UNIVERSITATEA „POLITEHNICA” DIN BUCUREȘTI
FACULTATEA TRANSPORTURI

Departamentul Telecomenzi și Electronică în Transporturi

Rețele de calculatoare
Curs

București
2017

Cuprins

Capitolul 1. Introducere

1.1 Rețele de calculatoare. Internet

O rețea de calculatoare constă dintr-un număr de calculatoare autonome, interconectate astfel încât să fie capabile să schimbe informație între ele, ce pot funcționa independent pe baza unui sistem de operare/software propriu [1]. Termenul *calculator* ce va fi utilizat în acest curs va include de fapt, și va fi substituit uneori de orice tip de mașină de calcul, computer, dispozitiv sau echipament ce se poate conecta la o rețea de comunicație.

Internetul este o rețea de rețele care interconectează sute de milioane de calculatoare din întreaga lume. Nu cu mult timp în urmă, acestea erau, în primul rând, computere tradiționale, stații de lucru Linux și așa-numitele servere care stochează și transmit informații cum ar fi paginile web sau mesajele de poștă electronică. Cu toate acestea, tot mai multe sisteme *netradiționale* cum ar fi laptopuri, telefoane inteligente (*smartphone*), tablete, televizoare, console de jocuri, camere web, automobile, senzori sau elemente de acționare sau control au început să fie conectate la Internet [2]. În jargonul de Internet, toate acestea sunt numite gazde (*host*) sau sisteme finale (*end system*).

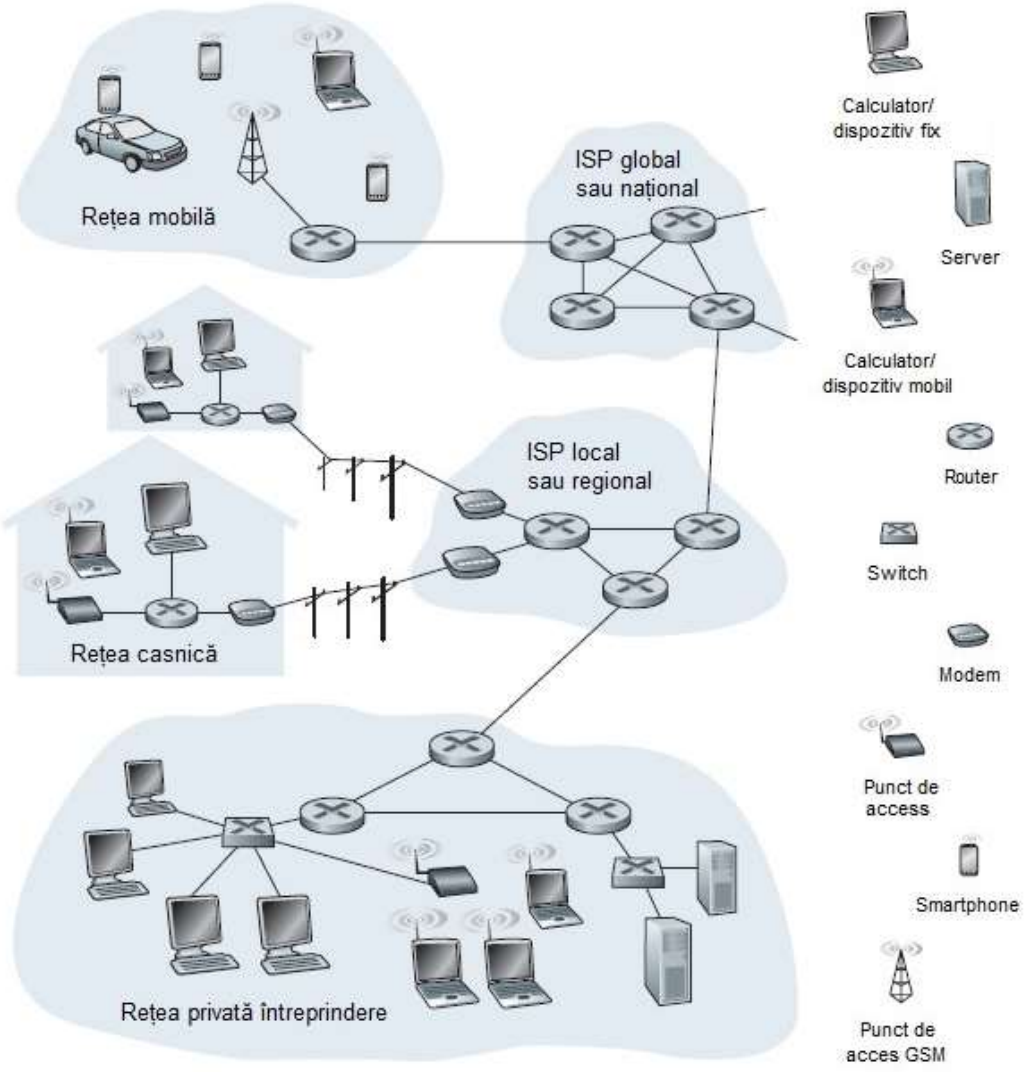

Figura 1. Structura tipică a Internetului [2]

Echipamentele de rețea menționate în Figura 1 sunt unele dintre cele mai des întâlnite în cadrul rețelei Internet, rolul acestora putând fi descris pe scurt astfel:

- Comutatorul (*Switch*) conectează în aceeași rețea mai multe calculatoare ce utilizează comunicații prin fir.
- Punctele de acces permit conectarea la o rețea a calculatoarelor ce utilizează comunicații fără fir.
- Modem-ul conectează între ele două rețele ce folosesc tehnologii diferite.
- Serverul gestionează rețelele și furnizează serviciile de rețea.
- *Router*-ul interconectează rețelele între ele.

Scopul principal al existenței Internetului este furnizarea de servicii pentru diverse aplicații: navigare pe Web, poștă electronică, control de la distanță, acces la rețele sociale, mesagerie instant, transfer de fișiere, Voice-over-IP, transmisii video și TV, jocuri, IoT, IoV, etc [2].

Conform datelor actuale sau statistice, numărul dispozitivelor conectate la Internet și evoluția acestuia sunt impresionante:

- În iulie 2011 existau aproape 2 miliarde [2]. În 2017 vor fi mai mult de 20 de miliarde (Figura 1) [3].
- La sfârșitul anului 2016 se estima că 328 de milioane de noi dispozitive sunt conectate la Internet în fiecare lună [4].
- 75 de miliarde vor fi conectate până în anul 2020 [5].
- Până în anul 2022 în fiecare casă vor exista circa 500 de dispozitive conectate [4].

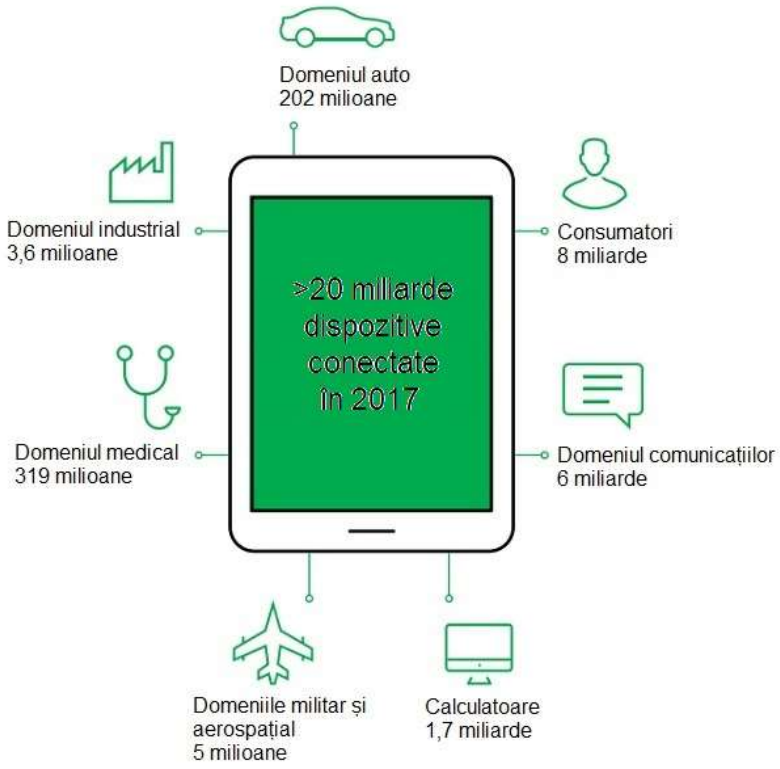

Figura 2. Evoluția numărului de dispozitive conectate la Internet pentru anul 2017 [3]

Așa cum se poate observa, sistemele finale sunt conectate împreună de o rețea de **legături de comunicație** și **comutatoare de pachete**. Atunci când un calculator are date de trimis către altul, expeditorul împarte datele în segmente mai mici și adaugă octeți de **antet** (*header*) fiecăruia. Segmentele de informații rezultate, cunoscute sub numele de **pachete** (*packet*), sunt apoi trimise prin rețea către calculatorul de destinație, unde sunt reasamblate în datele originale [2]. Un câmp de adresă din fiecare pachet specifică destinatarul.

Există două tipuri de legături de comunicație care sunt utilizate pe scară largă: punct-la-punct (*point-to-point*) și de difuzare (*broadcasting*) [6]:

- Legăturile punct-la-punct conectează perechi individuale de calculatoare. Pentru ca pachetele să ajungă de la sursă la destinație într-o rețea formată din legături punct-la-punct, de cele mai multe ori, acestea trec prin echipamente intermediare numite **comutatoare de pachete**. La primirea unui pachet, fiecare calculator verifică câmpul de adresă. Dacă pachetul este destinat acestuia va fi procesat; dacă este adresat unui alt destinatar, pachetul va fi ignorat sau transmis mai departe către un alt nod din rețea.
- Într-o rețea de difuzare, canalul de comunicații este împărțit de toate calculatoarele din rețea (de exemplu, rețelele *wireless*); pachetele trimise de oricare dintre acestea sunt primite de toate celelalte. Sistemele de difuzare de obicei oferă și posibilitatea trimiterii unui pachet către toate destinațiile prin utilizarea unui cod special în câmpul de adresă. Când un pachet cu acest cod este transmis, acesta este primit și procesat de fiecare calculator din rețea.

Un comutator de pachete ia un pachet care sosește pe una dintre legăturile de comunicație de intrare și îl trimite mai departe pe una dintre legăturile de ieșire. Cele mai cunoscute comutatoare de pachete sunt ***router***-ele și ***switch***-urile. *Switch*-urile sunt de obicei utilizate în rețelele de acces, iar *router*-ele în nucleele rețelelor. Succesiunea legăturilor de comunicație și a comutatoarelor traversate de un pachet de la calculatorul sursă către cel destinație este cunoscută ca o **rută** (*route*) sau o **cale** (*path*) prin rețea [2]. Deseori sunt posibile mai multe rute, de lungimi diferite, găsirea celor bune fiind un lucru important.

1.2 Performanța rețelelor

Performanța unei rețele definește calitatea serviciilor oferite utilizatorilor acesteia și este afectată de diverși factori precum lățimea de bandă, rata de transfer, întârzierile sau pierderile.

În domeniul telecomunicațiilor, mai exact când este vorba de procesarea semnalelor, lățimea de bandă (*bandwidth*) reprezintă dimensiunea unui domeniu de frecvențe disponibil și se măsoară în Hertzi. În cazul unei legături de comunicație dintre două calculatoare lățimea de bandă reprezintă o măsură a capacității **maxime** de transmitere a informației prin acea legătură într-un interval de timp și se măsoară în biți pe secundă.

Rata de transfer efectivă (*throughput* sau *data rate*) reprezintă cantitatea de informație ce se poate transmite la **un moment dat** printr-o legătură de comunicație, aceasta putând să

depindă de unele restricții, de numărul de utilizatori care folosesc rețeaua în cazul în care aceasta este partajată, de perturbațiile care o afectează, etc.

Cel mai adesea, comutatorul de pachete trebuie să transmită printr-o singură legătură de comunicație pachete venite de la mai multe calculatoare conectate la acesta. Cea mai utilizată metodă se numește *store-and-forward* (stochează și trimite mai departe) prin care pachetele sunt stocate într-o memorie tampon (*buffer*) și sunt transmise mai departe numai dacă legătura de comunicație este liberă. Aceasta duce la apariția unei întârzieri (*queuing delay*) a cărei valoare depinde de numărul de pachete primite de comutator precum și de gradul de congestie al rețelei.

Dacă întârzierile sunt prea mari și memoria tampon se umple, toate pachetele care sosesc la comutator pentru a fi trimise mai departe prin legătura de comunicație congestionată vor fi refuzate, ceea ce duce la pierderi de pachete (*packet loss*).

Alte întârzieri care pot să apară se datorează timpului necesar pentru procesarea pachetului pentru a afla prin care legătură de comunicație trebuie trimis, timpului necesar pentru a transmite un pachet (dimensiunea pachetului raportată la rata de transfer a legăturii de comunicație), precum și timpului de propagare prin mediul de comunicație (lungimea acestuia raportată la viteza de propagare, ce poate fi egală sau puțin mai mică decât viteza luminii, în funcție de caracteristicile fizice ale mediului).

Toate întârzierile menționate, precum și altele, cumulate determină latența rețelei (*latency*) ce poate fi testată și minimizată.

1.3 Tipuri de rețele

Dimensiunea unei rețele poate fi exprimată de aria geografică pe care o ocupă și de numărul de calculatoare care fac parte din rețea. Rețelele pot acoperi orice, de la o mână de calculatoare, într-o singură cameră, până la milioane distribuite pe întregul glob.

În funcție de dimensiunile lor, rețelele de calculatoare se pot clasifica conform criteriilor menționate în Figura 3.

1 m	Proximitate	Personal area network (PAN)
10 m	Cameră	Local area network (LAN)
100 m	Clădire	
1 km	Campus	
10 km	Localitate	Metropolitan area network (MAN)
100 km	Țară	Wide area network (WAN)
1000 km	Continent	
10,000 km	Planetă	Internet

Figura 3. Clasificarea rețelelor după dimensiunea lor [6]

1.3.1 Rețele personale (PAN)

Astfel de rețele permit dispozitivelor să comunice în proximitatea unei persoane. Un exemplu comun este o rețea fără fir care conectează un calculator cu perifericele acestuia. Aproape fiecare calculator are un monitor atașat, tastatură, mouse și imprimantă și în multe cazuri aceste conexiuni sunt realizate cu cabluri. Pentru a ajuta utilizatorul, unele companii s-au reunit pentru a proiecta o rețea *wireless* cu rază scurtă de acțiune numită Bluetooth, pentru a conecta aceste componente fără cabluri. Un alt exemplu de utilizare este conectarea telefonului mobil cu o cască fără fir cu scopul de a transmitere a convorbirilor sau cu un vehicul pentru a asculta muzică în format digital prin intermediul sistemului audio al acestuia [6].

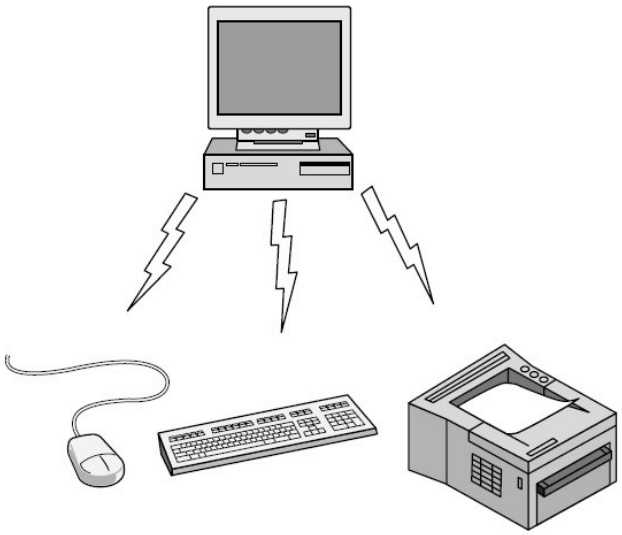

Figura 4. Rețea de tip PAN [6]

Rețelele de tip PAN pot fi construite și cu alte tehnologii care comunică pe distanțe scurte, cum ar fi RFID.

1.3.2 Rețele locale (LAN)

O rețea LAN este o rețea privată care funcționează în interiorul și în apropierea unei singure clădiri, cum ar fi o casă, un birou sau o fabrică. LAN-urile sunt utilizate pe scară largă pentru a conecta calculatoarele personale și alte echipamente electronice pentru a le permite să partajeze resurse (de exemplu, imprimante) și să facă schimb de informații.

Rețelele LAN fără fir sunt din ce în ce mai preferate de utilizatori deoarece oferă libertate și comoditate. În aceste sisteme, fiecare calculator are un modem radio și o antenă pe care o utilizează pentru a comunica în rețea. În majoritatea cazurilor, fiecare dintre ele discută cu un dispozitiv numit AP (Punct de Acces) sau cu un *router wireless* (Figura 5). Există un standard pentru rețelele fără fir numit IEEE 802.11, cunoscut popular sub numele de Wi-Fi, care permite transferul de date cu viteze până la sute de Mbps.

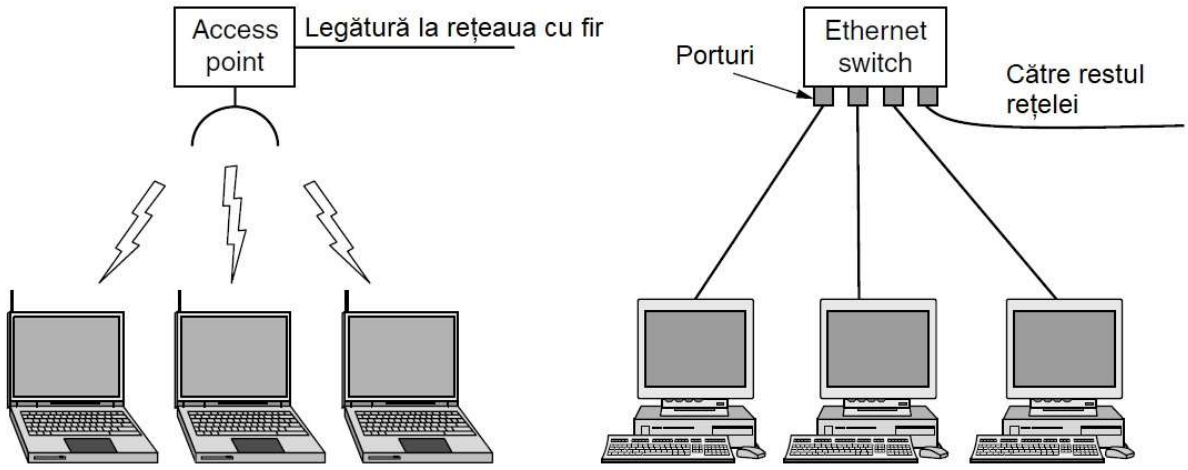

Figura 5. Rețea de tip LAN, fără fir și cu fir [6]

Rețelele LAN cu fir utilizează o gamă largă de tehnologii de transmisie diferite. Majoritatea folosesc fire de cupru, dar unele utilizează fibră optică, raza de acoperire a lor fiind restricționată din considerente fizice ale acestora. În mod obișnuit, LAN-urile cu fir transferă date cu viteze de 100 Mbps până la 1 Gbps, au o întârziere mică (microsecunde sau nanosecunde) iar gradul de apariție a erorilor este foarte mic. LAN-urile mai noi pot funcționa cu o viteză de până la 10 Gbps. Comparat cu rețelele fără fir, LAN-urile cu fir le depășesc în performanță.

Majoritatea rețelelor de tip LAN sunt construite din legături punct-la-punct, iar IEEE 802.3, numit popular *Ethernet*, este, de departe, cel mai cunoscut tip de rețea LAN cu fir. Într-o astfel de rețea fiecare calculator este conectat la un *switch*, în unul din porturile acestuia, printr-o legătură punct-la-punct (Figura 5). Rolul *switch*-ului este de a retransmite pachetele între calculatoarele atașate la acesta, utilizând adresa existentă în antetul fiecărui pachet pentru a determina care este destinatarul. Pentru a construi rețele LAN mai mari, *switch*-urile pot fi conectate între ele prin porturile lor.

De asemenea, este posibilă împărțirea unei rețele LAN fizice mari în două rețele LAN logice mai mici. Acest lucru poate fi util, de exemplu, atunci când două departamente ale unei companii ar putea avea calculatoare conectate în aceeași rețea fizică, deoarece acestea s-ar afla în aceeași aripă a clădirii. Rețeaua ar putea fi mai ușor gestionat dacă pachetele de date ar fi transmise separat doar între calculatoarele fiecărui departament. Acest lucru este posibil prin separarea logică a rețelei în două rețele de tip Virtual LAN sau VLAN, la nivelul *switch*-urilor [6].

1.3.3 Rețele metropolitane (MAN)

Rețelele metropolitane sunt de dimensiunile unei localități, conectează mai multe rețele de tip LAN iar legăturile și echipamentele sale de comunicații sunt, în general, deținute fie de un consorțiu de utilizatori, fie de un singur furnizor de rețea care vinde serviciul utilizatorilor.

Prin urmare, acest nivel al serviciului furnizat fiecărui utilizator trebuie să fie negociat cu operatorul MAN, iar în mod normal sunt specificate anumite garanții de performanță.

Rețelele de tip MAN sunt utilizate de furnizorii de Internet (ISP), de servicii de telefonie sau cablu TV pentru a furniza servicii către consumatori. Sunt extrem de eficiente și asigură o comunicare rapidă prin intermediul unor suporturi fizice de mare viteză, cum ar fi cablurile cu fibră optică.

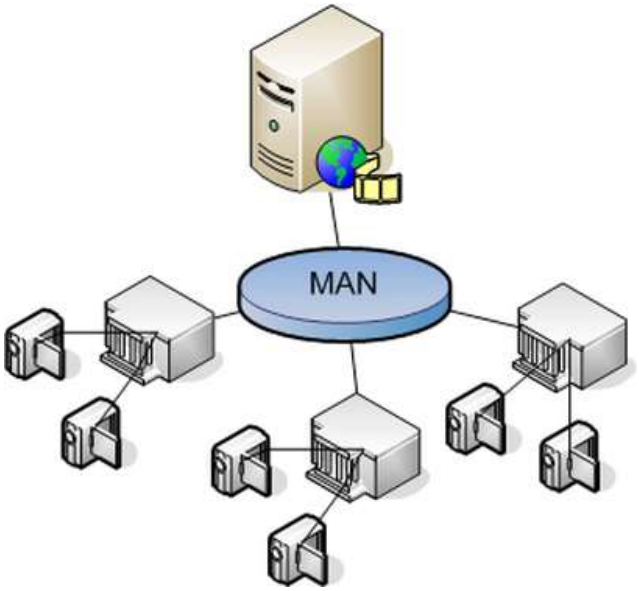

Figura 6. Rețea de tip MAN [6]

1.3.4 Rețele de arie largă (WAN)

Rețelele WAN se întind pe arii geografice mari, adesea o țară sau continent. Acestea sunt utilizate pentru a conecta rețele de tip LAN și alte tipuri de rețele împreună, astfel încât utilizatorii și calculatoarele dintr-o locație să poată comunica cu utilizatorii și calculatoarele din alte locații. Multe rețele WAN sunt construite pentru o anumită organizație și sunt private, iar adesea sunt construite folosind linii de comunicație închiriate ce pot fi foarte scumpe.

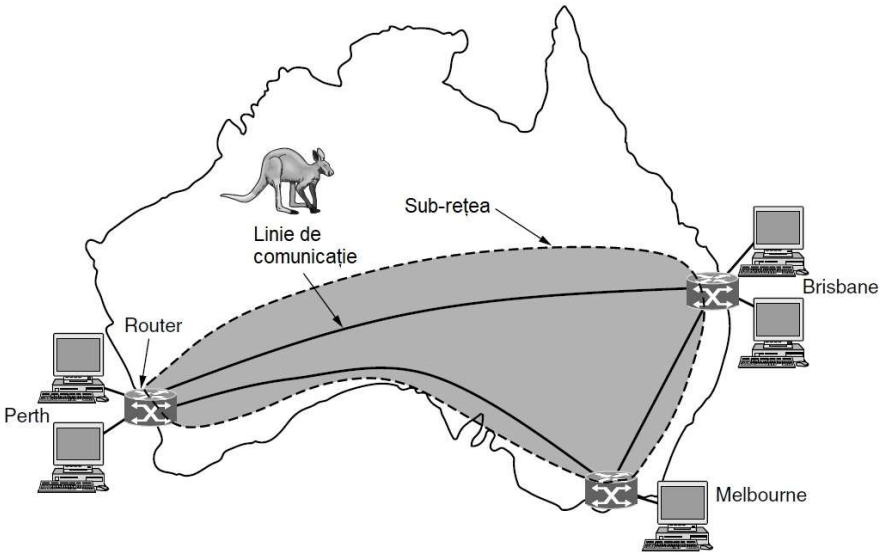

Figura 7. Rețea de tip WAN [6]

Alte rețele WAN sunt construite de furnizorii de servicii de Internet și oferă conexiuni între o rețea LAN a unei organizații și Internet. Astfel, utilizatorii pot comunica între locații îndepărtate prin legături virtuale care utilizează capacitatea de bază a conexiunii la Internet. Acest modalitate este denumită VPN (Rețea Virtuală Privată), având avantajul virtualizării, și anume faptul că oferă o reutilizare flexibilă a unei resurse, conectivitatea la Internet [6].

1.4 Topologii fizice de rețea

Topologia fizică a unei rețele de calculatoare se referă la structura acesteia la modul de plasare a diferitelor componente ale ei, inclusiv localizarea echipamentelor de interconectare și a mediilor de comunicație. Distanțele dintre noduri, conexiunile fizice, ratele de transmisie sau tehnologiile de comunicație pot diferi între două rețele, deși topologiile lor pot fi identice.

1.4.1 Topologie tip Magistrală (Bus)

O topologie de tip magistrală utilizează un singur cablu pentru a conecta mai multe calculatoare. De cele mai multe ori, pentru conectarea acestora la magistrală sunt utilizați conectori de tip T (numiți astfel deoarece au forma literei T) și cabluri coaxiale [7].

O altă componentă importantă a unei topologii de tip magistrală este necesitatea terminării. Pentru a împiedica reflectarea semnalelor electrice înapoi în cablu, la capetele acestuia se atașează niște dispozitive numite terminatoare. În lipsa lor, sau în cazul în care cablul se întrerupe undeva, rețeaua nu funcționează.

Într-o astfel de topologie doar un singur calculator poate transmite un pachet la un moment dat, iar acesta se deplasează în ambele direcții. Aceasta înseamnă că rețeaua este ocupată până când calculatorul de destinație acceptă pachetul. Calculatoarele din rețea ascultă tot traficul, dar acceptă numai pachetele care le sunt adresate. Pachetele de difuzare sunt o excepție, deoarece toate calculatoarele din rețea le acceptă [7].

Numărul de calculatoare dintr-o astfel de rețea are o influență majoră asupra performanței rețelei: cu cât numărul acestora și al pachetelor este mai mare, cu atât rețeaua funcționează mai greu [7].

Topologia de tip magistrală este una pasivă, calculatoarele ascultând sau trimițând date, fără a le retrimite sau regenera, deci, dacă unul dintre ele are probleme sau dispare din rețea, funcționarea acesteia nu este afectată [7].

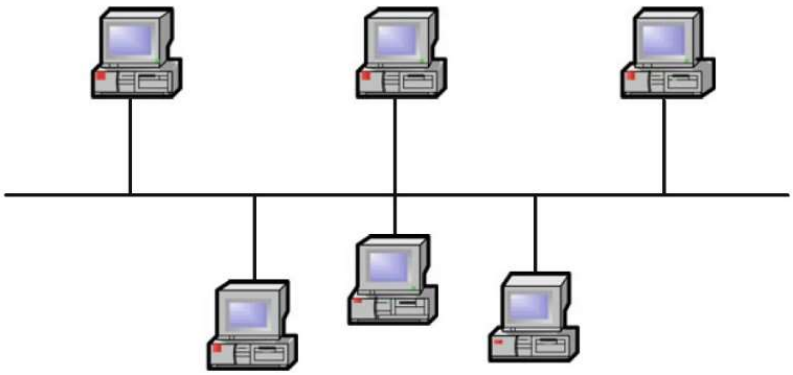

Figura 8. Topologie de tip magistrală [1]

Avantaje [7]:

- Costuri reduse.
- Ușurință în instalare.

Dezavantaje [7]:

- Depanare dificilă.
- Încetinirea rețelei o dată cu creșterea traficului.
- Scalabilitate redusă.

1.4.2 Topologie tip Stea (Star)

Este cea mai utilizată topologie de rețea, toate calculatoarele fiind conectate într-un *switch* central. Spre deosebire de topologia tip magistrală, dacă o legătură se întrerupe și afectează un calculator, celelalte își păstrează accesul la rețea.

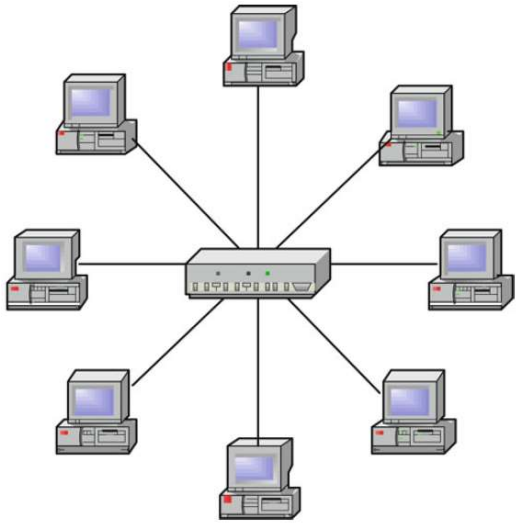

Figura 9. Topologie de tip stea [1]

Avantaje [7]:

- Centralizarea cablurilor.
- Ușurință în managementul și monitorizarea rețelei.
- Scalabilitate mărită.

Dezavantaje [7]:

- Dependența de echipamentul central.
- Costuri crescute.

1.4.3 Topologie tip Inel (Ring)

În topologia de tip inel, fiecare calculator este atașat de calculatoarele din apropiere prin legături punct-la-punct, astfel încât întreaga rețea are forma unui inel în care pachetele circulă într-o singură direcție, fiind transmise de la un calculator la altul. Fiecare verifică un pachet și, dacă nu îi este destinat, îl trimite mai departe. Nu există un capăt al rețelei și, prin urmare, nu este nevoie de elemente terminatoare. Dacă unul din calculatoare are probleme sau dispare din rețea, aceasta nu mai funcționează.

Fiecare calculator are acces în mod egal la rețea, astfel încât cele care folosesc rețeaua mai intens nu le afectează pe celelalte.

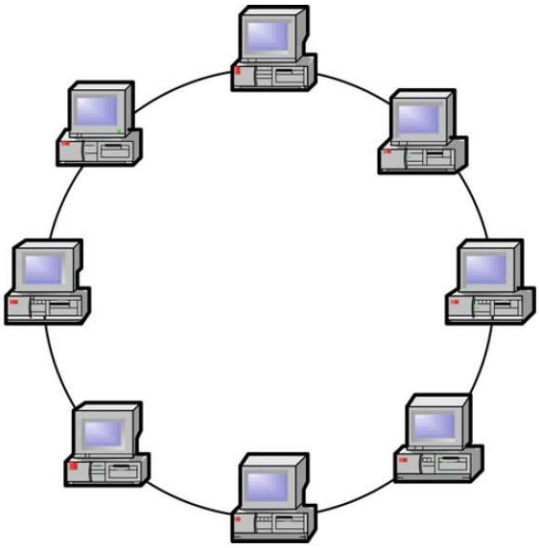

Figura 10. Topologie de tip inel [1]

Avantaje [7]:

- Performanță bună a rețelei.
- Degenerare redusă a semnalului.

Dezavantaje [7]:

- Dependența de funcționarea fiecărui calculator.
- Mentenanță dificilă.

1.4.4 Topologie tip Plasă (Mesh)

O topologie de tip plasă nu este foarte frecventă în rețelele de calculatoare ci mai curând în rețelele naționale de telefonie. Într-o astfel de rețea fiecare calculator are o legătură de comunicație cu fiecare componentă a rețelei [7]. Dacă o legătură între oricare dintre calculatoare nu mai funcționează, va fi disponibilă o rută alternativă. O topologie ca aceasta este costisitoare, dar poate fi necesară pentru aplicații unde este vital ca mașinile de calcul să nu piardă contactul între ele [1].

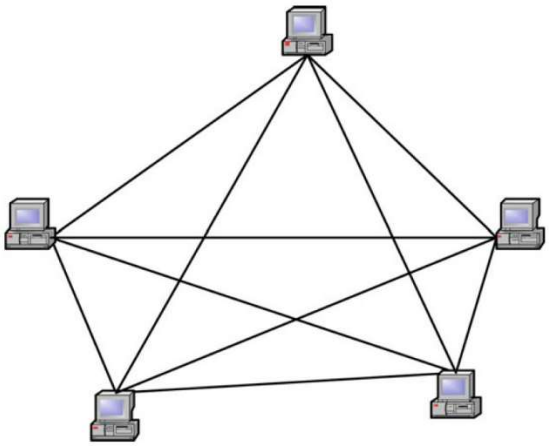

Figura 11. Topologie de tip plasă [1]

1.4.5 Topologie tip Arbore (Tree)

Topologia de tip arbore poate fi văzută ca o combinație de rețele de tip stea aranjate ierarhic. La periferia rețelei se află calculatoarele gazdă, în vârful ierarhiei se află cel care administrează rețeaua, iar nodurile intermediare constau în comutatoare de pachete (*switch*-uri). Ca și într-o rețea de tip stea, întreruperea unei legături de comunicație poate izola de rețea unul sau mai multe calculatoare.

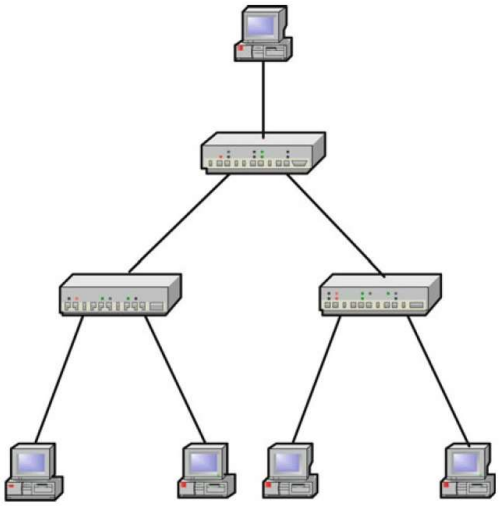

Figura 12. Topologie de tip arbore [1]

Avantaje [7]:

- Ușurință în managementul și monitorizarea rețelei.
- Scalabilitate mărită.

Dezavantaje [7]:

- Dependența de calculatorul central.
- Mentenanță dificilă pentru rețelele mari.

1.5 Medii de comunicație

Mediul de comunicație este suportul fizic prin intermediul căruia datele sunt transmise într-o rețea. Există mai multe tipuri de suporturi media, iar selectarea celui potrivit depinde de mai mulți factori, cum ar fi costul acestuia, eficiența transmiterii datelor sau rata de transfer. Pentru rețelele de calculatoare se utilizează trei clase de medii de transmisie: fir de cupru (cablu coaxial, cablu TP – *twisted pair*), datele fiind transmise sub formă de curent electric, fibră optică, prin care se transmit date sub formă de undă luminoasă și aer, caz în care se folosesc undele radio.

1.5.1 Cablu coaxial

Cablul coaxial este alcătuit din doi conductori din cupru, unul în interiorul celuilalt, separați de o izolație din plastic. Conductorul interior este un fir de cupru gros iar conductorul exterior este o plasă cilindrică din sârmă de cupru subțire ce acționează și ca un ecran pentru

conductorul interior, ajutând la reducerea interferențelor electromagnetice din afara cablului. Un manșon din plastic protejează cablul. Pentru conexiuni se utilizează mufe de tip BNC.

Figura 13. Cablu coaxial și mufa de tip BNC

A fost utilizat în anii ’80 – ’90 pentru primele rețele Ethernet cu viteze de până la 10 Mbps, rețele care, cel mai adesea, erau de tip magistrală. Aveau impedanță de 50 Ω, spre deosebire de cele utilizate pentru transmiterea semnalelor de televiziune, care aveau impedanța de 75 Ω.

1.5.2 Cablu TP

TP este prescurtarea de la *Twisted Pair*, însemnând Perechi Torsadate (răsucite). Un cablu TP pentru rețele de calculatoare conține 8 fire din cupru izolate individual și răsucite două câte două, formând 4 perechi de fire torsadate. Răsucirea firelor se face cu scopul anulării interferențelor electromagnetice datorate semnalelor electrice care circulă prin fire alăturate (*crosstalk*).

Cel mai cunoscut tip de astfel de cablu este UTP (*Unshielded Twisted Pair*) în care cele patru perechi de cabluri sunt acoperite cu o manta izolatoare, fără a avea vreun fel de ecranare față de perturbațiile din exterior. Legat de acest aspect, există alte trei tipuri de cabluri TP: cu ecranare globală a tuturor perechilor, cu ecranare individuală a fiecărei perechi și cu ambele variante utilizate în același cablu. Acestea se numesc FTP (*Foiled Twisted Pair*), STP (*Shielded Twisted Pair*) sau ScTP (*Screened Twisted Pair*) fiind folosite arbitrar, fără ca una din denumiri să descrie strict un anumit tip de cablu. Pentru conexiuni se utilizează mufe de tip RJ-45.

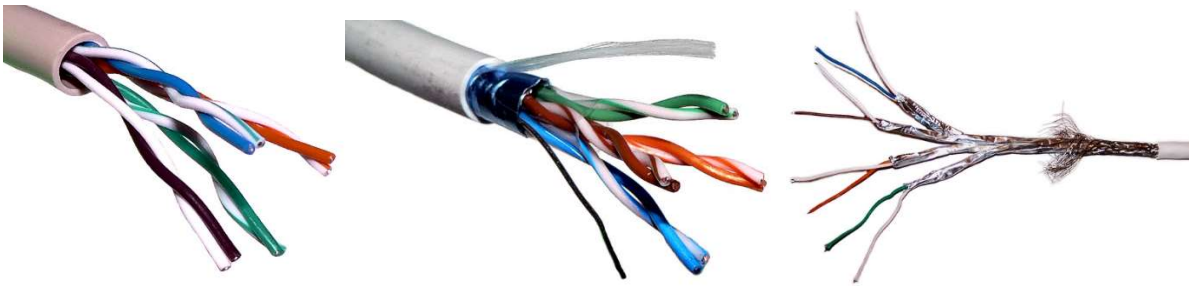

Figura 14. Cabluri TP neecranat și cu diferite tipuri de ecranare

Figura 15. Mufă de tip RJ-45

Utilizarea preponderentă a cablului UTP se datorează prețului scăzut, dimensiunilor reduse și ușurinței în instalare. Distanța maximă de comunicație este de 100 m iar viteza de transmitere a datelor variază între 100 Mbps și 10 Gbps în funcție de categoria cablului (CAT5, CAT5e, CAT6, CAT6a, CAT7), fiecare dintre acestea având lățimi de bandă diferite ce depind de caracteristicile constructive.

1.5.3 Fibră optică

Fibra optică constă dintr-un miez transparent și flexibil realizat din sticlă sau material plastic, un strat protector din sticlă transparentă și o manta exterioară din plastic cu rol de protecție împotriva factorilor externi. Deși miezul și stratul protector sunt ambele transparente, există o diferență majoră între ele: indexul de refracție, mic pentru stratul protector și mare pentru miez. Rezultatul acestei diferențe este obținerea unei reflexii interne totale, iar efectul este că lumina introdusă într-o fibră optică nu poate părăsi miezul, ci călătorește de la un capăt la celălalt al acesteia.

Avantajele fibrei optice constau în obținerea unor rate de transfer și a unor distanțe de comunicație mari. În plus, deoarece nu folosesc semnale electrice, nu sunt afectate de interferențele electromagnetice externe.

Dezavantajele acesteia constau în prețul mai mare decât al cablurilor din cupru, instalarea dificilă, pentru unirea (sudura) a două fire de fibră optică fiind necesare echipamente speciale destul de scumpe, precum și necesitatea utilizării unor echipamente de conversie a semnalelor optice în semnale electrice pentru interconectarea cu echipamentele electronice.

Figura 16. Cablu cu fibră optică și diverși conectori

Există două tipuri principale de fibră optică: multi-mod (*multimode*) și mono-mod (*singlemode*).

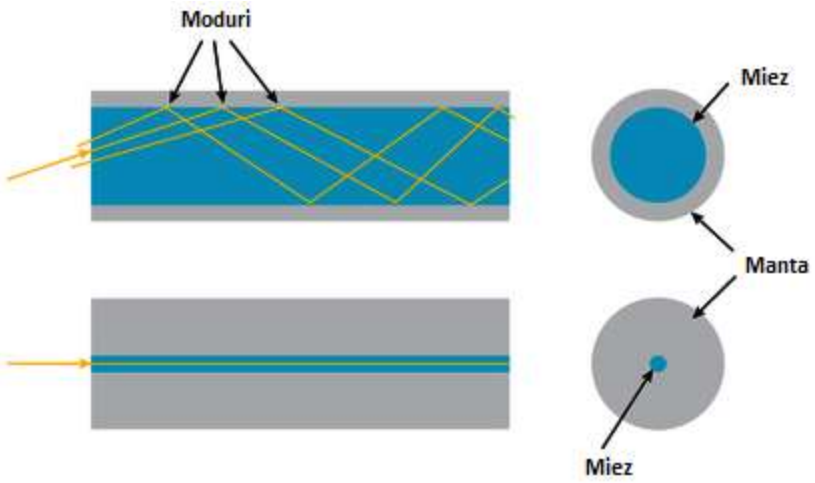

Figura 17. Fibra optică multi-mod și mono-mod

Fibra optică multi-mod are dimensiunea miezului între 50-100 µm și poate accepta până la 4 fascicule luminoase. Pentru emisia luminii se folosesc diode LED, iar din acest motiv conexiunile sunt mai ușor de realizat iar prețul echipamentelor este mai mic. Dezavantajul principal este obținerea unor distanțe de comunicație relativ scurte (până la 2 km) datorită dispersiei luminii în interiorul fibrei, dar suficiente pentru utilizarea în clădiri sau campusuri. Rata maximă de transfer variază în funcție de lungimea fibrei între 100 Mbps și 10 Gbps [8].

Fibra optică mono-mod are dimensiunea miezului între 8-10 µm și acceptă un singur fascicul luminos. Pentru emisia luminii se folosesc diode Laser, iar din acest motiv prețul echipamentelor este mai mare, dar și distanța de comunicație crește, putând depăși 100 km. Rata maximă de transfer variază în funcție de lungimea fibrei și poate ajunge până la 100Gbps [9].

1.5.4 Aer

Comunicațiile fără fir (*wireless*) reprezintă transferul de informații între două puncte prin intermediul undelor electromagnetice. Sunt necesare atunci când amplasarea unui cablu poate fi dificilă sau scumpă, sau atunci când se dorește conectarea la rețea a unui dispozitiv mobil. Acesta este și motivul pentru care acest tip de comunicații este foarte popular, deși în practică se observă că, în comparație cu comunicațiile prin cablu, ratele maxime de transfer sunt mai mici, probabilitatea de apariție a erorilor este mai mare, și sunt mai ușor influențate de către condițiile meteo, obstacole sau perturbații electromagnetice.

Transmiterea și recepția datelor se realizează cu ajutorul unei antene ce poate fi direcțională, canalizând undele electromagnetice într-o anumită direcție, sau omnidirecțională, caz în care semnalul este răspândit în toate direcțiile. Frecvențele între 30 MHz și 1 GHz (unde radio) sunt potrivite pentru utilizarea antenelor omnidirecționale, iar cele între 1 GHz și 100 GHz (microunde) pentru cele direcționale [11]. Cu cât frecvența este mai mare, cu atât rata de transfer este mai mare, dar și distanța de comunicație va fi mai mică datorită creșterii atenuării. Astfel, a apărut o mare diversitate de tehnologii de comunicație, fiecare cu avantaje și dezavantaje, dar adaptate unui anumit domeniu de aplicabilitate (Figura 18).

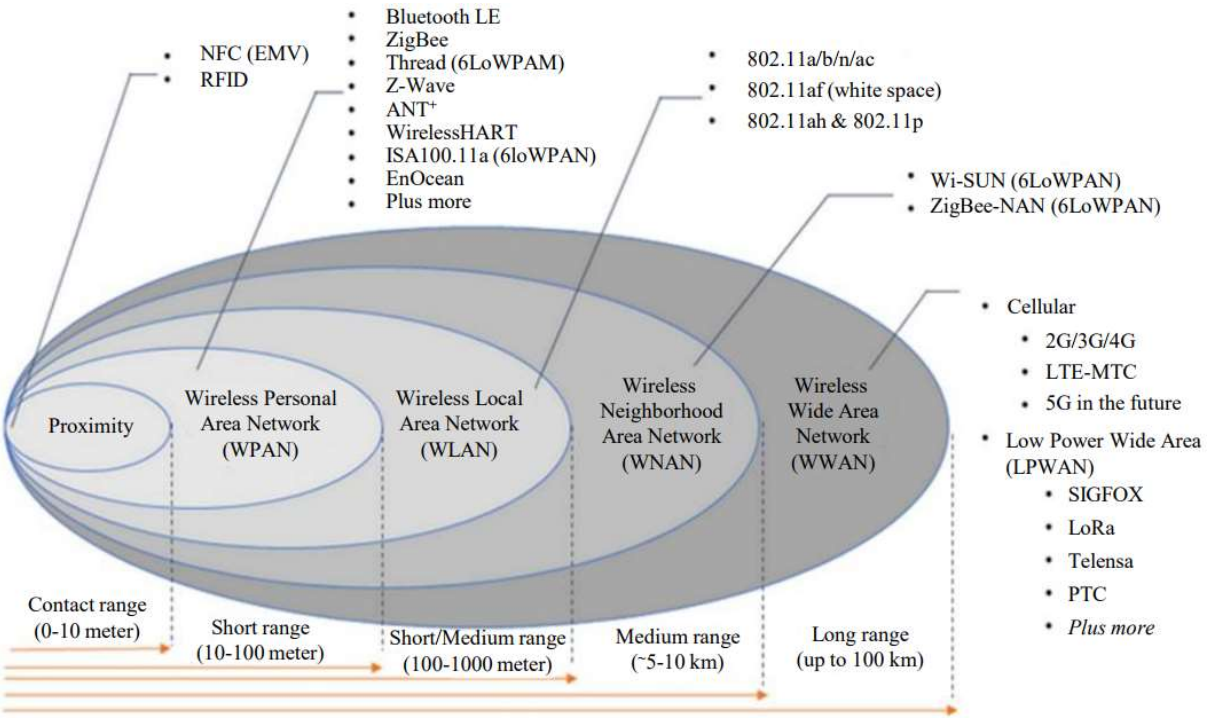

Figura 18. Caracteristicile a diferite tipuri de tehnologii de comunicație fără fir [10]

1.6 Protocoale de comunicație

1. Un protocol este un set standard de reguli și reglementări care permite ca două dispozitive electronice să se conecteze și să facă schimb de informații între ele [7].

2. Un protocol definește formatul și ordinea mesajelor schimbate între două sau mai multe entități care comunică, precum și acțiunile întreprinse în legătură cu transmiterea și / sau primirea unui mesaj sau a unui alt eveniment [2].

3. Un protocol este un acord între părțile care comunică despre modul în care urmează să se desfășoare comunicarea [12].

Deoarece rețelele de calculatoare sunt sisteme complexe și de mari dimensiuni, pentru proiectarea acestora nu a fost posibilă utilizarea unui singur protocol de comunicație. Astfel, arhitectura unei rețele a fost împărțită în **niveluri** (*layer*) așezate unul deasupra celuilalt, fiecare utilizând propriul protocol. Scopul fiecărui nivel este de a oferi servicii nivelului superior și de a utiliza servicii furnizate de nivelul inferior, fără a oferi detalii despre cum acestea sunt implementate. Numărul, denumirea, conținutul și funcțiile acestora sunt diferite de la o rețea la alta.

Un set de protocoale (câte unul pentru fiecare nivel) formează o **stivă de protocoale** (*stack*). Se numește stivă deoarece nivelurile sunt aranjate în mod ierarhic. Pentru ca două calculatoare să poată comunica, ambele trebuie să utilizeze aceeași stivă de protocoale, iar fiecare nivel al stivei unui calculator trebuie să comunice cu nivelul echivalent al celuilalt. Acest lucru permite calculatoarelor care rulează sisteme de operare diferite să poată comunica între ele.

Un set complet de niveluri și protocoale formează **arhitectura de rețea**. Cele mai cunoscute modele de referință ce pot fi utilizate pentru a crea o arhitectură de rețea sunt OSI și TCP/IP.

