

1. ISO/OSI(Open Systems Interconnection) Reference Model- teoretic

Three layer model= Network Access + Transport + App

- Layererele de nivel inalt folosesc serviciile oferite de layererele de nivel jos
- Layererele de nivel mai jos ofera servicii urmatorului layer mai inalt
- Nivelul cel mai de sus e exportat catre user
- Intre layere avem interfete(SAPs= service access points), independente de implementare, putem schimba in interiorul unui layer fara a le modifica si pe restul
- Nivelele de jos trebuie sa fie simple, se poate complica nivelul de aplicatie
- **Avantaje layere**: - operatii mai curate, design mai usor de modificat, se pot identifica bug-urile mai rapid si mai usor, framework independent(layere cu functionalitati independente)

ISO/OSI= model de referinta pentru retele si protocoale, descrie cum ar trebui structurata o implementare, nu descrie felul in care trebuie implementat

Debugging de la nivel 1 spre 7.

Outgoing= incapsulare

Incoming= decapsulare

Un protocol opereaza intre acelasi nivel a 2 sisteme diferite.

De jos in sus(7 nivele):

1. **Physical layer**

- Tipuri de cabluri, conexiuni fizice, cum se transmit si primesc inf, sincronizare
- Wireless: RF, IR, microwave
- Wired: cupru: UTP,FTP,STP si fibra optica

2. **Data Link layer**- se lucreaza la nivel de adresa MAC(identificator unic pe 48b al unei interfete de retea: primii 24b producatorul, urmatoorii 24b id unic)

- Cum se face management-ul intre dispozitive din aceeasi retea, adresare(MAC), protocoale, detectie de erori, etc
- Wireless: Access Point(AP)
- Wired: Switch

3. **Network layer**- se lucreaza la nivel de adresa de IPv4/ IPv6

- Cum se fac legaturile intre 2 retele diferite: adresare globala, rutare, cum se trimit pachetele, congestie, flow control, billing and accounting(facturare si contabilitate)
- Wireless: router wireless
- Wired: router

4. **Transport layer**

- Se folosesc numere de porturi
- Legatura capat la capat de la un serviciu la alt serviciu: multiplexare, adresare, management conexiuni, segmentare, controlul erorii, flow control

5. **Session layer**

- Controlul dialogului
- Management-ul sesiunilor logice de comunicare: half/full duplex, grupare, checkpoint, etc

6. **Presentation layer**

- Cum e formatul datelor: formatarea datelor, compresie, criptare, probleme de securitate, conversie de protocol, etc

7. **Application layer**

- Cum sunt definite interfetele pentru aplicatii- generice/ specializare, functii de management a traficului de retea, ce protocoale se ruleaza la final
- Jitter= variatia delay-urilor
- Nu e nevoie de bandwidth foarte mare la toate aplicatiile(elastic= se adapteaza la diferite latimi de banda)

TCP/IP Reference Model- practic

Un standard “de facto”, adoptat si folosit, dar nu e un standard oficial.

Acesta e folosit in implementare= asa se numeste stiva de protocoale. Se reduce nr de nivele deoarece e mult mai usor de implementat. Exista comunicare intre nivele neadiacente.

De jos in sus(4 nivele):

1. Network Access(Subnet)= Physical layer + Data Link layer

- Tipuri de cabluri, conexiuni fizice, cum se transmit si primesc inf, sincronizare
- Wireless: RF, IR, microwave
- Wired: cupru: UTP,FTP,STP si fibra optica
- se lucreaza la nivel de adresa MAC(identificator unic pe 48b al unei interfete de retea: primii 24b producatorul, urmatoorii 24b id unic)
- Cum se face management-ul intre dispozitive din aceeasi retea, adresare(MAC), protocoale, detectie de erori, etc
- Wireless: Access Point(AP)
- Wired: Switch

2. Network layer- se lucreaza la nivel de adresa de IPv4/ IPv6

- Cum se fac legaturile intre 2 retele diferite: adresare globala, rutare, cum se trimit pachetele, congestie, flow control, billing and accounting(facturare si contabilitate)
- Wireless: router wireless
- Wired: router

3. Transport layer

- Se folosesc numere de porturi
- Mecanism de reliability(TCP)

4. Application layer= session+ presentation+ application

- Controlul dialogului: comunicare intre procese si aplicatii(FTP, SMTP, etc)

2. Network Topologies

= cum aranzam device-urile intr-o retea la nivel logic(cum se transmit datele la nivel logic, conceptual)/ fizic(cum se conecteaza prin cabluri)

- **Pentru avantaje si dezavantaje vezi curs(slide 310)**
- **Avantaj topologii logice peste fizice:** - ca sa nu aglomeram reseaua dpdv fizic, adaugam restrictii logice de comunicare

a. Topologii verticale:

- **HIERARCHICAL(TREE)**
 - o Un nod central(radacina) la care sunt conectate alte noduri intermediare ca si intr-un arbore
 - o Frunzele sunt workstations
 - o Datele circula up-down folosind nivelele intermediare
- **MESH**
 - o Cel putin 2 noduri cu 2 sau mai multe cai intre ele
 - o La full-mesh prea multe cabluri, prea scump

b. Topologii orizontale:

- **STEA/ STEA EXTINSA- snowflake(ex. Switched Ethernet LAN)**
 - o Un nod central in mijloc la care sunt conectate alte noduri periferice
 - o Nodul central face broadcast la ce a primit de la un nod periferic catre toate celelalte noduri periferice
 - o Cele mai folosite, se pot conecta multe device-uri prin switch-uri(smart), se interconecteaza stelele
 - o Costisitor la switch-uri si cabluri
 - o De tip broadcast, nu se poate limita acest broadcast
- **BUS(ex: Pure Ethernet LAN, Token Bus)**
 - o O linie centrala(cablu) la care se conecteaza toate device-urile
 - o Statiile sunt conectate intre ele prin AUI(attachment units/ transceivers)
 - o Ieftina, e nevoie de un terminator pe cablu, altfel nu se pot transmite date
 - o E lenta, are limita de device-uri conectate, risc mare(daca a cazut bus-ul cade toata reseaua), probleme de coliziuni(datele se trimit in frame-uri- blocuri mici, terminatorul absoarbe cadrele la final de cablu)
 - o E de tip full-duplex(transmisie si receptie in acelasi timp)
- **TREE**
 - o De tip arbore, un nod in varf si tot coboara cu alte noduri interconectate
- **RING(ex. Token Ring LAN)**
 - o Fiecare nod e conectat la 2 vecini, ca si intr-un inel
 - o Statiile sunt conectate intre ele prin repetoare(leg pct la pct)

