

CONCEPTOS BASICOS DE REDES

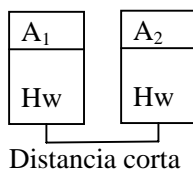
CONCEPTOS BASICOS DE REDES	1
TEMA 1.- INTRODUCCIÓN	3
Redes de comunicaciones	3
LAN (Local Area NetWork)	4
WAN (Wide Area NetWork)	5
Estandares y organizaciones	5
Estandarización	5
Modelos de referencia.....	6
Direccionamiento en Redes	7
TEMA 2.- TRANSMISIÓN DE DATOS (NIVEL FISICO)	8
Conceptos.....	8
Medios de transmisión.....	10
Par de hilos	10
Par trenzado	11
Coaxial.....	11
Fibra óptica	11
Codificación del canal.....	11
Modulaciones digitales	12
Técnicas de modulaciones	13
Digitalización.....	14
Multiplexación.....	14
Codificación de fuente	15
Modos de comunicación	16
Modos de transmisión.....	16
Sincronismo de reloj	16
Sincronismo de caracter.....	17
TEMA 3.- NIVEL ENLACE.....	18
Entramado.....	18
Detección de errores	18
Paridad	19
Block Sun Check (LRC)	19
CRC: Cyclic Redundancy Check.....	19
Control de errores.....	20
Stop&Wait	20
Retransmisión continua.....	23
Control de flujo	25
Ventana deslizante	25
Números de secuencia	26
TEMA 4.- REDES LAN	26
Introducción	26
Arquitectura de Niveles.....	26
Medios de transmisión.....	27
Guiados	27
No guiados	27
Topologías.....	28
Bus	28
Anillo	29
Estrella	29
Acceso al Medio.....	30
Token Ring	30

Anillos de baja velocidad ($\approx 4\text{Mbps}$).....	31
Anillos de alta velocidad ($\approx 20\text{Mbps}$).....	31
Ethernet	32
DIX vs IEEE	32
Acceso al Medio	33
Características del MAC Ethernet	35
Segmentación LAN	35
TEMA 5.- REDES WAN	37
Servicios	37
Orientados a la Conexión.....	37
No Orientados a la Conexión	38
Conmutación	38
Conmutación de circuitos	38
Conmutación de Paquetes por Circuitos Virtuales	39
Conmutación de Paquetes por Datagrama	40
Control de Flujo y de la Congestión	40
Encaminamiento	41
Técnicas de encaminamiento	42
TEMA 6.- INTERCONEXIÓN DE REDES	42

TEMA 1.- INTRODUCCIÓN

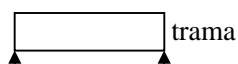
REDES DE COMUNICACIONES

Para comunicar dos aplicaciones se utilizan llamadas al S.O, es decir, tenemos la interfície de comunicaciones del sistema operativo (pipes, mailbox, FIFO, etc.)



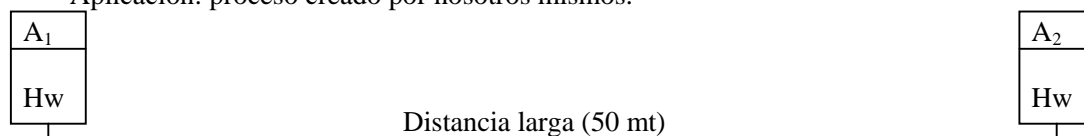
A la hora de comunicar aplicaciones que se encuentran en diferentes ordenadores, esta comunicación se ha de realizar mediante hardware, y no como antes que se hacía mediante ficheros utilizando las llamadas write y read. Ahora las aplicaciones siguen utilizando estas llamadas, pero el S.O las ha de interpretar de diferente forma.

La información se ha de convertir en señales eléctricas (voltajes) ya que el hardware es lo que utiliza. En el estado de reposo de un cable, puede haber un voltaje y tenemos que diferenciar cuando hay y cuando no hay información. Para diferenciar estas situaciones se encapsula la información en lo que se llama, una trama, es decir, sabemos cuando empieza y cuando acaba la información.



Por lo tanto necesitamos unas funciones para poder “encapsular” esta información (abrir fichero, encapsular información, convertir la información de voltajes, etc.). De este tipo de funciones surgió la idea de realizar la caracterización de las redes por niveles. Estos niveles son los siguientes:

- Físico: define el tipo de conector, la tarjeta..., el hardware necesario para crear la red.
- Enlace: proceso que actúa de interfície entre la aplicación y el hardware.
- Aplicación: proceso creado por nosotros mismos.

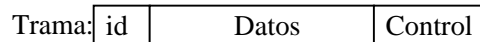


Con una distancia larga el voltaje que se recibe es mucho menor que la que se ha emitido (hasta puede tener diferente forma, que no sea cuadrada). Esta distorsión es debido a que el cable irradia, es decir, pierde información, y puede captar otras comunicaciones ya que actúa como antena. Para solucionar esto, el nivel enlace debe tener unas funciones que traten la gestión de errores:

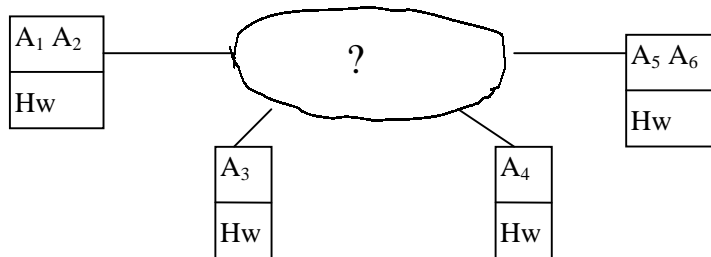
- Entramar.
- Detectar errores.
- Control de errores.
- Control de flujo: regular la cantidad de información que viaja a través de la red. Un ordenador puede ser mas lento que el que recibe la información.



Ahora tenemos dos ordenadores conectados pero donde puede haber mas de una aplicación ejecutándose a la vez. Ahora las aplicaciones llaman siempre al mismo proceso de enlace, y este no sabría a que aplicación dar la información que le llega, para solucionar esto tenemos la función de multiplexar. Para multiplexar hay que poner un identificador, este identificador de aplicación se guardará en la trama junto con la información de control de errores.



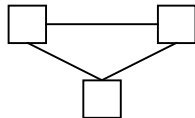
Ahora intentaremos conectar mas de dos ordenadores formando una red.



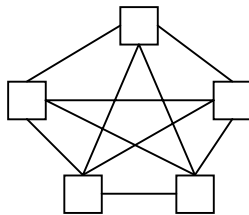
Con la conexión que hemos estado utilizando hasta ahora (tipo punto a punto) no es factible ya que aumenta de forma muy rápida el número de enlaces necesarios con pocos ordenadores, veamos un ejemplo:



2 ordenadores → 1 enlace



3 ordenadores → 3 enlaces



5 ordenadores → 10 enlaces

Con este tipo de conexión tenemos que con n ordenadores obtenemos:

$$\frac{N \cdot (N - 1)}{2}$$

enlaces.

Para solucionar estos problemas de conexión existen dos tipos de filosofías distintas para la conexión de ordenadores en red.

LAN (LOCAL AREA NETWORK)

- Esta conexión utiliza un medio (cable, infrarrojos, ondas, etc.) compartido por todos los ordenadores.
- Tienen que formar una topología (conexión en bus, en forma de anillo, estrella, etc.).

Estas dos características definen el nivel físico.

• Necesitaremos un algoritmo de arbitraje (acceso al medio) además de las funciones vistas anteriormente.

- Niveles del LAN:

- Aplicación.
- Transporte: busca errores, entrama la información, identifica la aplicación (dar un id al proceso).
- Interconexión (IP): unifica la información, busca direcciones (rutas), da un identificador de la red a la cual queremos acceder.
- Enlace:
 - LLC (Logical Link Control)
 - MAC (Medium Acces Control)
 - Realiza el entramado.
 - Acceso al medio.
 - Identificador del ordenador.
- Físico.

Ahora tenemos dos identificadores, uno para el ordenador y el otro para la aplicación que necesita la información.

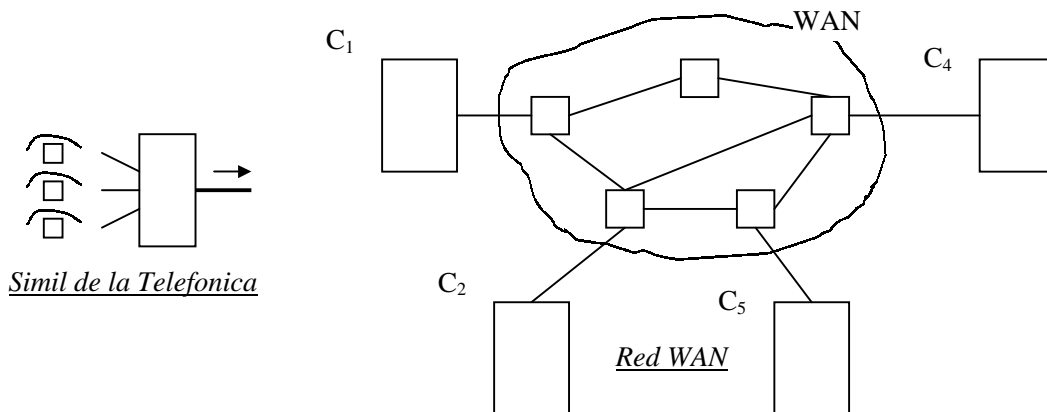
Para interconectar diferentes redes se ha de unificar la información, esta unificación la realiza el Router.

Ejemplo: telnet www.cisco.com
 ↑ ↑
 dirección IP
 identificador de la aplicación

WAN (WIDE AREA NETWORK)

Red de Area Extendida.

Se utiliza un medio muy rápido para la comunicación, pero solo tenemos un medio. Por lo tanto los ordenadores están conectados a un aparato que multiplexa y envía la información y busca rutas.



Los niveles de este tipo de redes son los siguientes:

- Aplicación.
- Transporte.
- Red: Interconexión (IP)
 - Red: busca rutas dentro de la WAN, conmutación.
- Enlace.
- Físico.

Necesitamos el mismo número de identificadores que en el caso de las redes LAN, aunque podrías utilizar más identificadores.

Las redes que se pueden encontrar actualmente son las siguientes:

- LAN: Ethernet (80% del mercado), Token Ring, FDDI, LAN ATM (10 % del mercado)
- WAN: Red telefónica Conmutación (RTC), RDSI, X25, Frame Relay, ATM.

ESTANDARES Y ORGANIZACIONES

Los sistemas cerrados se componen de un ordenador que dispone de software y hardware, propiedad de una empresa, para comunicarse con otros ordenadores. Si se instala otro software o hardware en otro equipo, no nos podremos comunicar.

En los sistemas abiertos tenemos un ordenador con software/hardware de diversos fabricantes y dispone de la posibilidad de conectar con otros ordenadores con otras características. Para establecer esta comunicación comprensible entre ambos, necesitamos una estandarización, organismo de estandarización, que define un estándar (definir reglas /algoritmos que permiten la comunicación entre varios subsistemas)

ESTANDARIZACIÓN

Las *ventajas* de una estandarización son las siguientes:

- Estimula la competitividad (sino hay un monopolio los precios bajan y por lo tanto se facilita el acceso a los usuarios).
- Flexibilidad a la hora de instalar la red (puedes poner equipos de distintos fabricantes). Ejemplo: tarjetas de distintas marcas, etc.

Las *desventajas* son las siguientes:

- Los organismos de estandarización son muy lentos (3 o 4 años aproximados para declarar un estándar).

- Quien compone los organismos de estandarización (empresas: interés por no dejarse aventajar por la competencia; política: comunicación de los votos, universidades: I+D..)

Ejemplo: Ethernet → IEEE 802.3

DIX (Digital-Intel-Xerox)

Ethernet II → Compatible mediante protocolo

- Demasiados organismos de estandarización.

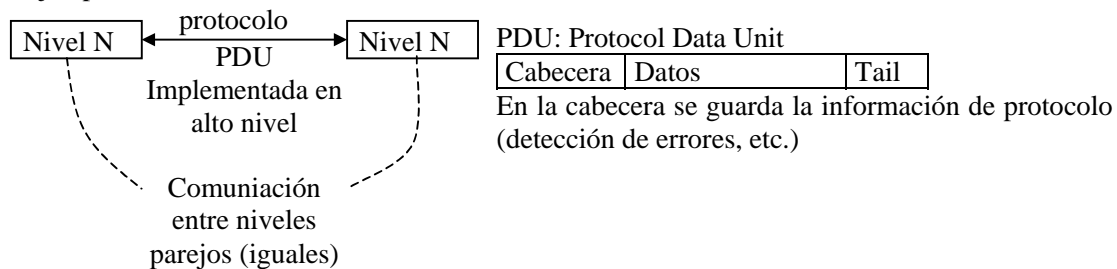
A continuación veremos unos organismos de estandarización:

- IEEE (Institution of Electrical and Electric Engineers): esta organización declaró el protocolo LAN pero no el LAN-ATM.
- EIA (Electrical Industries Asociation): declaró el cableado estructural.
- CCITT (International Telegraph and Telephone Consultatue Comittee): declaró la telefonía, actualmente esta absorbida por ITU (International Telecommunication Union), esta ultima declaró el ATM i la RDSI (comunicación digital)
- IETF (Internet Engineering Task Force): declaró el protocolo de Internet.
- ISO (International Standard Org): Modelos de referencia

MODELOS DE REFERENCIA

Los *modelos de referencia* intentan definir niveles. Cada modelo define sus funciones.

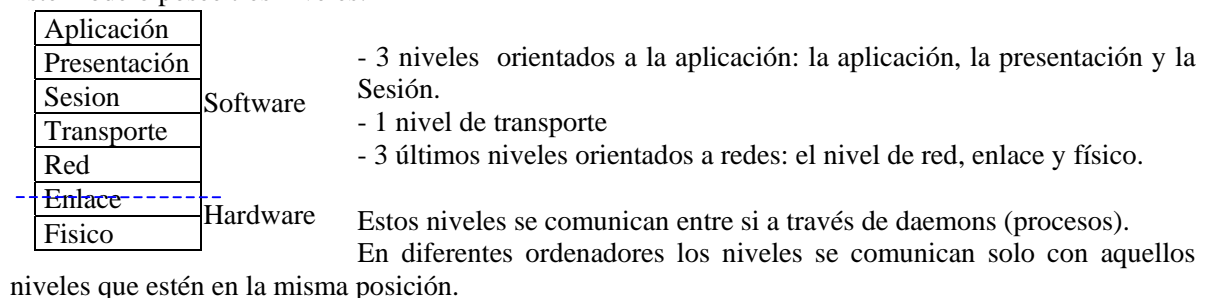
Ejemplo:



Un modelo de referencia es el siguiente:

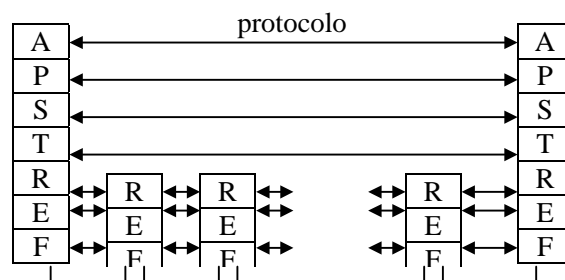
- OSI (Open System Interconexion) → El organismo de este modelo de referencia es ISO.

