**Apunts TXC**

***15/02/2019***

**T1-Introducció**

**Model de Shannon**: organitzat en blocs que acumulen funcions, i entre blocs tenim interfícies.

En xarxes, els bits sempre s’envien en sèrie.

· **Source**: envia bits que Destination ha de rebre.

· **Transmitter**: adapta els 0 i 1 que surten de la font per enviar-los per la línia.

· **Transmission System**: és el que haurà d’enviar els 0s i 1s.

· **Receiver**: encarregat de “desadaptar/reconvertirlo” les dades rebudes per la linia.

· **Destination**: rep els bits enviats pel Source.

\* Tots els sistemes de comunicacions són simètrics.

**TCP/UDP:** té control d’errors i control de flux.

Internet: els routers és un commutador IP.

Xarxa ATM: els router són commutadors ATMs (assincronous transfer mode). Actualment està obsoleta ja que ha estat superada per l’ethernet.

Lo únic comú en totes les xarxes són els paquets ip. La resta s’ha de distingir entre TV, telefon i dades ja que els temps de transmissió han de ser diferents (no és el mateix veure un streaming que carregar una pàgina web).

Inconvent: Telefònica, és el que té el monopoly.

Dominador: Vodafone i Telefònica.

Altres: ono, pepephone...

· Reply ability: Si un inconvent vol fer algun canvi tecnològic que no pot fer el dominador, l’estat no autoritza el canvi.

· Positive discrimination: Tot allò que sigui dolent per l’inconvent, és bo per la competència.

· Own network: tots els operadors tinguin la seva pròpia xarxa. Però es van oblidar de posar terminis. Per això, mentre un nou operador està fent la xarxa, l’inconvent està obligat a llogar la seva xarxa. El que ha passat és que els nous operadors han deixat de muntar la seva pròpia xarxa i utilitzen la de l’inconvent.

El problema pervers és que si els altres no tenen xarxes i utilitzen la de l’inconvent, aquest últim no vol ampliar la xarxa ja que no només l’utilitzarà ell sinó també totes les “rates” i els altres no tenen cap interès en crear-la. Per això, la xarxa s’ha aturat, no hi ha creixement.

La nova alternativa és tenir una xarxa única.

**T2-Model arquitectònic d’Internet:**

Model OSI, basat en nivells i definit per ISO.

**Abstracció de nivells:**

Definim el model en nivells que executa funcions i proporciona serveis al nivell superior.

· Per definir el nombre de nivells es va elaborar una llista de funcions.

· Després es separen les funcions i s’agrupen per afinitat (cada agrupació de les funcions seria el nivell).

· Es busca el menor flux entre els nivells, per això es va determinar que **eren 7 nivells**.

**Model TCP/IP**

- Encapçalament permet executar funcions.

- Cada nivell executa funcions i proporciona serveis al nivell superior.

S’estableixen dos tipus de comunicació:

· **Verticals o físiques:** s’anomenen “interface”

· **Horitzontals o virtuals:** s’anomenen “protocol”. Aquest protocol s’estableix a través de les capçaleres del nivell. El que es fa és posar informació a la capçalera, de tal manera, quan arriba a l’altra banda, l’únic que podrà llegir això serà la del mateix nivell.

Per poder executar aquestes funcions s’utilitza la tècnica de l’encapsulament que amb els paràmetres que dona permetran executar la funció a nivells superiors.

Si en un protocol s’ha de canviar alguna cosa, el que es fa és afegir un subnivell i així no cal modificar ni l’ethernet, ni l’ip...

· Funció principal del **nivell 4 Transport** (TCP): control d’errors i control de flux a nivell global a nivell end-to-end. Unitat de dades es segments.

· Funció principal del **nivell 3 Xarxa** (IP): enroutar, unitat de dades (paquets ip), no te sincronisme!

· Funció principal del **nivell 2 Enllaç** (Network Acces): control d’errors i control de flux a nivell local a nivell de trama. Té sicronisme!

Les adreces de nivell 2 poden estar repetides fora de la xarxa local.