- Dual ring pentru a transmite mai multe date in acelasi timp(pe fiecare cablu o anumita directie), un cablu pentru redundanta(in cazul in care celalalt cade)
- La scala mica e rapid, pe scala mare nu e rapid din cauza repetoarelor, se trece prin toate
- Datele se trimit in frame-uri, cand un frame e eliminat acesta se intoarce inapoi la sursa
- La TOKEN RING nu exista coliziuni, trimite doar statia care are token-ul
- **HYBRID**
 - **Star cu Bus**

3. HTTP(Hypertext Transfer Protocol) – nivel de aplicatie

- **WWW(World Wide Web)**= framework architectural pt accesul de documente linkate si distribuite pe internet, avem servicii date de browsere, retele sociale, etc; e bazat pe arhitectura client-server
- Hyperlink= prin click face legatura cu alt document
- Modul incognito sterge cache-urile dintr-o anumita sesiune
- Orice pagina e scrisa cu un mark up language, standardul e **HTML**(Hypertext Markup Language)
- O pagina se identifica printr-un **URL**(Uniform Resource Locator) dat de protocol(http, https, ftp, file, mailto), computer name(DNS name), portul TCP(optional), numele documentului(calea de pe computer catre pagina)
- Un identificator mai general e **URI**(Universal Resource Identifier), pot exista pagini duplicate
- **HTTP portul 80, HTTPS portul 443**
- **HTTP**= protocolul utilizat in cazul WWW
- **Nu e un protocol specific pentru transmitere de hypertext, se transmit si text plain, audio, video, imagini**
- Trimite informatie sub forma de client(Web browser) –Web server
- **Foloseste conexiuni TCP**
- E un protocol **fara stare**= fiecare tranzactie e tratata independent, nu depinde starea curenta de cea anterioara, o noua conexiune TCP pentru fiecare tranzitie, se incheie conexiunea cand tranzactia e completa
- Cu **cookies**(pastreaza inf relevante anterioare) facem ca protocolul sa fie **stateful**
- **Client**= browser care cere, primeste si afiseaza obiecte WWW
- **Server**= WWW server daemon care trimite obiecte ca si raspuns la cererile clientilor
- **Functionalitate:** clientul initiaza conexiunea TCP(socket catre server pe protul 80), server-ul accepta conexiunea de la client, se transmit mesaje HTTP intre browser si server si se incheie conexiunea
- **Caching**= se pastreaza anumite inf pentru utilizare ulterioara
- **Metode HTTP si status codes**

- **Un mesaj HTTP**=unitatea protocolului incapsulata: cereri/raspunsuri, anumita structura(vezi curs)
- **Sisteme intermediare HTTP:**
 - o **Proxy**= sistem intermediar catre client, actioneaza pt client facand forward la request-uri, **avantaj:** se pot folosi variante diferite de HTTP
 - o **Gateway**= sistem intermediar catre server, actioneaza pt server, dintr-un pct gateway-ul face filtrarea, **avantaj:** se poate folosi pt servere diferite de HTTP
 - o **Tunnel** = sistem intermediar catre ambele, niciunul nu vede ce se intampla, releu intre cele 2, **probleme:** nu e end-to-end service(legatura directa client-service)

4. Electronic mail- nivel de aplicatie

= cea mai folosita aplicatie din orice retea

- Procesare automata: sortare, reply
- RFC 822= request for comments, dupa ajunge la varianta finala, ca si standard
- Componente arhitecturale:
 - o **MUA(Message User Agent)**
 - Un client de mail care formateaza mesajul intr-un format care va putea fi accesat de MHS(ex: Thunderbird,Opera,Mail, etc)
 - Browser-ul ruleaza un MUA
 - o **MHS(Message Handling Service)**
 - **MSA(Mail Submission Agent)**= accepta mesaje de la MUA prin SMTP, verifica respectarea standardelor
 - **MTA(Message Transfer Agent)**= server intermediar prin care se tot transmite informatie
 - **MDA(Mail Delivery Agent)**= transfera mesajul de la MHS la MS
 - o **MS(Message Store)**
 - Inbox-ul user-ului
 - Localizat pe un server remote sau pe aceeaasi masina ca si MUA
 - Se foloseste protocolul POP(Post Office Protocol, nu e foarte reliable, mergea pe TCP, portul 110, **dezavantaj:**se sterge de pe server odata ce a fost preluat mesajul) sau IMAP(Internet Message Access Protocol, pastreaza mereu o copie a mesajului pe server, mai multe mailbox-uri)
 - o **Altele**
 - **Administrative management domain(ADMD)**= **releu**(antena care preia mesajul si il transmite mai departe)
 - **DNS**
- Mail- contine text(header RFC 822 si corpul mesajului) si o lista de destinatii
- CC(Carbon Copy)
- BCC(Blind Carbon Copy)= nu poti vedea restul destinatarilor

SMTP(Simple Mail Transfer Protocol) - nivel de aplicatie

- RFC 821
- Reliable, in general
- Foloseste portul TCP 25(potrivit pt ca intai se stabileste conexiunea, etc) pentru transmisie de mesaje intre client si server, mai poate fi folosit pt e-mail adr lookup sau verificarea de adr
- Comenzile sunt text ASCII si raspunsul e un status code si phrase
- 3 etape: handshaking, mail transfer, closure
- **Functionare sender:** se ia un mesaj din coada, se transmite
- Pot fi mai multi senders active
- Cand transmiterea e complete se sterge destinatia din lista, cand toate destinatiile sunt procesate se sterge mesajul=> **optimizare**
- Un mesaj pt mai multi useri cu acelasi host e trimis o singura data la host care il imparte intre useri
- Se utilizeaza o singura conexiune TCP, chiar daca sunt mai multe mesaje pregatite
- Nu garanteaza recuperarea mesajelor pierdute
- Nu informeaza originarul de trimitere
- Nu garanteaza detectarea unor erori
- ASCII ca si content type
- **Functionare receiver:** accepta mesajul primit in mailbox sau face forward

MIME(Multi-purpose Internet Mail Extension) - nivel de aplicatie

- Extensie pentru SMTP, foloseste RFC 2045,2056
- Prezinta linii aditionale in header pt a diferentia continutul mesajului:
 - o MIME version
 - o Content type
 - o Content transfer encoding(ex: base64)
 - o Content description
 - o Content id
- E extensibil
- E compatibil cu sistemele actuale de mail
- Totul e codificat ca si ASCII si au loc conversii
- Datele binare sunt incapsulate in ASCII mail