Este modelo posee tres niveles:



Cada equipo que forme parte de la red tendrá implementados los niveles de Red si sigue el modelo

OSI.



En diferentes ordenadores el nivel de aplicación solo podrá comunicarse con el nivel de aplicación del otro ordenador, así para todos los niveles. Si algún equipo intermedio no tiene todos los niveles de red, el nivel que en este presente tendrá una comunicación directa con el siguiente equipo que tenga el mismo nivel.

Esta comunicación se hace a través de un solo cable físico por lo tanto la comunicación entre aplicaciones se realiza de la siguiente forma:

La aplicación realiza un write y por lo tanto la aplicación que recibe la información ha de realizar un read.

El nivel de presentación encapsula la información de la aplicación y le pone una cabecera con el control de errores y/o flujo creando el PDU del nivel de presentación.

El nivel de sesión encapsula la PDU de presentación y añade su propia cabecera. La comunicación entre los diferentes niveles se realiza mediante una simple comunicación de procesos.

Así se va formando el PDU definitivo que llega al nivel de enlace, cuyo nivel encapsula el PDU de red le añade su cabecera y la tail (final del PDU) y lo envía al nivel físico el cual convierte dicha información en señales eléctricas y las envía a través de la red física.

Enlace-PDU		
Cabecera	R-PDU	Tail

Esquema del PDU de enlace, los demás PDU no tienen el tail

Las funciones que desempeñan los diferentes niveles son las siguientes:

- *Nivel A*: ofrece servicios de transferencia de archivos, gestión de correo electrónico, etc. Ofrece la posibilidad de crear sus propios servicios.
- *Nivel P*: está relacionado con la representación sintáctica de los datos (presentación de los datos). También está relacionado con la seguridad informática, es decir, con temas de encriptación.
- *Nivel S*: sincroniza las aplicaciones, por ejemplo cuando hay una caída de la red y al poco tiempo vuelve este nivel hace que las aplicaciones funcionen correctamente.
- *Nivel T*: intenta realizar una conexión correcta para esto realiza el control de flujo y de errores.
- *Nivel R*: busca rutas para llegar al destino, y da un identificador de red (dirección de ordenador y de interconexión)
- *Nivel E*: ofrece un servicio libre y seguro realizando el control de flujo y errores, pero a través de los terminales intermedios. Los niveles R no aseguran que hayan errores y por eso se realiza aquí, también, el control de flujo y errores.
- *Nivel F*: se ocupa de la electrónica y mecánica, tipologías, etc.

• Modelos TCP/IP

Los niveles de este modelo son:

- Aplicación
- Transporte: TCP-UDP. El TCP ofrece un control de errores (OSI) pero el UDP no ofrece este control de errores, al no utilizar este control tarda menos en realizar la comunicación y por lo tanto es muy útil para aplicaciones en tiempo real.
- Interconexión: IP
- Orientados a red: puede tener todos los niveles que se quieran.

La comunicación entre IP y el nivel de red se realiza mediante drivers.

El nivel A realiza las operaciones que se realizaban en los niveles A, P, S del modelo de referencia OSI.

DIRECCIONAMIENTO EN REDES

Normalmente debería haber una dirección por cada nivel, pero hay niveles internos que no necesitan, por lo tanto las direcciones necesarias para alcanzar una máquina remota son tres:

- Una dirección para identificar la *aplicación*: conocida como puerto, TCP/IP o A-SAP (Service Access Point)
- Dirección de *Internet* (IP). Identifica la red y el ordenador (la identificación de ordenador, en esta dirección solo es interesante para ordenadores conectados a la misma red).

- Dirección *física* o hardware, identifica la dirección propia de la tarjeta de red (identificador del ordenador real)

Ejemplo: LAN

A

TCP-UDP → puerto

IP → @ IP

LLC → la @ LLC identificaría la jerarquía de protocolos.

MAC → @ hardware

F

WAN: cada WAN tiene diferentes niveles.

- RDSI, RTC: solo definen el nivel físico. Los niveles R y E se comunican directamente. Necesitamos un aparato que nos adapte la señal digital a la del medio (si es telefónica un modem, si es RDSI otro aparato).
- X25: tiene los niveles F, E, R, es la única que sigue el modelo OSI.
- Frame Relay: define solo hasta el nivel enlace, ha pasado funciones del nivel R al nivel E y ha quitado funciones del nivel E.
- ATM: se define sus propios niveles: F, ATM (se corresponde al nivel E, pero totalmente diferente). En los terminales ofrece un nivel de adaptación de protocolos.

TEMA 2.- TRANSMISIÓN DE DATOS (NIVEL FISICO)

CONCEPTOS

El nivel físico define las características eléctricas de la transmisión.

Señal: onda electromagnética que se transmite, propaga por un medio de transmisión. Las señales se representa con un voltaje o una intensidad. Los diferentes tipos de señales son:

- *Continuas:* señal que no tiene saltos temporales. Función del tiempo (señal) continua (sin saltos).
- *Discretas:* señal que tiene discontinuidades.

Dominio temporal: representación de una señal en el tiempo.

Dominio frecuencial: representación de una señal en el eje de frecuencias. Esta representación se realiza mediante una transformación (*transformación de Fourier*).

Ancho de Banda: de la señal se define como aquellas componentes frecuenciales que contienen la energía de la señal. Si recuperamos estas señales recuperamos la señal.

Función de transferencia: función que depende de las frecuencias, esta función da el rango de frecuencias en las cuales oscila un circuito determinado.

Transmisión analógica: el receptor recupera con la máxima fidelidad la señal transmitida. El receptor ha de redibujar la misma forma de onda del transmisor. Una señal discreta se podría transmitir con la transmisión analógica, pero hay señales discretas que no.

Transmisión digital: esta transmisión define símbolos. El receptor toma decisiones durante un tiempo fijo para decidir que símbolo se ha recibido. En cada símbolo encapsula un número determinado de bits.

Velocidad de transmisión (V_t): número de bits que se transmite por unidad de tiempo (por segundo). Se mide en bits por segundo (bps)

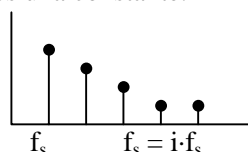
Velocidad de modulación (V_m): número de símbolos que se transmite por segundo. Se mide en baudios.

$$V_t = n \cdot V_m = V_m \cdot \log_2 M \quad n = \# \text{ bits} \wedge m = \# \text{ símbolos}$$

Tenemos que:

$$V_m = 1/T_{\text{símbolo}},$$

y que la frecuencia fundamental (la que lleva la mayor parte de la energía de la onda) es: $f_s = 1/T_s$, i la frecuencia de las señales restantes es $f_i = i \cdot f_s$. Donde i es una constante.



Ancho de Banda: esto es el camino por el cual discurren los datos de la comunicación de la red
 Por ejemplo: tenemos 2 símbolos (1 bit).

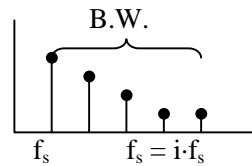
$$V_m = \frac{1}{T_s}; f_s \rightarrow B.W_s \text{ (ancho de banda de la señal)} = (f_i - f_s) = (i \cdot f_s - f_s) = f_s \cdot (i - 1) \approx \frac{1}{T_s} (i - 1) = \text{ancho de banda}$$

de la señal.

• Si T_s es muy grande entonces:

$$V_m' = \frac{1}{T_s'} \rightarrow \text{Si } T_s' \text{ aumenta entonces } V_m' \text{ disminuye} \rightarrow f_s \approx \frac{1}{T_s'} \text{ baja (la frecuencia fundamental}$$

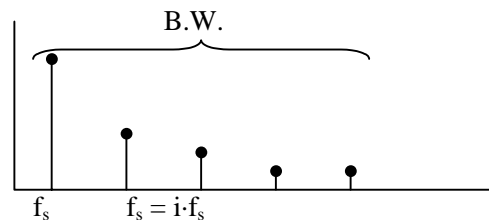
disminuye, i además el resto de las frecuencias se juntan) \rightarrow B.W. disminuye.



• Si T_s disminuye entonces:

$$V_m'' = \frac{1}{T_s''} \rightarrow \text{Si } T_s'' \text{ disminuye entonces } V_m'' \text{ aumenta} \rightarrow f_s \approx \frac{1}{T_s''} \text{ aumenta (la frecuencia fundamental}$$

se dispara, i además el resto de las frecuencias se alejan unas de otras) \rightarrow B.W. aumenta.



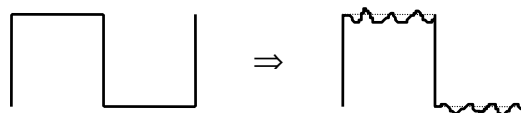
Aumenta el número de
símbolos transmisibles

Límite de la velocidad de modulación:

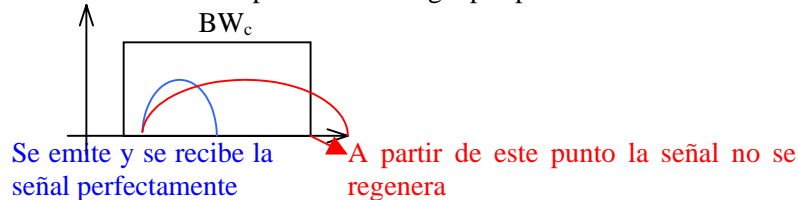
Si T_s es demasiado pequeña, tenemos el criterio de Nyquist, donde $V_{m(max)} = 2 \cdot B.W_c$ (ancho de banda del canal) eso si no se produce distorsión.

$$V_m = 2 \cdot B.W_s \leq V_{m(max)} = 2 \cdot B.W_c \rightarrow [B.W_s \leq B.W_c]$$

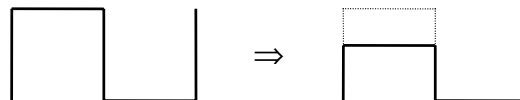
Distorsión: Al aumentar la velocidad de transmisión se producen retardos en las frecuencias (interferencia intersimbólica)



La interferencia intersimbólica es la pérdida de energía por parte de una de las señales.



Atenuación: reducción de la amplitud de la señal, es decir pérdida de energía de la señal.



Esta atenuación se debe al viaje a través del cable o medio de la señal, por lo tanto podemos acabar perdiendo la señal ya que el receptor no puede discernir señales. Nosotros la potencia la medimos en dB y no en vatios por lo tanto tenemos que transformar esta potencia en vatios a decibelios.

$$P \text{ dB} = 10 \cdot \log_{10} \frac{P}{P_{ref}} = 10 \cdot \log_{10} P \quad \text{donde } P_{ref}=1 \text{ W}$$

$$L = P(\text{atenuación}) = \frac{P_{emit}}{P_{recib}} \geq 1. \quad 10 \cdot \log L = L \text{ (dB)} = 10 \cdot \log_{10} \frac{P_e}{P_r} = (10 \cdot \log_{10} P_e) - (10 \cdot \log_{10} P_r)$$

$$L(\text{dB}) = P_e(\text{dB}) - P_r(\text{dB}) \rightarrow P_r = P_e - L.$$

De un cable se da la atenuación mediante un parámetro (α) que indica la atenuación del material según su longitud en kilómetros.

Ruido: el ruido es una señal no deseada en el medio de transmisión. Un cable aunque no este conectado a ningún aparato, este tiene una señal eléctrica, a esto se le conoce como ruido. Hay diferentes tipos de ruido:

- *Impulsivo:* ruido influenciado por otros medios, equipos, o otras señales eléctricas (tormenta eléctrica crea el efecto click). Tampoco se debe poner equipos cerca de circuitos de alta tensión, ya que el campo electromagnético es muy grande y esto afecta a la comunicación
- *Crosstalk (diafonía):* debido a que en la red de telefónica, los cables están pegados unos a otros, formando un gran cable. Los hilos que forman parte de este cable actúan como receptores y emisores a la vez, y al irradiar los hilos puede haber otros hilos que capten esta irradiación y si esta energía capturada de otro hilo es mayor que la que transmite, es esta la que se recibe al final. En equipos que transmiten y reciben también puede haber estos problemas ya que la entrada y salida de información esta muy cercana. Este tipo de ruidos se puede eliminar mediante circuitos especializados.
- *Térmico:* este ruido esta asociado al movimiento de los electrones. Estos se agitan con la temperatura y por lo tanto existe una señal eléctrica. La potencia del ruido medida en vatios se la calcula mediante la siguiente fórmula:

$$N(\text{watt}) = k \cdot T \cdot BW \quad \text{donde } k \text{ es la constante de Boltzman} = 13803 \cdot 10^{-23} \text{ J/}^\circ\text{K y } T \text{ la temperatura en } ^\circ\text{K}$$

La *relación señal ruido* (SNR o S/N) nos indica como de bueno es un receptor. S indica la sensibilidad del receptor (mínima potencia en el receptor para poder detectar la señal).

$$\left(\frac{S}{N} \right)_{\text{watts}} = \frac{P_r(\text{watts})}{N} = \frac{P_r(\text{watts})}{kTBw}$$

$$\left(\frac{S}{N} \right)_{\text{dB}} = 10 \log_{10} \left(\frac{S}{N} \right)_{\text{watts}}$$

La formula de Shanon indica la máxima velocidad a la que se puede transmitir por un medio, tal que la cantidad de errores es mínima (es decir que no nos afecta el propio ruido del cable).

$$C = \text{Capacidad del canal} = V_t \text{ maxima} = BW \cdot \log_2 \left(1 + \left(\frac{S}{N} \right)_{\text{watts}} \right) \text{ bps}$$

Si superamos C seguro que encontraremos errores. Si se transmite a una velocidad menor o igual a C tendremos un sistema que recupera la señal sin errores.

MEDIOS DE TRANSMISIÓN

Hay dos medios de transmisión:

- *Guiados:* encapsulan la onda electromagnética. (par de hilos, par trenzado, cable coaxial, fibra optica..)
- *No guiados:* no encapsulan la onda (radio, infrarrojos, satélite,...)

PAR DE HILOS

Aquí tenemos dos hilos, uno conectado a tierra y otro por el que se transmite. Solo permite una comunicación a muy corta distancia (30-40 metros). La velocidad de transmisión es menos a los kbps.

PAR TRENZADO

Para evitar el ruido que se producía en el caso anterior, trenzan los dos cables y entonces los dos cables cogen a la vez la diafonía y antes solo la cogía el cable de datos. Así se pueden transmitir a distancias mas largas y a velocidades anteriores.

El EIA 568-A estructuro este cable y así se consiguió la mayor velocidad de transmisión. Definió diferentes categorías:

- UTP-3 \rightarrow 16 Mbps
- UTP-4 \rightarrow 20 Mbps
- UTP-5 \rightarrow 100 Mbps

Todos pueden transmitir a una distancia de 100 metros. Este sistema es el mas utilizado.

