

2. Introducere în programarea rețelelor de calculatoare

O rețea de calculatoare reprezintă „o colecție de calculatoare autonome interconectate folosind o singură tehnologie” (A. Tanenbaum, 2003). Pornind de la această definiție, termenul *Internet* utilizat pe parcursul acestei lucrări face referire la o rețea ce interconectează două sau mai multe rețele de calculatoare autonome aflate la distanțe mari.

Întrucât schimbul de informații între calculatoare, denumite pe parcursul acestei lucrări *stații*, este posibil datorită interconectării acestora pe baza unor protocoale bine stabilite, pe parcursul acestui capitol vom prezenta principalele aspecte din domeniul programării rețelelor de calculatoare.

2.1 Modelul Client-Server

Modelul Client-Server, ilustrat în figura 2.1, este unul dintre cele mai răspândite modele de comunicare, fiind modelul pe care sunt construite o serie de rețele. În cadrul acestui model atât clientul cât și serverul trebuie să fie active, se pretează atât în cazurile în care clientul și serverul sunt situate în aceeași rețea, dar și în cazurile în care acestea sunt despărțite de mai multe rețele interconectate.

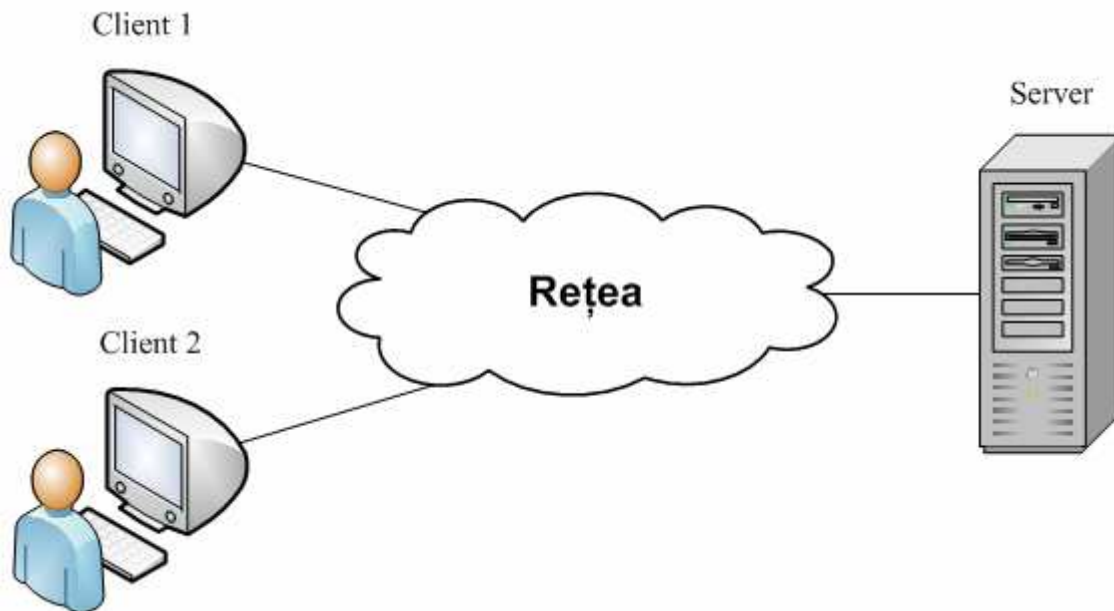


Figura 2.1 Modelul Client-Server

În cadrul acestui model, sunt implicate două procese: unul pe stația clientului și unul pe stația serverului. Comunicația presupune transmiterea unui mesaj din partea clientului și blocarea acestuia până la primirea răspunsului din partea serverului. Caracteristica principală a acestui model reprezintă faptul că serverul nu emite cereri, ci

le primește, răspunzând întotdeauna cu un mesaj reprezentând rezultatul procesării. De regulă, serverele lucrează cu un număr mare de clienți.

2.2 Modelele OSI și TCP/IP

Programarea aplicațiilor destinate rețelelor de calculatoare utilizează paradigma straturilor sau nivelelor, fiecare nivel fiind construit peste cel de dedesubt, cu proprietatea că un nivel de pe o mașină conversează cu același nivel de pe o altă mașină.

