei pe mai multe nivele are următoarele avantaje:

- Se împarte problema comunicaţiei din reţea în piese mai mici şi mai simple, deci mai uşor de analizat.
- Oamenii pot discuta şi învăţa mai uşor detalii ale specificaţiilor unui protocol de reţea.
- Se doreşte standardizarea componentele de reţea pentru a putea permite dezvoltare şi suport multi-producător.
- Standardizarea interfeţelor facilitează concepţia şi construcţia modulară, astfel încât diferite produse pot oferi funcţionalităţi doar pentru anumite nivele (spre exemplu, ruterele oferă funcţii pentru nivelele 1-3) iar unele produse pot oferi doar părţi ale funcţiilor unui protocol (spre exemplu, aplicaţia de e-mail Eudora care oferă suport pentru nivelul aplicaţie TCP/IP).
- Permite diferitor tipuri de hardware şi software din reţea să comunice între ele.
- Un nivel foloseşte serviciile nivelului imediat inferior; astfel, memorarea funcţiilor nivelelor se face mai uşor.

Este o modalitate de prevenire a faptului că o modificare ce apare la un nivel să afecteze celelalte nivele, astfel încât să se dezvolte mai rapid.

Naşterea Internetului a dus în 1973 la începutul dezvoltării stivei de protocoale TCP/IP, care se dorea a fi o colecţie de protocoale de reţea bazate pe software care să permită *oricărui* sistem să se conecteze cu *orice alt sistem,* folosind *orice topologie* de reţea. Cinci ani mai târziu, în 1978, era gata versiunea 4 IP, adică aceeaşi versiune care încă se mai foloseşte şi astăzi, dar care, treptat, este înlocuită de noua versiune IPv6 (în care adresele de reţea sunt stocate pe 128 de biţi). Imediat după aceea au început să apară o serie de semnale pozitive în legătură cu recunoaşterea TCP/IP: Universitatea Berkeley din California a încorporat stiva de protocoale TCP/IP în versiunea proprie de UNIX - FreeBSD (distribuită gratis) ce avea să devină cel mai folosit sistem de operare în comunităţile academice şi de cercetare.

Introducerea la scară largă a suitei de protocoale TCP/IP a produs o serie de schimbări majore în lumea reţelelor de calculatoare. În primul rând, topologia de bază a unei reţele era concentrată pe un nod central, în care fiecare sistem ataşat trimitea datele unui nod central (pe post de dispecer) pentru a fi procesate. Cu alte cuvinte, utilizatorii din reţea nu aveau independenţă în lucru, orice procesare, tipărire la imprimantă, etc. trebuind să treacă pe la nodul central.

Odată cu introducerea TCP/IP, lucrurile s-au schimbat: s-a introdus "descentralizarea", astfel încât fiecare echipament din reţea era tratat independent şi complet funcţional, fără a mai depinde de un nod central. Comunicarea cu alt echipament din reţea se putea face acum direct, fără să se comunice mai întâi cu nodul central. Reţelele bazate pe protocolul IP sunt oarecum anarhice, fiecare echipament acţionând pe cont propriu ca o unitate autonomă, responsabilă pentru serviciile de reţea proprii [Hall 2000]. Această concepţie arhitecturală a permis partajarea aplicaţiilor şi a resurselor la scară largă, având în vedere că un model centralizat top-down nu era viabil în cazul existenţei a milioane de echipamente larg răspândite. În plus, acest model oferea siguranţă în exploatare în cazul "căderii" unei componente din reţea, în contrast cu modelul centralizat în care toată funcţionarea se oprea în cazul "căderii" nodului central.

### 2.3.2 Modelul ISO-OSI

Lansat oficial în 1984, modelul ISO-OSI reprezintă modelul arhitectural principal pe baza căruia reţelele de calculatoare sunt proiectate, analizate, dezvoltate, implementate sau

depanate. Acest model este conceput să trateze reţelele de calculatoare pe mai multe nivele, făcând astfel ca problemele comunicaţiei (fluxurile din reţea) să fie divizate în probleme mai simple şi mai uşor de analizat, corespunzătoare unui nivel din reţea. Cu ajutorul modelului OSI se îmbunătăţeşte transferul datelor dintre nodurile unei reţele, având în vedere că una dintre caracteristicile sale principale este aceea de a asista modalitatea de transfer a datelor între două sisteme terminale din reţea.

Modelul OSI este practic un set de principii de bază pe care dezvoltatorii de aplicaţii de reţea îl pot folosi pentru a crea şi implementa aceste aplicaţii. De asemenea, modelul oferă cadrul specific pentru *crearea şi implementarea standardelor de reţea, a echipamentelor şi a schemelor de interconectare în reţea* [Lammle 2000]. Modelul OSI descrie modalitatea în care datele şi informaţiile din reţea sunt transmise de la o aplicaţie de pe un computer către o altă aplicaţie de pe alt computer; acest lucru se face folosind o abordare pe 7 nivele. Cele 7 nivele ale modelului OSI sunt împărţite în două grupuri. Primul grup, format din cele trei nivele superioare defineşte modul de comunicare între aplicaţiile de pe staţiile terminale din reţea şi modul de comunicare cu utilizatorii. Cel de-al doilea grup, format din cele 4 nivele inferioare defineşte modul de transmitere a datelor de la o sursă la o destinaţie. În tabelul 2.1 sunt prezentate cele 7 nivele ale modelului OSI împreună cu câteva caracteristici principale ale fiecărui nivel şi câteva exemple de protocoale ce activează la aceste nivele. Unele protocoale sunt definite pe mai multe nivele din modelul OSI; spre exemplu, NFS (Network File System) implementează elemente din cele trei nivele superioare (aplicaţie, prezentare şi sesiune) iar standardele Ethernet, IEEE 802.3 şi 802.5 cuprind detalii legate de nivelele fizic şi legătură de date.

Tabelul 2.1 Nivelele modelului ISO-OSI

| Denumirea nivelului | Scurtă descriere funcţională | Exemple de protocoale |
|---|---|---|
| **APLICAŢIE** | Interfaţa cu utilizatorul | Telnet, HTTP, FTP, browsere WWW, NFS, SMTP gateways, SNMP |
| **PREZENTARE** | Modalitatea de prezentare a datelor | JPEG, ASCII, EBCDIC, TIFF, GIF, PICT, MPEG, MIDI |
| **SESIUNE** | Separă datele diferitelor aplicaţii | RPC, SQL, NFS, nume NetBios, AppleTalk ASP |
| **TRANSPORT** | Asigură livrarea datelor la destinaţie<br>Asigură corecţia datelor înaintea transmiterii | TCP, UDP, SPX |
| **REŢEA** | Se ocupă cu adresarea logică pe care ruterele o utilizează pentru determinarea rutei până la destinaţie | IP, IPX, AppleTalk |
| **LEGĂTURĂ DE DATE** | Pachetele de date sunt transformate în octeţi şi octeţii în cadre<br>Oferă acces la mediu prin utilizarea adreselor MAC<br>Asigură detecţia erorilor | IEEE 802.3/802.2, HDLC, Frame Relay, PPP, FDDI, ATM, IEEE 802.5/ 802.2 |
| **FIZIC** | „Mută" şiruri de biţi între echipamente<br>Specifică tipul de cablare, viteza de transmisie, voltaje, tipuri de conectori, etc. | EIA/TIA-232, V.35, EIA/TIA- 449, V.24, RJ45, Ethernet, 802.3, 802.5, FDDI, NRZI, NRZ, B8ZS |

Modalitatea de funcţionare a nivelelor din modelul OSI este următoarea: fiecare nivel oferă servicii nivelului imediat superior (nivelul fizic oferă servicii nivelului legătură de date, nivelul legătură de date oferă servicii nivelului reţea ş.a.m.d.), excepţie făcând nivelul aplicaţie care nu are un alt nivel superior. Comunicarea între două sisteme terminale din reţea se face, de asemenea, pe baza unor protocoale corespunzătoare nivelelor din modelul OSI la

care acestea activează. Astfel, nivelul aplicaţie al unui sistem terminal comunică cu nivelul aplicaţie al celuilalt sistem, nivelul reţea cu nivelul reţea, etc.

În continuare vom prezenta funcţionalităţile fiecărui nivel în parte pe baza modelului top-down, plecând de la nivelul aplicaţie (de vârf) şi ajungând la nivelul fizic (de bază).

## 2.4 Îndrumar pentru autoverificare

### 2.4.1 Sinteza unității de studiu 2

În vastul domeniu al rețelelor de calculatoare, lucrurile trebuie să fie foarte bine structurate. Conform principiului clasic "divide et impera", modelele ierarhice constituite pentru studiul, proiectarea, implementarea, depanarea, studiul rețelelor de calculatoare sunt constituite pe nivele ierarhice, astfel încât fiecare nivel să fie bine documentat. În acest mod, structurarea pe nivele asigură un grad înalt de rigurozitate abordării domeniului rețelelor de calculatoare.

### 2.4.2 Concepte și termeni de reținut

| | |
|---|---|
| *ISO-OSI* | *Model ierarhic* |
| *Aplicație* | *Sesiune* |
| *Presentare* | *Transport* |
| *Rețea* | *TCP/IP* |
| *Legătură de date* | *Nivelul fizic* |

### 2.4.3 Întrebări pentru autoverificare

***Întrebarea 1.*** Ce nivel din modelul ISO-OSI definește serviciile pentru segmentarea și reasamblarea datelor comunicațiiilor individuale între aplicații ?

a) Aplicație
b) Transport
c) Legătură de date
d) Fizic
e) Sesiune

*Răspuns:b*

***Întrebarea 2.*** Care dintre următoarele protocoale acționează la nivelul Internet (alegeți două)?

a) ICMP
b) BOOTP
c) IP
d) PPP
e) POP

*Răspuns: a, c*

***Întrebarea 3.*** În cadrul comunicațiilor dintre computere, care este rolul codificării mesajelor?

a) Pentru negocierea unei sincronizări corecte a comunicației
b) Pentru a divide mesajele mai lungi în frame-uri de lungime mai mică
c) Pentru a interpreta informația
d) Pentru a converti informația într-o fromă specifică transmisiunii

*Răspuns:d*

### Întrebări de control și teme de dezbatere

1. Care sunt nivelele modelului ISO-OSI?
2. Care sunt asemănările și deosebirile între modelele arhitecturale ISO-OSI și TCP/IP?
3. Care sunt caracteristicile de bază ale protocolului TCP?
4. Care sunt caracteristicile de bază ale protocolului UDP?
5. Care sunt nivelele modelului TCP/IP?
6. Ce nivel din modelul ISO-OSI are aceleași funcții și același nume cu un nivel din modelul TCP/IP?

## 2.4.4 Bibliografie obligatorie

1. Răzvan Daniel Zota, Rețele de calculatoare, capitolul 2, Ed. ASE, 2013.

# UNITATEA DE STUDIU 3. Nivelul legătură de date

## 3.1 Introducere

Nivelul legătură de date din modelul OSI are o serie de caracteristici şi funcţionalităţi bine conturate, printre care: determinarea modului în care biţii sunt grupaţi în frame-uri, tratarea erorilor de transmisie, reglarea fluxului de date astfel încât receptorii terţi să nu fie supra-aglomeraţi de emiţători prea rapizi, oferirea de servicii nivelului reţea, etc. Nivelul legătură de date este împărţit, de fapt, în două subnivele (definite de către IEEE):

- ***subnivelul LLC (Logical Link Control)*** – reprezintă subnivelul superior ce defineşte procesele software ce oferă servicii către protocoalele nivelului reţea. Acest subnivel se ocupă de informaţiile din frame ce identifică protocolul de nivel reţea utilizat de către acel frame. Acest tip de informaţie permite utilizarea mai multor protocoale de nivel 3, precum IPv4 sau IPv6, pentru a utiliza în comun aceeaşi interfaţă de reţea şi acelaşi mediu de comunicaţie.
- ***Subnivelul MAC (Media Access Control)*** - acest subnivel inferior defineşte procesele de acces la mediul de comunicaţie realizate de hardware. Oferă adresarea de nivel 2, aşa numita adresare fizică sau adresare MAC. Delimitarea frame-urilor se face în concordanţă cu necesităţile de semnalizare fizică ale mediului de comunicaţie precum şi în funcţie de tipul protocolului de nivel legătură de date utilizat.

Organizaţie IEEE a creat subnivelul LLC pentru nevoia de a avea o parte a nivelului 2 independentă de tehnologiile utilizate. Ca subnivel, LLC participă la procesul de încapsulare a datelor, iar datagramele LLC sunt denumite deseori pachete LLC. Separarea nivelului legătură de date în subnivele permite ca un tip de frame definit la subnivelul superior LLC să poată accesa diferite tipuri de medii de comunicaţie definite la subnivelul inferior MAC. Acest lucru se întâmplă în cazul multor tehnologii de reţea, inclusiv în cazul celei mai răspândite tehnologii de reţea locală, care este tehnologia Ethernet. Astfel, subnivelul LLC comunică direct cu nivelul 3 (reţea), în timp ce subnivelul MAC permite accesul la diverse tehnologii de acces la reţea. Spre exemplu, subnivelul MAC poate comunica cu tehnologia Ethernet LAN pentru a trimite şi recepţiona frame-uri pe fir de cupru sau pe fibră optică. De asemenea, subnivelul MAC poate folosi tehnologii fără fir precum Wi-Fi sau Bluetooth pentru a trimite şi recepţiona frame-uri wireless.

Protocoalele de nivel 2 specifică tipul de încapsulare a unui pachet într-un frame precum şi tehnicile de recepţionare sau trimitere a pachetului încapsulat. Tehnica de manipulare a frame-urilor ce tranzitează mediul de comunicaţie poartă numele de metoda de *control al accesului la mediul de comunicaţie*. Pe drumul parcurs de la sursă către destinaţie, pachetele traversează, de regulă, medii de comunicaţie diferite. Aceste reţele fizice pot fi alcătuite din diverse medii de comunicaţie – bazate pe fir de cupru sau pe fibră optică, medii wireless, legături satelit, etc. Pachetele de date nu au o modalitate de accesare directă a mediului de comunicaţie. Aici intervine rolul nivelului legătură de date de a pregătu pachetele provenite de la nivelul reţea pentru transmisiunea în continuare şi pentru controlul accesului la mediu. Metodele de control al accesului la mediu definite de nivelul legătură de date stabilesc procesele prin care echipamentele de reţea pot accesa mediul de transmisie şi transmit frame-uri în diverse medii de reţea. Fără existenţa nivelului legătură de date, protocoalele nivelului reţea, precum este şi IP ar fi trebuit să se asigure de posibilitatea conectării la toate mediile de comunicaţie disponibile de-a lungul unei rute de comunicaţie în reţea. Mai mult, IP ar fi trebuit să se modifice de fiecare dată când apărea o nouă tehnologie sau un nou mediu de comunicaţie în reţea. Acest fapt constituie un element cheie şi un motiv puternic pentru o abordare separată pe nivele în cazul reţelelor de calculatoare în general.

Subnivelul MAC are de-a face cu protocoalele folosite de un calculator gazdă pentru a avea acces la mediul fizic. Adresele MAC sunt adrese de 48 de biţi lungime şi sunt reprezentate prin 12 cifre hexazecimale. Dintre acestea, primele 6 sunt administrate de către IEEE şi identifică producătorul – *OUI (Organizational Unique Identifier).* Celelalte 6 cifre hexazecimale reprezintă ceva asemănător cu un număr serial şi sunt administrate de către respectivul producător. Adresele MAC sunt uneori referite drept adrese de tip BIA (Burned-In Address), deoarece ele sunt „arse" în memoria *ROM (Read Only Memory)* a plăcii de reţea şi copiate în memoria *RAM (Random Access Memory)* a calculatorului odată cu iniţializarea plăcii de reţea (astfel funcţionează programele ce pot „falsifica" adresa MAC a unei plăci de reţea, suprascriind în memoria RAM o nouă valoare – dar adresa MAC „arsă" în memoria ROM a plăcii de reţea nu poate fi modificată). În cazul sistemului de operare Windows o modalitate rapidă de a vizualiza adresa MAC a unei plăci de reţea este aceea de a apela comanda ***ipconfig/all***; în acest caz adresa MAC a plăcii de reţea LAN Ethernet este identificată prin „Physical Address" (vezi figura 3.1).