5. DNS(Domain Name System)- nivel de aplicatie

- E un protocol la nivel de aplicatie, cu o arhitectura de tip client-server(client= query pt a transla un nume **nslookup**(resolver) si server= daemons, raspunde la query-uri cu bind, named), aplicatia cheama resolver-ul
- Oferă translatarea între nume simbolice(ex: nume de site www.utcluj.ro) pentru IP hosts si adresele lor reale de IP
- Oferă o vizualizare ierarhica a internetului, DNS= baza de date globala implementata ca si o ierarhie de nume de servere
- Server Google: 8.8.8.8, putem face resolving folosind acest server, nu e neaparat sa folosim server-ul local
- Distribuit in toata lumea pentru a nu aparea latentă, pentru mentenanta, pentru spatiu de stocare
- **13 servere mari de DNS in lume(root name servers)** care fac legături între toate celelalte servere DNS(1376 instante duplicate in toata lumea)
- **Mod de functionare:** caută in DNS-ul local(poate găsi adrese accesate anterior si pastrate in cache), dacă nu il găsește, caută in server-ul root name corespunzător, care redirectionează cautarea către un authoritative name server care știe să trimită un răspuns
- **DNS names:**pana la 255 caractere, separate prin . , structura ierarhica in nume de la dreapta la stanga, primul nume e dat de root, ultimul e top-level domain(ex: .ro)
- **DNS domain** e un concept logic si nu are nevoie de un corespondent fizic al locatiei
- Abrevierile sunt rezolvate de resolver, server-ul DNS știe doar nume complete(full-qualified domain names)
- Interogarea server-ului poate fi recursiva(de fiecare dată se întreaba un nou server DNS care întreaba un alt server DNS la randul lui pana când se găsește răspunsul) sau iterativa(se întreaba un server DNS, nu știe si îți trimite inapoi o adresă de IP al unui server care ar putea ști, se întreaba noul server DNS si tot așa)
- In database se pot pune alias-uri pentru nume, de ex utcluj.com o să ducă tot la utcluj.ro
- Cu un DDNS se poate asigura o adresă de IP

6. TCP(Transmission Control Protocol) –nivel de transport

- Orientat pe conexiunea dintre client si server
- **Reliable over unreliable IP**= ca si un stream, flux continuu de date, multe caracteristici prin care incearca sa duca la capat o conexiune cu succes
- Clientul stabileste conexiunea catre server si o incheie cand e cazul
- RFC 793
- Fiecare capat e un **socket(IP adr+ port nr)**, pot fi active mai multe conexiuni pe un socket la un moment dat
- Transmisorul nu o sa trimita prea multe date, controlul congestiei= daca reseaua e aglomerata se asteapta
- Nu tine cont de constrangeri de timing si nu garanteaza o latime de banda minima
- Full duplex(in ambele sensuri simultan)
- Legaturi punct la punct
- Nu se poate face broadcast sau multicast
- Se folosesc buffere pt a adauga cat mai multe inf=> flag-uri
- Server-ul deschide un port selectat si asteapta mesajele pe el, clientul selecteaza portul local si trimite mesaje pe el
- Porturi well-known(asignate de IANA) sau asignate dinamic, porturile garanteaza corectitudinea datelor primite
- Un port local e folosit pt un singur serviciu la un moment dat
- Daca la nivel de server toate porturile sunt asignate deja nu le putem asigna conexiuni noi-> scalarea infrastructurii, o masina noua care preia din cea existenta
- Facilitate de labeling: data stream push/ urgent data signal
- Endpoints sunt identificate prin **porturi(numere pe 16b)**
- **TCP unit= segmente** cu un header de cel putin 20B, un segment trebuie sa fie mai mic de 64 KB(MTU= **Maximum Transmission Unit**), poate aparea fragmentare daca datele sunt mai mari decat MTU
- **Header segment TCP**
 - o Port sursa= 16b
 - o Port destinatie= 16b
 - o Nr de ordine al segmentului
 - o ACK number= semnalul care indica ca pachetul a ajuns= nr ordine seg+1=> de aceea e reliable!!
 - o Data offset= de unde incep datele
 - o Reserved= modificari
 - o Flags= urg, psh, rst, syn(=1 se incepe conexiunea), fin(=1 se incheie conexiunea)
 - o Window= cate seg se pot trimite deodata fara a astepta primirea unui ACK dupa fiecare

- Checksum= suma de contor pt a verifica integritatea pachetului, se verifica daca e acelasi cu cel din header
- Urgent pointer= se tine cont doar daca e invatata aplicatia sa faca asta
- Options+ padding pt a fi multiplu de 32b
- **3 way handshake(stabilirea conexiunii):** clientul trimite un pachet cu SYN=1, daca sender-ul vrea sa il accepte ii trimite inapoi un pachet cu SYN=1 si ACK=1, iar apoi clientul ii trimite iar un pachet cu ACK=1, lucru care inseamna ca a fost stabilita conexiunea
- la inchiderea conexiunii la fel, folosind FIN=1(graceful), sau de tip ABORT, se reseteaza conexiunea
- retransmisie: primul din coada/ batch(toate din coada, un timer pt toate)/ individual(un timer pt fiecare seg din coada, se confirma pe rand, varianta buna pt seg mari de date)
- **controlul congestiei**= ce se intampla cand se trimit prea multe pachete odata
- **modalitati:** slow-down/ speed-up- diverse metode de a determina un timp dupa care sa se reincearca transmitia(daca nu s-a primit ACK un timp indelungat clientul se poate gandi sa trimita datele mai incet, creste valoarea timer-ului)
- **window management**= pt a asigura ca nu se aglomereaza reseaua se porneste cu un slow start, se trimite un segment, se primeste ACK in timp util, se trimit 2 seg, si se tot creste nr de seg pana cand nu se mai primeste ACK=> **window size dinamica** care se modifica in functie nivelul congestiei
- **dezavantaj:** lent(compartiv cu UDP) din cauza complexitatii

UDP(User Datagram Protocol)- nivel de transport

- Nu e bazat pe o conexiune existenta, e de tip best effort
- **Nu e reliable**, se trimit toate datele odata
- Nu ofera controlul flow-ului, setup, controlul congestiei, latime de banda, timing
- Mesajul trebuie sa incapa intr-o singura datagrama UDP, se trimite cate un mesaj independent pe rand
- **Avantaj:** e foarte rapid, low overhead
- Nu se retransmit datele
- Nu e nevoie de semnale de ACK
- Folosit in aplicatii real-time si pentru functii non-critice
- **Dezavantaj:** nu se garanteaza controlul si lipsa duplicatelor
- **Header:**
 - Port sursa= 16b
 - Port destinatie= 16b
 - Lungime mesaj
 - Checksum optional

7. IPv4(Internet Protocol)- nivel de retea

IP privat= cunoscut in interiorul propriei retele, nu putem trimite date in afara retelei cu un IP privat