Modelul OSI (en. „Open Systems Interconnection”) reprezintă o descriere abstractă a structurării rețelelor, având șapte nivele asociate:

- *Nivelul fizic:* are rolul de transmitere a biților printr-un canal de comunicație;
- *Nivelul legătură de date:* are rolul de a transforma un mijloc de transmisie fizic oarecare într-o linie care să fie disponibilă nivelului rețea fără erori de transmisie nedetectate;
- *Nivelul rețea:* are rolul de dirijare a pachetelor de la sursă la destinație;
- *Nivelul transport:* are rolul de a accepta date de la nivelul sesiune, de a le descompune în unități mai mici, de a transfera aceste unități nivelului rețea și de a se asigura că fragmentele sunt recepționate de nivelul superior în ordinea transmisiei acestora;
- *Nivelul sesiune:* are rolul de stabilire a sesiunilor pentru utilizatorii de e stații diferite. Sesiunile asigură o serie de servicii printre care se numără controlul dialogului, gestionarea jetonului și sincronizarea;
- *Nivelul prezentare:* are rolul de a asigura transferul unitar a datelor cu reprezentări diferite;
- *Nivelul aplicație:* are rolul de a asigura implementarea protocoalele ce utilizează celelalte nivele. Un exemplu de asemenea protocol este HTTP (en. „HyperText Transfer Protocol”).

Modelul TCP/IP poartă denumirea celor două protocoale fundamentale utilizate. Elaborarea acestui model s-a realizat ca o soluție pentru o comunicare fără întreruperi între două stații unde punctele intermediare pot eșua, își pot schimba arhitectura. Totodată, acest model este unul mult mai flexibil, corespunzător unei game largi de aplicații, începând de la transfer de fișiere și până la fluxuri de date multimedia în timp real.

Modelul TCP/IP definește următoarele 4 nivele:

- *Nivelul gazdă-rețea:* unifică nivelul fizic cu nivelul legătură de date din cadrul modelului OSI. De fapt, în cadrul modelului TCP/IP nu se spun foarte multe lucruri despre acest nivel, singura mențiune fiind că stația trebuie să se lege de rețea pentru a putea trimite pachete;
- *Nivelul internet:* reprezintă nivelul pe care se centrează întreaga arhitectură. Rolul său este clar definit, fiind acela de a permite stațiilor să emită pachete în orice rețea și de a face ca pachetele să circule independent până la

destinație. Acest nivel definește oficial protocolul utilizat: IP (en. „Internet Protocol”);

- *Nivelul transport*: a fost proiectat astfel încât să permită conversații între entitățile pereche de pe stațiile sursă și destinație, fiind similar cu nivelul transport din cadrul modelului OSI. Față de modelul OSI, unde nu sunt definite protocoalele acestui nivel, modelul TCP/IP definește două protocoale: TCP (en. „Transmission Control Protocol”) și UDP (en. „User Datagram Protocol”);
- *Nivelul aplicație*: este situat deasupra nivelului transport, conținând toate protocoalele de deasupra nivelului transport.

Cele două modele prezentate, precum și corespondențele dintre ele sunt ilustrate în figura 2.2. În continuarea acestui capitol vom prezenta sumar nivelele modelului TCP/IP, acesta fiind modelul utilizat în Internet.