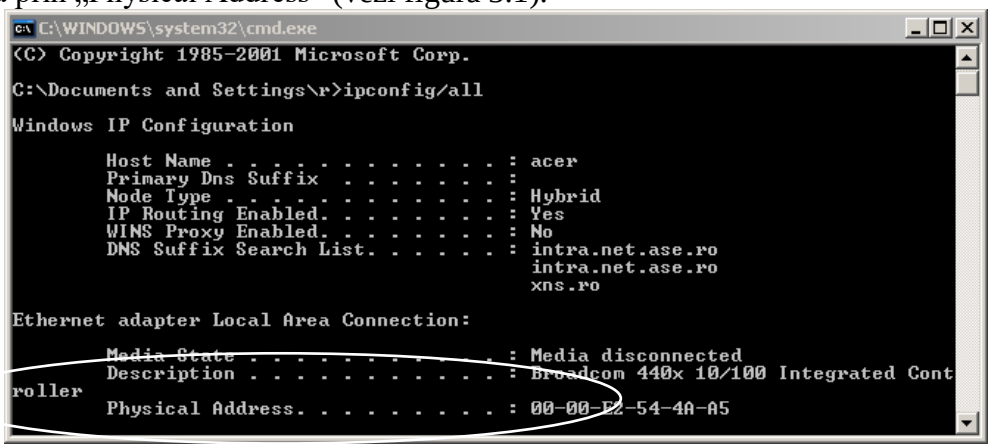



Figura 3.1 Exemplu de apel al comenzii *ipconfig/all*

Datagramele de nivel 2 – frame-urile au un format special în funcţie de tehnologia utilizată, dar care includ următoarele componente (vezi figura 3.2):

- ***Header*** – câmp de început ce conţine informaţii de control şi adresare
- ***Date*** – conţine header-ul IP, cel de nivel transport, precum şi datele de nivel aplicaţie
- ***Trailer*** – câmp de sfârşit ce conţine informaţii de control pentru detecţia erorilor.

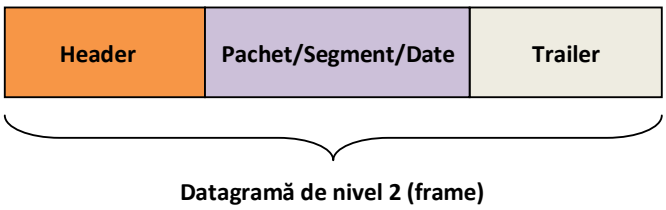



Figura 3.2 Structura simplificată a unui frame

## 3.2 Obiectivele și competențele unității de studiu

*Obiectivele unității de studiu*:

- ❑ identificarea modului în care se realizează controlul accesului la mediu în cadrul nivelului legătură de date oferă suport pentru comunicația de-a lungul unei rețele;

- descrierea scopului și funcțiilor nivelului legătură de date în pregătirea comunicației pentru transmisie pe mediul specific;
- compararea caracteristicilor specifice unor metode de control al accesului la mediu diferite în cazul topologiilor LAN și WAN;
- descrierea caracteristicilor și funcțiilor frame-urilor.

*Competențele unității de studiu*:

- studenții vor putea să definească conceptele de bază întâlnite în cadrul nivelului legătură de date;
- studenții vor cunoaște detalii legate de: structura frame-urilor nivelului legătură de date, tipurile de conexiuni half-duplex și full-duplex, precum și modalitatea de funcționare a echipamentelor de rețea de tip switch.

***Durata medie de studiu individual alocat unității:*** 4 ore

## 3.3 Conținutul unității de studiu

### 3.3.1 Structura unui frame

În momentul în care biţii traversează mediul de comunicaţie trebuie să existe o modalitate prin care să se identifice unde începe şi unde se termină o structură de tip frame. Împărţirea în frame-uri (operaţie denumită *framing*) face ca fluxul de date (biţi) de la nivelul fizic să fie inteligibil – să aibă o anumită structură ce poate fi recepţionată de nodurile din reţea şi să fie decodificată în pachete de date la destinaţie. Un frame generic are următoarele câmpuri generale (vezi figura 3.3):

- Indicatorii ***start*** şi ***stop*** ai frame-ului – utilizaţi de către subnivelul MAC pentru a identifica începutul, respectiv sfârşitul frame-ului;
- ***Adresare*** – informaţiile legate de adresare permit subnivelului MAC să identifice echipamentul sursă şi cel destinaţie – aici apar adresele MAC destinaţie şi sursă (în această ordine);
- ***Tipul –*** utilizat de subnivelul LLC pentru a identifica protocolul de nivel 3 folosit;
- ***Control –*** utilizat pentru identificarea de servicii speciale pentru controlul fluxului;
- ***Date –*** acest câmp conţine aşa numitul „data payload” – header-ul de nivel 4 (segment), header-ul de nivel 3 (reţea) precum şi datele de la nivelul aplicaţie;
- ***Detecţia erorilor –*** câmp utilizat pentru detecţia erorilor; împreună cu *frame stop* formează *trailer-ul* unui frame.

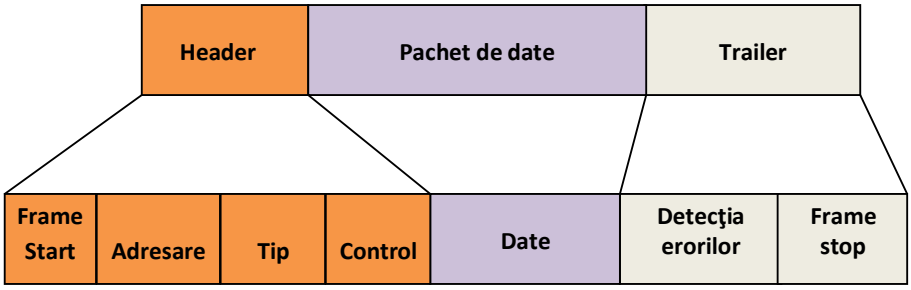



Figura 3.3 Structura generică a unui frame

Toate protocoalele de nivel legătură de date încapsulează datagramele de nivel 3 în cadrul *câmpului de date* din frame. Cu toate acestea, structura frame-urilor şi ale câmpurilor conţinute de acestea variază în funcţie de protocol. Nu toate protocoalele de nivel legătură de date includ toate aceste câmpuri; standardele ce reglementează protocoalele nivelului legătură de date definesc formatul respectiv de frame. În general, header-ul unui frame conţine informaţie de control specificată de protocolul de nivel legătură de date pentru respectiva topologie logică şi pentru mediul de comunicaţie folosit. Informaţia de control a frame-ului este unică fiecărui tip de protocol, fiind folosită de nivelul a pentru a oferi facilităţile cerute de mediul de comunicaţie. În figura 3.4 prezentăm câmpurile unui frame Ethernet, tehnologie definitorie pentru reţelele locale de astăzi:

- Câmpul ***Start Frame*** – indică începutul unui nou frame;
- Câmpurile ***Adresă Destinaţie şi Sursă*** – indică destinaţia şi sursa respectivului frame;
- Câmpul ***Tip/Lungime*** – indică serviciul de nivel superior conţinut în frame.


Figura 3.4 Structura header-ului unui frame Ethernet

Alte protocoale de nivel legătură de date pot utiliza alte câmpuri faţă de cele menţionate mai înainte. Spre exemplu, alte câmpuri ale header-ului unui frame de nivel 2 pot include:

- Câmp de ***Prioritate/Calitate a serviciilor (QoS)*** – indică procesarea unui tip particular de serviciu de comunicaţie;
- Câmpul ***Control fizic al legăturii*** – folosit pentru stabilirea legăturii pe mediul de comunicaţie;
- Câmpul ***Control logic al conexiunii*** – folosit pentru stabilirea unei legături logice între noduri;
- Câmpul ***Control al fluxului*** – folosit pentru pornirea sau oprirea traficului de date;
- Câmpul ***Control al congestiilor*** – indică apariţia unei congestii (blocaj) pe mediul de comunicaţie.

Protocoalele de nivel legătură de date adaugă un trailer la sfârşitul fiecărui frame. Acest trailer este utilizat pentru a determina dacă frame-ul a ajuns la destinaţie fără erori – proces denumit *detecţia erorilor*. Procesul de detecţie a erorilor este necesar deoarece semnalele transmise pe mediul de comunicaţie pot suferi unele distorsiuni, interferenţe sau pierderi ce modifică valoarea biţilor reprezentaţi de către aceste semnale.

Fiecare nod ce transmite în reţea creează un câmp de corecţie (de regulă prin operaţii ce implică algebra booleană) ce se numeşte *CRC (Cyclic Redundancy Check) – Control ciclic de redundanţă*, iar această valoare este plasată în câmpul ce se numeşte *FCS (Frame Check Sequence)* al unui frame. Pentru fiecare frame ajuns la destinaţie, nodul ce recepţionează frame-ul va calcula (conform controlului ciclic de redundanţă) câmpul FCS şi îl va compara cu cel existent în frame. Dacă cele două câmpuri coincid, înseamnă că frame-ul a ajuns corect la destinaţie; în caz contrar, frame-ul este invalid şi va fi înlăturat, urmând a fi retransmis. În acest mod simplu câmpul FCS este utilizat pentru detecţia erorilor apărute la transmisia frame-urilor de la sursă către destinaţie. Există totuşi şi o mică posibilitate ca un frame cu un câmp CRC bun să fie, de fapt, transmis greşit. În acest caz, protocoalele ce aparţin nivelelor superioare vor detecta şi corecta datele recepţionate greşit.

Într-o reţea bazată pe stiva de protocoale TCP/IP toate protocoalele de nivel 2 OSI

lucrează cu protocolul IP la nivelul 3, însă protocolul de nivel 2 folosit depinde de topologia logică a reţelei, precum şi de implementarea de nivel fizic. Având în vedere domeniul larg de medii de comunicaţii folosite pentru toate topologiile existente în lumea reţelelor de calculatoare, există un număr corespunzător de mare de protocoale de nivel 2 ce sunt folosite.

Fiecare protocol de nivel 2 defineşte controlul accesului la mediu iar acest fapt presupune că un număr de diferite echipamente de reţea pot acţiona ca noduri ce operează la nivel legătură de date atunci când sunt implementate aceste protocoale. Aceste echipamente de reţea includ adaptoare sau plăci de reţea (regăsite în engleză sub denumirea de *NICs - Network Interface Cards)* instalate pe calculatoare, laptop-uri, alte echipamente inteligente, rutere şi switch-uri de nivel 2. Protocolul de nivel 2 utilizat pentru o topologie particulară de reţea este determinat de tehnologia folosită pentru implementare. Această tehnologie, la rândul ei, este influenţată de dimensiunea reţelei – ca număr de gazde şi ca arie geografică de acoperire – precum şi de serviciile ce trebuie oferite în reţea.

O reţea locală (*LAN – Local Area Network*) foloseşte în mod uzual o tehnologie ce posedă o lăţime de bandă înaltă capabilă să ofere suport pentru un număr mare de gazde. Aria de acoperire a unei reţele locale este relativ restrânsă (de la câteva calculatoare conectate între ele într-un laborator până la reţele ce se întind pe o rază de câţiva kilometri – de dimensiunea unui campus universitar) iar densitatea mare de utilizatori ai reţelei face ca această tehnologie să fie eficientă din punct de vedere al costurilor. În cazul tehnologiilor de lăţime de bandă mare pe arii largi de acoperire (reţele WAN – Wide Area Network) eficienţa costurilor nu este aşa de mare iar costurile legăturilor pe distanţe mari şi tehnologia folosită pentru transmisia semnalelor determină capacităţi de lăţime de bandă mai mici decât în cazul reţelelor locale.

### 3.3.2 Topologii de rețea

Modalitatea de amplasare a frame-urilor pe mediul de comunicaţie este controlat de către subnivelul MAC al nivelului legăturii de date. Controlul accesului la mediu este echivalent cu regulile de trafic ce reglementează intrarea automobilelor pe o autostradă. Absenţa unui astfel de control ar fi echivalentul ignorării de către automobile a traficului de pe o autostradă şi de intrare în trafic fără a ţine cont de celelalte automobile. Cu toate acestea, nu toate drumurile şi nu toate intrările pe aceste drumuri sunt la fel; automobilele se pot alătura traficului existent pe o bandă separată, ele pot aştepta intrarea la un semafor sau trebuie să respecte nişte indicatoare de circulaţie. În concluzie, şoferii trebuie să respecte anumite reguli în funcţie de fiecare tip de intrare.

În mod asemănător, protocoalele de nivel legătură de date definesc regulile pentru accesul la diferite medii de comunicaţie. Diferite implementări ale protocoalelor de nivel legătură de date utilizează diverse metode de control al accesului la mediu. Aceste tehnici de control al accesului la mediu definesc modalitatea în care nodurile reţelei partajează mediul de comunicaţie. Metodele de control al accesului la mediu depind de *topologie* (modalitatea în care conexiunile dintre noduri apar nivelului legătură de date) şi de modalitatea de *partajare a mediului* (modul în care nodurile partajează mediul).

Topologia unei reţele reprezintă modalitatea de aranjare a echipamentelor în reţea precum şi relaţia de interconexiune între acestea. Există, în acest sens, două topologii de reţea:

- ***Topologia fizică*** – se referă la conexiunile fizice între echipamente şi identifică modul în care acestea (rutere, switch-uri, PC-uri, AP-uri) sunt interconectate;
- ***Topologia logică*** – reprezintă modul în care sunt transferate frame-urile de la un nod la altul, alcătuindu-se un sistem de circuite virtuale între nodurile reţelei. Aceste „drumuri logice” ale semnalelor transmise în reţea sunt definite de către

protocoalele nivelului legătură de date. Spre exemplu, o topologie logică de tip punct-la-punct este relativ simplă, în timp ce o topologie de partajare a mediului oferă varianta deterministă sau non-deterministă a accesului la mediu.

Nivelul legătură de date stabileşte topologia logică a unei reţele atunci când controlează accesul datelor la mediul de comunicaţie; astfel, varianta de topologie logică influenţează modalitatea de împărţire a datelor în frame-uri precum şi modul de control al accesului la mediu.

6.2.4.1 Topologii de reţea locală

Topologia fizică defineşte modalitatea în care sunt interconectate echipamentele de reţea. În cazul reţelelor locale LAN există următoarele topologii de reţea:

- Topologia de tip ***stea (star)*** – în acest caz echipamentele sunt conectate la un nod central (vezi figura 3.5); dacă iniţial topologiile de tip stea conţineau un hub central, astăzi se foloseşte un switch central. Topologia stea este cea mai populară topologie de reţea locală astăzi deoarece este uşor de instalat, scalabilă (este uşor să adaugi echipamente noi) şi uşor de depanat.

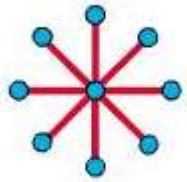


Figura 3.5 Topologia de tip *stea (star)*

- Topologia ***stea extinsă (extended star)*** – în figura 3.6 este ilustrată o astfel de topologie de reţea în care echipamentele centrale interconectează alte topologii stea. În cazul topologiilor ***hibride*** reţelele stea pot fi interconectate printr-o topologie de tip ***magistrală (bus)***.
- Topologia de tip ***magistrală (bus)*** – aici echipamentele sunt conectate unul cu celălalt sub forma unui lanţ, nefiind nevoie de echipamente de tip switch pentru interconexiune. Topologiile de tip magistrală (figura 3.7) erau utilizate în vechile reţele Ethernet deoarece erau ieftine şi uşor de instalat.

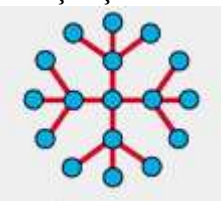


Figura 3.6 Topologia de tip *stea extinsă (extended star)*

- Topologia de tip ***inel (ring)*** – în acest caz (figura 3.8) echipamentele sunt conectate cu vecinii pentru a forma un inel. Spre deosebire de topologia magistrală, inelul nu trebuie să fie terminat cu o conexiune fizică. Topologia inel a fost utilizată în vechile reţele locale de tip Token Ring sau FDDI (Fiber Distributed Data Interface) – în acest din urmă caz existând un inel dual ce interconecta fiecare echipament cu vecinii săi.

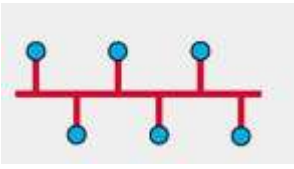


Figura 3.7 Topologia de tip *magistrală (bus)*

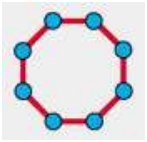


Figura 3.8 Topologia de tip inel (ring)

#### 6.2.4.2 Topologii de reţea de arie largă

Reţelele de arie largă *(WAN – Wide Area Network)* utilizează, de regulă, trei tipuri de topologii fizice:

- Topologia *punct-la-punct*
- Topologia „hub-and-spoke”
- Topologia „mesh”

Topologia *punct-la-punct* (figura 3.9) reprezintă cea mai simplă topologie şi constă în realizarea unei legături fizice permanente între două puncte aflate le extremităţile unei conexiuni, fiind una dintre cele mai utilizate tehnologii de tip WAN. În cazul acestui tip de legătură cele două noduri nu trebuie să partajeze mediul de comunicaţie cu alte gazde iar nodurile nu trebuie să analizeze permanent dacă un frame îi este destinat sau nu (este implicit acest lucru). În concluzie, protocoalele nivelului legătură de date în acest caz pot fi foarte simple din moment ce toate frame-urile ce traversează mediul se „plimbă” doar între cele două noduri implicate în conexiune. Aceste protocoale de comunicaţie ar putea fi mai „sofisticate”, oferind un control mai complex al accesului la mediu, dar acest lucru ar fi inutil şi ar adăuga informaţii suplimentare *(overhead)* datelor transmise.