·Funcio principal de **nivell 1 Fisic** (transmissió, codificació, multiplexació, sincronisme a nivell de bit!

\* Ethernet: LLC – MAC 802.3 – Physical Ethernet (mirar-ho com si fos una pila).

*Exemple Taller0 – Q6 pila dels nivells d’una xarxa*

*\*\*per comprovar si funcionaria correctament, es mira si a nivell horitzontal els protocols de la mateixa alçada han de ser el mateix, sino no funcionara correctament.*

*----------------------------------------------------------------------*

**Protocols Control Enllaç**

(Les solucions que es plantegen estan plantejades a nvivell, la gran majoria).

La trama és la velocitat de dades de nivell 2.

Sempre que s’ha de prendre una decisió es pren en el punt mig (es segueix el model gaussià).

Si un buffer s’omple es perd el paquet. El control de flux intenta evitar que hi hagi overflow als buffers controlant el que s’envia (no hi ha grantia).

***21/02/2019***

**Fourier**

Diferència entre temps de transmissió i propagació: la unitat de mesura Vt = bits/seg, i la de Vp = m/seg. La velocitats de transmissió depèn del sistema mentre que la velocitat de propagació depèn del mitjà de transmissió (medi), com podria ser el material.

**Longitud d’un bit = Vp/Vt (m/bit).**

**Temps transmissió d’un bit = Vt-1 (seg/bit)**

**Temps de propagació d’un bit = L/Vp (seg) on L és la longitud/distància.**

**Temps de transmissió: = #bits/Vt**

**Velocitat de transmissió = 2\*A(ample de banda)**

**Temps de propagació = L/Vp**

\* Si el temps de propagació és molt petit pot ser negligible.

\* Si la velocitat de transmissió és molt gran, el temps de transmissió es pot considerar negligible.

\* Com més gran sigui l’ampla de banda , més bits per segon es poden enviar.

***Exercici Taller0***

VT=2000 bps

Tbit=1/2000 seg/bit

Tbit=1/4000 seg/bit

T=8tbits=8/2000

T=8/4000

fo=1/T=2000/8=250Hz

fo=4000/8=500Hz

4fo=4\*250=1000Hz

4fo=4\*500=2000Hz

Bw=750Hz

Bw=1500Hz

**Exercici paper-TXC01**

***22/02/2019***

**Design Factors Determining Data Rate and Distance:**

Ampla de banda: són les freqüències que deixen passar (quines freqüències té).

L’ampla de banda del senyal ha de quadrar amb l‘ampla de banda del medi.

“+ freqüència -> + velocitat de transmissió.”

+ ample de banda -> + velocitat de transmissió.

Atenuació: pèrdua de potència, vé determinada per la distància.

+ dispositius -> + atenuació

Atenuació es molt important; les senyals de radio es nota molt el contrast entre el dia i la nit, ja que el sol genera bastantes interferències que fan disminuir la atenuació.

Unitat per mesurar la *relació senyal-soroll: decibels (dB) = 10log10(Pa/Pb)*

*Pa/Pb = 10³*

*XdB = a\*logb(Pa/Pb) Pa/Pb = b^X*

**Twisted Pair:**

Tipus de cablejat que s’usa per crear xarxes.

UTP(Unshielded twisted pair)

STP(shielded twisted pair): apantallat, calbe més car peró més segurs.

| Industry abbreviations | [ISO/IEC 11801](https://en.wikipedia.org/wiki/ISO/IEC_11801) designation | Cable shielding | Pair shielding |
| --- | --- | --- | --- |
| UTP, TP | U/UTP | None | None |
| STP, ScTP, PiMF | U/FTP | None | Foil |
| FTP, STP, ScTP | F/UTP | Foil | None |
| STP, ScTP | S/UTP | Braiding | None |
| SFTP, S-FTP, STP | SF/UTP | Braiding and Foil | None |
| FFTP, STP | F/FTP | Foil | Foil |
| SSTP, SFTP, STP, STP PiMF | S/FTP | Braiding | Foil |
| SSTP, SFTP, STP | SF/FTP | Braiding and Foil | Foil |

The code before the slash designates the shielding for the cable itself, while the code after the slash determines the shielding for the individual pairs:

U = unshielded

F = foil shielding

S = braided shielding (outer layer only)

TP = twisted pair

Parell de fils elèctrics de coure que permeten transmetre corrent. Estan cargolats i això provoca un Efecte Faraday que provoca que l’ampla de banda sigui més alta -> més velocitat de transmissió.