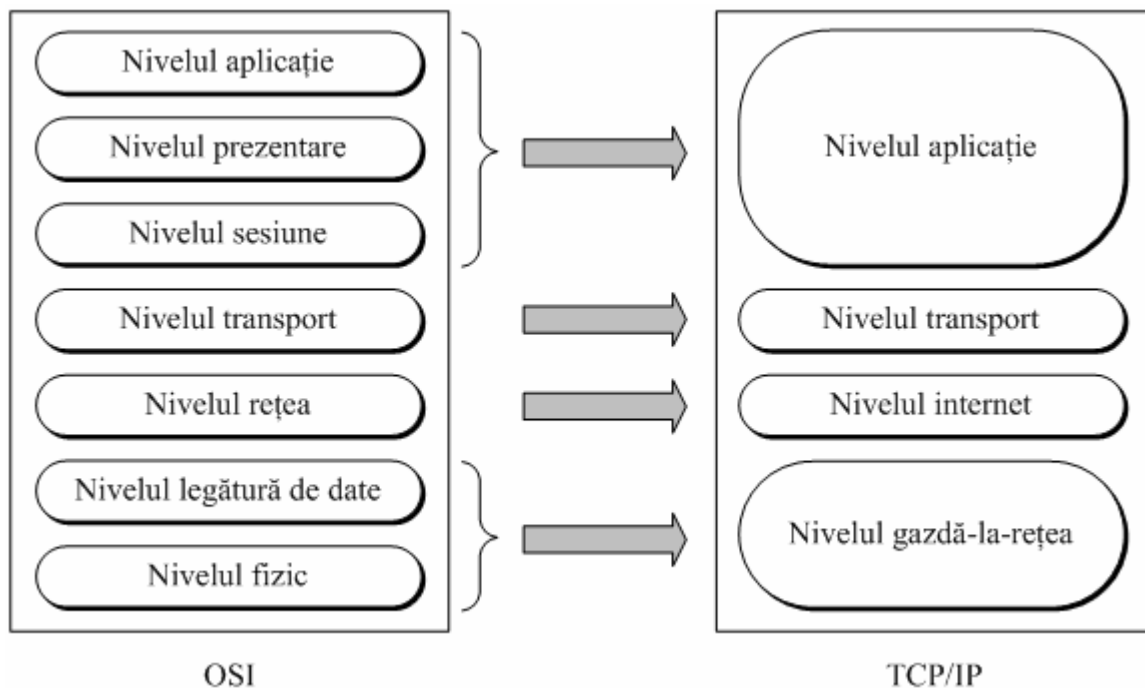


Figura 2.2 Modelele OSI și TCP/IP și corespondențele dintre cele două

2.2.1 Nivelul gazdă-la-rețea

Nivelul gazdă-la-rețea include atât nivelul fizic cât și nivelul legătură de date din cadrul modelului OSI.

Scopul *nivelului fizic* este acela de a transporta o secvență de biți de la o stație la alta. Pentru realizarea acestui transport pot fi utilizate o serie de medii fizice. Fiecare dintre acestea este caracterizată de o lărgime de bandă, întârziere, cost, precum și ușurința de instalare și întreținere. Există două clase de medii fizice: medii ghidate și medii neghidate. Exemple de medii ghidate includ cablul de cupru și fibrele optice, iar exemple de medii neghidate includ undele radio și laserul.

Cel mai des utilizat mediu fizic este *cablul torsadat*, o pereche torsadată fiind formată din două fire de cupru izolate, împletite sub formă elicoidală. Lărgimea de bandă depinde de grosimea firului dar și de lungimea acestuia, în cele mai multe cazuri aceasta ajungând la câțiva megabiți pe secundă pe distanțe de ordinul kilometrilor. Cablurile torsadate uzuale sunt cunoscute și sub denumirea de cabluri UTP (en. „Unshielded Twisted Pair”). Categoriile mai noi de asemenea cabluri sunt capabile să transfere semnale de frecvență de 600MHz. Un asemenea exemplu de cablu este ilustrat în figura 2.3.

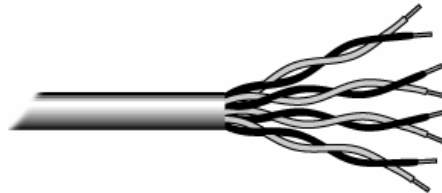


Figura 2.3 Exemplu de secțiune dintr-un cablu UTP

Un alt mediu uzual este *cablul coaxial*, ilustrat în figura 2.4. Acesta conține un singur conductor de cupru rigid în centru, învelit într-un material izolator. Acest material este la rândul său încapsulat într-un conductor circular acoperit cu un înveliș de plastic. Acest mediu oferă o ecranare mult mai bună decât cablul torsadat și poate fi folosit pe distanțe mult mai mari la rate de transfer mai mari, frecvențele de transfer ajungând și la 1GHz. Cu toate acestea, dificultatea instalării cablului datorită rigidității acestuia precum și costul ridicat au făcut ca acesta să fie înlocuit de cablurile UTP.

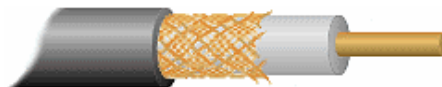


Figura 2.4 Exemplu de secțiune dintr-un cablu coaxial

Fibrele optice reprezintă o soluție modernă pentru problema transferului cât mai rapid a biților. De fapt, această tehnologie a fost cea care a învins în războiul dintre dezvoltarea comunicațiilor și cea a calculatoarelor. Motivul este faptul că lărgimea de bandă a acestora este mai mare de 50000 Gbps, însă o asemenea rată de transfer nu poate fi atinsă datorită imposibilității transformării semnalelor electrice în semnale optice, fapt ce limitează utilizarea practică a acestor fibre la 10Gbps.

Cablurile din fibră optică sunt formate dintr-un miez de sticlă prin care se propagă unda luminoasă și din materiale protective ce înconjoară acest centru, o secțiune dintr-un asemenea cablu fiind ilustrat în figura 2.5. De regulă, diametrul miezului variază între 8 și 50 micrometri. De obicei, mai multe fibre sunt grupate împreună, sunt învelite într-o teacă protectoare, formând astfel un cablu optic.

Avantajele fibrei optice față de cablurile din cupru sunt multiple. În primul rând, lărgime de bandă net superioară o face un candidat ideal pentru rețele de mare viteză. În al doilea rând, datorită atenuării scăzute, repetoarele sunt necesare la fiecare 30km, față de 5km pentru firele de cupru. Fibra nu este afectată de șocurile electrice, este mult mai ușoară decât firele din cupru.