COAXIAL

Se pone un hilo central protegido por un dieléctrico y este por una malla, así se aísla el hilo central de las interferencias del exterior. El dieléctrico o refracta o refleja las ondas del exterior. La distancia máxima de transmisión es de 500 metros (aunque podría llegar a 1 Km.) a una velocidad de 100 Mbps.

FIBRA ÓPTICA

Funciona con leds o láseres que emiten un haz de luz dentro de un cable fino. La única interferencia que podría haber seria la de la luz y para evitar esto el cable se encapsula con una capa opaca. No tiene ruido térmico y la longitud de transmisión puede ser muy grande al igual que la velocidad, el único inconveniente es que esta tecnología es muy cara.

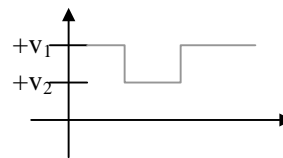
CODIFICACIÓN DEL CANAL

El cable transmite señales y no bits, por lo tanto se ha de codificar los bits en diferentes señales.

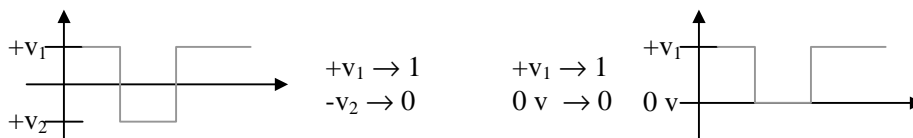
Hay so tipos de señales:

- *Unipolares*: todos los elementos de la señal (símbolos) o son positivos o negativos. Solo se transmite un bit por símbolo.

$$\begin{array}{ll} +v_1 \rightarrow 0 & -v_1 \rightarrow 0 \\ +v_2 \rightarrow 1 & -v_2 \rightarrow 1 \end{array}$$

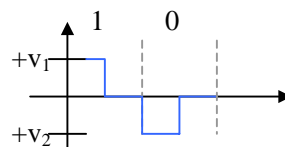


- *Polares*: los símbolos pueden ser positivos o negativos o 0.

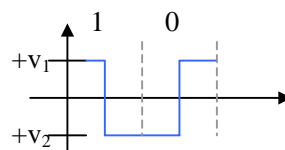


Existen dos tipos de categorías de señales polares:

- *Polares con retorno a cero*: en un tiempo de símbolo la señal eléctrica tiene que volver a cero y quedarse un tiempo:



- *Polares sin retorno a cero*: durante el tiempo de símbolo no vuelve a cero.



- En una transmisión digital existe un ancho de banda en banda base (es decir, tiene frecuencias cercanas a 0). En cambio la banda trasladada no está cercana a estas frecuencias nulas. Pero la red telefónica se comunica con banda trasladada de 300 Hz, por lo tanto interesa trasladar la frecuencia de la transmisión digital a esta frecuencia.
- No nos interesa que la señal tenga componente continua, ya que las componentes continuas tienen frecuencia cercana a cero. Al tener esta frecuencia, en el receptor los condensadores no dejan pasar esa frecuencia y por lo tanto esta señal no pasaría.
- Nos interesan señales que permitan una buena sincronización.

- *NRZ-L*: Non Return to Zero Level. Podría ser multinivel, es decir, tener símbolos para codificar mas bits a la vez.. Pueden producirse cambios de polaridad, y puede tener componentes continuas. No permite la extracción de reloj de la propia señal.

A graph of a step function on a coordinate plane. The x-axis is labeled from 0 to 10 with increments of 2. The y-axis is labeled from 0 to 4 with increments of 1. The function is defined by three horizontal segments: the first segment is at y=1 for x from 0 to 2, the second segment is at y=2 for x from 2 to 4, and the third segment is at y=3 for x from 4 to 6. The function is not defined for x > 6.

--	--

Figure 1 consists of eight bar charts arranged in two rows of four. The top row shows results for 'Visual' stimuli, and the bottom row shows results for 'Auditory' stimuli. Each chart has 'Correct' on the y-axis (0-100%) and 'Stimulus' on the x-axis (Visual, Auditory). The bars represent the percentage of correct responses for each combination of stimulus and response type.

Stimulus Type	Response Type	Visual Stimulus (%)	Auditory Stimulus (%)
Visual	Verbal	~95	~95
	Motor	~95	~95
Auditory	Verbal	~95	~95
	Motor	~95	~95

As se rompen las componentes continuas

CBXC- 12

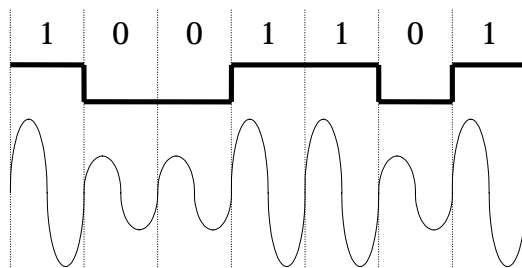
a las frecuencias adecuadas para la transmisión analógica, a este proceso se le denomina **modularización**. Este traslado de frecuencias se puede conseguir multiplicando la frecuencia inicial por una señal sinusoidal.

El receptor recupera la señal trasladando la frecuencia que viaja, hasta llevarla cerca de las frecuencias nulas.

TÉCNICAS DE MODULACIONES

- ASK (Amplitude Shifted Key): esta técnica codifica la señal según la amplitud. Esta técnica hace que la señal se pueda atenuar y por lo tanto confirmar cuando hay un 1 por un cero o viceversa. Se puede utilizar con fibra-optica la cual tiene una atenuación muy baja.

$$s(t) = \begin{cases} A_1 \sin(2\pi \cdot f \cdot t + \phi) & \rightarrow \text{si } 1, f, \phi \text{ son constantes} \\ A_2 \sin(2\pi \cdot f \cdot t + \phi) & \rightarrow \text{si } 0 \end{cases}$$

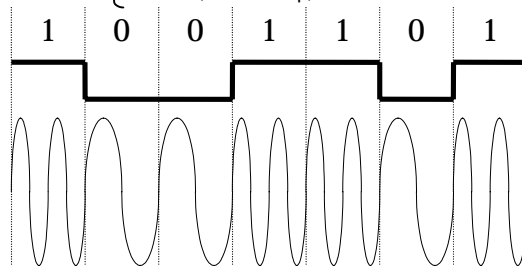


0 → A₀
1 → A₁

Poco utilizada
Mucha atenuación

- FSK (Frequency Shifted Key): esta técnica distingue los niveles según la frecuencia que encuentre. Esta técnica tiene como desventaja que se tiene que centrar en dos frecuencias diferentes y por lo tanto no se aprovecha todo el ancho de banda. Se utilizaba en módem de baja velocidad.

$$s(t) = \begin{cases} A \cdot \sin(2\pi \cdot f_1 \cdot t + \phi) & \rightarrow \text{si } 1, A, \phi \text{ son constantes y } f_1 \text{ varía} \\ A \cdot \sin(2\pi \cdot f_2 \cdot t + \phi) & \rightarrow \text{si } 0 \end{cases}$$

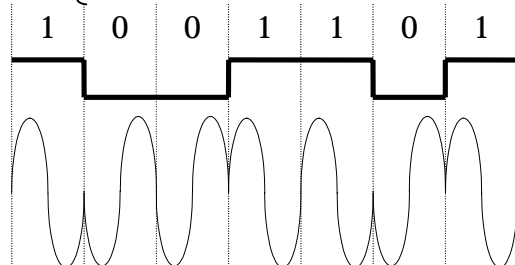


0 → f₀
1 → f₁

Modems de baja
velocidad (300-1200)

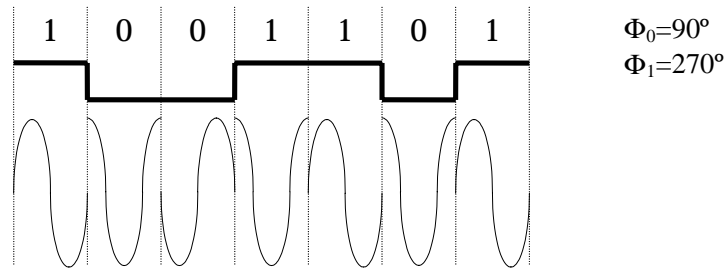
- PSK (Phase Shifted Key): la modulación se hace según la fase de la onda. Esta técnica sufre atenuación pero no tiene el problema de la técnica ASK. Puede utilizar todo el ancho de banda del canal, pero si existen bobinas en el camino estas hacen que se invierta la señal.

$$s(t) = \begin{cases} A \cdot \sin(2\pi \cdot f \cdot t + \phi_1) & \rightarrow \text{si } 1, A, f \text{ son constantes y } \phi_1 \text{ varía} \\ A \cdot \sin(2\pi \cdot f \cdot t + \phi_2) & \rightarrow \text{si } 0 \end{cases}$$



Φ₀ = 0°
Φ₁ = 180°

- PSK-diferencial: con esta técnica se soluciona el problema de la inversión del señal. Esta técnica lo que hace es sumar una fase a la frecuencia actual, cuando se cambia de símbolo.



También se puede modular señales multinivel con este método:

$$\phi_{11} = 0^\circ \rightarrow 11$$

$$\phi_{10} = 90^\circ \rightarrow 10$$

$$\phi_{01} = 180^\circ \rightarrow 01$$

$$\phi_{00} = 270^\circ \rightarrow 00$$

Este sería un PSK-4, es decir un PSK con 4 símbolos:

$$\text{Ejemplo: PSK-4 (4 fases)} \Rightarrow V_t = 2 \cdot V_m$$

Estas técnicas se pueden combinar para obtener más símbolos y por lo tanto ir más rápido. Pero esto tiene un límite por el hardware, y por lo tanto existe un límite de velocidad.

DIGITALIZACIÓN

Nosotros transmitimos en digital y la RTC inicialmente era totalmente analógica, actualmente tiene una parte digital (solo es analógico el proceso de recepción y el emisor). Por lo tanto la señal se ha de digitalizar para poder comunicarse con el resto de la RTC.



PAM: Pulse Amplitude Modulation

SAMPLER: muestreador.

El circuito va tomando muestras espaciadas T_m y cada muestra tiene su amplitud. Estas señales pasan por un cuantificador que lo que hace es asignar n bits a cada muestra. Se mira cada muestra y según esta tendrá unos determinados bits, y estos bits se codifican según unos de los métodos de codificación del canal.

$$T_m = \text{Intervalo a que se toman las muestras} = 1/f_m$$

$$\text{Cuantificador de niveles : } n \text{ bits/muestra} \Rightarrow 2^n \text{ niveles.}$$

$$\text{Teorema de muestreo de Nyquist : } f_m = \frac{1}{T_m} \geq 2 \cdot B_w$$

$$V_t = f_m \cdot n \text{ (N es el número de símbolos)}$$

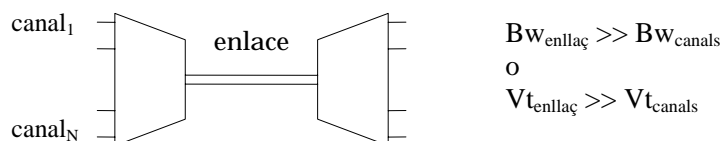
$$\text{Ejemplo : } B_w = 4 \text{ KHz, } 8 \text{ bits/muestra.}$$

$$f_m = 2 \cdot 4 \text{ KHz} = 8000 \text{ muestras/seg}$$

$$V_t = 8000 \text{ muestras/seg} \cdot 8 \text{ bits/muestra} = 64000 \text{ bps}$$

MULTIPLEXACIÓN

Tenemos N canales y cada canal con su determinada velocidad de transmisión, y su ancho de banda. Si queremos transmitir por el canal necesitaríamos N cables y esto es caro. Es más barato que solo haya una línea de comunicación entre dos puntos, y por lo tanto se han de juntar los canales en una sola línea, esta unión de canales la realizamos la **multiplexación**.

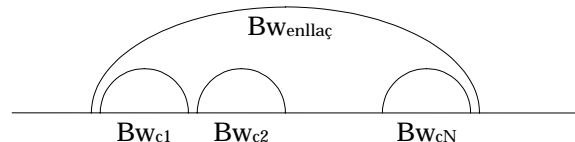


Hay diferentes formas de realizar esta multiplexación:

- **FDM** (Frequency Division Multiplexing): esta multiplexación se utiliza para transmisión analógicas. Modulan las frecuencias y las “suman”. Se han de preocupar muy bien de centrar estas frecuencias para que no se solapen y o superen el ancho de banda. A cada canal le asignamos una frecuencia diferente (portadora), con su BW correspondiente.

$$BW \geq \sum_{i=1}^N BW_i + \sum_{i=1}^N ABW_i \quad ABW_i \text{ es el ancho de banda para que no se solapen las frecuencias.}$$

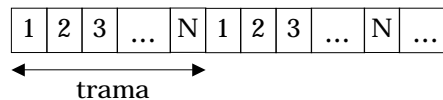
$$BW_{\text{enllaç}} > \sum_{i=1}^N BW_i \quad (\text{damos un ancho de banda de margen})$$



El receptor lo que hace es poner filtros en diferentes frecuencias y así tienen solo la señal de un canal que después es demodulada.

- **TDM** (Time Division Multiplexing): esta técnica se utiliza para las transmisiones en forma digital. Esta técnica va dando un cierto tiempo para transmitir a cada canal, por lo tanto se van alternando la información, esto es debido a que solo hay un ancho de banda. A cada canal se le asignan ranuras (slots). Existen diferentes formas de rellenar estos slots:

- **Sincrono(RDSI)**: se asigna un tiempo a cada canal tenga o no tenga información que transmitir. Una vez que termina con todos los canales, vuelve a empezar. A la unidad de slots de todos los canales se le llama *trama* digital. De todos los slots existe uno que lleva la información de control.



Trama = $N \cdot T_{\text{slot}}$ (N es el número de slots)

$$T_{\text{slots}} = \frac{\#bits}{slot} \cdot T_b \quad (T_b = \text{tiempo de bit} = \frac{1}{V_t}; V_t \text{ es la velocidad del enlace}).$$

Se ha de asegurar que $V_t \geq \sum_{i=1}^V V_{t_i}$ $V_{t_i} = V_t$ del canal y V_t del enlace

- **Asincrono(ATM)**: si un canal tiene información a transmitir se le asigna un periodo de tiempo y si no tiene información no se le asigna ningún tiempo. En esta técnica no existe el concepto de trama. Esta técnica permite que haya más usuarios que no en la anterior.

CODIFICACIÓN DE FUENTE

Los canales de transmisión transmiten bits y estos se convierten en señales eléctricas. Para enviar texto, este se ha de codificar de alguna forma en bits, para esto existen los siguientes códigos fuentes:

- ASCII: 7 bits por carácter.
- EBCDIC (IBM): 8 bits por carácter.

Los bits sufren una codificación de canal.

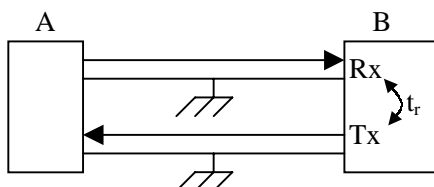
Todos los códigos tienen dos tipos de caracteres:

- **Imprimibles**: letras, números, signos, etc.
- **No imprimibles**: estos son los caracteres que se utilizan en redes. Algunos caracteres no imprimibles son: STX (start of text), ETX (end of text), DLE (Data Link Escape). Estos caracteres también tienen su codificación en ASCII o EBCDIC, por lo tanto cualquier usuario puede utilizar estos caracteres y por lo tanto tenemos que diferenciar cuando los utiliza el usuario y cuando la red para su control.