Normalment s’usen dins dels edificis per connectar terminals.

· UTP: més econòmic

· STP: és igual però sobre el cable cargolat li han posat una malla metàl·lica que forma una pantalla. Això fa que l’efecte Faraday sigui més intens.

Les freqüències de treball són de màxim 600 MHz.

**Coaxial Cable:**

Ha perdut importància perquè la fibra òptica el supera.

És un conductor asimètric: hi ha un conductor interior i un exterior, separats per un aïllant. Això fa que es creï una gàbia de Faraday.

Les freqüències de treball són de més de 2’2 GHz.

**Optical Fiber:**

És un conductor asimètric: té un conductor interior i un interior, però no necessiten aïllant. Al ser una fibra de vidre, ambdós conductors tenen diferents índex de refracció que fa que la llum es pugui propagar.

Si la atenuació és constant no es pot treballar perquè apareix una distorsió.

El problema es que varia la Vpropagació.

**Satellite Point-to-Pint Link:**

Els satèl·lits són repetidors que si col·loques a 36000 km de la terra (òrbita geostacionària), es manté fixe a l’horitzó. Qualsevol objecte que es col·loqui a aquesta distància no és mou (ni s’envà ni cau ja que la força centrípeta i centrífuga s’igualen). D’aquesta manera, les antenes no s’han d’anar movent.

L’inconvenient que hi ha és que la senyal tarda molt en arribar (125ms per viatge al satèl·lit).

Sincronisme: principi i final de dades.

**Sorroll:**

Hi ha dos tipus de soroll:

**Crosstalk:** quan sona el telefon, entre els dos extrem es produeix soroll.

**Impulse noise:** quan enxegas la llum, es produeix una descarga electrica de molta intensitat i poca potencia(explosió).

El material intrisicament te un soroll, mai es pot anul·lar per complet, només es pot reduir. Per tant mai hi haura 0 soroll

Com més alta es la relació senyal-soroll millor per la qualitat de la línia.

**Mecanisme Stop & Wait:**

S’envia una unitat de dades i no es pot enviar l’altre fins que no es rep la confirmació de que el paquet ha arribat.

Link Utilitzation:

a = B (#bits que hi ha en el link) / L (#fibts de la trama).

En el primer cas la linia està molt més ocupada que el segon cas ja que B < L. Seria més eficaç utilitzar l’Stop & Wait en el primer cas ja que estaria més temps sense fer res.

LSMA/CD: abans de retransmetre ha d’escoltar si el terminal vol que es retransmetria (mirar si hi ha variació elèctrica). Només transmet si no hi ha ningú transmetent. Però el que pot passar és que es pensi que no hi ha ningú transmetent perquè els paquets encara no han arribat i no han sigut detectats. Aquí es produiria una transmissió. Si això passa, abandona i s’espera un temps (back off) abans de tornar-ho a intentar.

Ethernet té una avantatge que és que no necessita ack. Per assegurar-se que quan envia una cosa no es produirà col·lisió abans de retransmetre escolta. Suposant que t és el temps que tarda un paquet d’anar de punta a punta de la xarxa, si en 2t no s’ha produït una col·lisió implicarà que no es produirà mai. Com a inconvenient, durant 2t la trama no ha d’abandonar el terminal per si s’ha produït la col·lisió.

Si la durada de la trama és més petita que 2t ethernet no funciona. És per això que les xarxes ethernet tenen unes longituds màximes determinades.