Cu toate acestea, fibra optica prezintă și câteva dezavantaje semnificative. În primul rând aceasta este o tehnologie mai puțin familiară, necesitând o pregătire semnificativă în domeniu. Fibrele pot fi întrerupte foarte ușor dacă sunt îndoite în mod repetat, iar comunicațiile bidirecționale necesită două fibre sau două benzi de frecvență diferită pe aceeași fibră.

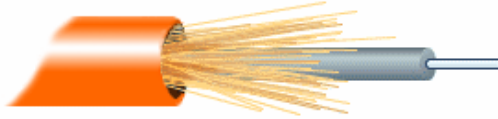


Figura 2.5 Exemplu de secțiune dintr-un cablu optic

O serie de rețele din zilele noastre utilizează comunicațiile fără fir, acestea fiind cunoscute și sub denumirea de *rețele fără fir*. Rețelele fără fir pot utiliza o gamă largă de modalități de comunicare printre care se numără undele radio și undele infraroșii.

Proprietățile *undelor radio* sunt date de frecvența de operare. Astfel, la frecvențe joase acestea se propagă bine prin obstacole dar puterea semnalului scade foarte mult o dată cu distanța parcursă. La frecvențe înalte undele radio tind să se propage în linie dreaptă și să ricoșeze din obstacole. Dezavantajul acestora este că toate undele radio sunt supuse interferențelor venite din partea echipamentelor electrice.

Undele infraroșii sunt folosite pentru comunicații pe distanțe mici. Exemple de echipamente ce folosesc asemenea unde sunt telecomenzile pentru televizoare, combine muzicale sau aparate video. Acestea sunt relativ ieftine, ușor de construit, însă prezintă dezavantajul major că nu penetrează obiectele solide.

Nivelul *legătură de date* are rolul de a furniza o interfață bine definită nivelului rețea, de a trata erorile de transmisie și de a regla fluxul cadrelor astfel încât receptorii mai lenți să nu fie inundați de emițători rapizi. Pentru îndeplinirea acestor funcționalități, nivelul legătură de date primește pachete de la nivelul rețea, le încapsulează în cadre și le transferă nivelului fizic.

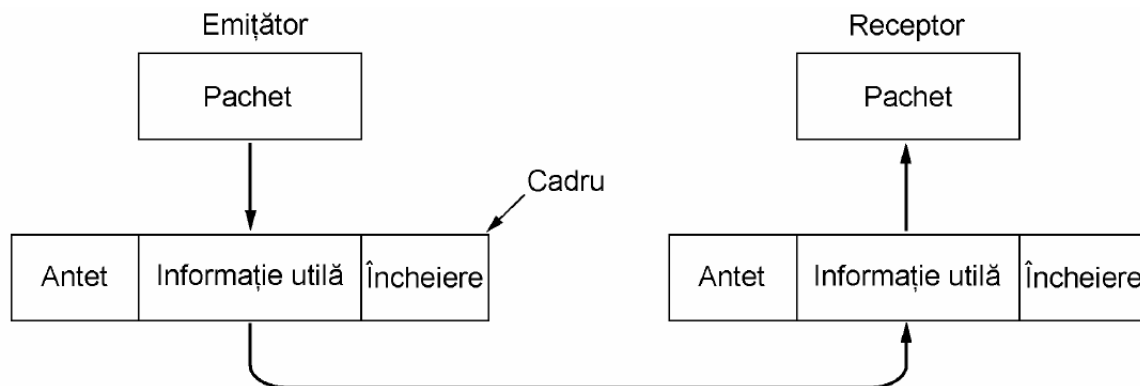


Figura 2.6 Relația dintre pachete și cadre

După cum se poate observa din figura 2.6 (A. Tanenbaum, 2003), pachetul primit de la nivelul superior de către emițător este încapsulat într-un cadru ce conține un antet și o secțiune de încheiere. În cazul în care pachetul este mult prea lung, acesta este împărțit în fragmente transmise separat. De partea cealaltă, receptorul extrage pachetul din una

sau mai multe cadre recepționate, pe care îl transmite nivelului superior. Cadrele sunt numerotate secvențial, iar în cazul erorilor, cadrele pierdute sunt retransmise.

Nivelul gazdă-la-rețea include pe lângă cele două nivele amintite anterior un *subnivel de acces la mediu* care controlează cine urmează a transmite într-un mediu multi-acces la un singur canal. Acest subnivel poartă denumirea de MAC (en. „Medium Access Control”), fiind important mai ales pentru rețelele de tip LAN (en. „local Area Network”). Rețelele de tip WAN (en. „Wide Area Network”) utilizează de regulă legături punct-la-punct, cu excepția rețelelor prin satelit.