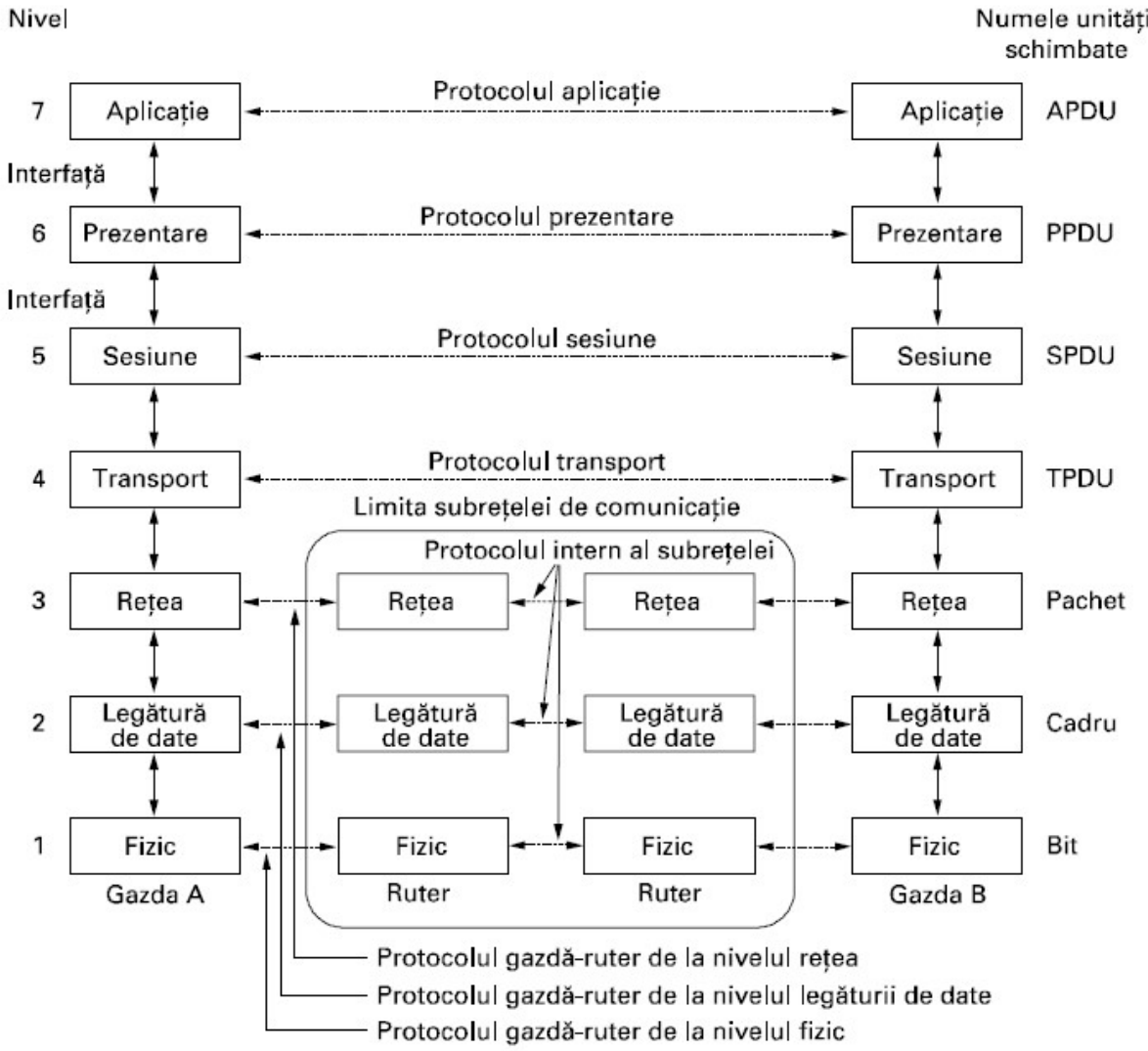

Figura 19. Modelul OSI [12]

Modelul OSI a fost realizat de Organizația Internațională pentru Standardizare (ISO) și publicat în anul 1984, fiind în acest moment un model teoretic. Acesta cuprinde șapte niveluri (Figura 19), fiecare având o funcție distinctă și utilizând o anumită unitate de date (PDU) specifică protocolului implementat:

1. Nivelul Fizic are rolul de transmitere a biților de la un nod al rețelei la altul prin mediul de comunicație, protocolul folosit depinzând de natura acestuia. Unitatea de date utilizată este bitul.
2. Nivelul Legătură de date se ocupă de transferul sigur al datelor între două noduri. Biții care trebuie transmiși sunt grupați în cadre (*frame* – unitatea de date a acestui nivel) și transmiși către destinatar, acesta răspunzând cu un mesaj de confirmare, pe baza căruia se rezolvă eventualele erori. Tot la acest nivel se realizează și controlul fluxului, în funcție de capacitatea de recepție a destinatarului.

3. Nivelul Rețea realizează conectivitatea, se ocupă de adresarea și selectarea rutei de transmitere a datelor în rețele multi-nod, rezolvă problemele datorate congestiilor, administrează și controlează traficul din rețea. Unitatea de date este pachetul (*packet*).
4. Nivelul Transport este responsabil cu transmiterea mesajelor între două calculatoare gazdă, realizând între acestea o conexiune logică. Mesajele ce trebuie transmise sunt împărțite în segmente (unitatea de date a acestui nivel) și se verifică recepționarea lor în ordinea în care au fost transmise, orice erori fiind detectate și corectate. Realizează și controlul fluxului de date.
5. Nivelul sesiune stabilește, administrează și încheie sesiuni între aplicații. Unele elemente de securitate, cum ar fi autentificarea, aparțin acestui nivel.
6. Nivelul prezentare preia datele furnizate de aplicație și se ocupă de formatarea, compresia și criptarea lor.
7. Nivelul aplicație furnizează serviciile de rețea necesare aplicațiilor. Nu reprezintă aplicațiile în sine, ci cadrul pe care acestea se bazează.

TCP/IP este un model utilizat la nivel mondial care conține protocoale adaptate pentru Internet. Modelul de referință are patru niveluri (Figura 20), dar poate avea și cinci (Nivelul de acces la rețea poate fi împărțit în Fizic și Legătură de date) în funcție de preferințe.

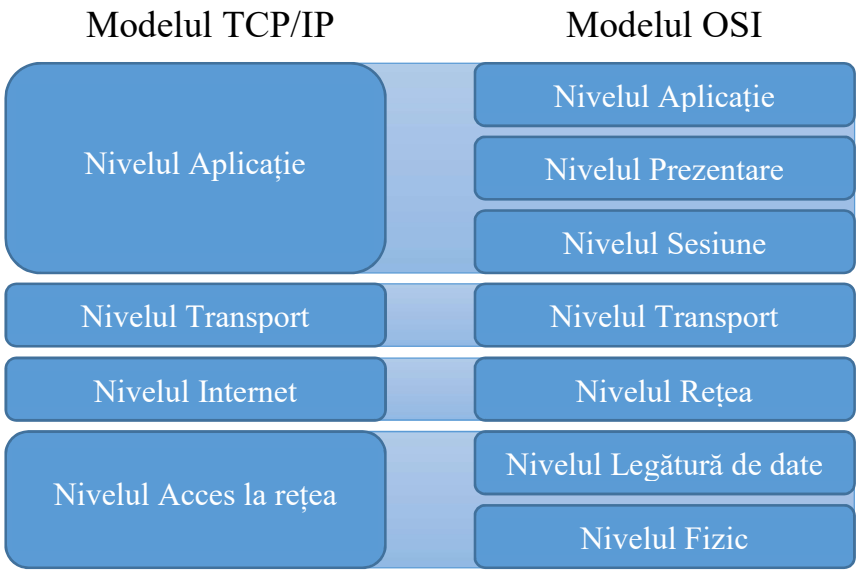

Figura 20. Modelul TCP/IP

Utilitatea celor patru niveluri este similară cu cea a nivelurilor modelului OSI și va fi descrisă pe larg în următoarele capitole.

Capitolul 2. Nivelul aplicație

Nivelul Aplicație interacționează cu aplicațiile software ce necesită acces la servicii de rețea, cuprinzând protocoale de comunicație și metode de interfață utilizate în comunicația directă dintre procese, fără a fi specificat un anumit format al datelor.

În funcție de modul în care o aplicație ce rulează pe un calculator client accesează serviciile de rețea necesare și de felul în care aceasta este structurată pe multitudinea de calculatoare client conectate la rețea, se pot defini două mari tipuri de arhitecturi de aplicații: **Client-Server** și **Peer-to-Peer** (P2P).

Arhitectura Client-Server se bazează pe existența unui calculator gazdă ce funcționează fără întrerupere (Server) și care preia cereri de la o mare varietate de Clienți, furnizându-le serviciile necesare. Acest tip de arhitectură este caracterizată de alte câteva particularități: doi Clienți nu pot comunica în mod direct unul cu celălalt, Serverul furnizează întotdeauna răspunsuri pe baza cererilor Clienților, iar pentru a putea fi contactat oricând, acesta are o **adresă IP** fixă și cunoscută. Unele dintre cele mai cunoscute aplicații de tip Client-Server sunt Web, FTP sau E-mail.

În cazul arhitecturilor de tip Peer-to-Peer comunicația are loc în mod direct între doi Clienți conectați unul cu celălalt, fără a fi necesară existența unui server dedicat. Aplicațiile de acest tip includ telefonia prin Internet, partajarea de fișiere, acceleratoare de descărcare a fișierelor. Uneori, aplicațiile de tip Peer-to-Peer folosesc totuși un Server (arhitectură hibridă) dar numai pentru obținerea unui suport minim, de exemplu identificarea utilizatorilor în aplicațiile de mesagerie instant, mesajele fiind apoi transmise în mod direct între aceștia.

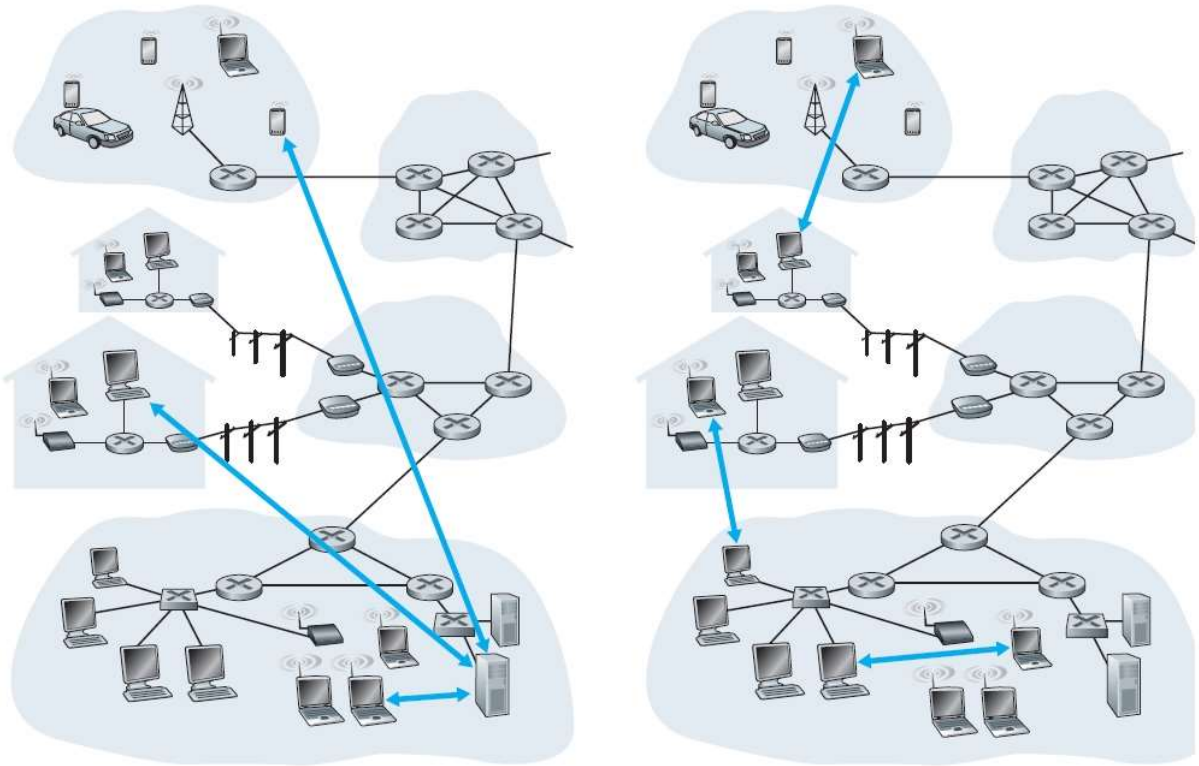

Figura 21. Arhitecturi Client-Server și Peer-to-Peer

Deoarece pe un calculator gazdă pot rula mai multe procese ce necesită sau furnizează acces la servicii de rețea, în afară de adresa IP este necesară identificarea procesului cu ajutorul unui **port** (un număr între 0 și 65535). De exemplu aplicațiile Web folosesc portul 80, FTP porturile 20 și 21 sau E-mail (prin protocolul SMTP) portul 25.

Asocierea dintre o adresa IP și un port formează un **soclu** (*socket*).

2.1 DNS – Sistemul numelor de domenii

Cu toate că teoretic programele ar putea să se refere la sistemele gazdă, la cutiile poștale și la alte resurse prin adresa lor de rețea (de exemplu prin adresa IP), aceste adrese sunt greu de memorat de către oameni. De asemenea, în trimiterea de poștă electronică la tana@128.111.24.41 ar însemna că dacă furnizorul de servicii Internet sau organizația Tanei mută serverul de poștă pe o mașină diferită, cu o adresă IP diferită, adresa ei de e-mail se va schimba. De aceea au fost introduse nume ASCII pentru a separa numele mașinilor de adresele mașinilor. În acest fel, adresa Tanei ar putea fi ceva de genul tana@art.ucsb.edu. Cu toate acestea, rețeaua înțelege numai adrese numerice, deci este necesar un mecanism care să convertească șirurile ASCII în adrese de rețea.

Esența DNS-ului constă într-o schemă ierarhică de **nume de domenii** și a unui sistem de baze de date distribuite pentru implementarea acestei scheme de nume. În principal este utilizat pentru a pune în corespondență numele sistemelor gazdă și adresele destinațiilor de e-mail cu adresele IP, dar poate fi utilizat și pentru alte scopuri. Foarte pe scurt, DNS este utilizat după cum urmează. Pentru a stabili corespondența dintre un nume și o adresă IP, programul de aplicație apelează o procedură de bibliotecă numită *resolver*, transferându-i numele ca parametru. *Resolver*-ul trimite un pachet la serverul DNS local, care caută numele și returnează adresa IP către *resolver*, care o returnează apelantului. Având adresa IP, programul poate stabili o conexiune cu destinația. Portul TCP standard pentru protocolul DNS este 53.

Conceptual, Internetul este divizat în peste 1000 domenii de nivel superior, fiecare domeniu cuprinzând mai multe sisteme gazdă. Fiecare domeniu este partiționat în subdomenii și acestea sunt, la rândul lor, partiționate ș.a.m.d. Toate aceste domenii pot fi reprezentate ca un arbore, așa cum se arată în Figura 21. Frunzele arborelui reprezintă domenii care nu au subdomenii (dar, bineînțeles, conțin sisteme). Un domeniu frunză poate conține un singur sistem gazdă sau poate reprezenta o firmă, deci să conțină mii de sisteme gazdă. Domeniile de pe primul nivel se împart în două categorii: generice și de țări. Domeniile generice sunt com (comercial), edu (instituții educaționale), gov (guvernul federal al SUA), int (organizații internaționale), mil (forțele armate ale SUA), net (furnizori Internet) și org (organizații nonprofit). Domeniile de țări includ o intrare pentru fiecare țară, cum se definește în ISO 3166.

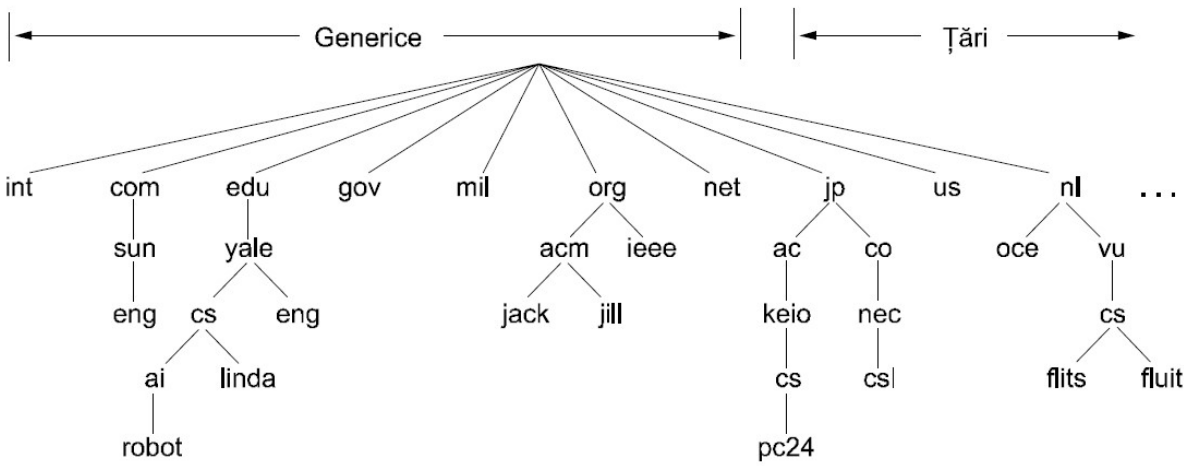

Figura 22. O porțiune a spațiului numelor de domenii din Internet

Fiecărui domeniu, fie că este un singur calculator gazdă, fie un domeniu de nivel superior, îi poate fi asociată o mulțime de înregistrări de resurse (*resource records*). Pentru un singur sistem gazdă, cea mai obișnuită înregistrare de resursă este chiar adresa IP, dar există multe alte tipuri de înregistrări de resurse. Atunci când un *resolver* trimite un nume de domeniu către un DNS, ceea ce va primi ca răspuns sunt înregistrările de resurse asociate acelui nume. Astfel, adevărata funcție a DNS este să realizeze corespondența dintre numele de domenii şi înregistrările de resurse.

O înregistrare de resursă este formată din cinci părți componente. Cu toate că, din rațiuni de eficientă, înregistrările de resurse sunt codificate binar, în majoritatea expunerilor ele sunt prezentate ca text ASCII, câte o înregistrare de resursă pe linie. Formatul pe care îl vom utiliza este următorul: **Nume domeniu Timp de viață Clasă Tip Valoare**

Nume domeniu (*domain_name*) precizează domeniul căruia i se aplică această înregistrare. În mod normal există mai multe înregistrări pentru fiecare domeniu şi fiecare copie a bazei de date păstrează informații despre mai multe domenii. Acest câmp este utilizat ca cheie de căutare primară pentru a satisface cererile. Ordinea înregistrărilor în baza de date nu este semnificativă.

Câmpul **Timp de viață** (*time_to_live*) dă o indicație despre cât de stabilă este înregistrarea. Informația care este foarte stabilă are asigurată o valoare mare, cum ar fi 86400 (numărul de secunde dintr-o zi). Informației instabile îi este atribuită o valoare mică, cum ar fi 60 (1 minut).

Al treilea câmp dintr-o înregistrare de resursă este **Clasa** (*class*). Pentru informațiile legate de Internet este tot timpul IN. Pentru alte informații pot fi folosite alte coduri, însă în practică acestea se întâlnesc rar.

Câmpul **Tip** (*type*) precizează tipul înregistrării. Cele mai importante tipuri sunt prezentate în Figura 21.

Câmpul **Valoare** (*value*) poate fi un număr, un nume de domeniu sau un şir ASCII. Semantica depinde de tipul de înregistrare.

Tip	Semnificație	Valoare
SOA	Start autoritate	Parametrii pentru această zonă
A	Adresa IP a unui sistem gazdă	Întreg pe 32 de biți
MX	Schimb de poștă	Prioritate, domeniu dispus să accepte poștă electronică
NS	Server de Nume	Numele serverului pentru acest domeniu
CNAME	Nume canonic	Numele domeniului
PTR	Pointer	Pseudonim pentru adresa IP
HINFO	Descriere sistem gazdă	Unitate centrală și sistem de operare în ASCII
TXT	Text	Text ASCII neinterpretat

Figura 23. Principalele tipuri de înregistrări de resurse DNS

O înregistrare SOA furnizează numele sursei primare de informații despre zona serverului de nume (descrisă mai jos), adresa de e-mail a administratorului, un identificator unic și diverși indicatori și contoare de timp.

Cel mai important tip de înregistrare este înregistrarea A (adresă). Ea păstrează adresa IP de 32 de biți a unui sistem gazdă.

Următoarea ca importanță este înregistrarea MX. Aceasta precizează numele sistemului gazdă pregătit să accepte poșta electronică pentru domeniul specificat.

Înregistrările NS specifică serverele de nume. De exemplu, fiecare bază de date DNS are în mod normal o înregistrare NS pentru fiecare domeniu.

Înregistrările CNAME permit crearea pseudonimelor.

Înregistrările HINFO permit aflarea tipului de mașină și de sistem de operare cărora le corespunde domeniul.

Un exemplu de informație ce se poate găsi în baza de date DNS a unui domeniu este prezentat în Figura 24.

```
;Baza de date pentru cs.vu.nl
cs.vu.nl.       86400   IN   SOA      star boss (9527, 7200, 7200, 241920, 86400)
cs.vu.nl.       86400   IN   TXT      „Divisie Wiskunde en Informatica."
cs.vu.nl.       86400   IN   TXT      „Vrije Universiteit Amsterdam."
cs.vu.nl.       86400   IN   MX       1 zephyr.cs.vu.nl.
cs.vu.nl.       86400   IN   MX       2 top.cs.vu.nl.
flits.cs.vu.nl. 86400   IN   HINFO    Sun Unix
flits.cs.vu.nl. 86400   IN   A        130.37.16.112
flits.cs.vu.nl. 86400   IN   A        192.31.231.165
flits.cs.vu.nl. 86400   IN   MX       1 flits.cs.vu.nl.
flits.cs.vu.nl. 86400   IN   MX       2 zephyr.cs.vu.nl.
flits.cs.vu.nl. 86400   IN   MX       3 top.cs.vu.nl.
www.cs.vu.nl.   86400   IN   CNAME    star.cs.vu.nl.
ftp.cs.vu.nl.   86400   IN   CNAME    zephyr.cs.vu.nl.
rowboat                 IN   A        130.37.56.201
                        IN   MX       1 rowboat
                        IN   MX       2 zephyr
                        IN   HINFO    Sun Unix

little-sister           IN   A        130.37.62.23
                        IN   HINFO    Mac MacOS

laserjet                IN   A        192.31.231.216
                        IN   HINFO    „HP Laserjet IIISi" Proprietary
```

Figura 24. O parte dintr-o posibilă bază de date DNS pentru cs.vu.nl

Teoretic, un singur server de nume poate conține întreaga bază de date DNS și poate să răspundă tuturor cererilor. În practică, acest server poate fi atât de încărcat, încât să devină de neutilizat. În afară de aceasta, dacă se defectează, va fi afectat întregul Internet. Pentru a evita problemele asociate cu existența unei singure surse de informație, spațiul de nume DNS este împărțit în zone care nu se suprapun. Fiecare zonă conține câte o parte a arborelui precum și numele serverelor care păstrează informația autorizată despre acea zonă.

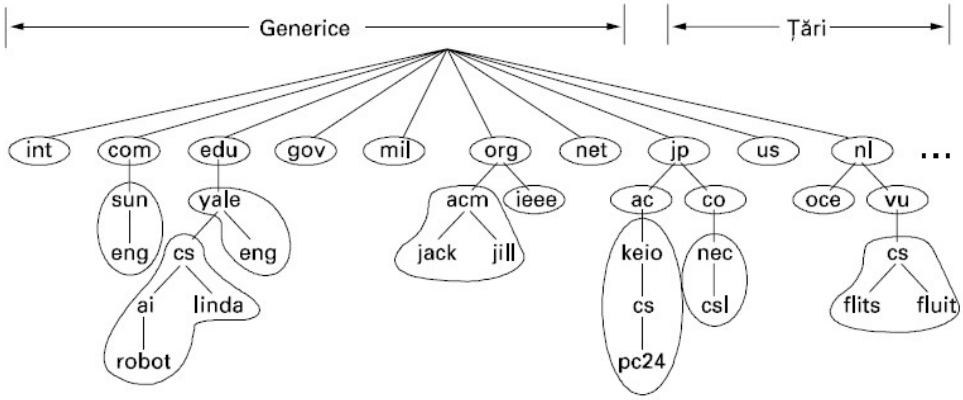

Figura 25. O parte din spațiul numelor DNS prezentând împărțirea în zone

Atunci când un *resolver* are o cerere referitoare la un nume de domeniu, el transferă cererea unuia din serverele locale de nume. Dacă domeniul căutat este sub jurisdicția serverului de nume, el reîntoarce înregistrări de resurse autorizate. O înregistrare autorizată (*authoritative record*) este cea care vine de la autoritatea care administrează înregistrarea și astfel este întotdeauna corectă.

Dacă, totuși, domeniul se află la distanță, iar local nu este disponibilă nici o informație despre domeniul cerut, atunci serverul de nume trimite un mesaj de cerere la serverul de nume de pe primul nivel al domeniului solicitat. Odată ce aceste înregistrări de resurse ajung înapoi la serverul de nume care le-a cerut, ele vor fi depuse în memoria ascunsă a acestuia, pentru a fi folosite ulterior. Totuși, această informație nu este autorizată, deoarece orice schimbare făcută la acea înregistrare nu se va propaga spre toate serverele care au folosit-o. Din acest motiv intrările în memoria ascunsă nu ar trebui să aibă viață prea lungă. Acesta este motivul pentru care câmpul Timp_de_viaţă este inclus în fiecare înregistrare de resursă.

2.2 Web și HTTP

Din punctul de vedere al utilizatorului, Web-ul constă dintr-o colecție imensă de documente sau pagini de Web (*Web pages*), adesea numite prescurtat pagini, răspândite în toată lumea. Fiecare pagină poate să conțină legături către alte pagini, aflate oriunde în lume.

Paginile pot fi vizualizate cu ajutorul unui program de navigare (*browser*). Programul de navigare aduce pagina cerută, interpretează textul și comenzile de formatare conținute în text și afișează pagina, formatată corespunzător, pe ecran.

Adresarea unei pagini se face prin nume folosind URL-uri (Localizatoare Uniforme de Resurse). Un URL tipic este http://www.itu.org/home/index.html și are trei părți: numele

protocolului (*http*), numele calculatorului pe care se găsește pagina (*www.itu.com*) și calea către fișierul care conține pagina (*home/index.html*).

Etapele parcurse pentru obținerea unei pagini Web sunt următoarele:

- Programul de navigare determină URL (pe baza selecției).
- Programul de navigare întreabă DNS care este adresa IP pentru www.itu.org.
- DNS răspunde cu 156.106.192.32.
- Programul de navigare realizează conexiunea TCP cu portul 80 al 156.106.192.32.
- Trimite o cerere pentru fișierul /home/index.html.
- Serverul www.itu.org transmite fișierul /home/index.html.
- Conexiunea TCP este eliberată.
- Programul de navigare afișează textul din /home/index.html.
- Programul de navigare aduce și afișează toate imaginile din acest fișier.

HTTP este un protocol elaborat pentru transferul dinspre server spre client a fișierelor cu informații disponibile public. Acesta specifică ce mesaje pot trimite clienții către servere și ce răspunsuri primesc înapoi. Deși numele protocolului face referire la hipertext, el poate utilizat pentru a transfera orice fel de conținut.

Protocolul de bază constă în trimiterea de către client a unei cereri, în care informația principală este numele fișierului cerut. Răspunsul serverului conține niște informații despre fișier și conținutul efectiv al fișierului. Implicit, conexiunea se închide după transferul unui fișier. Dacă clientul dorește mai multe fișiere de pe același server, va trebui să deschidă câte o conexiune pentru fiecare fișier.

Modul uzual prin care un program de navigare contactează un server este de a stabili o conexiune TCP pe portul 80 pe mașina serverului. Avantajul de a folosi TCP este că nici programele de navigare și nici serverele nu trebuie sa se preocupe de mesajele pierdute, mesajele duplicate, mesajele lungi, sau mesajele de confirmare. Toate aceste aspecte sunt tratate de implementarea TCP.

Cu toate că HTTP a fost proiectat pentru utilizarea în Web, el a fost creat intenționat mai general decât era necesar în perspectiva aplicațiilor orientate pe obiecte. Pentru aceasta sunt suportate operațiile, denumite **metode**, care fac mai mult decât a cere o pagină Web.

Fiecare cerere constă din una sau mai multe linii de text ASCII, în care primul cuvânt din prima linie este numele metodei cerute. Metodele incorporate sunt listate în Figura 26.

Metoda	Descriere
GET	Cerere de citire a unei pagini Web
HEAD	Cerere de citire a antetului unei pagini de Web
PUT	Cerere de memorare a unei pagini de Web
POST	Adăugarea la o resursă anume (de exemplu o pagină de Web)
DELETE	Ștergerea unei pagini de Web
TRACE	Tipărirea cererii care a sosit
CONNECT	Rezervat pentru o utilizare în viitor
OPTIONS	Interogarea anumitor opțiuni

Figura 26. Metode de cerere standard pentru HTTP

Metoda GET cere serverului să trimită pagina (prin care noi înțelegem obiect, în cel mai general caz, dar în practică de obicei doar un fișier). Forma uzuală a metodei GET este *GET fişier HTTP-1.1*, unde fișier denumește resursa (fișierul) ce va fi adusă.

Metoda HEAD cere doar antetul mesajului, fără să ceară și pagina propriu-zisă. Această metodă poate să fie utilizată pentru a afla când s-a făcut ultima modificare, pentru a obține informații pentru indexare, sau numai pentru a verifica corectitudinea unui URL.

Metoda PUT este inversa metodei GET: în loc să citească o pagină, o scrie. Această metodă permite crearea unei colecții de pagini de Web pe un server la distanta. Corpul cererii conține pagina.

Similară metodei PUT este metoda POST. Și ea conține un URL, dar în loc să înlocuiască date existente, noile date se vor adăuga într-un mod generalizat. De exemplu, se poate transmite un mesaj la un grup de știri.

DELETE realizează ștergerea unei pagini. Ca și la PUT, autentificarea și drepturile de acces joacă aici un rol important.

Metoda TRACE este pentru verificarea corectitudinii. Ea cere serverului să trimită înapoi cererea. Această metodă este utilă când cererile nu sunt procesate corect și clientul vrea să știe ce fel de cerere a ajuns de fapt la server.

Metoda CONNECT nu este utilizată în prezent. Este rezervată pentru utilizări ulterioare.

Metoda OPTIONS asigură o modalitate pentru client de a interoga serverul despre proprietățile acestuia sau despre cele ale unui anumit fișier.

Fiecare cerere obține un răspuns ce constă din linia de stare și posibile informații suplimentare (de exemplu, o parte sau toată pagina Web). Linia de stare conține un cod de stare de trei cifre, indicând dacă cererea a fost satisfăcută și dacă nu, cauza. Prima cifră este utilizată pentru împărțirea răspunsurilor în cinci mari grupuri, ca în Figura 27.

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

Figura 27. Metode de cerere standard pentru HTTP

2.3 E-mail – Poșta electronică

Noțiunea de e-mail are semnificația de scrisoare electronică, iar sistemul în sine (care se ocupă cu transmiterea, preluarea și interpretarea conținutului mesajelor electronice) se numește sistem de poștă electronică.

În general, sistemele de poștă electronică pun la dispoziție cinci funcții de bază:

- Compunerea se referă la procesul de creare a mesajelor și a răspunsurilor. Deși pentru corpul mesajului poate fi folosit orice editor de texte, sistemul însuși poate acorda asistență la adresare și la completarea numeroaselor câmpuri antet

atașate fiecărui mesaj. De exemplu, când se răspunde la un mesaj, sistemul poate extrage adresa inițiatorului din mesajul primit și o poate insera automat în locul potrivit din cadrul răspunsului.

- Transferul se referă la deplasarea mesajului de la autor la receptor. În mare, aceasta necesită stabilirea unei conexiuni la destinație, sau la o mașină intermediară, emiterea mesajului și eliberarea conexiunii. Sistemul de poștă ar trebui să facă acest lucru singur, fără a deranja utilizatorul.
- Raportarea se referă la informarea inițiatorului despre ce s-a întâmplat cu mesajul.
- Afișarea mesajelor primite este necesară pentru ca utilizatorii să-și poată citi poșta. Uneori sunt necesare conversii sau trebuie apelat un program de vizualizare special.
- Dispoziția este pasul final și se referă la ceea ce face receptorul cu mesajul, după ce l-a primit. Posibilitățile includ eliminarea sa înainte de a-l citi, aruncarea sa după citire, salvarea sa ș.a.m.d. Ar trebui de asemenea să fie posibilă regăsirea și recitirea de mesaje deja salvate, trimiterea lor mai departe, sau procesarea lor în alte moduri.

2.3.1 Transmiterea mesajelor

Protocolul folosit pentru a trimite un mesaj de pe calculatorul unui client către un server destinație (fie cel final, al destinatarului, fie unul intermediar) se numește SMTP. Portul TCP standard pentru protocolul SMTP este 25. Sarcina acestui protocol este de a permite transferul mesajelor într-un mod eficient, și este un sistem independent care necesită stabilirea unui canal de comunicație duplex între cele două calculatoare care participă la schimbul de mesaje.

Protocolul SMTP definește un limbaj de comunicare între echipamentul care transmite (client) și echipamentul care primește mesajul electronic (server). Comunicația între echipamentul client și echipamentul server se efectuează în modul următor: clientul trimite o comanda server-ului, acesta o execută și o returnează clientului un cod numeric.

Comenzile SMTP (o combinație de prescurtări de cuvinte specifice din limba engleză) constau din codul comenzii format din patru litere și urmat opțional de un parametru. Principalele comenzi definite de protocolul SMTP sunt:

- HELO <hostname> - reprezintă comanda care inițializează dialogul dintre procesul client și procesul server; procesul client va identifica serverul cu numele calculatorului pe care rulează, specificat prin parametrul <hostname>.
- MAIL FROM: <expeditor> - informează procesul server că urmează să primească un e-mail de la expeditor.
- RCPT TO: <destinatar> - specifică procesului server adresa destinatarului (prin parametrul <destinatar>) căruia îi este adresat mesajul e-mail care urmează a fi transmis.

- DATA – specifică procesului server că urmează să primească de la client conținutul unui mesaj electronic (e-mail).
- QUIT - închide canalul de comunicație dintre client și server.

Pentru fiecare comandă trimisă de către clientul SMTP către serverul SMTP, acesta din urmă returnează un cod numeric care reprezintă codul rezultat în urma execuției operației specificate de către client. Principalele coduri numerice (și semnificațiile lor) returnate de procesul server sunt:

- 220 – *Service ready*, procesul server este disponibil pentru a prelua un mesaj.
- 221 – *Service closing transmission channel*, procesul server urmează a închide canalul de comunicaţie cu procesul client.
- 250 – *Request mail action okay*, *completed*, specifică procesului client că operația specificată de acesta a fost executată cu succes.
- 251 – *User not local*, informează procesul client că nu cunoaște adresa destinatarului și va redirecționa mesajul respectiv către un alt calculator server.
- 354 – *Start mail input*, specifică procesului client că acesta poate începe transmisia conținutului mesajului (e-mail-ului);
- 502 – *Command not implemented*, cod de eroare returnat atunci când comanda specificată de către procesul client nu este cunoscută / implementată de către procesul server.

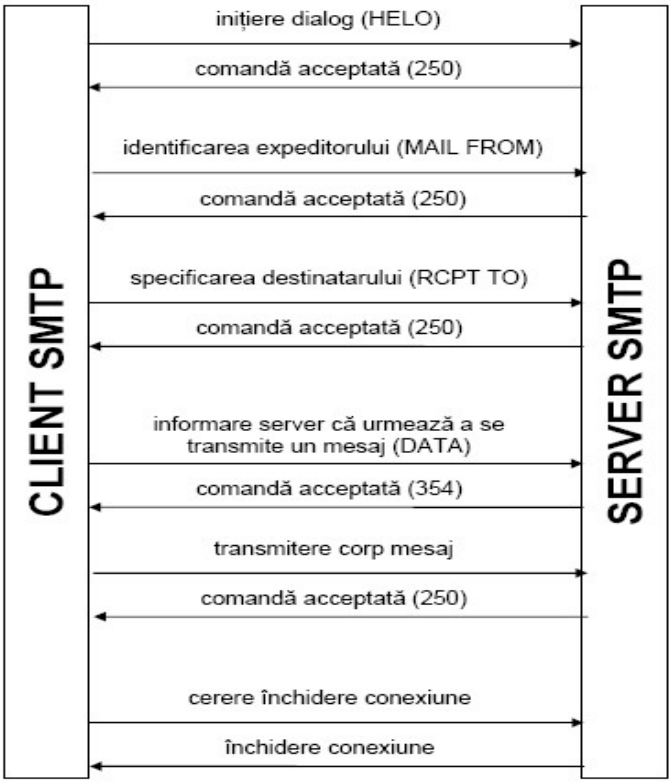

Figura 28. Scenariu de transmitere a unui mesaj

2.3.2 Recepționarea mesajelor

2.3.2.1 Protocolul POP3

Etapa de recepționare a unui e-mail presupune că utilizatorul căruia îi este destinat mesajul să pornească aplicația client pentru serviciul de poștă electronică și să îi specifice acesteia să extragă de pe calculatorul server (care are rolul de oficiu poștal) noile mesaje asociate căsuței sale poștale.

Protocolul utilizat pentru extragerea mesajelor unui utilizator de pe un calculator server care îi gestionează căsuța poștală se numește POP3 (Post Office Protocol Version 3). Portul TCP standard pentru protocolul POP3 este 110.

Rolul acestui protocol este de a permite utilizatorilor să își aducă mesajele de pe calculatorul server (care are rolul de oficiu poștal) pe propriul calculator.

Protocolul POP3 definește un limbaj de comunicare între procesul care cere informațiile (client) și procesul care executa comenzile și transmite mesajele cerute de către client (server).

Principalele facilități oferite de către acest protocol sunt:

- extragerea mesajelor de pe calculatorul server;
- ștergerea mesajelor (care au fost sau nu recepționate) de pe calculatorul server;
- posibilitatea utilizării versiunii securizate, POPS3, care criptează informațiile transmise între procesul client și procesul server, pentru a preveni astfel interceptarea acestora. Comunicația între procesul client și procesul server se efectuează în modul următor: clientul trimite o comanda serverului, acesta o execută și returnează clientului un cod numeric.

Principalele comenzi definite de protocolul POP3 sunt:

- USER <utilizator> - specifică procesului server numele utilizatorului pentru care să deschidă căsuța poștală.
- PASS <parola> - trimite procesului server parola contului de utilizator asociată cu contul de utilizator specificat la comanda precedentă.
- LIST [<număr_mesaj>] - cere procesului server să listeze mesajele utilizatorului.
- RETR <număr_mesaj> - cere procesului server să listeze conținutul mesajului cu numărul de identificare specificat de parametrul <număr_mesaj>.
- DELE <număr_mesaj> - șterge mesajul cu numărul specificat de parametrul <număr_mesaj>.
- QUIT - închide canalul de comunicație dintre client și server.
- STAT - cere procesului server să afișeze informații statistice despre căsuța poștală a utilizatorului curent (și numărul de mesaje din căsuța poștală și dimensiunea totală a acestora).
- LAST - cere procesului server să afișeze numărul de identificare al ultimului mesaj venit în căsuța poștală.

- TOP <număr_mesaj> <număr_linii> - specifică procesului server să listeze din mesajul cu numărul de identificare specificat de parametrul <număr_mesaj> primele <număr_linii> de conţinut;
- RSET - resetează starea mesajelor din căsuța poștală (refăcând mesajele șterse).

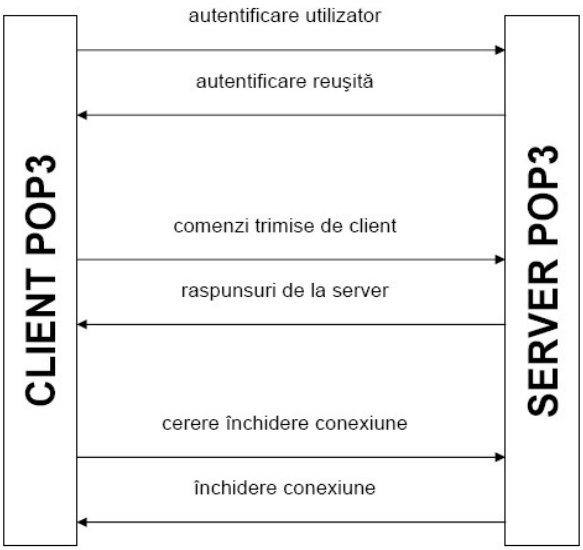

Figura 29. Procesul de comunicare între server și client utilizând protocolul POP3

2.3.2.2 Protocolul IMAP

IMAP este un protocol care a fost proiectat pentru a ajuta utilizatorii care accesează căsuța poștală de pe mai multe calculatoare. Spre deosebire de POP3, care în mod normal presupune că utilizatorul își va goli căsuța poștală la fiecare conectare și va lucra deconectat de la rețea (*off-line*) după aceea, IMAP presupune că tot conținutul căsuței va rămâne pe server putând fi accesat de pe oricâte calculatoare. Portul TCP standard pentru protocolul IMAP este 143.

Protocolul asigură mecanisme pentru crearea, ștergerea și manipularea mai multor directoare pe server. Astfel, un utilizator poate păstra un director pentru fiecare corespondent și poate muta aici mesajele din *Inbox* după ce acestea au fost citite.

Stilul general al protocolului IMAP este similar cu cel al POP3-ului cu excepția faptului că există zeci de comenzi. La fel și scenariul de conectare a clientului cu serverul.

2.4 FTP – Protocolul pentru transfer de fișiere

FTP este protocolul care oferă facilități pentru transferul fișierelor pe sau de pe un calculator din rețea. Transferul poate fi de două tipuri:

- *Upload* - fișierele sunt transferate de pe calculatorul local pe cel de la distanță.
- *Download* - fișierele sunt transferate de pe calculatorul aflat la distanță pe cel local.

Pentru a se realiza transferul fișierelor este necesar să existe:

- Aplicație server – care este instalată pe un calculator care astfel devine server FTP. Prin FTP server administratorul de sistem creează conturi FTP și stabilește în ce zonă se poate conecta clientul și ce poate face în acea zonă.
- Aplicație client - care este instalată pe un alt calculator care astfel devine client FTP.

Clientul deschide o conexiune TCP către portul 21 al serverului; această conexiune se numește conexiune de control. Prin conexiunea de control, clientul transmite comenzi în format text, câte o comandă pe o linie. Fiecare comandă începe cu numele comenzii urmat de eventuali parametrii. Parametrii sunt separați prin spații, atât față de numele comenzii cât și între ei. Serverul răspunde tot în format text, fiecare răspuns începând cu un cod care arată dacă comanda s-a executat cu succes sau ce erori s-au produs, după care urmează un text ce descrie, în limbaj natural, rezultatul execuției comenzii. Cu o singură excepție (în cazul comenzii PASV, descrisă mai jos), textul din răspuns nu este interpretat de către aplicația client. El este însă afișat, de obicei, pe ecran utilizatorului aplicației client.

Autentificarea se face la solicitarea clientului. Clientul trimite succesiv două comenzi, USER și PASS, având ca parametrii respectiv numele utilizatorului și parola acestuia. Serverul refuză execuția majorității comenzilor clientului înainte de autentificarea cu succes a acestuia. După autentificare, serverul acceptă să efectueze operațiile cerute de client doar dacă utilizatorul în contul căruia s-a făcut autentificarea are dreptul la operațiile respective.

Pentru transferul de fișiere publice, serverul este configurat să accepte autentificare cu numele de utilizator ftp sau *anonymous* fără să solicite parolă sau acceptând orice șir de caractere pe post de parolă. în vremurile de început ale Internet-ului, se obișnuia ca un utilizator ce dorea acces la fișiere publice să-și dea, pe post de parolă, adresa sa de poștă electronică. O dată cu răspândirea spam-urilor, s-a renunțat la acest obicei.

Transferul fișierelor se cere prin comenzile SEND (dinspre client spre server) și RETR (dinspre server spre client). Comenzile au ca argument numele de pe server al fișierului de transferat. Transferul propriu-zis are loc printr-o conexiune separată, numită conexiune de date. Pentru fiecare fișier se deschide o nouă conexiune de date, care se închide la finalul transferului fișierului. Dimensiunea fișierului nu este specificată explicit nicăieri, receptorul fișierului obținând lungimea din faptul că emițătorul închide conexiunea de date la finalul fișierului.

Există două moduri de deschidere a conexiunii de date:

- Modul activ prevede că serverul deschide conexiunea de date ca o conexiune TCP dinspre portul 20 al serverului către un port specificat de client. Clientul specifică portul pe care așteaptă conexiunea prin comanda PORT. Conexiunea se deschide ca urmare a comenzii de transfer (SEND sau RETR), nu imediat după primirea comenzii PORT.
- Modul pasiv prevede deschiderea conexiunii de date de către client, dinspre un port oarecare al său, către un port specificat de server. Portul specificat de server se obține ca răspuns al comenzii PASV date de client. Acesta este singurul caz

în care clientul interpretează din răspunsul serverului și altceva decât codul returnat.

Listarea fișierelor de pe server este solicitată de client prin comanda LIST. Transferul listei de fișiere se face tot printr-o conexiune de date, ca și în cazul comenzii RETR.

Capitolul 3. Nivelul Transport

Nivelul Transport al modelului TCP/IP administrează transmisia de date de la un calculator la altul, asigurând calitatea serviciului de comunicare, siguranța liniei de transport, controlul fluxului, detecția și corecția erorilor. În lumea reală el îndeplinește importanta funcție de a izola nivelurile superioare de tehnologia, arhitectura și imperfecțiunile subrețelei.

Acest nivel oferă două tipuri de servicii: **orientate pe conexiune** (protocolul TCP) sau **datagramă** (protocolul UDP), împărțind datele în segmente mai mici (Figura 30) pentru a fi transportate ușor prin rețea. El este proiectat astfel încât să permită conversații între gazdele sursă, respectiv, destinație. Hardware-ul și/sau software-ul care se ocupă de toate acestea în cadrul nivelului transport poartă numele de **entitate de transport**. Entitatea de transport poate aparține nucleului sistemului de operare, unui proces distinct, unei biblioteci legate de aplicațiile de rețea sau poate fi găsită în cadrul plăcii de rețea.

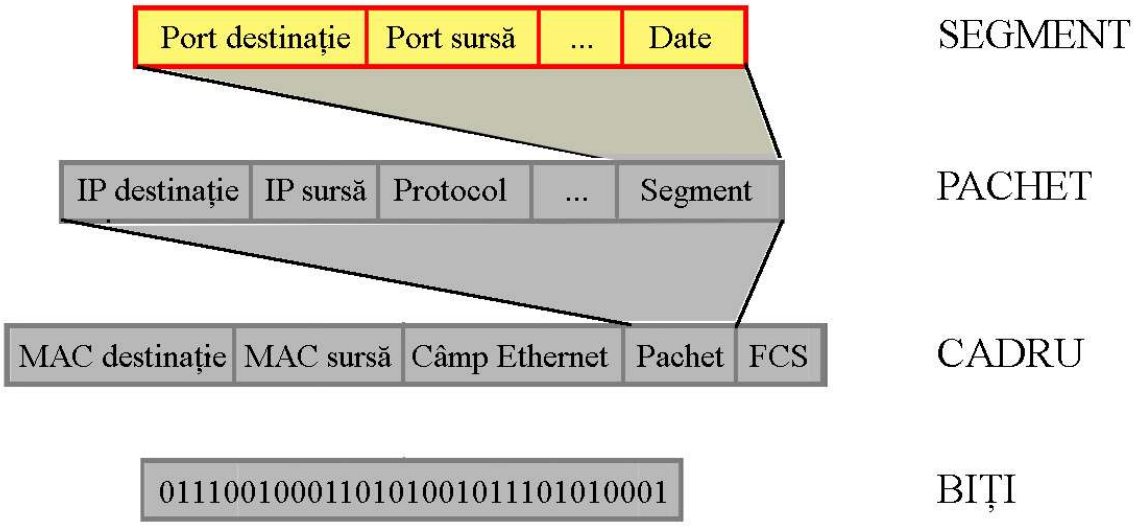

Figura 30. Unitatea de date pentru Nivelul Transport

Deoarece utilizatorii nu pot controla nivelurile inferioare, ei nu pot rezolva problema unor servicii de proastă calitate folosind rutere mai bune sau adăugând o tratare a erorilor mai sofisticată. Singura posibilitate este ca acest nivel să amelioreze calitatea serviciilor. Dacă pe o subrețea orientată pe conexiune, o entitate de transport este informată la jumătatea transmisiei că a fost închisă brusc conexiunea sa la un nivel inferior, fără nici o indicație despre ceea ce s-a întâmplat cu datele aflate în acel moment în tranzit, ea poate iniția o altă conexiune cu entitatea de transport aflată la distanță. Folosind această nouă conexiune, ea poate întreba care date au ajuns la destinație și care nu, și poate continua comunicarea din locul de unde a fost întreruptă.