IP public= face conexiunea cu alte calculatoare din retea

- Serviciu de tip best effort, nu e reliable, se transmit datele dar nu se garanteaza ca ajung la destinatie
- Internet= o colectie globala de mii de retele individuale
- IP = cheia care conecteaza toate celelalte protocoale
- Conceptul de pachete de date
- Primitive(functii care pot avea loc): send, deliver
- Parametrii(transmitere de date si inf de control): adr sursa, adr destinatie, protocol, tipul serviciului
- La acest nivel de retea se trimit datele sub forma unui sir de bytes
- **Structura unui pachet IP(header):**
 - o Versiune: 4/6
 - o Internet header length(IHL)= 32b words
 - o Type of service
 - o Total length= al datelor, in octeti
 - o Id= pt a identifica pachetul
 - o Flags
 - o Fragment offset= de unde incep datele
 - o Time to Live(TTL)= prin cate hop-uri(routere) poate trece un pachet de date de la sursa la destinatie, iar cand se atinge acea limita, pachetul va fi eliminat daca nu a ajuns inca la destinatie, pentru a nu se plimba in retea intr-o bucla infinita; la fiecare hop prin care trece pachetul se decrementeaza TTL, daca ajunge la 0 si pachetul nu a ajuns la destinatie atunci acesta se elimina
 - o Protocol
 - o Header checksum
 - o Adr sursa=32b
 - o Adr dest=32b
 - o Options+ padding la 32b
 - o + Data field
- **IPv4(32b) = network ID + host ID**
- Clasele se identifica dupa primii biti
- Unicast= comunicare 1 la 1
- Broadcast= comunicare 1 la toti
- Multicast= comunicare 1 la cativa(N)
- Masca de retea= identificatorul de network
- **Subnetare**= impartirea unei retele in subretele, ajuta la management, debugging, se imprumuta din bitii de host

Internet Control Message Protocol(ICMP)

- Permite sa trimitem pachete in retea pentru a vedea ce s-a intamplat cu alte pachete, de exemplu facem ping
- RFC 792
- Transferul unor mesaje de control de la routere/ hosts la alte hosts, mod de testare/ controlare a retelei
- Mecanism de raportare a erorilor
- ICMP-ul e incapsulat in pachete IP, foarte importante deoarece anunta daca checksum-ul nu e corect, TTL a expirat, nu exista ruta, etc
- Nu e foarte reliable
- Mesajele ICMP sunt trimise ca si raspuns la niste datagrame cu problem
- Un mesaj ICMP nu e trimis pt un mesaj ICMP
- Tipul pachetului ICMP e dat de field-ul type

8. IPv6- nivel de retea

- Functioneaza in paralel cu IPv4
- A fost nevoie de o plaja mai mare de valori pentru adresele IP
- Adresele sunt de 128b= 16 nr in hexa(4 prefix, 4 interfcae ID), separate prin : => miliarde de adrese
- **Autoconfigurare** = adresare dinamica a adreselor
- **Anycast**= comunicare 1 la oricare, de obicei cea mai apropiata interfata
- **Multicast**(incepe cu FF)= comunicare 1 la N(toate interfetele identificate), de exemplu pt stream-uri de tip multimedia
- **True multicast**- se trimite un singur pachet catre router care verifica daca are un grup pt care e acel pachet, daca da, duplica pachetul si il trimite tuturor din grup, astfel doar la nivelul routerelor apar mai multe pachete care trebuie transmise => protocolul **IGMP(reducerea nr de pachete existente in retea)**
- **Nu mai avem broadcast**
- Permite etichetarea pachetelor
- Nu e compatibil cu IPv4(nu se poate da ping de la o adr ipv4 la una ipv6), dar e compatibil cu protocoale de nivel mai inalt
- Se inlocuieste field-ul type of service de la IPv4
- Header-ul mai scurt decat cel de la IPv4, apare un header de extindere care se adauga daca e necesar: hop by hop options, routing, fragment, authentication, encapsulating security payload, destination options
- **IPv6(128b) = prefix(001-global+global routing prefix+subnet) + interface ID**
- Prefixul indica in ce retea ne aflam
- /64 e lungimea comuna a prefixului

- Fragmentarea se poate face doar de la sursa, nu putem fragmenta pe parcurs pachetele (la IPv4 se poate), doar la final se reasambleaza datele
- Link-local FE80/64
- Global 2000/64
- Loopback ::1/128
- :: comprimare de 0
- Registry= ofera adresele publice, primii biti din prefix
- ISP= genereaza adr de IP
- Adrese link-local: Interface ID de obicei automat din adr MAC a dispozitivului
- Adrese globale: prefix de la DHCP sau de la un router
- Adrese site-local: in interiorul unei organizatii
- La rutare statica trebuie sa stim next hop, default e o adresa de ISP
- La IPv6 avem mai multe adrese de IP pe o interfata
- **Dual stack routers:** avem pt un device 2 adrese: IPv4 si IPv6 si o folosim pe cea potrivita
- **Tunnel:** trimitem adr IPv4, primul router il transforma in IPv6 si comunica cu urmatorul care il converteste iar la IPv4
- **NAT(Network Address Translation) Protocol Translation:** un IPV6 intra intr-un device cu NAT care va transla adresa in una de tip IPv4, celalalt trimite cu IPv4 si device-ul translateaza in IPV6

9. Flow Control Techniques(tehnici de limitare a cantitatii de date transmise)- la nivel 2(Data Link):

- Data unit la nivel 2= frame
- **Frame synchronization**= se detecteaza inceputul si finalul frame-ului, flag-uri pentru sincronizare, pot exista in ambele parti
- **Flow control**= trebuie sa nu suprasolicitam receptorul, deoarece el retine in buffere foarte multa informatie
- **Timp de transmisie**= timpul necesar pt o statie sa trimita toti bitii in mediu
- **Timp de propagare**= timpul pt un bit sa traverseze link-ul de la sursa la destinatie

Stop and Wait

- **Functionare:** sursa trimite la o destinatie un frame, destinatia primeste frame-ul, daca doreste sa se continue, raspunde cu un ACK, sursa asteapta ACK inainte de a trimite un nou pachet
- **Timeout** care creste dupa fiecare trasnmisie= $t_1 + \text{delta} + ..$
- **Inefficient** pentru frame-uri mici
- Functioneaza bine pentru cadre de dimensiuni foarte mari
- Cand timpul de propagare > timpul de transmisie linia nu e utilizata la capacitatea sa

Sliding Window(cel mai utilizat, dinamic)