MODOS DE COMUNICACIÓN

Existen tres modos de comunicación:

1. **Simplex**: la información solo va en un sentido (la señal de un medidor de temperatura).
2. **Half Duplex (HDX)**: la comunicación se puede realizar en los dos sentidos pero no al mismo tiempo.



T_r : tiempo que tarda un equipo en cambiar de ser transmisor a ser receptor y viceversa (*tiempo de inversión de circuitos*). Este tiempo es aproximado al tiempo de propagación.

3. **Full Duplex (FDX)**: la comunicación se puede realizar en ambos sentidos al mismo tiempo. No puede haber un solo par de hilos, sino que se necesitan dos pares de hilos (uno para cada sentido de la comunicación). No necesita invertir los circuitos para poder recibir o transmitir. Si la señal es analógica se podría utilizar un solo par de hilos ya que se puede utilizar diferentes frecuencias para las comunicaciones en los diferentes sentidos.

MODOS DE TRANSMISIÓN

Existen dos modos de transmisión:

- **Asíncrono**: orientado a carácter: Se envía un carácter por trama, normalmente, si se envían mas caracteres la eficiencia disminuye.
- **Síncrono**: puede ser orientado a carácter o orientado a bit

La comunicación del nivel de enlace al nivel físico se realiza mediante tramas, y estas pueden ser de caracteres o de bits (en cuyo caso no se distingue ninguna codificación).

Independientemente de la transmisión, existen dos tipos de sincronismo:

- **Sincronismo de reloj**: (bit) esta sincronización intenta recuperar todos los bits.
- **Sincronismo de carácter**: si tenemos un modo de transmisión orientado a carácter el nivel físico tiene que detectar cual es la frontera entre caracteres.

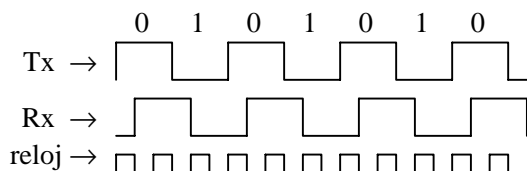
SINCRONISMO DE RELOJ

Asíncrono

- Se transmite con codificaciones de canal sencillas (NRZ), se utilizan para una comunicación a poca distancia entre equipos.
- Definen un estado de reposo, que será el estado convencional (todos unos). De esta forma en el estado de reposo (cuando nadie transmite) se ven V voltios.
- Tenemos que decidir cuando se toma una muestra de reloj.

El sincronismo de reloj ha de diferenciar si tenemos un uno o un cero, para ello toma una muestra de la información que se transmite en medio del bit.

En el sincronismo de reloj con una transmisión asíncrona los relojes de la transmisión y de la recepción son independientes.



$$f_{\text{reloj}} = N \cdot v_t \rightarrow T_{\text{reloj}} = 1/f_{\text{reloj}} = 1/(N \cdot v_t) \quad v_t = 1/T_b \rightarrow T_{\text{reloj}} = T_b/N \rightarrow T_b = N \cdot T_{\text{reloj}}$$

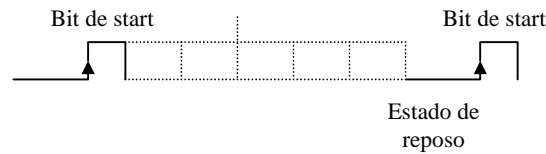
T_b = tiempo de bit

N = numero de pulsos del reloj por bit.

El inicio de T_b no tiene porque coincidir con el T_{reloj} . Lo que nos interesa es que la N sea grande (mayor que 16).

Tenemos que diferenciar cuando empieza y acaba el carácter, pero la línea cuando no transmite está en reposo, es decir, como si se indicara que tenemos un uno, y si al comenzar una transmisión el primer carácter empieza por un bit uno no se sabría cuando empieza el carácter ni la transmisión. Para solucionar esto se añade el bit de *start*.

- **Bit start:** facilita el sincronismo de carácter (un flanco de subida y además indica cuando empieza el sincronismo de bit.

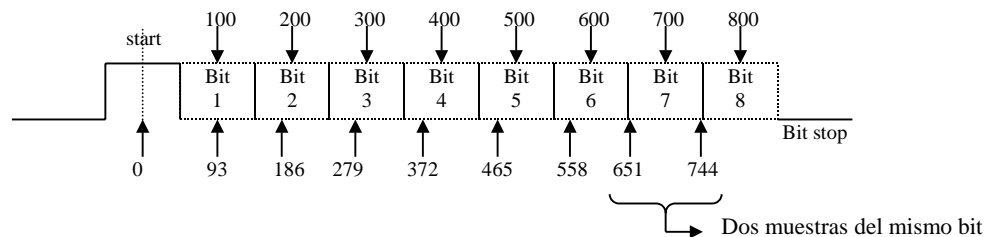


Si la línea de reposo es cero el bit de start tendrá que ser uno, y si el estado de reposo es uno el bit de start será cero.

Cuando termina el carácter (el ultimo bit del carácter) la línea se mantiene un cierto tiempo T en estado de reposo ($1, 1.5, 2 T_b$) para distinguir cuando se ha llegado al final del carácter.

Esta sincronización no es buena ya que el reloj de recepción se puede ir desfasando y llegar un momento en confundir un uno por un cero o viceversa. Se puede observar en el siguiente ejemplo:

$V_t = 100 \text{ Kbps}$ $T_b = 1/V_t = 100 \mu s$
 Reloj Rx \rightarrow desplazamiento de 7%
 8 bits por caracter



De esto se deduce que el sincronismo se ha de hacer por cada carácter, y por lo tanto no es muy eficiente.

Sincrono

- Las codificaciones de canal son mas complejas (Manchester).
- El reloj del receptor está sincronizado con el de transmisión.
- La frecuencia del reloj es $32 \cdot v_t$ (32 pulsos por bit) y utilizan un circuito DPLL (Digital Phase Lock Loop), que persigue la fase del reloj y intenta mantener el sincronismo.

La codificación Manchester cambia siempre de estado (es decir siempre hay transiciones). En cada transición el reloj cuenta 32 pulsos y si se pasa o se queda corto el circuito mencionado anteriormente totaliza el numero de pulsos hasta 32.

SINCRONISMO DE CARACTER

Si el modo de transmisión es orientado a carácter se necesita el sincronismo de carácter. Este sincronismo consiste en que el transmisor introduce dos caracteres al principio del carácter que nos interesa.

Transmisor

SYN		
-----	--	--

SYN es un carácter de sincronismo.

La técnica *scrambler* provoca las transiciones multiplicando la información por alguna constante.

TEMA 3.- NIVEL ENLACE

El nivel de enlace tiene las siguientes funciones:

- Enramado.
- Códigos detectores de errores (poner el código).
- Control de errores.
- Control de flujo.

La recepción es mas lenta que la transmisión, por lo tanto los buffers se pueden llenar y perder la información, el control de flujo hace reducir la velocidad de transmisión hasta la velocidad de la recepción.

ENTRAMADO

La información que le llega al enlace se empaqueta y se le añade una cabecera y una cola formando así la **trama**.

Head	Paquete	Tail
------	---------	------

Generalmente la cola es el código detector de errores y opcionalmente puede estar el sincronismo de trama. En la cabecera suele estar el sincronismo de trama y la información de control (una dirección, en redes LAN la dirección MAC).

Sincronismo de trama: tenemos que diferenciar cuando empieza y acaba una trama, el nivel físico no ve tramas sino caracteres. Según cual sea sincronización del nivel físico podemos tener dos tipos de distinción de tramas:

- Dentro de la trama es *orientado a carácter*: para diferenciar el comienzo y final de trama utilizaremos caracteres especiales.

STX → Start of Text → principio de trama

ETX → End of Text → final de trama.

STX	Contr	Datos	Cod	ETX
-----	-------	-------	-----	-----

Si dentro de la trama hay un ETX hay que diferenciarlo, para ello existe la técnica Character Stuffing (transparencia de la información) que utiliza el carácter DLE (Data Link Escape) el cual se añade al STX y ETX de principio y final de trama.

DLE	STX	Datos	DLE	ETX
-----	-----	-------	-----	-----

Pero, también nos podemos encontrar la secuencia DLE+ETX dentro de los datos de la trama, por lo tanto se pueden producir errores, para solucionarlo añadimos un DLE si se encuentra otro carácter DLE

DLE	STX	DLE.DLE	ETX	DLE	ETX
-----	-----	---------	-----	-----	-----

Ejemplo: datos → A DLE ETX

Trama → DLE + STX | A + ~~DLE~~ + DLE + ETX | DLE + ETC

Otra técnica para delimitar tramas sería indicar la longitud del campo de datos.

SOH	Control	Longitud	Datos	código
-----	---------	----------	-------	--------

SOH → Start Of Head, indicación de principio de cabecera.

- *Orientado a bit*: podemos utilizar un campo para indicar la longitud o utilizar flags de apertura y cierre. Un flag de apertura y cierre es el siguiente

01111110 → seis unos

Dentro de los datos puede haber una secuencia como el del flag de apertura y cierre. Por lo tanto hay que utilizar otra vez la transparencia de la información (bit Stuffing). Esta técnica lo que hace es que justo después de cinco unos inserta un cero y continua.

Ejemplo: 01111110 | 100101011111010011 | 01111110

01111100 → 01111100

DETECCIÓN DE ERRORES

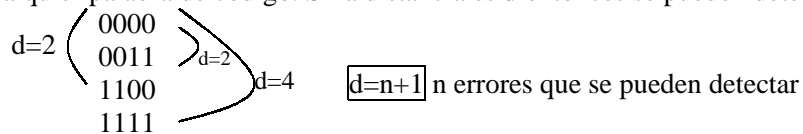
La detección de errores añade información redundante para detectar los errores.

Existen dos técnicas para la detección de errores:

- FEC (Forward Error Control): detecta y corrige errores.

- Feedback Error Control: esta técnica solo detecta errores, y por lo tanto una vez detectado se ha de realizar un control de flujo (volver a enviar la información, es decir, la trama).

Los dos sistemas se basan en la distancia hamming, que define el mínimo número de bits diferentes entre cualquier palabra de código. Si la distancia es d entonces se pueden detectar $n+1$ errores.



Para poder corregir un error se ha de cumplir que la distancia hamming sea $d = 2n+1$ y así se pueden corregir n errores.

PARIDAD

Esta técnica añade un bit de paridad a la palabra. La paridad puede ser par o impar:

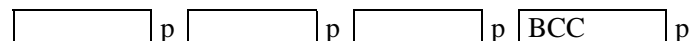
- Paridad par: la suma de unos tiene que ser par: 1011101 1
- Paridad impar: la suma de unos tiene que ser impar: 1011101 0

Esta técnica solo sirve para detectar un error.

BLOCK SUM CHECK (LRC)

Longitudinal Redundancy Check

Se van transmitiendo caracteres con su propia paridad y al final se pone un carácter de paridades (*Block Check Character- BCC*). Se mira la paridad en sentido horizontal (paridad de un carácter) y en sentido vertical para formar el BCC.



		bit paridad
C_1	→ 1011101	1 → par
C_2	→ 0110101	0
C_3	→ 0001111	0
C_4	→ 1110110	1
BCC	→ 1101110	1
	↓	
		impar

Este sistema detecta mas de un bit erróneo dentro de un carácter, aunque hay errores que no se pueden detectar.

El **Checksum** es otra técnica de detección de errores per es utilizado en TCP y IP. Lo que hace es sumar palabras de 16 bits en complemento a uno.

CRC: CYCLIC REDUNDANCY CHECK

Esta técnica es utilizada en redes LAN.

Se basa en códigos polinomiales, ya no tenemos caracteres, sino una secuencia de bits.

BER: Bit Error Rate, probabilidad de que un bit sea erróneo (P_b), este parámetro depende del medio, ruido, etc.

A partir de P_b podemos saber la probabilidad de que una trama sea errónea o no (P_f). Si tenemos una trama de L bits se puede deducir lo siguiente:

$$P_f = 1 - (1 - P_b)^L \approx L \cdot P_b \text{ si } L \cdot P_b \ll 1 \quad (1 - P_b) \rightarrow \text{Probabilidad de trama.}$$

Si la longitud de trama L es muy grande entonces $(1 - P_b)^L$ tiende a cero y por lo tanto la P_f de trama tenderá a uno, y por lo tanto casi siempre se producirá un error.

En CRC se define la secuencia de bits a transmitir como un polinomio:

$$M(x) = S_{k-1}x^{k-1} + S_{k-2}x^{k-2} + \dots + S_1x + S_0$$

Donde la k representa el número de bits que contiene datos. Estos k bits de datos se pueden representar por un polinomio de grado $k-1$.

$$S_j=0 \text{ si } T_x=0$$

$$S_j=1 \text{ si } T_x=1$$

Para calcular el CRC se necesita un polinomio generador ($G(x)$), el grado de este polinomio tiene que ser L donde L es el número de bit del CRC) con el que se realizan las siguientes operaciones para calcular el CRC y para recuperar la secuencia de bits:

$$CRC = resto \left[\frac{M(x) \cdot x^L}{G(x)} \right] \quad \frac{M(x) + CRC}{G(x)} = 0$$

Ejemplo:

datos: 11100110 $k=8$ bits

CRC= 4 bits = L

$$M(x) = x^7 + x^6 + x^5 + x^2 + x$$

$G(x) = x^4 + x^3 + 1$ escogido al azar.

$$\begin{array}{r} M(x) \cdot x^L = x^{11} + x^{10} + x^9 + x^6 + x^5 \\ \underline{x^{11} + x^{10} + x^7} \\ x^9 + x^7 + x^6 + x^5 \\ \underline{x^9 + x^8 + x^5} \\ x^8 + x^7 + x^6 \\ \underline{x^8 + x^7 + x^4} \\ x^6 + x^4 \\ \underline{x^6 + x^5 + x^2} \\ x^5 + x^4 + x^2 \\ \underline{x^5 + x^4 + x} \\ x^2 + x \end{array}$$

$$x^2 + x \rightarrow L = 4 \text{ bits} \rightarrow 0110$$

11100110	0110
Datos	CRC

Con un $G(x)$ bien escogido se pueden detectar:

- errores de un solo bit.
- errores de un número impar de bits.
- dos bits erróneos.
- ráfagas de tamaño menor que k bits.
- algunas ráfagas de tamaño mayor que k .

Las *ráfagas* son el número de bits que hay entre bits erróneos: 11001100 ráfaga = 6.

Un $G(x)$ es el CRC-CCITT de 16 bits = $x^{16} + x^{15} + x^2 + 1$.

CONTROL DE ERRORES

Una vez detectado un error se puede realizar:

- Echo checking: se transmite y se recibe lo que se ha enviado.
- ARQ's (automarting Repeat Request): el receptor al recibir los errores pide que se le envíe otra vez la información automáticamente.