Figura 3.9 Topologia *punct-la-punct*

Nodurile comunicante în cazul topologiei *punct-la-punct* pot fi interconectate fizic cu ajutorul mai multor dispozitive intermediare de reţea, fără ca aceste echipamente să afecteze topologia logică a reţelei. În figura 3.10 nodurile A şi B sunt conectate indirect peste o arie geografică largă, prin utilizarea unui circuit virtual între acestea. Un circuit virtual reprezintă o conexiune logică creată între două noduri ale unei reţele. Cele două noduri terminale ale conexiunii schimbă frame-uri între ele, chiar dacă aceste frame-uri trec prin diverse echipamente intermediare existente în „norul” reţelei. Circuitele virtuale reprezintă un concept important al comunicaţiei logice utilizat de unele dintre tehnologiile de nivel legătură de date. Metoda de acces la mediu utilizată de către aceste protocoale de nivel 2 este determinată de către topologia punct-la-punct logică şi nu de către cea fizică, ceea ce înseamnă faptul că o legătură logică punct-la-punct între două noduri nu trebuie să corespundă în mod neapărat cu legătura fizică punct-la-punct dintre două noduri aflate la capetele unei legături.


Figura 3.10 Topologie logică de tip *punct-la-punct*

Topologia *„hub-and-spoke”* (figura 3.11) este versiunea WAN a topologiei LAN de tip stea în care un sediu central interconectează alte sedii folosind legături fizice de tip *punct-la-punct*.

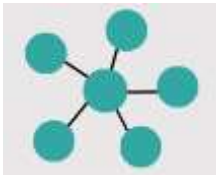


Figura 3.11 Topologia *hub-and-spoke*

Topologia *completă (mesh)* oferă o redundanţă remarcabilă, însă necesită ca fiecare nod al reţelei să fie legat de toate celelalte noduri printr-o legătură fizică, ceea ce determină costuri ridicate de realizare şi administrare. Fiecare legătură din topologia *mesh* (figura 3.12) este practic o legătură de tip *punct-la-punct*. Numărul de legături creşte foarte mult odată cu creşterea numărului de noduri (pentru ***n*** noduri vor exista ***n*(n-1)/2*** legături bidirecţionale). Există şi versiuni parţiale de tip *mesh*, în care nu toate legăturile dintre noduri sunt prezente.

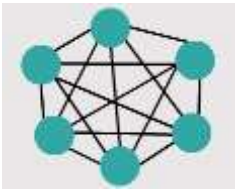


Figura 3.12 Topologia de tip *mesh*.

## 3.4 Îndrumar pentru autoverificare

### 3.4.1 Sinteza unității de studiu 3

Nivelul legătură de date încapsulează datele (de regulă pachete IPv4 sau IPv6) provenite de la nivelul rețea pentru a le pregăti în vederea transportului acestora pe mediul de comunicație. Procedeul de încapsulare presupune adăugarea unui header și a unui trailer și crearea unui frame. Nivelul legătură de date este responsabil cu comunicațiile placă de rețea-placă de rețea din cadrul aceleiași rețele. Protocolul de nivel 2 ce este folosit aici pentru o anumită topologie de rețea este determinat de tehnologia folosită pentru implementarea acesteia. Protocoalele de nivel legătură de date sunt: Ethernet, 802.11 Wireless, PPP, HDLC și Frame Relay.

### 3.4.2 Concepte și termeni de reținut

| | |
|---|---|
| *Legătură de date* | *PPP* |
| *HDLC* | *Frame Relay* |
| *802.11 Wireless* | *Header* |
| *Trailer* | *FCS* |
| *CSMA/CD* | *CSMA/CA* |

### 3.4.3 Întrebări pentru autoverificare

***Întrebarea 1.*** Ce tip de identificator este folosit la nivelul legătură de date pentru a identifica un echipament Ethernet?

a) Adresa MAC
b) Adresa IP
c) Numărul de secvență
d) Numărul de port TCP
e) Numărul de port UDP

*Răspuns:a*

***Întrebarea 2.*** Ce atribut al plăcii de rețea o plasează la nivel legătură de date din modelul ISO-OSI?

a) TCP/IP protocol stack
b) Portul RJ-45
c) Adresa IP
d) Adresa MAC
e) Cablul Ethernet

*Răspuns: d*

***Întrebarea 3.*** Ce metodă este folosită pentru administrarea accesului într-o rețea wireless?

a) Token passing
b) CSMA/CD
c) CSMA/CA
d) Ordonarea priorităților

*Răspuns:c*

#### Întrebări de control și teme de dezbatere

1. Care sunt sub-nivelele nivelului legătură de date?
2. Care este diferența între CSMA/CD și CSMA/CA?
3. Ce topologii fizice de rețea locală cunoașteți?
4. Ce topologii fizice de rețea de arie largă cunoașteți?
5. Unde se află nivelul legătură de date în modelul TCP/IP?

### 3.4.4 Bibliografie obligatorie

1. Răzvan Daniel Zota, Rețele de calculatoare, capitolul 6, Ed. ASE, 2013.

# UNITATEA DE STUDIU 4. Nivelul rețea



## 4.1 Introducere

Nivelul reţea din modelul ISO-OSI (echivalentul nivelului Internet din modelul TCP/IP) are două funcţii de bază: prima este aceea legată de asigurarea *adresării logice* a echipamentelor din reţea (este vorba aici despre adresarea IP – Internet Protocol) iar a doua funcţie este aceea de a asigura direcţionarea corectă a pachetelor de date în drumul lor de la sursă către destinaţie (această operaţie se numeşte *rutare* iar echipamentele specializate care realizează acest lucru sunt *ruterele*). Pe lângă adresare logică şi rutare, la nivelul Internet au loc încapsularea şi decapsularea datelor. Vom folosi în continuare denumirea de *nivel reţea* – identificând atât nivelul Internet (TCP/IP) cât şi nivelul reţea ISO-OSI.

În ceea ce priveşte încapsularea, nivelul reţea primeşte o datagramă (denumită, de regulă, *segment*) de la nivelul transport căreia îi adaugă un header de informaţie IP ce conţine adresa IP sursă şi adresa IP destinaţie. După ce informaţia header este adăugată la datagramă, aceasta se numeşte *pachet*. În cazul decapsulării, atunci când un pachet ajunge de la nivelul legătură de date la nivelul reţea, se verifică header-ul IP al acestuia. Dacă adresa IP destinaţie coincide cu adresa IP a calculatorului local, atunci header-ul IP este înlăturat prin procedeul de decapsulare şi rezultă segmentul de nivel 4 (transport).

Principalele protocoale ce funcţionează la nivel reţea sunt:

- ***Internet Protocol versiunea 4 (IPv4)*** – ajută în stabilirea de legături neorientate pe conexiune, de tip „best-effort", fără a ţine cont de conţinutul pachetelor (datagramelor). Caută, în schimb, o cale pentru a trimite pachetele către destinaţie. Adresele IPv4 sunt adrese pe 32 de biţi.
- ***Internet Protocol versiunea 6 (IPv6)*** – versiune mai nouă a lui IPv4, cu adrese pe 128 de biţi.
- ***Internet Control Message Protocol (ICMP)*** – oferă funcţionalităţi de control şi mesagerie.
- ***Address Resolution Protocol (ARP)*** – ajută la determinarea adresei fizice (MAC) a unui echipament destinaţie atunci când este cunoscută adresa IP.
- ***Reverse Address Resolution Protocol (RARP)*** – ajută la determinarea adresei IP destinaţie atunci când este cunoscută adresa fizică (MAC).

Există şi o serie de protocoale de nivel reţea mai vechi (*legacy*), dintre care amintim: IPX (Internetwork Packet Exchange) de la Novell, AppleTalk de la Apple, CLNS (Connectionless Network Service) şi DECNet.

În figura 4.1 este prezentat header-ul IPv4.

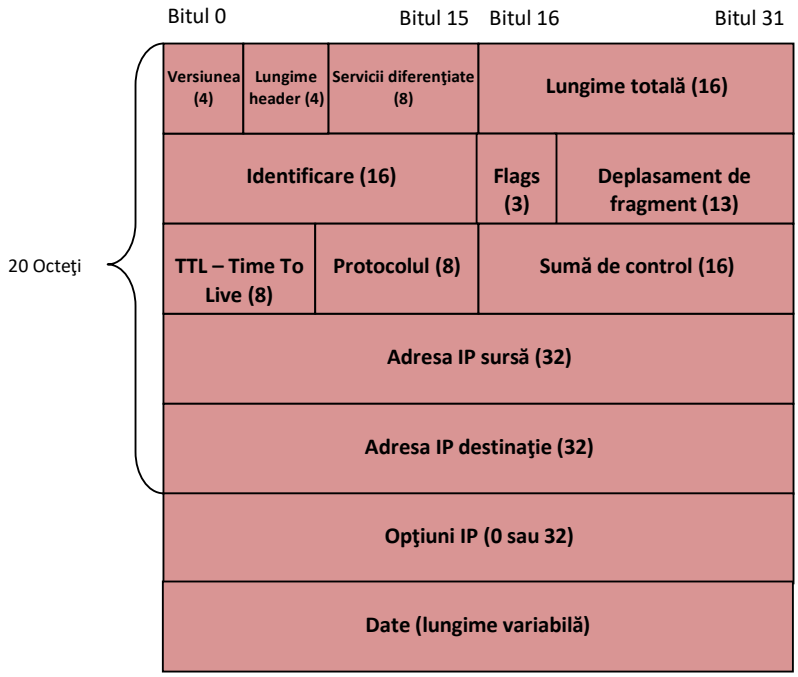



Figura 4.1 Câmpurile header-ului IPv4

## 4.2 Obiectivele și competențele unității de studiu

*Obiectivele unității de studiu*:

- Prezentarea funcționalităților nivelului rețea din modelul ISO-OSI;
- Prezentarea caracteristicilor protocolului IPv4;
- Prezentarea caracteristicilor protocolului IPv6;
- Împărțirea în subrețele de lungime fixă și variabilă în cazul IPv4.

*Competențele unității de studiu*:

- studenții vor putea să definească conceptele de bază întâlnite în cadrul nivelului rețea;
- studenții vor cunoaște detalii legate de: structura pachetelor IPv4 și IPv6, modalitatea de funcționare a ruterelor.

***Durata medie de studiu individual alocat unității:*** 4 ore

## 4.3 Conținutul unității de studiu

### 4.3.1 Funcționalitățile protocolului IP

Protocolul IP (Internet Protocol) reprezintă principalul protocol de nivel rețea din suita de protocoale TCP/IP şi principalul protocol de adresare în Internet. Publicat iniţial în documentul RFC 791 în anul 1981 sub denumirea completă *„Internet Protocol - DARPA*

*Internet Program Protocol Specification ",* protocolul IP are, în consecinţă, o lungă istorie ce începe cu primii ani ai dezvoltării reţelei Internet şi continuă cu succes şi astăzi. De la început, protocolul IP a fost proiectat ca fiind un protocol cu minimum de informaţii redundante adăugate în header-ul de reţea. În definirea funcţionalităţilor sale din RFC 791 se precizează că IP este „limitat la a oferi funcţiile necesare pentru transportul unui pachet de biţi (datagramă internet) de la o sursă la o destinaţie într-un sistem interconectat de reţele. Nu există mecanisme să ofere fiabilitatea transmisiei datelor, controlul fluxului, secvenţializarea sau alte servicii uzuale întâlnite în protocoale de tip host-to-host. Protocolul IP se poate baza pe alte servicii ale reţelelor folosite pentru a obţine diverse tipuri de calităţi ale serviciilor". De asemenea, protocolul IP nu a fost conceput să administreze fluxul de transmisie a pachetelor de la o sursă la destinaţie, iar principalele trei caracteristici ale sale sunt:

- *Protocol de tip „best-effort"* – nu garantează livrarea pachetelor la destinaţie, nu este fiabil. IP se bazează pe fiabilitatea reţelei sau a altor protocoale de reţea pentru livrarea cu succes a pachetelor de date de la sursă către destinaţie.
- *Protocol neorientat pe conexiune* – acest lucru presupune faptul că nu există o etapă iniţială de stabilire a conexiunii între două gazde, ci datele sunt transmise imediat.
- *Protocol independent de mediul de comunicaţie* – funcţionarea acestuia este independentă de tipul de mediu de comunicaţie folosit.

***Caracteristica „best-effort"***

Rolul nivelului reţea este acela de a ajuta la transportul pachetelor de date între dispozitive, folosind cât mai puţină informaţie redundantă. Nivelul reţea nu este interesat de tipul de comunicaţie explicitat în interiorul unui pachet de date, fiind un protocol neorientat pe conexiune, adică nu este creată nicio conexiune dedicată de tip *punct-la-punct* înainte de transmisia datelor de la sursă către destinaţie. Cel mai simplu exemplu (din lumea reală) de comunicaţie neorientată pe conexiune este acela al trimiterii unei scrisori clasice (pe hârtie), caz în care destinatarul nu este anunţat în prealabil de primirea scrisorii. De asemenea, ca şi în cazul unei simple scrisori, expeditorul nu este anunţat dacă scrisoarea a ajuns la destinaţie sau dacă a parcurs un anumit drum. Protocolul IP funcţionează în acelaşi mod. Lipsa unor informaţii suplimentare legate de confirmarea primirii sau de conţinutul pachetelor de date face ca protocolul IP să fie un protocol „suplu", fără un avea un header supra-încărcat.

Cu toate că protocolul IP este considerat nefiabil, aceasta nu înseamnă că datele nu sunt transmise cât se poate de eficient şi corect către destinaţie. Nefiabilitatea protocolului constă în faptul că protocolul IP nu are capacitatea de a administra retransmisia pachetelor greşite sau pierdute de date. Acest lucru se întâmplă datorită faptului că pachetele sunt trimise pe baza adresei destinaţie dar nu există informaţii în legătură cu informarea expeditorului cu privire la recepţionarea cu succes a acestora. Nu există informaţii de sincronizare incluse în header-ul IP pentru a ţine cont de ordinea recepţionării pachetelor. De asemenea, nu există confirmări de primire a pachetelor şi nici informaţii pentru controlul erorilor pentru a verifica faptul că pachetele au ajuns nemodificate la destinaţie. Există posibilitatea ca pachetele să ajungă la destinaţie modificate, în altă ordine, sau să nu ajungă deloc. Pe baza informaţiilor conţinute în header-ul IP nu există posibilitatea retransmisiei pachetelor dacă apar astfel de erori la transmitere. În cazul în care apar pachete pierdute sau recepţionate în altă ordine decât cea iniţială, atunci protocoalele nivelelor superioare, spre exemplu TCP, trebuie să rezolve astfel de probleme şi să permită astfel protocolului IP să funcţioneze în mod eficient. Dacă în header-ul IP ar fi fost introduse informaţii suplimentare legate de fiabilitate, atunci comunicaţiile ce nu au nevoie de conexiuni iniţiale sau fiabilitate ar fi consumat inutil lăţime

de bandă cu aceste informaţii. În suita de protocoale TCP/IP nivelul transport poate decide să folosească TCP sau UDP în funcţie de necesităţile legate de fiabilitate. Lăsând decizia legată de fiabilitate nivelului transport, protocolul IP poate fi mai adaptabil şi poate oferi suport pentru diverse tipuri de comunicaţie.

***Independenţa de mediu***

O altă caracteristică importantă a nivelului reţea este aceea că nu ţine cont de caracteristicile mediilor de comunicaţie pe care sunt transportate pachetele de date. Protocolul IP funcţionează independent de mediul de comunicaţie pe care sunt transportate datele la nivelele inferioare din stiva de protocoale. Pachetele IP pot fi transportate pe medii bazate pe fir de cupru, pe medii de fibră optică sau folosindu-se mediul fără fir (wireless). Nivelul legătură de date este responsabil cu preluarea pachetelor IP provenite de la nivelul reţea şi pregătirea acestora pentru transmisia pe mediul de comunicaţie respectiv. Acest lucru presupune faptul că transportul pachetelor IP în reţea nu este limitat la nici un mediu de comunicaţie în particular. Există, însă, o caracteristică importantă a mediului de comunicaţie pe care o ia în consideraţie nivelul reţea şi anume dimensiunea maximă a datagramelor pe care fiecare mediu de comunicaţie le poate transmite; această caracteristică se numeşte *MTU (Maximum Transmission Unit).* Stabilirea unei dimensiuni maxime pentru pachetele transmise este parte integrantă a controlului comunicaţiei dintre nivelul legătură de date şi nivelul reţea. Nivelul legătură de date transmite valoarea MTU către nivelul reţea şi astfel se determină cât de mari pot fi pachetele transmise. În unele cazuri, echipamente intermediare (de regulă rutere) pot împărţi în mai multe bucăţi un pachet atunci când îl transmit mai departe către alt mediu de comunicaţie ce posedă o valoare MTU mai mică; această operaţiune se numeşte *fragmentarea pachetelor*, sau, pe scurt, *fragmentare*.

***Construirea pachetelor IP***

Conform procedeului general de încapsulare a datelor, la nivelul reţea protocolul IP încapsulează (împachetează) segmentul de nivel transport prin adăugarea unui header IP, conform figurii 4.2.