- Permite existenta mai multor frame-uri in transit pe legatura
- Efficient pentru full duplex si transmiteri de viteze mari, frame-uri mici, multe
- **Functionare:** receiver-ul are un buffer W de lung, sender-ul poate trimite pana la W cadre fara a astepta dupa un ACK, acesta numereaza cadrele pt a se stii cate s-au trimis, receiver-ul trimite un frame ACK impreuna cu numarul urmatorului frame asteptat
- Cadrele sunt numerotate modulo 2^k
- Marimea ferestrei e de obicei mica
- Sender-ul tine o lista cu numerele cadrelor pe care le poate transmite, receiver-ul tine o lista cu numerele cadrelor pe care e pregatit sa le primeasca=> sliding window flow control
- Se primeste un ACK pt toate frame-urile transmise odata, nu dupa fiecare
- Se poate ca W sa fie variabil(pt TCP), acesta sa tot creasca daca receiver-ul e liber, sa trimita W+N

Error detection vs error correction

- Se detecteaza prin biti de paritate, se corecteaza prin retransmisii

10.HDLC(High-level Data Link Control) protocol and frame structure

- Ex: PPP derivat din HDLC(router wireless la ISB), cu disp intermediare e PPoE
- Era inainte SDLC(Synchronous), dezvoltat de IBM
- E folosit de router
- Tipuri de statii
 - o **Primara**= master, da comenzi
 - o **Secundara**= slave, ofera raspunsuri
 - o **Combinata** = peer 2 peer, fiecare joaca rol de client si server
- Se poate folosi in 2 moduri:
 - o **Balanced:** 2 statii combinate
 - o **Unbalanced:** o statie primara si una sau mai multe secundare
- **Normal Response Mode(NRM)**= configuratii unbalanced, polling= statia principala intreaba pe fiecare in parte daca are ceva de transmis
- **Asynchronous Balanced Mode(ABM)**= configuratii balanced, cel mai utilizat, fara polling
- **Asynchronous Response Mode(ARM)**= configuratii unbalanced, slave poate initia transmitia fara permisiune de la master
- **Frame structure:**
 - o Flag= pt sincronizare(nu poate fi 11111111), se introduce cate un 0 pt a echilibra linia, sa existe tranzitii

- Address= o singura adr pt ca avem legatura punct la punct
- Control= ce fel de date transmit
- Information= datele
- FCS= pt detectia erorilor
- Flag
- Header= flag+ adr + control
- Trailer= FCS+ flag
- Transmitem deodata si control si datele
- **Header compression**= un pachet cu header si trailer si la al doilea pachet zicem ca header-ul e la fel, lasam doar 1-2 biti

11.CSMA/CD (Carrier Sense Multiple Access/ Collision Detection)

- Ethernet-ul se bazeaza pe CSMA/CD, standard IEEE 802.3(min frame 64B)
- E o metoda de acces la mediu
- **folosit doar in retele half-duplex** = datele comunica in ambele directii, dar nu in acelasi timp
- o statie verifica daca mediul e idle, daca da, transmite
- daca mediul e ocupat, se asteapta sa fie idle si apoi transmite
- collision => 2 statii trimit in acelasi timp, se opresc amandoua, trimit un **jamming signal** care anunta toata retea ca a avut loc o coliziune, dupa care fiecare asteapta un timp random diferit inainte sa asculte din nou mediul si sa incerce iar o transmisie
- bazat pe competitie

FDDI(Fiber Distributed Data Interface)

- retea de tip TOKEN RING, cu fibra optica(pt viteze foarte mari)
- 2 inele cu fibra optica: unul primar si unul de back-up cu flow-uri opuse
- E o metoda de acces la mediu
- 3 tipuri de statii:
 - **DAS**(Dual Attachement Station)
 - **DAC**(Dual Attachement Concentrator) = cel care face management-ul fibrei optice si a comunicatiei
 - **SAS**(Single Attachment Station)= end device-uri
- Intre DAC si SAS se poate folosi si cablu UTP
- Se foloseste o codificare de tip 4B5B(4 biti sunt inlocuiti cu 5b) pt a introduce noi tranzitii, sa nu fie totul pe 0 cand vrem sa facem sincronizare

12. FastEthernet- protocol de nivel 2

- Viteza de 100 Mbps
- Cost redus
- Se ilustreaza standardul 100BaseX, unde X indica distanta pe care pot fi transmise datele
- Lungime minima cadru=512 biti
- Cadrul contine: preambul, (SOF- start of frame delimiter), adr destinatie, adr sursa, tipul, datele, FCS
- 100BaseTX- cablu de cupru UTP Cat 5
 - o **Codificari:** 4B/5B, MLT-3(multi-level transmit) Codificarea MLT-3 foloseste 3 nivele de tensiune si presupune prezenta sau absenta tranzitiei la mijlocul bitului indicand valoarea binara 1 respectiv 0.
 - o Atat half-duplex cat si full-duplex
 - o Topologie tip stea sau stea extinsa
- 100BaseFX- fibra optica
 - o **Codificari:** 4B/5B, NRZI. Codificarea NRZI presupune prezenta sau absenta tranzitiei la mijlocul bitului indicand valoarea binara 1 respectiv 0.
 - o full-duplex avand doua cai separate pentru transmisie si receptie
 - o cablarea verticala
- se pot folosi maxim 1 repetor de clasa I sau 2 de clasa II

GigabitEthernet(1000BaseX)

- Viteza de 1000Mbps
- O extensie cu lungimea min cadru= 4096 biti
- Frame bursting= se trimit multe cadre mici consecutive fara controlul CSMA/CD, evita overhead-ul
- 1000BaseSX, 1000BaseLX- fibra optica, 8B/10B si NRZ, full-duplex, stea/stea extinsa, 1 singur repetor intre 2 statii
- 1000BaseCX
- 1000BaseTX- cablu UTP Cat 5e, full si half duplex, codificare 4D-PAM5, stea sau stea extinsa
- Acelasi format de cadru ca si FastEthernet

10 GigabitEthernet(10GBaseX)

- Viteza de 10Gbps
- Range de la 300m la 40 km
- Interfete optice standard
- Poate fi folosit in MANs si WAN
- 10GBASE-S, 10GBASE-L, 10GBASE-E, 10GBASE-LX4

- Deoarece 10GbE folosește doar conexiuni full-duplex pe fibră optică CSMA/CD nu mai este necesar.