Una dintre cele mai răspândite rețele este Ethernet-ul, cunoscută și sub denumirea de IEEE802.3. Aceasta este o rețea cu difuzare bazată pe magistrală cu control descentralizat lucrând la viteze între 10Mbps și 10Gbps. Principiul care stă la baza acestei rețele este că stațiile pot să transmită când doresc, iar dacă două sau mai multe pachete se ciocnesc, fiecare stație așteaptă o perioadă de timp aleatorie, după care încearcă din nou.

Formatul cadrelor Ethernet este ilustrat în figura 2.7 (A. Tanenbaum, 2003) și include următoarele câmpuri:

- *Preambul*: 7 octeți de forma 10101010, utilizați de receptor pentru sincronizarea ceasurilor;
- *Delimitator de cadru de început*: un singur octet cu valoarea 10101011, ce indică începutul unui cadru;
- *Adresa destinației și adresa sursei*: adresa MAC a destinației și a sursei, cu bitul cel mai semnificativ 0 pentru adrese locale și 1 pentru difuzare. Fiecare interfață Ethernet are asignată o adresă Ethernet unică globală;
- *Lungime*: indică numărul de octeți de date;
- *Date*: reprezintă octeții datelor incluși într-un cadru;
- *Sumă de control*: reprezintă o sumă de control pentru identificarea erorilor de transmisie.

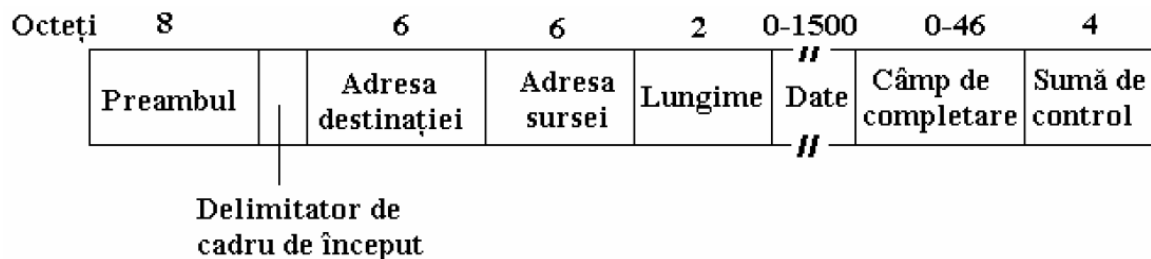


Figura 2.7 Formatul cadrelor Ethernet

2.2.2 Nivelul Internet

Rolul nivelului Internet este acela de a asigura transferul pachetelor de la sursă la destinație, chiar dacă stațiile sunt despărțite de mai multe rețele. Acest nivel nu asigură o fragmentare a pachetelor, nici o detectare a pierderilor de pachete, aceasta fiind sarcina nivelelor superioare.

Protocolul pe care s-a construit Internetul este implementat la acest nivel și poartă denumirea de IP (en. „Internet Protocol”). În cadrul protocolului IP se definește datagrama IP, reprezentând pachetul construit la acest nivel. Structura datagramii IPv4 este ilustrată în figura 2.8.