3.1 Primitive ale serviciilor de transport

Pentru a permite utilizatorului să acceseze serviciile de transport, nivelul transport trebuie să ofere unele operații programelor aplicație, adică o interfață a serviciului transport. Astfel, multe programe (și programatori) folosesc primitivele de transport. Acestea sunt în număr de cinci (Figura 31), interfața fiind într-adevăr simplă, dar prezintă trăsăturile de bază

ale oricărei interfețe orientate pe conexiune a nivelului transport. Ea permite programelor de aplicație să stabilească, să utilizeze și să elibereze conexiuni, ceea ce este suficient pentru multe aplicații.

Primitiva	Unitatea de date trimisă	Explicaţii
LISTEN	(nimic)	Se blochează până când un proces încearcă să se conecteze
CONNECT	CONNECTION REQ.	Încearcă să stabilească conexiunea
SEND	DATE	Transmite informaţie
RECEIVE	(nimic)	Se blochează până când primeşte date trimise
DISCONNECT	DISCONNECTION REQ.	Trimisă de partea care vrea să se deconecteze

Figura 31. Primitivele unui serviciu de transport simplu

3.2 Adresarea

Atunci când un proces ce rulează pe o gazdă dorește să stabilească o conexiune cu un proces ce rulează pe o altă gazdă aflată la distanță, el trebuie să specifice cu care proces dorește să se conecteze. Metoda folosită în mod normal este de a defini adrese de transport la care procesele pot să aștepte cereri de conexiune. În Internet acestea se numesc porturi. În continuare se va folosi pentru acestea termenul generic SAP (Punct de Access la Servicii), iar pentru nivelul Transport se vor numi TSAP - punct de acces la serviciul de transport. Punctele similare în cazul nivelului rețea (adică adresele la nivel rețea) sunt numite NSAP. Adresele IP sunt exemple de NSAP-uri.

Necesitatea de a avea mai multe TSAP-uri se datorează faptului că, de obicei, fiecare calculator gazdă are un singur NSAP, deci cumva este nevoie să se distingă mai multe adrese de transport pentru a putea utiliza multiple aplicații ce rulează pe același calculator.

Un scenariu posibil pentru stabilirea unei conexiuni la nivel transport este următorul:

1. Un proces server care furnizează ora exactă și care rulează pe gazda 2 se atașează la TSAP 1522 propriu, așteptând un apel. Poate fi utilizat un apel de tip LISTEN.
2. Un proces aplicație de pe gazda 1 dorește să afle ora exactă; atunci el generează un apel CONNECT specificând TSAP 1208 ca sursă și TSAP 1522 ca destinație. Această acțiune are ca rezultat în cele din urmă stabilirea unei conexiuni la nivel Transport între procesele aplicație de pe gazda 1 și serverul de pe gazda 2.
3. Procesul aplicație trimite o cerere o cerere pentru timp.
4. Procesul server de timp răspunde cu timpul curent.
5. Conexiunea transport este apoi eliberată.

3.3 Protocolul UDP

UDP (Protocol cu Datagrame Utilizator) este un protocol de transport fără conexiune ce oferă aplicațiilor o modalitate de a trimite datagrame IP încapsulate fără a fi nevoie să stabilească o conexiune. UDP este descris în RFC 768 [13]. Este adesea folosit pentru interogări rapide întrebare-răspuns, client-server și pentru aplicații în care comunicarea promptă este mai

importantă decât comunicarea cu acuratețe, așa cum sunt aplicațiile de transmisie a vocii și a imaginilor video.

UDP transmite segmente constând într-un antet de 8 octeți urmat de informația utilă. Antetul este prezentat în Figura 32. Cele două porturi servesc la identificarea punctelor terminale ale calculatoarelor sursă și destinație. Când ajunge un pachet UDP, conținutul său este predat procesului atașat portului destinație.

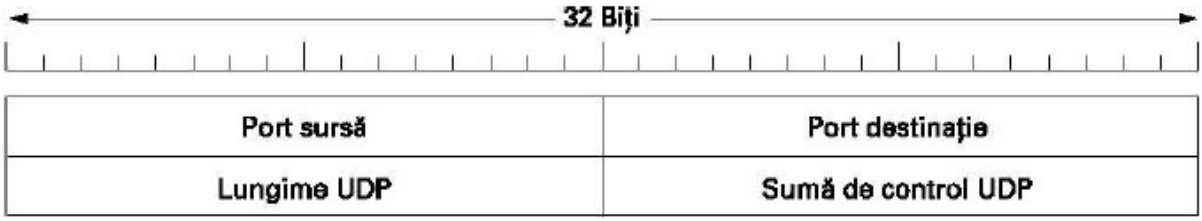

Figura 32. Antetul UDP

Portul sursă este în primul rând necesar atunci când un răspuns trebuie transmis înapoi la sursă. Prin copierea câmpului port sursă din segmentul care sosește în câmpul port destinație al segmentului care pleacă, procesul ce trimite răspunsul specifică ce proces de pe calculatorul de trimitere urmează să-l primească.

Protocolul UDP nu realizează controlul fluxului, controlul erorii sau retransmiterea unui segment incorect primit. Toate acestea depind de procesele utilizatorului.

Avantajul acestui protocol se observă în situația utilizării aplicațiilor de tip client-server. Deseori, clientul trimite o cerință scurtă server-ului și așteaptă înapoi un răspuns scurt. Dacă se pierde ori cererea ori răspunsul, clientul poate pur și simplu să încerce din nou după ce a expirat timpul. Nu numai că va fi mai simplu codul, dar sunt necesare și mai puține mesaje (câte unul în fiecare direcție) decât la un protocol care solicită o inițializare inițială.

O aplicație care folosește protocolul UDP este DNS. Pe scurt, un program care trebuie să caute adresele de IP ale unor nume de domenii, poate trimite un pachet UDP conținând numele gazdei către un server DNS. Serverul răspunde cu un pachet UDP conținând adresa de IP a gazdei. Nu este necesară nici o inițializare în avans și nici o închidere de sesiune. Doar două mesaje traversează rețeaua.

3.4 Protocolul TCP

TCP (Protocolul de Control al Transmisiei) este un protocol orientat pe conexiune care permite ca un segment trimis de la un calculator să ajungă fără erori pe orice alt calculator din Internet. Dacă pe calculatorul destinație un segment ajunge cu erori, TCP cere retrimiterea lui. Orientarea pe conexiune nu semnifică faptul că există un circuit între calculatoarele care comunică, ci faptul că segmentele călătoresc bidirecțional între două gazde care sunt conectate logic pentru o anumită perioadă.

Internetul este compus din nenumărate rețele ce diferă ca topologie, lărgime de bandă, întârzieri, dimensiunea pachetelor și alți parametri. TCP a fost proiectat să se adapteze în mod dinamic la proprietățile acestuia și să fie robust în ceea ce privește mai multe tipuri de defecte. TCP a fost definit în mod oficial în RFC 793 [14]. O dată cu trecerea timpului, au fost detectate

diverse erori și inconsistențe și au fost modificate cerințele în anumite subdomenii. Aceste clarificări, precum și corectarea câtorva erori sunt detaliate în RFC 1122. Extensiile sunt furnizate în RFC 1323.

Fiecare mașină care suportă TCP dispune de o entitate de transport, fie ca proces utilizator, fie ca procedură de bibliotecă, fie ca parte a nucleului sistemului de operare. În toate aceste cazuri, ea gestionează fluxurile TCP și interfețele către nivelul inferior, nivelul Internet.

Una din sarcinile protocolului TCP este să detecteze erorile de livrare a segmentelor și să efectueze o retransmisie atunci când situația o impune. Segmentele care ajung (totuși) la destinație pot sosi într-o ordine eronată; este, de asemenea, sarcina TCP-ului să le reasambleze în mesaje respectând ordinea corectă (de secvență).

Serviciul TCP este obținut prin crearea atât de către emițător, cât și de către receptor, a unor puncte finale, *socket*. Fiecare dintre acestea este format din adresa IP a calculatorului gazdă și un port cu lungime de 16 biți, local gazdei respective. Portul este pentru TCP punctul de acces la servicii. Pentru a obține o conexiune TCP, trebuie stabilită explicit o conexiune între un *socket* de pe mașina emițătoare și unul de pe mașina receptoare.

Numerele de port mai mici decât 256 se numesc porturi general cunoscute și sunt rezervate serviciilor standard (Figura 33).

Port	Protocol	Utilitate
21	FTP	Transfer de fişiere
23	Telnet	Login la distanţă
25	SMTP	E-mail
69	TFTP	Protocol de transfer de fişiere trivial
79	Finger	Căutare de informaţii despre un utilizator
80	HTTP	World Wide Web
110	POP-3	Acces prin e-mail la distanţă
119	NNTP	Ştiri USENET

Figura 33. Porturi uzuale

Toate conexiunile TCP sunt duplex integral și punct-la-punct. Duplex integral înseamnă că traficul se poate desfășura în ambele sensuri în același timp. Punct-la-punct indică faptul că fiecare conexiune are exact două puncte finale.

Entitățile TCP de transmisie și de recepție inter-schimbă informație sub formă de segmente. Un segment TCP constă dintr-un antet de exact 20 de octeți (plus o parte opțională) urmat de zero sau mai mulți octeți de date. Programul TCP este cel care decide cât de mari trebuie să fie aceste segmente. El poate acumula informație provenită din mai multe cereri într-un singur segment sau poate fragmenta informația provenind dintr-o singură cerere în mai multe segmente. Există două limite care restricționează dimensiunea unui segment. În primul rând, fiecare segment, inclusiv antetul TCP, trebuie să încapă în cei 65.535 de octeți de informație utilă IP. În al doilea rând, fiecare rețea are o unitate maximă de transfer (MTU), deci fiecare segment trebuie să încapă în acest MTU. În realitate, MTU este în general de 1500 octeți

(dimensiunea informației utile din Ethernet), definind astfel o limită superioară a dimensiunii unui segment.

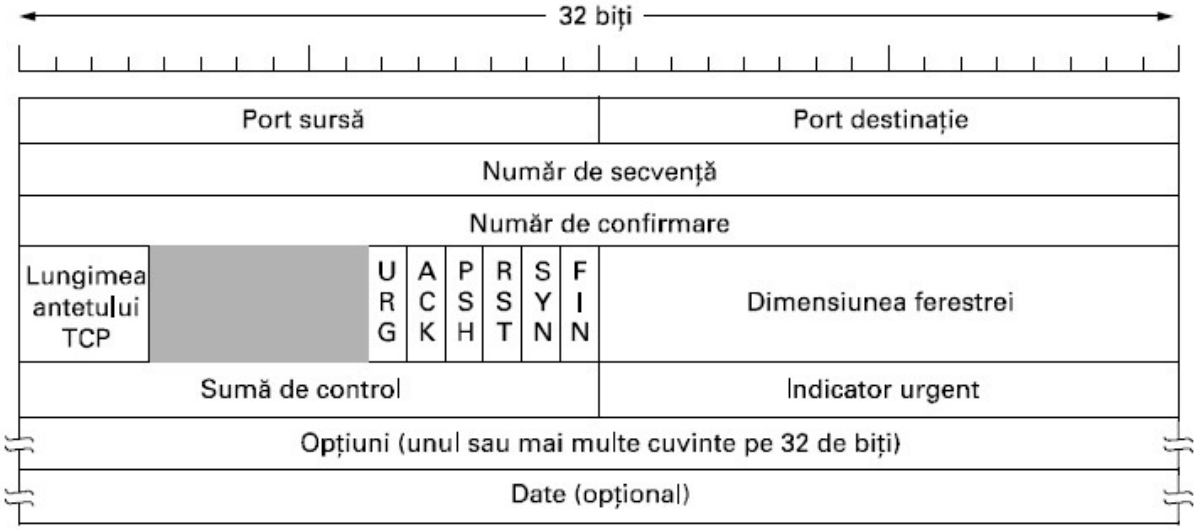

Figura 34. Antetul TCP

Protocolul de bază utilizat de către entitățile TCP este protocolul cu fereastră glisantă (*sliding window*). Atunci când un emițător transmite un segment, el pornește un cronometru. Atunci când un segment ajunge la destinație, entitatea TCP receptoare trimite înapoi un segment (cu informație utilă, dacă aceasta există, sau fără, în caz contrar) care conține totodată și numărul de secvență următor pe care aceasta se așteaptă să-l recepționeze. Dacă cronometrul emițătorului depășește o anumită valoare înaintea primirii confirmării, emițătorul retransmite segmentul neconfirmat.

În TCP conexiunile sunt stabilite utilizând „înțelegerea în trei pași”.

1. Pentru a stabili o conexiune, una din părți – de exemplu serverul - așteaptă în mod pasiv o cerere de conexiune prin execuția primitivelor LISTEN și ACCEPT, putând specifica o sursă anume sau nici o sursă în mod particular.
2. Cealaltă parte – de exemplu clientul - execută o primitivă CONNECT, indicând adresa IP și numărul de port la care dorește să se conecteze, dimensiunea maximă a segmentului TCP pe care este dispusă să o accepte și, opțional, o informație utilizator (de exemplu o parolă).
3. Când un segment sosește la destinație, entitatea TCP receptoare verifică dacă există un proces care a executat LISTEN pe numărul de port specificat în câmpul Port destinație, în caz contrar refuzând conexiunea. Dacă există, segmentul TCP recepționat va fi dirijat către procesul respectiv. Acesta poate accepta sau refuza conexiunea. Dacă o acceptă, trimite înapoi expeditorului un segment de confirmare.

Atunci când încărcarea la care este supusă o rețea este mai mare decât poate aceasta să suporte, apare congestia. Controlul congestiei este realizat de către TCP prin micșorarea ratei de transfer a informației. Atunci când se stabilește o conexiune, trebuie să se aleagă dimensiunea potrivită a segmentelor. Receptorul poate o poate specifica bazându-se pe

dimensiunea buffer-ului propriu. Dacă emițătorul acceptă această dimensiune, nu mai pot apărea probleme datorită umplerii buffer-ului la recepție, dar pot apărea în schimb datorită congestiei interne în rețea (a se observa analogia din Figura 35).

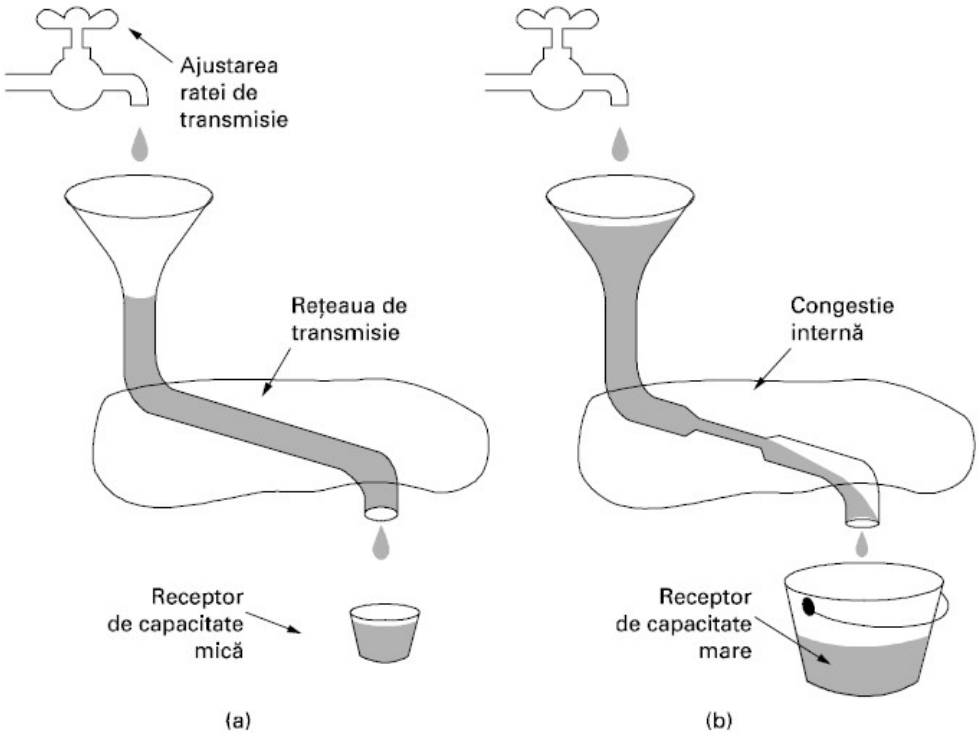

Figura 35. (a) O rețea rapidă alimentând un receptor de capacitate mică. (b) O rețea lentă alimentând un receptor de mare capacitate.

Un exemplu de selecție a cantității de date transmisă este următorul: Dacă receptorul spune: „Trimite 8KB”, dar emițătorul știe că o rafală de mai mult de 4KB poate aglomera excesiv rețeaua, el va trimite 4KB. Din alt punct de vedere, dacă receptorul spune: „Trimite 8KB” și emițătorul știe că o rafală de 32KB poate străbate fără efort rețeaua, el va trimite toți cei 8KB ceruți.

Capitolul 4. Nivelul Internet

Nivelul Internet este un grup de metode, protocoale și specificații de interconectare utilizate pentru a transporta pachete de la gazda sursă la gazda destinație specificată de o adresă IP care este definită în acest scop de Protocolul de Internet (IP).

Scopul interconectării rețelelor este de a permite utilizatorilor din orice rețea să comunice cu utilizatorii celorlalte rețele și de asemenea de a permite unui utilizator din orice rețea să acceseze date pe orice rețea. Realizarea acestui scop înseamnă trimiterea pachetelor dintr-o rețea în alta. Cum rețelele diferă deseori în puncte esențiale, transmiterea acestora nu este întotdeauna ușoară.

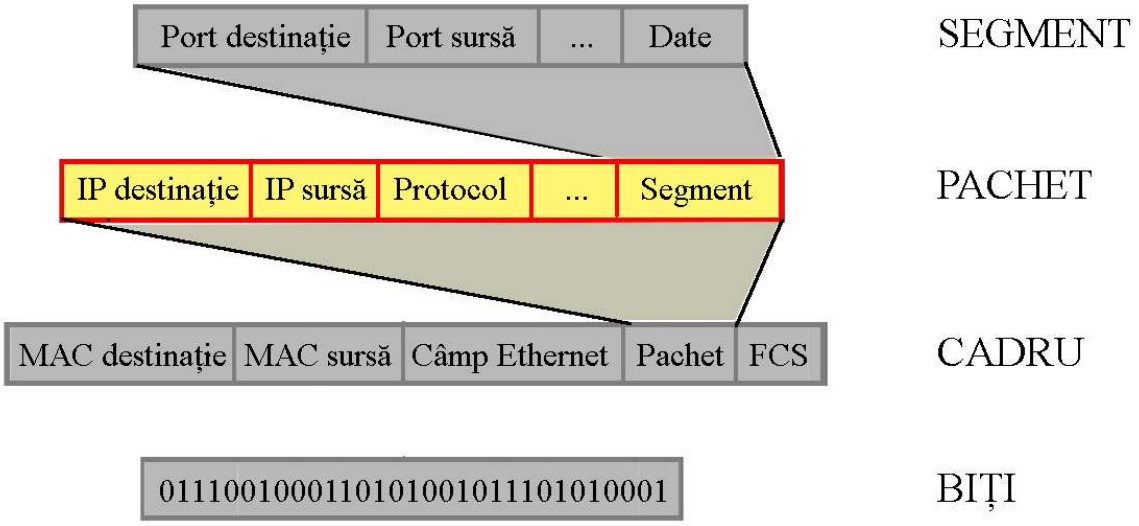

Figura 36. Unitatea de date pentru Nivelul Internet

Rețelele pot fi interconectate prin diferite dispozitive. La nivelul fizic, rețelele pot fi conectate prin repetoare sau noduri (*hub*), care doar transferă biții între două rețele identice. Acestea sunt în marea lor majoritate dispozitive analogice și nu cunosc protocoalele numerice (doar regenerează semnale).

Cu un nivel mai sus întâlnim punțile (*bridge*) și comutatoarele (*switch*), care operează la nivelul legăturii de date. Acestea acceptă cadre, examinează adresele MAC și retransmit cadrele către o rețea diferită, efectuând traduceri de protocol minore, ca de exemplu de la Ethernet la FDDI sau la 802.11.

La nivelul Internet există rutere care pot conecta două rețele. Ruterele comunică între ele prin intermediul unor protocoale de rutare special proiectate, fie protocoale de *gateway* interior, fie protocoale de *gateway* exterior, în funcție de topologia rețelei.

O diferență esențială între cazul utilizării unui *switch* și cel al utilizării unui ruter este următoarea. În cazul *switch*-ului este transportat întregul cadru, pe baza adresei MAC. În cazul unui ruter pachetul este extras din cadru, iar adresa IP din pachet este utilizată pentru a decide unde să fie trimis. Comutatoarele nu trebuie să înțeleagă protocolul nivelului Internet, dar ruterele da.

4.1 Protocolul IP

Protocolul de Internet este responsabil pentru adresarea gazdelor, încapsularea datelor în datagrame (inclusiv fragmentarea și reasamblarea) și rutarea datagramelor de la o gazdă sursă la o gazdă de destinație în una sau mai multe rețele IP. În aceste scopuri, Protocolul de Internet definește formatul pachetelor și oferă un sistem de adresare.

Fiecare datagramă are două componente: un antet și datele care trebuie transportate. Antetul IP include adresa IP sursă, adresa IP de destinație și alte meta-date necesare pentru a direcționa și a livra datagrama. Această metodă de îngloba datele într-un pachet cu un antet se numește încapsulare.

4.1.1 IP v4

Protocolul de Internet versiunea 4 (IPv4) este unul dintre protocoalele de bază ale metodelor de interconectare, bazate pe standarde, din Internet și a fost implementat în 1983.

IPv4 este un protocol fără conexiune pentru utilizarea în rețelele cu comutare de pachete. Funcționează pe baza unui model de livrare cu cel mai bun efort (*best effort delivery*), deoarece nu garantează livrarea și nu asigură secvențierea corespunzătoare sau evitarea dublei livrări. Aceste aspecte, inclusiv integritatea datelor, sunt abordate de către protocolul de transport de nivel superior, cum ar fi TCP.

O datagramă IPv4 constă dintr-o parte de antet și o parte de text. Antetul are o parte fixă de 20 de octeți și o parte opțională cu lungime variabilă (Figura 37).

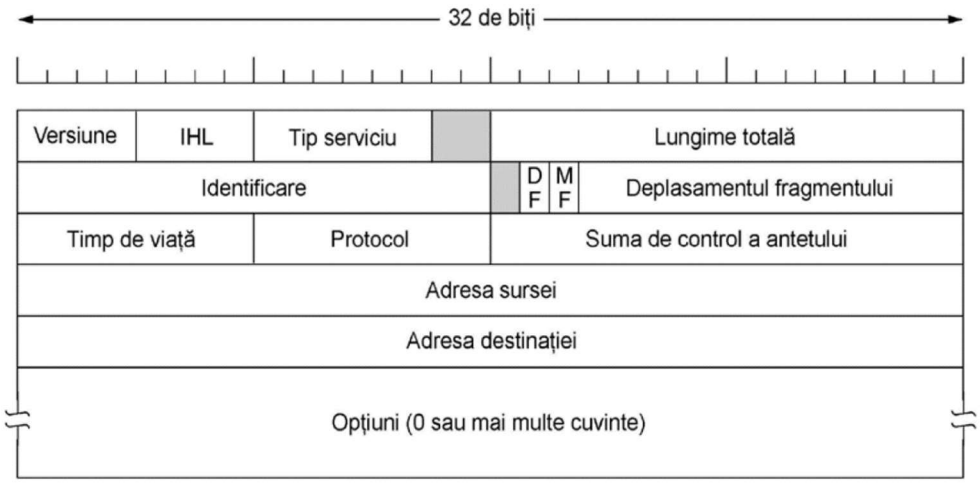

Figura 37. Antetul IPv4

Fiecare gazdă și ruter din Internet are o adresă IP, care codifică adresa sa de rețea și de gazdă. Combinația este unică: în principiu nu există două mașini cu aceeași adresă IP. Toate adresele IP sunt de 32 de biți lungime și sunt folosite în câmpurile Adresă sursă și Adresă destinație ale pachetelor IP. Este important de observat că o adresă IP nu se referă de fapt la o gazdă. Se referă de fapt la o interfață de rețea, deci dacă o gazdă este în două rețele, trebuie să folosească două adrese IP. Totuși în practică, cele mai multe gazde sunt conectate la o singură rețea și deci au o adresă IP.

Timp de mai multe decenii, adresele IPv4 erau împărțite în cinci categorii ilustrate în Figura 38. Acest model de alocare a fost denumit **clase de adrese**. Nu mai este folosit, dar referințele la acest model sunt în continuare des întâlnite în literatură.

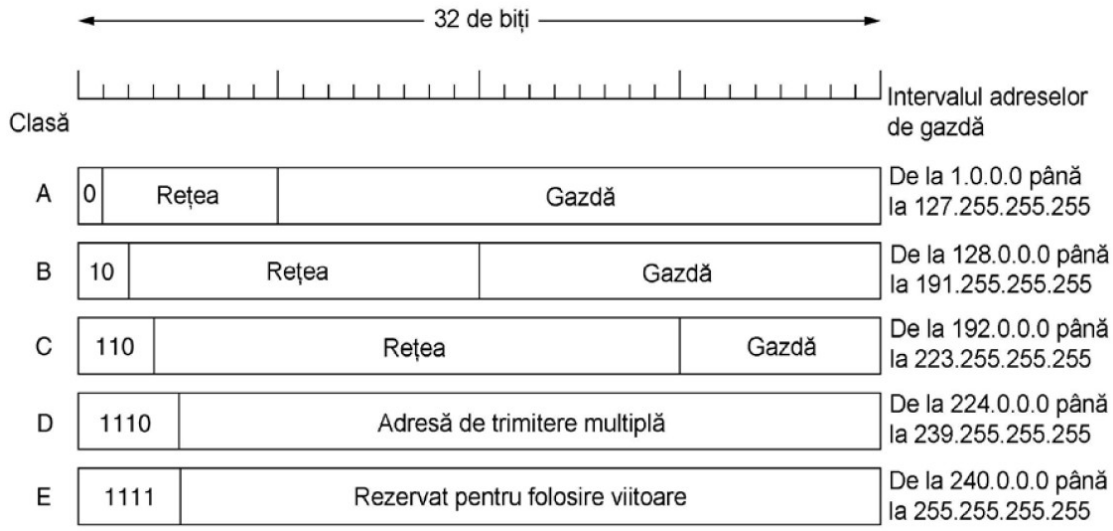

Figura 38. Formatul adreselor IPv4

Formatele de clasă A, B, C permit până la 128 rețele cu 16 milioane de gazde fiecare, 16.384 rețele cu până la 64K gazde, 2 milioane de rețele (de exemplu, LAN-uri) cu până la 256 gazde fiecare (deși unele dintre acestea sunt speciale). Pentru a evita conflictele numerele de rețea sunt atribuite de ICANN (Corporația Internet pentru Numere și Nume Atribuite). La rândul său, ICANN a împuternicit diverse autorități regionale să administreze părți din spațiul de adrese și acestea, la rândul lor, au împărțit adrese ISP-urilor și altor companii.

Adresele de rețea, care sunt numere de 32 de biți, sunt scrise în mod uzual în notația zecimală cu punct. În acest format, fiecare din cei 4 octeți este scris în zecimal, de la 0 la 255. Cea mai mică adresă IP este 0.0.0.0 și cea mai mare este 255.255.255.255.

Adrese IP speciale:

- Adresa IP 0.0.0.0 este folosită de gazde atunci când sunt pornite. Adresele IP cu 0 ca ID de rețea se referă la rețeaua curentă.
- Adresele care constau numai din 1-uri permit difuzarea în rețeaua curentă, în mod uzual un LAN.
- Adresele cu un ID exact de rețea și numai 1-uri în câmpul gazdă permit calculatoarelor să trimită pachete de difuzare în LAN-uri la distanță, aflate oriunde în Internet (deși mulți administratori de sistem dezactivează această opțiune).
- Toate adresele de forma 127.xx.yy.zz sunt rezervate pentru testări în buclă locală (*loopback*). Pachetele trimise către această adresă nu sunt trimise prin cablu; ele sunt prelucrate local și tratate ca pachete sosite.

O singură adresă din clasele definite mai sus se referă la o singură rețea. Deoarece clasele A sau B conțin în teorie un număr mare de gazde, iar în practică nu există rețele atât de mari, s-a permis ca o rețea să fie divizată în mai multe părți pentru uz intern, dar pentru lumea

exterioară să se comporte tot ca o singură rețea și astfel au apărut subrețelele. Pentru aceasta, un număr de biți din ID-ul gazdei sunt folosiți pentru a crea un număr de subrețea.

Pentru a se putea folosi subrețele, ruterul principal are nevoie de o mască de subrețea, care indică separarea dintre ID-urile rețea + subrețea și gazdă: toți biții măștii de subrețea ce corespund ID-ului rețelei au valoarea 1, iar toți biții corespunzători ID-ului gazdei sunt 0.

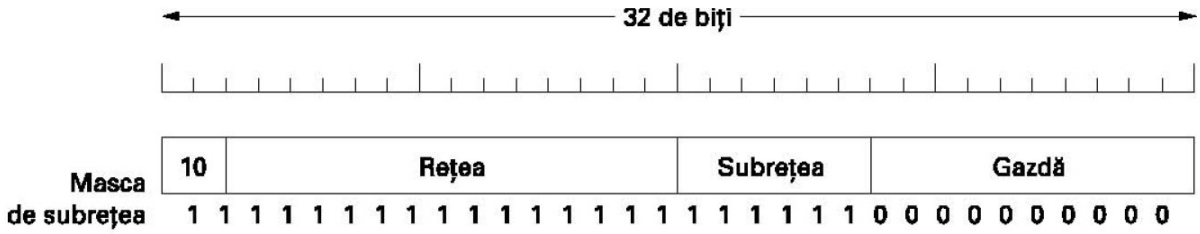

Figura 39. O rețea de clasă B împărțită în 64 de subrețele

4.1.2 IP v6

În situația epuizării adreselor IPv4, IETF a început să lucreze în 1990 la o nouă versiune de IP, una care să nu își epuizeze niciodată adresele, să rezolve o gamă largă de alte probleme și să fie totodată mai flexibilă și mai eficientă. Obiectivele majore au fost:

1. Să suporte miliarde de gazde, chiar cu alocare ineficientă a spațiului de adrese.
2. Să reducă dimensiunea tabelelor de rutare.
3. Să simplifice protocolul, pentru a permite ruterelor să proceseze pachetele mai rapid.
4. Să asigure o securitate mai bună (autentificare și confidențialitate) față de IP-ul curent.
5. Să acorde o mai mari atenție tipului de serviciu, în special pentru datele de timp real.
6. Să ajute trimiterea multiplă, permițând specificarea de domenii.
7. Să creeze condițiile pentru ca o gazdă să poată migra fără schimbarea adresei sale.
8. Să permită evoluția protocolului în viitor.
9. Să permită coexistența noului și vechiului protocol pentru câțiva ani.

Astfel a apărut IPv6, care, deși nu este compatibil cu IPv4, este compatibil cu celelalte protocoale Internet auxiliare, incluzând TCP, UDP, ICMP, IGMP, OSPF, BGP și DNS, câteodată fiind necesare mici modificări (majoritatea pentru a putea lucra cu adrese mai lungi). Mai multe informații despre el pot fi găsite în RFC 2460 până la RFC 2466.

IPv6 are adrese mai lungi decât IPv4. Ele au o lungime de 16 octeți, ceea ce rezolvă problema pentru a cărei soluționare a fost creat IPv6: să furnizeze o sursă efectiv nelimitată de adrese Internet. Mai exact, sunt 2^{128} adrese, care reprezintă aproximativ $3{,}4 \times 10^{38}$. Chiar și în cel mai pesimist scenariu, vor fi totuși mult peste 1000 de adrese IP pe metru pătrat de suprafață planetară. În orice scenariu credibil, vor fi trilioane de adrese pe metru pătrat. Pe scurt, pare improbabil că vor fi epuizate adresele în viitorul previzibil.

A doua mare îmbunătățire a lui IPv6 este simplificarea antetului. El conține numai 7 câmpuri (față de 13 în IPv4). Această schimbare permite ruterelor să prelucreze pachetele mai rapid, îmbunătățind astfel productivitatea și întârzierea.

A treia mare îmbunătățire a fost suportul mai bun pentru opțiuni. Această schimbare a fost esențială în noul antet, deoarece câmpurile care erau necesare anterior sunt acum opționale. În plus, modul în care sunt reprezentate opțiunile este diferit, ușurând ruterelor saltul peste opțiunile care nu le sunt destinate. Această caracteristică accelerează timpul de prelucrare a pachetelor.

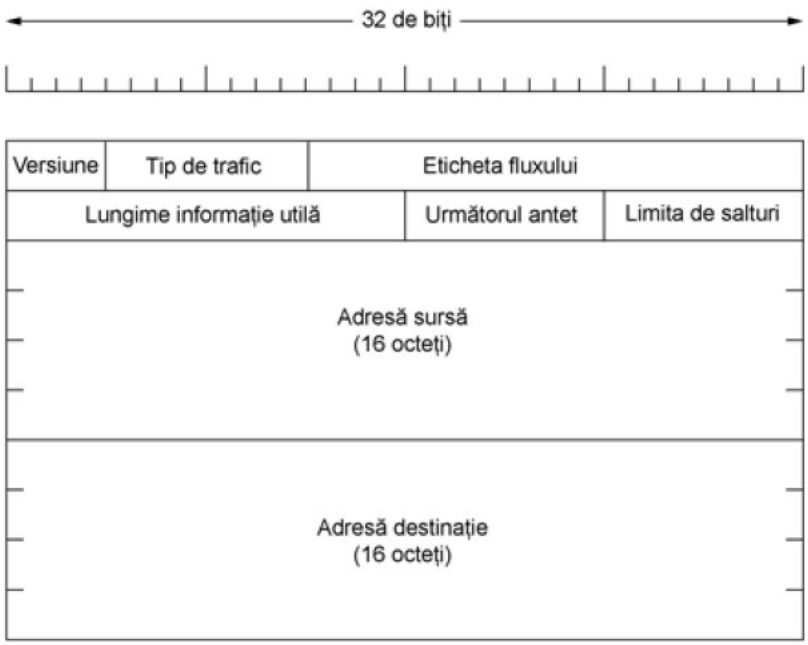

Figura 40. Antetul IPv6

Pentru scrierea adreselor de 16 octeți a fost inventată o nouă notație. Ele sunt scrise ca opt grupuri de câte patru cifre hexazecimale cu semnul : (două puncte) între grupuri, astfel:

8000:0000:0000:0000:0123:4567:89AB:CDEF

Din moment ce multe adrese vor avea multe zerouri în interiorul lor, au fost autorizate unele optimizări. Mai întâi, zerourile de la începutul unui grup pot fi omise, astfel încât 0123 poate fi scris ca 123. În al doilea rând, unul sau mai multe grupuri de 16 zerouri pot fi înlocuite de o pereche de semne două puncte (::). Astfel, adresa de mai sus devine:

8000::123:4567:89AB:CDEF

4.2 Translatarea adreselor de rețea

În perioada de tranziție dintre IPv4 și IPv6, pentru a depăși problema insuficienței adreselor de tip IPv4, s-a propus o rezolvare rapidă pe termen scurt: translatarea adreselor de rețea (NAT) care este descrisă în RFC 3022.

Ideea de bază a NAT-ului este de a aloca fiecărei companii o singură adresă IP (sau cel mult un număr mic de adrese) pentru traficul Internet. În interiorul companiei, fiecare calculator primește o adresă IP unică, care este folosită pentru traficul intern. Totuși, atunci când un pachet părăsește compania și se duce la ISP, are loc o translatare de adresă (conversia adresei interne

în cea reală). Pentru a face posibil acest lucru, au fost declarate ca fiind private trei intervale de adrese IP. Companiile le pot folosi intern așa cum doresc. Singura regulă este ca nici un pachet conținând aceste adrese să nu apară pe Internet. Cele trei intervale rezervate sunt:

- 10.0.0.0 -10.255.255.255/8 (16.777.216 gazde)
- 172.16.0.0 -172.31.255.255/12 (1.048.576 gazde)
- 192.168.0.0 -192.168.255.255/16 (65.536 gazde)

Când un pachet părăsește compania, trece printr-o unitate NAT (*NAT box*) care convertește adresa IP internă în adresa IP reală a companiei. Apare totuși o problemă atunci când sosește înapoi răspunsul, iar unitatea NAT trebuie să-l trimită în rețeaua locală către gazda corectă.

Folosind câmpul Port sursă se rezolvă problema de corespondență. De fiecare dată când un pachet pleacă, el intră în unitatea NAT și adresa sursă este înlocuită cu adresa reală a companiei. În plus, câmpul TCP Port sursă este înlocuit cu un index în tabela de translatare a unității NAT, care are 65.536 intrări. Această tabelă conține adresa IP inițială și portul inițial. În final, sunt recalculate și inserate în pachet sumele de control ale antetelor IP și TCP. Câmpul Port sursă trebuie înlocuit pentru că s-ar putea întâmpla, de exemplu, ca două stații din aceeași rețea să aibă ambele conexiuni care să folosească un anumit port, deci câmpul Port sursă nu este suficient pentru a identifica procesul emițător.

Atunci când la unitatea NAT sosește un pachet de la ISP, Portul sursă din antetul TCP este extras și folosit ca index în tabela de corespondență a unității NAT. Din intrarea localizată sunt extrase și inserate în pachet adresa IP internă și Portul sursă TCP inițial. Apoi sunt recalculate sumele de control TCP și IP și inserate în pachet. După aceea pachetul este transferat ruterului companiei pentru transmitere normală folosind adresa internă.

4.3 Protocoale de control în Internet

Pe lângă IP, care este folosit pentru transferul de date, Internetul are câteva protocoale de control la nivelul rețea, incluzând ICMP, ARP, RARP, BOOTP și DHCP.

4.3.1 Protocolul mesajelor de control din Internet

Operarea Internet-ului este strâns monitorizată de către rutere. Atunci când se întâmplă ceva neobișnuit, evenimentul este raportat prin ICMP (Protocolul mesajelor de control din Internet). Cele mai importante mesaje sunt enumerate în Figura 41. Fiecare tip de mesaj ICMP este încapsulat într-un pachet IP.

Tipul mesajului	Descriere
Destinaţie inaccesibilă	Pachetul nu poate fi livrat
Timp depăşit	Câmpul timp de viaţă a ajuns la 0
Problemă de parametru	Câmp invalid în antet
Oprire sursă	Pachet de şoc
Redirectare	Învaţă un ruter despre geografie
Cerere de ecou	Întreabă o maşină dacă este activă
Răspuns ecou	Da, sunt activă
Cerere de amprentă de timp	La fel ca cererea de ecou, dar cu amprentă de timp
Răspuns cu amprentă de timp	La fel ca răspunsul ecou, dar cu amprentă de timp

Figura 41. Tipuri de mesaje ICMP

4.3.2 Protocolul de rezoluție a adresei

ARP este un protocol de comunicație utilizat pentru descoperirea adresei MAC a unui calculator asociată unei adrese IP cunoscută.

Fabricanții plăcilor de rețea cer un spațiu de adrese de la o autoritate centrală pentru a se asigura că nu există două plăci cu aceeași adresă (pentru a evita conflictele care ar apărea dacă cele două plăci ar fi în aceeași rețea). Plăcile trimit și primesc cadre pe baza adresei MAC de 48 biți. Deși fiecare calculator din Internet are una sau mai multe adrese IP, acestea nu pot fi folosite de fapt pentru trimiterea pachetelor deoarece hardware-ul nivelului legăturii de date nu înțelege adresele Internet.

Protocolul ARP permite unei gazde să trimită un pachet de difuzare în rețea prin care să întrebe cine este proprietarul unei adrese IP specificate în acesta, la acest pachet răspunzând cu adresa MAC numai gazda cu adresa IP menționată. ARP este definit în RFC 826.

4.3.3 Protocolul Dinamic de Configurare a Gazdei

DHCP (Protocol Dinamic de Configurare a Gazdei) permite atribuirea manuală sau automată de adrese IP calculatoarelor dintr-o rețea și este bazat pe ideea unui server special care atribuie adrese IP gazdelor care cer una. Este descris în RFC-urile 2131 și 2132.

Pentru a obține o adresă IP, un calculator tocmai pornit difuzează un pachet DHCP DISCOVER. Serverul DHCP din rețea interceptează toate difuzările de acest gen și alocă acestuia o adresă IP.

O problemă care apare cu atribuirea automată a adreselor IP dintr-o rezervă comună este cât timp ar trebui alocată o adresă IP. Dacă o gazdă părăsește rețeaua și nu returnează adresa sa IP serverului DHCP, acea adresă va fi pierdută permanent. După o perioadă de timp vor fi pierdute multe adrese. Pentru a preveni aceasta, atribuirea adresei IP va fi pentru o perioadă fixă de timp, o tehnică numită închiriere (*lease*). Chiar înainte ca perioada de închiriere să expire, gazda trebuie să îi ceară DHCP-ului o reînnoire. Dacă nu reușește să facă cererea sau dacă cererea este respinsă, gazda nu va mai putea folosi adresa IP primită anterior.

4.4 Protocoale de rutare

Internetul este construit dintr-un număr mare de sisteme autonome (AS). Fiecare AS este administrat de o organizație diferită și poate folosi propriul algoritm de rutare a pachetelor

în interior. Un algoritm de rutare în interiorul unui AS este numit protocol de *gateway* interior; un algoritm de rutare utilizat între AS-uri este numit protocol de *gateway* exterior.

4.4.1 Protocolul de gateway interior

Primul protocol de acest gen a fost unul de rutare pe baza vectorilor distanță (RIP) ce utilizau numărarea hop-urilor (nodurilor) și unele metrici de rutare (lungimea căii, lățime de bandă, încărcare, întârziere, etc.), ce funcționa bine în sisteme mici.

A fost înlocuit cu un protocol bazat pe starea legăturilor, mai exact pe calea cea mai scurtă numit OSPF (Protocol Public (deschis) bazat pe Calea cea mai Scurtă).

OSPF suportă trei tipuri de conexiuni și rețele:

- Linii punct-la-punct între exact două rutere.
- Rețele multi-acces cu difuzare (de exemplu, cele mai multe LAN-uri).
- Rețele multi-acces fără difuzare (de exemplu, cele mai multe WAN-uri cu comutare de pachete).

OSPF funcționează prin abstractizarea colecției de rețele, rutere și linii reale într-un graf orientat în care fiecărui arc îi este atribuit un cost (distanță, întârziere etc.). Apoi calculează cea mai scurtă cale bazându-se pe ponderile arcelor. O conexiune serială între două rutere este reprezentată de o pereche de arce, câte unul în fiecare direcție. Ponderile lor pot fi diferite. O rețea multi-acces este reprezentată de un nod pentru rețeaua însăși, plus câte un nod pentru fiecare ruter. Arcele de la nodul rețea la rutere au pondere 0 și sunt omise din graf.

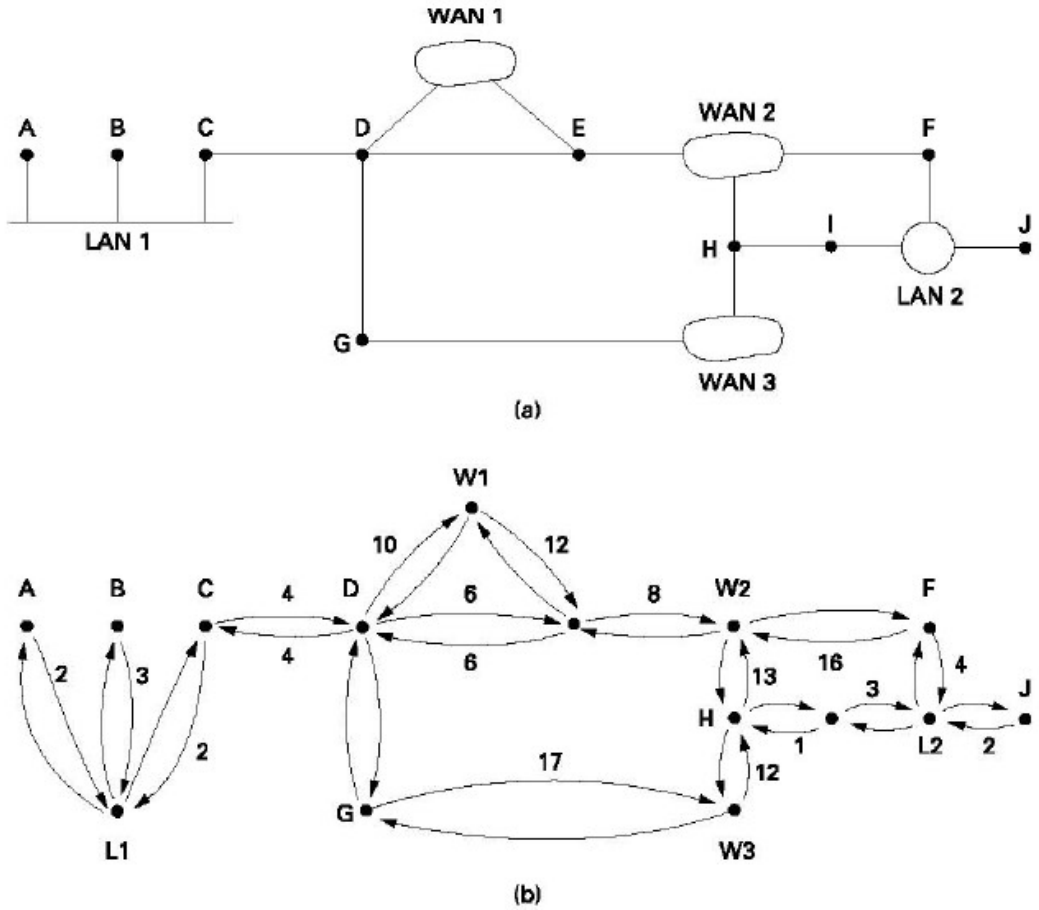

Figura 42. (a) Un sistem autonom. (b) O reprezentare de tip graf a lui (a)

4.4.2 Protocolul de gateway exterior

Între AS-uri se folosește un protocol diferit, BGP (Protocolul Porților de Graniță), pentru că scopurile unui protocol pentru *gateway* interior și ale unui protocol pentru *gateway* exterior sunt diferite. Tot ce trebuie să facă un protocol pentru *gateway* interior este să mute pachetele cât mai eficient de la sursă la destinație.

Ruterele ce folosesc protocolul de *gateway* exterior trebuie să țină cont într-o mare măsură de politică. De exemplu, un AS al unei corporații poate dori facilitatea de a trimite pachete oricărui *site* Internet și să recepționeze pachete de la orice *site* Internet. Cu toate acestea, poate să nu dorească să asigure tranzitarea pentru pachetele originare dintr-un AS străin destinate unui AS străin diferit, chiar dacă prin AS-ul propriu trece cea mai scurtă cale dintre cele două AS-uri străine („Asta este problema lor, nu a noastră.”). Pe de altă parte, poate fi dispus să asigure tranzitarea pentru vecinii săi, sau chiar pentru anumite AS-uri care au plătit pentru acest serviciu. Companiile telefonice, de exemplu, pot acționa ca un purtător pentru clienții lor, dar nu și pentru alții. Protocoalele pentru *gateway* exterior, în general și BGP în particular, au fost proiectate pentru a permite forțarea multor tipuri de politici de rutare pentru traficul dintre AS-uri.

Din punctul de vedere al unui ruter BGP, lumea constă din AS-uri și liniile care le conectează. Două AS-uri sunt considerate conectate dacă există o linie între două rutere de graniță din fiecare. Dat fiind interesul special al BGP-ului pentru traficul în tranzit, rețelele sunt grupate în trei categorii.

1. Rețele ciot (*stub networks*), care au doar o conexiune la graful BGP. Acestea nu pot fi folosite pentru traficul în tranzit pentru că nu este nimeni la capătul celălalt.
2. Rețele multi-conectate. Acestea pot fi folosite pentru traficul în tranzit, cu excepția celor care refuză.
3. Rețele de tranzit, cum ar fi coloanele vertebrale, care sunt doritoare să manevreze pachetele altora, eventual cu unele restricții, și de obicei pentru o plată.