Existen dos tipos de ARQ's:

- **Idle-RQ**: la técnica se llama Stop&Wait. Se espera respuesta de si lo que se ha enviado es correcto o no.
- **Continuos-RQ**: tenemos la *selective Repeat* (se envía solo la información que ha sido errónea) y *Go_back_N* (envía toda la información otra vez).

STOP&WAIT

Se definen dos tipos de tramas:

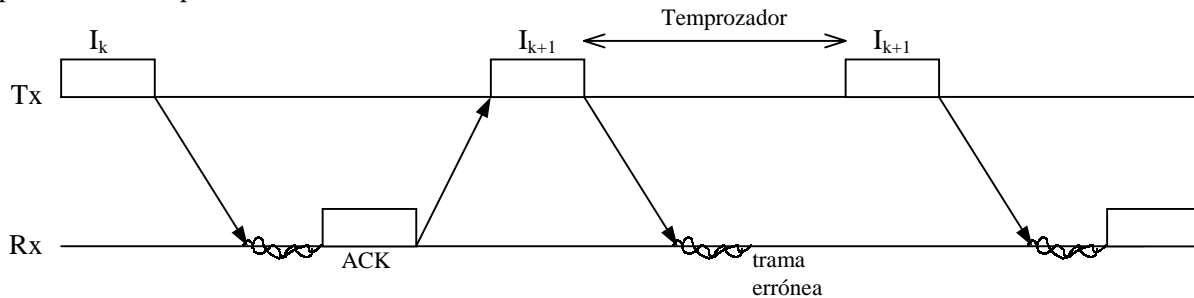
- I_k = trama de información.
- Tramas de reconocimiento: estas tramas pueden ser **ACK** (acknowledge, reconocimiento positivo) o **NACK** (NON-ACK, reconocimiento negativo). Estas tramas son de pocos caracteres, entre 3 i 4 caracteres.

Existen dos tipos de Stop&Wait:

- *Implícito*: utiliza solo ACK.
- *Explícito*: utiliza los dos tipos de reconocimiento ACK y NACK.

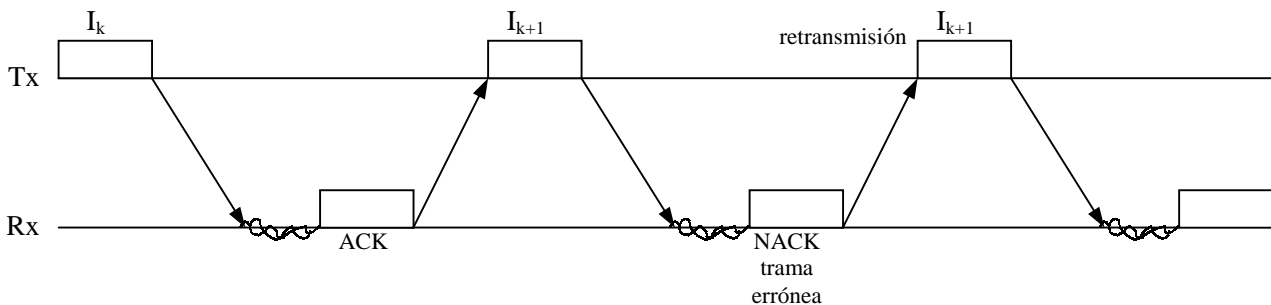
Implícito

Si hay un error el receptor no envía ninguna trama de reconocimiento, solo lo envía cuando se ha recibido correctamente la información. Por lo tanto el transmisor cada vez que envía una trama activa un temporizador, si se consume el tiempo de dicho temporizador antes de enviar el ACK el receptor, entonces el transmisor lo identifica como si hubiera habido un error. este temporizador se activa cada vez que se envía el primer bit de la primera trama a transmitir.



Explícito

Si hay un error el receptor transmite una trama de tipo NACK, pero el temporizador se sigue manteniendo debido a que la transmisión de receptor a transmisor puede ser errónea.



Eficiencia en el protocolo (U_p) (sin errores)

La eficiencia en el protocolo es el tanto por ciento de tiempo en el que el transmisor está enviando tramas. Nos interesa que la eficiencia sea 1, es decir, que el 100% del tiempo se este transmitiendo información.

$$U_p = 1 \rightarrow \text{objetivo} \quad U_p = \frac{T_t}{T_c}$$

T_t = tiempo de transmisión de una trama

$$= L \cdot T_b = \frac{L}{V_t}$$

T_{ack} = tiempo de transmisión de la trama ACK

$$= L_{ack} \cdot T_b = \frac{L_{ACK}}{V_t}$$

T_p = tiempo de propagación, tiempo que tarda 1 bit en viajar desde el transmisor al receptor.

$$= \frac{\text{dist(metros)}}{V_p(m/s)} \quad \text{donde } V_p = \text{velocidad de propagación} = ctn \cdot C$$

$L \rightarrow$ # bits de una trama

$T_t \rightarrow$ tiempo de trama

$T_c \rightarrow$ tiempo de ciclo

$T_b \rightarrow$ tiempo de bit

$C \rightarrow$ velocidad de la luz

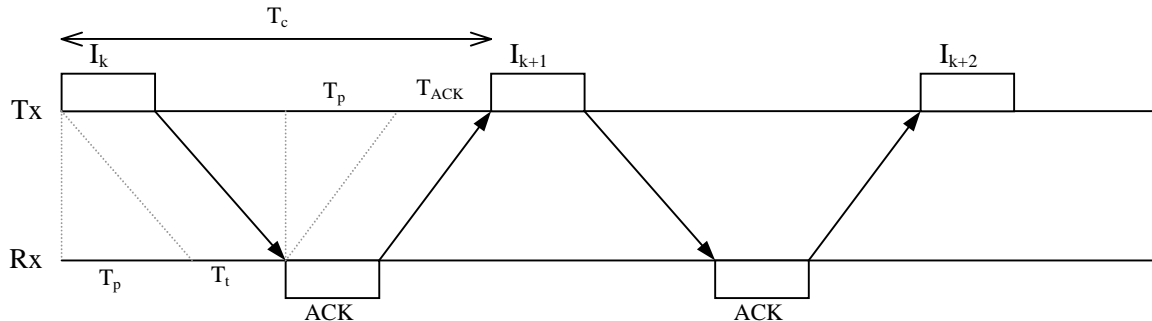
$V_p \rightarrow$ velocidad de propagación

T_{proc} = tiempo de proceso de una trama o 1 ACK. Este tiempo es tan pequeño que no nos afecta.

$$= T_{proc} \ll T_t + T_p + T_{ack}$$

T_c = Tiempo entre el primer bit de la trama hasta el primer bit de la siguiente trama.

$$= 2 \cdot T_p + T_t + T_{ack} + 2 \cdot T_{proc}$$



$$U_p = \frac{T_t}{T_c} = \frac{T_t}{2 \cdot T_p + T_t + T_{ACK}} \stackrel{L_{ack} \ll L_t}{\approx} \frac{T_t}{2 \cdot T_p + T_t} \approx \frac{1}{1 + 2 \cdot \frac{T_p}{T_t}} \text{ esto ha de tender a uno.}$$

Para que U_p tienda a uno, podemos hacer que:

- $\frac{T_p}{T_t}$ tienda a cero $\rightarrow T_p$ será muy pequeño, tiempo de propagación será muy pequeño (distancia entre Tx y Rx ha de ser muy pequeña).
- T_t crezca: pero si crece mucho es porque L crece (el tamaño de trama) y por lo tanto la probabilidad de errores de la trama también crece.

Temporizador

hay que asegurarse de que en el peor de los casos, se de el tiempo suficiente para que llega el ACK del receptor. un valor aproximado debería ser:

$$T_{temp} \geq 2 \cdot T_p + T_{ack} \text{ de esta forma nos aseguramos de que el ACK llega a tiempo.}$$

Si tenemos un circuito half duplex se ha de sumar el tiempo de inversión del circuito al tiempo de ciclo.

Eficiencia en el protocolo (con errores)

Ahora tenemos un tiempo de ciclo diferente:

T_c' : tiempo que transcurre desde que se envía una trama hasta que se puede enviar la siguiente trama que no sea la misma.

N_c = numero medio de transmisiones de una trama.

$$U_{pmedia} = \frac{T_t}{T_c'} = \frac{T_t}{N_c \cdot T_c}$$

$P_f \rightarrow$ probabilidad de que una trama sea errónea.

$1 - P_f \rightarrow$ probabilidad de que una trama sea correcta.

$$T_c' = T_c \cdot (1 - P_f) + 2 \cdot T_c \cdot (1 - P_f) \cdot P_f + 3 \cdot T_c \cdot (1 - P_f) \cdot P_f^2 + \dots = (1 - P_f) \cdot T_c \cdot \sum_{i=1}^{\infty} i \cdot P_f^{i-1} = T_c \cdot N_r$$

$$N_r = (1 - P_f) \cdot \sum_{i=1}^{\infty} i \cdot P_f^{i-1}$$

$$\sum_{i=1}^{\infty} i \cdot P_f^{i-1} = \frac{d}{dP_f} \left(\sum_{i=1}^{\infty} P_f^i \right) = \frac{d}{dP_f} (1 + \sum_{i=1}^{\infty} P_f^i - 1) = \frac{d}{dP_f} \left(\sum_{i=0}^{\infty} P_f^i - 1 \right) = \frac{d}{dP_f} \left(\frac{1}{1-P_f} - 1 \right) = \frac{d}{dP_f} \left(\frac{P_f}{1-P_f} \right) = \frac{1-P_f+P_f}{(1-P_f)^2} = \frac{1}{(1-P_f)^2}$$

$$N_r = \frac{1}{1-P_f}; \quad U_{p_{media}} = \frac{T_t}{T_c} = \frac{T_t}{N_r \cdot T_c} = (1-P_f) \cdot \frac{T_t}{T_c} = (1-P_f) \cdot U_{p_{max}}$$

El Stop&Wait es muy malo en largas distancias, ya que es el tiempo de propagación (T_p) donde influye la distancia i este está en el denominador.

$$U_p = \frac{T_t}{2 \cdot T_p + T_t + T_{ACK}}$$

El temporizador solo ha de preocuparse de dar tiempo ha llegar al ACK, y este no importa en la eficiencia.

RETRANSMISIÓN CONTINUA

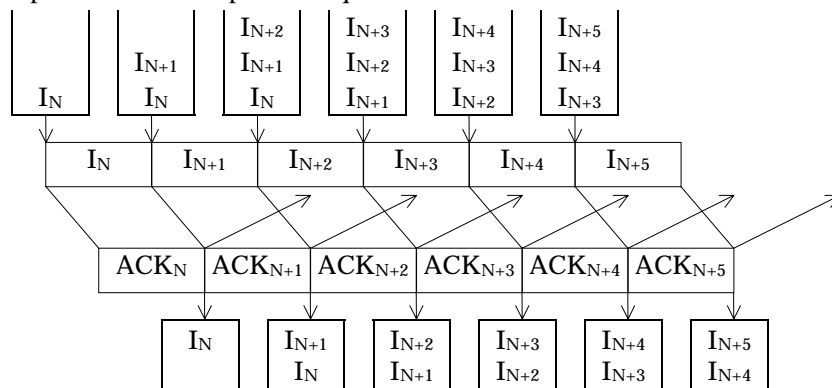
El problema del Stop&Wait es el tiempo de bloqueo o de espera a que llegue el ACK.

$$T_{bloqueo} = T_c - T_t = 2 \cdot T_p + T_{ack}$$

El transmisor, por lo tanto solo transmite en $T_c - T_{bloqueo}$.

$$U_{bloqueo} = 1 - U_p = \frac{T_c - T_t}{T_c}$$

En este tiempo de bloqueo el transmisor solo espera ha transmitir y no hace nada más, por lo tanto se pierde tiempo. Para aprovechar el tiempo de bloqueo esta la **retransmisión continua**.



Con esta retransmisión tenemos que buscar una estrategia de retransmisión por si alguna trama es errónea. En este caso la eficiencia máxima si no hay errores es:

$$U_p = 1 \quad U_{bloqueo} = 0$$

ya que siempre estamos transmitiendo, y si no se transmite es que no hay nada que transmitir, y no porque estemos bloqueados.

En este sistema se necesitan memorias para guardar la trama enviada hasta que nos llegue la trama ACK correspondiente a la trama, para poder retransmitir la trama en caso de error. Esta memoria es el *buffer de transmisión*.

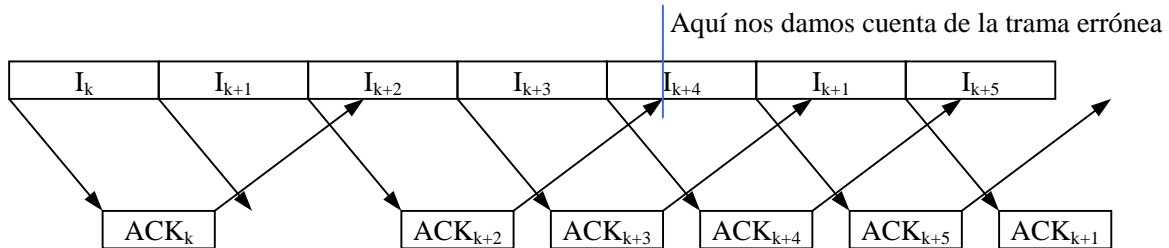
En el receptor tenemos el buffer de transmisión, porque tenemos que ordenar las tramas, por si el receptor es mas lento, y para poder realizar copias (para saber si una trama es copia de otra o no).

Existen dos estrategias de retransmisión: la **repetición selectiva** i el **Go_back_N**.

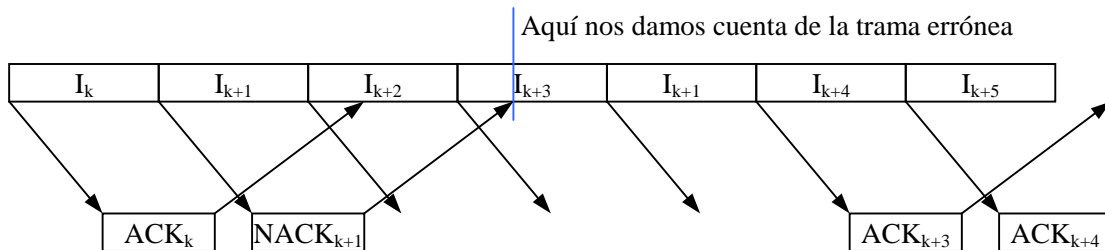
Repetición selectiva

En este caso solo retransmitimos las tramas erróneas. Hay dos formas de trabajo:

- **Implícito:** el receptor solo transmite los ACK. Cuando una transmisión es incorrecta, al transmisor no le llega el ACK pero se sigue transmitiendo tramas, hasta que se da cuenta que le falta una trama que es cuando la saca del buffer y la vuelve a enviar, una vez enviada se siguen enviando tramas normalmente. También utiliza el temporizador, por lo tanto se utiliza al igual que en el Stop&Wait, se utiliza por si hay tramas consecutivas erróneas (y así se impide que se llenen los buffers), o por si es la última trama.