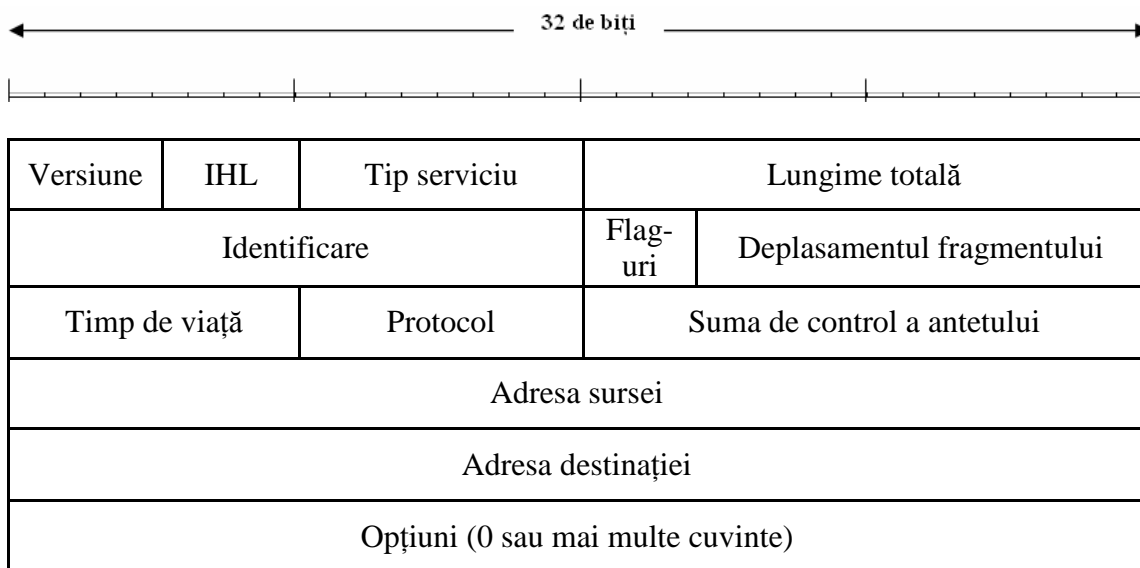


Figura 2.8 Formatul antetului IPv4

Câmpul *Versiune* este pe 4 biți și reprezintă versiunea antetului. Pentru IPv4, acest câmp are valoarea 4. *IHL* (en. „Internet Header Length”) este pe 4 biți și reprezintă lungimea antetului IP, exprimat în multiplii de 32 de cuvinte (un cuvânt = 32 de biți), valoarea minimă pentru acest câmp fiind 5, când nu apar opțiuni.

Câmpul *Tip serviciu* este pe 8 biți, dintre care ultimii doi biți sunt rezervați. Acest câmp se referă la parametrii tipului de serviciu specificat. Acești parametri pot fi utilizați pentru determinarea modului de tratare a pachetului primit.

Câmpul *Lungime totală* este pe 16 biți și conține valoarea dimensiunii totale a pachetului (datagramei) cu tot cu dimensiunea antetului IP. Câmpul acesta fiind pe 16 biți, dimensiunea maximă permisă pentru un pachet este de 65535 octeți. Câmpul *Identificare*, pe 16 biți, este folosit pentru a deosebi fragmentele unui pachet de fragmentele unui alt pachet.

Flag-urile, pe 3 biți conține un bit neutilizat și doi biți utilizați: DF și MF. DF (en. „Don’t Fragment”) controlează fragmentarea pachetului, dacă valoarea bitului este 0 pachetul poate fi fragmentat dacă este necesar, iar dacă valoarea bitului este 1, atunci acesta nu trebuie fragmentat. MF (en. „More Fragments”) este utilizat pentru a indica dacă au ajuns toate fragmentele. MF are valoarea 0 dacă este ultimul fragment și valoarea 1 dacă urmează alte fragmente.

Deplasamentul fragmentului este un câmp pe 13 biți ce indică numărul de secvență al fragmentului. Acest număr este utilizat la reasamblarea pachetelor pe stația destinație.

Câmpul *Timp de viață* pe 8 biți este utilizat pentru a limita timpul de viață a datagramei. Valoarea acestui câmp este decrementată la fiecare salt (i.e. trecere dintr-o rețea în alta). Când valoarea ei ajunge la 0 pachetul este distrus și se trimite înapoi la sursă un pachet avertisment.

Câmpul *Protocol*, pe 8 biți denotă protocolul transport ce trebuie să preia pachetul. Printre protocoale posibile se numără TCP și UDP.

Suma de control este pe 16 biți și se aplică doar asupra antetului. Aceasta se calculează adunând jumătățile de cuvinte ale antetului folosind complement față de unu. Această sumă este recalculată de destinatar considerând valoarea 0 pentru câmpul sumă de control, fiind folosită pentru detectarea erorilor de transmisie. Ca mențiune, întrucât

câmpul *Timp de viață* se modifică la fiecare salt, valoarea sumei de control trebuie recalculată de fiecare dată.

Următoarele două câmpuri, *Adresa sursei* și *Adresa destinației*, reprezintă adresele IP ale, respectiv, sursei și destinației. Acestea sunt fiecare pe 32 de biți, formatul și clasele de adrese IPv4 disponibile fiind ilustrate în figura 2.9.

Câmpul *Opțiuni*, de lungime variabilă, permite versiunilor ulterioare să includă o serie de informații care au fost omise din proiectarea inițială.