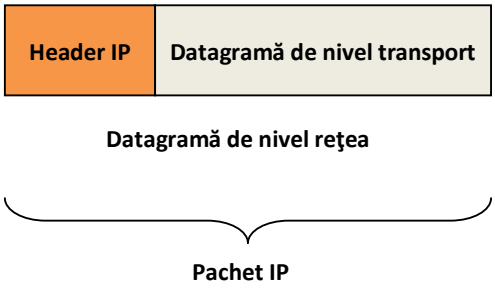



**Figura 4.1 Încapsularea la nivel reţea creează pachetul IP**

Header-ul IP este folosit pentru a livra pachetul la destinaţie, el rămânând pe loc din momentul în care pachetul părăseşte nivelul reţea al calculatorului sursă până în momentul în care ajunge la nivelul reţea al calculatorului destinaţie. Procesul de încapsulare a datelor permite dezvoltarea serviciilor la fiecare nivel în parte fără a afecta alte nivele. Acest lucru permite, spre exemplu, ca segmentele nivelului transport să fie împachetate folosind protocolul IPv4, IPv6 sau oricare alt protocol nou de nivel reţea. Ruterele pot implementa diferite protocoale de nivel reţea ce pot să opereze concomitent într-o reţea. Operaţia de rutare realizată de către aceste echipamente intermediare iau în considerare doar conţinutul

header-ului pachetului ce încapsulează segmentul, iar porţiunea de date din pachet rămâne neschimbată în timpul proceselor ce au loc la nivelul reţea.

### 4.3.2 Protocolul IPv6

Odată cu trecerea anilor, protocolul IPv4 a fost actualizat pentru a face faţă noilor provocări apărute. Dintre noutăţile apărute se evidenţiază în mod deosebit notaţia CIDR (Classless Internet Domain Routing – RFC 4632/2006) şi adresarea privată (RFC 1918/1996). Cu toate acestea, protocolul IPv4 are totuşi trei probleme majore:

- Epuizarea spaţiului de adrese IPv4 – IPv4 are un număr limitat de adrese IP publice disponibile. Cu toate că sunt peste 4 miliarde de adrese disponibile, numărul tot mai crescut de echipamente noi bazate pe IP, noile conexiuni permanente şi creşterea potenţială a zonelor mai puţin dezvoltate conduc la o cerere din ce în ce mai mare de adrese IP publice.
- Extinderea tabelelor de rutare în Internet – tabelele de rutare sunt folosite de către rutere pentru a face alegeri de direcţionare a pachetelor în drumul lor de la sursă către destinaţie. Pe măsură ce numărul de servere (noduri) conectate la Internet creşte, la fel creşte şi numărul de rute în reţea. Aceste rute IPv4 consumă o mare cantitate de resurse de memorie şi procesare pe ruterele ce fac să funcţioneze reţeaua Internet.
- Lipsa conectivităţii capăt-la-capăt – Tehnologia NAT (Network Address Translation) permite reţelelor IPv4 să folosească o singură adresă publică partajată pentru a permite conectarea la reţea a mai multor echipamente. Datorită faptului că adresa IP publică este partajată între mai multe echipamente, adresele IP interne ale echipamentelor sunt ascunse, iar acest lucru poate cauza probleme în cazul tehnologiilor ce necesită conectivitate capăt-la-capăt.

Explozia Internetului începută în anii 1990 a făcut ca tot mai multe calculatoare să fie conectate la reţea şi, în consecinţă, au apărut probleme legate de epuizarea spaţiului de adrese IPv4. În această perioadă, grupul de standardizare IETF a început lucrul la această problemă ce trebuia rezolvată mai devreme sau mai târziu. În afara limitării spaţiului de adrese, IPv4 are şi alte limitări legate de calitatea serviciilor (QoS – Quality of Service), criptarea comunicaţiei capăt-la-capăt, autentificarea pachetelor şi rutare. Printre soluţiile propuse pentru a înlocui IPv4 s-au numărat TUBA (TCP/UDP over Bigger Addresses) – protocol bazat pe protocolul CLNP (ConnectionLess Networking Protocol) şi NSAP (Network Service Access Protocol) cu 20 de octeţi pentru adresare (care a fost respins datorită lipsei caracteristicilor legate de multicasting, calitatea serviciilor şi altele).

În cele din urmă însă, câştigător a fost un alt protocol compatibil cu IP, CLNP şi IPX . Acest protocol, cunoscut iniţial ca SIPP (Simple IP Plus) a crescut spaţiul de adresare de la 32 la 64 de biţi şi a reglat câteva dintre caracteristicile lui IPv4. Propunerea iniţială a suferit câteva modificări iar în final spaţiul de adresare a crescut la 128 de biţi iar denumirea aleasă a fost IPv6 (versiunea 5 fusese deja folosită!). În aceste condiţii IPv6 a apărut ca fiind protocolul ce poate rezolva problemele de scalabilitate ale Internetului.

## 4.4 Îndrumar pentru autoverificare

### 4.4.1 Sinteza unității de studiu 4

Nivelul rețea încapsulează datele provenite de la nivelul transport pentru a le pregăti în vederea direcționării acestora către destinație. Procedeul de încapsulare presupune adăugarea unui header IP (Internet Protocol) și crearea unui pachet. Nivelul rețea este responsabil cu

comunicațiile cu dispozitive din afara rețelei locale. Protocoalele IPv4 sau IPv6 sunt folosite astăzi în mod dual, până în momentul în care IPv6 va lua complet locul "vechiului" IPv4.

### 4.4.2 Concepte și termeni de reținut

| | |
|---|---|
| *Nivel rețea* | *IPv4* |
| *IPv6* | *Rutare* |
| *Pachete de date* | *Header IP* |
| *IPsec* | *Adresare logică* |
| *Subnetting* | *VLSM* |

### 4.4.3 Întrebări pentru autoverificare

***Întrebarea 1.*** Care este reprezentarea binară a adresei IPv4 223.1.3.27 ?

*Răspuns: 11011111.00000001.00000011.00011011*

***Întrebarea 2.*** Ce nivel din modelul TCP/IP corespunde nivelului rețea din modelul ISO-OSI ?

a) Acces la rețea
b) Internet
c) Aplicație
d) Transport
e) Ethernet

*Răspuns: b*

***Întrebarea 3.*** Care este numărul maxim de subrețele ce pot fi asignate pentru adresa IP 172.16.0.0 cu un subnet mask de 255.255.240.0 ?

a) 16
b) 32
c) 30
d) 14
e) Valoarea lui subnet mask este invalidă

*Răspuns:a*

### Întrebări de control și teme de dezbatere

1. Care sunt funcțiile de bază ale nivelului rețea?
2. Comparați header-ele IPv4 și IPv6. Au câmpuri în comun?
3. Fie un ruter cu trei interfețe. Presupunem că aceste interfețe folosesc adrese IPv4 de clasă C. Vor avea în mod necesar adresele IP ale acestor trei interfețe primii opt biți egali?
4. Cum definiți proprietatea de scalabilitate a unei rețele?

5. Căutați pe web pentru a afla care sunt cele mai cunoscute protocoale de rutare în Internet.

### 4.4.4 Bibliografie obligatorie

1. Răzvan Daniel Zota, Rețele de calculatoare, capitolul 5, Ed. ASE, 2013.

# UNITATEA DE STUDIU 5. Caracteristicile protocolului IPv6



## 5.1 Obiectivele și competențele unității de studiu

*Obiectivele unității de studiu*:

- Prezentarea caracteristicilor protocolului IPv6;
- Exemplificarea tipurilor de adrese de rețea noi concepute în IPv6.

*Competențele unității de studiu*:

- studenții vor putea să definească conceptele noului protocol rutat IPv6;
- studenții vor cunoaște detalii legate de adresarea în cadrul noului protocol IPv6.

***Durata medie de studiu individual alocat unității:*** 4 ore

## 5.2 Conținutul unității de studiu

### 5.2.1 Modalități de reprezentare a adreselor IPv6

Adresele IPv6 sunt adrese pe 128 de biţi, scrise sub formă de 32 de cifre hexazecimale (având în vedere că o cifră hexazecimală se scrie pe 4 biţi, 32*4=128). Dacă în cazul adreselor IPv4 scrierea obişnuită este cea „zecimală cu punct”, de genul 192.168.250.1, în cazul adreselor IPv6 cifrele hexazecimale sunt despărţite de semnul „:”, astfel încât o adresă IPv6 are formatul: „x:x:x:x:x:x:x:x”, unde „x” reprezintă un grup de 4 cifre hexazecimale. În terminologia neoficială IPv6 fiecare grup „x” este un „hextet”, adică un grup de 16 cifre binare.

Formatul preferat de scriere a unei adrese IPv6 este acela în care apar toate cele 32 de cifre hexazecimale, dar acest lucru nu înseamnă că acest format este şi cel ideal. Atunci când într-o adresă IPv6 avem multe cifre hexazecimale egale cu 0 se folosesc anumite convenţii ce simplifică scrierea lor. Vom prezenta în continuare două reguli de simplificare a scrierii adreselor IPv6. Prima regulă este aceea care spune că orice 0 înaintea altor cifre dintr-un hextet poate fi omis. Spre exemplu:

- 01CD poate fi reprezentat ca 1CD
- 0B00 poate fi reprezentat ca B00
- 001A poate fi reprezentat ca 1A
- 000F poate fi reprezentat ca F

Cea de-a doua regulă specifică faptul că semnul „::” poate înlocui orice secvenţă continuă formată dintr-unul sau mai multe “hextete” egale cu zero. Semnul „::” poate fi

folosit o singură dată în cadrul unei adrese, altfel notaţia putând conduce la ambiguităţi. Folosind această tehnică, notaţia adreselor IPv6 poate fi deseori mult simplificată faţă de notaţia clasică. Prezentăm în continuare un exemplu de adresă incorectă:

- 2002:0DC9::1234::1234

  Această adresă poate conduce la mai multe extinderi posibile ale adresei comprimate:
- 2002:0DC9::1234:0000:0000:1234
- 2002:0DC9::1234:0000:0000:0000:1234
- 2002:0DC9:0000:1234::1234
- 2002:0DC9:0000:0000:1234::1234

  Alt exemplu de scriere simplificată a unei adrese IPv6 este prezentat în tabelul 5.1.

| Reprezentare | Valoare |
|---|---|
| Adresa IPv6 în format preferat | **2002:0DC9:0000:0000:1234:0000:0000:0234** |
| Adresa IPv6 fără zerourile de început de hextet | **2002:DC9:0:0:1234:0:0:234** |
| Adresa IPv6 în format comprimat | **2002:DC9::1234:0:0:234**<br>***sau***<br>**2002:DC9:0:0:1234::234** |

Tabelul 5.1 Formate de scriere pentru o adresă IPv6

De asemenea, în tabelul 5.2 sunt prezentate mai multe formate de scriere pentru o adresă IPv6.

| Reprezentare | Valoare |
|---|---|
| Format preferat | FEC0:0000:0000:0000:0000:0000:0000:0000/10 |
| Format comprimat (1) | FEC0:0:0:0:0:0:0:0/10 |
| Format comprimat (2) | FEC0::/10 |
| Format binar | Cei mai semnificativi 10 biţi sunt setaţi la valoarea 1111 1110 11 |

Tabelul 5.2 Formate de scriere pentru o adresă IPv6

### 5.2.2 Tipuri de adrese IPv6

Există trei mari tipuri de adrese IPv6:

- ***Unicast*** – O adresă IPv6 de tip *unicast* identifică în mod unic o interfaţă a unui echipament compatibil IPv6. De regulă, adresele IPv6 sursă trebuie să fie adrese *unicast*.
- ***Multicast*** – O adresă IPv6 de tip *multicast* este folosită pentru a trimite un singur pachet IPv6 către mai multe destinaţii în acelaşi timp.
- ***Anycast*** – O adresă IPv6 de tip *anycast* reprezintă orice adresă IPv6 de tip *unicast* ce poate fi atribuită mai multor dispozitive. Un pachet de date trimis către o adresă de tip *anycast* este direcţionat către cel mai apropiat echipament ce posedă acea adresă.

Spre deosebire de IPv4, protocolul IPv6 nu are adrese de broadcast, dar există o adresă IPv6 de tipul multicast care se numeşte „all-nodes” ce poate fi folosită în acest scop.

Să ne reamintim că porţiunea de reţea a unei adrese IPv4 se poate reprezenta în formatul CIDR prin sufixul respectiv (spre exemplu, adresa 192.168.250.1 cu masca de sub-reţea 255.255.255.0 poate fi reprezentată sub forma 192.168.250.1/24 ). Asemănător, în cazul IPv6 prefixul determină porţiunea de reţea a adresei IPv6 folosind notaţia respectivă, hexazecimală cu „:”. Lungimea valorii scrisă după semnul „/” poate varia între 0 şi 128 (numărul de cifre binare al unei adrese IPv6). Un sufix IPv6 tipic folosit pentru reţele locale şi pentru alte tipuri de reţele este /64. Acest lucru semnifică faptul că porţiunea de reţea este de 64 de biţi, în timp ce restul de 64 de biţi vor identifica porţiunea de host a adresei. O adresă IPv6 unicast identifică în mod unic o interfaţă a unui echipament IPv6; un pachet de date trimis către o adresă unicast este recepţionat de către interfaţa ce are atribuită acea adresă. În mod asemănător cu IPv4, o adresă sursă IPv6 trebuie să fie o adresă unicast. O adresă destinaţie IPv6 poate fi sau adresă unicast, sau adresă multicast.

**Adrese IPv6 unicast**

Există 6 tipuri de adrese unicast:

- Adrese ***unicast globale*** – asemănătoare cu adresele publice IPv4, fiind adrese globale unice, rutabile în Internet. Adresele globale unicast pot fi configurate static sau atribuite în mod dinamic. Există unele deosebiri majore în modalitatea de atribuire dinamică a unei adrese IPv6 în comparaţie cu DHCP pentru IPv4.
- Adrese ***link-local*** – sunt adrese utilizate pentru a comunica cu alte dispozitive în aceeaşi reţea locală. În cazul IPv6, termenul de ***link*** se referă la o sub-reţea. Adresele de tip ***link-local*** sunt atribuite unui singur *link*. Unicitatea acestora se referă doar la acel link, deoarece nu sunt adrese rutabile în afara link-ului (ruterele nu trimit mai departe pachete de date cu adrese sursă sau destinaţie de tip *link-local*).
- Adresa de ***loopback*** – asemănătoare cu adresa de loopback în cazul IPv4, folosită ca adresă proprie a unui dispozitiv. Putem testa conectivitatea la reţea a unui calculator prin comanda ***ping localhost***. Adresa de loopback IPv6 este formată numai din zero-uri, cu excepţia ultimului bit şi poate fi reprezentată ca **::1/128** sau doar **::1** în format comprimat. Cu alte cuvinte, testul de loopback în IPv6 se poate face folosind comanda **ping ::1**, al cărui output este prezentat în figura 5.3.
- Adresa ***nespecificată*** – este o adresă formată numai din zerouri reprezentată ca ::/128 sau :: (în format comprimat). Nu poate fi atribuită unei interfeţe şi este utilizată doar ca adresă sursă a unui pachet IPv6. O adresă nespecificată este folosită ca adresă sursă atunci când un echipament nu are încă o adresă IPv6 permanentă sau atunci când sursa unui pachet de date este irelevantă pentru destinaţie.


Figura 5.3 Apelul comenzii ***ping*** cu adresa de ***loopback IPv6***

- Adresa ***locală unică*** – adresele IPv6 de acest tip sunt asemănătoare cu adresele IPv4 private RFC 1918, cu unele deosebiri însă. Adresele ***locale unice*** sunt folosite

pentru adresarea locală în cadrul unui site sau între un număr limitat de site-uri. Aceste adrese nu sunt rutabile la nivel global IPv6. Adresele locale unice sunt în domeniul **FC00::/7 - FDFF::/7**. În cazul IPv4 adresele private sunt combinate cu mecanismul NAT/PAT pentru a oferi translatări de adrese private/publice (datorită limitării spaţului de adresare IPv4). Multe site-uri folosesc adresarea privată RFC 1918 pentru a securiza şi a „ascunde" reţeaua de posibile riscuri externe.

- Adresă ***înglobată IPv4*** – reprezintă adrese utilizate pentru a favoriza tranziţia de la IPv4 la IPv6.