Sliding Windows Flow Control (nivell 2):

La finestra serveix per controlar el flux que pugui arribar sobre els buffers, de manera que no es pugui enviar més del que els buffers puguin emmagatzemar.

Perquè fos més rendible que el mecanisme Stop & Wait es permet seguir transmetent sense haver d’esperar a rebre l’ack. Es van enviant paquets deixant-los en mode “espera” fins que es confirmi.

Les trames hauran de ser numerades ja que n’hi ha varies pendents de confirmar. Aquesta numeració s’anomena mòdul: nosaltres usarem mòdul 8 (2^3) o 128 (2^7). La garantia que ha de tenir és que el temps que passa entre que he enviat la trama i he rebut la confirmació és el temps mínim. Si es vol que no s’aturi mai, la finestra ha de ser de mida T/Tt on té és el temps mínim. Està calculat que amb una numeració de 8 és suficient si volem que passi això en qualsevol punt de la terra.

La finestra no pot ser més gran que la numeració, tot i així, si les trames són de 2^k, la finestra ha de ser com a màxim 2^k -1. Això és degut perquè si la finestra té 8 espais (0-7), si es rep l’ack0 implica que tots els anteriors estan bé i que espera el paquet 0. Si es perd un ack hi ha el mecanisme de time out i s’hauria de tornar a enviar el primer paquet, en aquest cas el 0. El receptor es pensaria que el 0 és el “nou 0” i no el que s’havia perdut. És per això que no es deixa fins a k.

La numeració de 7 bits es podria aplicar per enviar en una distància molt gran, on el temps de propagació és molt gran i per tant encara no hauria donat temps d’esperar la confirmació de la primera i per tant, tindríem el mateix problema que el stop & wait. Això s’aplica en satèl·lits.

Si la finestra de recepció és 1, la finestra només apunta a un paquet i només pot arribar aquest. Exigeix un ordre que permet detectar immediatament si un paquet es perd (han d’arribar d’un en un).

Mida de la finestra òptima:

Time out mínim = (bits trama/Vc) + tp1 + (#bits ack/Vt) + tp2

Mida òptima finestra = time out/temps trama

***28/02/2019***

**2.2 Codificació**

HDLC

· Primary: són les estacions que manen ja que és un protocol master-slave. Envien ordres i reben respostes (només pot haver una primary).

· Secondary: són estacions que reben ordres i envien respostes.

· Combined: és convinada, tant rep ordres com envia respostes i viceversa.

Amb aquest sistema anirà més lent ja que només hi ha un “jefe” però pot haver una carrega de treball del 100% mentre que per ethernet pot anar molt ràpid però si la càrrega de treball és del 20% ja rebenta.

* Les estacions secundaries no es poden connectar entre si. Només pot haver flux entre la primària i la secundaria i veceversa.

Quan s’envien coses entre ells, només cal posar un sol camp d’adreça ja que el primari sap totes les dels secundaris:

· primari  secundari: es posa l’adreça del destí.

· secundari  primari: es posa l’adreça de l’origen.

HDLC

El mecanisme que té aquest protocol per identificar la trama (és poc eficient però molt utilitzat) és posar un flag al començament i al final que ha de ser irrepetible i que no es pugui donar mai en una posició que no sigui la del flag.

Per fer-ho usem la tècnica “bit Stuffing” que consisteix en posar un 0 després d’una posició determinada. El transmisor, fora del flag, si detecta 5 1s seguits, inserta un 0. El receptor elimina els 0 després de 5 1s seguits. L’únic que portarà 5 1s seguits sense cap 0 a darrere seran els flags, que se’ls hi posarà un 1 en comtes d’un 0.

Piggybacking: superposició

NRM: P-> Poll

Per acabar el procés envia RR,F i la resposta és I,F

ABM: enviar P vol dir actuació immediata (confirmar), i la resposta porta el bit F activat.

Si el camp d’informació és molt petit hi ha un excessiu overhead, però si les trames són molt grans, si hi ha errors quan l’hagis de tornar a enviar s’haurà d’enviar molts més octets.