BGP este la bază un protocol bazat pe vectori distanță, dar destul de diferit de celelalte protocoale cum ar fi RIP. În loc să mențină doar costul până la fiecare destinație, fiecare ruter BGP memorează calea exactă folosită. Similar, în loc să trimită periodic fiecărui vecin costul său estimat către fiecare destinație posibilă, fiecare ruter BGP comunică vecinilor calea exactă pe care o folosește. Definirea BGP-ului se găsește în RFC 1771 și 1774.

Capitolul 5. Nivelul Acces la rețea

La baza stivei de protocoale TCP/IP este nivelul de Acces la Rețea, o colecție de servicii și specificații care furnizează și gestionează accesul la echipamentele de rețea. Acest nivel administrează toate serviciile și funcțiile necesare pentru pregătirea datelor pentru rețeaua fizică, responsabilitățile lui incluzând:

- Interfațarea cu adaptorul de rețea al calculatorului.
- Coordonarea transmiterii datelor, aplicând regulile metodei de acces corespunzătoare.
- Conversia datelor într-un format potrivit pentru transmiterea unui flux de impulsuri electrice sau luminoase prin mediul de transmisie.
- Verificarea de erori a datelor primite.
- Adăugarea de informații de verificare a erorilor în datele de ieșire, astfel încât calculatorul destinație să poată verifica datele pentru erori.

Nivelul de Acces la Rețea definește procedurile de interfațare cu echipamentele de rețea și accesarea mediului de transmisie. Din fericire, este aproape invizibil pentru utilizatorul de zi cu zi. Driverul adaptorului de rețea, împreună cu componente cheie de nivel inferior ale sistemului de operare, gestionează cele mai multe dintre sarcinile nivelului de Acces la Rețea, iar câțiva pași de configurare scurți sunt de obicei tot ceea ce este necesar unui utilizator. Acești pași sunt simplificați prin îmbunătățirea funcțiilor *plug-and-play* și autoconfigurare ale sistemelor de operare.

Acest nivel TCP/IP corespunde cu aproximație nivelurilor Fizic și Legătură de Date ale modelului OSI. Unitățile de date corespunzătoare sunt cadrul și șirurile de biți (Figura 43).

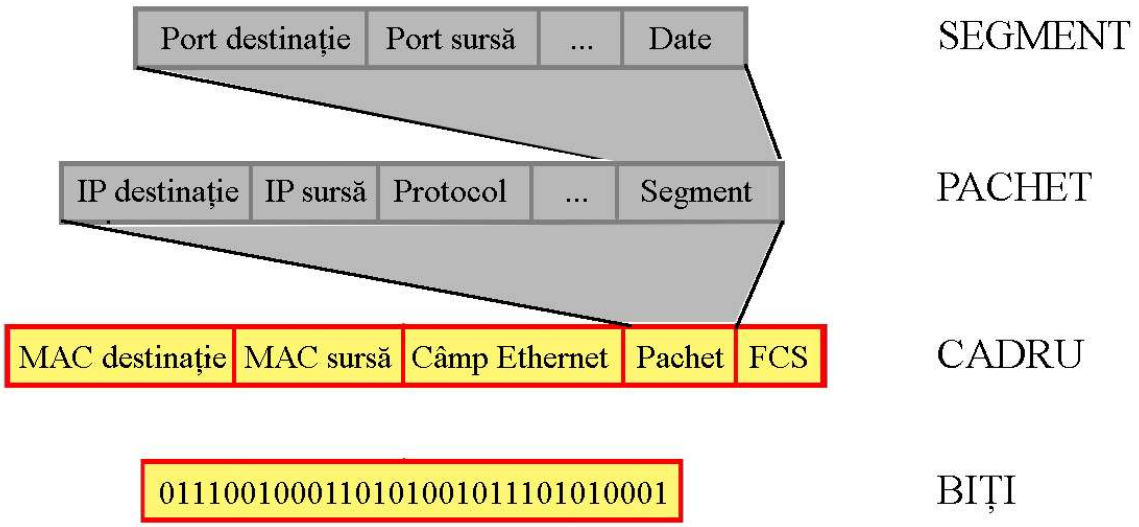

Figura 43. Unitățile de date pentru Nivelul Acces la Rețea

Nivelul Fizic OSI este responsabil pentru transformarea cadrului de date într-un flux de biți adecvat mediului de transmisie. Cu alte cuvinte, acesta gestionează și sincronizează impulsurile electrice sau luminoase care formează transmisia reală. La destinație, nivelul Fizic reasamblează aceste impulsuri într-un cadru de date.

Nivelul de Legătură de Date efectuează două funcții separate și, prin urmare, este împărțit în următoarele două sub-niveluri:

- Controlul Accesului la Mediu (MAC): acest sub-nivel oferă o interfață cu adaptorul de rețea. Adresa hardware a adaptorului de rețea, încorporată în acesta din fabrică, este adesea denumită adresa MAC.
- Controlul Logic al Conexiunii (LLC): Acest sub-nivel efectuează funcții de verificare a erorilor pentru cadrele livrate prin rețea și gestionează legăturile dintre dispozitivele care comunică.

5.1 Arhitectura rețelei

O arhitectură de rețea, cum ar fi Ethernet, oferă un pachet de specificații care reglementează adresarea fizică, accesul la și interacțiunea cu mediul de comunicație. Practic, proiectarea unei arhitecturi de rețea înseamnă de fapt proiectarea nivelului de Acces la Rețea.

O arhitectură de rețea este reprezentată de proiectarea unei rețele fizice și de stabilirea unei colecții de specificații care definesc comunicațiile în acea rețea fizică. Detaliile de comunicare depind de detaliile fizice, astfel încât specificațiile vin de obicei împreună, ca un pachet complet. Aceste specificații includ considerente precum:

- Metoda de acces: este un set de reguli care definesc modul în care calculatoarele vor partaja mediul de transmisie. Pentru a evita coliziunile de date, calculatoarele trebuie să respecte aceste reguli atunci când efectuează transmisii.
- Formatul cadrului de date: datagramele de la nivelul Internet sunt încapsulate în cadre de date cu un format predefinit. Datele din antet trebuie să furnizeze informațiile necesare pentru a livra date în rețeaua fizică.
- Tipul de cabluri: tipul de cablu utilizat pentru o rețea are efect asupra anumitor parametri de proiectare, cum ar fi proprietățile șirului de biți transmiși de adaptor.
- Norme de cablare: protocoalele, tipul cablurilor și proprietățile transmisiei au un efect asupra lungimilor maxime și minime ale cablului și ale specificațiilor conectorului cablului.

Detaliile, cum ar fi tipul de cablu și tipul de conector, nu reprezintă responsabilitatea directă a nivelului de Acces la Rețea dar, pentru a proiecta componentele software ale acestui nivel, dezvoltatorii trebuie să își asume un set specific de caracteristici pentru rețeaua fizică. Important este că nivelurile superioare nu trebuie să cunoască detalii despre componentele hardware ale rețelei. Stiva de protocoale TCP/IP este proiectată astfel încât toate detaliile de interacțiune cu acestea să apară la nivelul de Acces la Rețea. Acest design permite ca TCP/IP să funcționeze pe o mare varietate de medii de transmisie diferite.

Exemple de arhitecturi de rețea dezvoltate a acest nivel sunt:

- IEEE 802.3 (Ethernet): rețea bazată pe cabluri folosită în majoritatea locuințelor și birourilor.

- IEEE 802.11 (Wi-Fi): rețea fără fir folosită în majoritatea locuințelor și birourilor.
- IEEE 802.16 (WiMAX): o tehnologie fără fir folosită pentru comunicații pe distanțe mari.
- Point-to-Point Protocol (PPP): protocol utilizat pentru conexiuni între modem-uri.

5.2 Adresarea fizică

Nivelul de Acces la Rețea este necesar pentru a face legătura între adresa de IP, care este configurată prin software-ul de protocol la nivelul Internet, cu adresa fizică reală permanentă a adaptorului de rețea. Această adresă fizică este denumită adesea adresă MAC deoarece, în cadrul modelului OSI, adresarea fizică este responsabilitatea sub-nivelului de Control al Accesului la Mediu (MAC). Deoarece sistemul de adresare fizică este încapsulat în nivelul de Acces la Rețea, adresa poate avea un format diferit, în funcție de specificațiile arhitecturii rețelei.

În cazul rețelei Ethernet, adresa fizică este, de obicei, încorporată în adaptorul de rețea din fabrică, dar marea majoritate a acestora oferă posibilitatea de a o schimba în sistemul de operare cu ajutorul driverului. Cadrele de date trimise prin intermediul rețelei trebuie să utilizeze această adresă fizică pentru a identifica adaptoarele sursă și destinație.

O adresă MAC este formată dintr-o succesiune de 48 de biți. Primii 24 identifică producătorul adaptorului de rețea, iar ultimii 24 identifică adaptorul. Primii doi biți ai primului octet au funcții speciale: primul identifică dacă mesajele trimise de adaptor sunt destinate unui singur calculator (*unicast*) sau mai multor calculatoare (*multicast*), iar al doilea dacă adresa MAC a fost alocată de către producător sau modificată de administratorul de rețea.

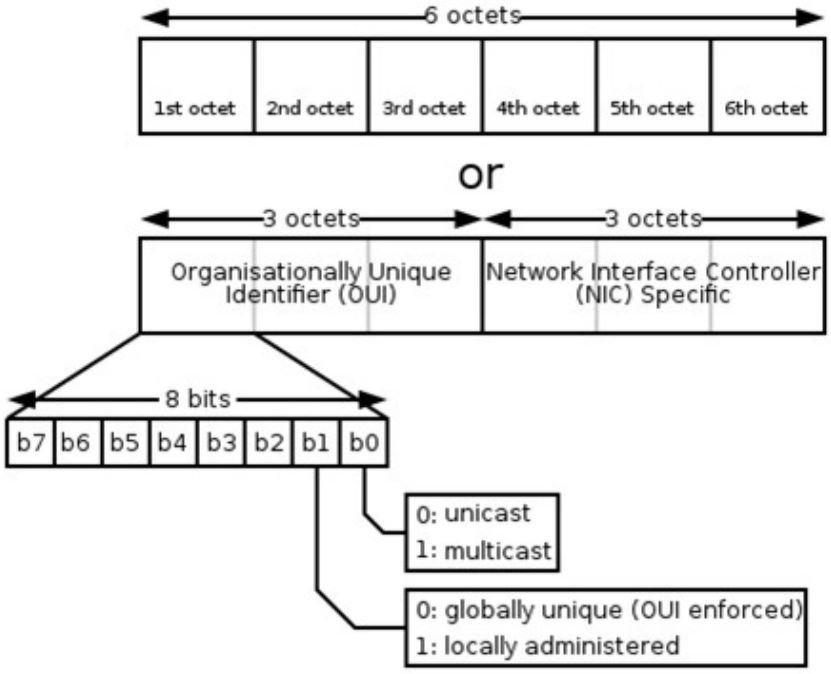

Figura 44. Structura adresei MAC

5.3 Ethernet

Ethernet este, fără îndoială, cea mai populară tehnologie LAN utilizată în prezent. Arhitectura Ethernet a devenit populară datorită prețului său scăzut: cablul Ethernet este ieftin și ușor de instalat. Adaptoarele de rețea Ethernet și componentele hardware Ethernet sunt, de asemenea, relativ ieftine. Răspândirea rețelelor fără fir nu a diminuat importanța Ethernet-ului. O formă importantă de rețea LAN fără fir este uneori numită "Ethernet fără fir" deoarece încorporează multe dintre principiile specificațiilor inițiale ale rețelei Ethernet.

Într-o rețea clasică Ethernet, toate computerele au un mediu de transmisie comun. Ethernet utilizează o metodă de acces numită Acces Multiplu cu Detecția Purtătoarei și Detecția Coliziunilor (CSMA/CD) pentru a determina când un calculator este liber să transmită date prin mediul de comunicație. Utilizând CSMA/CD, toate calculatoarele monitorizează mediul și așteaptă până când linia este disponibilă înainte de a transmite. Dacă două calculatoare încearcă să transmită în același timp, apare o coliziune. Calculatoarele se opresc, așteaptă un interval de timp aleatoriu și încearcă să transmită din nou.

Rețelele Ethernet tradițională funcționează bine pentru trafic ușor până la moderat, rata coliziunilor crescând în condiții de utilizare intensă. În rețelele Ethernet moderne, dispozitive precum *switch*-urile de rețea gestionează traficul pentru a reduce incidența coliziunilor, permițând astfel Ethernet-ului să funcționeze mai eficient, prin împărțirea unei rețele în mai multe domenii de coliziune.

Domeniul de coliziune reprezintă un grup de calculatoare conectate fizic prin dispozitive ce lucrează la nivelul de Rețea (repetor, hub, *tranceiver*) în care se pot produce coliziuni. *Switch*-urile, lucrând la nivelul de Legătură de Date, dirijează cadrele de date numai către calculatoarele cărora le sunt destinate, pe baza adresei MAC. Prin urmare acestea separă o rețea în mai multe domenii de coliziune, în funcție de numărul de porturi.

Structura unui cadru Ethernet este prezentată în Figura 45.

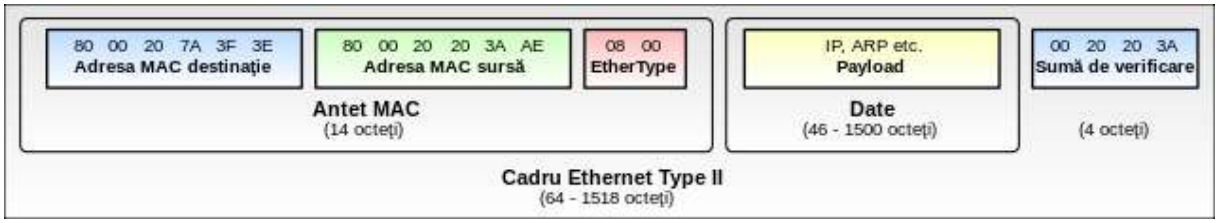

Figura 45. Structura unui cadru Ethernet

5.4 Codificarea Manchester

În ceea ce privește transmiterea șirurilor de biți prin mediul de comunicație în rețelele Ethernet nu se folosește o codificare binară directă, cu 0 volți pentru un bit 0 și 5 volți pentru un bit 1, deoarece aceasta conduce la ambiguități. Dacă o stație trimite șirul de biți 00010000, altele l-ar putea interpreta fals ca 10000000 sau 01000000 întrucât nu pot distinge diferența între un emițător inactiv (0 volți) și un bit 0 (0 volți).

Această problemă poate fi rezolvată utilizându-se codificarea Manchester în care fiecare perioadă a unui bit este împărțită în două intervale egale. Un bit 1 este trimis stabilind un voltaj

ridicat în timpul primului interval și scăzut în cel de-al doilea. Un 0 binar este trimis exact invers: întâi nivelul scăzut iar apoi cel ridicat. Această strategie asigură că fiecare perioadă a unui bit are o tranziție la mijloc, ușurând sincronizarea între emițător și receptor.

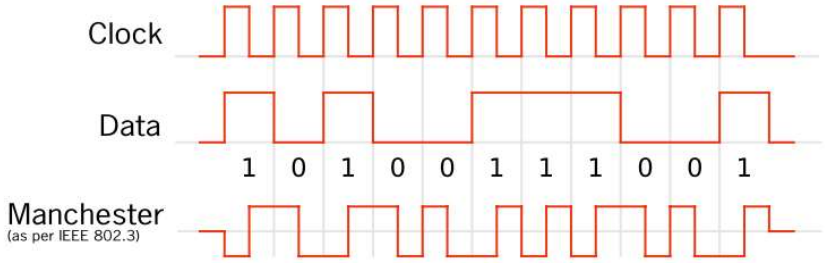

Figura 46. Codificarea Manchester

Capitolul 6. Rețele fără fir

O rețea fără fir este formată din gazde, ce pot fi mobile sau nu, asociate unor stații de bază / puncte de acces, cu rol administrare a rețelei și de trimitere și recepție a datelor către / de la gazde, între acestea stabilindu-se legături de comunicație.

În funcție de elementele componente ale unei rețele fără fir (dacă există sau nu puncte de acces) și de modul în care acestea comunică (dacă este necesară o singură legătură de comunicație – *single-hop* – sau mai multe – *multi-hop*) rețelele fără fir se pot clasifica astfel [2]:

- *Single-hop* bazată pe infrastructură, în care există un punct de acces și toate gazdele se conectează la acesta (de exemplu în cazul rețelelor Wi-Fi și GSM).
- *Single-hop* fără infrastructură, în care una din gazde coordonează schimbul de date cu celelalte gazde (de exemplu în cazul rețelelor Bluetooth).
- *Multi-hop* bazată pe infrastructură, în care deși există un punct de acces, unele gazde nu se vor conecta la acesta în mod direct ci prin intermediul altor gazde (de exemplu în cazul rețelelor WSN).
- *Multi-hop* fără infrastructură, în care nu există puncte de acces, iar gazdele pot comunica între ele și prin intermediul altor gazde care fac parte din rețea (de exemplu în cazul rețelelor MANET sau VANET).

Deoarece aerul este un mediu partajat iar spectrul radio alocat comunicațiilor ce nu necesită licență de operare este redus, accesarea acestuia de către dispozitive care folosesc aceeași frecvență poate duce la apariția de interferențe. Efectul acestora depinde de distanța până la dispozitivul care o produce și de puterea de emisie a acestuia. De exemplu, banda de frecvențe nelicențiată de 2,4 GHz este utilizată de către telefoanele fără fir, rețelele Bluetooth, Wi-Fi sau ZigBee, sistemele de alarmă auto, cuptoare cu microunde, camere video sau microfoane.

Cele mai răspândite tehnologii de comunicație pentru rețele fără fir sunt Wi-Fi, Bluetooth, ZigBee, WiMAX și GSM.

6.1 Rețele Wi-Fi

Rețelele Wi-Fi se bazează pe seria de standarde 802.11, primul dintre acestea fiind lansat în 1997. Acesta putea opera în două moduri: unul în care calculatoarele se puteau conecta la un punct acces ce avea legătură prin cablu cu rețeaua de bază, iar altul în care două calculatoare se puteau conecta direct unul cu celălalt.

Standardul 802.11 folosește o parte din banda de comunicații ISM (Industrial, Științific și Medical). Aceasta permite utilizarea fără licență și a fost populară pentru sistemele automate pentru uși de garaj, telefoane fără fir și alte dispozitive electronice de larg consum. În Europa, cele mai utilizate sunt benzile de 2,4 GHz și 5 GHz, fiecare bandă fiind divizată în mai multe canale.

În Figura 47 sunt reprezentate cele 13 canale disponibile în rețelele Wi-Fi ce folosesc banda de 2,4 GHz (unele dintre ele sunt interzise, în funcție de zona de pe glob), fiecare canal având o anumită lățime de bandă: 20 MHz, 22 MHz sau 40 MHz. Deoarece spațierea între frecvențele centrale a două canale este de doar 5 MHz, se constată o suprapunere a celor 13 canale. Dar, așa cum se observă din Figura 47, dacă într-o anumită zonă se utilizează numai canalele 1, 6 sau 11, suprapunerea se poate evita.

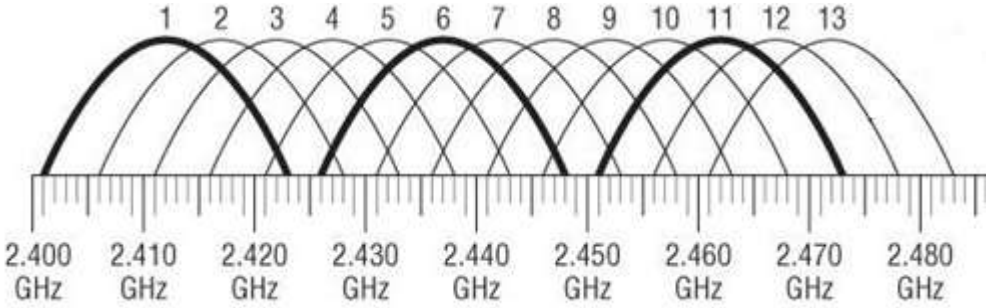

Figura 47. Canale Wi-Fi 2,4 GHz

Pentru banda de 5 GHz sunt disponibile mai multe canale (Figura 48) având lățimi de bandă de 20 MHz, 40 MHz, 80 MHz sau 160 MHz. Canalele nu se suprapun atâta timp cât în aceeași zonă se utilizează o singură valoare pentru lățimea de bandă.

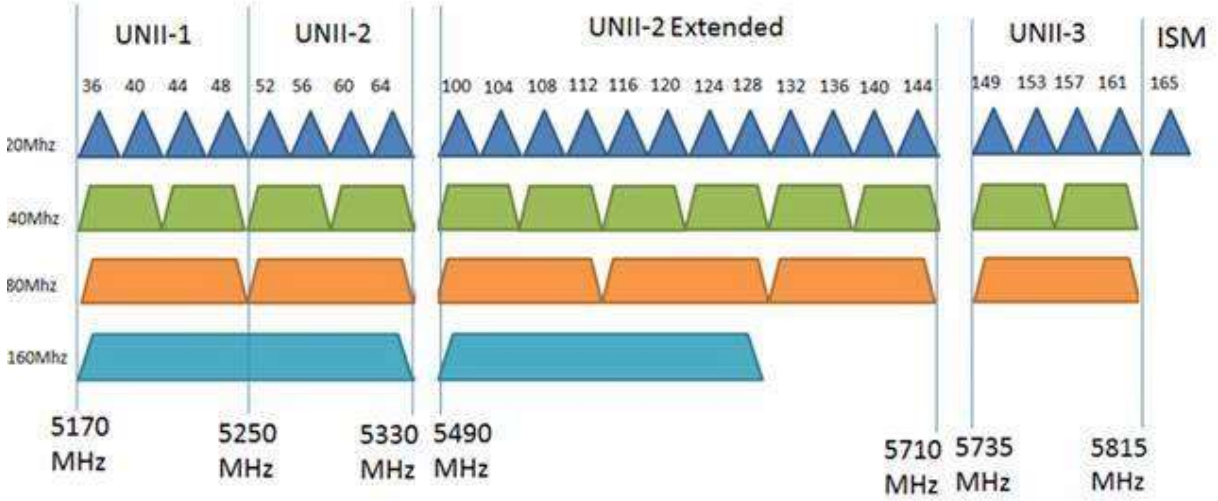

Figura 48. Canale Wi-Fi 5,8 GHz

Familia 802.11 (Figura 49) este alcătuită dintr-o serie de tehnici de modulație *half-duplex*, care utilizează același protocol de bază. 802.11-1997 a fost primul standard de rețea fără fir din familie, însă 802.11b a fost primul care a fost acceptat, urmat de 802.11a, 802.11g, 802.11n și 802.11ac. Alte standarde din familie (c-f, h, j) reprezintă amendamente, modificări ale serviciilor care sunt utilizate, pentru a extinde domeniul de aplicare actual al standardului existent, care pot include și corecții la o specificație anterioară.

802.11 network PHY standards								
802.11 protocol	Release date	Frequency (GHz)	Bandwidth (MHz)	Stream data rate (Mbit/s)	Allowable MIMO streams	Modulation	Approximate range	
							Indoor	Outdoor
802.11-1997	Jun 1997	2.4	22	1, 2	N/A	DSSS, FHSS	20 m (66 ft)	100 m (330 ft)
a	Sep 1999	5	20	6, 9, 12, 18, 24, 36, 48, 54	N/A	OFDM	35 m (115 ft)	120 m (390 ft)
		3.7						5,000 m (16,000 ft)[A]
b	Sep 1999	2.4	22	1, 2, 5.5, 11	N/A	DSSS	35 m (115 ft)	140 m (460 ft)
g	Jun 2003	2.4	20	6, 9, 12, 18, 24, 36, 48, 54	N/A	OFDM	38 m (125 ft)	140 m (460 ft)
n	Oct 2009	2.4/5	20	Up to 288.8	4	MIMO-OFDM	70 m (230 ft)	250 m (820 ft)
			40	Up to 600				
ac	Dec 2013	5	20	Up to 346.8	8		35 m (115 ft)	
			40	Up to 800				
			80	Up to 1733.2				
			160	Up to 3466.8				
		0.054-0.79	6-8	Up to 568.9	4			
ad	Dec 2012	60	2,160	Up to 6,757 (6.7 Gbit/s)	N/A	OFDM, single carrier, low-power single carrier	3.3 m (11 ft)	
ah	Dec 2016	0.9	1-16	Up to 347	4	MIMO-OFDM		

Figura 49. Variante ale standardului 802.11

Un caz special de amendament al standardului 802.11 este 802.11p ce permite realizarea de sisteme de comunicație pentru mediul vehicular. Denumit și WAVE (Acces Wireless în Medii Vehiculare), definește îmbunătățirile necesare pentru suportul Sistemelor Inteligente pentru Transporturi. Acestea includ schimbul de date între vehiculele de mare viteză (V2V) și între vehicule și infrastructura rutieră (V2I), în banda de 5,9 GHz.

Primele tipuri rețele Wi-Fi utilizau tehnici de acces al canalului de tip OFDM sau DSSS. Multiplexarea cu diviziune ortogonală de frecvență (OFDM) împarte un canal radio într-un număr mare de sub-purtătoare apropiate ce nu interferează, deși se suprapun în frecvență, datorită distanței dintre acestea aleasă corespunzător, iar tehnica de tip Spectru împrăștiat cu secvență directă (DSSS) multiplică datele ce trebuie transmise cu un semnal de tip zgomot. Ambele tehnici permit partajarea unui canal de comunicație între mai mulți utilizatori precum și obținerea unei mai mari imunități la perturbații și interferențe.

Creșterea ratelor de transfer s-a putut realiza prin Multiplexarea cu diviziune ortogonală de frecvență – Multiple intrări și Multiple ieșiri (MIMO-OFDM), aceasta fiind interfața dominantă pentru comunicații fără fir în bandă largă. Aceasta combină tehnologia MIMO, care multiplică rata de transfer prin transmiterea fluxului de date cu ajutorul mai multor antene, cu multiplexarea cu diviziune ortogonală de frecvență.

6.2 Rețele Bluetooth

Tehnologia Bluetooth a fost inventată de Ericsson în 1994 și este gestionată de Grupul de Interese Speciale Bluetooth (SIG), care deține peste 30.000 de companii membre din domeniul telecomunicațiilor, calculatoarelor, rețelelor și electronicii de consum. IEEE a standardizat Bluetooth ca IEEE 802.15.1, dar în momentul de față nu îl mai actualizează.

Numele "Bluetooth" este o versiune anglicizată a numelui scandinav Blåtand / Blåtann, epitetul regelui Harald Bluetooth din secolul al zecelea, care a unit triburile daneze disonante într-un singur regat și, conform legendei, a introdus și creștinismul.

Versiunile Bluetooth apărute de-a lungul timpului sunt următoarele:

- Versiunile 1.0 și 1.0B au apărut în anul 1999 și au avut multe probleme în funcționare.

- Versiunea 1.1 a apărut în anul 2001 fiind corectate unele erori și adăugându-se un indicator de tip RSSI. Versiunea a fost ratificată de IEEE prin standardul 802.15.1-2002.
- Versiunea 1.2 a apărut în 2003 fiind adăugate noi facilități precum Saltul Adaptiv în Frecvență (AFH), controlul fluxului, moduri de retransmitere, Conexiuni Sincronizate Extinse (eSCO), rată de transfer până la 721 kbps.
- Versiunea 2.0 + EDR a fost lansată în 2004 și a mărit rata de transfer maximă până la 2,1 Mbps datorită utilizării EDR (Rate de Date Îmbunătățite).
- Versiunea 2.1 + EDR a apărut în 2007 și a adus ca noutate principală modalitatea de împerechere simplă și sigura (SSP) ce a îmbunătățit procedura de împerechere a dispozitivelor Bluetooth dar și securitatea.
- Versiunea 3.0 + HS a fost lansată în 2004 și a mărit rata de transfer maximă până la 24 Mbps dar nu prin legătura Bluetooth. Aceasta era folosită doar pentru inițierea și stabilirea conexiunii, transferul de date făcându-se printr-o legătura de tip 802.11.
- Versiunea 4.0 + LE a fost adoptată în 2010 și a introdus un mod de consum redus de energie care a permis comunicarea cu periferice și senzori pentru aplicații medicale și industriale, la prețuri mici și cu menținerea distanțelor de comunicație.
- Versiunea 5 a fost dezvăluită în 2016 iar noile sale caracteristici se concentrează, în principal, pe tehnologia IoT. Pentru BLE furnizează dublul vitezei (2 Mbps în rafală scurtă) în detrimentul distanței, sau de până la patru ori mai mare distanța în detrimentul ratei de transfer și de opt ori mai mare capacitatea de transmisie a datelor prin creșterea lungimilor pachetelor.

Bluetooth este un standard pentru comunicații fără fir dedicat schimbului de date pe distanțe scurte (folosind banda ISM de la 2,4 la 2,4835 GHz) între dispozitive fixe și mobile și construirii rețelelor personale (PAN).

Versiunile standard *Basic Rate / Enhanced Data Rate* utilizează spectrul împărțit în 79 de canale, fiecare având lățimea de bandă de 1 MHz.

Tehnica de acces a canalului de comunicație este AFHSS (Spectru Împrăștiat cu Salt Adaptiv în Frecvență), denumită pe scurt AFH, ce presupune transmiterea datelor pe mai multe canale prin comutare rapidă pe baza unei secvențe pseudo-aleatorie. Selectarea canalului de comunicație permite Bluetooth să se adapteze mediului prin identificarea canalelor afectate de surse fixe de interferență și excluderea acestora din lista celor disponibile.

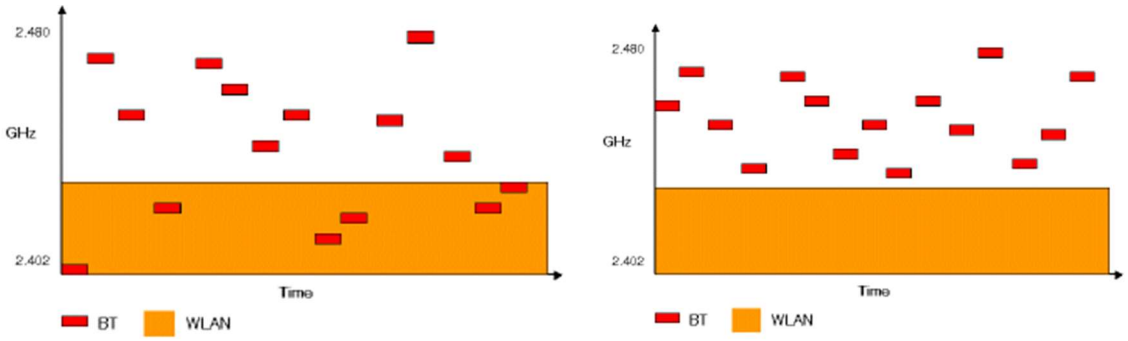

Figura 50. Utilizarea canalelor cu și fără AFH

În cazul versiunii Low Energy sunt utilizate 40 de canale (Figura 50) fiecare având lățimea de bandă de 2 MHz. 37 dintre acestea sunt dedicate transferului de date, iar 3 sunt folosite pentru *advertising* (descoperirea dispozitivelor, stabilirea conexiunii, transmisii de tip *broadcast*).

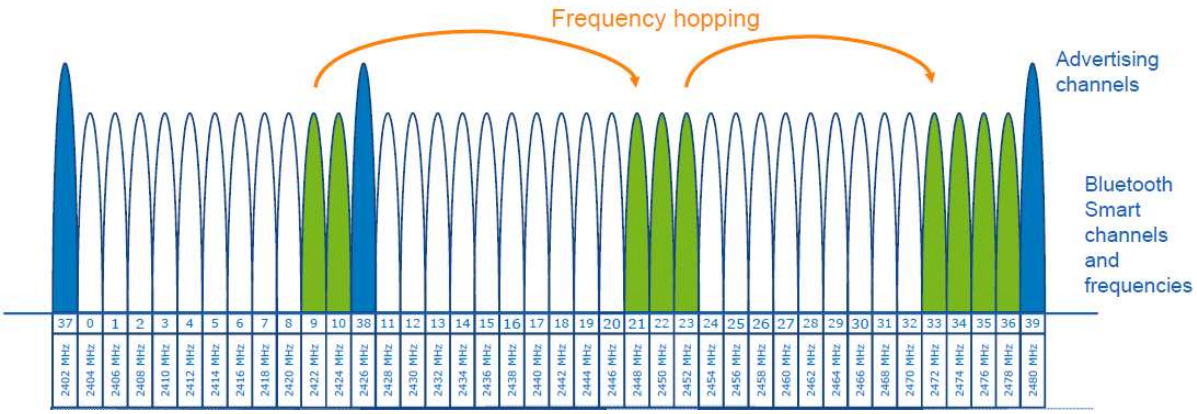

Figura 51. Canalele utilizate de Bluetooth LE

Raza de comunicație a modulelor Bluetooth depinde de clasa în care sunt acestea încadrate:

- Clasa 1, putere de emisie 100 mW, raza de comunicație aprox. 100 m.
- Clasa 2, putere de emisie 2,5 mW, raza de comunicație aprox. 10 m.
- Clasa 3, putere de emisie 1 mW, raza de comunicație aprox. 1 m.

Rețeaua de bază pentru Bluetooth se numește *Piconet* și are până la 8 dispozitive active într-o relație *master-slave*, adică 1 *master* și 7 *slave*. Un dispozitiv *slave* poate comunica doar cu dispozitivul *master* și numai atunci când este permisă de acesta. *Master*-ul determină alegerea canalului, adică secvența de salt în frecvență care trebuie utilizată de toate dispozitivele slave din *Piconet*.

Structura unui *Piconet* este prezentată în Figura 50. Dispozitivele sunt conectate într-un mod ad-hoc, iar desemnarea dispozitivelor *master* sau *slave* va fi valabilă pe toată durata de viață a *Piconet*-ului. Fiecare *Piconet* are un model unic de salt în frecvență, determinat de dispozitivul *master*, iar dispozitivele *slave* trebuie să se sincronizeze cu acesta.

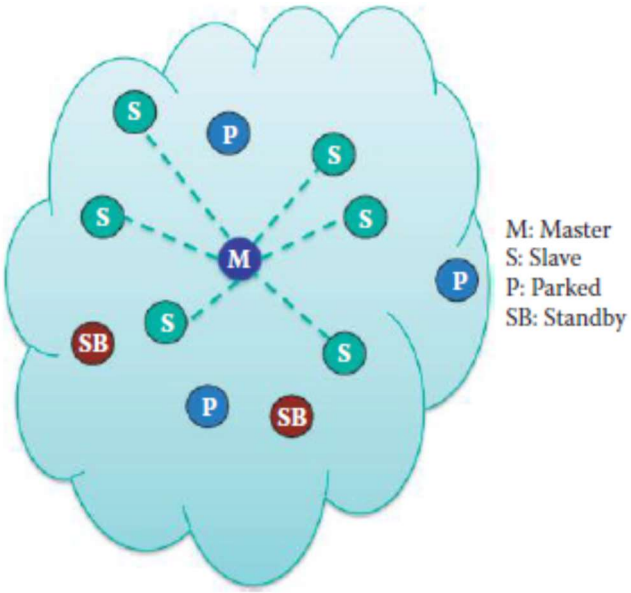

Figura 52. Structura unei rețele *Piconet*

Dispozitivele *slave* pot fi în una din cele trei stări majore: Așteptare, Conexiune și Parcat. Starea de Așteptare este starea implicită a dispozitivului. În această stare, dispozitivul poate fi într-un mod de consum redus de energie. În starea Conexiune, conexiunea a fost stabilită și pachetele pot fi trimise înainte și înapoi. În starea Parcat, dispozitivul *slave* va fi inactiv până când *master*-ul îl va reactiva. Starea Parcat nu mai este disponibilă începând cu versiunea 5.

6.3 Rețele ZigBee

Rețelele ZigBee sunt rețele personale (PAN) cu rate de transfer scăzute bazate pe standardul IEEE 802.15.4, destinate aplicațiilor industriale, rezidențiale și medicale ce necesită costuri și consum redus de energie, precum și cerințe reduse privind rata de transfer sau QoS.

Numele ZigBee se pare că provine de la zborul în zig-zag al albinelor care își transmit informații referitoare la poziția sursei de hrană.

Prima apariție a acestei tehnologii a fost în 1998, ca urmare a nevoii de o interfață mai ieftină decât Bluetooth pentru aplicații cu mulți senzori în care rețeaua se auto-configurează la intrarea sau ieșirea din activitate a unora dintre aceștia.

Modulele ZigBee pot lucra în modul punct la punct sau punct la multipunct, o rețea de astfel de dispozitive necesitând un dispozitiv cu funcția de Coordonator. Rețeaua de tip *mesh* permite conexiuni de date între dispozitive mai îndepărtate decât raza de acțiune radio prin interpunerea unor noduri ZigBee intermediare, iar defectarea unui nod poate fi transparentă prin preluarea sarcinilor de către alt nod.

Banda alocată în Europa este cea de 2,4GHz, intervalul de bandă folosit fiind între 2,405GHz și 2,480GHz, împărțit în 16 canale fiecare cu o lățime de bandă de 5MHz (Figura 53). Canalele sunt numerotate începând cu canalul 11 până la 26. Canalele ZigBee nu se suprapun, însă, în Figura 53 se poate observa suprapunerea acestora cu cele ale tehnologiilor

Wi-Fi și Bluetooth, lucru care poate duce la probleme de coexistență datorită interferențelor create de acestea.

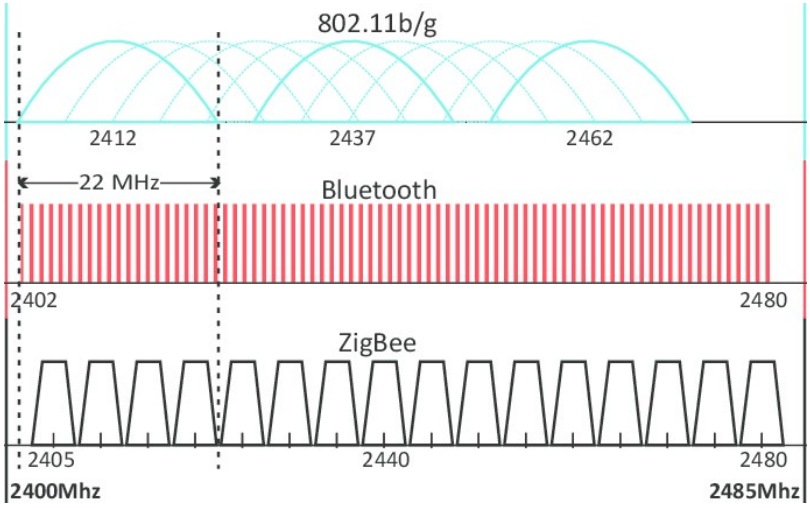

Figura 53. Canalele utilizate de ZigBee

Modulele ZigBee uzuale asigură o rată de transfer de 250kbps la distanțe de maximum 100m în spații închise și 1,6km în spații fără obstacole, viteze posibile fiind între 1200bps și 1Mbps. Comunicarea radio poate fi criptată (AES) iar corectitudinea transmisiei este asigurată de un mecanism de confirmare (ACK) și reîncercare.

Metoda de acces la mediu este numită Acces Multiplu cu Detecția Purtătoarei și Evitarea Coliziunilor (CSMA/CA) pentru a determina când un dispozitiv este liber să transmită date prin mediul de comunicație, iar pentru transmiterea datelor se utilizează tehnica de tip Spectru împrăștiat cu secvență directă (DSSS).

O rețea ZigBee poate cuprinde trei tipuri de componente, care îndeplinesc funcționalități specifice:

1. Coordonator
    - Orice rețea ZigBee are nevoie în orice moment de timp de un Coordonator. Acesta trebuie să fie unic la nivel de rețea.
    - Un Coordonator este responsabil cu formarea topologiei logice a rețelei, oferirea și configurarea adreselor și managementul altor funcționalități ce definesc rețeaua, o securizează și o păstrează funcțională pentru intervale lungi de timp.
    - Rolul unui Coordonator poate fi asumat de către un alt nod, în cazul în care acesta devine indisponibil.
    - Se recomandă să aibă o sursă de energie cât mai sigură, pentru a nu pune în pericol funcționarea rețelei. De asemenea, Coordonator se alege și nodul care are cele mai bune resurse hardware, pentru a putea face față unui număr mare de conexiuni active în orice moment de timp.
2. Router
    - O rețea poate avea mai multe noduri care să îndeplinească funcționalitatea de *Router*.

- Se poate conecta în mod activ la o rețea (nu are deci nevoie de un administrator care să-i spună să se conecteze) sau poate cere să se alăture unei rețele, dacă aceasta este securizată.
- Poate trimite și primi informații de la alte noduri, dar cel mai important – poate ruta informațiile primite. Acest lucru înseamnă că dacă un mesaj a ajuns la *Router* dar nu îi este destinat acestuia, el se poate ocupa cu trimiterea mai departe a informațiilor, către nodul corespunzător, cu ajutorul tabelei de rutare.
- *Router*-ele trebuie de asemenea să aibă o sursă de alimentare constantă, pentru a nu se apărea discontinuități în procesul de comunicare.

3. End-device (dispozitiv terminal)
    - *End-device*-urile pot să se alăture în mod pasiv rețelelor din jurul său și sunt capabile să trimită și să primească informații.
    - Un *End-device* are nevoie în orice moment de timp de un *Router* sau un Coordonator la care să se conecteze.
    - Un *End-device* nu se poate conecta în mod activ într-o rețea și depinde de existența unui ”părinte” – acesta îi ajută să se conecteze la rețea și de asemenea stochează mesaje când *End–device*-urile sunt în stand-by.
    - O rețea poate conține un număr foarte mare de *End-device*-uri, până la 65536.
    - La un *End-device*, de cele mai multe ori, se vor conecta senzori – direct sau indirect.

În cadrul rețelelor ZigBee pot exista, în principiu, patru tipuri de topologii (Figura 54):

- Pereche (*pair*), reprezintă cea mai simplă modalitate de interconectare a oricare două noduri, unul dintre ele fiind Coordonator. Această topologie este ideală pentru scenariile simple și pentru detectarea și rezolvarea problemelor dintr-o topologie mai mare – prin izolarea pe rând a nodurilor.
- Stea (*star*), în care un nod ce îndeplinește funcția de Coordonator se află în centrul topologiei, iar el se conectează un număr de *End-device*-uri. Orice mesaj trimis în cadrul acestui tip de rețea trebuie să treacă, în mod obligatoriu, prin Coordonator care se ocupă cu rutarea lor în mod corespunzător către destinația finală.
- Plasă (*mesh*), în care sunt prezente în mod concomitent toate cele trei tipuri de componente – Coordonator, *Router* și *End-device*. Rețelele ce sunt construite pe baza acestei topologii au un grad ridicat de redundanță deoarece dacă apare o problemă cu o cale de comunicare, există întotdeauna una redundantă.
- Arbore (*cluster-tree*), *Router*-ele formează o formă de ”*backbone*” a rețelei și se conectează doar la Coordonator. Nu mai există niciun fel de

redundanță a rețelei, iar defecțiunea unui router, aduce o dată cu sine, imposibilitatea de comunicare a anumitor *End-device*-uri.

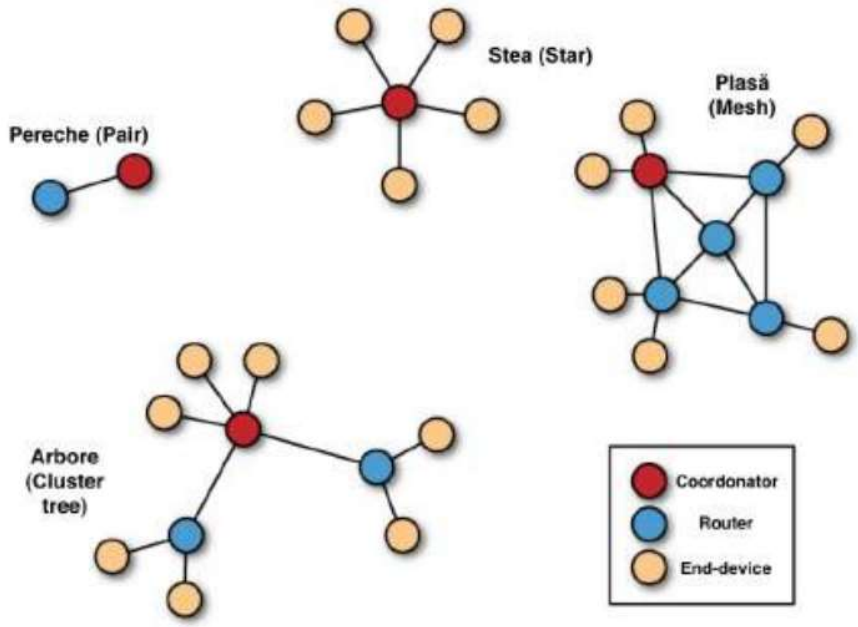

Figura 54. Tipuri de topologii pentru rețele ZigBee

6.4 Rețele WiMAX

Standardul IEEE 802.16 "*Air Interface for Fixed Broadband Wireless Access Systems*", cunoscut și ca WiMAX, a fost proiectat să ofere acces fără fir, de bandă largă, în rețele MAN cu performanțe comparabile cu cablul tradițional.

Avantajele sistemelor bazate pe 802.16 sunt multiple: abilitatea de a porni rapid acest serviciu chiar și în zone unde ar fi greu de ajuns cu interfețe pe bază de cablu, evitarea costurilor mari de instalare, și posibilitatea de a depăși limitările fizice ale infrastructurilor tradiționale cu conexiune prin fir.

Lansat în anul 2001, standardul 802.16 era dedicat rețelelor punct-la-multipunct bazate pe LoS (linie de vizibilitate directă) inițial în banda licențiată 10-66 GHz, apoi, prin emiterea unor amendamente, și în benzile licențiate sau nelicențiate 2-11 GHz chiar și cu capabilități de tip non-LoS. În situațiile de tip LoS se puteau obține rate de transfer de 10 Mbps până la 10 km, și până la 2 km pentru cele de tip non-LoS.

În mod frecvent se folosesc benzile de 2,5 GHz, 3,5 GHz și 5,8 GHz, cu lățime de bandă flexibilă a canalului, între 1,25 MHz și 20 MHz.

Amendamentul 802.16e a permis utilizarea stațiilor mobile la viteze vehiculare precum și utilizarea de tehnici MIMO pentru îmbunătățirea razei de acoperire. Standardul 802.16m a permis mărirea ratelor de transfer până la 100 Mbps pentru stațiile mobile și 1 Gbps pentru cele fixe, distanța de comunicație atingând 50 km.

WiMAX, însă, nu poate livra 70 Mbps pe o distanță de 50 km. Ca toate tehnologiile wireless, WiMAX poate funcționa cu rate de transfer mai mari sau pe distanțe mai mari, dar nu și în ambele situații simultan. Operarea la o rază maximă de 50 km crește rata de eroare a biților

și are ca rezultat o rată de transfer mult scăzută. În cazul reducerii distanței (sub 1 km) se permite unui dispozitiv să funcționeze la rate de transfer mari.

Pentru accesul la mediu se utilizează tehnica CSMA/CA iar pentru transmiterea datelor se folosesc atât transmisia duplex cu diviziune în timp (TDD) cât și cu diviziune în frecvență (FDD). Diviziunea presupune utilizarea unui canal de comunicație *half-duplex* (cum este aerul) pentru a emula o comunicație *full-duplex* prin utilizarea alternativă a unor sloturi de timp, în cazul TDD, sau a unor frecvențe purtătoare diferite, în cazul FDD, pentru transmisie și recepție.

O arhitectură generală a unei rețele WiMAX este prezentată în figura 55. Elementele din cadrul acestei rețele sunt stațiile de bază (BS), stațiile fixe (SS), stațiile mobile (MS) și o poartă către rețeaua de acces la servicii (ASN-GW), care conectează rețeaua WiMAX la o rețea *backbone*, care la rândul său se conectează la Internet. Un turn WiMAX este capabil să acopere o suprafață de 8.000 de kilometri pătrați.

Stația de bază WiMAX permite comunicația punct-la-multipunct, utilizând fie o antenă omnidirecțională, fie una sectorială. Cu toate acestea, comunicațiile între stațiile de bază sunt realizate utilizând o antenă punct-la-punct.