***Adrese IPv6 link-local***

O adresă de acest tip permite unui echipament să comunice cu celelalte echipamente IPv6 din aceeaşi sub-reţea (link). Pachetele ce au sursa sau destinaţia adrese de tip link-local nu pot fi direcţionate în afara sub-reţelei de origine. În cazul protocolului IPv6, aceste adrese au un rol special, deoarece fiecare interfaţă de reţea trebuie să aibă o adresă *link-local*. Dacă o adresă link-local nu este configurată manual pe o interfaţă, atunci echipamentul respectiv îşi va crea automat o astfel de adresă fără a comunica cu un server DHCP. Gazdele pe care este implementat protocolul IPv6 îşi creează o adresă de tip link-local chiar dacă echipamentului nu i s-a atribuit o adresă IPv6 unicast globală. Acest lucru permite echipamentelor IPv6 să comunice cu alte echipamente IPv6 din aceeaşi sub-reţea, inclusiv cu poarta implicită (*default gateway*). Adresele link-local se află în domeniul **FE80::/10**. Sufixul /10 semnifică faptul că primii 10 biţi sunt egali cu **1111 1110 10** (primul hextet este de forma **1111 1110 10xxx xxxx**). Domeniul de valori pentru primul hextet este: **1111 1110 1000 0000 ÷ 1111 1110 1011 1111** (**FE80 ÷ FEBH** în hexazecimal). În figura 5.4 este prezentat un exemplu de comunicaţie în sub-reţea între două echipamente ce au atribuite adresele de tip *link-local* **FE80::AAAA/64** şi **FE80::DDDD/64**.

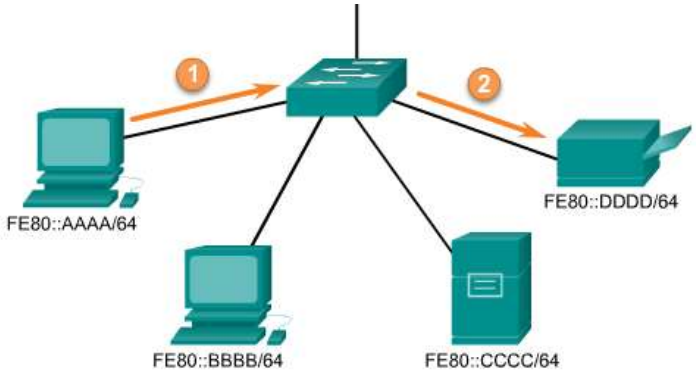



Figura 5.4 Exemplu de comunicaţie pe baza unor adrese IPv6 de tip *link-local*

Adresele IPv6 de tip *link-local* sunt utilizate, de asemenea, de către protocoalele de rutare IPv6 pentru a inter-schimba mesaje şi pentru a fi atribuite ca adrese *next-hop* într-o tabelă de rutare IPv6. În figura 5.5 este prezentat formatul unei adrese IPv6 de tip *link-local*.

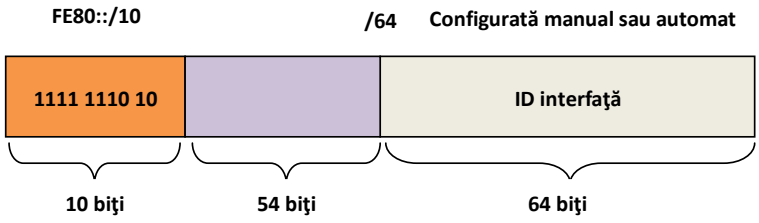



Figura 5.5 Formatul de reprezentare a unei adrese IPv6 de tip *link-local*

***Adrese IPv6 unicast globale (Global Unicast Addresses – GUAs)***

Aceste adrese sunt adrese unice şi pot fi rutate în Internetul IPv6 public, asemănătoare cu adresele IPv4 publice. Organizaţiile internaţionale *ICANN (Internet Committee for Assigned Names and Numbers), IANA (Internet Asigned Numbers Authority)* sunt responsabile cu alocarea blocurilor de adrese IPv6 celor 5 centre regionale de tip *RIR (Regional Internet Registry)*[1]. În momentul de faţă sunt alocate adrese ce au primii trei biţi egali cu **001 (2000::/3)**, ceea ce reprezintă o optime din spaţiul total de adresare IPv6, fără a ţine cont de un mic număr de adrese speciale unicast şi multicast.

Adresa 2001:0DB8::/32 este rezervată pentru documentare şi este folosită în exemplele prezentate în continuare. În figura 5.6 este prezentat formatul general al unei adrese unicast globale. Acest format este constituit din trei părţi: *prefixul global de rutare*, *identificatorul de sub-reţea (subnet ID)* şi *identificatorul de interfaţă (interface ID)*.

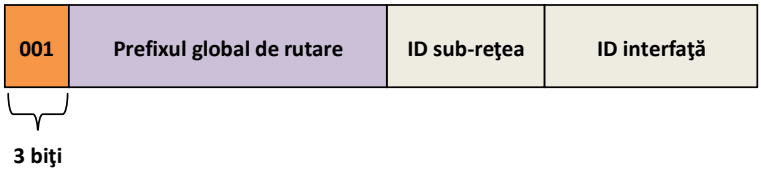



Figura 5.6. Formatul general al unei adrese IPv6 unicast globale

·***Prefixul global de rutare*** – reprezintă porţiunea de reţea a adresei atribuite de către furnizorul de servicii de reţea/Internet unui client/site. În momentul de faţă, autorităţile regionale atribuie un sufix /48 clienţilor şi astfel sunt folosite adrese IPv6/48 al cărui format este cel din figura 5.7. Cei 48 de biţi ai sufixului global de rutare împreună cu cei 16 biţi ai identificatorului de sub-reţea generează o adresă IPv6 cu sufix /64 (48 + 16 = 64). Spre exemplu, adresa IPv6 2001:ODB8:AAAA::/48 are un sufix ce ne arată că primii 48 de biți (3 hexteţi) reprezintă porțiunea de rețea a adresei. Reamintim că semnul „::” de la sfârşitul adresei semnifică faptul că ultimii biţi ai adresei au valoarea zero.

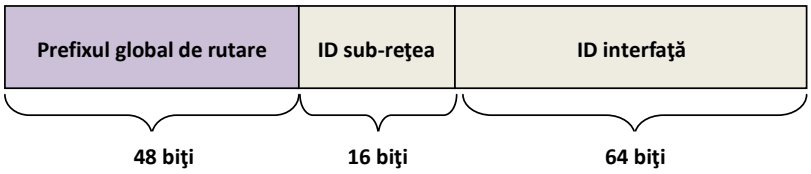



Figura 5.7. Adresă IPv6 globală cu sufix /48

[1] Cele 5 centre regionale RIR sunt:

- *AfriNIC (African Network Information Center)* pentru Africa;
- *ARIN (American Registry for Internet Numbers)* pentru SUA, Canada, Antartica şi o parte din zona Caraibelor;
- *APNIC (Asia-Pacific Network Information Center)* pentru Asia, Australia, Noua Zeelandă şi ţările învecinate;
- *LACNIC (Latin America and Caribbean Network Information Center)* pentru America Latină şi o parte din zona Caraibelor;
- *RIPE NCC (Réseaux IP Européens Network Coordination Centre)* pentru Europa, Rusia, Orientul Mijlociu şi Asia Centrală.

·***Identificatorul de sub-reţea*** – este folosit în interiorul unei organizaţii pentru a specifica sub-reţeaua din care face parte acea adresă.

·***Identificatorul de interfaţă (interface-ID)***– acesta este echivalent cu porţiunea de host dintr-o adresă IPv4. În acest caz se utilizează termenul de interfaţă deoarece un singur host poate avea mai multe interfeţe, fiecare interfaţă fiind identificată printr-una sau mai multe adrese IPv6. În cazul adreselor IPv6 pot fi atribuite şi adrese de host ce conţin toate valorile zero sau unu în această porţiune. În cazul tuturor biţilor egali cu zero adresa este însă rezervată pentru tipul *„subnet router anycast”* şi trebuie atribuită doar ruterelor. Se recomandă ca în majoritatea cazurilor să fie folosite subrețele /64, ceea ce creează un interface-ID de 64 de biți. Un interface-ID pe 64 biți oferă posibilitatea alocării a 18 cvintilioane de echipamente (host-uri) per subrețea.

## 5.3 Îndrumar pentru autoverificare

### 5.3.1 Sinteza unității de studiu 5

Protocolul IPv6 reprezintă viitorul adresării în rețeaua globală Internet. Pe lângă un spațiu de adresare mult mai mare, IPv6 aduce față de IPv4 o structură a headerului simplificată, facilitând rutarea pachetelor de date în Internet.

### 5.3.2 Concepte și termeni de reținut

| | |
|---|---|
| *Rutare în Internet* | *Adresă global unicat* |
| *IPv6* | *Rutare* |
| *Adresă link local* | *Identificator de interfață* |
| *IANA* | *RIPE* |
| *Reprezentare prin hexteți* | *APNIC* |

### 5.3.3 Întrebări pentru autoverificare

***Întrebarea 1.***
Cum se scrie prescurtat adresa IPv6 2001:0404:0001:1000:0000:0000:0EF0:BC00 ?

*Răspuns:* 2001:404:1:1000::EF0:BC00

***Întrebarea 2.*** Ce tip de adresă IPv6 nu este rutabilă și este folosită doar pentru comunicația în cadrul unei singure subrețele?

a) Adresă unică locală
b) Adresă loopback
c) Adresă nespecificată
d) Adresă global unicast
e) Adresă link-local

*Răspuns: e*

***Întrebarea 3.*** Rețelei organizației dumneavoastră i s-a asociat adresa IPv6 2001:db8:130f::/48 de către furnizorul de servicii de rețea. Cu acest prefix, câți biți sunt disponibili pentru organizație pentru a crea /64 subrețele dacă biții interface ID nu sunt împrumutați?

a) 128
b) 16
c) 8
d) 80
e) 10

*Răspuns:b*

### Întrebări de control și teme de dezbatere

1. Cât timp credeți că se va continua folosirea IPv4 în dualitate cu IPv6?
2. Căutați pe web pentru a vedea care este procentajul actual de utilizare a protocolului IPv6.

## 5.3.4 Bibliografie obligatorie

1. Răzvan Daniel Zota, Rețele de calculatoare, capitolul 5, Ed. ASE, 2013.

# UNITATEA DE STUDIU 6. Nivelul transport

## 6.1 Introducere

Având în vedere poziţionarea sa, nivelul transport reprezintă nivelul de mijloc în cadrul ierarhiei nivelelor modelului ISO-OSI. În acest context, nivelele aplicaţie, prezentare şi sesiune sunt considerate nivelele superioare iar nivelele reţea, legătură de date şi fizic reprezintă nivelele inferioare. Nivelul transport are funcţia specială de asigurare a serviciilor de comunicare direct către procesele asociate aplicaţiilor ce rulează pe diverse calculatoare gazdă. Protocoalele ce acţionează la nivelul transport asigură organizarea logică a comunicaţiei între aplicaţii. Cu toate că, de regulă, procesele asociate aplicaţiilor nu se află pe dispozitive direct interconectate fizic, modalitatea de lucru a protocoalelor nivelului transport face ca acestea să pară interconectate în mod direct. Astfel, procesele asociate aplicaţiilor folosesc comunicaţia logică oferită de protocoalele nivelului transport pentru a-şi trimite mesaje între ele, fără a fi preocupate de detalii ale infrastructurii fizice ce este utilizată pentru transportul efectiv al datelor.

În mod practic, un calculator conectat la reţea poate dispune de mai multe protocoale de transport disponibile pentru diverse aplicaţii de reţea. Spre exemplu, reţeaua Internet foloseşte două protocoale de transport de bază: *TCP (Transmission Control Protocol)* şi *UDP (User Datagram Protocol)*; fiecare dintre aceste două protocoale oferă servicii de transport diferite pentru aplicaţii. În general, toate protocoalele de nivel transport oferă câteva servicii de bază aplicaţiilor, dintre care enumerăm: multiplexarea/demultiplexarea, transferul fiabil al datelor, garantarea lăţimii de bandă şi administrarea întârzierilor.

Nivelul transport poate asigura segmentarea datelor (împărţirea datelor în segmente) precum şi controlul necesar pentru a reasambla segmentele create în ordinea corectă în momentul ajungerii lor la destinaţie. Funcţiile de bază ale nivelului transport sunt următoarele:

- Urmărirea secvenţelor individuale de comunicaţie între aplicaţii de la sursă către destinaţie;
- Segmentarea datelor şi administrarea segmentelor de date;
- Reasamblarea la destinaţie a segmentelor în fluxuri de date;
- Identificarea aplicaţiilor comunicante.
-

***Urmărirea secvenţelor de comunicaţie***

Un calculator conectat într-o reţea poate avea mai multe aplicaţii ce comunică în reţea. Fiecare dintre aceste aplicaţii comunică la rândul lor cu diverse aplicaţii ce rulează pe calculatoare diferite. Una dintre responsabilităţile principale ale nivelului transport este aceea de a asigura existenţa mai multor fluxuri de comunicaţie între diverse aplicaţii.

***Segmentarea datelor***

Având în vedere că fiecare aplicaţie generează un flux de date ce este transmis către altă aplicaţie ce rulează pe un calculator la distanţă, datele ce urmează a fi transmise prin intermediul mediului de comunicaţie către destinaţie, trebuie împărţite în segmente de date ce pot fi administrate mai uşor. Astfel, protocoalele nivelului transport descriu servicii de segmentare (împărţire în segmente) a datelor provenite de la nivelul superior, incluzând metoda de încapsulare necesară pentru fiecare segment de date. Fiecare datagramă ce provine de la nivelul superior necesită un header ce trebuie adăugat la nivelul transport pentru a indica tipul de comunicaţie necesar.

***Reasamblarea segmentelor***

În momentul în care ajung la destinaţie, datele trebuie direcţionate către aplicaţia corespunzătoare şi, de asemenea, trebuie rearanjate într-un flux de date complet pentru a fi transmise mai departe către nivelul superior. Protocoalele nivelului transport descriu modalitatea în care informaţiile conţinute în header-ul de transport sunt folosite pentru rearanjarea (reasamblarea) datelor în fluxuri de date ce vor fi transmise mai departe către nivelul aplicaţie.

***Identificarea aplicaţiilor comunicante***

Pentru a asigura la destinaţie transmisia datelor către aplicaţia corespunzătoare, nivelul transport trebuie să stabilească o modalitate de identificare a aplicaţiei destinaţie. În acest scop, nivelul transport atribuie un identificator aplicaţiei, identificator ce reprezintă un *număr de port* de comunicaţie. Fiecărui proces implicat în comunicaţia în reţea i se atribuie un număr de port unic pentru echipamentul respectiv. Acest număr de port este utilizat în cadrul header-ului transport pentru a indica cărei aplicaţii îi sunt asociate datele respective.

Nivelul transport reprezintă legătura dintre nivelul aplicaţie din modelul TCP/IP şi nivelul inferior responsabil pentru transmiterea datelor în reţea (figura 6.1). Acest nivel primeşte date provenite de la mai multe conversaţii pe care le transmite mai departe nivelelor inferioare ca segmente de date ce pot fi administrate şi eventual *multiplexate* pe mediul de comunicaţie.

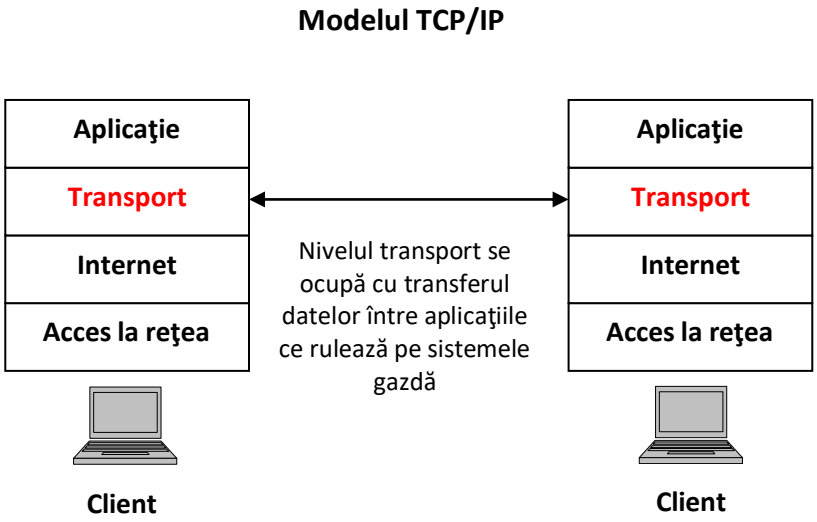



Figura 6.1 În modelul TCP/IP, nivelul transport asigură legătura dintre nivelul aplicaţie şi nivelul Internet

Aplicaţiile nu trebuie să cunoască detaliile operaţionale ale reţelei pe care o folosesc pentru comunicaţie. Aplicaţiile generează date care sunt transmise către alte aplicaţii, fără a ţine cont de tipul echipamentului destinaţie, de mediul de comunicaţie, de ruta pe care circulă datele către destinaţie, de gradul de congestie al reţelei, de dimensiunea reţelei, etc. De asemenea, nivelele inferioare nu cunosc detalii legate de faptul că mai multe aplicaţii transmit date de-a lungul reţelei iar funcţia lor este aceea de a transmite datele către echipamentul specificat. Nivelul transport se ocupă cu rearanjarea segmentelor de date înainte ca acestea să fie transmise către aplicaţia corespunzătoare. Datorită faptului că diversele aplicaţii pot avea diferite necesităţi cu privire la asigurarea transportului datelor către destinaţie, există mai multe protocoale

ce funcţionează la nivelul transport. În cazul anumitor aplicaţii segmentele de date trebuie să ajungă la destinaţie într-o ordine specificată pentru a fi procesate în mod corect. În unele cazuri, *toate datele* trebuie să fie recepţionate pentru ca acestea să poată fi procesate, iar în alte cazuri aplicaţiile pot tolera anumite pierderi de date intervenite în timpul unei transmisiuni în reţea.