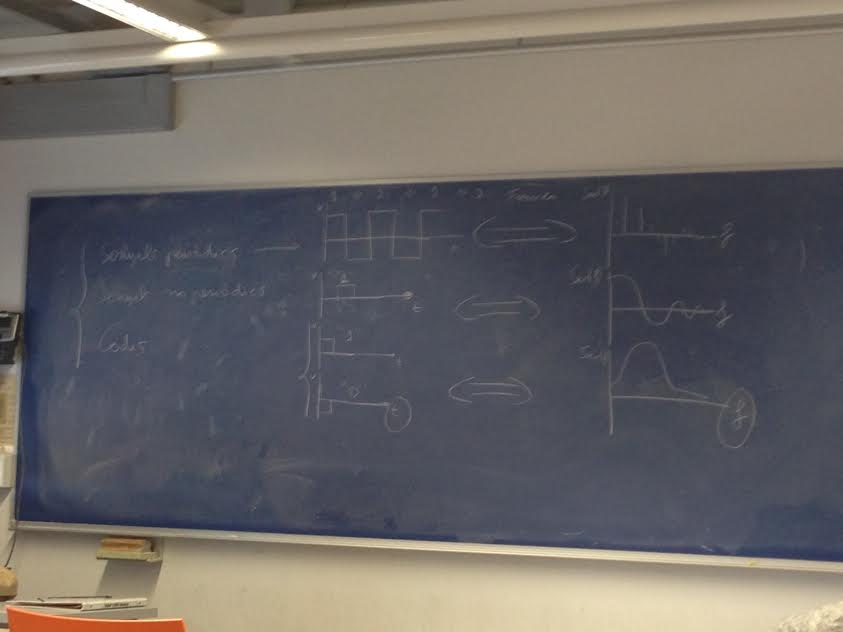
· Finestra òptima: compromís entre controlar el flux i poder transmetre contínuament.

HDLC Operation

· Inicialització: es posa tot a zero.

· Data transfer: comença a transmetre dades.

· Disconnect: indica que ha acabat.



**Distorció d’atenuació**: si l’atenuació varia en funció de la freqüencia, es produeix una distorsió ja que la suma de les diferents freqüencies ja no serà la mateixa que la primera. La solució és aplanar la curva atenuació-freqüència [equalització].

El que limita l’ample de banda són els símbols/segon. Si es desplaça tot cap a la dreta, l’ample de banda augmentarà ja que hi ha més àrea i energia. El punt clau és que la intersecció entre l’eix y i la gràfica no es mourà encara que tot es tiri cap a la dreta.

**Codificació:** independentment de si va bé o no, ocupa tot el canal, ja que no és mou el “punt clau”. Si es vol augmentar la velocitat s’hauria d’estirar la gràfica però sense tocar el punt. Per això, la codificació l’utilitzarem quan no hi ha ningú més al canal (medi de transmissió).

S’usarà quan estigui sol al canal/medi de transmissió (ethernet).

**Els sistemes de codificació:**

* Clocking (te sincronisme i es va reajustant el clock)
* Detecció d’errors
* poden millorar la relació senyal-soroll.

**Clock:** per mantenir el sincronisme a nivell físic, és importantíssim que hi hagi transicions (positiu-negatiu). Per això hi ha sistemes de codificació que implícitament fan això.

**Tipus de encoding schema:**

Si tenim nivell 2 HDLC no podríem triar sense perill el codificador **NRZ-L**. El problema seria si vinguessin molts 1’s o 0’s seguits. Però tot i així, amb HDLC no hi haurà mai sis 1’s seguits (amb 6 1’s no es perd el sincronisme), però si que podem tenir més de 6 0’s seguits.

**NRZI:** un 1 canvia i un 0 no canvia. Si fos del revés, 1 no canvi i 0 canvia, es podria utiltizar amb HDLC. No permet detectar errors, no permet clocking.