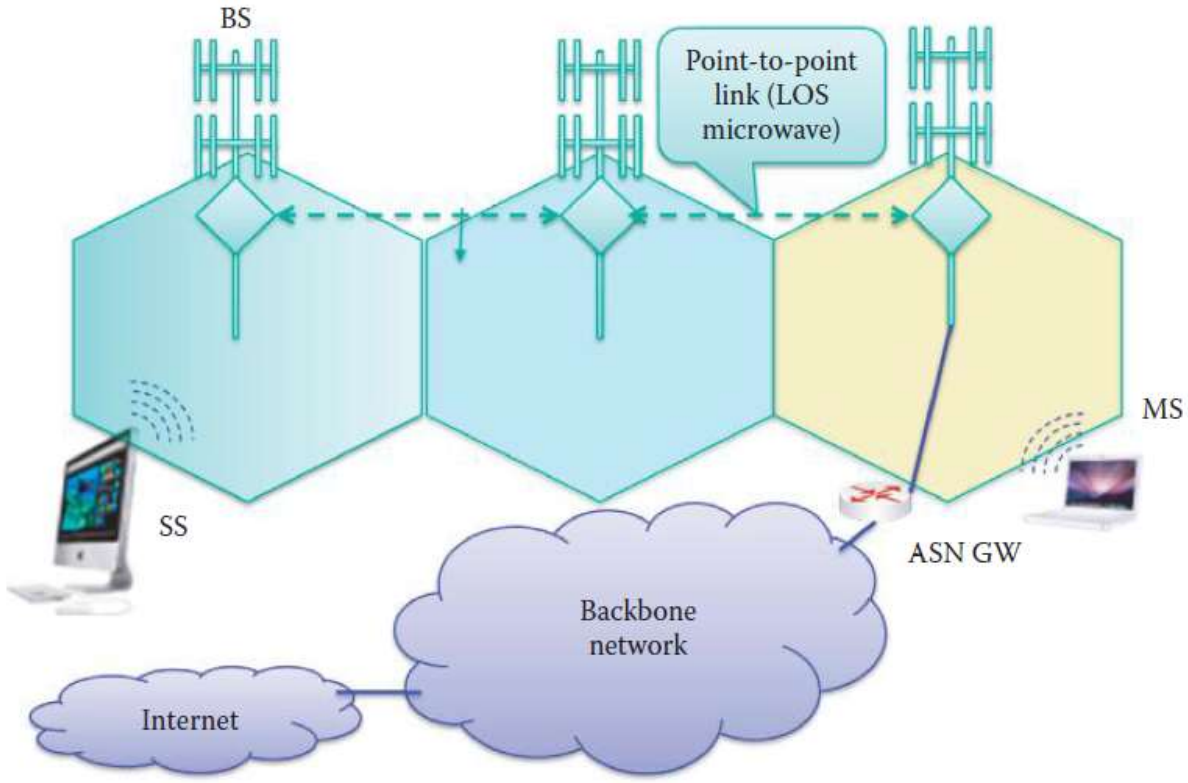

Figura 55. Arhitectura generală a unei rețele WiMAX

6.5 Rețele GSM

GSM (Sistemul Global de comunicații Mobile) este un standard elaborat de Institutul European de Standardizare în Telecomunicații (ETSI) pentru a descrie protocoalele pentru rețelele celulare de a doua generație utilizate de dispozitive mobile.

6.5.1 Generația 2G

Rețelele de generație 2G au fost dezvoltate pentru a înlocui rețelele celulare analogice de primă generație (1G), iar standardul GSM inițial era descris ca o rețea digitală cu comutare de circuite (*circuit switching*), optimizată pentru telefonia vocală *full-duplex*. Pentru aceasta se utiliza tehnica accesului multiplu prin diviziunea timpului (TDMA) care acordă diferiților utilizatori sloturi diferite de timp pe un canal. Serviciile de date cuprindeau transmiterea de SMS-uri (mesaje text) și MMS (mesaje multimedia).

Tehnologia a evoluat în timp pentru a include comunicațiile de date, mai întâi prin comutare de circuite, apoi prin comutare de pachete (*packet switching*) prin GPRS (generația 2,5G) și EDGE (generația 2,75G). GPRS oferă o rată maximă de transfer de 50 kbps, iar EDGE de maxim 1 Mbps.

Cele mai utilizate benzi de frecvență în Europa sunt 900 MHz și 1800 MHz.

6.5.2 Generația 3G

Sistemele de telefonie mobilă de generația a treia (3G) utilizează în general o formă de Acces multiplu prin diviziune în cod (CDMA). Fiecare stație transmite în mod constant folosind întreg spectrul de frecvențe. Un canal de transmisie poate transporta mai multe semnale de la diferiți utilizatori în același timp, fără interferențe între utilizatori, deoarece diferiților utilizatori li se alocă coduri diferite pentru a asigura accesul la sistem. Semnalul care transporta informația este multiplicat cu un alt semnal care este mai rapid și are o lărgime de bandă mai mare - o secvență de pseudo-zgomot (PN). Semnalul rezultat, amestecat, se aseamănă foarte mult cu un semnal de zgomot. Receptorul extrage informațiile folosind aceeași secvență PN ca și transmițătorul. Semnalele de la diferiți utilizatori se disting prin utilizarea de diferite secvențe PN. CDMA se bazează pe metoda de transmitere DSSS.

CDMA cu bandă largă (WCDMA) este o variantă a CDMA care poate suporta comunicații multimedia la viteze mai mari decât au fost posibile anterior (până la 384 kbps). Acesta este utilizat în rețelele 3G de tip UMTS.

HSDPA și HSUPA sunt adăugiri la infrastructura 3G standard care permit obținerea de rate de transfer mai mari, până la 14,4 Mbps pentru *download* și 5,76 Mbps pentru *upload*.

O altă îmbunătățire a fost adusă prin standardul HSPA+ ce permite rate de transfer de până la 42,2 Mbps pentru *download* și 22 Mbps pentru *upload*.

Cele mai utilizate benzi de frecvență în Europa sunt 900 MHz și 2100 MHz.

6.5.3 Generația 4G

A patra generație de sisteme de telefonie mobilă (4G) oferă rate de transfer de peste 100 Mbps pentru *download* și peste 50 Mbps pentru *upload*, putând fi astfel sprijinite servicii și aplicații avansate, precum televiziune interactivă, bloguri video mobile și jocuri *multiplayer* avansate.

Tehnologiile LTE-Advanced și WirelessMAN-Advanced (parte din standardul WiMAX 2) sunt ambele standarde 4G oficiale.

LTE poate utiliza un număr mare de benzi de frecvență diferite, ceea ce ajută la transformarea acesteia într-o tehnologie foarte flexibilă. Acest lucru este necesar deoarece aceleași zone de spectru radio nu sunt disponibile în întreaga lume. În plus, LTE acceptă atât modurile de duplex cu divizare în frecvență (FDD), cât și modurile duplex cu divizare în timp (TDD).

Metoda de modulare utilizată în LTE este OFDM, datele de mare viteză care sunt transmise fiind împărțite în mai multe semnale de viteză mai mică, în benzi de frecvență înguste (numite subpurtătoare). Aceasta înseamnă că se pierde puțin spectru radio. Prin proiectare, semnalele subpurtătoarelor sunt independente unul de altul (ortogonale) și astfel produc un semnal global care este foarte puțin afectat de interferențe. Procesoare puternice sunt utilizate pentru a efectua calculele transformatei Fourier rapide care sunt necesare pentru a separa subpurtătoarele.

Ratele de transfer mari se obțin și prin utilizarea tehnologiilor MIMO, creșterea densității stațiilor de bază sau utilizarea de frecvențe înalte.

Cele mai utilizate benzi de frecvență în Europa sunt 800 MHz, 900MHz, 1800Mhz, 2100 MHz și 2600 MHz.

6.5.4 Generația 5G

A cincea generație de sisteme de telefonie mobilă (5G) urmează a fi lansată în anul 2018 și promite rate de transfer de până la 20 Gbps, utilizând benzi de frecvență înaltă de până la 6 GHz precum și tehnologii MIMO masive cu 64 până la 256 de antene.

6.5.5 Arhitectura generală

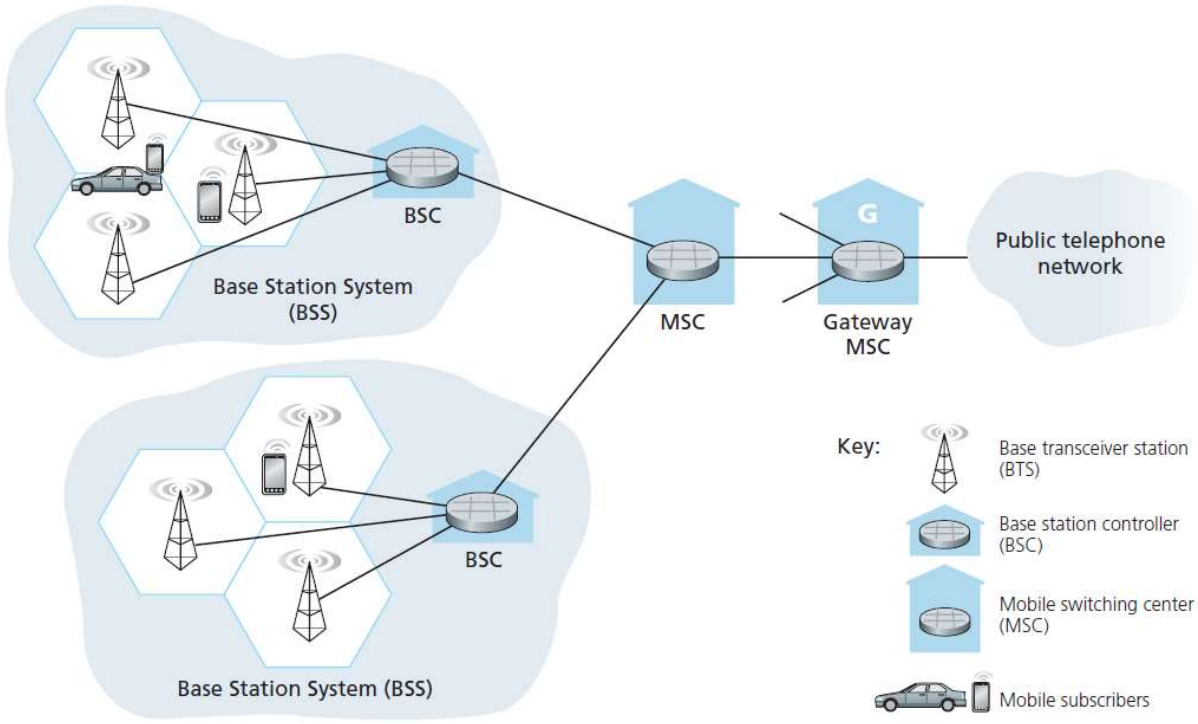

Figura 56. Rețea de telefonie mobilă

Rețelele de telefonie mobilă sunt rețele celulare. După cum se poate vedea în Figura 56, **unitatea mobilă** (fie că este vorba de un telefon sau de un calculator) comunică cu o **stație de bază**. Stația de bază este echivalentul punctului de acces dintr-o rețea WLAN și constă dintr-un transceiver (emițător/receptor) și un controler al stației de bază. Zona de acoperire a unei stații de bază se numește **celulă**. În realitate, celulele sunt oarecum dezordonate, dar în arhitecturile de rețea sunt de obicei prezentate ca fiind hexagonale sau circulare. În centrele orașelor, celulele sunt mult mai mici (datorită densității populației pe care trebuie să o servească) decât celulele din localitățile rurale.

Există numeroase stații de bază într-o rețea celulară. Legături prin fibră optică sau fără fir de tip punct-la-punct conectează stațiile de bază la un **Centru de comutare al serviciilor mobile** (MSC). MSC conectează apelurile de pe telefoanele fixe către telefoanele mobile și comută apelurile între celule, pe măsură ce dispozitivele mobile se mișcă dintr-o celulă în alta (*handover*).

Aceeași frecvență radio poate fi utilizată în mai multe celule, atâta timp cât aceste celule nu sunt una lângă alta. Reutilizarea frecvențelor în acest mod mărește capacitatea rețelei fără a provoca interferențe între celule.

Capitolul 7. Multimedia

Aplicațiile multimedia sunt aplicații care presupun utilizarea de fluxuri video și audio. Acestea pot fi clasificate fie ca fluxuri audio/video stocate, conversații prin voce/video-over-IP, sau fluxuri audio/video în timp real (*live*). Fiecare clasă de aplicații are propriul său set unic de cerințe privind serviciul și de probleme de proiectare.

7.1 Proprietățile unui clip video

Caracteristica cea mai importantă a unui clip video este rata de transmisie a biților ridicată (*bitrate* - numărul de biți utilizați per unitate de timp de redare, de obicei secundă). Un clip video distribuit pe Internet conține de obicei de la 100 kbps pentru videoconferințe de calitate scăzută, la 2 Mbps pentru vizionarea de filme cu definiție standard, 5 Mbps pentru filmele HD și 9 Mbps pentru cele UHD.

O altă caracteristică importantă a unui videoclip este aceea că poate fi comprimat, prin urmare, se poate reduce calitatea și mări *bitrate*-ul. Un videoclip este o secvență de imagini, de obicei fiind afișat la o rată constantă, de exemplu, de 24 sau 30 imagini pe secundă. O imagine necomprimată, codificată digital, constă dintr-o matrice de pixeli, fiecare pixel fiind codificat într-un număr de biți pentru a reprezenta luminanța și culoarea. Există două tipuri de redundanță în clipurile video, ambele putând fi exploatate prin compresie:

- Redundanța spațială este redundanța dintr-o imagine dată. Intuitiv, o imagine care constă din spațiu în cea mai mare parte alb are un grad ridicat de redundanță și poate fi comprimată eficient fără a sacrifica în mod semnificativ calitatea ei.
- Redundanța temporală reflectă repetarea pixelilor de la o imagine curentă la cea ulterioară. Dacă, de exemplu, imaginea curentă și cea ulterioară sunt exact aceleași, nu există niciun motiv pentru recodarea imaginii ulterioare; este mai eficientă indicarea în timpul codificării că imaginea ulterioară este exact aceeași.

Compresia poate fi utilizată pentru a crea mai multe versiuni ale aceluiași videoclip, fiecare la un nivel de calitate diferit. De exemplu, se poate folosi compresia pentru a crea, să zicem, trei versiuni ale aceluiași videoclip, la rate de 300 kbps, 1 Mbps și 3 Mbps. Utilizatorii pot decide apoi ce versiune doresc să vizioneze în funcție de lățimea de bandă disponibilă.

Două dintre cele mai utilizate metode de compresie sunt MPEG-2 și H.264.

7.2 Proprietățile unui clip audio

Clipurile audio (inclusiv vorbirea și muzica digitalizate) au cerințe de lățime de bandă semnificativ mai reduse decât clipurile video.

O caracteristică importantă este necesitatea conversiei semnalului audio analogic în unul digital ce presupune următoarele:

- Semnalul analogic este eșantionat cu o anumită rată, măsurată în eșantioane pe secundă (*sample per second*). Valoarea fiecărui eșantion este un număr real arbitrar.

- Fiecare dintre eșantioane este apoi rotunjit la o valoare dintr-un set finit. Această operație este denumită **cuantificare**. Numărul acestor valori finite - numite valori de cuantificare - provine de obicei din puterea lui 2.
- Fiecare dintre valorile de cuantificare este reprezentată de un număr fix de biți.

Reprezentările în biți ale tuturor eșantioanelor sunt concatenate împreună pentru a forma reprezentarea digitală a semnalului. De exemplu, dacă un semnal audio analog este prelevat la 8.000 de eșantioane pe secundă și fiecare eșantion este cuantizat și reprezentat pe 8 biți, atunci semnalul digital rezultat va avea o rată de 64.000 de biți pe secundă.

Pentru redarea prin difuzoare audio, semnalul digital poate fi transformat înapoi, adică decodificat, într-un semnal analogic. Cu toate acestea, semnalul analogic decodificat este doar o aproximare a semnalului original, iar calitatea sunetului poate fi considerabil degradată.

Această tehnică de codare de bază se numește modulare prin codificarea impulsurilor (PCM). Codarea vocală folosește adesea PCM, cu o rată de 8.000 de eșantioane pe secundă și 8 biți per eșantion, rezultând o rată de 64 kbps. CD-ul audio utilizează, de asemenea, PCM, cu o rată de eșantionare de 44,100 de eșantioane pe secundă cu 16 biți per eșantion; acest lucru oferă o rată de 705,6 kbps pentru înregistrările mono canal și 1,411 Mbps pentru cele stereo.

Cu toate acestea, vorbirea și muzica codificate PCM sunt rareori utilizate în Internet. În schimb, ca și în cazul clipurilor video, tehnicile de compresie sunt utilizate pentru a reduce ratele de biți ale fluxului. Discursul uman poate fi comprimat la mai puțin de 10 kbps și poate fi în continuare inteligibil. O tehnică populară de compresie pentru muzica stereo apropiată de calitatea CD-urilor este MPEG 1 layer 3, cunoscut sub numele de MP3. Pentru aceasta, 128 kbps este cea mai obișnuită rată de codare și produce o degradare foarte mică a sunetului. Un standard similar este Advanced Audio Coding (AAC), popularizat de Apple.

7.3 Transmiterea de fluxuri audio/video stocate

Transmiterea unui flux (*streaming*) video stocat presupune combinarea de componente video și audio. Transmiterea unui flux audio stocat este similară, deși rata biților este de obicei mult mai mică.

În această clasă de aplicații, mediul de bază este un videoclip preînregistrat plasat pe un server, iar utilizatorii trimit solicitări către acesta pentru a-l vizualiza, la cerere (*on demand*). Multe companii de internet oferă astăzi video *streaming*, inclusiv YouTube sau Netflix.

Transmiterea unui flux video stocat are trei caracteristici cheie:

- Transmiterea sub formă de flux. În acest mod aplicația client începe, de obicei, redarea video în câteva secunde după ce începe să primească videoclipul de la server. Acest lucru înseamnă că aplicația client va fi reda o anumită parte din videoclip, în același timp primind alte părți ale videoclipului de pe server. Această tehnică, cunoscută sub numele de *streaming*, evită descărcarea întregului fișier video înainte de începerea redării.
- Interactivitatea. Deoarece clipul video este preînregistrat, utilizatorul poate întrerupe, repoziționa înainte, repoziționa înapoi, derula rapid înainte și așa mai

departe, prin conținutul video. Timpul de la momentul în care utilizatorul face o astfel de cerere până când acțiunea are loc la client ar trebui să fie mai mic de câteva secunde pentru o reacție acceptabilă.

- Derulare continuă. Odată ce începe redarea videoclipului, derularea se va desfășura în timp conform înregistrării. Prin urmare, datele trebuie să fie primite de la server în timp util pentru redarea videoclipului de către aplicația client; în caz contrar, utilizatorii se vor confrunta cu înghețarea cadrului video (atunci când clientul așteaptă cadrele întârziate) sau omiterea unor cadre (când clientul trece peste cadrele întârziate).

De departe, cea mai importantă măsură de performanță a unei rețele pentru *streaming* video este rata de transfer medie. Pentru a asigura redarea continuă, aceasta trebuie să furnizeze o rată medie pentru aplicația de *streaming* care să fie cel puțin egală cu rata de biți a videoclipului în sine. Prin utilizarea de memorii tampon (*buffere*) și a pre-descărcare (*prefetching*), este posibilă asigurarea redării continue chiar și atunci când rata de transfer variază, atâta timp cât rata medie (de-a lungul a 5-10 secunde) rămâne peste rata de înregistrare a clipului video.

7.3.1 Tehnica buffering

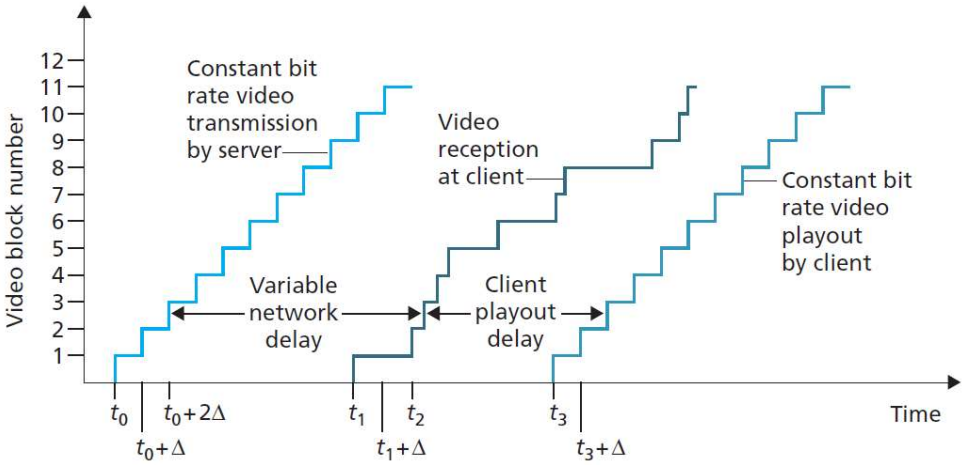

Figura 57. Întârzierea derulării unui flux video

Această tehnică este implementată la nivelul aplicației clientul pentru a atenua efectele diferitelor întârzieri sau ale variației lățimii de bandă disponibilă între server și client. Pentru *streaming* video (atât stocate, cât și în timp real) utilizatorii pot tolera în general o întârziere inițială de câteva secunde între momentul când clientul solicită un videoclip și când începe redarea acestuia la client. În consecință, atunci când videoclipul începe să sosească la client, acesta nu va începe imediat redarea, ci va construi o rezervă de flux video într-o memorie buffer a aplicației. Odată ce clientul a creat o rezervă de câteva secunde de video neredat încă, acesta poate începe redarea video.

Există două avantaje importante oferite de o astfel de tehnică. Mai întâi, se pot absorbi variații datorate întârzierilor dintre server și client. Dacă o anumită parte de date video întârzie să sosească, atâta timp cât aceasta ajunge înainte de epuizarea rezervelor video, această

întârziere lungă nu va fi observată. În al doilea rând, dacă lățimea de bandă a legăturii server-client scade pentru scurt timp sub rata de biți a clipului video, un utilizator poate continua să se bucure de redarea continuă, atât timp cât buffer-ul aplicației client nu devine complet gol.

7.3.2 Tehnica prefetching

Așa cum s-a văzut anterior, în cazul tehnicii *buffering*, serverul transmite un clip video cu rata necesară pentru ca acesta să poată fi vizualizat. Cu toate acestea, pentru transmiterea de fluxuri video stocate, clientul poate încerca să descarce un clip video la o rată mai mare decât rata de biți a clipului video, pre-descărcând astfel cadre video care urmează să fie afișate în viitor. Această parte din videoclip este în mod natural stocată în *buffer*-ul aplicației client.

O astfel de tehnică este aplicată în mod natural când transmiterea fluxului video se face prin TCP, deoarece mecanismul de evitare a congestiei inclus în protocol va încerca să utilizeze întreaga lățime de bandă disponibilă între server și client.

7.4 Transmiterea de conversații prin voce/video-over-IP

Transmiterea de voce în timp real pe Internet este adesea numită telefonie prin Internet, deoarece, din perspectiva utilizatorului, este similară cu serviciul telefonic tradițional. Este, de asemenea, numită în mod obișnuit Voice-over-IP (VoIP). Conversația video este similară în multe privințe cu VoIP, cu excepția faptului că include imaginea video a participanților, precum și vocile acestora.

Protocolul de bază al nivelului Rețea pentru Internet, IP, oferă servicii de tip *best-effort* (cel mai bun efort). Adică serviciul depune toate eforturile pentru a muta fiecare datagramă de la sursă la destinație cât mai repede posibil, dar nu face nici o promisiune în ceea ce privește livrarea pachetului la destinație în anumite limite de timp, sau în legătură cu o limită a procentajului de pachete pierdute. Lipsa unor astfel de garanții reprezintă provocări semnificative pentru proiectarea aplicațiilor de voce în timp real, care sunt extrem de sensibile la întârzierea pachetelor, jitter și pierdere.

7.4.1 Pierderea de pachete

Un segment UDP generat de aplicația VoIP este încapsulat într-o datagramă IP. Pe măsură ce datagrama circulă prin rețea către destinație, aceasta trece prin memorii *buffer ale* router-elor (adică așteaptă în cozi). Este posibil ca unul sau mai multe buffere din calea parcursă să fie pline, caz în care datagrama IP se va pierde și nu va mai ajunge niciodată la aplicația destinație.

Pierderea de pachete ar putea fi eliminată prin trimiterea pachetelor pe TCP (care asigură transferul de date fiabil), mai degrabă decât peste UDP. Cu toate acestea, mecanismele de retransmisie sunt adesea considerate inacceptabile pentru aplicațiile audio în timp real, cum ar fi VoIP, deoarece ele măresc întârzierea capăt-la-capăt. Mai mult, datorită controlului congestiei TCP, pierderea de pachete poate duce la o reducere a ratei de transmisie a expeditorului TCP ce poate determina o golire a buffer-ului destinație, iar acest lucru poate avea

un impact sever asupra inteligibilității vocii. Din aceste motive, majoritatea aplicațiilor VoIP existente rulează în mod implicit prin UDP.

Pierderea de pachete nu este neapărat dezastruoasă, ratele de pierdere între 1 și 20% putând fi tolerate, în funcție de modul în care vocea este codată și transmisă și de modul în care pierderea este ascunsă la receptor (de exemplu prin corecția de eroare de tip FEC). Prin FEC, informații redundante (de exemplu secvențe cu o rată de bit scăzută – Figura 58) sunt transmise împreună cu informațiile originale, astfel încât unele dintre datele originale pierdute pot fi recuperate din acestea.

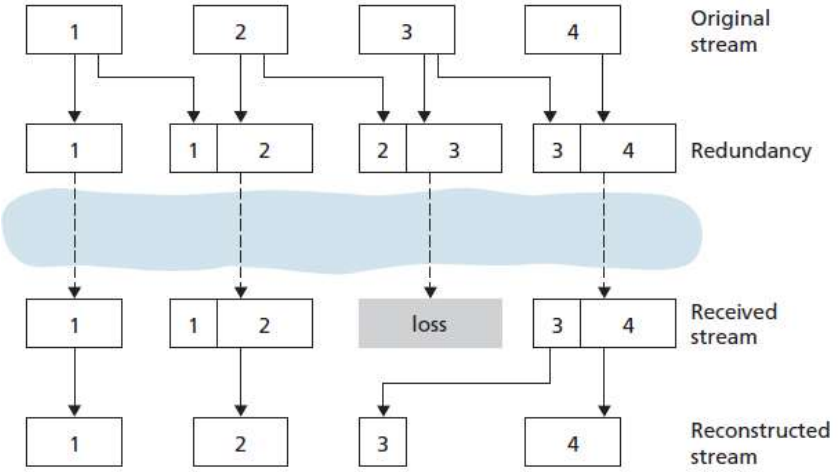

Figura 58. Corecția erorilor prin utilizarea de informații redundante

O altă metodă atenuare a efectului pierderii de pachete este intercalarea (*interleaving*) prin care expeditorul re-aranjează secvențele de date audio înainte de transmisie, astfel încât unități adiacente inițial sunt separate de o anumită distanță în fluxul transmis (Figura 59).

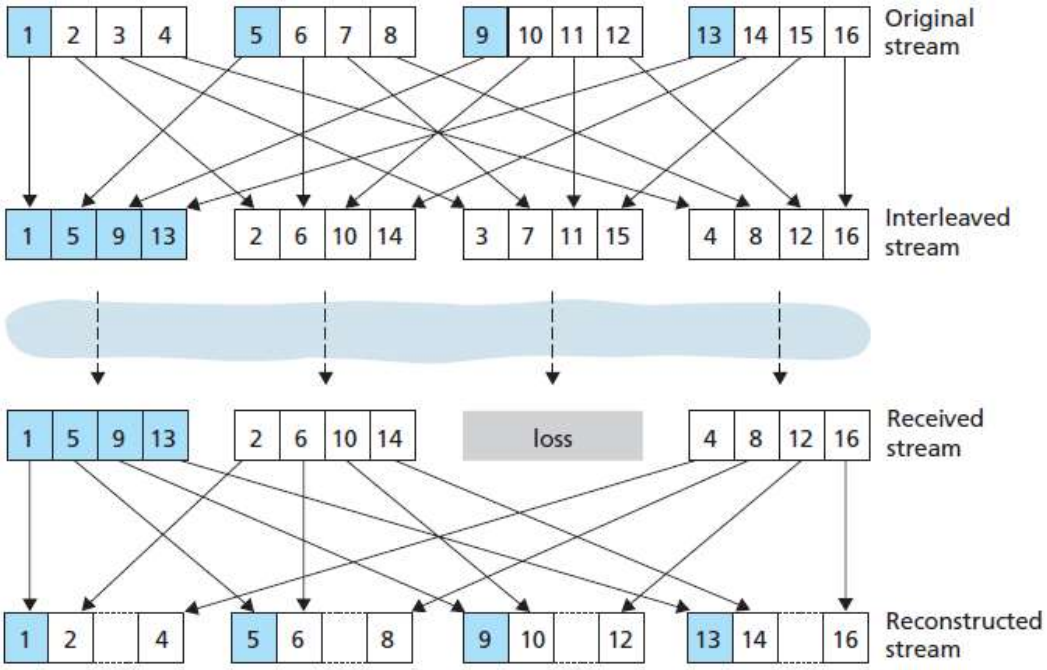

Figura 59. Transmiterea intercalată a unui flux audio

Cu toate acestea, dacă una sau mai multe legături dintre expeditor și receptor sunt puternic aglomerate, iar pierderea pachetelor depășește 10-20% (de exemplu, pe o conexiune

fără fir), atunci nu se poate face nimic pentru a obține o calitate audio acceptabilă. În mod clar, serviciul cu cel mai bun efort are limitările sale.

7.4.2 Întârzierea capăt-la-capăt

Reprezintă acumularea de întârzieri prin transmisie, prelucrare și așteptare la nivelul router-elor; întârzieri de propagare în legături; întârzieri în procesarea la nivelul sistemului final. Pentru aplicațiile conversaționale în timp real, cum ar fi VoIP, dacă aceste întârzieri cumulate sunt mai mici de 150 ms ele nu sunt percepute de un ascultător uman; întârzierile între 150 și 400 ms pot fi acceptabile, dar nu sunt ideale; iar întârzierile de peste 400 ms pot împiedica serios interactivitatea în conversațiile vocale. Aplicațiile VoIP de pe calculatoarele de destinație vor ignora de obicei orice pachete care sunt întârziate mai mult decât un anumit prag, de exemplu, mai mult de 400 ms, acestea fiind efectiv pierdute.

7.4.3 Jitter-ul

O componentă crucială a întârzierii capăt-la-capăt este *întârzierea variabilă* datorată timpilor de așteptare diferiți pe care un pachet îi are în router-ele rețelei. Din cauza acestor întârzieri diferite, timpul scurs de la generarea unui pachet până când acesta ajunge la receptor poate fluctua de la pachet la pachet. Acest fenomen se numește *jitter*.

Dacă receptorul ignoră prezența *jitter*-ului și redă secvențe din fluxul audio imediat ce acestea ajung, atunci fluxul rezultat poate deveni cu ușurință neinteligibil la receptor. Din fericire, *jitter*-ul poate fi deseori eliminat prin utilizarea numerotării secvențelor, a marcajelor de timp și a întârzierilor în redare.

7.5 Transmiterea de fluxuri audio/video în timp real

Introducerea serviciilor de *streaming live* permite utilizatorilor să urmărească simultan mai multe canale TV prin Internet. În *streaming live*, fluxurile video sunt generate în același timp cu descărcarea și vizualizarea lor de către clienți. Deci, avem de-a face cu distribuirea unui fișier de lungime necunoscută și imprevizibilă în care datele sunt disponibile doar pentru o perioadă scurtă de timp. În acest caz, cea mai importantă provocare este întârzierea afișării unui flux video, adică timpul scurs între producția conținutului și afișarea lui. Experiența utilizatorului final este similară unei emisiuni TV *live*, deoarece toți utilizatorii vor intenționa să urmărească cel mai recent conținut generat. Serviciul popular de *streaming* video *live* este Protocolul de Televiziune prin Internet (IPTV).

http://www.explainthatstuff.com/how-iptv-works.html

Capitolul 8. Securitatea rețelelor

SECURITÁTE s. f. **1. Faptul de a fi la adăpost de orice pericol; sentiment de încredere și de liniște pe care îl dă cuiva absența oricărui pericol. ♦ Protecție, apărare.** ◊ *Securitate colectivă* = stare a relațiilor dintre state, creată prin luarea pe cale de tratat a unor măsuri de apărare comună împotriva unei agresiuni. *Securitate socială* = totalitatea reglementărilor juridice pentru asigurarea stării de siguranță socială la nivel de persoană, grup social sau populație totală, precum și pentru protejarea persoanelor defavorizate sau marginalizate. **2.** (Ieșit din uz) Organ de stat represiv care avea ca sarcină apărarea prin orice mijloace a sistemului comunist din România. – Din fr. **sécurité,** lat. **securitas, -atis.** [15]

În cazul rețelelor de date, a fi la adăpost de orice pericol, a te proteja și apăra de orice infracțiune ce poate fi comisă în spațiul cibernetic, presupune implicarea tuturor componentelor unui astfel de sistem: protocoale, tehnologii, sisteme, instrumente și tehnici de lucru.

Securitatea unei rețele de calculatoare înseamnă în primul rând securitatea informației. Pentru aceasta trebuie avute în vedere trei aspecte importante (triada C-I-A [16]):

- **confidențialitatea**, care reprezintă calitatea de a permite, doar persoanelor autorizate, accesul la informație;
- **integritatea**, care reprezintă garanția că informația nu a fost modificată de persoane neautorizate;
- **disponibilitatea**, care este definită ca fiind asigurarea accesului la informație atunci când aceasta este necesară.

Fiind un domeniu extrem de complex, ISO (Organization for Standardization) și IEC (International Electrotechnical Commission) au stabilit 14 subdomenii ale securității informației, prin standardul ISO 27001:

- **Politica de Securitate** este un document care tratează măsurile coercitive și comportamentul membrilor unei organizații și specifică cum vor fi accesate datele, ce date sunt accesibile și cui.
- **Organizarea Securității Informației** e un model de guvernare elaborat de o organizație pentru securitatea informației.
- **Securitatea Resurselor Umane** definește procedurile de securitate privind angajarea, detașarea și părăsirea de către un angajat a organizației din care va face, face sau a făcut parte.
- **Administrarea Bunurilor** reprezintă un inventar potrivit unei scheme clasificate pentru bunurile informaționale.
- **Controlul Accesului** privește restricțiile aplicate accesului direct la rețea, sisteme, aplicații și date.
- **Criptografia** definește modalitățile de criptare a informației și administrarea cheilor.

- **Securitatea Fizică și a Mediului** descrie măsurile de protecție pentru centrele de date din cadrul unei organizații.
- **Securitatea Operațională** controlează serviciile de rețea, securitatea acesteia, transferul de informație, etc.
- **Securitatea Comunicațiilor** definește cerințele de securitate pentru procesele de dezvoltare și suport.
- **Achiziția, Dezvoltarea și Mentenanța Sistemelor** definește aplicarea măsurilor de securitate în aplicații.
- **Relațiile cu Furnizorii** din perspectiva controlului asupra înțelegerilor cu aceștia și a monitorizării lor.
- **Administrarea Incidentelor de Securitate a Informației** tratează felul în care sistemul anticipează și răspunde în cazul unei breșe de securitate.
- **Administrarea Continuității Afacerii** descrie măsurile de protecție, întreținere și verificare și revizuire pentru afacere și sisteme.
- **Conformitatea** descrie procesul de asigurare a conformității cu politicile de securitate a informației, standarde și reguli.

Starea de securitate poate fi garantată prin implementarea a patru mecanisme de protecție: descurajare, prevenire, detectare și răspuns [17].

Descurajarea este de obicei prima linie de apărare împotriva intrușilor care pot încerca să obțină acces la o rețea. Presupune crearea unei atmosfere menite să sperie intrușii, iar uneori, acest lucru poate implica transmiterea unor avertismente.

Prevenirea este procesul încercării de a împiedica intrușii să obțină acces la resursele unui sistem sau al unei rețele prin utilizarea de *firewall*-uri, zonele demilitarizate (DMZ) sau utilizarea elementelor de acces pentru a permite utilizarea și accesul numai utilizatorilor autorizați.

Detectarea are loc atunci când intrusul a reușit sau este în curs de accesare a sistemului sau rețelei. Semnalele din procesul de detectare includ alerte la existența unui intrus. Uneori aceste alerte pot fi în timp real sau pot fi stocate pentru analize suplimentare de către personalul de securitate.

Răspunsul este un mecanism care încearcă să răspundă la eșecul primelor trei mecanisme. Funcționează încercând să oprească și/sau să prevină deteriorarea sistemului sau rețelei.

8.1 Aspecte privind securitatea rețelelor

8.1.1 Autentificarea, autorizarea și monitorizarea activității utilizatorilor

Autentificarea unui persoane sau a unei aplicații reprezintă conceptul de bază al asigurării securității unei rețele sau a unui sistem și este de cele mai multe ori făcută printr-o parolă și un nume de utilizator. Una din problemele acestei metode este că parolele nu sunt foarte sigure, pot fi furate sau chiar ghicite, iar mulți utilizatori aleg unele foarte simple ce

folosesc cuvinte foarte evidente ori și le salvează în locații sau documente la care pot avea acces mai multe persoane. De asemenea, au apărut în ultimii ani din ce în ce mai multe metode sau programe cu ajutorul cărora se pot „fura” parole.

Autentificarea este cu atât mai sigură cu cât parolele sunt mai complicate și se schimbă la intervale scurte de timp, sau dacă se utilizează metode alternative precum identificarea retinei sau amprentei, folosirea de carduri de acces sau a dispozitivelor de tip *token* care generează parole de unică folosință.

După autentificare, utilizatorul trebuie să primească autorizarea de a realiza anumite sarcini. Autorizarea activității acestuia este făcută de un administrator care are permisiunea de a-i controla accesul către diverse resurse sau servicii. Monitorizarea presupune măsurarea resurselor consumate de către utilizator în timpul sesiunii de lucru și poate include durata de timp cât acesta a utilizat resursele, cantitatea de date trimisă sau primită, resursele accesate în mod explicit. Toate acestea, de obicei, se salvează în înregistrări automate și pot fi utilizate pentru control, analize statistice sau planificare.

8.1.2 Asigurarea confidențialității și integrității datelor

Serviciul de confidențialitate protejează datele și informațiile împotriva accesărilor neautorizate, prin folosirea de algoritmi de criptare și decriptare. Criptarea protejează datele pe parcursul transferului lor prin canalul de comunicație, decriptarea făcându-se de către destinatar.

Criptarea datelor se poate realiza prin două metode: algoritmi cu cheie simetrică sau cu cheie asimetrică.

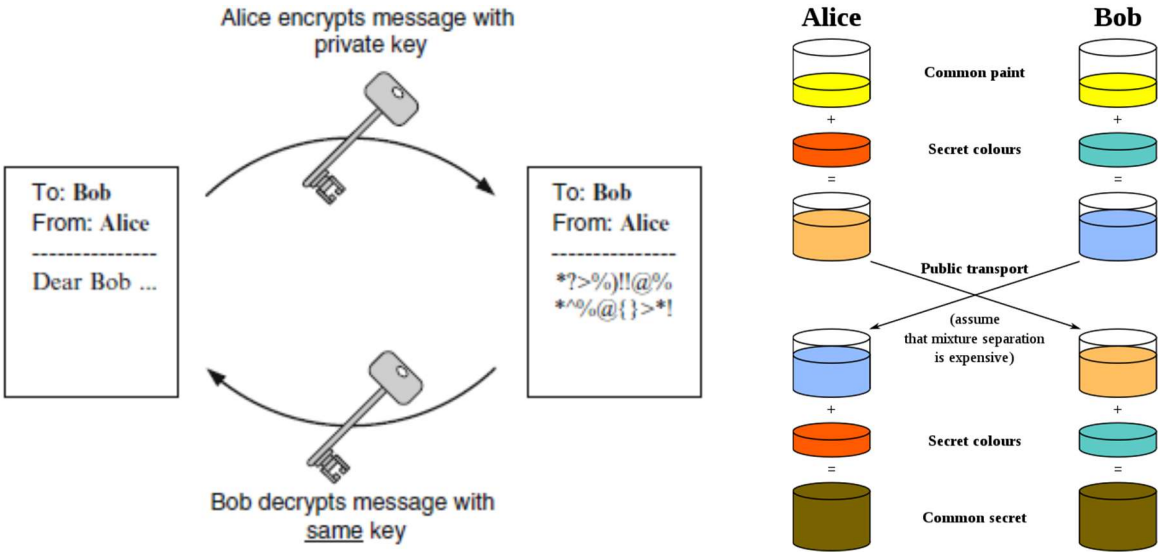

Figura 60. Criptarea cu cheie simetrică și metoda Diffie-Hellman

Algoritmii cu cheie simetrică utilizează aceeași cheie atât pentru criptare cât și pentru decriptare, cerința principală, dar și marele dezavantaj, fiind ca ambele părți care comunică să aibă acces la acea cheie. Dificultatea apare din cauza necesității existenței unei metode prin care să se poate transmite această cheie de la emițător la receptor, fără a intra în posesia altora. Transmiterea personală este cea mai sigură, dar imposibilă dacă este vorba de o comunicare

între două puncte diferite ale planetei. Pentru aceste situații se utilizează metode de schimb în siguranță al cheii de criptare (de exemplu metoda Diffie-Hellman) sau un protocol cu cheie publică. Un exemplu de algoritm cu cheie simetrică utilizat în rețelele de calculatoare este AES (Standard Avansat de Criptare).

Algoritmii cu cheie asimetrică folosesc două tipuri de chei, una privată și una publică. Oricine poate cripta un mesaj folosind cheia publică, însă numai deținătorul cheii private (pereche cu cheia publică folosită pentru criptare) poate decripta mesajul. Pentru mai multă siguranță, cheia publică face parte dintr-un certificat digital eliberat de o autoritate de certificare. Cele mai cunoscute aplicații ale acestui tip de algoritm sunt criptarea cu cheie publică, pentru transmiterea de mesaje, și crearea de semnături digitale. În cazul rețelelor de calculatoare aplicațiile client-server folosesc un astfel de algoritm în cadrul protocolului TLS (Securitatea Nivelului de Transport).

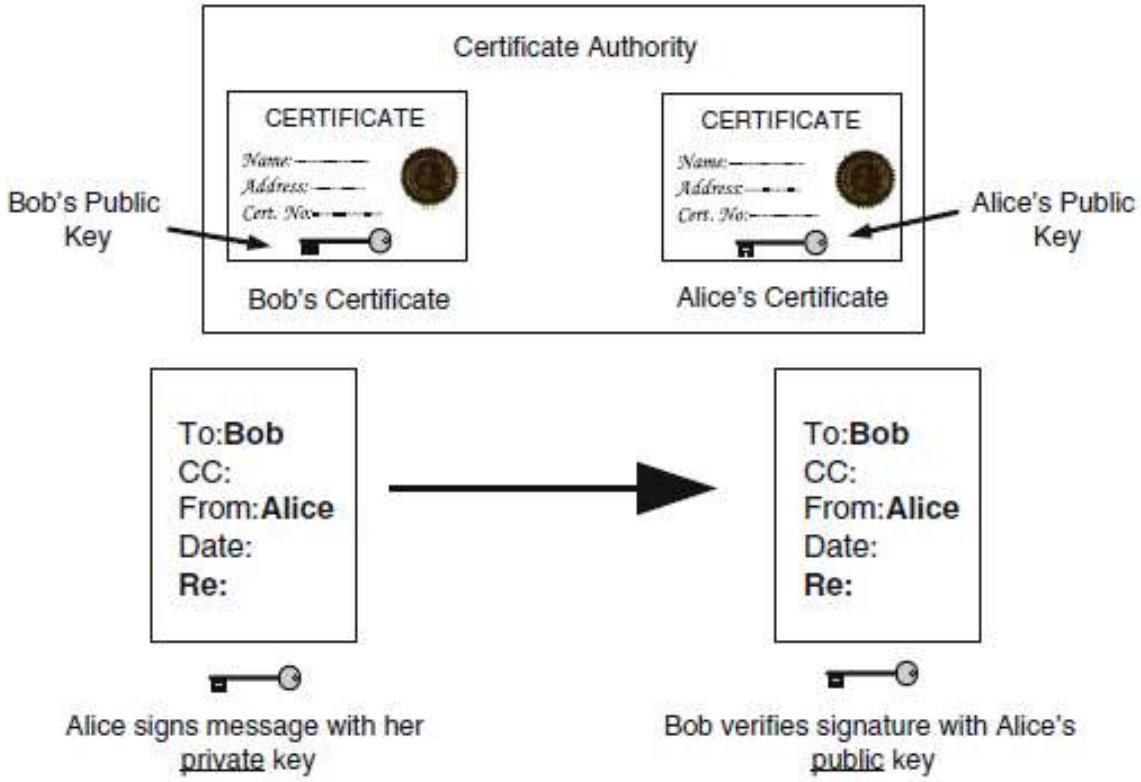

Figura 61. Criptarea cu cheie asimetrică

Datele aflate în tranzit între emițător și receptor pot fi alterate în mod intenționat. Verificarea integrității acestora se realizează de obicei folosind coduri *hash*. Un cod *hash* este o valoare numerică de lungime fixă (de obicei 128 de biți) care identifică în mod unic datele și se obține din acestea pe baza unei funcții *one-way-hash* (nu se poate aplica și invers pentru a determina datele pe baza codului *hash*). Codul este apoi criptat și transmis împreună cu mesajul. Receptorul decriptează codul, aplică funcția pe mesajul primit și verifică dacă valoarea *hash* a datelor primite este identică cu valoarea *hash* a datelor trimise, pentru a determina dacă datele au fost modificate.

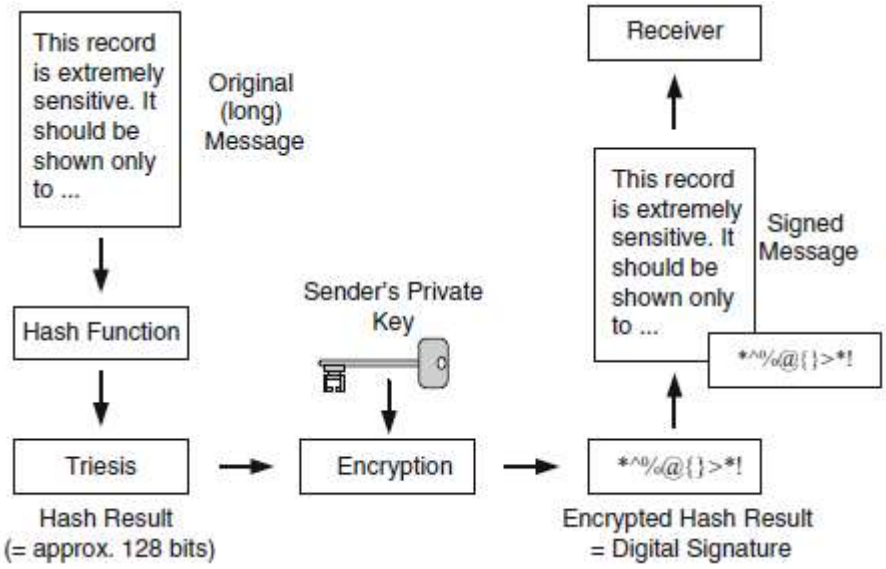

Figura 62. Verificarea integrității datelor cu coduri *hash*

8.1.3 Securizarea perimetrului rețelei

Perimetrul unei rețele este definit de acea graniță care separă zona privată, deținută și administrată local, de zona publică administrată de un furnizor de servicii.