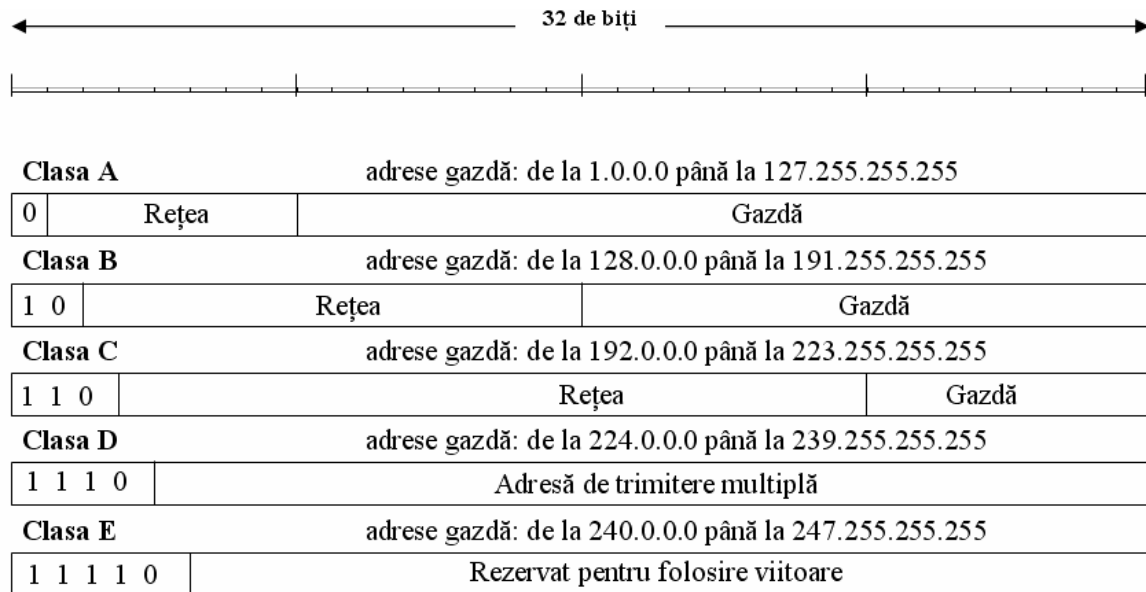


Figura 2.9 Formatul și clasele de adrese IPv4 disponibile

2.2.3 Nivelul transport

Rolul nivelului transport este acela de a oferi servicii sigure și eficiente nivelului aplicație. Pentru a atinge acest scop, acest nivel definește două tipuri de servicii: datagramă și orientate conexiune.

În cadrul modelului TCP/IP, serviciul datagramă este implementat prin intermediul protocolului UDP (en. „User Datagram Protocol”). UDP este un protocol transport fără conexiune ce oferă posibilitatea aplicațiilor să trimită pachete fără să fie nevoie de stabilirea unei conexiuni în prealabil. Cu toate acestea, UDP este un protocol fără confirmare, ceea ce înseamnă că pachetele se pot pierde fără ca sursa sau destinația să fie conștiente de aceste pierderi.



Figura 2.10 Formatul antetului UDP

Forma antetului UDP este ilustrată în figura 2.10. Câmpurile *Port sursă* și *port destinație* sunt pe câte 16 biți și identifică punctele terminale de pe mașinile sursă, respectiv destinație. Câmpul *Lungime UDP*, pe 16 biți, include dimensiunea antetului și cea a datelor. Câmpul *Sumă de control UDP* pe 16 biți este utilizat pentru identificarea erorilor de transmisie. Acest ultim câmp este opțional în sensul că calcularea acestuia este una opțională, în cazul dezactivării acesteia se utilizează valoarea 0.

TCP (en. „Transmission Control Protocol”) reprezintă implementarea serviciului orientat conexiune. Acesta asigură „un flux sigur de octeți de la un capăt la celălalt al conexiunii într-o inter-rețea nesigură” (A. Tanenbaum, 2003). Formatul antetului TCP este ilustrat în figura 2.11.