**Bipolar-AMI**: el 0 és absència de senyal, i l’1 canvia a dalt i a baix de forma alternada. Si hi ha dos 1 seguits a dalt, es detecta un error. El mateix codi a nivell físic (nivell 1) ja ens serveix per detectar errors. No serveix amb HDLC ja que els zeros no canvien els senyals. No permet clocking, però si detecció d’errors.

**Pseudoternary:** aquest és com l’anterior però els 0 canvien i els 1 no. Aquest si que es pot usar amb HDLC ja que com que no pot haver més de 6 1’s no passarà res i al 0 canviaran. *No permet clocking, però si detecció d’errors.*

**Manchester:** el 0 és cap a baix i l’1 és cap a munt. Una gran avantatge és que té totes les transicions possibles. Cada bit és una transició (si venen 2 1’s seguits i hi transició, si venen 2 0’s seguits hi ha transició). Un aspecte negatiu que per un sol bit posem dos símbols, això implica que la velocitat de transmissió es divideix per dos respecte els símbols/segon. S’aplica a xarxes d’alta velocitats.

\*Multisímbol: poder transmetre més d’un símbol en un bit augmentant la velocitat de transmissió sense augmentar els símbols per segons.

*Aquest sistema té clocking!!! Tenen transició, tenen una bona relació senyal-soroll! Però no detecta errors!!*

**Differential Manchester:** similar a l’anterior. Però al reves amb en comptes de fer dos 0’s. Te clocking, transicions però no detecta errors! Cada bit te dos simbols (per tant necessitarem el doble de l’ample de banda per obtenir la mateix velocitat.

Codis d’un bit per símbol, s’utilitzen normalment en llargues distàncies/ altes velocitats ja que te detecció d’errors però no permet clocking, per això van fer dos variants:

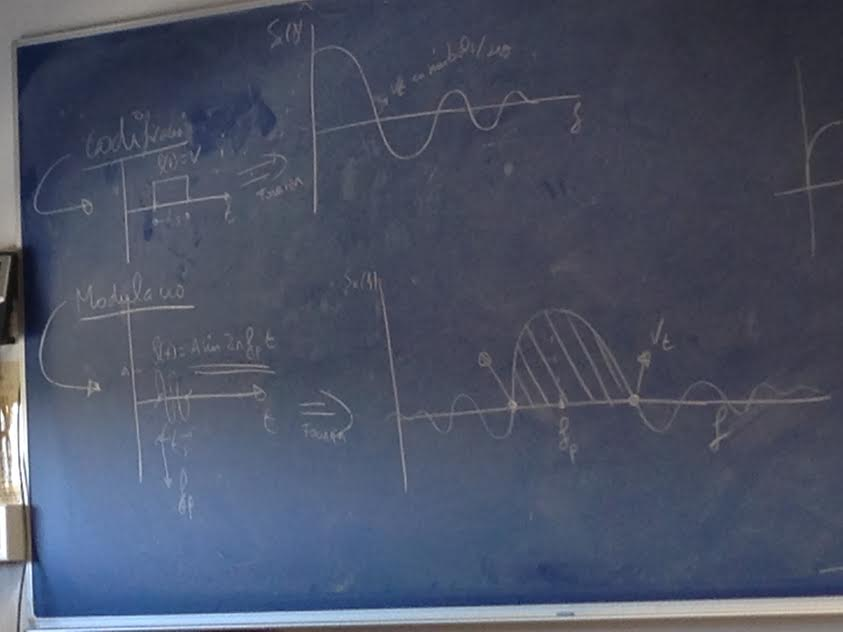
**B8ZS:** (EEUU) utilitza el bipolar-AMI però amb una modificació: quan hi ha vuit 0 seguits, els canvia amb un algorisme.

**HDB3:** (UE) utilitza el bipolar-AMI però amb una modificació: quan hi ha quatre 0 seguits, els canvia amb un algorisme.

La codificació serveix perquè l’ample de banda no sigui un problema, però a canvi de que només podem retransmetre nosaltres.

No perdre el sincronisme i detectar errors.

Contra més sofisticats són els codis, menys relació senyal-soroll tenen.