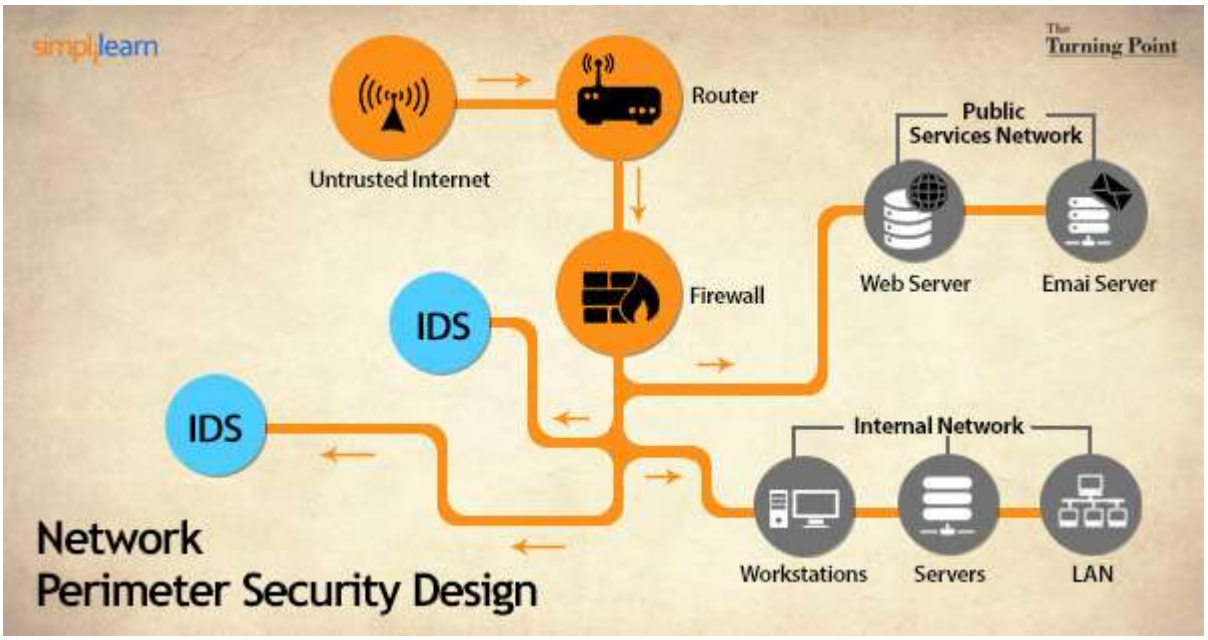

Figura 63. Securizarea perimetrului unei rețele

Un singur nivel de securitate nu va fi niciodată suficient, în afara ruterului de graniță, ce conține de obicei un *firewall* de bază (*firewall*-ul este o barieră care poate fi configurată să controleze ce anume poate trece din rețeaua internă în exterior și invers.), fiind necesare niveluri suplimentare precum împărțirea rețelei în sisteme cu acces în rețeaua publică sau în cea privată, un *firewall* de ultimă generație care poate controla traficul la nivel de aplicație și de utilizator sau un echipament de tip IDS (Sistem de Detecție a Intrușilor).

Securitatea rețelei poate fi îmbunătățită prin crearea unei zone demilitarizate (DMZ) care presupune existența a două *firewall*-uri, unul pentru traficul extern și unul pentru cel intern.

Toate componentele utilizate în cadrul rețelei trebuie actualizate permanent, din punct de vedere al configurației hardware, al aplicațiilor sau sistemelor de operare, și trebuie configurate în mod corespunzător, astfel încât rețeaua să poată fi protejată nu numai față de amenințările curente, ci și față de cele care evoluează de-a lungul timpului.

8.1.4 Monitorizarea rețelei

Monitorizarea rețelei implică managementul configurației rețelei și controlarea activităților normale dintr-o rețea: monitorizarea accesului, a ruterelor sau *switch*-urilor, a sistemelor de tip *firewall*, a sistemelor de tip server sau client. În acest caz, administratorii rețelei se ocupă de controlarea și supravegherea modului în care funcționează aceasta, folosind unelte software specializate. Aceștia au nevoie să știe de existența tuturor componentelor, denumirile și adreselor acestora și, de asemenea, detaliile de rutare. Trebuie cunoscute relațiile dintre componente și operațiile caracteristice fiecăruia.

Monitorizarea presupune, de obicei, determinarea unor parametri precum disponibilitatea, timpul de răspuns sau de funcționare a unui sistem, aplicație sau serviciu. De obicei sunt trimise mesaje prin rețea pentru a verifica felul în care acestea răspund la cereri. De exemplu, verificarea stării unui site web se poate face prin trimiterea de cereri periodice pentru a descărca o anumită pagină. În cazul apariției unor erori sau a unor răspunsuri lente se vor trimite alerte ce vor fi vizibile administratorilor de rețea. Astfel de teste se pot realiza cu comenzi de tip *ping* sau cu protocoale de monitorizare și administrare precum SNMP (Protocol Simplu de Administrare a Rețelei).

8.2 Surse de vulnerabilități

Greșeli de proiectare. Pot să apară în oricare din cele două componente majore ale unei rețele de calculatoare, hardware sau software. Sistemele hardware sunt mai puțin susceptibile la astfel de greșeli datorită experienței în domeniul ingineriei și complexității mai reduse ce le face mai ușor de testat. Cele mai mari probleme de vulnerabilitate se datorează greșelilor în proiectarea componentelor software, trei factori majori contribuind în mare măsură: factorul uman, complexitatea software-ului și sursele software de încredere.

Gestionarea deficitară a securității. Este rezultatul unui control scăzut asupra implementării, administrării și monitorizării securității. Este un eșec în a avea un control solid asupra situației de securitate a organizației atunci când administratorul de securitate nu știe cine stabilește politica de securitate a organizației, cine administrează respectarea securității și configurațiile de securitate ale rețelei și cine are sarcina de a trata evenimentele și incidentele de securitate.

Implementarea incorectă. Este de obicei rezultatul utilizării unor interfețe incompatibile între două produse hardware sau software care trebuie să lucreze împreună, a utilizării unei arhitecturi de rețea nesigură, a liniilor de comunicație neprotejate, sau a configurării insuficiente și incorecte a produselor.

Factorul uman. Securitatea unei rețele este atât asigurată cât și afectată de persoanele care o administrează, întrucât orice situație care necesită implicarea directă a acestora poate avea diverse efecte: lipsa atenției, uitarea temporară, graba de a finaliza o activitate, utilizarea unor algoritmi netestați, rea-intenție.

8.3 Clasificarea atacurilor

În funcție de vulnerabilitățile rețelei, atacurile se pot manifesta pe mai multe planuri:

- Accesarea neautorizată a rețelei sau a unor resurse ale acesteia din interiorul organizației sau din afara acesteia.
- Tentative de perturbare sau de întrerupere a funcționării rețelei la nivel fizic (prin factori mecanici, de întrerupere a unor cabluri sau scoatere din funcțiune a unor echipamente din rețea; factori electrici, de bruiaj în cazul rețelelor radio, semnale de interferență în rețelele cablate).
- Tentative de întrerupere sau de încărcare excesivă a traficului din rețea prin transmiterea unui număr foarte mare de pachete către unul sau mai multe noduri din rețea (*flooding*).
- Atacuri asupra software-ului echipamentelor de rețea care concentrează și dirijează fluxurile în noduri critice (*switch*, *router*, *access point* etc.) prin modificarea fișierelor de configurare și a drepturilor de acces stabilite de personalul autorizat.
- Modificarea sau distrugerea informației, adică atacul la integritatea fizică a datelor.
- Preluarea și folosirea neautorizată a informațiilor, adică încălcarea confidențialității și a dreptului de autor.

În funcție de locul de unde se execută, atacurile pot fi:

- **Atacul local**, ce presupune spargerea securității unei rețele de calculatoare de către un utilizator local, adică o persoană care face parte din rețea și care dispune de un cont și de o parolă de utilizator care îi dau drept de acces la o parte din resursele sistemului. De asemenea, persoana respectivă poate să aibă cunoștințe despre arhitectura sistemului de securitate al rețelei, putând astfel lansa atacuri mult mai periculoase, principalele riscuri constând în accesarea informațiilor la care nu are drept de acces, găsirea punctelor vulnerabile ale rețelei prin încărcarea unor programe care să scaneze rețeaua.
- **Atacul la distanță** (*remote attack*) este un atac lansat împotriva unei rețele de comunicații sau a unui echipament din rețea, față de care atacatorul nu deține nici un fel de control. Accesul de la distanță la resursele unei rețele este mai riscant decât accesul din rețeaua locală deoarece în Internet sunt câteva miliarde de utilizatori ceea ce face ca numărul posibililor atacatori externi să fie mult mai mare decât al celor interni.

În funcție de modul în care acționează, ca sursă și destinație, atacurile pot fi:

- **Centrate pe o singură entitate** (de exemplu, este atacat un anumit server din rețea de pe un singur echipament).
- **Distribuite** (lansate din mai multe locații sau către mai multe mașini simultan).

În funcție de metodele utilizate, atacurile pot fi:

- **Nestructurate**, ce sunt inițiate de indivizi neexperimentați ce utilizează exploit-uri disponibile pe Internet. Exploit-urile sunt programe ce exploatează vulnerabilitățile pentru a ocoli politica de securitate implementată într-o rețea.
- **Structurate**, ce sunt inițiate de indivizi mult mai bine motivați și cu cunoștințe tehnice competente. Acești indivizi cunosc vulnerabilități de sistem și le pot folosi pentru a căpăta acces în rețea, pot detecta noi vulnerabilități de sistem și pot dezvolta cod și scripturi pentru a le exploata.

Atacurile asupra unui sistem pot fi executate atât pentru culegere de date, cât și pentru modificarea acestora. Din acest punct de vedere atacurile se clasifică în **atacuri pasive** și **atacuri active**.

Atacurile pasive sunt acele atacuri care au ca țintă furtul datelor și drepturilor utilizatorilor autorizați. Aceste date pot fi utilizate ulterior de atacator în accesarea diverselor componente ale sistemului ca și când acesta ar fi utilizatorul de drept al datelor. Atacurile pasive pot fi:

- **monitorizarea transmisiei** dintre două entități (*eavesdropping*) și furtul informației (*packet sniffing*) care este transmisă între ele. Atacatorul nu intenționează să întrerupă serviciul, sau să cauzeze un efect, ci doar să intre în posesia informației.
- **analiza traficului** – dacă informația este criptată, va fi mult mai dificil să fie citită, dar atacatorul nu doar observă informația, ci și încearcă să înțeleagă ceva din ea; sau pur și simplu să determine identitatea și locația celor două părți implicate în conversație sau să descopere modul și cheia de criptare.

Atacurile pasive sunt greu de detectat din moment ce există un impact foarte mic asupra informației comunicate.

Atacurile active au scopul de a cauza o întrerupere, și de obicei sunt ușor de recunoscut. Spre deosebire de atacul pasiv, un atac activ modifică informația, poate șterge, insera sau întârzia mesaje sau întrerupe un serviciu. Exemple de atacuri active:

- **Mascarada** (*masquerade*) – atacatorul pretinde a fi altcineva cu intenția de a obține date secrete. Multe dintre atacurile de acest tip pot fi evitate prin adoptarea unor politici de securitate adecvate, care presupun responsabilizarea utilizatorilor, implementarea unor metode de acces robuste, folosirea unor metode de autentificare cât mai eficiente.
- **Reluarea** – se produce atunci când un mesaj sau o parte a acestuia este reluată (repetată), cu intenția de a produce un efect neautorizat (autentificarea atacatorului folosind informații de identificare valide, transmise de un utilizator

autorizat al rețelei). Acest tip de atac poate fi prevenit prin etichetarea fiecărei componente criptate cu un ID de sesiune și un număr de componentă.

- **Modificarea mesajelor** – face ca datele mesajului sa fie alterate prin modificare, inserare sau ștergere. Poate fi folosită pentru a se schimba beneficiarul unui credit în transferul electronic de fonduri sau pentru a modifica valoarea acelui credit. O altă utilizare poate fi modificarea câmpului destinatar/expeditor al poștei electronice.
- **Refuzul serviciului** – se produce când o entitate nu izbutește să îndeplinească propria funcție. Acest lucru se realizează prin supraîncărcarea serverelor cu cereri din partea atacatorului și consumarea resurselor, astfel încât acele servicii să nu poată fi oferite și altor utilizatori.

8.4 Tipuri de atacuri și metode de protecție

8.4.1 Ingineria Socială

Reprezintă unul din cele mai simple și eficiente atacuri, dar totuși nu necesită cunoștințe în domeniul tehnologiilor. Ea presupune manipularea persoanelor ce au o autoritate în sistemul ce urmează a fi spart, pentru a face anumite lucruri, ce l-ar ajuta pe hacker să execute atacul. De obicei, ingineria socială este însoțită de alte tipuri de atacuri, astfel devenind o armă puternică în mâna atacantului.

Ingineria socială poate fi evitată prin implementarea tehnicilor de securitate ce protejează de accesul liber al persoanelor neautorizate în încăperile companiei, instruirea personalului, anunțarea lucrătorilor în caz că apare o persoană nouă autorizată, etc.

Există mai multe metode de atacuri de acest tip, printre care *phishing*-ul şi *baiting*-ul.

Phishing-ul reprezintă un atac în care se simulează o organizație legitimă, care cere informații confidențiale de la utilizator, de exemplu, pe e-mailul victimei vine un mesaj ce conține site-ul emulat al unei organizații reale, cu emblema sa, iar când utilizatorul încearcă să se autentifice, parola sa este trimisă atacatorului.

Baiting-ul este un tip de atac prin care victima introduce codul dăunător în calculatorul său. Hackerul poate lăsa codul pe un disc sau un flash drive USB, ce se va instala automat când acestea sunt introduse în calculator, de obicei din curiozitate pentru a vedea ce se află pe mediul de stocare respectiv.

8.4.2 Spargerea parolelor

Reprezintă un atac pe care hackerul îl efectuează ca să se poată autoriza și autentifica într-un sistem pentru a-i obține resursele. În majoritatea cazurilor, acesta obține nu parolele, ci *hash*-ul acelor parole. Deoarece funcția de criptare a parolelor nu este una reversibilă, parola

este aflată prin încercarea tuturor variantelor posibile, sau conform unui dicționar, până când parola criptată coincide cu *hash*-ul căpătat.

Pentru a evita riscul ca parolele să fie sparte, e nevoie ca ele să aibă o dificultate mare, să conțină litere mici, majuscule, cifre, semne de punctuație, totodată trebuie ca ele să fie schimbate la intervale regulate, pentru a nu da șanse atacatorului să reușească să le spargă în acest interval de timp.

8.4.3 Flooding

Flooding-ul presupune blocarea un server sau client cu o cantitate anormală de pachete, având ca scop supraîncărcarea acestuia. Atacurile sunt periculoase doar în cazul în care lățimea de bandă a victimei este cu mult mai mică decât cea a atacatorului. În prezent astfel tipuri de atacuri nu prezintă pericol, deoarece lățimile de bandă de obicei sunt destul de mari pentru a suporta cantitățile mari de pachete.

8.4.4 Spoofing

Spoofing-ul nu este mereu un atac, dar de obicei este însoțit de un atac. Reprezintă ascunderea informației despre calculatorul atacator, de exemplu a adresei IP, adresei MAC, serverului DHCP, DNS, etc. și a face mai dificilă găsirea acestuia.

Spoofing-ul se realizează prin servere *proxy*, vulnerabilități în protocoalele TCP/IP sau prin serviciile anonime de pe internet.

Cea mai utilizată metodă de securizare împotriva *Spoofing*-ului, este criptarea datelor între rutere și gazde externe, ce micșorează șansa ca hackerul să afle datele despre calculatoare în timp rezonabil.

8.4.5 Sniffing

Sniffing-ul reprezintă procesul de capturare și analiză a traficului. Programele folosite pentru *sniffing* se numesc *sniffere* sau analizatoare de protocoale. Ele analizează pachetele transmise prin rețea, capturând parolele sau alte date confidențiale transmise în formă de text simplu.

Pentru protecție se utilizează protocoale pentru securizarea comunicațiilor (IPSec) care criptează traficul din rețea, astfel datele capturate de hacker nu vor fi ușor descifrabile. Altă metodă ar fi folosirea programelor anti-*sniffer*, ce verifică dacă rețeaua este monitorizată.

8.4.6 Denial of Service (DoS)

Scopurile atacurilor DoS nu sunt captarea datelor, parolelor, ci prevenirea utilizatorilor legitimi de a se folosi de anumite resurse ale rețelei. Atacurile DoS se pot manifesta în 2 moduri: prin inundarea cu informație invalidă a serverului, sau prin căderea activității lui. Cele mai

frecvente atacuri DoS sunt bazate pe protocoalele TCP/IP și funcționează prin una din următoarele metode:

- Consumul resurselor computaționale, precum lățimea benzii de transfer, spațiu de stocare, puterea procesorului, etc.
- Coruperea informației
- Coruperea stării informației, de exemplu întreruperea nesolicitată a conexiunilor TCP/IP.
- Distrugerea fizică a componentelor rețelei.
- Împiedicarea comunicării dintre 2 calculatoare.

Există multe tipuri diferite de atacuri DoS, cele mai grave dintre acestea fiind:

- **Ping of Death**, ce trimite pachete de tip *ping* de mărime mai mare decât mărimea maximă 65535 B, astfel pachetul ICMP este fragmentat și stația victimă va trebui să îl reasambleze, dar în acest timp el mai primește altele, ajungându-se astfel la supraîncărcarea sistemului.
- **Permanent Denial of Service** (PDoS), cunoscut ca și *Phlashing*, reprezintă atacuri care distrug sistemul într-atât încât e necesară înlocuirea componentelor hardware, sau chiar a întregului sistem. Atacatorul obține acces la o imprimantă, ruter sau alte componente din rețea și îi poate modifica *firmware*-ul cu o imagine invalidă, coruptă, sau modificată, astfel distrugând acea componentă.
- O tehnică specială de atac DoS e reprezentată de *„Banana attack”*, ce redirecționează toate pachetele trimise de client unui server, înapoi clientului, inundându-l cu aceleași pachete trimise.
- DDoS (*Distributed Denial of Service*) reprezintă atacul în care un singur server este atacat de multe calculatoarea *Zombie*, care sunt infectate de hacker prin diverse metode, de obicei prin intermediul programelor malware, în care este înscrisă adresa IP a victimei. Nu este nevoie de interacțiunea atacatorului pentru a realiza atacul, deși în unele cazuri el poate prelua controlul asupra calculatoarelor infectate. În prezent nu există metode eficiente de evitare a atacurilor DDoS, totodată nu poate fi aflată ușor proveniența atacului.
- DRDoS (*Distributed Reflected Denial of Service Attack*) presupune trimiterea unor cereri false către un număr mare de calculatoare, iar cu ajutorul IP *Spoofing* se redirecționează toate răspunsurile către gazda cu IP-ul emulat.

8.4.7 Man-in-the-Middle (MITM)

Atacul MITM presupune ca hackerul să intercepteze și, dacă are nevoie, să modifice conținutul mesajelor dintre două calculatoare, făcând ambele victime să creadă că comunică una cu cealaltă, conversația fiind de fapt controlată de atacator. Acest atac poate fi obținut prin intermediul *Spoofing*-ului *DHCP,* adică un calculator din rețea va pretinde că el este serverul DHCP, luând informația despre gazde de la serverul DHCP real. Dacă atacatorul indică calculatoarelor din rețea date greșite, acesta poate aplica apoi *sniffing*-ul pentru a afla toate

datele confidențiale trimise de calculatoarele din rețea altor rețele externe. Acest *Spoofing* DHCP poate fi detectat utilizând programele special destinate.

8.4.8 DNS cache poisoning

Presupune modificarea bazei de date *cache* a serverului DNS, astfel încât el va asocia adrese ale site-urilor Web cu IP-uri greșite, redirecționând de fapt utilizatorul spre un alt site. ce poate conține un virus, vierme, sau cal troian, astfel, infectând calculatorul victimei prin intermediul vulnerabilităților în serverul DNS.

8.4.9 Malware

Este un tip de software proiectat intenționat pentru deteriorarea unui calculator sau infiltrarea în el, sau/și deteriorarea ori infiltrarea în întregi rețele de calculatoare, fără consimțământul deținătorului. Noțiunea se utilizează generalizat de către informaticieni pentru a desemna orice formă ostilă, intruzivă sau supărătoare de software sau cod de program. De cele mai multe ori, software-ul dăunător este folosit pentru a lua, fără voia proprietarului, informații personale din computerul infectat, cum ar fi: parole, date bancare, alte informații confidențiale. Protecția împotriva acestui tip de atac se realizează prin utilizarea antivirușilor.

8.4.10 Spyware

Sunt atașate de obicei la programe gratuite (jocuri, programe de partajat fișiere, programe de chat, etc.), captează pe ascuns date de marketing (prin analiza siturilor pe care le vizitează utilizatorul) și le folosesc apoi pentru a transmite utilizatorului reclame corespunzătoare dar nesolicitate. În 99 % din cazuri programul spion este instalat de însuși utilizatorul calculatorului, în mod voit sau nu, necitind licența programului ce conține *spyware*, prin apăsarea la instalare pe un buton de genul „Da, sunt de acord”.

Programele de tip spion nu sunt considerate viruși informatici, deoarece, în general, ele nu caută să infecteze programe și nici să atace calculatoarele altor. Ele sunt considerate doar amenințări la adresa sferei private a utilizatorilor. Unele programe antivirus nu detectează niciun fel de program spion. Pentru înlăturarea programelor spion sunt folosite programele anti-*spyware*.

8.4.11 Trojan horse

Calul troian este un tip de software spion care pare că ar realiza ceva util, dar care în realitate realizează funcții ce permit accesarea neautorizată a unui calculator, respectiv copierea fișierelor, și chiar controlarea comenzilor acestuia. Caii troieni, care tehnic nu sunt viruși informatici, pot fi descărcați cu ușurință și în necunoștință de cauză. Spre exemplu, dacă un joc pe calculator este astfel proiectat ca la executarea sa de către un utilizator să deschidă o ușă de intrare (*backdoor*) pentru un hacker, care poate prelua ulterior controlul computerului, se spune

despre acel joc că este un cal troian. Protecția împotriva acestora se poate realiza prin utilizarea antivirușilor.

8.4.12 Ransomware

Este un software rău intenționat care, după ce se instalează pe dispozitivul victimei (calculator, smartphone) criptează datele acesteia ținându-le „ostatice" sau șantajează victima, pe care o amenință că îi va publica datele dacă aceasta nu plătește o „răscumpărare". Se răspândește în rețea ca un cal troian. Ca metodă de protecție se folosește un antivirus, dar dacă datele au fost criptate se vor folosi aplicații de decriptare specifice fiecărui *ransomware*.

8.4.13 Viruși

Sunt programe care se instalează singure, fără voia utilizatorului, și pot provoca pagube atât în sistemul de operare cât și în elementele hardware (fizice) ale computerului. Acestea pot fi orientate spre componentele hardware (le solicită din punct de vedere fizic până la deteriorarea acestora) sau spre componentele software (afectează toate tipurile de fișiere, inclusiv pe cele ale sistemului de operare). Câteva dintre efectele pe care le generează virușii software:

- distrugerea unor fișiere;
- modificarea dimensiunii fișierelor;
- ștergerea totală a informațiilor de pe disc, inclusiv formatarea acestuia;
- distrugerea tabelei de alocare a fișierelor, care duce la imposibilitatea citirii informației de pe disc;
- diverse efecte grafice/sonore, inofensive sau și dăunătoare;
- încetinirea vitezei de lucru (utilă) a calculatorului, până la blocarea acestuia;
- înmulțirea fișierelor până la umplerea memoriei;
- ascunderea fișierelor si blocarea anumitor spații.

Pot fi detectați și eliminați prin utilizarea antivirușilor.

8.4.14 Viermi (worms)

Sunt programe care se instalează singure, se auto-multiplică și se răspândesc în rețea prin utilizarea vulnerabilităților sistemelor de operare. Mulți viermi care au fost creați sunt concepuți doar pentru a se răspândi și nu încearcă să schimbe sistemele prin care trec. Aproape întotdeauna provoacă cel puțin unele daune rețelei, chiar și numai prin consumul de lățime de bandă.

Capitolul 9. Rețele de senzori fără fir

9.1 Descriere generală

Rețelele de senzori fără fir (WSN) sunt formate dintr-un număr mare de senzori inteligenți, denumiți în continuare noduri, dispuși pe o suprafață exterioară sau în interiorul unei clădiri, cu capacități de comunicație fără fir, mobili sau ficși, care, prin acțiuni ce implică colaborarea, formează o rețea cu scopul de a implementa o anumită aplicație ca un tot unitar.

Nodurile pot fi elemente familiare, cum ar fi vehicule sau telefoane, sau pot fi mici dispozitive separate. De exemplu, un vehicul ar putea colecta date despre locație, viteză, vibrații sau eficiența combustibilului de la sistemul său de diagnosticare și le va încărca într-o bază de date externă [6].

O astfel de rețea este formată dintr-un număr foarte mare de noduri și, din acest considerent, o caracteristică ce trebuie avută în vedere pentru fiecare nod este costul. Se optează de obicei pe o soluție cu un preț cât mai scăzut dat fiind numărul mare de noduri. În același timp trebuie să se țină cont de faptul că în marea majoritate a situațiilor nodurile sunt alimentate din baterie și se dorește ca durata de viață a unui nod să fie cât mai lungă. Pentru a se realiza acest lucru nu este suficient să se folosească componente hardware optimizate pentru consum redus de energie ci și din punct de vedere software trebuie să se țină cont de tehnici de programare și proiectare software orientate pe reducerea consumului de energie. În cazul aplicațiilor în timp real toate tehnicile atât de reducere a consumului de energie cât și cele de reducerea costului nu trebuie să afecteze cerințele stricte de timp impuse de aplicație.

În cadrul proiectării unei rețele de senzori trebuie avute în vedere câteva caracteristici importante:

- Numărul mare de senzori – pentru a utiliza în mod eficient dimensiunile mici și costul redus al senzorilor, rețelele de senzori pot conține mii de noduri. Administrarea acestor uriașe rețele este o problemă majoră. Împărțirea în grupuri (*clustering*) este o soluție la aceasta problemă. Astfel, nodurile apropiate se unesc pentru a forma un grup (*cluster*) și aleg un coordonator pentru a administra grupul.
- Constrângerile legate de timpul de viață a unui nod, date de către modul de alimentare – în multe aplicații nodurile se vor afla într-o locație îndepărtată în care nu se va putea face întreținerea acestuia. Astfel durata de funcționare a unui nod poate fi determinată de timpul de viață al bateriei acestuia, drept urmare senzorul trebuie să consume cât mai puțină energie.
- Probleme în cadrul procesului de comunicație generate de către mediul prin care se face transmisia fără fir – aici se încadrează interferențele ce pot apărea în mediul în care funcționează nodurile. Acestea pot fi cauzate de dispozitive ce transmit de asemenea datele într-o manieră fără fir și folosesc o frecvență apropiată de cea a tehnologiei utilizată în rețeaua de senzori.

- Abilitatea de auto-configurare și auto-vindecare, fără a fi necesar un grad de intervenție ridicat din partea administratorilor, odată ce sistemul este configurat și funcțional. Nodurile se pot alătura rețelei, pot ieși din rețea sau se pot defecta. În momentul în care apare o problemă, iar datele nu mai pot fi transmise de la nodurile ce realizează achiziția de date către nodul central, reconfigurarea se face în mod automat, fiind căutate alternative.
- Utilizarea eficientă a memoriei și puterii de calcul – la construirea unei rețele de senzori, ținând cont de probleme precum construirea unor tabele de rutare, răspunsuri la fluxuri de date și probleme de securitate, trebuie avută în vedere memoria și puterea de calcul limitate de care dispun nodurile rețelei.
- Acumularea de informații – numărul, uneori uriaș, de noduri poate duce la congestia rețelei datorită cantității mari de informații. Pentru a rezolva această problema unele noduri, cum ar fi coordonatorii, pot acumula informația și pot face diverse calcule (medii, sume, calcul de maxime și minime), pentru a realiza un rezumat pe care mai apoi să-l transmită în rețea.

9.2 Clasificarea rețelelor de senzori fără fir

În funcție de dimensiunea rețelei sau distanța dintre nodul de bază (în cazul unei rețele organizate după o topologie coordonată) și cele mai îndepărtate noduri ale rețelei:

- Rețele single-hop, în care fiecare nod comunică doar cu coordonatorul rețelei. O astfel de rețea este de obicei de mici dimensiuni dar se caracterizează prin simplitate deoarece nu este nevoie de algoritmi complecși care să realizeze rutarea informației. Nodurile comunică doar atunci când au ceva de transmis nodului coordonator și astfel își economisesc mult din energia bateriei.
- Rețele multi-hop, în care puține noduri sunt în raza de comunicație a coordonatorului rețelei și astfel ele nu își pot transmite datele direct la coordonator ci doar prin noduri intermediare, fiind necesară implementarea unor algoritmi specializați care să realizeze rutarea informației de la nodul transmițător până la nodul receptor, de obicei coordonatorul rețelei.

În funcție de densitatea rețelei și dependența de date:

- Rețele cu agregare, în care fiecare nod transmite coordonatorului toate datele pe care le deține fără să se aplice vreo transformare asupra lor. În cazul unei rețele cu multe noduri acest lucru generează trafic ridicat ceea ce implicit duce la scăderea duratei de viață a rețelei în urma consumului mare de energie a fiecărui nod.
- Rețele fără agregare, în care nodurile sunt grupate de obicei în funcție de localizare, datele sunt agregate și prelucrate local, iar coordonatorului i se transmite doar un rezultat și astfel traficul scade considerabil. De obicei această prelucrare a datelor se rezumă ori la o medie ori la eliminarea datelor redundante.

În funcție de modul de distribuție a nodurilor:

- Rețele deterministe, în care plasarea nodurilor este cunoscută și fixă. În această situație întreaga gestiune a rețelei devine mult mai facilă, mulți algoritmi (de exemplu de rutare, localizare) ne mai fiind necesari. O astfel de rețea are un grad mare de rigiditate.
- Rețele dinamice, în care nu se cunoaște de la început poziția fiecărui nod. Motivul necunoașterii poziției este fie plasarea în mod aleatoriu a nodurilor, fie nodurile au și capacitate de mobilitate. În acest caz toate operațiunilor realizate în cadrul rețelei se complică simțitor, dar se oferă un mare grad de flexibilitate și scalabilitate.

În funcție de politica de control:

- Rețele cu autoconfigurare, în care se permite nodurilor rețelei să se organizeze atât din punct de vedere al comunicării cât și din punct de vedere al sarcinilor. În acest caz, o entitate superioară de control nu trebuie decât să traseze sarcina în linii mari iar nodurile se auto-organizează și, formând un mediu de colaborare, decid în colectiv fiecare pas.
- Rețele fără autoconfigurare, în care nodurile nu au capacitatea de a se auto-organiza ci au nevoie de o entitate de control care să comande ce să facă la fiecare pas. Acesta politică de control nu poate fi folosită decât pentru rețele de mici dimensiuni și unde nu se dorește scalabilitate.

În funcție de modul de administrare al rețelei:

- Rețele Ad-Hoc, în care senzorii comunică direct între ei.
- Rețele gestionate de un coordonator.

9.3 Configurații de rețea

Rețelele de tip Ad-Hoc caracterizează cel mai mult ideea de rețea de senzori fără fir, în general. Rețelele de acest tip sunt acelea care nu au nevoie de un coordonator pentru a-și îndeplini funcția. Prin procese de colaborare senzorii, care în principiu sunt dispuși aleatoriu, se auto-organizează pentru a realiza funcția destinată. Acest tip de rețele prezintă un grad mare de scalabilitate și adaptabilitate.

Sunt posibile două situații de organizare. O situație în care toate nodurile sunt în aceeași zonă de acoperire, adică orice nod poate comunica în mod direct cu oricare alt nod. De obicei această situație este destul de greu de obținut. Un alt caz este acela când raza de acoperire a nodurilor este mică și astfel apărând situația în care unele noduri trebuie să realizeze și funcții de rutare a informației. În acest caz rețeaua are facilități întâlnite la o rețea de tip plasă.

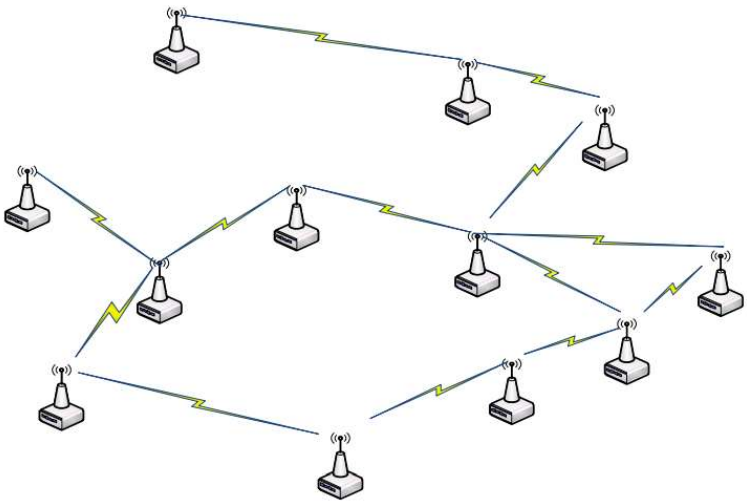

Figura 64. Exemplu de rețea Ad-Hoc

Marea majoritate a rețelelor de senzori sunt gestionate de un coordonator. În această configurație există noduri cu *funcții reduse* în ceea ce privește comunicarea (noduri finale, senzori) și noduri cu *funcții evoluate* (ruter, coordonator) care au rolul de a le conduce pe celelalte, formându-se astfel o structură ierarhizată de noduri. Spre deosebire de un nod final, un ruter este mereu activ (treaz) fiind proiectate pentru a utiliza alimentare externă.

În general un nod cu funcții evoluate de comunicare împreună cu nodurile pe care le conduce formează un *cluster*.

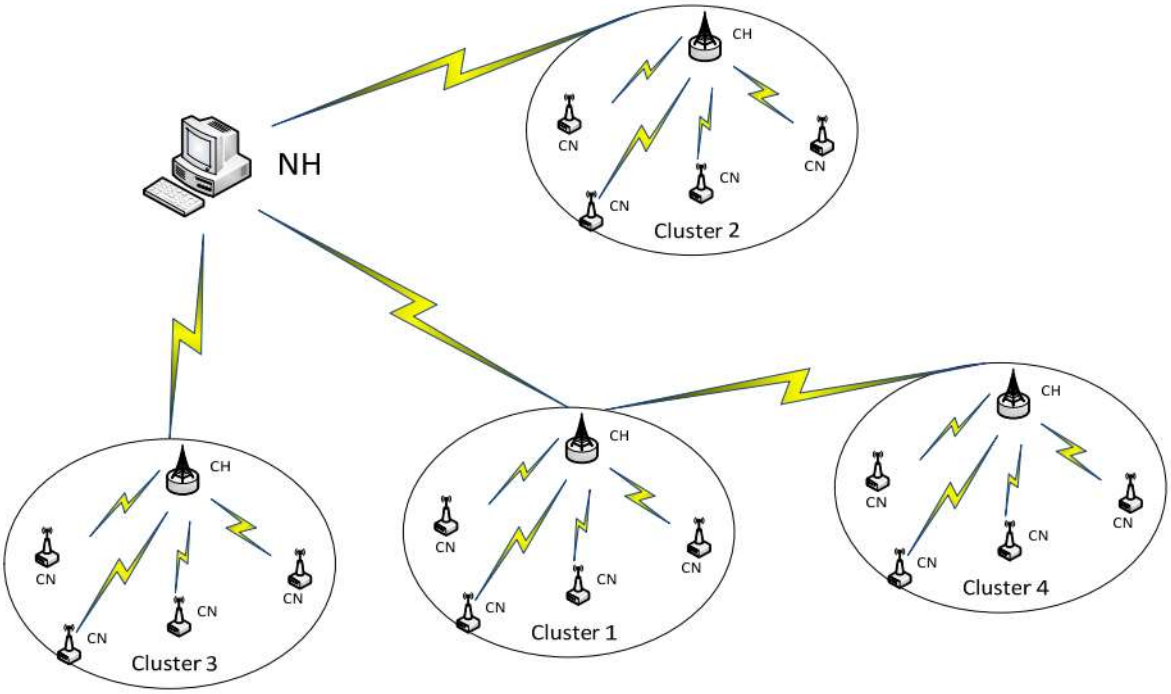

Figura 65. Exemplu de rețea organizată pe *clustere*

În Figura 65 se prezintă un exemplu de rețea organizată în structuri de clustere. Nodurile desemnate cu CN (*Cluster Node*) reprezintă senzorii și nu pot comunica decât cu noduri de tip CH (*Cluster Head*) și în concluzie au capabilități reduse de comunicare. Fiecare nod CN trebuie să fie în raza de acoperire a nodului CH ce conduce clusterul din care face parte și, în principiu,

nu poate comunica decât cu acesta. Așadar un cluster este format dintr-un coordonator sau conducător de cluster (CH) și un număr de noduri de tip CN.

Coordonatorul unui cluster (numit de obicei ruter) este de obicei un nod mai bogat în resurse decât nodurile pe care le conduce putând implementa nu doar funcții de bază de comunicare ci și funcții mai evoluate cum ar fi rutarea informației. Deși sunt mai bogate în resurse, nodurile de tip CH sunt și cele la care durata de viață a bateriei poate să scadă mult mai repede față de celelalte noduri deoarece acestea, având și funcția de rutare de informație, consumă multă energie electrică pentru comunicarea radio.

În figura prezentată mai sus se observă cum *clusterul* 4 nu este în raza directă de acoperire a coordonatorului rețelei și astfel CH1 trebuie să realizeze rutarea informației între acesta rețelei și CH4.

O rețea de senzori într-o asemenea structură este formată din mai multe *clustere*, toate fiind conduse de către un coordonator de rețea simbolizat în figură prin NH (*Network Head*). De obicei acest tip de nod dispune de multe resurse cum ar fi energia electrică, memorie, putere de calcul. În unele situații NH este chiar un calculator clasic și nu neapărat un sistem integrat.

Coordonatorul și ruterele lucrează împreună pentru a forma o rețea de tip plasă. Două exemple de rețele de acest fel se pot observa în Figura 66. Deoarece numărul de salturi al datelor prin rețea este bine să fie cât mai mic, a doua variantă este de preferat în locul primei.

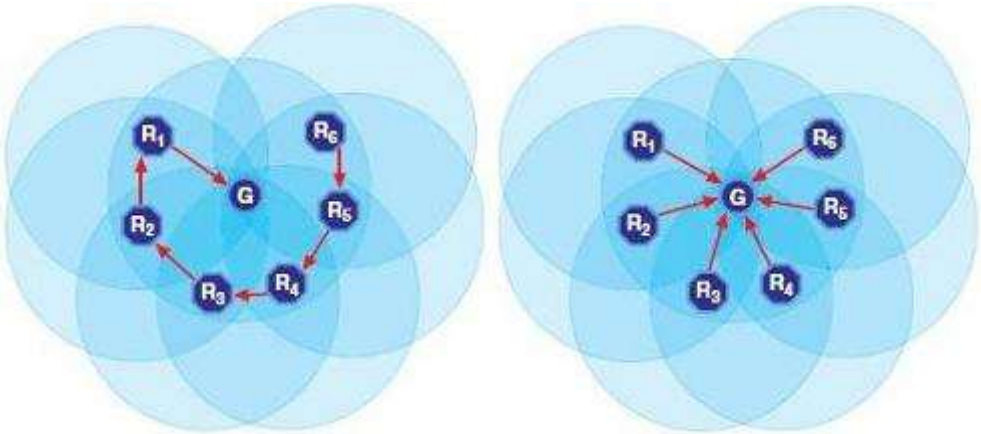

Figura 66. Configurații de rețea tip plasă

Mărirea distanței de acoperire a rețelei de senzori se poate face cu ajutorul ruterelor prin plasarea acestora conform Figurii 67.

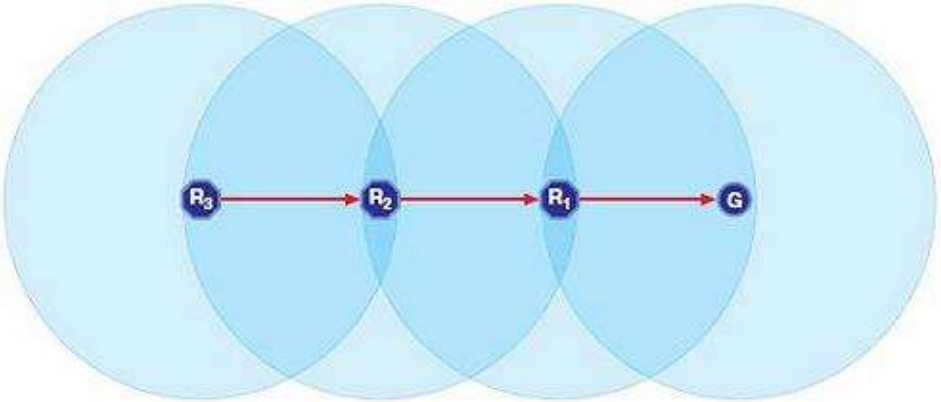

Figura 67. Extinderea distanței de acoperire a unei rețele

Dacă se dorește să se pună în aplicare o configurație tip stea, în care niciun ruter nu este utilizat, se pot conecta doar noduri finale conectate la coordonator, așa cum se vede în Figura 68.

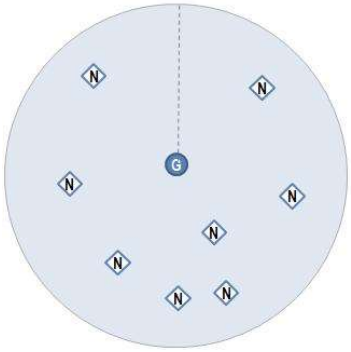

Figura 68. Configurații de rețea tip stea

Dacă densitatea nodurilor este mai importantă decât distanța de acoperire a rețelei se poate utiliza modelul de configurație din Figura 69.

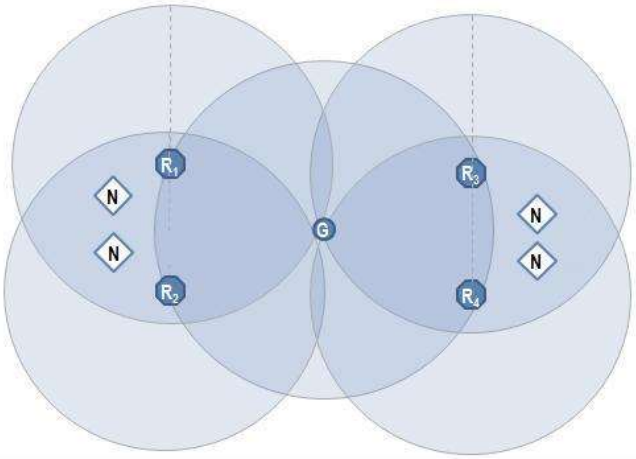

Figura 69. Topologie de rețea tip plasă cu densitate ridicată și distanțe medii

Dacă distanța de acoperire a rețelei este mai importantă decât densitatea nodurilor se poate utiliza modelul de configurație din Figura 70.

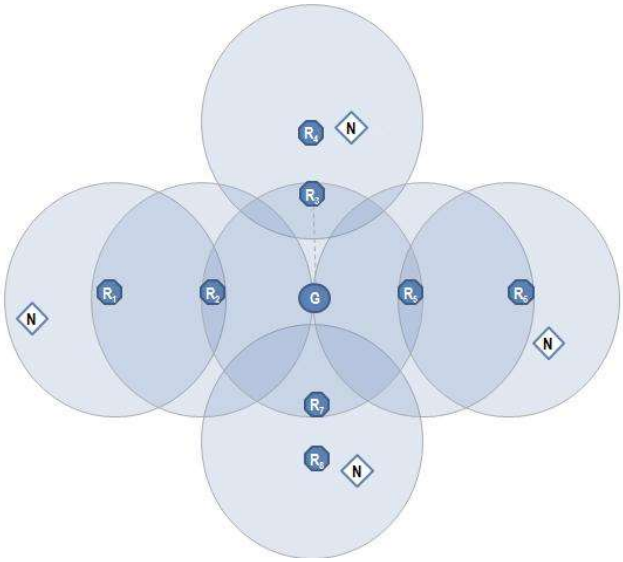

Figura 70. Topologie de rețea tip plasă cu densitate ridicată și distanțe medii

9.4 Nodurile unei rețele de senzori

Un nod are, de obicei, ca și componentă centrală un microcontroler cu putere mică de calcul și cu consum redus de energie electrică, ce asigură întreaga funcționare a acestuia.

O configurație minimală a unui nod cuprinde:

- un microcontroler ce realizează toată partea de calcul și interfațare cu componentele externe;
- interfață de comunicație cu sau fără fir care, de obicei, transmite date în ambele direcții și este una din componentele care consumă cea mai multă energie;
- memorie externă care permite stocarea permanentă a unei cantități mari de date, fără a fi nevoie să fie transmise prin interfața de comunicație;
- senzorul;
- sursă de alimentare: baterie, acumulator, celulă solară.

Într-o asemenea configurație minimă rețeaua de senzori poate fi folosită în principiu doar pentru monitorizare într-o topologie fixă cu un grad scăzut de dinamism.

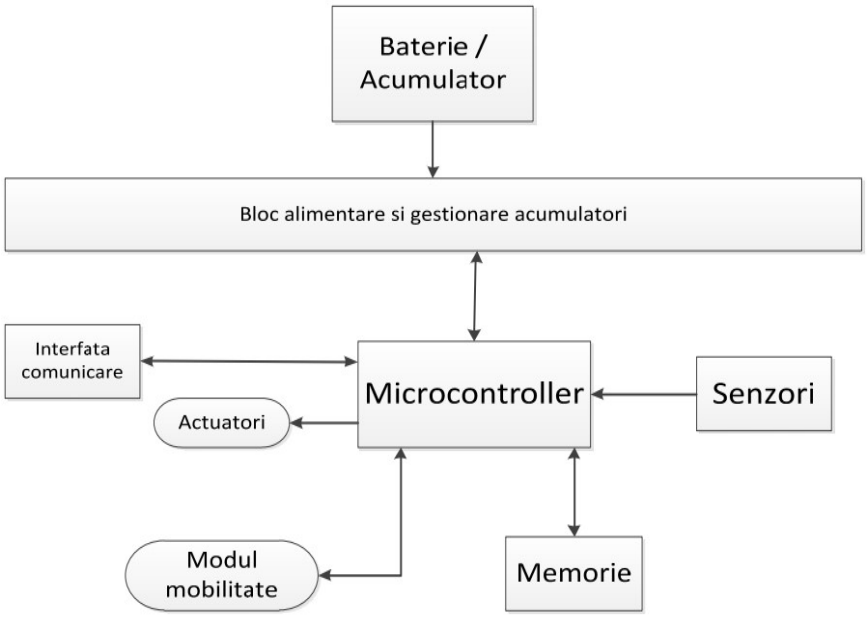

Figura 71. Configurația unui nod de rețea

În cazul în care se adaugă diferite module opționale cum ar fi cele din Figura 71 caracteristicile rețelei se schimbă. Spre exemplu, în situația adăugării unor elemente de acționare rețeaua primește și facilitate de control nu doar de monitorizare, iar în cazul adăugării și a unui modul de mobilitate rețeaua devine puternic dinamică. Deși o astfel de rețea este de dorit în multe situații, dinamismul introduce și o serie de probleme atât pe parte de mobilitate și consum de energie cât mai ales pe partea de comunicație fără fir.

Una din cele mai importante proprietăți este consumul de energie, cantitatea disponibilă fiind de obicei mult redusă. În majoritatea aplicațiilor, nodurile sunt în cea mai mare parte a timpului într-un mod de ”așteptare” (*sleep*), în care interfața de comunicație este oprită și pornită periodic, generând trafic doar atunci când au loc evenimente de interes. În felul acesta

se urmărește o eficientizare a consumului de energie. Există totuși și domenii în care este nevoie de monitorizare și generare continuă de date, cum ar fi supravegherea mediului înconjurător.

În Figura 72 se poate observa cantitatea de energie consumată de unele exemple de componente ale unui nod.

Component	Mode	Current Draw
Microcontroller (TI MSP430)	Active Sleep	1.8 mA 5.1 μA
RF Transceiver (CC2420)	Receive Transmit (at 0 dBm) Sleep	19.7 mA 17.4 mA 0.01 mA
Accelerometer (ADXL345)	Standby Active	0.0001 mA 0.04 – 0.145 mA
External flash (Micron M25P16)	Write Read Sleep	15 mA 4 mA 0.001 mA
Temperature sensor (TMP102)	Sense Sleep	0.015 mA 0.001 mA

Figura 72. Exemple de consum de energie ale componentelor unui nod

O metodă utilă ce permite detectarea evenimentelor de interes în timp real dar și reducerea consumului de energie este utilizarea întreruperilor. Software-ul nodului ce așteaptă date de la senzor (fie executând alte funcții, fie aflându-se în modul *sleep*) preia datele în momentul detecției întreruperii (de obicei semnalată prin modificarea stării logice a unui port al microcontrolerului), iar după aceasta revine la sarcinile obișnuite sau în modul *sleep*.

9.5 Protocoale de comunicație

Dat fiind faptul că nodurile unei rețele de senzori wireless sunt sisteme integrate, sărace în resurse, protocoalele de comunicație nu respectă ad litteram modelul OSI, atât din considerentul că în acest caz poate nu sunt necesare toate nivelurile, cât și din cauza resurselor puține, ceea ce face imposibilă folosirea unor protocoale existente pentru sisteme de calcul clasice.