- **Explícito:** el receptor transmite NACK, además de los ACK. transmite los NACK si la trama ha sido errónea. Los ACK son **acumuladores**, es decir, si transmite el ACK_{k+5}, esto nos indica que ha reconocido todas las tramas hasta la I_{k+5}, un solo reconocimiento de una trama reconoce esa trama y todas las anteriores. Cuando se envía el NACK no se pueden enviar ACK hasta que no nos llegue el ACK correspondiente a la trama que había provocado el error.



En todos estos sistemas existe el temporizador por si la transmisión es errónea en los dos sentidos o de receptor a transmisor.

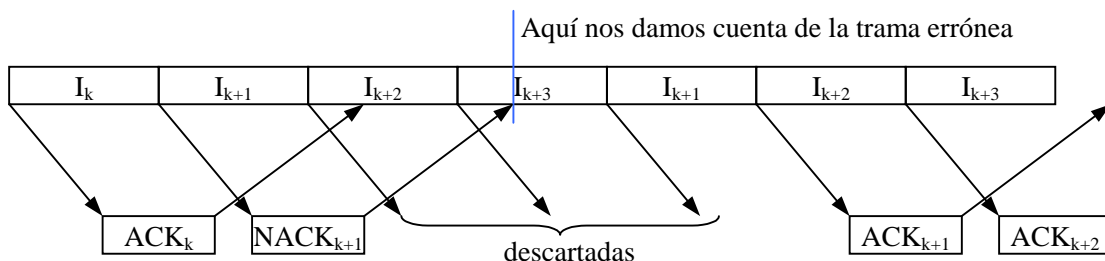
La eficiencia es la siguiente:

- Sin errores: $U_p=1$
- Con errores: $N_t \rightarrow$ número de transmisiones para una trama. $N_t=1/(1-P_f)$.

$$U_p = \frac{T_t}{N_t \cdot T_t} = \frac{1}{N_t}$$

Go_Back_N

Este sistema es explícito y cuando una trama es errónea, descarta las tramas que vengan hasta que la trama sea correcta. Este sistema recibe las tramas ordenadas.



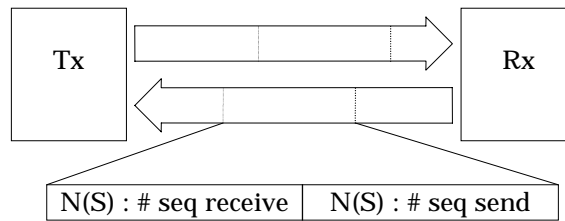
La eficiencia de este protocolo es:

- Sin errores: $U_p=1$
- Con errores:

$$U_p = \frac{T_t}{(N_t - 1) \cdot T_c + T_t} \stackrel{T_{ACK} \rightarrow 0}{=} \frac{T_t}{(N_t - 1) \cdot (2 \cdot T_p + T_t) + T_t} = \frac{T_t}{(N_t - 1) \cdot 2 \cdot T_p + (N_t - 1) \cdot T_t + T_t} = \frac{T_t}{(N_t - 1) \cdot 2 \cdot T_p + N_t \cdot T_t}$$

Si $N_t \rightarrow 1$ entonces $U_p = 1/N_t$.

Piggy Backing



$$U_p = \frac{T_t}{T_c} = \frac{T_t}{2 \cdot T_p + T_t + T_{ack}} \stackrel{T_t \gg T_{ack}}{=} \frac{T_t}{2 \cdot T_p + T_t} \text{ caso óptimo}$$

CONTROL DE FLUJO

El receptor controla la velocidad a la que el transmisor emite tramas, porque el receptor puede ser mas lento que el transmisor. Para hacer el control de flujo el receptor bloquea al transmisor.

Un método es la **ventana deslizante** (sliding window).

VENTANA DESLIZANTE

K_t = ventana de transmisión, que normalmente será una constante. La ventana de transmisión es el número de tramas que se pueden transmitir sin bloquearse.

L_{inf} = límite inferior de la ventana.

L_{sup} = límite superior de la ventana.

Si se quiere transmitir una trama se mira si tenemos espacio en la ventana para transmitir, esto se hace según el algoritmo siguiente:

si $L_{sup} - L_{inf} < K_t \rightarrow$ podemos transmitir;

$L_{sup} ++$;

sino ($L_{sup} - L_{inf} = K_t$) \rightarrow bloqueo del transmisor;

Si $L_{sup} - L_{inf} = K_t$ se bloquea el transmisor hasta que se reciba un ACK, con el que se aumenta el L_{inf} :

Solo se pueden transmitir K_t tramas, de esta forma nos aseguramos que el buffer del transmisor haya, como mucho, K_t tramas..

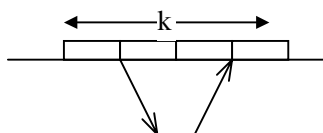
K_r = ventana de recepción, se define como el tamaño del buffer de recepción. Esta ventana se utiliza para controlar las tramas repetidas que puedan llegar al receptor.

A continuación podemos ver valores de K según el protocolo:

	K_t	K_r
Stop&Wait	1	1
Repetición selectiva	k	k
Go_back_N	k	1

\rightarrow porque descarta las tramas y solo le importa la última que haya llegado

Eficiencia de una transmisión con ARQ continuo y control de flujo con ventana deslizante y sin errores:

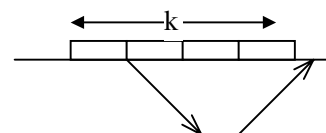


Si ACK llega antes del límite de la ventana no nos bloquearemos nunca

Si $k \cdot T_t < T_c \rightarrow$ hay bloqueo; $U_p = k \cdot T_t / T_c$

si $k \cdot T_t \geq T_c \rightarrow$ No hay bloqueo; $U_p = 1$.

Si hay errores se ha de dividir la eficiencia del protocolo por N_t , en cualquiera de los casos.



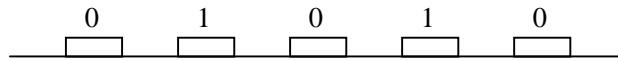
Si ACK llega después del límite de la tramas nos bloquearemos.

NÚMEROS DE SECUENCIA

Los números de secuencia sirven para la identificación de las tramas.

Si tenemos n bits entonces podemos secuencias 2^n tramas. Por lo tanto tenemos 2^n tramas que podemos generar.

· Stop&Wait: solo necesitamos un bit (2 tramas) para las secuencias ya que solo nos hace falta numerar 2 tramas, se ha de poder distinguir las tramas.



· Repetición selectiva: $N \geq 2 \cdot K_t$

· Go_Back_N: $N \geq K_t + 1$ ya que si uno es incorrecto descarta las demás.

La ventana óptima (K_{opt}) es aquella ventana que produce una eficiencia muy cercana a uno:

$$U_p = 1 \text{ si } K_t \cdot T_t \geq T_c \quad K_{opt} \geq \left\lceil \frac{T_c}{T_t} \right\rceil$$

TEMA 4.- REDES LAN

INTRODUCCIÓN

Una **LAN** es un sistema que permite conectar directamente varias estaciones entre si.

Un *sistema* nos implica que va haber componentes hardware y software.

Conectar: queremos que los terminales se transmitan información entre ellos.

Directamente: que no se comunica un terminal con otro terminal con otra red diferente. Para que los terminales se comuniquen directamente se podría:

- Utilizar un cable para comunicar un terminal con todos los demás.
- Utilizar un solo cable (un solo medio de transmisión) y los terminales están conectados a este cable.
- Utilizar un dispositivo que conecte todas las estaciones, el dispositivo puede ser un Hub o Switch.

Al haber varias estaciones deberemos tener un sistema de direcciones que identifiquen las estaciones.

Estas direcciones son: @MAC, @física (id de la estación dentro de una red), @hardware.

La dirección más importante es la de acceso al medio (MAC) que es la que permite llegar a la estación.

Tiene que haber una topología o forma de conexión entre los diferentes terminales.

Las LAN's se comunican con la filosofía *red broadcast* o *por difusión*. En esta filosofía una trama es vista por todas las demás estaciones y se copiará en la estación en que tenga el identificador igual que el identificador destino de la trama.

Las LAN necesitan un algoritmo de acceso al medio, ya que podría pasar que todos los terminales quisieran transmitir al mismo tiempo y de esta forma se produciría una colisión, por lo tanto se ha de controlar el acceso al medio.

ARQUITECTURA DE NIVELES

Se pueden escoger diferentes estándares sobre que arquitectura de niveles utilizar (OSI, TCP/IP, AppleTalk, Novell, DEC, SNA,)

Nosotros mismos nos podemos definir una arquitectura de niveles, lo único importante es que el último nivel pueda discernir que tipo de jerarquía (arquitectura) se utiliza, pero todos los terminales tendrán que seguir esta jerarquía.

En **Ethernet** tenemos dos tipos de tarjetas:

- Ethernet-DIX (Digital, Xerox, Intel)
- y IEEE 8023).

IEEE divide el nivel de enlace en dos en cambio DIX no quería dividir el enlace. Por lo tanto si teníamos tarjetas DIX, estas no podrían comunicarse con las IEEE.

Al final IEEE cambio su jerarquía para compatibilizar las tarjetas. este define tres niveles:

- Nivel **LLC**: define una trama, el código detector de errores y un servicio confirmado o no confirmado (ACK, NACK). Podría permitir un control de flujo.
- Nivel **MAC**: define como se comparte el medio, también define una trama y dentro de la trama define el sistema de direcciones físicas y un código detector de errores no confirmado. El código detector de errores es un CRC. También define el sincronismo de trama (campo que indica la longitud de la trama).
- Nivel **físico**: define el medio de transmisión, los conectores, la topología, la codificación de canal y el diámetro de la red (distancia entre las dos estaciones mas lejanas).

MEDIOS DE TRANSMISIÓN

GUIADOS

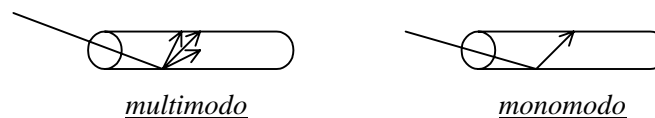
La onda electromagnética va encapsulada, normalmente, en un cable. (LAN cableadas: Ethernet, token Ring, FDDI, DODB, 100VG Any LAN, LAN ATM).

El cable utilizado pueden ser coaxial, par trenzado (UTP) o fibra óptica. Estos cables permiten velocidades altas y las LAN's son normalmente muy rápidas (>100 Mbps).

- **Coaxial**: diámetro de 500 metros a 10 Mbps. Se pueden conseguir 150 Mbps pero con pocos metros. Este cable tiende a desaparecer.
- **UTP**: es un cable estructurado. Ha habido un estándar y por lo tanto está definido el hardware y el software a utilizar. Hay tres tipos:
 - UTP-3 → BW=16Mhz, 100 mt
 - UTP-4 → BW=20Mhz, 100 mt
 - UTP-5 → BW=100Mhz, 100 mt.

El estándar lo definió EIA 568 (Electronic Industries Association), también definió el cableado en un edificio (normalmente se utiliza una topología en estrella), es decir la topología o cableado estructurado.

- **Fibra óptica**: se diferencia entre el *multimodo* y el *monomodo*. El *multimodo*, el diámetro de cable es grande y por lo tanto se producen más de un haz de luz al rebotar la luz con las paredes del cable. Tiene más distorsión que el monomodo y puede producir atenuación. La distancia entre terminales puede ser de 400 metros. El *monomodo* solo tiene un haz de luz y por lo tanto puede ser más rápida y puede llegar a una distancia mayor (2 Km).



NO GUIADOS

La onda electromagnética no va encapsulada (infrarrojos, microondas, ondas de radio, etc.). A estas LAN's se las conoce como **wireless** (sin hilos).

El problema que se nos presenta es que las baterías nos limitan el diámetro de la red. Si nos salimos del diámetro de la red, dependeremos de otra estación base, este cambio de estación crea un clic con el cual podemos perder mucha información.

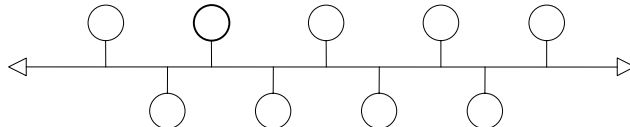
La dirección de Internet no es estática en redes inalámbricas, y por lo tanto no sabemos donde nos encontramos.

TOPOLOGIAS

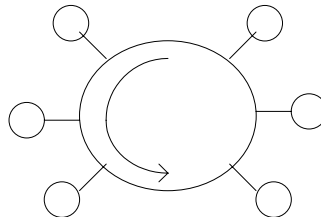
Actualmente todas las redes LAN se implementan en estrella.

Topología física: es la topología que forman las estaciones a nivel físico, estas pueden ser:

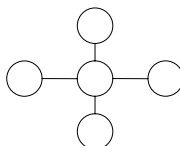
- Bus: todas las estaciones están conectadas a un cable.



- Anillo: todas las estaciones están conectadas de una a otra.

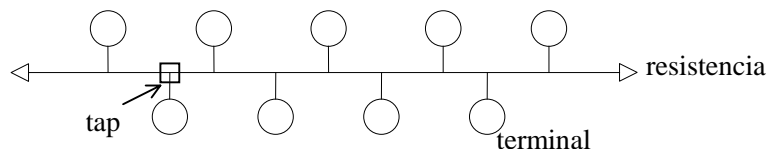


- Estrella: tenemos un centro donde están conectadas todas las estaciones.



Topología lógica: implica como esta funcionando una red realmente, es decir podemos hacer que una topología física en estrella trabaje como bus o como anillo.

BUS



El medio de transmisión termina en dos resistencias, una a cada lado. La conexión de un terminal a un bus se realiza mediante un **tap** (toma de conexión). Esta topología forma una red por difusión.

Cuando un terminal transmite el tap deja pasar la corriente y la distribuye en los dos sentidos. De esta forma todas las estaciones ven el flujo de bits. La información la recoge el terminal que tiene el identificador igual al identificador de destino de la trama.

Las resistencias eliminan la señal, de esta forma la señal no rebota y se puede utilizar de nuevo el cable, sin tener que esperar a que se elimine la señal emitida por si sola.

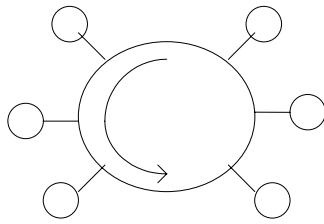
Esta topología en bus también se le llama **segmento**.

ETHERNET	10 base 2	10 base 5
Vt	10 Mbps	10 Mps
long max seg	185 mt 0,2 pulgadas	500 mt 0,4 pulgadas
cable	coaxial	coaxial
diámetro red	1 Km	2,5 Km
#terminales/seg	30	100
dist entre seg	0,5 mt	2,5 mt

multiplos

ANILLO

El medio de transmisión forma un bucle cerrado.



Al igual que la topología en bus los terminales se conectan al anillo a través de tomas de conexión, que en este caso se llaman **repetidores** (regeneran la señal y hay otros que permiten ampliar la red). La dirección de la información solo sigue un sentido ya que si lo hiciera en los dos sentidos, se producirían colisiones. El repetidor tiene un buffer para regenerar los bits y por lo tanto produce un retardo ($\text{retardo} \leq T_{\text{bit}}$). El retardo se puede producir a miles de bits y de esta forma podemos detectar patrones de bits y modificarlos.