Transmiterea unui anumit tip de date (un flux video, spre exemplu) de-a lungul mediului de comunicaţie poate utiliza întreaga lăţime de bandă disponibilă, ceea ce va face imposibilă existenţa transmisiunii altor tipuri de date în reţea. În figura 6.2 putem vedea cum împărţirea datelor în fragmente mai mici (segmente) permite existenţa mai multor tipuri de comunicaţie în reţea, din partea mai multor utilizatori, prin intercalarea (***multiplexarea***) acestor fragmente pe acelaşi mediu de comunicaţie. Fără existenţa procesului de segmentare a datelor nu am putea recepţiona email-uri, schimba mesaje pe chat sau vizualiza pagini web în timp ce urmărim un filmuleţ video, spre exemplu. Pentru a identifica fiecare segment de date, nivelul transport adaugă un header ce conţine informaţii binare, mai precis câmpuri de biţi. Valorile acestor câmpuri permit diferitelor protocoale de nivel transport să exercite anumite funcţiuni pentru administrarea comunicaţiilor de date.

Deoarece reţelele de astăzi sunt reţele convergente de date, voce şi video, aplicaţii cu nevoi diferite comunică pe baza aceleiaşi infrastructuri de reţea. Diferitele protocoale de nivel transport au reguli diferite ce permit echipamentelor să trateze în mod diferite cerinţele aplicaţiilor. Unele protocoale oferă funcţionalităţile de bază pentru transmiterea eficientă a datelor către destinaţie; de regulă, acest tip de protocoale sunt folosite în cazul aplicaţiilor în timp real, pentru care viteza de transmisie primează. Alte protocoale de transport descriu procedee suplimentare ce oferă funcţionalităţi suplimentare, oferind o fiabilitatea crescută transportului datelor între aplicaţii. Aceste funcţionalităţi suplimentare presupun însă creşterea volumului de date transmise rezultând necesităţi crescute ale lăţimii de bandă.

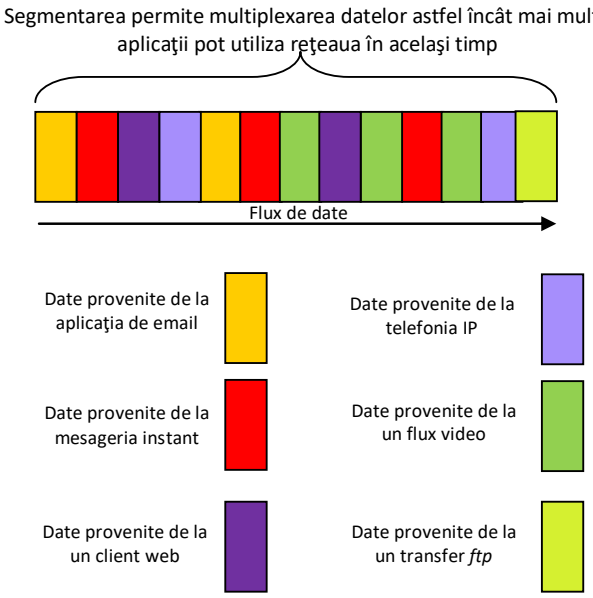



Figura 6.2 Multiplexarea datelor provenite de la aplicaţii diferite

## 6.2 Obiectivele și competențele unității de studiu

*Obiectivele unității de studiu*:

- Prezentarea funcționalităților nivelului transport din modelul ISO-OSI;
- Prezentarea caracteristicilor protocolului TCP;
- Prezentarea caracteristicilor protocolului UDP.

*Competențele unității de studiu*:

- Studenții vor putea să definească conceptele de bază întâlnite în cadrul nivelului transport;
- Studenții vor cunoaște detalii legate de funcționarea protocoalelor TCP și UDP.

***Durata medie de studiu individual alocat unității:*** 4 ore

## 6.3 Conținutul unității de studiu

### 6.3.1 Protocolul TCP

Descris iniţial în RFC 793, protocolul TCP este un protocol fiabil, orientat pe conexiune. În cazul unei comunicaţii orientate pe conexiune, înainte ca transferul de date să înceapă, se stabileşte o conexiune virtuală între cei doi parteneri de comunicaţie. Faza iniţială de stabilire a sesiunii de comunicaţie pregăteşte pentru comunicaţie entităţile implicate, astfel că în această etapă se negociază cantitatea de date care poate fi transmisă la un moment dat iar sesiunea este încheiată doar după ce toate datele au fost transmise. TCP este protocolul responsabil cu: împărţirea mesajelor în segmente, numerotarea acestora, reasamblarea (aranjarea în ordinea corectă) lor la destinaţie şi refacerea mesajelor transmise iniţial. De asemenea, în sarcina protocolului TCP revine şi retransmiterea segmentelor ce nu au fost recepţionate la destinaţie.

***Controlul fluxului*** este asigurat tot în cadrul nivelului transport; atunci când disponibilitatea lăţimii de bandă este limitată, TCP trimite o cerere de reducere a fluxului de date transmis. În acest mod, protocolul TCP reglează dinamic cantitatea de date pe care o sursă o transmite către destinaţie. Prin controlul fluxului se poate preveni pierderea de segmente de date transmise în reţea şi astfel se poate evita retransmisia acestora.

#### *Formatul header-ului TCP*

Faptul că TCP este un protocol orientat pe conexiune presupune că staţiile implicate în comunicaţie sunt permanent în alertă cu privire la starea conexiunii. Un exemplu de comunicaţie orientată pe conexiune poate fi acela al unei conversaţii telefonice uzuale, pentru care „protocolul” de comunicaţie implică faptul că iniţierea conversaţiei se face prin cuvântul „Alo!”. Printre aplicaţiile ce folosesc protocolul TCP se numără poşta electronică, transferul de fişiere şi navigatoarele web.

În figura 6.3 este prezentat header-ul TCP ce conţine câmpurile necesare pentru controlul conversaţiei între staţiile implicate în comunicaţie. De asemenea, protocolul TCP este considerat un protocol dinamic (în engleză: *stateful protocol*), având în vedere faptul că din momentul iniţierii unei sesiuni de comunicaţie se ţine seama continuu de starea acesteia. Spre exemplu, atunci când sunt transmise date folosind TCP, expeditorul aşteaptă confirmări de

primire al acestora din partea destinatarului. TCP ţine cont de datele trimise şi de cele confirmate. Dacă nu există confirmări de primire, expeditorul presupune că datele nu au ajuns la destinaţie şi le va retransmite. Sesiunea dinamică de comunicaţie durează din momentul iniţializării conexiunii până în momentul încheierii conexiunii. Informaţiile legate de starea conexiunii nu sunt necesare în cazul unui protocol neorientat pe conexiune, cum este, spre exemplu, protocolul UDP.

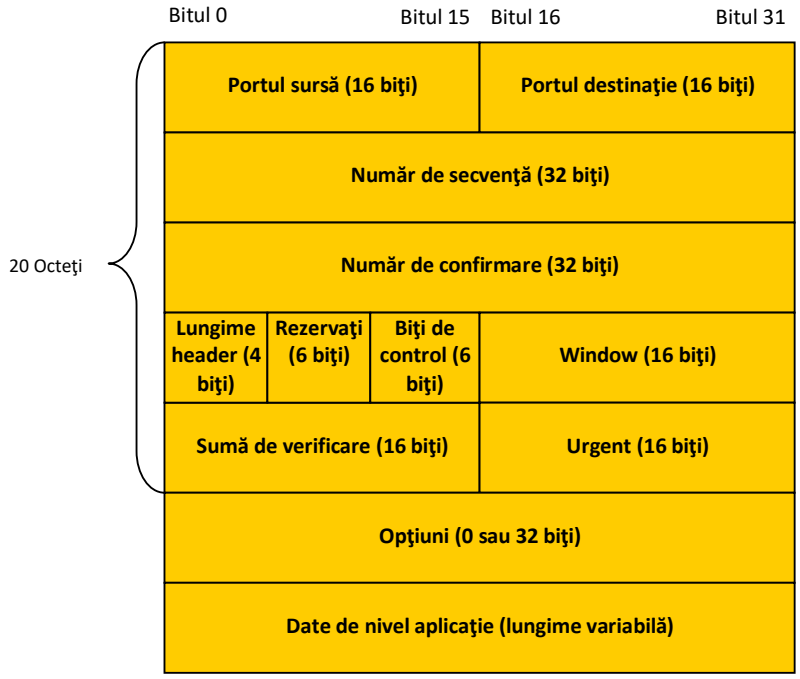



Figura 6.3. Header-ul TCP

După cum putem observa în figura 6.3, TCP implică informaţii suplimentare grupate în câmpurile prezentate, câte 20 de octeţi pentru fiecare segment TCP (spre deosebire de cei doar 8 octeţi prezenţi în header-ul UDP). Aceste câmpuri sunt următoarele:

- **Numărul de secvenţă** (32 de biţi) – folosit pentru reasamblarea datelor la destinaţie
- **Numărul de confirmare** (32 de biţi) – indică recepţionarea datelor
- **Lungimea header-ului** (4 biţi) – indică lungimea header-ului segmentului TCP (se mai numeşte *data offset*)
- **Biţi rezervaţi** (6 biţi) – rezervaţi pentru posibile utilizări viitoare
- **Biţi de control** (6 biţi) – câmp ce conţine biţi de codificare (indicatori) ce semnifică rolul şi funcţia segmentului TCP
- **Window** (16 biţi) – dimensiunea „ferestrei” – reprezintă numărul de segmente ce pot fi transmise/recepţionate la un moment dat, fără a se primi o confirmare.
- **Sumă de verificare** (16 biţi) – reprezintă câmpul folosit pentru verificare datelor din header şi datele de nivel aplicaţie

**Urgent** (16 biţi) – indică faptul că datele sunt urgente sau nu..

### 6.3.2 Protocolul UDP

.

Protocolul *User Datagram Protocol (UDP)* este un protocol documentat în RFC 768, fiind un protocol neorientat pe conexiune, nefiabil. Protocoalele de acest tip se numesc, de obicei, protocoale de tip „*best-effort delivery*", care nu garantează ajungerea sigură a datelor la destinaţie, dar care „fac tot ce pot" pentru a transmite corect datele către destinaţie. Pentru a înţelege cel mai bine acest concept, putem face o analogie cu modul de expediere a unei scrisori clasice folosind serviciile poştale. În acest sens, avem de regulă două variante: fie să trimitem scrisoarea aplicând un simplu timbru şi introducând-o în cutia poştală, fie să plătim mai mult şi să trimitem o scrisoare cu confirmare de primire, caz în care vom primi o confirmare a faptului că scrisoarea noastră a ajuns (sau nu) la destinatar.
Primul caz, în care nu vom primi nicio confirmare de primire şi în care avem doar speranţa că serviciile poştale îşi vor face „datoria" şi vor livra scrisoarea la destinatar, este asemănător cu ceea ce se întâmplă în cazul protocolului UDP. Acest tip de serviciu poartă denumirea de serviciu de tip *„best-effort"*, adică se face tot posibilul ca serviciul să funcţioneze normal, dar nu se asigură *niciun fel de garanţii*.
Al doilea caz, cel al trimiterii unei scrisori cu confirmare de primire, seamănă cu modul de funcţionare al protocolului TCP: fiabil, cu confirmări de primire ce ne *oferă garanţia* că datele au ajuns corect la destinaţie. Există şi un *„preţ"* plătit pentru acest lucru: dimensiunea mare a datelor suplimentare introduse în header-ul TCP (20 de octeţi faţă de cei 6 octeţi în cazul UDP).
În aceste condiţii, UDP este un protocol de transport ce oferă segmentarea datelor şi reasamblarea lor la destinaţie, fără a asigura fiabilitatea şi controlul fluxului (asigurate de TCP). Caracteristicile generale ale protocolului UDP sunt următoarele:

- ***Neorientare pe conexiune*** – nu stabileşte o conexiune virtuală între expeditor şi receptor înaintea transmisiei datelor;
- ***Livrare nefiabilă*** – UDP nu oferă servicii ce asigură faptul că datele vor fi transmise în mod fiabil. Nu există mecanisme ale protocolului UDP prin care transmiţătorul poate retrimite datele dacă acestea nu ajung corect la destinaţie;
- ***Reasamblarea datelor fără ordonarea lor*** – UDP nu oferă niciun procedeu de reasamblare a datelor în secvenţa originală. Datele sunt transmise către nivelul aplicaţie în ordinea în care au fost recepţionate;
- ***Nu există controlul fluxului*** – UDP nu are mecanisme prin care să se controleze cantitatea de date trimisă de o sursă pentru a preveni cazurile de supraîncărcare cu date a destinaţiei. Dacă destinaţia devine supraîncărcată, datele trimise ulterior se vor pierde până în momentul în care lucrurile vor reveni la normal. Nu există un mecanism de retransmisie automată a datelor pierdute precum în cazul TCP.

#### *Formatul header-ului UDP*

Formatul simplu al header-ului UDP este prezentat în figura 6.4. Cu toate că acest format nu include mecanisme ce asigură fiabilitatea şi controlul fluxului (ca în cazul TCP), faptul că UDP are puţine date suplimentare (*overhead*) face ca acesta să fie protocolul ideal de transport pentru aplicaţii ce pot avea câteva pierderi de date dar pentru care viteza este extrem de importantă. Este vorba aici despre aplicaţii în timp real, de genul telefoniei IP sau a transmisiilor fluxurilor video, unde viteza este mult mai importantă decât fiabilitatea (imaginaţi-vă o convorbire prin telefonia IP în care întârzierile de transmisie dintre

interlocutori să fie de domeniul zecilor de secunde!). Mesajele comunicate prin intermediul UDP sunt împărţite în datagrame UDP ce sunt transmise către destinaţie prin intermediul comunicaţiei de tip *„best effort”*, prezentată anterior. Cele mai cunoscute aplicaţii ce folosesc protocolul UDP ca protocol de transport sunt: *DNS (Domain Name System), VoIP (Voice over IP)* şi transmiterea de fluxuri video *(video streaming)*.

UDP este un protocol de tip *stateless*, adică un protocol ce nu ţine cont de starea legăturii (conexiunii). Nici clientul, nici serverul nu sunt obligate să aibă informaţii legate de starea sesiunii de comunicaţie la un moment dat. Având în vedere că header-ul UDP nu conţine informaţii legate de fiabilitate şi controlul fluxului, datele se pot pierde în drumul lor către destinaţie sau pot fi recepţionate greşit. În cazul în care este nevoie de fiabilitate pentru datele transmise, fie se foloseşte protocolul TCP, fie, dacă se foloseşte UDP, fiabilitatea va fi asigurată de nivelul superior (nivelul aplicaţie).

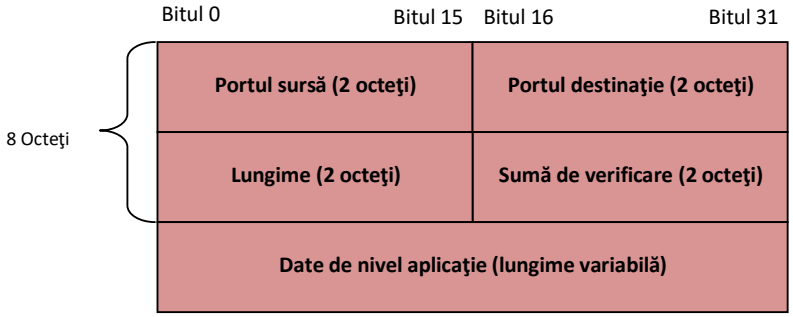



Figura 6.4. Header-ul UDP

## 6.4 Îndrumar pentru autoverificare

### 6.4.1 Sinteza unității de studiu 6

Nivelul transport încapsulează datele provenite de la nivelul sesiune pentru a le pregăti în vederea asigurării transportului acestora către destinație. Protocoalele de bază ce acționează la acest nivel sunt TCP și UDP. În timp ce TCP este un protocol fiabil, orientat pe conexiune, protocolul UDP este un protocol nefiabil, neorientat pe conexiune, potrivit pentru aplicațiile care necesită un overhead minim (și o viteză sporită) în comunicația din rețea.