***01/03/2019***

***2.3 Modulació***

Desplaçem la ona fins a fp(frequencia portadora), que depen de Vt(velocitat de transmissió). Es el BW efectiu màxim.

Les operadores treballen a frequencies portadores concretes perquè no es barregin. Per aixó es barallen quan surten lliures noves bandes de frequencies lliures que es treballin en telefon mobils.

**ASK:** variació de l’amplada en el mateix temps de simbol. (A)

**BFSK:** variació de la frquencia portadora en el mateix temps de simbol. (fpt)

**BPSK:** es molt complex de construir, però te una molt bona relacio senyal-soroll. Variació de la fase en el mateix temps de simbol . (I)

**f(t)=A\*sin(2n\*fpt)+I(fase)**

**QAM:**  aquest es el quin s’utilitza actualment. Variació de l’amplitud i la fase es la millor opció, amb una freqüència portadora fixa, es pot codificar amb 3 bits ja que és un multinivell que té 8 nivells. Tots els punts es dibuixen sempre de forma simètrica. Si hi haguessin molt punts, costaria més distingir els diferents nivells a causa del soroll.

**128QAM =** 128 PUNTS en la malla, cadascun d’ells amb la seva amplitud i fase.

**El limit de punts es la relació senyal-soroll**

**QAM s’usa per tots aquells medi de transmissio que no s’hagi de compartir l’ampla de banda.(Radio, wifi,cable, fibra,..)\**

**La llum funciona amb ASK.**

Trellis: triem els 256 punts que millor treballin i que no siguin simètrics, es pot treballar amb 7 bits.

Sistema de codificació, però no es el moviment de modulació de lespectra portadora.

Com digitalitzar un senyal analògic, utilitzem el teorema de mostreig:

**PCM(Pulse Code Modulation):** és el mètode més estès de digitalització de la veu (el de la xarxa telefònica). “ Si un senyal és mostrejat de forma regular, a un ritme que com a mínim és el doble de la màxima freqüència que porta el senyal, no es perd res d’informació.”

**El mostreig no depèn del ample de banda sino de la màxima frequencia.**

**La mostre es treu amb PAM (pulse amplitude modulation)**

Això demostra que és possible que en dos punts es pot representar tot un senyal sabent la freqüència.

**Per tant, com que el canal vocal va des de 0Hz a 4000Hz, si agafem 8000 mostres per segon, no perdríem informació. Segons el teoria Nyquist, per no perdre informació es necessita el doble de la max freq (2\*MaxFreq).**

**Exercici PCM – ritme ha de llegir el fitxer. R=(2\*Fmax)[mostres/seg]s\*8bits[bits/mostra]**

**R = (4.000Hz\*2) \* 8 bits/mostres = 64 kbps**

**Tbit = 1/(64 kbps)**

**Ttrama = (1/64\*10³)\*8= 125 microseg**

Tot i així, hi ha un error a l’agafar el nombre de l’eix y ja que els nivells són finits, i per tant, s’agafa el que està més a prop provocant que hi hagi error. Perque estem forçats a fer una aproximació al fer de cofidicació digital.

**DADA IMPORTANT!**

**Velocitat que s’usa standard en els medis de transmissió = 64 kbps**

Delta Modulation: enviar menys bits amb la mateixa qualitat. En lloc de codificar amb 8 bits, es codifica amb 1 bit. Això es possible perquè utilitzen la funció escala (o puja o baixa).

***07/03/2019***

**2.4 Multiplexació:**

L’objectiu és optimitzar l’ús del link.

Wavelength division multiplexing: dividim les longituds d’ona, cada una d’elles a 10Gb/s. No es pot enviar més perquè no està normalitzat.

Synchronous time division multiplexing: el que dividim és el temps.

Statistical TDM: Un terminal ocuparà un canal només quan el necessiti, ja que de les maneres anteriors si li toca i no té res a enviar doncs no fa res, però podria ser que quan tingues paquets no els pogués enviar.