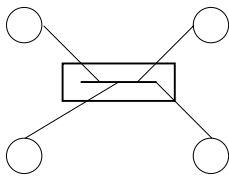
Las tomas de conexión pueden trabajar en tres modos diferentes:

- Modo *escucha*: se regenera la señal que le entra, la copia en el terminal y la deja pasar al siguiente terminal.
- Modo *transmisión*: pone un flujo de bits en el cable y lo envía en un solo sentido y cuando le vuelve puede eliminar dicha información que había emitido.
- Modo *cortocircuito*: la señal se regenera pero no se envía la información al terminal. De esta forma se pueden quitar terminales sin necesidad de tocar la red. En este caso el retardo es 0.

Algunas de las redes que utilizan una topología en anillo son Token Ring y FDDI.

ESTRELLA

En este caso tenemos un centro al cual están conectadas todas las estaciones, el centro es quien toma las decisiones sobre el acceso al medio, de forma que el centro va mirando que estación transmite y quien tiene que recibir dicha información (habiendo una especie de Polling).



Una de las redes que utilizan la topología en estrella es 100VGAnyLAN

Si tenemos un cableado estructurado el centro de la estrella se compone de un bus (**HUB**: Broadcast (difusión)). normalmente el cableado es un par trenzado (UTP) o fibra óptica.

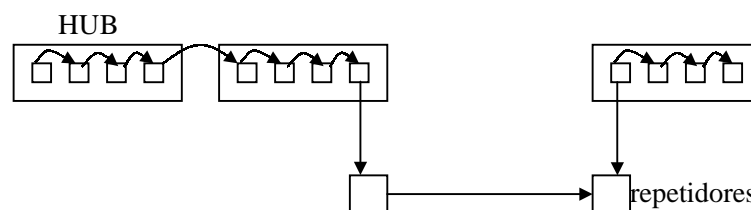
El problema de la estrella es el límite de la distancia de cable (100 metros) y que por tanto el diámetro de la red son 200 metros.

Si un terminal se quiere quitar, se puede quitar sin necesidad de tocar la red, incluso el propio HUB puede desconectar un terminal si su tarjeta de red estuviera dañada (es decir inhibe el puerto).

El HUB también permite seguridad informática (si viene una trama con @MAC que no corresponde a la que permite puede descartarla).

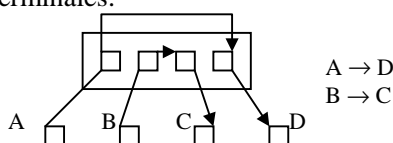
Existen los HUB MAU los cuales tienen repetidores y de esta forma se puede implementar una topología en anillo.

Para extender una red solo necesitaríamos repetidores.

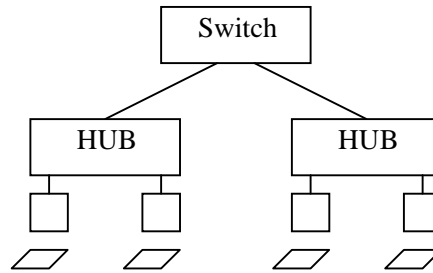


El centro de las redes además de poder ser un HUB pueden ser SWITCHES.

Los **Switches** se utilizan en redes Ethernet. En los HUB una trama se propaga a todas las estaciones, en cambio el switch tiene buffers, y gracias a ellos puede identificar la dirección destino y de esta forma dirigir la información directamente al terminal deseado, sin necesidad de pasar por el resto de las estaciones. Con los switch se aumenta la velocidad de transmisión ya que se pueden transmitir diferentes informaciones entre diferentes terminales.



Los HUB y los switches se pueden conectar entre si, sin necesidad de haber conflictos. Una posible conexión podría ser la siguiente:



A continuación tenemos algunas Ethernet que utilizan la topología en estrella:

- 10BaseT: 10 Mbps, Manchester, UTP
- 10BaseF: 10 Mbps, Manchester, Fibra Óptica.

En Fase Ethernet tenemos las siguientes redes:

- 100BaseF: 100Mbps, Codificación NML, Fibra Óptica.
- 100BaseT: 100Mbps, Codificación NML, UTP-5, FDX.
- 100Base4T: 100Mbps, Codificación NML, UTP-3, HDX.

ACCESO AL MEDIO

El acceso al medio consiste en una política o algoritmo que permite a las estaciones acceder/compartir un medio de transmisión (dispositivo HUB, MAU, switch, ...) de forma ordenada o coherente.

Las políticas que existen actualmente son:

- Paso de testigo en anillo.
- CSMA/CD en Ethernet.

En las LAN el enlace es parte software y parte hardware (parte del subnivel MAC).

En MAC se implementa una trama, detector de errores (CRC) y la política de acceso al medio.

TOKEN RING

El Token Ring tiene una topología en anillo y el acceso al medio se realiza por paso de testigo (Token Passing).

El testigo es una trama especial, el cual permite transmitir solo al terminal que lo tenga en su posesión.

Por lo tanto hay dos tipos de tramas: datos y token.

Cuando un terminal tiene el token, está en modo de transmisión y el resto de las estaciones están en modo escucha o en cortocircuito.

Los parámetros que influyen en un anillo son los siguientes:

→ D: distancia del cable, desde un punto al mismo.

$$T_p = \frac{D}{V_p} \text{ (seg) tiempo de propagación.}$$

→ L: longitud de la trama en bits.

$$\text{Tiempo de transmisión de la trama: } T_t = \frac{L_t}{V_t} \text{ (seg)}$$

→ L_k: longitud del token.

$$\text{Tiempo de transmisión del token: } T_k = \frac{L_k}{V_t} \text{ (seg)}$$

→ Latencia del anillo: tiempo que tarda un bit en recorrer el anillo.

$$\zeta' = T_p + \frac{M \cdot B}{V_t}$$

→ M: número de estaciones

→ B: numero de bits de retardo que introduce cada repetidor.

→ T_{pk} : propagación entre estaciones. Suponemos que las estaciones son equidistantes.

$$T_{pk} = \frac{\zeta'}{M}$$

→ T_{oi} : tiempo de ocupación del anillo. Tiempo desde que una estación transmite el primer bit de la trama hasta que puede transmitir el primer bit de la siguiente estación.

→ U_p : eficiencia del protocolo.

$$U_p = \frac{T_t}{T_{oi}}$$

→ V_{ef} depende de la gente que esta transmitiendo y por lo tanto es variables

$$V_{ef_{min}} \leq V_{ef} \leq V_{ef_{max}}$$

$V_{ef_{min}}$: cuando libera el testigo y todos las demás estaciones quieren transmitir.

$V_{ef_{max}}$: cuando libera el testigo y nadie mas quiere transmitir.

ANILLOS DE BAJA VELOCIDAD ($\approx 4\text{MBPS}$)

Funcionan de dos formas diferentes:

Single packet

Se transmite una trama y se libera el token cuando le llega el último bit de la trama.

$$U_p = \frac{T_t}{T_{oi}} = \frac{T_t}{\zeta' + T_t + T_{pk} + T_k}$$

$$T_{oi_{sp}} = \zeta' + T_t + T_{pk} + T_k$$

para que $U_p \approx 1 \rightarrow \zeta' + T_{pk} + T_k \approx 0$ lo que es imposible por la latencia.

$$V_{ef_{max}} = V_t \cdot E_t \cdot \frac{T_t}{T_{oi_{min}}}; V_{ef_{min}} = V_t \cdot E_t \cdot \frac{T_t}{T_{oi_{max}}}$$

$$T_{oi_{min}} = \zeta' + T_t + \zeta' + T_k; T_{oi_{max}} = M \cdot T_{oi}$$

Los sistemas *no exhaustivos* permiten transmitir una trama y después liberar el token, mientras que los *exhaustivos* permiten vaciar un buffer entero de tramas antes de liberar el token.

Single Token

Libera el token cuando el transmisor recibe el primer bit de la trama. Puede ocurrir que cuando nos llega el primer bit de la trama, todavía no hayamos terminado de enviar la trama y por lo tanto nos tendríamos que esperar a que se terminará de transmitir la trama.:

$T_t \geq \zeta' \rightarrow$ la estación no ha terminado de transmitir la trama cuando llega el primer bit.

$$T_{oi} = \zeta' + (T_t - \zeta') + T_{pk} + T_k = T_t + T_{pk} + T_k$$

$T_t < \zeta' \rightarrow$ la trama ya se ha enviado cuando nos llega el primer bit de esta y por lo tanto podemos transmitir el token.

$$T_{oi} = \zeta' + T_{pk} + T_k$$

$$V_{ef_{min}} = V_t \cdot E_t \cdot \frac{T_t}{T_{oi_{max}}}; T_{oi_{max}} = M \cdot T_{oi} \quad T_t \geq \zeta' \Rightarrow T_{oi_{min}} = T_t + \zeta' + T_k$$

$$V_{ef_{max}} = V_t \cdot E_t \cdot \frac{T_t}{T_{oi_{min}}} \quad T_t < \zeta' \Rightarrow T_{oi_{min}} = \zeta' + \zeta' + T_k$$

ANILLOS DE ALTA VELOCIDAD ($\approx 20\text{MBPS}$)

El modo de funcionamiento de estos anillos de alta velocidad es mediante el **múltiple token**, este modo libera el token tan pronto como acabamos de transmitir la trama.

$$U_{p_{max}} = \frac{T_t}{T_{oi}}$$

$$T_{oi} = T_t + T_{pk} + T_k$$

$$T_{oi_{max}} = M \cdot T_{oi}; T_{oi_{min}} = T_t + \zeta' + T_k$$

ETHERNET

Esta red es la mas abundante en el mundo. Existen dos tipos de tarjetas:

- Ethernet II (DIX)
- IEEE 802.3

DIX VS IEEE

DIX definió los niveles físico y de enlace, y por encima se encontrarían todos los demás protocolos (ApplleTalk, Novell, TCP/IP, OSI...)

pila protocolos
MAC
Fisico

El nivel físico asume que la topología, físicamente hablando, es un bus, pero que lógicamente puede ser un bus o una estrella.

El nivel MAC define el acceso al medio y parte lo hace por hardware y parte por software..

IEEE mantiene el mismo nivel físico, pero diferencia un nivel mas, este nivel es el LLC que se encuentra por encima del nivel MAC. El nivel LLC esta estandarizado mediante el IEEE 802.2. La diferencia con DIX se encontró en el formato de la trama del nivel MAC.

pila protocolos
LLC
MAC
Fisico

Nivel LLC

Define el formato de la trama LLC y el tipo de servicios que ofrece (servicios confirmados y no confirmados). Un tipo confirmado es un protocolo que pide retransmisión, mediante el ACK, los no confirmados transmiten y si hay algún error se descarta.

Ethernet utiliza los servicios no confirmados, solo se utilizaran los servicios confirmados en redes con transmisiones muy erróneas, como pueden ser la redes inalámbricas.

El LLC también ofrece la multiplexación a protocolos de nivel superior, que es en realidad el único servicio que se utiliza del LLC en Ethernet.

El formato de la trama LLC es el siguiente:

DSAP	SSAP	Control	Datos
------	------	---------	-------

DSAP: destino (destination) Service Acces Point

SSAP: fuente (source)

Algunos de los posibles valores de DSAP y SSAP:

- 06h: Arpanet IP
- FEh: OSI
- 80h: Xerox
- Eoh: Novell.

Datos: es la trama del nivel superior.

Nivel MAC

Define el algoritmo de acceso al medio, el formato de la trama de este nivel y el direccionamiento (identificador de las estaciones). El formato de la trama es el siguiente:

Datos				
@MAC destino	@MAC fuente	long	trama LLC	CRC
6 bytes	6 bytes	2 bytes	variable (46- 1500 bytes)	

El formato de las direcciones MAC es el siguiente:

1 bit	1 bit	46 bits
I/G	U/L	OUI OUA

- **I/G** (individual/Group): define si la trama va a ser **unicast** (0, comunicación de una estación a otra) o **multicast** (1, comunicación que va desde una estación a N estaciones). Un caso especial de multicast es el **broadcast** el cual tiene como código todo unos.
- **U/L** (Universal/Local): indica si la dirección esta administrada localmente o universalmente. Todas las tarjetas, por defecto, tiene este bit a 0, este valor indica que la dirección es única. Si es 1 es el administrador quien puede elegir la dirección (¡¡¡¡Mucho cuidado!!!!)
- **OUI** (Organization Unique Identifier): este identificador lo da el IEEE y será común para la empresa que lo ha solicitado, de esta forma no hay dos tarjetas con el mismo identificador, ya que el fabricante va variando el campo OUA.

Ethernet-DIX definió el siguiente formato de trama:

@MAC destino	@MAC origen	type	LLC	CRC
--------------	-------------	------	-----	-----

El campo type identifica el protocolo de nivel superior y no la longitud de la trama LLC. En cambio IEEE define el protocolo superior en el nivel LLC

Con IEEE tenemos que la trama del nivel LLC debe ser e al menos 46 bytes y si la trama no llega a los 46 bytes se utiliza la técnica *padding* la cual rellena de 0 la trama hasta llegar a los 46 bytes mínimos. Entonces el campo long indica la longitud de la trama LLC.

En cambio DIX no definió este campo y definió el campo type.

Al acuerdo que llegaron fue que los tipos no tuvieran un código inferior a 1500.

ACCESO AL MEDIO

Ethernet utiliza como acceso al medio CSMA/CD. Pero antes de llegar a este protocolo se tuvo que pasar por Aloha, CSMA i finalmente a CSMA/CD.

Aloha

La idea es de un protocolo de acceso al medio aleatorio.. En anillo el testigo va pasando de estación a estación, por lo tanto tiene un orden, en cambio en Ethernet si hay una trama a transmitir se transmite y espera un tiempo ($2T_{p(max)} + \Delta t$), este tiempo de espera es para ver si nos llega el ACK (la confirmación de que la trama ha llegado a su destino).

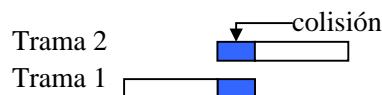
Si recibe el ACK entonces es que ha ido todo bien, en cambio si no se recibe se retransmite la trama pero solo un numero determinado de veces, por lo tanto si en este numero de veces no ha llegado la trama esta se descarta.

El receptor en este caso son todas las estaciones que escuchan el medio. Las estaciones comprueban la @MAC destino y el CRC , si este es correcto y la dirección de destino se corresponde con su dirección, devuelve el ACK y si no es correcto no devuelve nada.

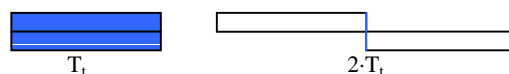
Con este acceso al medio se pueden producir y se producen colisiones (dos o mas estaciones que transmiten al mismo tiempo, es decir, dos señales digitales que se encuentran en el medio). Lo malo seria que hubiera muchas colisiones pero se pueden permitir un número bajo de colisiones.