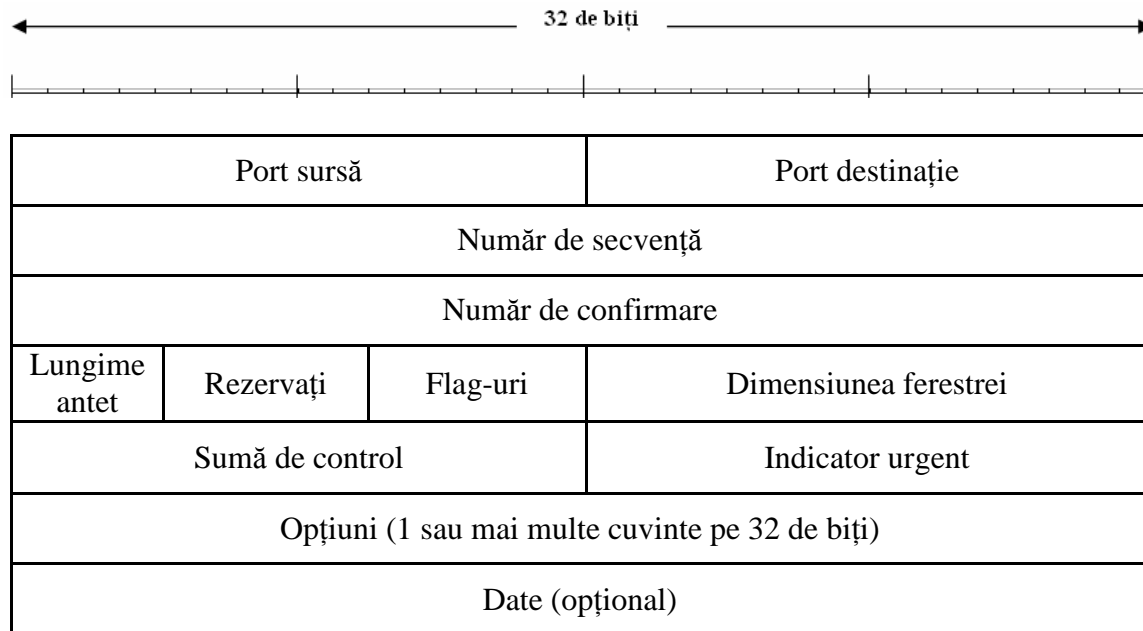


Figura 2.11 Formatul antetului TCP

Similar UDP, câmpurile *Port sursă* și *Port destinație* identifică punctele finale ale conexiunii. Câmpul *Număr de secvență* este pe 32 de biți și reprezintă numărul de secvență al pachetului curent. Câmpul *Număr de confirmare* este pe 32 de biți și reprezintă numărul confirmare pentru pachetul recepționat.

Câmpul *Lungime antet* este pe 4 biți și indică numărul de cuvinte pe 32 de biți conținute în antetul TCP. Acest câmp este necesar datorită opțiunilor care pot urma antetul.

Flag-uri este un câmp pe 6 biți, fiecare bit fiind un indicator: URG, ACK, PSH, RST, SYN, FIN. Bitul URG este indicatorul Urgent, dacă este activ, atunci câmpul *Indicator urgent* este și el activ, iar acesta din urmă va conține deplasamentul datelor urgente. Dacă bitul ACK este activ, atunci *Numărul de confirmare* este valid. Dacă bitul PSH este activ, atunci aceasta indică faptul că informația cerută trebuie livrată cât mai repede, iar aplicația este rugată să nu aștepte umplerea memoriei tampon de comunicație. Bitul RST este utilizat pentru a desființa o conexiune devenită inutilizabilă, bitul SYN este utilizat la stabilirea unei conexiuni noi, iar bitul FIN este folosit pentru a încheia o conexiune.

TCP este un protocol bazat pe ferestre glisante. Astfel, câmpul *Dimensiunea ferestrei*, pe 16 biți, indică numărul de octeți ce pot fi transmiși. Câmpul *Sumă de control* este calculat asupra antetului și datelor opționale care urmează. Câmpul *Opțiuni* a fost proiectat pentru a permite adăugarea ulterioară de facilități suplimentare.

2.2.4 Nivelul aplicație

Ultimul nivel din cadrul modelului TCP/IP include toate protocoalele implementate deasupra nivelului transport. Printre aceste protocoale se numără FTP, SFTP, SMTP, HTTP, SSH, SSL, TLS sau HTTPS.

2.3 Programarea rețelelor prin socluri (en. „SOCKET”)

Socurile reprezintă punctele finale ale unei comunicații uni sau bidirecționale. Din punctul de vedere al programatorului, soclurile reprezintă interfața prin care se programează aplicațiile de rețea.

Fiecare soclu are atașat un număr de soclu format din adresa IP a stației și un număr pe 16 biți reprezentând portul alocat. Soclurile sunt create de către sistemul de operare prin intermediul unui API dedicat programării cu socluri. În momentul creării soclurilor se specifică protocolul precum și o serie de alți parametri. În caz de succes, sistemul de operare returnează identificatorul unui soclu creat, identificator ce este de regulă un număr pe 32 sau 64 de biți.

Există nenumărate API-uri pentru programarea soclurilor. În cadrul acestei lucrări se vor prezenta 3 dintre cele mai des întâlnite și utilizate abordări:

- API-ul MFC prin intermediul CAsyncSocket;
- API-ul Win32;
- API-ul Posix pentru sisteme Unix.

Fiecare din cele 3 abordări amintite prezintă avantaje și dezavantaje, care vor fi evidențiate în capitolele următoare. Exemplificarea utilizării acestora se va realiza prin construirea unor aplicații Client-Server pentru fiecare din cele două protocoale transport amintite în cadrul acestui capitol: TCP și UDP.