### 6.4.2 Concepte și termeni de reținut

| | |
|---|---|
| *Nivelul transport* | *TCP* |
| *UDP* | *Segmentare* |
| *Segmente de date* | *Datagrame* |
| *Aplicații real-time* | *Numere de porturi* |
| *Multiplexare* | *Orientare pe conexiune* |

### 6.4.3 Întrebări pentru autoverificare

***Întrebarea 1.*** Ce caracteristică a nivelului transport este utilizată pentru a stabili o sesiune orientată pe conexiune?

a) UDP Ack flag
b) 3-way handshake
c) TCP port number
d) UDP port number

*Răspuns: b*

***Întrebarea 2.*** Care este intervalul de valori pentru bine cunoscutele porturi TCP și UDP?

a) 0-1023
b) 0-255
c) 0-1000
d) 1024-49151

*Răspuns: a*

***Întrebarea 3.*** Ce tip de aplicații sunt potrivite pentru folosirea UDP?

a) Aplicații ce necesită retransmisia segmentelor pierdute de date
b) Aplicații ce necesită o transmisie fiabilă
c) Aplicații ce sunt sensibile la întârzieri
d) Aplicații ce sunt sensibile la pierderile de pachete de date

*Răspuns:c*

#### Întrebări de control și teme de dezbatere

1. Care sunt funcțiile de bază ale nivelului transport?
2. Descrieți procedeul de fereastră glisantă utilizat de către protocolul TCP.
3. Considerând un flux de streaming audio, ce protocol de transport considerați că trebuie folosit: TCP sau UDP?

### 6.4.4 Bibliografie obligatorie

1. Răzvan Daniel Zota, Rețele de calculatoare, capitolul 4, Ed. ASE, 2013.

# UNITATEA DE STUDIU 7. Nivelul aplicație



## 7.1 Introducere

Nivelul aplicaţie din modelul ISO-OSI are legătură cu aplicaţiile de reţea ce rulează pe un dispozitiv (calculator sau alt echipament). În cadrul modelului TCP-IP, nivelul aplicație corespunde funcțional cu cele trei nivele superioare din modelul ISO-OSI: nivelul sesiune, nivel prezentare și nivelul aplicație.

De regulă, cele mai cunoscute aplicaţii de reţea includ accesul pe un calculator la distanţă, poşta electronică, transferul de fişiere, chat-ul, web-ul, telefonia Internet, video conferinţa, schimbul, de fişiere în reţele peer-to-peer, etc. Cu toate că aplicaţiile de reţea diferă mult între ele, partea de software se află totdeauna în prim plan. Software-ul ce ţine de aplicaţia de reţea este distribuit între două sau mai multe sisteme terminale (calculatoare gazdă). Spre exemplu, în cazul aplicaţiei (serviciului) web există două piese de software ce comunică între ele: aplicaţia de tip browser instalată pe calculatorul gazdă şi aplicaţia server (software-ul) web instalată pe serverul (fizic) de web. Şi în cazul aplicaţiei de conexiune la distanţă - *telnet*, întâlnim din nou două piese software: software-ul (de tip client) instalat pe clientul local şi software-ul (de tip server) instalat pe calculatorul la distanţă. Majoritatea aplicaţiilor de reţea folosesc pentru comunicaţia între ele modelul client-server (figura 7.1), prin care unul sau mai mulţi clienţi simultan trimit cereri unui server care oferă servicii (de conexiune şi comunicaţie) clienţilor, de regulă pe baza unei autentificări.

O singură aplicaţie poate presupune servicii de nivel aplicaţie diferite, astfel încât ceea ce pentru utilizator apare ca o simplă cerere pentru o pagină web,poate fi compusă din zeci de cereri individuale, iar pentru fiecare cerere în parte trebuie să se execute mai multe procese. Spre exemplu, un client poate avea nevoie de mai multe procese individuale pentru a iniţia o singură cerere către un server.

Mai mult, serverele au în mod obişnuit mai multe cereri client simultane. Spre exemplu, un server *ssh* poate avea mai multe cereri client de conexiune în acelaşi timp. Aceste cereri individuale trebuie să fie tratate în mod individual şi simultan pentru asigurarea comunicaţiei în reţea. Procesele și serviciile nivelului aplicaţie se bazează pe suportul nivelelor inferioare pentru a administra eficient mai multe conversaţii în acelaşi timp. Cu toate că datele sunt transferate, de regulă, dinspre server spre client, există şi date transferate dinspre client spre server; uneori acest din urmă flux de date poate fi mai masiv decât cel dinspre server spre client. Spre exemplu, un client poate transfera un fişier către server pentru stocare. Transferul de date dinspre client spre server se numeşte *upload (încărcare)* în timp ce transferul de date dinspre server spre client se numește *download (descărcare)*.

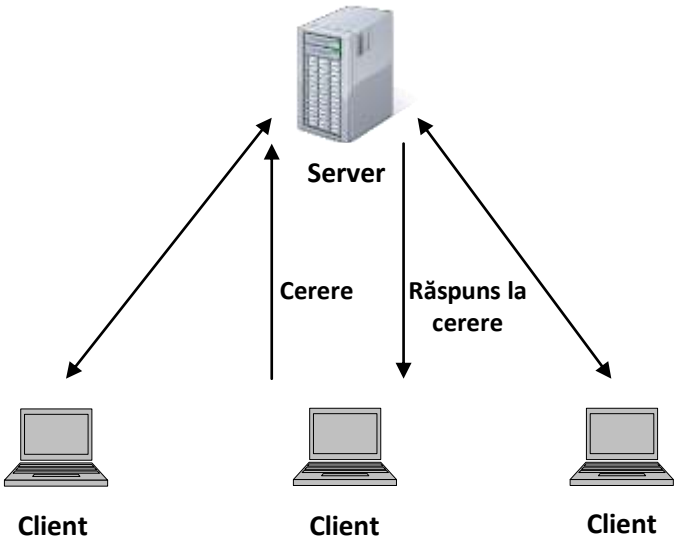



Figura 7.1 Modelul de comunicaţie client-server

În cazul modelului client-server, pe server rulează un *serviciu (proces),* uneori denumit *daemon* (denumirea iniţială provine de la procesele de tip daemon ce rulează pe servere UNIX/Linux). Precum majoritatea serviciilor, procesele de tip daemon rulează în fundal şi nu se află sub controlul direct al utilizatorului. Procesele de tip daemon sunt descrise ca „ascultând" cereri din partea clienţilor, fiind programaţi să răspundă de fiecare dată când serverul primeşte o cerere de serviciu din partea unui client. În momentul în care procesul de tip daemon descoperă o cerere venită din partea unui client, iniţiază schimbul de mesaje cu clientul conform protocolului de comunicaţie utilizat şi transmite datele necesare clientului în formatul corespunzător.

În afara modelului client-server de comunicaţie în reţea, există şi modelul *peer-to-peer* (pereche). Acesta are două forme de reprezentare: designul de reţea *peer-to-peer* şi aplicaţiile peer-to-peer (P2P). Ambele forme prezintă caracteristici similare dar în practică lucrează în mod diferit.

Reţelele peer-to-peer sunt compuse din două sau mai multe calculatoare conectate într-o reţea şi pot partaja diverse resurse fără existenţa unui server dedicat. Fiecare echipament conectat (denumit *peer*) poate funcţiona, pe rând, atât drept client cât şi ca server. Spre exemplu, un calculator poate avea rolul de server pentru o tranzacţie şi rolul de client în alt caz. Rolurile de client sau de server sunt stabilite pe baza cererilor. Cel mai simplu exemplu de reţea peer-to-peer poate fi o reţea în care sunt conectate două calculatoare ce partajează aceeaşi imprimantă. În acest caz, fiecare utilizator poate folosi computerul pentru a partaja fişiere, a juca jocuri în reţea sau pentru a partaja o conexiune Internet.

Un alt exemplu îl poate constitui o reţea mai mare în care două calculatoare folosesc aplicaţii de reţea pentru partajarea resurselor. Spre deosebire de modelul client-server ce foloseşte servere dedicate, reţelele pereche asigură descentralizarea resurselor unei reţele. În loc ca informaţia să fie partajată pe servere dedicate, aceasta poate fi oriunde în reţea, pe orice echipament conectat. Majoritatea sistemelor de operare de astăzi asigură servicii de partajare a fişierelor şi imprimantelor fără necesitatea unui software adiţional. Deoarece reţelele peer-to-peer nu necesită, de regulă, conturi de acces sau alte informaţii centralizate, securitatea este mai greu de realizat în astfel de reţele. Conturile utilizator şi drepturile de acces trebuie configurate pe fiecare calculator în parte conectat la respectiva reţea peer-to-peer.

Unul dintre cele mai cunoscute exemple de aplicaţii peer-to-peer îl reprezintă aplicaţiile

de tip *file sharing* (partajare de fişiere); în acest sens, protocolul *Gnutella* este unul dintre cele mai cunoscute protocoale folosite în reţelele de partajare de fişiere.

## 7.2 Obiectivele și competențele unității de studiu

*Obiectivele unității de studiu*:

- Prezentarea funcționalităților nivelului aplicație din modelul ISO-OSI;
- Prezentarea caracteristicilor protocoalelor de nivel aplicație;
- Prezentarea aplicațiilor, serviciilor și proceselor ce activează la nivel aplicație;
- Prezentarea serviciului DNS.

*Competențele unității de studiu*:

- Studenții vor putea să definească conceptele de bază întâlnite în cadrul nivelului aplicație;
- Studenții vor cunoaște detalii legate de funcționarea protocoalelor de la nivelul aplicație.

***Durata medie de studiu individual alocat unității:*** 4 ore

## 7.3 Conținutul unității de studiu

### 7.3.1 Aplicații, servicii și procese

Funcţiile asociate cu nivelul aplicaţie permit utilizatorilor umani să interacţioneze cu reţeaua de date. La nivelul aplicaţie avem de-a face cu trei noţiuni distincte, dar legate între ele: aplicaţii, servicii şi procese. Atunci când deschidem un browser web sau o fereastră prin care trimitem un email, o aplicaţie este pornită, iar aceasta este încărcată în memorie de unde este executată. O instanţă a unui program aflat în execuţie se numeşte *proces*. Pe un calculator există, de regulă, două tipuri de programe software ce oferă acces la datele din reţea: aplicaţii şi servicii.

Aplicaţiile de reţea sunt programele software folosite de utilizatori pentru a comunica în reţea; acestea implementează protocoalele nivelului aplicaţie şi pot comunica direct cu nivelele inferioare din stiva de protocoale. Navigatoarele web şi aplicaţiile de email sunt cele mai utilizate tipuri de aplicaţii de reţea. În figura 7.2 observăm în lista de programe încărcate în memorie browserul (aplicaţia) Internet Explorer.

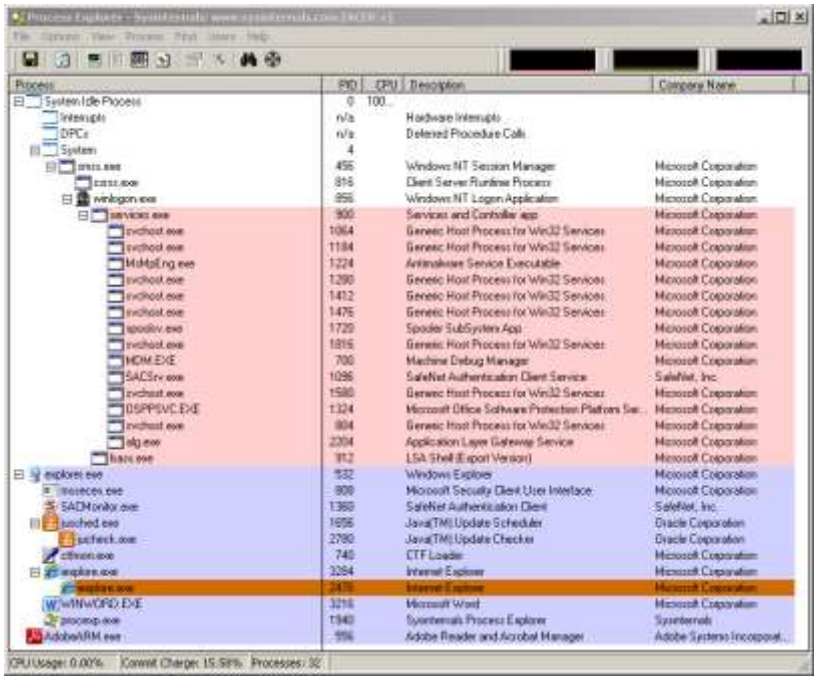


Figura 7.2 Aplicaţia Internet Explorer apare în lista de programe deschise

### Servicii

Există şi programe ce au nevoie de asistenţa serviciilor nivelului aplicaţie pentru a utiliza resursele de reţea, precum transferul de fişiere sau tipărirea în reţea. Aceste servicii sunt transparente faţă de utilizator, asigură interfaţa cu reţeaua şi pregătesc datele pentru transferul în reţea. Diferite tipuri de date (text, grafică, audio, video) necesită diferite servicii de reţea pentru a asigura faptul că datele sunt pregătite corespunzător pentru procesarea acestora de funcţiile nivelelor inferioare din modelul ISO-OSI. În figura 7.3 *alg.exe* este un serviciu ce rulează în cazul sistemului de operare Windows.

Fiecare aplicaţie sau serviciu de reţea foloseşte protocoale ce definesc standardele şi formatele de date ce vor fi utilizate. Fără existenţa protocoalelor, datele din reţea nu au o modalitate de formatare sau direcţionare. Pentru a înţelege funcţiile diferitelor servicii de reţea este necesară înţelegerea protocoalelor ce stau la baza operării acestora.

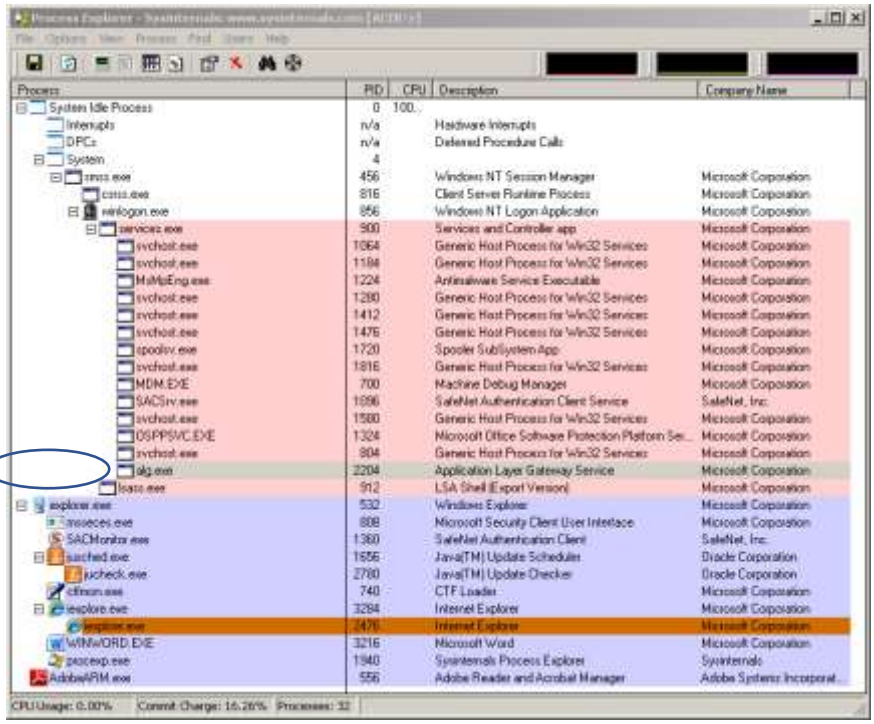



Figura 7.3. În lista de procese active apare serviciul de reţea alg.exe

De asemenea, pentru că un proces reprezintă doar *o instanţă* a unui program aflat în execuţie, există posibilitatea ca un program să ruleze de mai multe ori, existând astfel mai multe instanţe (procese) ale acestuia în memorie. Spre exemplu, putem observa în figura 7.4 că *svchost.exe* apare încărcat în memorie de 8 ori (8 procese active, deci 8 instanţe de execuţie diferite).

În concluzie, nivelul aplicaţie foloseşte protocoale ce sunt implementate în cadrul aplicaţiilor şi serviciilor. În timp ce aplicaţiile oferă utilizatorilor o modalitate de a crea mesaje ce vor fi transmise în reţea iar serviciile nivelului aplicaţie stabilesc interfaţa cu reţeaua, protocoalele definesc regulile şi formatele ce guvernează modul în care are sunt reprezentate datele. Toate cele trei componente pot fi utilizate de către un singur program executabil şi pot avea chiar şi acelaşi nume. Spre exemplu, atunci când vorbim despre *ftp*, ne putem referi la *aplicaţia de ftp*, la *serviciul ftp* sau la *protocolul ftp*.

În modelul ISO-OSI aplicaţiile interacţionează direct cu utilizatorii ce se consideră a fi în vârful stivei de protocoale. Având în vedere că în acest model fiecare nivel oferă servicii nivelului imediat superior, nivelul aplicaţie neavând alt nivel superior, se consideră că oferă servicii direct utilizatorilor. Ca toate celelalte nivele, nivelul aplicaţie se bazează pe funcţiile nivelelor inferioare pentru a asigura procesul de comunicaţie în reţea. În cadrul nivelului aplicaţie protocoalele specifică mesajele ce sunt schimbate între sursă şi destinaţie, sintaxa comenzilor de control, tipul şi formatul datelor ce sunt transmise şi metodele specifice pentru notificarea şi refacerea erorilor.