En un moment determinat això es podria congestionar i que no es pogués enviar tot alhora. Per solucionar-ho es posa un buffer on els paquets s’esperaran fins que puguin. Si tot i tenir els buffers entren en overflow vol dir que t’has equivocat i hauries d’haver posat una altra tecnologia.

Tot i així, es treballa conceptualment com l’Statistical però es treballa com l’Synchronous.

Per aconseguir l’Statiical TDM es treballar en nivell 2 on se li posa una trama amb l’adreça del terminal i les dades que estiguin enviant. Es pot enviar una trama hdlc per terminal o una per múltiples. En aquest últim cas es posa un camp de longitud per saber quan acaba un terminal i comença l’altre. Si les dades són molt grans es posa sol, però si és molt petita és pot posar amb múltiples.

Primer standard!!

**Normalització PDH:** busca la màxima eficàcia i el millor rendiment sobre les línies reals.

En el multiplexor entren 30 cables de 64Kbps i en surt un de 2Mbps(2048Kbps) que en realitat en suportarien 32. Un es fa servir per la telefonia (el 16) i un pel sincronisme.

*El que fan és dels 32 canals usen el primer per sincronitzar i l’últim per senyalitzar.* Aquestes trames tenen 32 canals i són de 125 microsegons.

Per tant 8bits/125\*10^-6=64kbps

El total que surt son 32 canals\*64 kbps/canal=2Mbps

I canals utils tindran una sortida real de 30\*64=1920 Kbps

Limit de funcionament de PDH es de 565 Mbps

E1 – 30 utils (payload) ( 32 total)

T1 – 24 totals i utils.

Com que les fibres òptiques tenen molta més velocitat el que fan és que del cable que surt del primer multiplexor es posa en un altre i així fins a 565Mbps que és el màxim que permet el procés de sincronisme (5 multiplexors).

Per saber els terminals que sortirien es fa: 30·4^4.

(Syncron

**SDH (SONET[EEUU])**:

Es fa el mateix que en el cas anterior però amb una distribuicio dels bits diferent.

1xSDH = 3xSONET

- la velocitat màxima és 10Gbps.

SONET: Tenim 90 clumnes x 9 files cada 125 microsegons (Amèrica).

Tenen 3 columnes son de la capçalera, la resta 87 son de payload.

Velocitat en un dels bits = 8/125\*10^-6 =64kbps

Total de bits en la matriu 9 files \* 90 columnes \* 8 = 6480bits/125\*10^-6 = 51,84 Mbps.

SDH: 270 x 9 cada 125 microsegons (Europa).

Tenen 9 columnes de capçalera i 261 de payload.

Vel = 9\*270\*8/125\*10^-6 =155,52Mbps

Vtpaquet = (261\*9\*8)/125\*10^-6 =150,3 Mbps

Per identificar un paquet a nivell 2 es fa a través d’un sistema de sincronització (flag)

Nivell físic:

· màx 565Mbps  PDH: canals i paquets de 64 Kbps.

· màx 40 Gbps  SDH: canals i paquets de 64 Kbps.

· màx 100 Gbps  ETH: paquets

Commutació circuits: thoughput constant · Delay fixe (molt petit, de l’ordre de microsegons)

Paquets: thoughput variable · Delay variable (de molt gran a molt petit de ms a microsegons)

\* Thoughput = quantitat de bps (útils)

**3.2 Commutació de cel·les**

· Trames: trames de longitud variable, trames grans i circuits virtuals  Frame Relay [Amb trames grans es busca màxim throughput (eficiència)]

· Cel·les: cel·les de longitud fixa, cel·les són molt petites (53 sockets), circuits virtuals. [Amb cel·les petites busquem mínima latència].  ATM

**Header Error Control:**

Utilitzem el CRC per detectar la capçalera. Si el resultat de calcular el crc hem dona tot lo anterior, això serà la capçalera. Sinó em de deixar passar un bit més i tornar-ho a calcular (0.7 · 10^-6 és el temps d’una trama).

Mentre la capçalera vagi bé indica que està sincronitzat.