100 GigabitEthernet(100GBaseX)

- viteza de 100Gbps
- in data centers, enterprise LANs(data, voice, video) , etc
- medii industriale

13.Wireless LAN

- E mai accesibil dpdv al infrastructurii
- Standardul IEEE 802.11, versiuni: 802.11, 802.11b, 802.11e, 802.11f, 802.11i
- PAN, disp mobil cu smartwatch-uri, etc, pe Bluetooth
- E mai ieftin, nu mai e nevoie de cablaj
- Performante ridicate
- Conexiune celulara, pe bucati
- Avem AP(access points) cablate, de la AP se merge pe WIFI
- Diferentele pt wifi sunt la nivelele 1 si 2, nu mai avem cabluri si folosim radio frecventa
- AP sunt limitate in putere pt a nu interfera cu restul, avem o raza de acoperire care nu se suprapune foarte tare cu restul razelor de acoperire
- **Utilizare:**
 - o **Mod infrastructura:** Leg pct la pct intre cladiri, pe distante km, prin AP-uri, ne conectam cablat la un singur element central
 - o **Ad hoc:** nu mai avem AP central(Bluetooth), ex: masini autonome
- License free operation(pe ce frecventa se va transmite?)
- Router wireless= router+ AP, se extinde cu AP-uri, nu cu routere, duc un nr limitat de conexiuni
- AP-ul face buffering pe pachetele primite(pt streaming nu e prea ok)
- 802.11ac- canale mult mai late de 80/160 MHz, pt o sursa se poate permite utilizarea mai multor canale odata
- WIFI6(802.11ax) permite mult mai multe conexiuni paralele decat WIFI5(802.11ac)
- Cu 60GHz putem transmite intr-o singura camera

CSMA/CA access(Collision Avoidance)

- E o metoda de acces la mediu
- In retele Wireless
- Se face collision avoidance si nu detection deoarece pe wireless(mediu complet partajat si distribuit de toata lumea) nu putem detecta daca transmit mai multe statii

- odata deoarece apar interferente incontinuu de la diverse dispozitive, nu doar de la propriul dispozitiv
- Chiar daca a determinat ca nu se transmite nimic mai asteapta un timp scurt random inainte sa trimita datele, cel care a primit datele trimite un **acknowledgement signal**, daca nu il primeste, considera ca datele nu s-au trimis
 - Mecanisme RTC(request to send) si CTS(clear to send) -> un computer trimite un RTS la wireless access point, acesta opreste celelalte computere, ii trimite inapoi un CTS pt a-l anunta ca poate trimite datele
 - bazat pe competitie

14.Spanning-Tree- protocol de nivel 2

- Duce la eliminarea unei posibile bucle fizice intre switch-uri, se obtine o topologie de tip arbore
- La Ethernet nu mai avem campul TTL ca si la IP, de aceea e posibil ca un pachet sa se plimbe intr-o bucla infinita
- Operatiile bridge-urilor
 - o **Frame forwarding**= se invata adresele pe fiecare port, se cauta o adresa primita in tabela de switching, daca nu exista intrari in tabela, se face broadcast pe toate celelalte porturi, invata adresa MAC in urma raspunsului primit
 - o **Address learning**= se pot invata adresele in mod automat, se poate preincarca tabela, se pun niste timere pe intrari(arp –a pt a vedea corespondentele intre IP si MAC)
 - o **Loop resolution**= switch-urile trimit pachete(BPDU) odata la fiecare 2 secunde(anunta ca exista conexiune)
 - root-ul e switch-ul cu bridge ID cel mai mic, adica adresa MAC cea mai mica
 - fiecare switch alege o cale catre root, daca toate au acelasi tip de retea(ex: gigabit) se alege interfata cu un nr de port mai mica, daca costul nu e egal se alege linia cu cel mai mare cost
 - interfata aleasa va fi in mod forwarding, iar cele nealese vor fi in mod de blocare, dar vor asculta mereu pentru BPDU-uri, pt ca daca un port de tip forward va da fail atunci unul din modul de blocare trebuie sa il inlocuiasca
 - daca a cazut o legatura se recalculeaza, se reiau alegerile
- cu cat avem o viteza mai mare costul e mai mic, deoarece dorim sa fie alese legaturile mai rapide
- pentru un switch cu management se poate configura spanning tree-ul(prioritatile unui switch) pt a avea un arbore echilibrat
- reseaua e smart in mod automat, nu are inteligenta artificiala
- e mediu complet distribuit, colaborativ, nu avem arhitectura de tip client-server

- e mai scump sa punem procesari complexe la nivel 2 decat la cele mai inalte

15. Routing strategies

- Decizie de rutare= unde se trimite pachetul? Shortest path
- **Rutare fixa(statica)**
 - o Se calculeaza shortest path o singura data, rute statice configurate manual
 - o Apare atunci cand exista o unica legatura intre 2 dispozitive
 - o Ar trebui modificate costurile la fiecare modificare a topologiei
 - o Foarte rapida
- **Rutare prin inundare**
 - o Nu e nevoie sa stim inf despre retea
 - o Trimitem broadcast catre toti vecinii, inundare
 - o Se vor intoarce pachete dar vor ajunge si la destinatie
 - o Se viziteaza toate nodurile
 - o Circuite virtuale
- **Rutare aleatoare**
 - o Se selecteaza urmatorul nod random pe baza unei probabilitati sau round robin
 - o Mai buna decat flooding, nu e nevoie de inf despre retea
- **Rutare adaptiva-** cea mai folosita
 - o Se tine mereu cont de starea retelei: congestie, latime de banda, etc
 - o Inf e retinuta la nivelul fiecarui router
 - o Decizii complexe
 - o Shortest path pt toate combinatiile, se recalculeaza automat

16. Nyquist theorem and Shannon's theorem

- **channel speed** e limitat de capacitatea liniei vs actual speed
- teoreme folosite pt a obtine viteza maxima a canalului de comunicatie
- **Nyquist:** Pentru un canal ideal(fara pierderi sau zgomot) viteza maxima este:
 $V = 2 * H * \log_2 N$, unde H= banda de frecventa, N= nr de niveluri de codare(binary=2)
- **Shannon:** Pentru un canal real, afectat de zgomot viteza maxima este:
 $V = H * \log_2(1 + S/N)$, unde H= banda de frecventa, S= puterea semnalului transmis, N= puterea zgomotului
- S/N= signal per noise ratio($10 * \log_{10}(S/N)$ in dB, atenuare

TDM(Time Division Multiplexing)

- pt semnale **digitale**

- se utilizeaza toata gama de frecventa pentru o singura sursa, in acel semnal se pot codifica diferite elemente
- nu putem utiliza sursa la infinit, deci se imparte timpul, cateva secunde pe un anumit canal, impartim canalele in functie de timp
- se utilizeaza toata gama de frecventa, dar trebuie sa ne incadram in timp
- poate fi:
 - **sincron:** mai multe semnale merg pe aceeasi cale si sunt intercalate in timp la nivel de bloc sau de biti, time slot-urile sunt pre-asignate, fixe; daca slot-ul nu e alocat, se pierde acel slot, latime de banda scazuta; trebuie sa existe sincronizare si biti de control
 - **asincron**

FDM(Frequency Division Multiplexing)