Principalele niveluri care se regăsesc în protocoalele de comunicație din cadrul unei rețele de senzori fără fir sunt:

- Nivelul fizic – se ocupă cu accesul la mediul fizic de comunicație. Acest nivel este obligatoriu să fie implementat pe un nod al unei rețele. Aici se regăsesc cu precădere părți din protocoalele de tip MAC.
- Nivelul de Acces la Mediu – este nivelul unde are loc administrarea modului de acces la mediul de comunicație: ascultare, trimite de date, așteptare, etc.
- Nivelul de Management al Legăturii – asigură comunicația de tip punct la punct între două noduri ale rețelei aflate în aceeași arie de acoperire, în acest caz realizându-se o comunicație directă între ele. Aici se regăsesc de asemenea protocoale de tip MAC cât și protocoale pentru detecția și corecția erorilor și controlul retransmisiilor.

- Nivelul de Rutare – abstractizează rețeaua sub forma unui graf pe baza căilor de comunicație existente între noduri și asigură o comunicație fiabilă punct la punct (*end-to-end*) în cadrul rețelei. Aici se regăsesc cu precădere protocoalele de rutare a informației.
- Nivelul Aplicație – este nivelul unde datele utile sunt preluate, pre-procesate și puse în formatul corespunzător pentru a fi transmise apoi nivelurilor inferioare. Aici se mai regăsesc de multe ori și protocoalele de sincronizare.

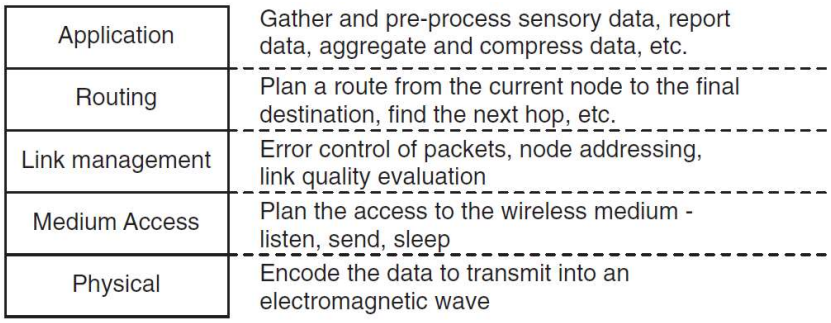

Figura 73. Modelul OSI simplificat pentru rețelele de senzori [18]

Protocoalele MAC sunt de o importanță majoră în proiectarea și realizarea rețelelor de senzori fără fir, fiind responsabile cu prevenirea interferențelor și a coruperii pachetelor de date, în același timp urmărind maximizarea ratei de transfer și minimizând consumul de energie electrică. În plus, față de rețelele de comunicații obișnuite, trebuie să implementeze metode de gestionare a comunicației cu dispozitivele deconectate cât și de minimizare a timpului cât un nod de află în stare de ascultare a canalului de comunicație, astfel încât acestea să evite risipa de energie electrică.

Astfel, se pot defini patru criterii de proiectare a protocoalelor MAC:

- Minimizarea coliziunilor, ce permite evitarea situațiilor în care trebuie retransmise pachete de date, ducând, evident, la mărirea ratei de transfer și scăderea cantității de energie utilizată.
- Minimizarea prelucrării inutile de pachete, ce are loc atunci când un nod primește un pachet care nu îi era destinat, pachet ce trebuie distrus. Aceasta este o sarcină dificilă deoarece nodul trebuie să știe când îi este destinat un pachet sau nu și când să intre în mod *sleep* sau nu.
- Minimizarea ascultării inutile a mediului, ce are loc atunci când nodul ascultă canalul de comunicație și nu se întâmplă nimic. Aceasta duce la irosirea energiei deoarece nodul folosește aceeași cantitate de energie și când ascultă canalul, și când primește date.
- Minimizarea antetului pachetelor, deoarece fiecare bit care trebuie transmis consumă energie.

Ca și în cazul rețelelor fără fir pentru calculatoare, și protocoalele MAC pentru rețelele de senzori folosesc tehnici răspândite precum TDMA, CSMA-CD, CSMA-CA, dar și unele adaptate acestora, precum Sensor MAC sau Berkley MAC ce permit nodurilor să intre în mod *sleep* și să-și realizeze transferul datelor numai atunci când sunt active.

9.6 Legături de date

Legăturile fără fir de date sunt nesigure, asimetrice și cu mari fluctuații în raport cu timpul și spațiul. Caracterizarea acestor legături, a fiabilității și calității lor, se poate face cu următorii parametri:

- Rata de Recepție a Pachetelor (PRR), este raportul între pachetele livrate cu succes și numărul total de pachete trimise. Se măsoară în procente și este ușor de calculat.
- Indicatorul de Putere a Semnalului Recepționat (RSSI), este furnizat pentru fiecare pachet recepționat și se măsoară în dBm (decibel per metru). Cu cât valoarea este mai mare cu atât semnalul este mai bun, iar valorile uzuale sunt negative.
- Indicatorul de Calitate a Legăturii (LQI), este un scor dat pentru fiecare pachet în parte de către transmițătorul radio, parte din standardul 802.15, dar implementat în diverse variante și de către alți producători. De obicei constă dintr-un număr între 50 și 110, valorile mai mari indicând o calitate mai bună a legăturii.
- Raportul Semnal Zgomot (SNR), este raportul dintre semnalul util și nivelul zgomotului din mediul de transmisie. Se măsoară în decibeli. Se poate calcula măsurând RSSI al zgomotului și RSSI al unui pachet de date.

Datorită atenuării undelor electromagnetice în raport cu distanța parcursă de acestea, pe măsură ce aceasta crește vor apărea pierderi de pachete din ce în ce mai mari, până când se ajunge la imposibilitatea de menținere a legăturii de date. Rata de Recepție a Pachetelor descrie cel mai bine această problemă (Figura 74), observându-se însă că, deși teoretic semnalul ar fi trebuit să se atenueze gradual cu creșterea distanței, pentru măsurări efectuate la aceeași distanță, rezultatele obținute sunt mult diferite. Astfel, legăturile de date pot fi caracterizate ca fiind impredictibile.

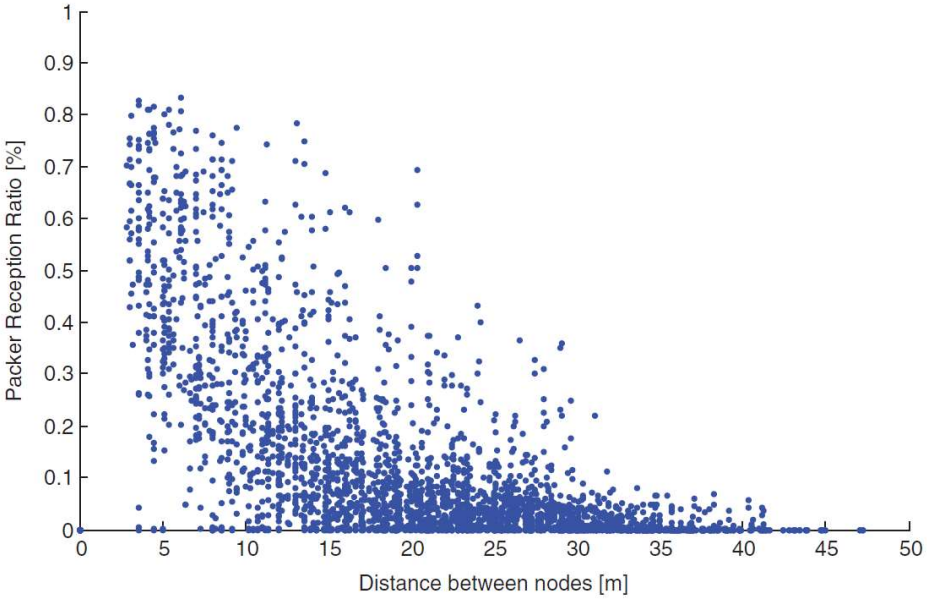

Figura 74. PRR pentru o comunicație între două echipamente la diferite distanțe [18]

O zonă de comunicație în care se poate stabili o legătură de date cu o anumită certitudine poate fi încadrată în trei moduri [19]:

- Zona de comunicație efectivă, în care probabilitatea de livrare a pachetelor este de cel puțin 90%.
- Zona de comunicație de tranziție, în care probabilitatea de livrare poate varia între 100% și 10%.
- Zona fără comunicație, în care livrarea de pachete este aproape imposibilă.

Legăturile dintre două noduri sunt utilizate, de obicei, pentru transferul de date în ambele sensuri. Deoarece aceste transferuri sunt afectate de mediul înconjurător și de interferențele din acesta, succesul transmiterii de date într-un sens nu înseamnă în mod obligatoriu și succesul transmiterii în sens invers. Practic, razele de acoperire a două noduri pot fi diferite în momente diferite, deși ele se află la aceeași distanță unul de celălalt, apărând astfel legăturile asimetrice între noduri (Figura 75).

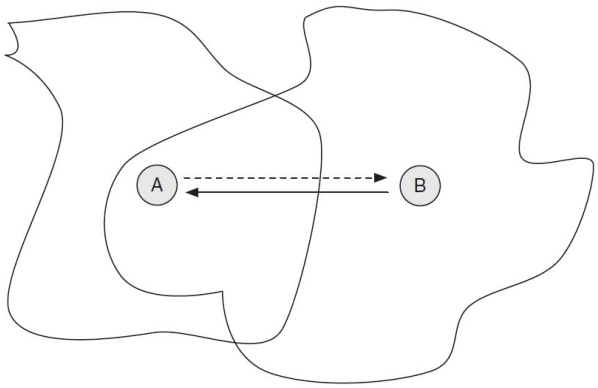

Figura 75. Legătură de date asimetrică

Afectarea unei legături de date nu duce doar la pierderea de pachete datorită imposibilității realizării acesteia, ci poate duce la apariția de erori ce are ca efect eliminarea de pachete și necesitatea retransmiterii lor. Există totuși metode de control al erorilor (FEC), fie la emisie, fie la recepție (BEC).

Controlul erorilor la emisie pleacă de la premisa că retransmiterea unui pachet consumă mai multă energie decât includerea în acesta a unor biți suplimentari, redundanți. Cea mai cunoscută metodă presupune repetarea datelor efective care trebuie transmise, de un număr de ori, datele corecte fiind extrase la recepție prin suprapunerea grupurilor de date primite.

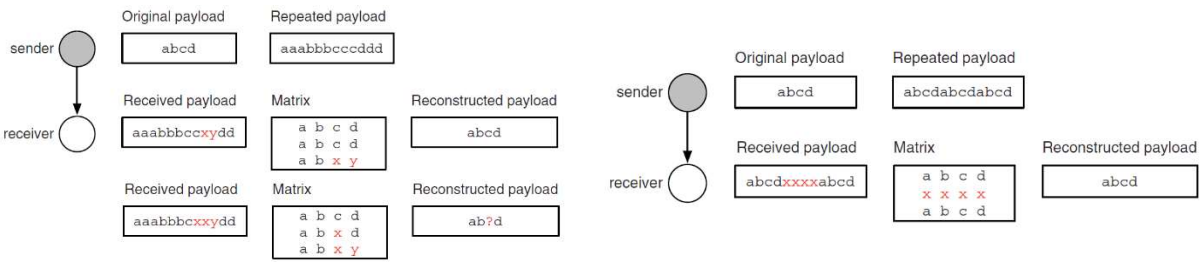

Figura 76. Exemplu de verificare FEC, fără și cu intercalare

Controlul erorilor la recepție presupune detecția acestora și realizarea unor cereri de retransmitere a pachetelor eronate. Cea mai cunoscută metodă de detecție se numește Controlul

Redundant Ciclic (CRC), iar cea mai simplă verificare presupune însumarea biților dintr-un pachet (*checksum*) și memorarea acestei sume într-un bit de paritate (0 pentru număr par de biți 1, sau 1 pentru număr impar de biți 1) (Figura 77).

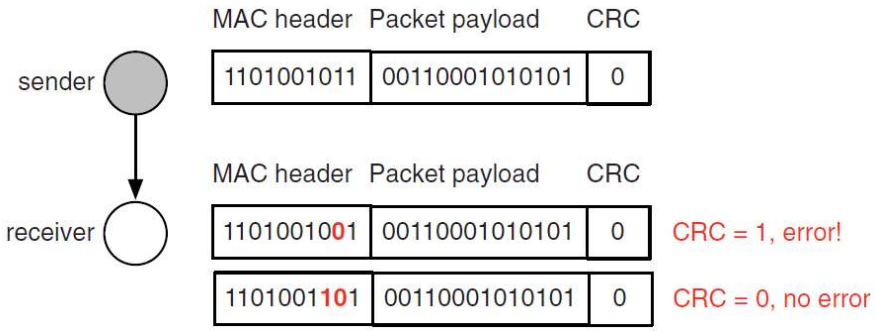

Figura 77. Exemplu de verificare CRC simplă

În practică se utilizează coduri CRC complexe, pentru a evita situația din Figura 77 în care modificarea a doi biți nu duce la schimbarea codului rezultat, dar cel mai adesea nu utilizatorul le implementează, ci ele sunt incluse în protocoalele tehnologiilor de comunicație.

9.7 Comunicații multi-hop

Comunicațiile multi-hop se referă la schimbul de date între noduri îndepărtate, iar oricare dintre acestea nu se află în raza de acoperire a nodului cu care comunică. Astfel, pentru a transmite cu succes datele este necesară rutarea lor prin rețea cu ajutorul altor noduri.

Rolurile nodurilor unei rețele de senzori, în ceea ce privește transferul datelor, sunt [18]:

- Sursa de date, este nodul care produce datele necesare și este capabil să-l transmită altor noduri din rețea.
- Destinația datelor, este nodul care necesită datele și este capabil să le recepționeze de la alte noduri din rețea.
- Transportorul de date, este orice nod din rețea, care nu este nici sursă din destinație a datelor, care este capabil să le recepționeze de la un nod și să le transmită către altul.
- Colectorul de date, este un nod dedicat al rețelei care are rolul de destinație implicită al oricăror date transmise în rețea. Acesta are, de obicei, o altă legătură de comunicație de bandă largă, cu sau fără fir, cu un alt echipament mai puternic (de ex. un calculator).

În funcție de sursa și destinația datelor se pot deosebi patru tipuri de scenarii [18]:

- Difuzare la nivelul întregii rețele (*full network broadcast*), când există o singură sursă a datelor și toate nodurile din rețea reprezintă destinații ale acestora.
- Transmisie nod-la-nod (*unicast*), când există o singură sursă și o singură destinație a datelor.
- Transmisie nod-la-multi-nod (*multicast*), când există o singură sursă și mai multe noduri destinație.

- Transmisie multi-nod-la-nod (*convergecast*), când toate nodurile din rețea sunt surse de date, un singur nod (de obicei colectorul de date) fiind destinația acestora. Se utilizează în cazul rețelelor al căror scop este colectarea de date.

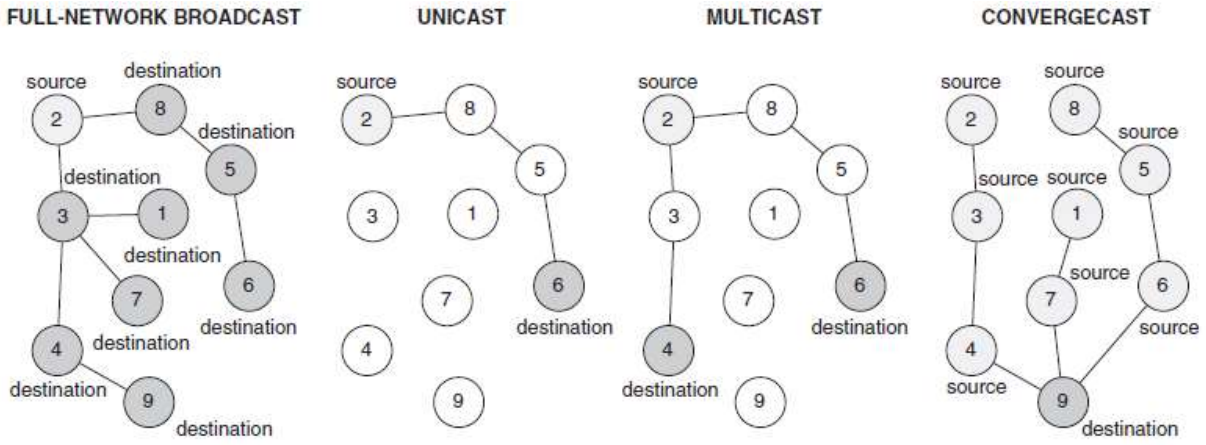

Figura 78. Scenarii de transmisie a datelor

Rutarea într-o rețea multi-hop este procesul de selectare a unei secvențe de noduri din rețea (**calea de rutare**), începând cu sursa de date și terminând cu destinația lor. O cale de rutare validă conține un număr finit de noduri și nici o buclă. Nodurile individuale ce formează o cale de rutare se numesc **hopuri**.

O cale de rutare poate fi **prestabilită** sau **la cerere**. Căile prestabilite permit transmiterea datelor imediat, deci având întârzieri mici, dar implementarea lor este costisitoare și s-ar putea să nu fie folosite niciodată. Căile obținute la cerere sunt create atunci când sunt necesare, iar primul pachet transmis are, de obicei, întârziere mare.

Maniera de stabilire a căilor de rutare poate fi de tip **centralizat** sau de tip **distribuit**. Căile stabilite în mod centralizat sunt calculate sau identificate de un nod special din rețea (de obicei nodul colector) și apoi transmise nodurilor individuale. Abordarea distribuită se bazează pe identificarea căilor direct la nivelul nodurilor individuale, ținând cont de caracteristicile de moment, cum ar fi nodurile vecine existente sau energia disponibilă. Stabilirea distribuită a căilor de rutare este o soluție bună în cazul apariției de defecte în rețea sau în cazul nodurilor mobile.

În rețelele de senzori rutarea este întotdeauna stabilită hop cu hop, nici un nod din rețea (fie sursă, fie colector) necunoscând întreaga cale de rutare până la un nod destinație, ci doar hopul următor. Aceasta este o diferență majoră față de rețelele în care rutarea se face pe bază de adrese (IP de exemplu), fiind necesară deoarece o defectare a unui nod de-a lungul căii de rutare ar necesita retransmiterea datelor.

Parametrii de rutare stau la baza acesteia și reprezintă modalitatea de comparație a gradului de vecinătate al nodurilor în raport cu destinația. Aceștia pot fi:

- **Localizarea sau vecinătatea geografică**. Folosește locații geografice reale sau distanțe aproximate, cerința principală fiind ca toate nodurile să-și cunoască poziția proprie și poziția nodului destinație. Dezavantajul principal al metodei

este că nu ține cont de proprietățile rețelei de comunicație. Alt dezavantaj este că proximitatea geografică nu garantează și drumul cel mai scurt.

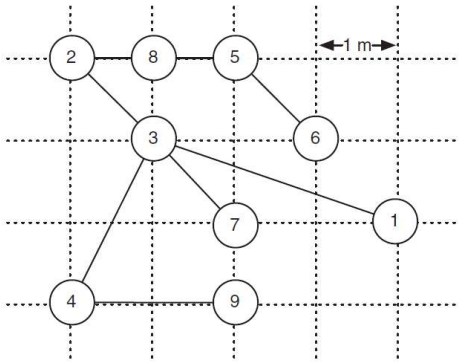

Figura 79. Rutare pe baza localizării nodurilor

- **Numărul de hopuri**. Stabilirea căii de rutare pleacă de la presupunerea că un hop corespunde unei distanțe în rețea, prin urmare mai multe hopuri înseamnă o distanță mai mare. Numărul de hopuri este corelat cu distanța geografică dar și cu topologia rețelei și numărul de legături disponibile.

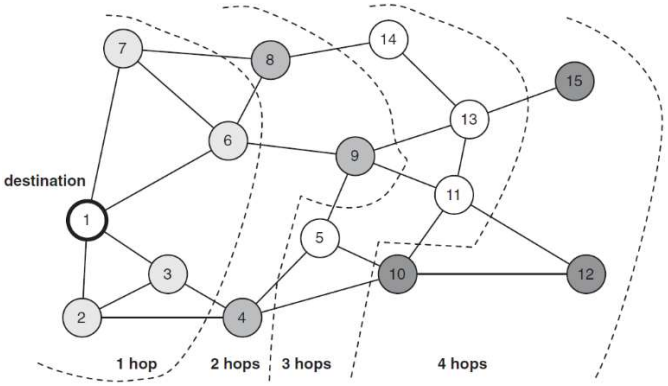

Figura 80. Rutare pe baza numărului de hopuri

- **Numărul de retransmisii**. Caracterizează fiecare hop în parte, pachetele ce sunt transportate prin rețea necesitând uneori retransmitere de către nodurile transportoare, de obicei datorită interferențelor sau legăturilor mai slabe pe distanțe mari, ce pot cauza pierderi de pachete. Precizia nu este foarte bună datorită fluctuațiilor în timp, folosindu-se adesea valori statistice ale numărului de retransmisii.

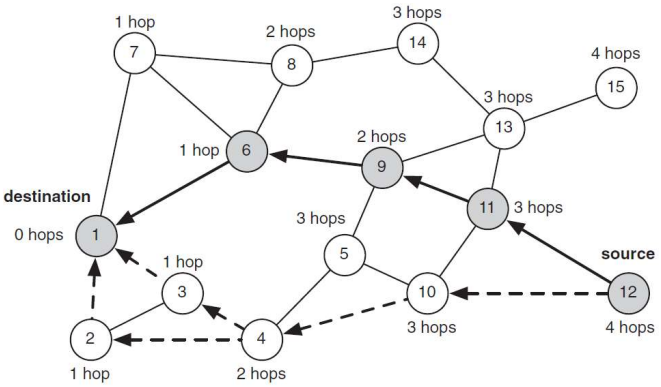

Figura 81. Rutare pe baza numărului de retransmisii

- **Timpul de întârziere**. Reprezintă timpul trecut de la trimiterea unui pachet de către nodul sursă până la recepția acestuia de către nodul destinație, indiferent de motivul întârzierii. Aceasta se poate datora metodelor de acces la mediu de tip TDMA, procesărilor la nivel de aplicație sau a comunicațiilor nesigure.

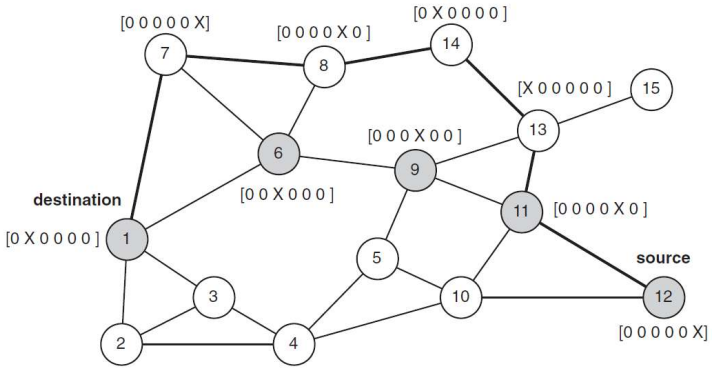

Figura 82. Rutare pe baza timpului de întârziere

9.8 Aplicații ale rețelelor de senzori fără fir

- **Monitorizarea mediului înconjurător**: condiții meteo, habitate naturale.
- **Clădiri inteligente**: monitorizarea umidității, temperaturii, calității aerului.
- **Monitorizarea conductelor**: detecția scurgerilor, măsurarea nivelurilor, presiunii, a calității apei.
- **Monitorizarea stării de sănătate**: pentru boli ca Parkinson, epilepsie, probleme cardiace.
- **Agricultură de precizie**: monitorizarea solului, culturilor, climei.
- **Managementul lanțului de producție**: urmărirea coletelor sau containerelor.
- **Monitorizarea vulcanilor**: temperatură, mișcare, nivel gaze.
- **Transporturi**: detecția vehiculelor, starea drumurilor.
- **Mineritul în subteran**: deplasări ale pereților minei, calitatea aerului, incendii
- **Monitorizarea stării structurale a construcțiilor**: poduri, baraje.
- **Detecția dezastrelor naturale**: incendii, alunecări de teren, cutremure.
- **Aplicații militare**: supraveghere, detecția armelor biologice sau chimice.

Capitolul 10. Internet of Things

Sintagma „Internet of Things” – IoT (“Internetul obiectelor/lucrurilor”) denumește o rețea de obiecte fizice „inteligente” (*smart objects*) interconectate, care au încorporată tehnologia necesară pentru a fi capabile de a sesiza și comunica date despre starea lor internă, precum și de a interacționa cu aceasta și cu mediul extern.

Se apreciază că până în anul 2020 numărul diverselor obiecte inteligente interconectate va tinde spre circa 200 de miliarde, de la tablete, telefoane, televizoare și alte obiecte electrocasnice inteligente, la dispozitive capabile să monitorizeze și să transmită parametrii de sănătate și mobilitate a oamenilor sau animalelor, de calitate a apei sau aerului, sau la dispozitive și elemente de monitorizare și control al parametrilor unor echipamente industriale complexe sau al produselor aflate în containere pentru livrare.

Schimbarea de viziune pe care o aduce IoT constă în:

- extinderea diversității acestor obiecte prin standardizarea soluțiilor de comunicare și interacțiune;
- valorificarea datelor privind evoluția stării acestor obiecte,
    - prin achiziția, transmiterea și stocarea lor în infrastructuri dedicate, centralizate sau de tip *cloud*, precum și
    - prin analiza avansată a acestora, utilizând servicii specializate, pentru a extrage, sintetiza și utiliza informația relevantă.

În ultimii ani, au fost dezvoltate numeroase soluții de interconectare a obiectelor inteligente în sisteme cu diferite scale și obiective. Se vorbește deja despre orașe inteligente (*smart cities*), case inteligente (*smart homes*), monitorizare digitală a sănătății (*digital health*), a unor procese industriale sau a poluării mediului etc.

Principalele funcționalități oferite de sistemele IoT pot fi grupate în 5 categorii, astfel:

- culegerea și pregătirea datelor;
- conectivitate, protocoale de comunicații;
- servicii de monitorizare, control și descoperire dispozitive;
- autentificare, autorizare, controlul integrității și securitatea datelor;
- analiza și procesarea datelor, asigurarea interfeței utilizator pentru acces la funcțiunile sistemului.

Se estimează că următorii ani vor însemna o perioadă de avânt spectaculos pentru sistemele IoT, ale căror complexitate și potențiale utilizări vor crește semnificativ. În Strategia Națională de Cercetare-Inovare 2014-2020, IoT este vizat în 3 din cele 4 subdomenii prioritare de dezvoltare a TIC ca domeniu de specializare inteligentă: în primul rând în „Internetul viitorului”, dar și în „Analiza și securitatea datelor de mari dimensiuni” și „Calculul de înaltă performanță și noi modele computaționale”.

10.1 Standardizarea sistemelor de tip IoT

IEEE descrie IoT, într-un raport despre Internet of Things din martie 2014, ca fiind o rețea de elemente, fiecare încorporând senzori, ce sunt conectate la Internet. Un proiect al organizației numit IEEE P2413 definește arhitectura cadru, abstractizările, adresează descrierea diverselor domenii de aplicabilitate și identificarea punctelor comune între acestea. Scopul acestui proiect este:

- să accelereze creșterea pieței IoT, permițând interacțiunea între domenii și unificarea platformelor prin creșterea compatibilității sistemelor, a interoperabilității și a substituirii funcționale;
- să definească o arhitectură cadru a IoT care acoperă nevoile arhitecturale ale diferitelor domenii ale aplicațiilor IoT;
- să crească transparența arhitecturilor de sistem pentru a sprijini evaluarea acestuia, siguranța și securitatea;
- să reducă fragmentarea industriei;
- să îmbunătățească cadrul de lucru existent.

IEEE P2413 definește în momentul de față o arhitectură bazată pe trei niveluri (Figura 83).

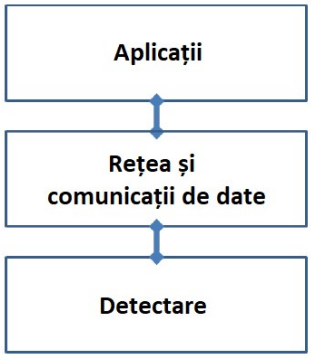

Figura 83. Arhitectura IEEE a sistemelor IoT

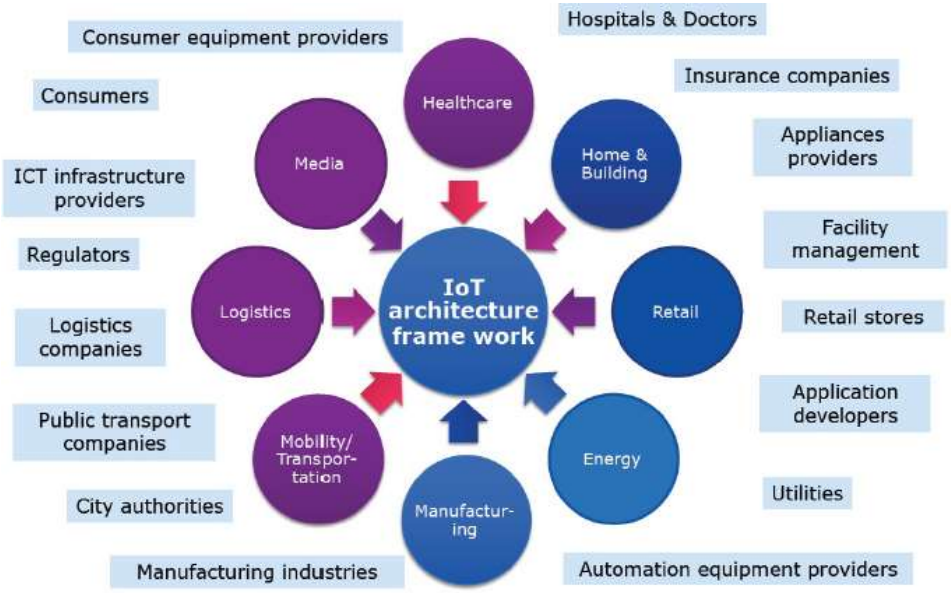

Figura 84. Piața sistemelor IoT și actorii implicați

O altă organizație de standardizare, ETSI, nu menționează cuvintele Internet of Things dar discută un concept similar numit *comunicație mașină-mașină* (M2M) ce reprezintă comunicația între două sau mai multe entități care nu necesită neapărat o intervenție umană. Serviciile M2M au scopul de a automatiza procesele de decizie și comunicație. Arhitectura de bază este prezentată în Figura 85.

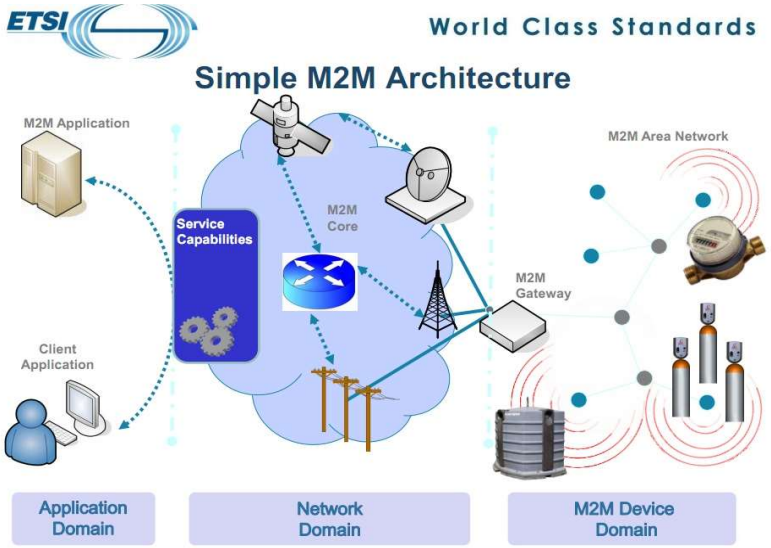

Figura 85. Arhitectura ETSI pentru comunicații M2M

Dispozitivele M2M rulează aplicații de bază necesare furnizării unor servicii și se conectează la rețeaua de acces fie **direct** (putând furniza servicii și altor dispozitive conectate la acesta), fie prin intermediul unui ***proxy*** de rețea sau ***gateway***.

Rețeaua de acces permite dispozitivelor M2M și celor de tip *gateway* să comunice cu rețeaua centrală, aceasta furnizând conectivitatea IP, funcțiile de control al rețelei și al serviciilor și interconectarea cu alte rețele.

Aplicațiile de furnizare a serviciilor sunt accesibile, de obicei, prin interfețe deschise.

ITU, agenția pentru tehnologiile informației și comunicațiilor, definește IoT ca o rețea **disponibilă oriunde, oricând, pentru orice sau oricine** (Figura 86).

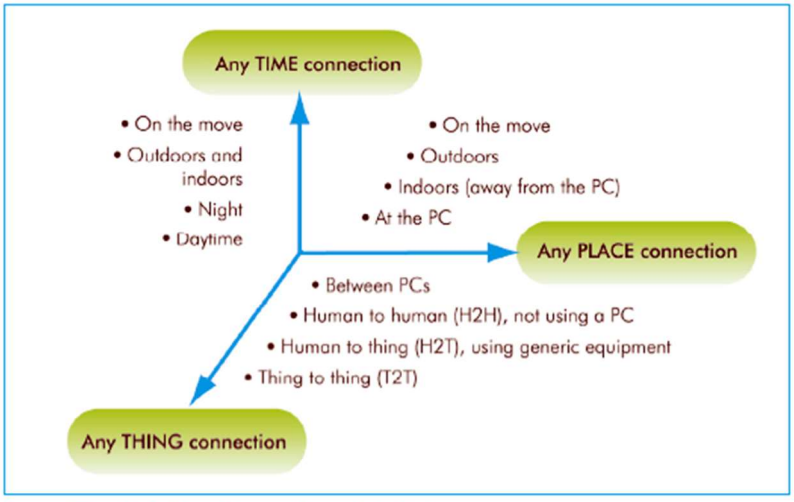

Figura 86. Definiția IoT a ITU

Sunt descrise și tehnologiile necesare pentru a putea fi realizate sistemele de tip IoT: RFID pentru etichetarea obiectelor, utilizarea senzorilor pentru obiectele care ”simt”, implementarea inteligenței pentru a face obiectele să ”gândească” și folosirea de nanotehnologii pentru micșorarea obiectelor.

10.2 Arhitectura unui sistem IoT

Din punct de vedere hardware, un sistem IoT poate conține:

- Senzori / elemente de acționare
- Unități de procesare
- Unități de stocare
- Unități de comunicație

Scenariul generic al unui sistem IoT constă din nevoia unui utilizator de a interacționa cu o entitate fizică, posibil aflată la distanță. Utilizatorul se definește ca o persoană sau un fel de entitate digitală activă (serviciu, aplicație, etc.) care urmărește un scop ce poate fi atins numai prin interacționarea cu mediul fizic. Entitatea fizică poate fi definită ca fiind o parte discretă, identificabilă a acestuia, ce poate lua orice formă, de la oameni și animale până la vehicule, containere, calculatoare sau electrocasnice. Entitățile fizice sunt reprezentate în mod digital prin entități virtuale (modele, elemente dintr-o bază de date, obiecte într-un limbaj de programare, etc.).

Dispozitivele dintr-un sistem IoT sunt utilizate pentru a asocia entitatea virtuală cu cea fizică. Aceasta se realizează prin încorporarea, atașarea sau plasarea acestora în apropierea entității fizice. Dispozitivele furnizează interfața tehnologică necesară interacționării cu sau obținerii de informații despre entitatea fizică.

Dispozitivele pot fi de mai multe feluri:

- Etichete, ce permit identificarea unică a unui lucru conectat la Internet.
- Cititoare, care citesc datele din etichete.
- Senzori, ce furnizează informații despre entitatea fizică pe care o monitorizează.
- Elemente de acționare, care pot modifica starea fizică a unei entități fizice.

Elementele unui sistem IoT funcționează pe baza unui sistem de operare. Ideală ar fi utilizarea de sisteme de operare complet dezvoltate (Linux, Unix, Windows) însă cerințele minime ale acestora nu îndeplinesc cerințele stricte ale dispozitivelor IoT ce funcționează pe baza unor microcontrolere de mică putere.

Inițial au fost utilizate sisteme de operare dedicate rețelelor de senzori fără fir ce puteau fi de două feluri: bazate pe evenimente (temporizare, întrerupere din exterior) sau cu mai multe fire de execuție (*thread*, în care sistemul de operare multiplexează în timp execuția mai multor sarcini). Două exemple de astfel de sisteme de operare sunt TinyOS sau Contiki.

Deoarece s-a considerat că cerințele unui sistem IoT diferă de cele ale unei rețele de senzori fără fir, au fost dezvoltate sisteme de operare dedicate, precum RIOT, care îndeplinesc următoarele:

- Cerințe minime de memorie și putere de procesare

- Abilitatea de a rula pe sisteme afectate de constrângeri hardware
- Suport pentru o mare varietate de platforme hardware
- Eficiență energetică mare
- Interfață de programare standard
- Suport pentru limbaje de programare de nivel înalt
- O stivă de rețea adaptivă și modulară
- Fiabilitate

O comparație între diferite sisteme de operare se poate observa în tabelul următor.

SO	Min RAM	Min ROM	Suport C	Suport C++	Multi Thread	Modula-ritate	Timp real	IPv6	TCP
Contiki	<2 kB	<30 KB	Parțial	Nu	Parțial	Parțial	Parțial	Da	Parțial
TinyOS	<1 kB	<4 kB	Nu	Nu	Parțial	Nu	Nu	Nu	Parțial
Linux	~1 MB	~1 MB	Da	Da	Da	Da	Parțial	Da	Da
RIOT	~1,5 kB	~5 kB	Da	Da	Da	Da	Da	Da	Da

Resursele sunt componente software care furnizează informație despre entitățile fizice sau permit controlul dispozitivelor. Pot fi localizate la nivelul dispozitivelor (software instalat local) sau disponibile undeva în rețea. Stocarea este un tip special de resursă ce memorează informațiile despre entitățile fizice.

Identificarea ”lucrurilor” în Internet se face pe baza adreselor, cele mai utilizate fiind de tip IPv6. În plus, a fost dezvoltat un alt sistem de adresare dedicat sistemelor IoT bazat pe adrese EPC (Cod Electronic de Produs) ce conțin 64 sau 94 de biți și sunt administrate la nivel global de EPCglobal.

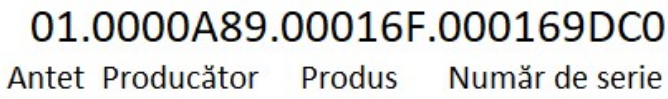

Figura 87. Adresă de tip EPC

O comparație între cele două tipuri de adrese se poate observa în tabelul următor.

	IPv6	EPC
Obiectul identificat	Interfața de rețea	Obiectul fizic
Aplicație primară	Adresă de rutare	Indicator al informației
Adresă alocată de	Administratorul rețelei	Producătorul obiectului
Adresă unică	Da	Da
Lungime adresă (biți)	128	64, 96 sau alta
Se poate schimba adresa?	Da	Nu
Sfera de dificultate	Mobilitatea	Lipsa informației despre localizare

10.3 Criterii de evaluare a platformelor IoT

Există astăzi o abundență de soluții de platforme IoT ce oferă conectivitate la Internet pentru senzori și elemente de acționare (*actuators*), permițând utilizatorilor finali să interacționeze cu obiectele inteligente dotate cu astfel de senzori și/sau elemente de acționare. Particularitățile și nivelurile tehnologice pe care se bazează, dar și măsura în care aceste platforme sunt capabile să satisfacă cerințele și așteptările diferiților actori din ecosistemul IoT (furnizori de dispozitive, dezvoltatori de aplicații, utilizatori finali etc.), pot constitui atât criterii de evaluare / comparare a lor, cât și elemente pe baza cărora se pot estima și direcțiile de evoluție ale acestora.

a) Tipurile de dispozitive suportate de platformă: platformele care necesită o poartă de acces (*gateway*) proprietară pentru conectarea dispozitivelor IoT sunt dependente de furnizorii platformelor respective pentru a adopta noi protocoale și a extinde varietatea și numărul de dispozitive IoT eterogene.

b) Tipul platformei IoT: în cele mai multe cazuri platformele sunt furnizate dintr-un *cloud*, fie ca Platform as a Service (PaaS), fie ca Software as a Service (SaaS). În cazul PaaS, platformele furnizează servicii de *cloud computing* pentru dispozitive IoT și date (de ex. de stocare, management dispozitive, conectivitate dispozitive, mecanisme de backup sau suport online), în timp ce, în cazul SaaS, accentul este pus pe interconectarea surselor de date utilizând capabilitățile *cloud computing*.

c) Tipul arhitecturii: arhitecturile de tip centralizat sunt specifice soluțiilor independente, în timp ce cele de tip descentralizat includ mai multe subrețele de senzori și dispozitive de acționare, fiecare controlată independent (de ex. acestea sunt denumite „site-uri” în cazul LinkSmart sau „hub-uri” în cazul OpenIoT).

Corelația între caracteristicile b) și c) este importantă pentru acoperirea unei mari diversități de cerințe de implementare. Astfel, o soluție de tip PaaS cu arhitectură descentralizată este adecvată pentru un sistem IoT la nivel de locuință, în timp ce o soluție centralizată bazată pe *cloud* este recomandabilă în cazul unor rețele de mari dimensiuni de senzori și elemente de acționare.

d) Gradul de deschidere: platformele *open source* sunt considerate mai de perspectivă decât alternativele proprietare deoarece permit o mai rapidă integrare a noilor soluții IoT pentru diverse domenii de aplicație și totodată s-a constatat că utilizarea de soluții *open source* accelerează adoptarea unei tehnologii software în manieră *bottom-up*. Totodată, s-a observat că soluțiile *open source* generează efecte economice benefice mai mari pentru domeniile aplicative în care sunt utilizate decât platformele proprietare.

e) Controlul accesului la date: este un criteriu relevant pentru platformele care nu arhivează datele local și implementează diverse niveluri de control al accesului la distanță. Există mai multe niveluri de granularitate privind controlul accesului, pornind de la accesul de tip privat / public, până la un control mai nuanțat al accesului atunci când datele pot fi private, protejate, publice sau anonime. Acest din urmă caz conferă flexibilitatea necesară pentru maximizarea reutilizării datelor de către servicii la distanță ale unor terțe părți.

f) Securitatea și confidențialitatea: implementarea unor asemenea mecanisme reprezintă un criteriu fundamental de evaluare pentru platformele IoT. Platformele IoT bazate pe *cloud* sunt predispuse la atacuri de securitate web și de rețea tradiționale, ca de exemplu: refuzul serviciului (*denial of service* – DoS), *man-in-the-middle*, *eavesdropping*, *spoofing* și *controlling*. Pentru asigurarea securității și confidențialității, atât în scenarii centralizate cât și distribuite, sunt necesare protocoale de nivel scăzut (*low level*). Pentru asigurarea securității în cazul unor configurații extinse de dispozitive încorporate trebuie avută în vedere depășirea limitărilor unor astfel de dispozitive (de ex. memoria, puterea de procesare, comunicația, timpii de răspuns, consumul de energie).

10.4 Platforme de tip IoT

Amazon Web Services (AWS) IoT

(https://aws.amazon.com/iot/)

Furnizează ca principale facilități: conectarea facilă a senzorilor pentru o mare varietate de aplicații, pe baza unui kit de dezvoltare software (SDK) pentru diapozitivele suportate, monitorizarea stării dispozitivelor IoT, poartă de acces (gateway) securizată, motor bazat pe reguli pentru evaluarea mesajelor de date recepționate. Există parteneriate cu firme producătoare de dispozitive și echipamente IoT (Intel, Texas Instruments, Broadcom, Qualcomm) pentru crearea de kit-uri compatibile cu platformele acestora.

Microsoft Azure IoT Suite

(https://www.microsoft.com/en-us/internet-of-things/azure-iot-suite)

Include următoarele facilități: monitorizarea stării dispozitivelor IoT, motor bazat pe reguli pentru validare mesaje de intrare, registru de identități, analiză avansată în timp real a unor fluxuri masive de date prin Azure Stream Analytics.

ThingWorx

(https://www.thingworx.com/)

Este considerat soluție IoT leader pentru domeniul industrial, cu următoarele facilități: simplitate în conectarea dispozitivelor la platformă, decuplarea dezvoltării aplicațiilor de întreprindere de detaliile tehnice specifice IoT, partajarea resurselor platformei între dezvoltatori pentru reutilizare și creștere a productivității, soluții de învățare automată pentru Big Data Analytics, versiuni diferite de distribuție (bazate pe *cloud*, integrate în sisteme de întreprindere, autonome).

IBM Watson IoT

(http://www.ibm.com/internet-of-things/)

Este o platformă bazată pe *cloud* PaaS Bluemix pentru dezvoltarea facilă de aplicații, care furnizează: managementul dispozitivelor, comunicații sigure, administrarea schimburilor de date în timp real, capacitate de stocare / memorare date.

Cisco IoT Cloud Connect

(https://www.cisco.com/c/en/us/solutions/service-provider/iot-cloud-connect/index.html

Este orientată spre operatori de comunicații mobile și oferă conectivitate de date și de voce, management al ciclului de viață al SIM, controlul sesiunilor de comunicații IP, facturare și raportare.

Salesforce IoT Cloud

(http://www.salesforce.com/iot-cloud/)

Este centrată pe client, pentru care oferă generare de oferte de vânzare, generare de comenzi de service, notificare automată a clienților, analiză a stocurilor disponibile.

Carriots

(https://www.carriots.com/)

Este o platformă PaaS de dezvoltare și găzduire de aplicații IoT, precum și pentru dezvoltare de soluții M2M (*Machine to Machine*). Oferă facilități de management dispozitive, SDK pentru aplicații, API pentru managementul cheilor de securitate, export date către alte aplicații, managementul utilizatorilor, un tablou de control pentru utilizatori.

Oracle Integrated Cloud IoT

(https://cloud.oracle.com/iot)

Oferă analiză în timp real a datelor IoT, virtualizare dispozitive, managementul punctelor de colectare date, mesagerie de viteză mare, notificarea utilizatorilor cu privire la dispozitivele lor.

General Electric's Predix

(https://www.ge.com/digital/predix)

Este o platformă PaaS pentru suport decizional în timp real, prin dezvoltare de aplicații IoT în sectoare prioritare (sănătate, transporturi, aviație, energie). Predix Machine este destinat nivelului de comunicații între senzori, controlere, platforma *cloud* și soluțiile de Analytics. Firma colaborează cu Intel, Verizon și Cisco pentru producerea de senzori compatibili Predix.

Kaa

(http://www.kaaproject.org/)

Este o soluție *open source* destinată să scurteze procesul și să reducă costurile de dezvoltare a soluțiilor IoT. Poate fi utilizată și ca platformă de găzduire pentru o varietate de aplicații IoT. Poate administra un număr foarte mare de dispozitive.

10.5 Aplicații ale IoT

10.5.1 Orașe inteligente

Un oraș este inteligent dacă integrează tehnologiile informației și comunicațiilor pentru utilizarea eficientă a resurselor și infrastructurii în scopul asigurării necesităților cetățenilor săi. Utilizarea tehnologiilor inovative are un impact pozitiv asupra calității vieții cetățenilor, protejării mediului, dezvoltării mediului de afaceri și dezvoltării durabile a comunităților locale și societății în general.

Figura 88. Domeniile de aplicabilitate ale IoT pentru orașe inteligente

Telecomunicațiile reprezintă elementul de bază și trebuie urmărită utilizarea tehnologiilor moderne astfel încât să fie asigurată o lărgime de bandă suficientă pentru funcționarea corespunzătoare a echipamentelor și aplicațiilor.

Figura 89. Utilizarea telecomunicațiilor în orașele inteligente

Un alt domeniu de aplicabilitate al IoT este cel al energiei electrice, prin transformarea rețelelor de distribuție a energiei electrice în rețele inteligente (*smart grid*) ce pot integra în mod inteligent comportamentul și acțiunile tuturor utilizatorilor conectați la acestea – generatoare, consumatori – pentru asigurarea unui proces sustenabil, economic și sigur.

Câteva dintre avantaje sunt utilizarea eficientă a energiei (telecontorizare, iluminat public inteligent) reducerea timpului de remediere a defecțiunilor și cheltuielilor de mentenanță.