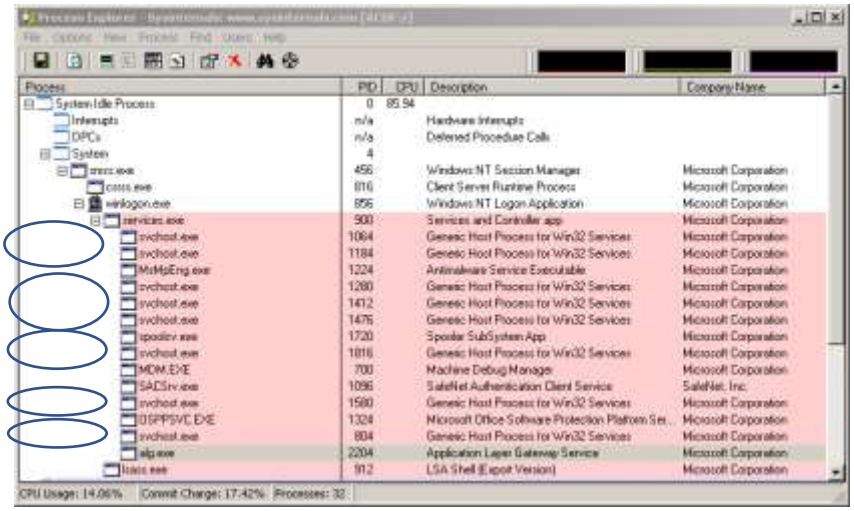


Figura 7.4. Programul svchost.exe apare încărcat în memorie de mai multe ori

**Funcțiile protocoalelor de nivel aplicație**

Protocoalele nivelului aplicaţie sunt folosite atât de către sursă cât şi de către destinaţie în timpul unei sesiuni de comunicaţie. Pentru ca schimbul de mesaje să aibă loc cu succes trebuie ca protocoalele nivelului aplicaţie folosite atât la sursă cât şi la destinaţie să coincidă. Protocoalele stabilesc reguli consistente pentru schimbul datelor între aplicaţiile şi serviciile ce se execută pe calculatoarele ce participă la procesul de comunicaţie. Protocoalele specifică modalitatea în care datele din interiorul mesajelor sunt structurate şi tipul mesajelor ce sunt schimbate între sursă şi destinaţie. Aceste mesaje pot fi cereri pentru servicii, confirmări de primire a datelor, mesaje de date sau mesaje de eroare. De asemenea, protocoalele definesc dialogurile între mesaje, asigurând faptul că un mesaj transmis este recepţionat în mod corect iar serviciile corespunzătoare sunt apelate. Există numeroase aplicaţii ce pot comunica într-o reţea; din această cauză nivelul aplicaţie trebuie să asigure implementarea unor protocoale ce pot asigura comunicaţia în reţea.

Fiecare protocol are un anumit scop şi anumite funcţii ce asigură atingerea acelui scop. Detaliile fiecărui protocol trebuie respectate pentru ca funcţiile ce asigură interfaţa la un anumit nivel din modelul OSI să corespundă cu serviciile unui nivelul inferior. Aplicaţiile şi serviciile pot utiliza mai multe protocoale de comunicaţie în cadrul unei singure conversaţii. Un protocol poate descrie cum se realizează conexiunea iar alt protocol poate descrie procesul de transfer al datelor atunci când mesajul trece la un nivel inferior.

.

## 7.3.2 Exemple de protocoale și servicii la nivelul aplicație

**Protocolul DNS**

Protocolul DNS defineşte un serviciu automat ce realizează corespondenţa numelor cu adresele numerice IP. Acesta include formatul cererilor, răspunsurilor şi al datelor transmise în reţea. Comunicaţiile protocolului DNS utilizează un format unic denumit mesaj. Acest format de mesaj este folosit pentru toate tipurile de cereri venite din partea clienţilor, pentru răspunsurile din partea serverului, pentru mesajele de eroare şi pentru transferul informaţiilor între servere.

Un server DNS realizează corespondenţa de nume – adresa IP folosind BIND

(Berkeley Internet Name Domain) – serviciul de tip daemon denumit named. Dezvoltat iniţial în anii 1980 la Universitatea Berkeley din California, formatul de mesaj DNS folosit de BIND reprezintă cel mai utilizat format DNS din Internet. Serverul DNS memorează diferite tipuri de înregistrări folosite pentru a rezolva nume de adrese. Câteva tipuri de înregistrări sunt:

- A – adresă de dispozitiv terminal
- NS – server de nume
- CNAME – nume canonic pentru un alias; folosit atunci când mai multe servicii au aceeaşi adresă de reţea, dar fiecare serviciu are propria intrare în DNS
- MX – înregistrare de schimb de mail; face corespondenţa unui nume de domeniu cu o listă de servere de mail pentru acel domeniu.

Atunci când un client realizează o cerere, procesul BIND al serverului caută în propriile înregistrări pentru a face corespondenţa numelui. Dacă nu este posibil acest lucru, încearcă să contacteze alte servere pentru a rezolva numele. Cererea poate fi transmisă către mai multe servere iar acest lucru poate necesita mai mult timp şi lăţime de bandă. După ce s-a găsit o corespondenţă şi aceasta a fost returnată serverului original, serverul va stoca temporar adresa IP corespunzătoare numelui în memoria de tip cache. Dacă acelaşi nume este cerut din nou, serverul iniţial poate returna adresa IP folosind conţinutul memoriei cache. Operaţia de salvare în memoria cache a corespondenţelor „nume-adresa IP" poate reduce interogările DNS şi traficul din reţea. Pe calculatoarele Windows, serviciul client DNS optimizează performanţa rezoluţiei de nume DNS prin stocarea în memorie a corespondenţelor de nume aflate anterior. Comanda ipconfig/displaydns afişează toate intrările din DNS stocate în memorie pe un calculator Windows.

**Serviciul DNS**

Serviciul DNS este un serviciu ce acţionează pe baza principiului client-server, însă diferă de alte servicii client-server pe care le vom examina. Dacă alte servicii de acest tip utilizează o aplicaţie client (precum un browser web sau un client de e-mail), aplicaţia client DNS rulează ca un serviciu de sine stătător. Clientul DNS, denumit în engleză şi *DNS resolver* – îi putem spune *translator DNS*, oferă suport pentru translatarea numelor în adrese IP pentru toate celelalte aplicaţii sau servicii de reţea ce necesită acest serviciu.

Atunci când se configurează un echipament de reţea se introduce o adresă (sau mai multe) de server DNS pe care clientul DNS îl poate folosi pentru translatarea numelor în adrese IP. În mod uzual, furnizorul de Internet ne oferă adresele serverelor de DNS pe care trebuie să le utilizăm la configurare. În momentul în care o aplicaţie utilizator necesită conectarea la un echipament în reţea (pentru care se cunoaşte numele, nu şi adresa IP), clientul DNS apelat trimite o cerere către unul dintre aceste servere de nume pentru a afla adresa IP corespondentă.

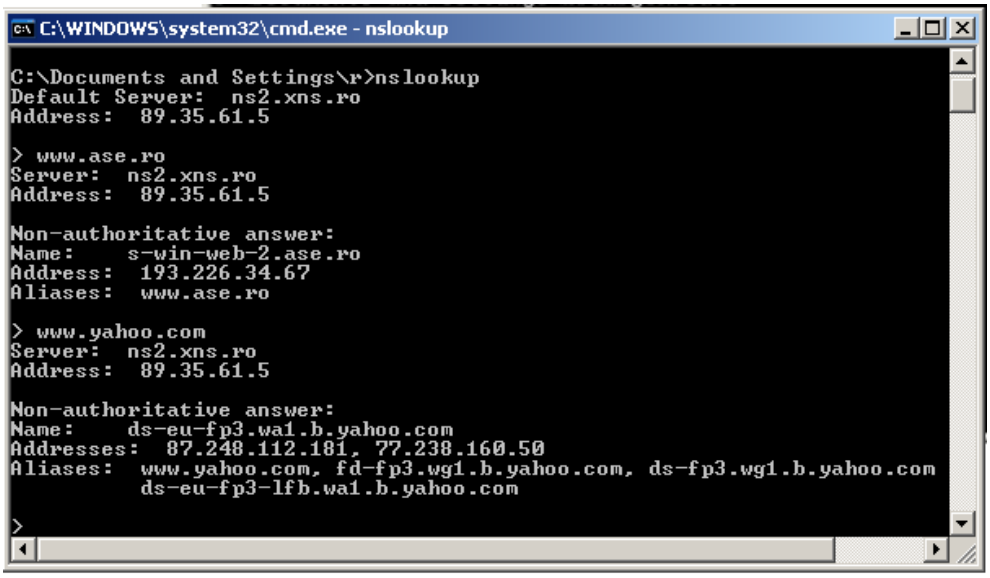



Figura 7.5. Exemplu de apel al comenzii *nslookup* în cazul sistemului de operare Windows

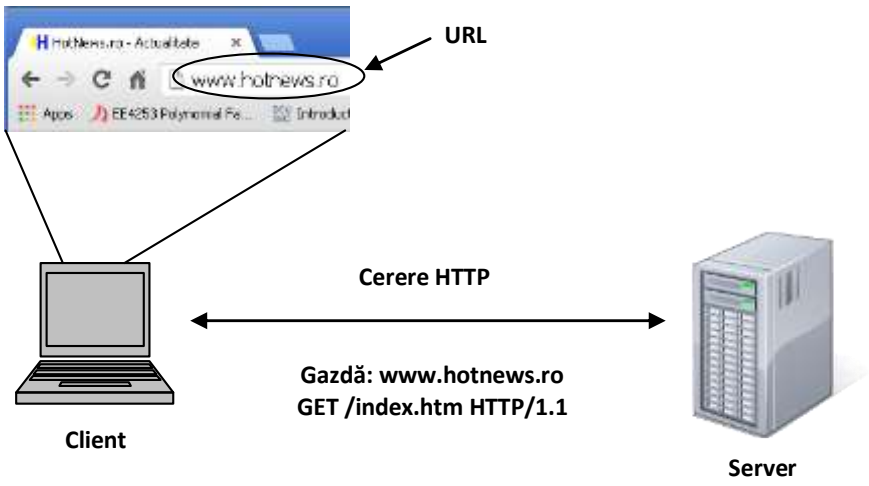
Sistemele de operare posedă, de asemenea, un utilitar denumit *nslookup* ce permite unui utilizator să interogheze manual serverele de nume pentru a afla adresa IP a unui calculator pentru care se cunoaşte adresa de nume. Acest utilitar poate fi de asemenea, folosit pentru a rezolva probleme de translatare a adreselor de nume şi pentru a verifica starea curentă a serverelor de DNS. În figura 7.5, apelul simplu al comenzii *nslookup* în Windows ne arată serverul implicit de DNS din configurare, care se numeşte în cazul nostru *ns2.xns.ro* şi care are adresa IP *89.35.61.5*. La prompterul ce apare după lansarea comenzii *nslookup* putem introduce adrese de nume pentru care dorim să aflăm adresa IP. Observăm astfel, tot în figura 3.8, că adresa de nume *www.ase.ro* are adresa IP *193.226.34.67* iar www.yahoo.com are adresele IP corespondente *87.248.112.181* şi *77.238.160.50*.

**Protocolul HTTP şi limbajul HTML**

Una dintre aplicaţiile cele mai utilizate în ultimele două decenii este aceea legată de navigarea pe web, adică *navigatorul* sau *browserul web*. Navigatoarele web sunt aplicaţii client instalate pe calculatoare pentru a asigura conectarea la World Wide Web şi pentru a putea accesa resurse stocate pe un aşa numit *server web*. Ca şi majoritatea proceselor server, serverul web rulează ca un serviciu în fundal asigurând accesul la diverse tipuri de fişiere. Navigatoarele pot interpreta şi prezenta diferite tipuri de date, începând cu simplu text sau fişiere sursă ce folosesc *HTML (Hypertext Markup Language).* Alte tipuri de date pot avea nevoie de alte programe sau servicii pentru a putea fi vizualizate. Acestea sunt elemente suplimentare de tipul *„plug-in”* sau *„add-on”* ce pot extinde funcţionalitatea standard a unui browser. Printre cele mai populare elemente suplimentare de acest tip sunt *Shockwave Flash Object* (care ajută la vizualizarea obiectelor generate cu ajutorul Macromedia Flash) şi *Adobe PDF Reader Link Helper* (care ajută la încărcarea unui document *.pdf* direct în fereastra navigatorului). Cele mai populare programe de tip browser de astăzi sunt: Google Chrome, Internet Explorer, Mozilla Firefox, Safari şi Opera.

În momentul în care scriem o adresă web (care se numeşte şi *URL – Uniform Resource Locator*, despre care putem afla mai multe detalii consultând RFC-ul 1738 la adresa http://www.ietf.org/rfc/rfc1738.txt) în bara de navigare a unui browser web, acesta stabileşte o conexiune cu serviciul web ce rulează pe un server ce foloseşte protocolul HTTP. Resursele de tip *URL* şi *URI (Uniform Resource Identifier)* reprezintă nume ale unor resurse disponibile în reţeaua Internet, prin intermediul serviciului *WWW (World Wide Web).*

Adresa URL *http://www.ase.ro/index.html* este un exemplu de URL ce face referire la pagina web denumită index.html pe serverul denumit *ase.ro*. Pentru a accesa conţinutul paginilor web, clienţii web realizează conexiuni cu serverul şi trimiţând o cerere de acces la acea resursă. Serverul oferă un răspuns ce conţine acea resursă şi pe baza informaţiilor obţinute, browserul interpretează datele şi le prezintă utilizatorului. În figura 7.6 este prezentată pe scurt funcţionarea protocolului HTTP ce stă la baza funcţionării serviciului *WWW* (pentru care adesea se foloseşte prescurtarea *web*). Pentru a înţelege mai bine cum funcţionează interacţiunea între browserul web şi serverul de web să luăm exemplul în care scriem în browser adresa URL următoare: *http://www.hotnews.ro/index.htm.* În primul rând, browserul analizează cele trei părţi ale acestui URL: (1) *http (protocolul utilizat),* (2) *www.hotnews.ro (numele serverului)* şi (3) *index.htm (numele fişierului cerut).* În continuare, browserul foloseşte un server DNS pentru a afla adresa IP asociată numelui *www.hotnews.ro*, pe care o va folosi pentru conectarea la serverul de web. Folosind cerinţele protocolului HTTP, browserul trimite o cerere de tip GET către server pentru fişierul index.htm iar serverul răspunde navigatorului cu codul HTML al acestei pagini. În cele din urmă, browserul decodifică codul HTML şi celelalte formate utilizate în pagina web afişând conţinutul paginii în fereastra sa.




**Introducerea adresei *http://www.hotnews.ro* în bara de adrese a browserului generează mesajul HTTP de tip GET.**

Figura 7.6 Funcţionarea protocolului HTTP

## 7.4 Îndrumar pentru autoverificare

### 7.4.1 Sinteza unității de studiu 7

Nivelul aplicație din modelul ISO-OSI (sau în modelul TCP/IP) este nivelul cel mai apropiat de către utilizator. Protocoalele nivelului aplicație sunt folosite pentru a schimba date între programele ce rulează pe echipamentele sursă și destinație. Nivelul prezentare are trei funcții de bază: formatarea, sau prezentarea datelor de la sursă într-un format compatibil cu recepția la destinație, comprimarea datelor la sursă (într-un așa mod astfel încât să poată fi

decomprimate la destinație) și criptarea datelor pentru transmisie și decriptarea datelor la destinație.

Nivelul sesiune creează și menține dialoguri între aplicațiile sursă și destinație. Acest nivel administrează schimbul de informație pentru a iniția dialoguri, menținerea activității acestora și repornirea sesiunilor întrerupte sau blocate pentru o perioadă mai lungă de timp.

### 7.4.2 Concepte și termeni de reținut

| | |
|---|---|
| *Nivelul aplicație* | *HTTP* |
| *Date* | *Procese* |
| *WWW* | *FTP* |
| *E-mail* | *Telnet* |
| *SSH* | *DNS* |

### 7.4.3 Întrebări pentru autoverificare

***Întrebarea 1.*** Într-o rețea de acasă, ce echipament va oferi adresare dinamică a adreselor IPv4 clienților rețelei?

a) Un ruter wireless
b) Un server de fișiere dedicat
c) Un server DHCP dedicat al ISP-ului
d) Un server DNS

*Răspuns: a*

***Întrebarea 2.*** Care dintre următoarele protocoale folosește criptarea datelor?

a) DNS
b) DHCP
c) HTTPS
d) FTP
e) UDP

*Răspuns: c*

***Întrebarea 3.*** Ce tip de mesaj este folosit de către un client HTTP pentru a încărca date de pe un server web?

a) GET
b) POST
c) ACK
d) GETACK
e) PUT

*Răspuns:a*

### Întrebări de control și teme de dezbatere

1. Care sunt funcțiile de bază ale nivelului aplicație din modelul ISO-OSI?
2. Dați 5 exemple de protocoale ce acționează la nivelul aplicație din modelul TCP/IP.
3. Enumerați două funcții ale nivelului aplicație din modelul TCP/IP?

### 7.4.4 Bibliografie obligatorie

1. Răzvan Daniel Zota, Rețele de calculatoare, capitolul 3, Ed. ASE, 2013.