- pt semnale **analogice**
- mai multe semnale analogice si gama de frecvente se imparte in mai multe canale, pe fiecare canal se poate pune alta frecventa(sursa)
- echivaleaza cu utilizarea in mod paralel a unei frecvente pentru mai multe surse
- daca bandwidth-ul total >> bandwidth un singur semnal, atunci se moduleaza fiecare semnal in anumite frecvente, fara overlapping=> se lasa niste frecvente de garda intre intervale(spatii) pt a nu se suprapune una peste alta
- **dezavantaj frecvente de garda:** raman frecvente nefolosite
- este un filtru combinatie de low-pass si high-pass

WDM(Wavelength Division Multiplexing)

- pt semnale **analogice**
- duce la o crestere foarte mare a vitezei pe fibra optica
- pe o singura fibra optica se moduleaza mai multe semnale cu diverse lungimi de unda
- exista niste prisme speciale(switch-uri optice) care combina fibrele optice de diverse lungimi de unda

Synchronous vs. Asynchronous transmission

- **asincrona:**
 - transmisie la nivel de caracter(intre 5-8 biti)
 - sincronizarea se face la nivel de caracter prin biti de sincronizare(start si stop)
 - biti de paritate pt detectia erorii
 - nu e nevoie de clk pt sincronizare
 - simplu, dar se transmit mai multi biti decat am dori(cei de sincronizare)
 - ieftina
- **sincrona:**
 - cea importanta la comunicatii

- se lucreaza la nivel de blocuri de date(caractere), apar niste flag-uri si biti de control
- complex, se transmit cam cati biti dorim
- scumpa

Switching techniques: packet switching

- switching= cum merg datele dintr-o parte in alta(comutare a pachetelor)
- **circuit switching**= cai fizice si noduri de retea, se faceau legaturi prin tablou; 3 etape(vezi curs); pot fi si circuite virtuale=> rutare rapida
- **message switching**= se trimite o bucata de date+ control+ adresare; tehnica de store and forward; nu se dau cai dedicate, e calculata in functie de cost, status
- **packet switching**= se preia ideea de la message switching, dar pachetul are o lungime limita mai mica, pana la 1000B(MTU); daca se pierde un pachet se va retrimite doar el, dureaza mai putin; 2 metode: folosind **datagram**(cu sau fara ACK, flexibilitate, unele pachete pe o cale, altele pe alte cale- avantaj major) sau **circuite virtuale**(3 etape: stabilirea conexiunii, transfer, deconectare), trebuie sa stim doar gateway-ul care va alege ruta cea mai scurta, daca e congestionata o schimba=> **dinamic**

17.Transmission Media: TP(twisted pair cables) and fiber optics

Exista 2 tipuri:

- Hard-wired= cablu de cupru:UTP,FTP,STP sau fibra optica
- Soft-wired= radio, satelit, IR, microunde, RF
- 0V= 0 logic, 5V= 1 logic
- **Fibra optica**= Tx si Rx(2 fire) pt a avea full duplex
 - o Single-mode= pe distante mult mai mari
 - o Multi-mode= pe distante mai scurte
 - o Informatia e transmisa sub forma de lumina
 - o **Indice de refractie**= viteza lumina in vid/ viteza lumina in mediu= viteza cu care trece lumina printr-un anumit mediu
 - o **Refractie**= bending of light-> atunci cand lumina trece dintr-un mediu cu un indice de refractie in alt mediu cu alt indice de refractie => **n_{Air} < n_{Glass}**
 - o Cu cat **creste indicele de refractie** cu atat lumina se “indoaie” mai tare si se intoarce in mediul initial la **critical angle**=> **reflexie interna totala la fibra optica**
 - o Se adauga un **layer de protectie numit cladding** => **n_{Cladding} < n_{Core}**
 - o Transporta date pe **100 km**
 - o **Optical fiber**-> **copper shell**-> **insulator**
 - o Latime de banda mai mare
 - o Viteza mai mare, Securitate buna
- **Cablu de cupru**(mai multe perechi de transmisie-receptie)

- **UTP (Unshielded)**
- **STP (Shielded) – industrial**
- **Straight** - ambele capete ale cablului folosesc **acelasi standard** => LAN
-conecteaza device-uri de **nivele diferite**
- **Crossover** – capetele folosesc **standarde diferite**
-conecteaza device-uri de **acelasi fel(nivel)**
- Standarde 568A, 568B
- Categori:3,5,5e,6,6a,7,8-> cel putin 5e pt internet

Media for Wireless transmissions; spread spectrum

- Spectru= gama de frecvente disponibile
- Spread spectru= ma intind pe toata gama de frecventa cu transmisia de date, cu cat gama de frecv e mai mare cu atat viteza e mai mare, se pot transmite mai multe informatii
- La RF putem modifica amplitudinea, faza pt a obtine un 0/1 logic, lucreaza la frecvente 2,4GHz sau 5 GHz(cu o anumita marja de eroare, un range de frecvente)
- Canalele de frecevente folosite trebuie sa nu fie overlapped, la 5 GHz sunt mult mai multe canale non overlapping
- La IR nu pot trece datele daca avem obstacole, la RF putem avea o difuzie a semnalului si ocoleste obstacolele
- Consum redus de energie(putere mica)
- Semnalul e intins pe o banda mai larga, prin 2 tehnici
 - **FHSS(Frequency hopping):** se fragmenteaza datele si fiecare bucata se transmite pe o frecventa anume si are loc cate un salt intre fiecare transmisie, exista si pause, astfel nu se tine o frecventa ocupata incontinuu; receiver-ul va face aceleasi salturi pentru a receptiona datele primite de pe frecventele bune
 - **DSSS(Direct sequence):** se incearca umplerea intregii game de frecventa cu semnalul respectiv, daca semnalul e lent el va trebui sa se schimbe foarte repede si se vor genera foarte multi biti, se ia un sir de date random si se face XOR cu input-ul(codificare), de unde rezulta sirul care va fi transmis fiind extins pe intreaga banda de frecventa

18.ADSL(Asymmetric Digital Subscriber Line)

- Potrivit pentru internet de viteza mare, video-on-demand
- Asimetric deoarece downstream-ul are valoare mare si upstream-ul are valoare mica
- Se folosesc frecventele pt voce si pt restul se face FDM intre downstream si upstream
- Nu se lasa frecventa de garda intre upstream si downstream pt a nu se influenta, se trimit pe aceeasi linie si ce se primeste se scade din ce s-a trimis(echo cancellation)
- Se imparte bandwidth ul intre mai multe surse
- ADSL in cazul internetului pe linii telefonice

Terms and concepts (short definitions):

1. **Claude Shannon's model of communication** = (telecomunicatie= comunicatie la distanta) se bazeaza pe existenta unor actori:

- *Sursa informatiei*= genereaza un mesaj care trebuie transmis
- *Transmitatorul*= converteste informatiile in date transmisibile(semnale)
- *Canalul*= transmite datele primite folosind semnale, poate fi afectat de zgomote din partea mediului(o singura linie de transmisie/ retea complexa)
- *Receiver*= primeste semnalele de pe canal si le converteste in date
- *Destinatie*= primeste mesajul transmis de sursa

Probleme:

- One way
- No feedback
- Nu e potrivit pt group communications
- Nu exista o explicatie pentru procesul de trimitere/ primire a datelor

2. **Comunicare**

- **Directa**: legaturi punct-la-punct sau punct la multi-punct
- **Indirecta**: mai multe retele comutate si disp care le conecteaza intre ele, nu putem face management-ul

Retea

- **Monolith**: retea dintr-o singura bucata
- **Structured**: retea divizata pe layere

Comunicare

- **Simetrica**: comunicatie de tip peer-to-peer, fiecare element poate fi provider sau client(ambele functionalitati)
- **Asimetrica**: comunicatie de tip client-server(master-slave), nu se pot inversa rolurile, clientul e cel care solicita, server-ul e provider-ul

Comunicare

- **Standard**: general valabila
- **Nonstandard**: pentru anumite computere/ task-uri

Comunicare

- **Fizica**: traseu exact fizic, de la un nod(router) la altul, pct de IP
- **Logica**: traseu de la un client la un server, pe nivel de aplicatie(HTTP, etc)

3. **Data Encapsulation**= adaugarea unor inf de control pentru date intr-un header:

- inf legate de adresa:ID,IP(sursa), destinatia
- Biti de paritate/checksum(pentru detectia erorilor)
- controlul protocolului

Data Segmentation (fragmentation la TCP/IP)= impartirea datelor in blocuri mai mici care se transmit rapid=>blocuri ATM(53 octeti), blocuri Ethernet(pana la 1526 octeti)

Avantaj: - putem utiliza mult mai usor mai multe resurse din retea, transmitand date in locuri diferite simultan

-daca se pierde un singur bloc, doar acela trebuie retransmis, nu intreg cadrul

Dezavantaj: - overhead => trebuie sa punem intr-un buffer toate bucatile, sa verificam daca s-au transmis-> overhead de calcul, spatiu, memorie

Data Reassembly= trebuie sa punem cadrele inapoi in ordinea corecta intr-un singur cadru, sa ne asiguram ca toate sunt corecte

4. **PAN(Personal Area Networks)**

- Distanțe mici(Bluetooth, IR, NFC)
- Conectăm telefoane, dispozitive dintr-o casa, pc-uri, laptop-uri

LAN(Local Area Networks)

- Acces într-o zonă geografică restransă: home/office, enterprise
- În general cineva deține rețeaua(department IT)
- De obicei sunt de tip broadcast(se transmite la toți)
- Ethernet/ Token Ring

MAN(Metropolitan Area Networks)

- LAN<MAN<WAN
- Interconectare de LAN-uri
- Operate de o singură entitate(organizație)
- Propria infrastructură

WAN(Wide Area Networks)

- Internet= un WAN
- Poate fi oricât de mare(global)
- Interconectare de hosts, LAN-uri, MAN-uri
- Operat de un provider de servicii de telecomunicații
- Fixed, satelit, mobil latime de bandă îngustă/ lăta
- Infrastructură fizică mare

5. **Analog data**= semnale continue cu valori într-un anumit interval(sunet, video)

Digital data= valori discrete(text, întregi) ,cost redus, atenuare redusă, puțin zgomot, securitate ridicată

6. **Role and examples of Data encoding**

= cu scopul păstrării datelor digitale sub aceste forme mai greu de descifrat, cât mai puțin susceptibile la zgomot

- **NRZ** = dacă se schimbă fața de perioadă anterioară avem 1 altfel 0
- **NRZI(NRZ-M)** =
- **Manchester**= se introduce o tranziție la mijloc de interval, detectezi erori mai ușor
- **Manchester Differential**
- **MLT-3**

= codificări din digital în date analogice:

- **Amplitude Shift Keying**
- **Frequency Shift Keying**
- **Phase Shift Keying**

7. **Repetor= nivel 1**, amplifica date si le transmite

Hub= nivel 1, repetor cu mai multe port-uri, transmite tuturor device-urilor conectate la el, nu stie diferentia intre adrese

Bridge= nivel 2(adrese MAC, logice), conecteaza 2 parti ale unei retele intre ele(switch cu 2 porturi), poate face legaturi intre retele cu protocoale diferite, transmite unui dispozitiv pe baza de adresa MAC

Switch= nivel 2, bridge-uri cu multe porturi, conectare dispozitive in interiorul aceleiasi retele

Router= nivel 3, conectare intre retele(LAN-uri)

O retea se extinde cu AP sau switch-uri, e nevoie de un singur ruter wireless.

Router wireless(unul singur in retea)= AP(wireless)+ router(+ switch+ servere(DHCP, media))

8. **Structured Cabling** – necessity and elements

= un set de standarde care ajuta la integrarea tuturor serviciilor disponibile(voce, date, video) in cadrul unei cladiri

- Trebuie actualizat din cand in cand
- Servicii integrate
- Standardizat de EIA/TIA
- Cablare verticala= etajele intre ele, fibra optica single-mode(backbone)
- Cablare orizontala= la nivel de un etaj, fibra optica multimode
- Standardele depend si de tipul cladirii(casa, birou, etc)
- Topologia de cablare: tip stea extinsa
- Tipurile de cabluri utilizate: FO sau TP
- Documentatia proiectului
- Nr angajati=> dimensiunea retelei
- Ce aplicatii se ruleaza=> latime de banda necesara
- MC(main cross connect)= centrul stelei, pt toata zona cablata
- IC(intermediate cross connect)= al 2-lea nivel, pt o cladire
- TC(telecommunication closet)= al 3-lea nivel, pt un grup de camere

9. **Bit stuffing role and necessity (example on HDLC protocol)**

= folosita pentru a asigura transparenta datelor si a evita confuziile care pot aparea in cazul unor tranzitii

Regula: se insereaza un 0 dupa fiecare secventa de 5 de 1 in partea sender-ului pentru a ne asigura ca exista o tranzitie

- Daca receiver-ul primeste 5 de 1 verifica urmatorul bit:
 - o Daca e 0 il sterge
 - o Daca e 1 si urmatorul e 0 il accepta ca si flag
 - o Daca urmatorii 2 biti sunt 1, abort

10. **Transmissions of data and control information**

piggybacking method= procesul de intarziere a trimiterii semnalelor de control(ACK) pentru a fi capturate impreuna cu urmatorul frame, sunt incluse in cadrul frame-ului