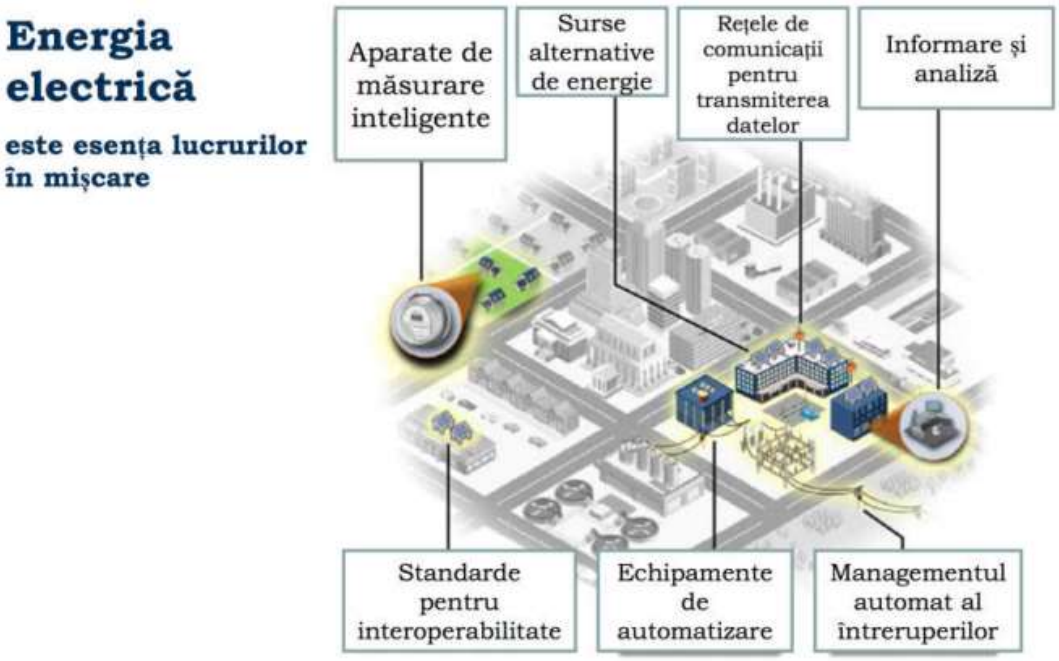

Figura 90. Utilizarea IoT în domeniul energiei electrice

În ceea ce privește rețeaua de alimentare cu apă potabilă și de canalizare, sistemele IoT pot contribui la eliminarea pierderilor, telecontorizare, managementul consumului și tarifarea secvențială, reducere timpului de remediere a defecțiunilor și cheltuielilor de mentenanță, reducerea cantității de apă consumate, controlul gradului de poluare, irigarea controlată, inteligentă a spațiilor verzi, etc.

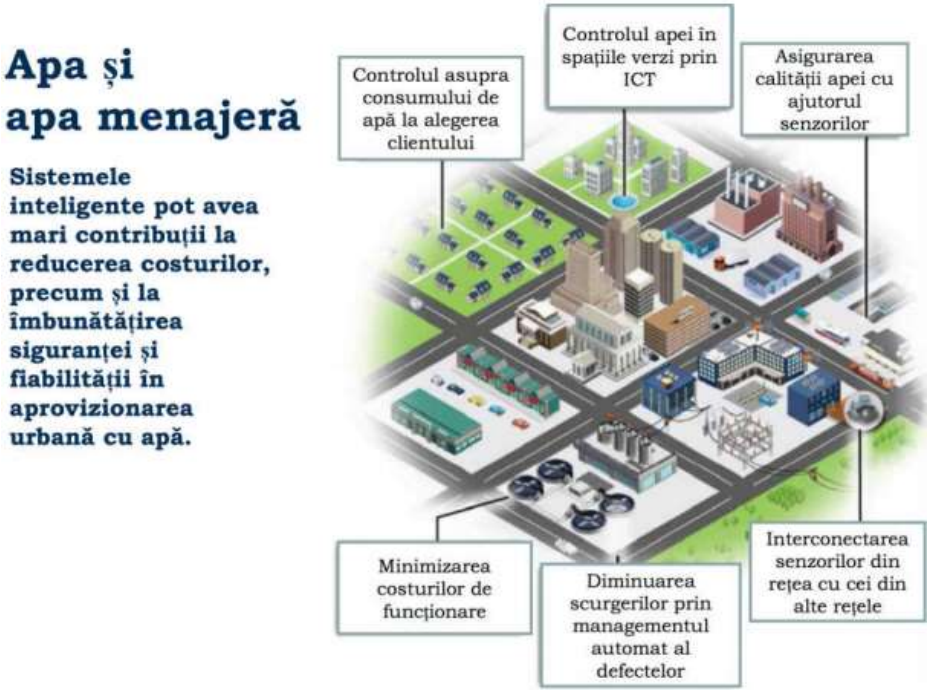

Figura 91. Utilizarea IoT în rețelele de apă și canalizare

În ceea ce privește rețeaua de transport, sistemele IoT pot ajuta la reducerea costurilor, a timpilor de așteptare în stații, a timpilor de călătorie, a nivelului poluării, fluidizarea traficului, adaptabilitatea flotei de vehicule de transport în comun, intervenția rapidă în caz de defecțiuni, managementul inteligent al parcărilor, etc.

Figura 92. Utilizarea IoT în rețeaua de transport

Sistemele IoT pot fi utilizate în domeniul siguranței publice pentru supravegherea și monitorizarea unor zone precum unitățile de învățământ, locurile de joacă, străzi și intersecții, detectarea incidentelor precum ambuteiaje, aglomerări umane neautorizate, probleme în trafic, apelarea și prioritizarea serviciilor de urgență, avertizarea cu privire la situații climaterice sau de mediu deosebite, etc.

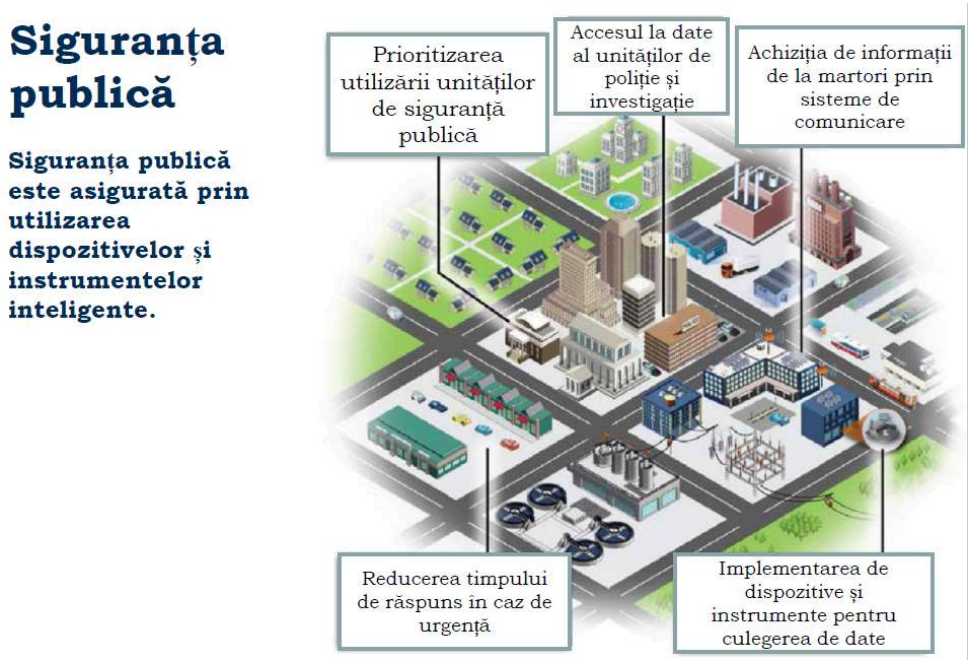

Figura 93. Utilizarea IoT în siguranța publică

Implementarea sistemelor IoT în domeniul medical va duce la dezvoltarea continuă a tele-medicinei (totalitatea sistemelor care ajută la procesul de îngrijire a sănătății prin schimbul eficient de informație medicală), va permite monitorizarea la distanță, optimizarea fluxului de pacienți, controlul costurilor și a calității serviciilor, realizarea de intervenții chirurgicale robotizate, partajarea de informații între pacient și furnizorii de servicii medicale.

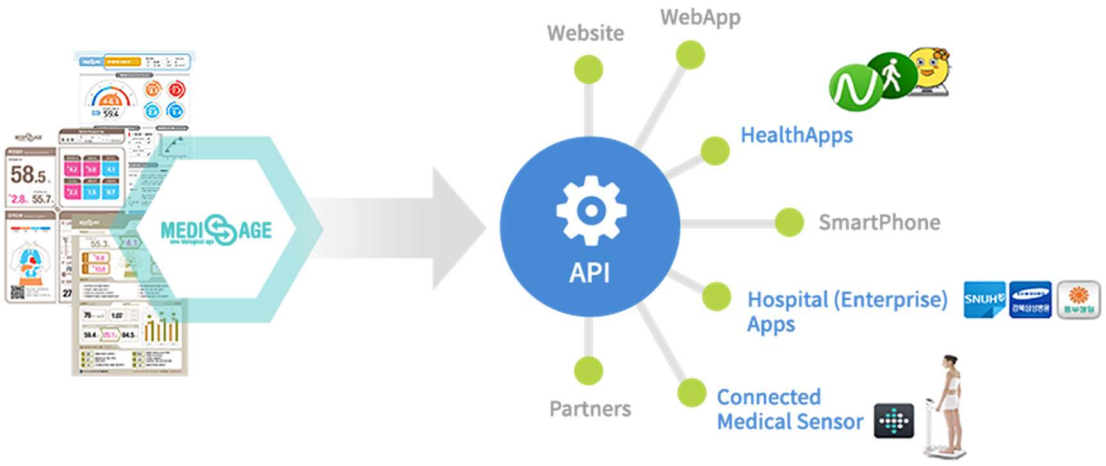

Figura 94. Utilizarea IoT în domeniul medical

10.5.2 Case și clădiri inteligente

Casa sau clădirea inteligentă oferă utilizatorilor acesteia un mediu confortabil și productiv cu ajutorul echipamentelor automate de măsură, comunicare și control, integrate în sistemele care o alcătuiesc.

Câteva elemente a căror prezență este obligatorie într-o casă sau clădire inteligentă: senzori care culeg informații diverse, elemente de acționare care permit comanda sistemelor instalate în casă, o rețea de comunicație și o unitate centrală care rulează programe ce memorează și monitorizează informațiile, ia decizii și emite comenzi în funcție de aceste decizii. În plus, cu ajutorul unor interfețe hardware sau software, proprietarul casei poate accesa toate informațiile culese de unitatea centrală, poate apela toate comenzile casei individual sau grupate în scenarii și poate configura modul în care sistemul ia decizii. Deciziile sunt cele a căror complexitate împarte casele în case inteligente sau doar automatizate.

Figura 95. Utilizarea IoT în casele inteligente

Un sistem IoT pentru case inteligente poate include:

- Sistem de control al iluminării, controlul tuturor luminilor prin intermediul întrerupătoarelor clasice, automat prin definirea de scenarii, de exemplu: a apus soarele, aprinde lumina în curte etc, automat prin senzorii de mișcare, lumina "se ține" după proprietar, eficientizarea consumului de energie electrică prin închiderea automată a luminilor aprinse atunci când se armează alarma pe toată casa etc. Controlul luminilor se poate realiza și de la distanță prin intermediul telefonului, tabletei sau a laptopului.
- Sistem control automat al temperaturii prin intermediul senzorilor de temperatură, umiditate amplasați în camere. Se pot defini scenarii gen: "am plecat", casa trece în conservare, "mă întorc", casa revine la funcționalitatea normală pentru a oferi un maxim de confort, iar comenzile "mă întorc" și "am plecat" se pot realiza și automat în cazul în care proprietarul are un sistem de localizare pe mașină care comunică poziția acesteia către casa inteligentă etc. Setarea temperaturilor se poate realiza și de la distanță prin intermediul telefonului, tabletei sau a laptopului.
- Sistem de monitorizare consumuri/producție energie electrică din panouri solare fotovoltaice, sistem ce eficientizează consumul, comutând consumatorii în funcție de producția de energie realizată de panouri.
- Senzori pentru evenimente cum ar fi inundație, incendiu, efracție. Pot fi definite scenarii ca in cazul declanșării senzorului de inundație, apa sa fie oprită automat, instalațiile golite, proprietarul anunțat prin sms/email de eveniment și dacă este definit în sistem anunțat și instalatorul să intervină etc.
- Senzori pentru evenimente zilnice/sezoniere cum ar fi zi/noapte, îngheț, ploaie etc. Se pot defini scenarii pentru diverse acționări în funcție de aceste evenimente, pornire degivrări, aprins lumini, udat gradina etc.

Capitolul 11. Cloud computing

11.1 Introducere

Termenul „Cloud Computing” se referă la stocarea, procesarea și utilizarea de date pe sisteme aflate la distanță și accesate prin intermediul Internet-ului. Aceasta înseamnă că utilizatorii pot dispune la cerere de o putere de calcul aproape nelimitată, că nu trebuie să facă investiții de capital majore pentru a-și satisface exigențele și că își pot accesa datele din orice loc, cu ajutorul unei conexiuni la Internet.

Cloud Computing-ul are potențialul de a reduce cheltuielile IT ale utilizatorilor și de a favoriza dezvoltarea unui număr mare de noi servicii. Utilizând Cloud Computing-ul, chiar și cele mai mici întreprinderi se pot adresa unor piețe din ce în ce mai mari, iar administrațiile pot spori atractivitatea și eficiența serviciilor lor, ținând în același timp sub control cheltuielile.

În comparație cu World Wide Web ce pune informația la dispoziția tuturor, în orice loc din lume, Cloud Computing-ul pune puterea de calcul la dispoziția tuturor, în orice loc din lume. Ca și web-ul, Cloud Computing-ul reprezintă o tehnologie inovatoare care a apărut în urmă cu ceva timp și care continuă să se dezvolte. Diagrama conceptuală a Cloud Computing-ului este prezentată în Figura 96.

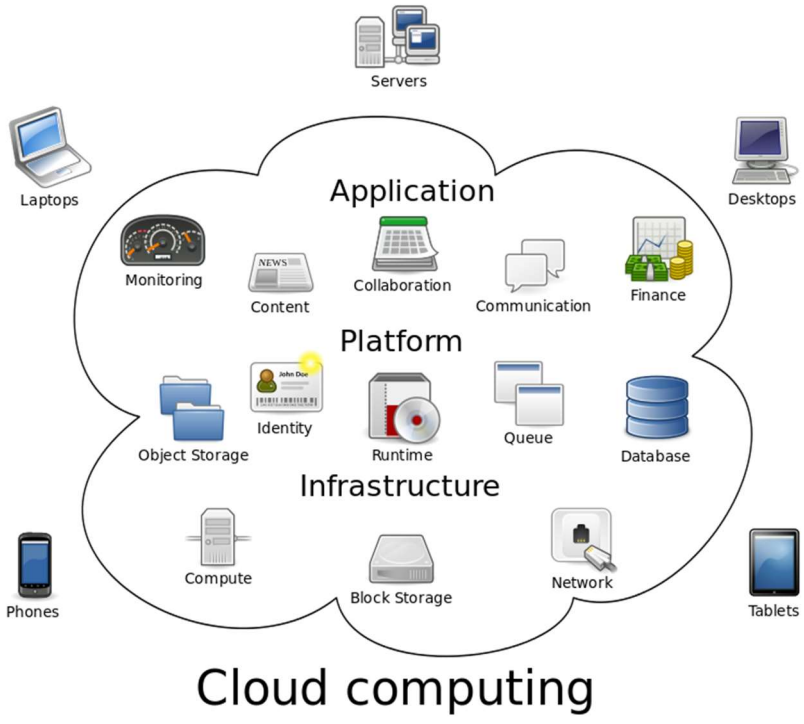

Figura 96. Diagrama conceptuală a Cloud Computing

Chiar dacă acest concept a fost explorat încă din anii 1960, nu s-au făcut progrese foarte mari până când nu au fost făcute investiții uriașe în infrastructura rețelelor din anii 1990. Consumul mare de lățime de bandă pe care îl generează traficul pe Internet în zilele noastre este ceea ce ajută ca această tehnologie să funcționeze, în timp ce rețelele anterioare erau mult mai

puțin dinamice din cauza vitezelor mici de încărcare și descărcare care erau disponibile în acele vremuri.

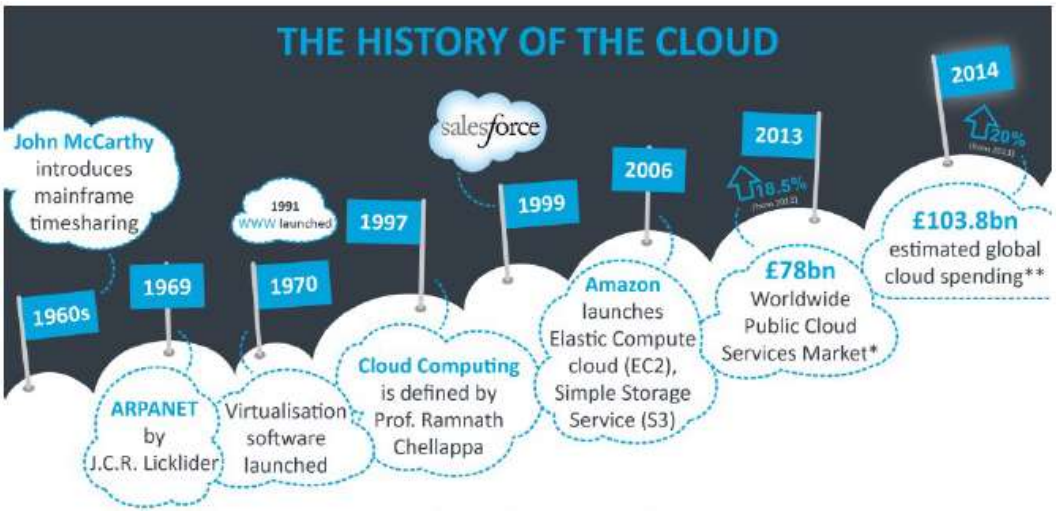

Figura 97. Evoluția Cloud Computing-ului în istorie

Economia globală a produs o schimbare majoră și s-a trecut de la conceptul de dezvoltare prin fabricare de echipamente hardware la o eră nouă în care prelucrarea informației este predominantă. Informația în sine va deveni o comoditate așa cum sunt lucrurile fabricate astăzi, iar o fermă de servere ar putea fi considerată ca echivalentul modern al unei fabrici. Această nouă „fabrică" este motorul din spatele creșterii puterii de prelucrare a datelor și posibilitatea de a reduce costurile, informația devenind produsul viitorului.

În ultima vreme tot mai multe companii renunță la a-și mai construi propriile infrastructuri IT, alegând ca alternativă să-și depoziteze bazele de date și aplicațiile software sau alte tipuri de informații utilizând servicii de tip Cloud, astfel având acces la informațiile stocate prin intermediul Internetului. Serviciile de tip Cloud au luat amploare datorită mobilității, disponibilității și a prețului scăzut, însă folosirea acestui tip de servicii aduce cu sine și o serie de amenințări de securitate la adresa datelor și informațiilor.

Una dintre amenințările cele mai mari provine chiar din natura conceptului și anume faptul că permite datelor să fie trimise și salvate aproape oriunde, în unele cazuri datele fiind stocate în locații diferite. Deși dispersia datelor îmbunătățește performanțele și reduce costurile, dezavantajul constă în faptul că datele pot ajunge în locații în care legile privind confidențialitatea sunt slabe sau, în unele cazuri, nici nu există.

Principalul beneficiu al conceptului din spatele tehnologiei Cloud Computing este acela că utilizatorul nu are nevoie de un computer extrem de performant pentru a prelucra de exemplu indexări ale unor baze de date foarte complexe, sarcini grele pe care o „fermă" de servere le poate executa. În schimb, utilizatorii se pot conecta cu ușurință în acest mediu, care poate fi numit un punct de contact cu rețele foarte mari. Din acest punct utilizatorii din întreaga lume pot profita de avantajele puterii enorme de procesare fără a investi un capital imens sau să aibă cunoștințe tehnice.

Un concept fundamental al modelului Cloud Computing este acela că prin acest model tehnologia informației este pusă la dispoziția utilizatorilor sub formă de servicii (așa cum, prin

analogie, tehnologia de comunicații este pusă la dispoziția utilizatorilor sub formă de servicii de telefonie/voce, servicii de date etc.), utilizate sub forma unor subscripții periodice (așa cum serviciile de telefonie sunt utilizate sub formă de abonament lunar).

Sfârșitul primului deceniu al secolului 21 a fost descris ca fiind un „punct de cotitură istorică" în dezvoltarea serviciilor de e-guvernare și „trecerea spre maturitate". Mediul IT folosit în prezent în administrația publică, este caracterizat de fragmentarea accesului la resurse, sisteme duplicat, slabă utilizare a resurselor disponibile, proceduri de achiziții complicate, în general un mediu greu de administrat și controlat, cu efect imediat asupra calității serviciilor prestate de administrația publică către cetățeni. Un factor important în dezvoltarea serviciilor de e-Guvernare îl reprezintă Cloud Computing-ul, cu potențial de a juca un rol major în a adresa aceste ineficiențe și de a îmbunătății modul livrării serviciilor de către administrația publică. Modelul de Cloud Computing poate în mod semnificativ ajuta administrația publică prin servicii de înaltă disponibilitate, servicii inovative, accesibile imediat trecând peste bariera disponibilității resurselor specifice unui mediu IT tradițional.

În sectorul privat, furnizorii de Cloud Computing și-au extins oferta de servicii specifice, acum acesta incluzând întreaga stivă de servicii specifică IT-ului tradițional: infrastructura software și hardware, platforma *middleware*, componente de aplicații și servicii, aplicații la cheie. Sectorul privat a identificat acest avantaj de a folosi și oferi servicii de tip Cloud Computing, pentru a îmbunătății modul de utilizare al resurselor, disponibilitatea, accesul și inovativitatea.

11.2 Modele Cloud

Se pot identifica patru modele principale de folosință: Privat, Public, Hibrid, Comunitar, și un model derivat, Instituțional (Figura 97).

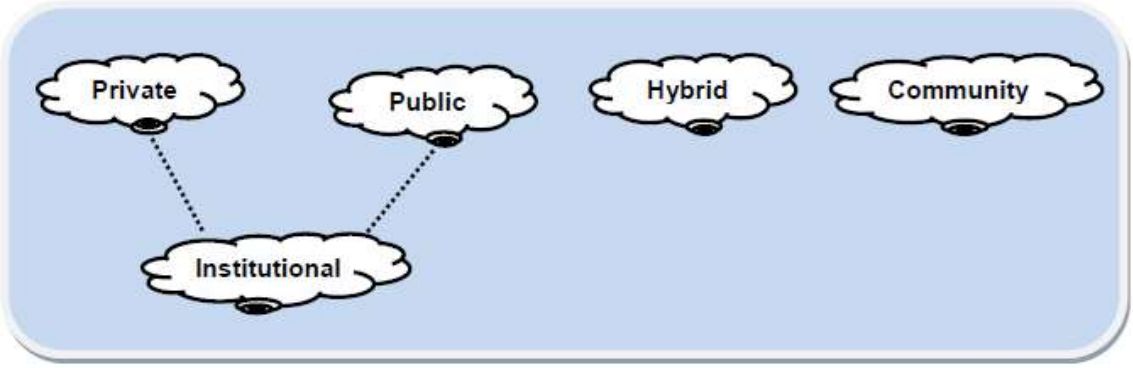

Figura 98. Modele Cloud

Modelele de Cloud folosite în organizații sunt în majoritatea cazurilor de tip Privat (Private) sau Hibrid (Hybrid), pe când furnizorii de servicii folosesc modelul de tip Public și Hibrid (Hybrid).

Cloud Privat (Private Cloud): Infrastructura IT este folosită de către o singură organizație formată din mai mulți consumatori și poate fi administrată de către organizația însăși sau externalizată către un terț. În *Private Cloud*, spre deosebire de un Centru de Date

tradițional, sunt optimizate resursele disponibile. Există multe organizații care au implementat propriul sistem de Cloud privat cum ar fi IBM, HP, Microsoft etc.

Cloud Public (Public Cloud): Infrastructura IT este disponibilă publicului sau unei părți a publicului în baza unor criterii, sau unui segment industrial sau zonă de interes. În cadrul acestui model infrastructura IT este deținută și administrată de către un furnizor de servicii (*service provider*) (organizație comercială, guvernamentală, academică sau mixtă). Serviciile pot fi accesate prin intermediul Internetului, iar protecția datelor este asigurată de furnizorul de servicii. Exemple de servicii de *Public Cloud*: Windows Azure Platform de la Microsoft, AWS de la Amazon, AppEngine și Gmail de la Google, etc.

Cloud Hibrid (Hybrid Cloud): Infrastructura IT este compusă din una sau mai multe componente Cloud de tip private sau publice care sunt considerate ca un întreg folosind aceeași tehnologie. Companiile pot rula aplicații în Cloud-ul public în timp ce datele și aplicațiile private sunt stocate în Cloud-ul privat.

Cloud Comunitar (Community Cloud): Infrastructura IT este partajată de către mai multe organizații pentru a asigura servicii unei anumite comunități, ce împărtășesc aceleași cerințe funcționale. În cadrul acestui model infrastructura IT este deținută și administrată de către una sau mai multe dintre organizațiile din comunitate, o terță parte sau o combinație a acestora și poate exista fizic în interiorul sau în afara organizației. Exemple: Google Apps for Government, Microsoft Government Community Cloud.

Cloud Instituțional (Institutional Cloud): Infrastructură cloud care este administrată de o organizație și folosită de mai mulți utilizatori. Administratorul, care este și furnizor de servicii folosește infrastructura și pentru activitățile proprii asigurând în același timp servicii de consultanță și mentenanță pentru clienți. Protecția comunicațiilor și a datelor este asigurată de administrator. Cloud-ul instituțional reprezintă o combinație între cloud-ul privat și cloud-ul public.

11.3 Servicii de tip Cloud Computing

În funcție de cerințele utilizatorilor, există mai multe soluții de implementare pentru Cloud Computing disponibile pe piață. Acestea sunt definite de NIST (National Institute of Standards and Technology) în trei categorii principale sau „modele de servicii” (Figura 98).

IaaS (*Infrastructure as a Service* – Infrastructură ca Serviciu): primul model care respectă caracteristicile Cloud Computing NIST (National Institute of Standards and Technology). În cadrul acestui model un furnizor (*service provider*) închiriază infrastructura IT, adică mașini virtuale aflate la distanță, care pot înlocui infrastructura IT din cadrul companiilor. IaaS include întreaga stivă de resurse de infrastructură oferind automatizări până la nivelul de virtualizare și oferind de asemenea facilități cum ar fi soluții de răcire, energie electrică etc. pentru platformele hardware găzduite. Furnizorul unor astfel de servicii este responsabil pentru gestionarea unor eventuale defecțiuni hardware.

Exemple de IaaS: Amazon Web Service (AWS), Google Compute Engine (GCE), Rackspace Open Cloud, IBM SmartCloud Enterprise, HP Enterprise Converged Infrastructure.

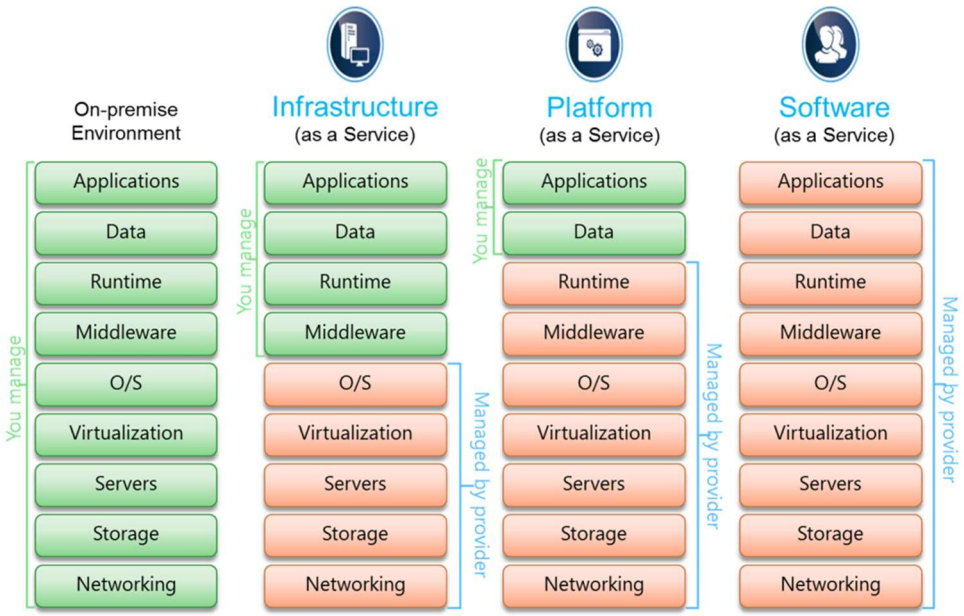

Figura 99. Modele de servicii în Cloud Computing

PaaS (*Platform as a Service* – Platformă ca Serviciu): furnizorul întreține și oferă componente pre-configurate inclusiv limbaje de programare, servere de aplicații și baze de date pentru dezvoltatorii de aplicații web. PaaS se află între IaaS și SaaS, acesta nefiind un produs finit care poate fi accesat direct de către utilizatorii finali, dar adaugă față de IaaS un nivel suplimentar de integrare cu *framework*-urile de dezvoltare de aplicații, capabilități de middleware și funcții, cum ar fi baze de date, stive de interogări.

Exemple de PaaS: Engine Yard, Red Hat OpenShift, Google App Engine, Heroku, AppFog, Windows Azure Cloud Services, Amazon Web Services AWS, Caspio.

SaaS (*Software as a Service* – Software ca Serviciu): cea mai comună formă de Cloud pentru utilizatorul obișnuit - furnizorul oferă aplicații la cerere - „*on-demand*" - utilizatorilor finali. Aceste aplicații, plătite de obicei pe bază de abonament, sunt găzduite și gestionate de către furnizorul de servicii și înlocuiesc aplicațiile tradiționale instalate de utilizatori pe echipamentele lor. Astfel de servicii sunt în mare măsură destinate aplicațiilor de birou bazate pe web cum ar fi poștă electronică, procesare de text, calcul tabelar, prezentări, calendare, agende etc.

Exemple de SaaS: Microsoft Office 365, Google Gmail, Google Docs, Zoho Office, Salesforce, Citrix GoToMeeting, Cisco WebEx.

Raportându-ne la cele trei modele de servicii (IaaS, PaaS, SaaS) prezentate anterior trebuie avut în vedere faptul că există compromisuri importante pentru fiecare, pe de o parte de caracteristici integrate, de complexitate și pe de altă parte de extensibilitate și de securitate. Compromisurile între cele trei modele de implementare Cloud includ:

- SaaS oferă cele mai integrate funcționalități, cu cea mai mică posibilitate de extensibilitate și un nivel relativ ridicat de securitate integrat (furnizorul poartă o parte importantă de responsabilitate pentru securitate);

- PaaS permite dezvoltatorilor să construiască propriile aplicații în zona superioară a platformei. Ca urmare, tinde să fie mai extensibil decât SaaS. Acest compromis se extinde la elementele de securitate, dar oferă tocmai datorită flexibilității posibilitatea integrării unui strat de securitate suplimentar;
- IaaS oferă cea mai multă extensibilitate. Acest lucru înseamnă că, în general, capacitățile de securitate și funcționalitățile nu trec dincolo de protejarea infrastructurii în sine. Acest model presupune că sistemele de operare, aplicațiile și conținutul vor fi gestionate și securizate de către consumatorul de astfel de servicii.

11.4 Caracteristici principale ale Cloud Computing-ului

Cel mai important element al Cloud Computing-ului este structura serverelor. Aceasta joacă un rol major deoarece este considerată creierul din spatele întregului mediu de procesare. În cazul Cloud Computing-ului, componentele fizice NU trebuie să aibă obligatoriu o performanță individuală extrem de mare. Mai degrabă beneficiul cheie al acestei tehnologii este posibilitatea ca o organizație să valorifice puterea de calcul a unor echipamente ieftine pe o scară mare, spre deosebire de cazul în care se folosesc un număr mai redus de servere de înaltă performanță.

Pentru o companie poate fi foarte util să beneficieze de capabilitățile de calcul pe care le oferă această tehnologie deoarece permite tuturor utilizatorilor să aibă acces la informații de oriunde ar fi și oricând ar avea nevoie, ceea ce poate ajuta la prevenirea pierderii datelor sau organizarea proastă a fișierelor digitale. De asemenea se permite ca o companie să aibă o structură împărțită în multe noduri localizate pe tot globul, organizare care devine din ce în ce mai populară deoarece companiile încearcă să se integreze la nivel global și să aibă mai multă flexibilitate în același timp.

Faptul că toate informațiile sunt găzduite într-un singur spațiu fizic permite administrarea mai eficientă a componentelor hardware și software de către un grup specializat care poate întreține mai ușor această arhitectură. Acest mod de administrare a tehnologiei hardware este exact beneficiul pe care companiile vor să îl aibă prin soluțiile de Cloud Computing deoarece este mult mai ieftin să desemneze pe altcineva care să gestioneze din punct de vedere tehnic această tehnologie, astfel încât personalul companiei să se poată concentra doar asupra afacerii.

Factorul principal la care se gândesc companiile este scalabilitatea sistemului și realizează că dacă altcineva se va ocupa doar de întreținerea acestor mari servere o va face mai rapid și mai eficient decât ar putea să o facă prin eforturi proprii. Dacă afacerea lor devine mai complexă, Cloud-ul va trebui să devină și el mai mare iar compania nu va trebui să aibă grija acestei probleme. Sunt multe probleme de care nu va mai trebui să se ocupe compania atunci când externalizează acest serviciu cum ar fi să pună la dispoziție un spațiu special amenajat pentru echipamentele de calcul, costurile asociate cu angajarea de tehnicieni sau achiziționarea periodică a unor componente hardware costisitoare pentru ca totul să decurgă fără probleme.

În Tabelul următor sunt prezentate caracteristicile utilizării modelului Cloud în comparație cu modelul tradițional.

Model Tradițional	Model „Cloud"
Fiecare entitate întreține propria infrastructura IT	Infrastructura este partajată și utilizată după necesități de mai multe entități
Sistemele sunt eterogene și complexe	Platforma este omogenizată, simplificată și controlată unitar
Gestiunea infrastructurii cade în sarcina responsabililor de procese	Infrastructura este virtualizată, optimizată și gestionată de un grup specializat
Nivel redus de suport disponibil din partea personalului autorizat	Nivel ridicat de suport în exploatare
Nivel de securitate redus și necesar pentru fiecare componentă a procesului	Securitate ridicată la nivelul întregului sistem
Utilizare intensivă a resurselor energetice pentru funcționarea unui număr ridicat de centre de date	Utilizare optimizată a resurselor energetice prin agregarea centrelor de date

Cloud Computing-ul prezintă o serie de caracteristici și avantaje:

- furnizorul de servicii de Cloud Computing are în gestiune sistemele și dispozitivele de stocare (hardware-ul) și nu utilizatorul, care interacționează cu acesta prin Internet;
- sistemele sunt virtualizate într-o rețea, iar utilizatorul nu cunoaște cu precizie locația exactă a datelor sau a proceselor, ci numai punctul de acces la infrastructură;
- utilizatorul poate modifica foarte ușor și rapid volumul hardware utilizat, ca de exemplu mărirea capacității de stocare;
- utilizatorul își poate accesa datele și utiliza programele atunci când are nevoie folosind un dispozitiv (calculator, laptop, tabletă, smartphone) conectat la Internet;
- sincronizarea datelor este simplificată pentru un utilizator care folosește mai multe dispozitive conectate la Cloud;
- furnizorul de servicii de Cloud poate migra anumite procese ale utilizatorilor pentru o mai bună optimizare a resurselor disponibile;
- utilizatorul plătește în funcție de cât a consumat, asemănător unui serviciu de utilitate publică (de exemplu serviciul de energie electrică), neavând costuri legate de configurarea și exploatarea sistemelor informatice.

Dezavantaje ale Cloud Computing-ului privesc câteva aspecte principale prezentate mai jos:

- Internet rapid și comunicații sigure - utilizatorul are nevoie de o legătură stabilă și rapidă la Internet;
- securitatea datelor – toate datele și înregistrările sunt la furnizor ceea ce poate duce la neîncrederea utilizatorului în păstrarea confidențialității și integrității acestora;

- atacurile nedorite – atacurile de tipul DDoS (Distributed Denial of Service) sunt mult mai frecvente în Cloud Computing;
- prelucrarea datelor cu caracter personal și libera circulație a acestor date (lipsa controlului utilizatorului asupra datelor respective și informații insuficiente cu privire la modalitatea, locul și entitatea de prelucrare/sub-prelucrare a datelor), utilizatorul nu știe în ce loc se găsesc datele sale, care pot fi în aceeași țară sau în străinătate;
- infrastructură concentrată – hardware, sofrware, date - în caz de incident major, utilizatorul poate pierde integral date și programe, dacă sistemul cloud-computing nu dispune de măsuri de siguranță specifice (data recovery, sisteme de back-up);
- cadru legal adecvat – cadrul legal de funcționare a sistemelor cloud nu este suficient de cuprinzător pentru a reglementa situații posibile, uneori nedorite.

11.5 Disponibilitate și fiabilitate

Disponibilitatea reprezintă aptitudinea unui sistem, subsistem, sau echipament de a se afla într-o stare specificată de operare, presupunând un mediu conform cu specificațiile de funcționare, indiferent de momentul de timp. Un sistem Cloud necesită o disponibilitate ridicată, de tipul "*Five Nines*", adică de 99.999% din timp.

Fiabilitatea este probabilitatea ca un sistem sau o componentă să-și îndeplinească funcțiile specificate sub anumite condiții pentru o anumită perioadă de timp.

Aceste proprietăți se pot asigura prin soluții de toleranță la defecte, de menținere a furnizării serviciilor în caz de defectare sau de securitate.

Toleranța la defecte este proprietatea sistemului de a continua să funcționeze corect chiar în prezența apariției unui defect în oricare dintre componentele acestuia. Patru aspecte sunt importante pentru asigurarea acesteia:

- Evitarea existenței unui punct unitar de defectare (*single point of failure*) ce reprezintă o parte a sistemului care, dacă cedează, conduce la oprirea completă a acestuia. Evaluarea locațiilor de apariție a defectelor poate arăta care sunt acele componente critice ale sistemului.
- Implementarea de soluții de detecție a defectelor (*fault detection*) și izolare a unei componente defecte cu ajutorul unui sistem de monitorizare ce realizează identificarea unui defect, specificarea tipului și locației acestuia.
- Prevenirea propagării deoarece unele defecte pot opri sistemul prin propagarea de la componenta defectă către alte componente, fiind necesare mecanisme pentru izolarea acesteia.
- Existența unor moduri de revenire în funcțiune.

Menținerea furnizării serviciilor în caz de defectare (*resilience*) reprezintă abilitatea de a furniza și menține un nivel acceptabil al serviciilor, chiar și în prezența defectelor sau întreruperii alimentării cu energie electrică. Aceasta este adesea realizată prin utilizarea

componentelor, subsistemelor sau sistemelor redundante, salvarea datelor *off-site* la intervale regulate. Atunci când un element eșuează sau suferă o întrerupere, elementul redundant prelucrează fără probleme și continuă să sprijine furnizarea serviciilor. În mod ideal, utilizatorii unui astfel de sistem nu știu niciodată că a apărut o perturbare.

Securitatea în Cloud reprezintă un sub-domeniu în continuă dezvoltare legat de securitatea calculatoarelor, a rețelelor de calculatoare, sau în general de securitatea informatică. Aspectele importante ale acesteia sunt protejarea datelor, gestiunea identităților pentru controlul accesului la informații și resurse computaționale, securitatea aplicațiilor și intimitatea.

Clustering. Load balancing. Parallel processing. Job scheduling. Virtualization.

11.6 Organizații de standardizare și reglementare

National Institute of Standards and Technology (NIST) (Institutul Național de Standarde și Tehnologie) – este o agenție federală fondată în 1901 și cunoscută între 1901-1988 ca National Bureau of Standards (NBS) (Biroul Național de Standarde), iar în prezent face parte din Departamentul de Comerț al SUA. Rolul NIST este de a susține economia și industria prin elaborarea de tehnologii și standarde și de a promova inovația și competitivitatea industrială prin avansarea științei, a standardelor și tehnologiilor astfel încât să sporească securitatea economică în folosul cetățeanului. În publicația specială „Special Publication 800-145" NIST definește Cloud Computing-ul atribuindu-i cinci caracteristici esențiale (*On-demand self-service, Broad network access, Resource pooling, Rapid elasticity, Measured service*), trei modele de servicii (*Software as a Service* (SaaS), *Platform as a Service* (PaaS), *Infrastructure as a Service* (IaaS)) și patru modele de implementare (*Private Cloud, Community Cloud, Public Cloud, Hybrid Cloud*).

Cloud Security Alliance (CSA) (Alianța de securitate Cloud) este o organizație non-profit cu misiunea de a promova utilizarea celor mai bune practici pentru asigurarea securității în Cloud Computing și de a oferi educație cu privire la utilizarea Cloud Computing-ului. Cloud Security Alliance este condusă de o coaliție largă de practicieni din industrie, corporații, asociații și alte părți interesate.

Information Systems Audit and Control Association (ISACA) este o asociație independentă non-profit, la nivel mondial angajată în dezvoltarea, adoptarea și utilizarea la nivel global a cunoștințelor și a practicilor pentru sistemele informatice de vârf. Misiunea ISACA este sprijinirea obiectivelor companiilor prin dezvoltarea, furnizarea și promovarea cercetării, a standardelor, competențelor și practicilor eficiente de guvernare, control și asigurare a sistemelor informaționale și tehnologiilor.

European Cloud Partnership (ECP) (Parteneriatul Cloud European) reunește industria și sectorul public pentru a stabili o piață unică digitală pentru Cloud Computing în Europa. Acesta a fost stabilit în cadrul Strategiei Europene de Cloud (European Cloud Strategy).

Capitolul 12. Rețele Vehiculare. Rețele Ad-Hoc

Introduction to Computer Networking pag 8
Communications and Networking An Introduction pag 197

Dicționar explicativ de termeni și abrevieri

AES	- Advanced Encryption Standard
AFH	- Adaptive Frequency-Hopping
AFHSS	- Adaptive Frequency Hopping Spread Spectrum
AP	- Access Point
ARP	- Address Resolution Protocol
AS	- Autonomous System
ASN-GW	- Access Service Network Gateway
BEC	- Backward Error Control
BGP	- Border Gateway Protocol
BNC	- Bayonet Neill–Concelman
BS	- Base Station
CD	- Compact Disk
CDMA	- Code Division Multiple Access
C-I-A	- Confidentiality - Integrity – Availability
CRC	- Cyclic Redundancy Check
CSMA/CD	- Carrier Sense Multiple Access with Collision Detection
CSMA/CA	- Carrier Sense Multiple Access with Collision Avoidance
DHCP	- Dynamic Host Configuration Protocol
DMZ	- De-Militarized Zone
DNS	- Domain Name System
DSSS	- Direct-Sequence Spread Spectrum
EDGE	- Enhanced Data rates for GSM Evolution
EDR	- Enhanced Data Rate
EPC	- Electronic Product Code
ETSI	- European Telecommunications Standards Institute
eSCO	- Extended Synchronous Connections
ETSI	- European Telecommunications Standards Institute
FDD	- Frequency Division Duplex
FEC	- Forward Error Correction/Control
FTP	- File Transfer Protocol
GPRS	- General Packet Radio Services
GSM	- Global System for Mobile Communications
HSDPA	- High-Speed Downlink Packet Access
HSPA+	- Evolved High Speed Packet Access
HSUPA	- High-Speed Uplink Packet Access
HTTP	- HyperText Transmission Protocol
ICANN	- Internet Corporation for Assigned Names and Numbers
ICMP	- Internet Control Message Protocol

IDS - Intrusion Detection System
IEEE - Institute of Electrical and Electronics Engineers
IETF - Internet Engineering Task Force
IMAP - Interactive Mail Access Protocol
IMAP - Internet Message Access Protocol
IoT - Internet of Things
IoV - Internet of Vehicles
IP - Internet Protocol
IPsec - Internet Protocol Security
IPTV - Internet Protocol Television
ISM - Industrial, Scientific, and Medical
ISO - International Organization for Standardization
ISP - Internet Service Provider
LAN - Local Area Network
LED - Light Emitting Diode
LLC - Logical Link Control
LoS - Line of Sight
LQI - Link Quality Indicator
LTE - Long Term Evolution
M2M - Machine to Machine
MAC - Media Access Control
MAN - Metropolitan Area Network
MANET - Mobile Ad-hoc Network
MIMO - Multiple-Input Multiple-Output
MMS - Multi-Media Service
MS - Mobile Station
MSC - Mobile Switching Centre
MTU - Maximum Transfer Unit
NAT - Network Address Translation
NSAP - Network Service Access Point
OFDM - Orthogonal Frequency-Division Multiplexing
OSI - Open Systems Interconnection
OSPF - Open Shortest Path First
PAN - Personal Area Network
PCM - Pulse-Code Modulation
PDU - Protocol Data Unit
PN - Pseudo Noise
PRR - Packet Reception Rate
Q0S - Quality of Service
RIP - Routing Information Protocol

RFID	- Radio-Frequency Identification
RSSI	- Received Signal Strength Indicator
SAP	- Service Access Point
SIG	- Bluetooth Special Interest Group
SMS	- Short Message Service
SMTP	- Simple Mail Transfer Protocol
SNMP	- Simple Network Management Protocol
SNR	- Signal to Noise Ratio
SS	- Stationary Station
SSP	- Secure Simple Pairing
TCP/IP	- Transmission Control Protocol/Internet Protocol
TDD	- Time Division Duplex
TLS	- Transport Layer Security
TSAP	- Transport Service Access Point
UDP	- User Datagram Protocol
UMTS	- Universal Mobile Telecommunications System
URL	- Uniform Resource Locators
V2I	- Vehicle to Infrastructure
V2V	- Vehicle to Vehicle
VANET	- Vehicular Ad-hoc Network
VPN	- Virtual Private Network
WAN	- Wide Area Network
WAVE	- Wireless Access for Vehicular Environments
WCDMA	- Wideband Code Division Multiple Access
WLAN	- Wireless Local Area Network
WiMAX	- Worldwide Interoperability for Microwave Access
WSN	- Wireless Sensor Networks

Bibliografie

[1] John Cowley, *Communications and Networking. An Introduction*, Ediția a 2-a, Editura Springer, 2012.

[2] James F. Kurose și Keith W. Ross, *Computer Networking. A top-down approach*, Ediția a 6-a, Editura Pearson, 2012.

[3] Peter Brown, *20 Billion Connected Internet of Things Devices in 2017, IHS Markit Says*, http://electronics360.globalspec.com, 25 ianuarie 2017

[4] Michael Yardney, *How many devices are connected to the Internet? [Infographic],* https://propertyupdate.com.au, 29 septembrie 2016.

[5] Tony Danova, *Morgan Stanley: 75 Billion Devices Will Be Connected To The Internet Of Things By 2020*, http://www.businessinsider.com, 2 octombrie 2013.

[6] Andrew S. Tanenbaum, David J. Wetherall, *Computer Networks*, Ediția a 5-a, Editura Prentice Hall, 2011.

[7] Narasimha Karumanchi, Damodaram A., Sreenivasa Rao M., *Elements of Computer Networking: An Integrated Approach - Concepts, Problems and Interview Questions*, Editura CareerMonk Publications, 2017.

[8] Tony Irujo, *OM4 - The Next Generation of Multimode Fiber*, Furukawa Electric North America, 2011.

[9] LanPro, *How to select the proper type of Optical Fiber*, White Paper, M7200010_TT_ENB01W.

[10] Mahmoud Shuker Mahmoud, Auday A. H. Mohamad, *A study of efficient power consumption wireless communication techniques/modules for Internet of Things (IoT) applications*, Advances in Internet of Things, Vol. 6, Nr.2, pag. 19-29, 2016.

[11] William Stallings, *Wireless communications and networks. Second edition*, Editura Pearson Prentice Hall, 2005.

[12] Andrew S. Tanenbaum, David J. Wetherall, *Computer Networks, 5th Edition*, Editura Pearson, 2011.

[13] Internet Engineering Task Force (IETF), RFC 768 – User Datagram Protocol, https://tools.ietf.org/html/rfc768, 28 August 1980.

[14] Internet Engineering Task Force (IETF), RFC 793 – Transmission Control Protocol, https://tools.ietf.org/html/rfc793, Septembrie 1981.

[15] https://dexonline.ro/definitie/securitate

[16] Sagar Ajay Rahalkar, *Certified Ethical Hacker (CEH) Foundation Guide*, Ed. Apress, 2016.

[17] Joseph Migga Kizza, *Guide to computer network security. Fourth Edition*, Ed. Springer, 2017.

[18] Anna Förster, *Introduction to Wireless Sensor Networks*, Ed. Wiley-IEEE Press, 2016.

[19] A. Woo, T. Tong, and D. Culler. *Taming the underlying challenges of reliable multihop routing in sensor networks*. Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys), pag. 14–27, Los Angeles, CA, USA, 2003.

[20] Doina Banciu, Neculai Andrei, Mihail Dumitrache, Ionuț Eugen Sandu. *Cloud Computing. Curs Partea 1.* Institutul Național de Cercetare-Dezvoltare în Informatică ICI București.