El tiempo de vulnerabilidad (T_v) es el intervalo de tiempo en que en la LAN se pueden producir colisiones. Este tiempo es:

$$T_v = 2 \cdot T_t$$



La duración de la colisión esta acotado entre T_t y $2 \cdot T_t$. ($T_t \leq T_c \leq 2 \cdot T_t$)



Sea $N(t)$ el número de tramas sin colisiones desde 0 a t , i $C(t)$ el número de tramas que colisionan. Definimos la **Eficiencia** o **Troughput (S)** como el número de tramas que llegan a su destino sin producirse ninguna colisión por unidad de tiempo.

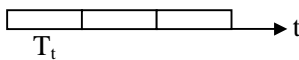
$$S = \lim_{t \rightarrow \infty} \frac{N(t) \cdot T_t}{t} = \lim_{t \rightarrow \infty} \frac{N(t)}{t/T_t}$$

La eficiencia máxima que puede conseguir este método es de 18% ($S=0,18$).

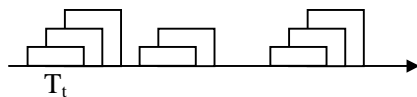
Definimos la **carga ofrecida (G)** como el número de tramas que se ofrecen al sistema, ya sean sin colisiones como con colisiones.

$$G = \lim_{t \rightarrow \infty} \frac{(N(t) + C(t)) \cdot T_t}{t} = \lim_{t \rightarrow \infty} \frac{N(t) + C(t)}{t/T_t}$$

El caso ideal sería en que tenemos una eficiencia de 1 ($S = 1$), este caso se produciría si hubiera una trama en cada instante de trama, es decir siempre se estaría ocupando el sistema. Pero este caso es muy difícil que se produzca ya que siempre habrá colisiones.



El caso real es que obtengamos una $S < 1$ y que la G este entre 0 i ∞ , es decir que por cada instante de trama transmitimos una trama pero en medio puede llegar otra trama. Por lo tanto, en un tiempo de trama puede haber de una trama. Al ser este el caso real podemos deducir que el sistema será inestable.



Para conseguir que el sistema sea mas estable se debería de utilizar un algoritmo de **back off**, el cual no encola la trama que haya producido una colisión sino que le indica que se vuelva a transmitir en un tiempo aleatorio mas tarde.

El **Aloha ranurado** obliga a todos los terminales a transmitir dentro de cada ranura, de esta forma se duplica la eficiencia pero tiene los mismos problemas que el Aloha puro.

Las estaciones transmiten cuando quieren, una posible mejora sería que las estaciones miren el medio antes de transmitir, este método lo utiliza el CSMA.

CSMA (Carry Sense Multiple Acces)

Este sistema observa si alguna estación esta transmitiendo antes de transmitir, de esta forma el sistema pasa de ser aleatorio a competencia-reserva (durante un tiempo miramos si podemos utilizar el medio y si podemos cogemos el medio y a partir de ese momento el medio es solo de la estación hasta que termina de transmitir). La observación es física, electrónica, es decir que se realiza sobre el cable.

Para *transmitirse* observa si el medio esta ocupado, si no lo esta se transmite y si está ocupado tenemos dos opciones:

- **1_persistente**: sigue mirando el medio hasta que se libera el medio. Observa hasta que el sistema esté desocupado (espera un cierto tiempo fijo – IPG) y transmite
- **No_persistente**: este método va observando el medio cada un cierto tiempo aleatorio.

Se pueden producir igualmente colisiones, ya que si dos estaciones están observando y el medio se libera estas dos estaciones transmitirán a la vez.

Este método llega a una eficiencia del 0,5 o mayor.

El 1_persistente puede degradarse si aumenta la G , ya que habrá mas gente esperando a transmitir, este sistema al igual que Aloha, se le puede aplicar un algoritmo de back off para que cuando halla mucha carga se mantenga la máxima eficiencia.

$$T_v = 2 \cdot T_{p(max)}$$

El primer bit de una trama tarda un tiempo de propagación para pasar de estación a estación, por lo tato se puede producir colisiones si se transmite en un tiempo $\pm T_p$. Este T_p será el tiempo de propagación entre las estaciones mas lejanas (diámetro de la red).

$$T_t \leq T_{col} \leq T_t + T_p$$

Para mejorar la eficiencia podríamos reducir el tiempo de colisiones y para esto una vez que transmitimos también escuchamos y si hay una colisión se para de transmitir. Este sistema lo realiza el CSMA/CD.

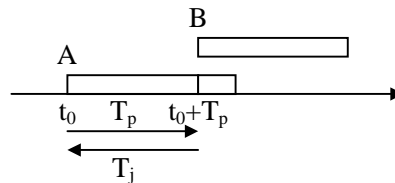
CSMA/CD (Carrier Sense Multiple Acces with Collision Detection)

Carrier Sense = Escucha portadora.

Es lo mismo que CSMA pero una vez que se ha transmitido se escucha. Para esto el cable es en realidad 4 pares trenzados (uno transmite, uno escucha, ...). Cuando uno detecta que hay una colisión este envía una interferencia (**jamming signal**) y cada estación que lo recibe para de transmitir, ejecuta un algoritmo de back off y después vuelve a inicializar el back off.

$$\text{jamming signal} \rightarrow T_j = \frac{J(\text{bits})}{V_t}$$

$$T_v = 2 \cdot T_{p(\max)}$$

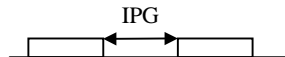


La primera estación que se entra del jamming es B la cual envía el jamming, y este tiene que llegar a A (T_p mas tarde).

$$T_{p(\max)} + T_j \leq T_{col} \leq 2 \cdot T_{p(\max)} + T_j$$

CARACTERÍSTICAS DEL MAC ETHERNET

- CSMA/CD 1_Persistente
- $V_t = 10 \text{ Mbps} \rightarrow T_b = 100 \text{ ns}$
- IPG = $9,6 \mu\text{s}$ (96 bits) (Inter Packet Group: tiempo fijo que pasa desde una trama a otra).



- $T_v = \text{ventana de colisiones} = 512 \text{ bits}$
- Tamaño de trama $\rightarrow 64 \leq L_t \leq 1518 \text{ bytes}$
- Límite de intentos de RTx = 16
- Límite de Back_off = $10 = n \rightarrow \text{random}(0..2^n) \cdot 2 \cdot T_{p(\max)}$
- Jamming de 32 bits.
- Parámetros de diseño:

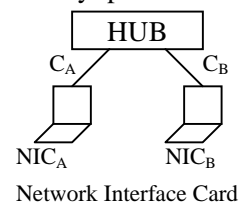
$$T_v = 2 \cdot T_{p(\max)} = 512 \cdot T_b$$

Sirve para fijar el tamaño mínimo de la trama ($T_{t(\min)} \geq T_v$: si $T_{t(\min)} < T_v$ entonces hay una estación que no detecta la colisión y da por transmitida dicha trama y por lo tanto no retransmite la trama errónea) y el diámetro máxima de la red.

$$\Sigma \text{retardos} \leq T_v = 512 \cdot T_b = 2 \cdot T_p$$

$$2 \cdot (\text{NIC}_A + C_A + T_{\text{hub}} + C_B + \text{NIC}_B) \leq 512 \cdot T_b$$

Donde $C_A = \text{Dist}(A, \text{hub}) / V_p$.



SEGMENTACIÓN LAN

La segmentación se utiliza para alargar la red y consiste en ir poniendo repetidores a la distancia máxima admisible hasta llegar a la longitud deseada.



Tenemos dos segmentos pero un solo dominio de colisiones (cualquier lugar donde puede haber colisiones). Si en un segmento hay colisiones estas se propagarán por los demás segmentos.

Para extender la red tenemos los siguientes elementos:

- Repetidores
- Bridge/switch

- *Routes*

El **dominio de colisiones** son todos aquellos segmentos que ven las colisiones producidas por una estación. En este caso tenemos que un segmento es de por sí un dominio de colisiones.

Si conectamos un HUB, este retransmite por todos los puertos de salida, excepto por el que transmite, por lo tanto todas las estaciones conectadas a este HUB forman un dominio de colisiones. Por lo tanto tenemos N segmentos pero un solo dominio de colisiones.

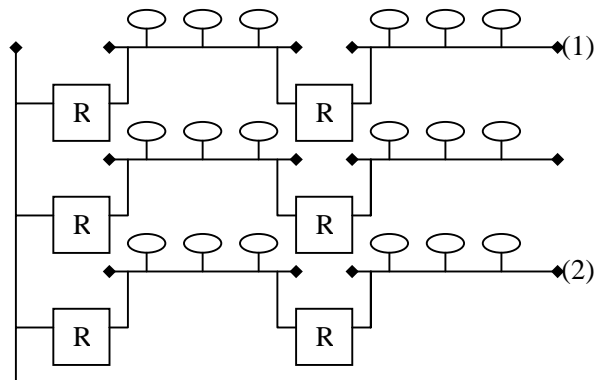
Si conectamos un Switch (una estación transmite y el Switch mira la dirección MAC y se lo retransmite solamente al destinatario, esto es gracias a que estos dispositivos tienen buffers), tendremos N segmentos y N dominios de colisiones en un Half Duplex pero no en Full Duplex (donde tenemos un cable para transmitir y otro cable para recibir).

Un **repetidor** aumenta los segmentos (alarga la red) pero deja igual el número de dominios de colisiones. Un repetidor regenera la señal.

El número de repetidores que se pueden poner para alargar la red no es infinito y está acotado por la siguiente fórmula:

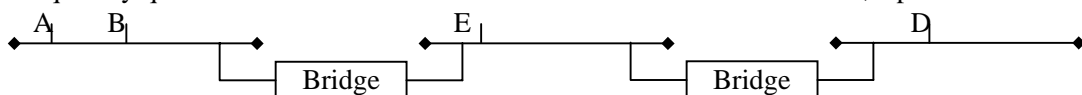
$$T_v = 2 \cdot T_{p(max)} \leq 512 \cdot T_b$$

Para saber el número de repetidores que se pueden usar hay que seguir la **regla 5 – 4 – 3**. Esta regla nos indica que podemos poner 5 segmentos, 4 repetidores y que solo haya 3 segmentos poblados para conseguir la máxima distancia de la red.



La regla no se cumple ya que tenemos más de tres segmentos poblados, pero sí en el 5 y en el 4 (mirar como se va de 1 a 2).

Sigue habiendo un solo dominio de colisiones aunque haya muchos segmentos. Si hay muchas colisiones lo que hay que hacer es crear varios dominios de colisiones con los switch, a partir de la red inicial.



Un **Bridge** trabaja a nivel MAC y físico. Si A transmite a B, A libera la trama y B la coge, al llegar al bridge este ve que el destino B está en el mismo segmento que A, y lo filtra, ya no lo retransmite a los otros segmentos.

Si fuera de A a E pasaría por el primer bridge pero no por el segundo. De A a D, la información llega al primer bridge y como no sabe donde está D, este retransmite por todos los puertos de salida.

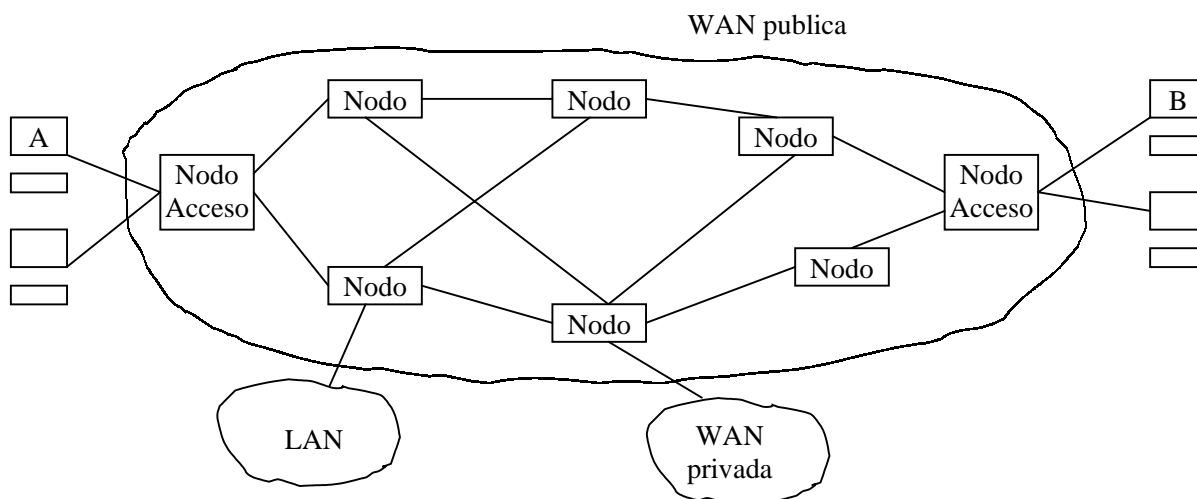
Hay tres dominios de colisiones y tres segmentos.

Un bridge es lento, en una CPU con memoria tiene que hacer cambios de contexto etc., un switch en cambio hace lo mismo pero todo está integrado y por lo tanto es más rápido y eficiente. Con los Switch tenemos un dominio por cada puerto del switch.

TEMA 5.- REDES WAN

Las WAN se componen de diferentes **nodos**, estos nodos son equipos que interconexionan estaciones o sirven de intermediario para la comunicación entre los diferentes terminales conectados a la WAN.

Los **terminales** se conectan al **nodo de acceso**, los cuales se conectaran al resto de nodos. A los nodos se pueden conectar otras WAN o LAN.



SERVICIOS

- Los nodos ofrecen multiplexación (por ejemplo multiplexación TDM).
- La conexión entre los nodos tiene que ser de alta velocidad.
- Hay que hacer conmutación.

Los equipos tendrán N líneas de entrada y M de salida donde cada salida esta multiplexada, por lo tanto hay que asignar una salida a una entrada, y esta asignación es la **conmutación** (que cada canal sea capaz de salir por una salida).

• Hay varios caminos posibles para llegar a un destino, es decir, que se puede llegar al destino por diversos caminos. De estos caminos hay que escoger uno (es decir se ha de ejecutar un algoritmo de **encaminamiento**).

Algunas redes WAN se pueden encontrar en el mercado son:

- RDSI
- ATM
- X25
- FrameRelay
- RTC
- Internet: es una red global y puede tener cualquier tipo de conectores, es decir, no es una WAN homogénea como las anteriores).

Los servicios que se ofrecen en las WAN son de dos tipos:

- Orientados a la conexión.
- No orientados a la conexión.

ORIENTADOS A LA CONEXIÓN

Ofrece una conexión física o lógica entre dos puntos terminales. La conexión física o lógica implica una reserva de recursos (ancho de banda, velocidad de transmisión, buffers, etc.).

Reserva física

Implica que los recursos son dedicados a esa conexión. Implica que los recursos son nuestros y de nadie mas, y por lo tanto sino los utilizamos nadie los utilizará.

Reserva lógica

Los recursos son compartidos. Solo utilizaremos los recursos cuando los necesitemos y solo en ese momento los recursos son nuestros. Con esta reserva se ha de considerar la calidad del servicio y debido a esto son difíciles de gestionar ya que permite muchas comunicaciones.

Hay tres fases en la comunicación:

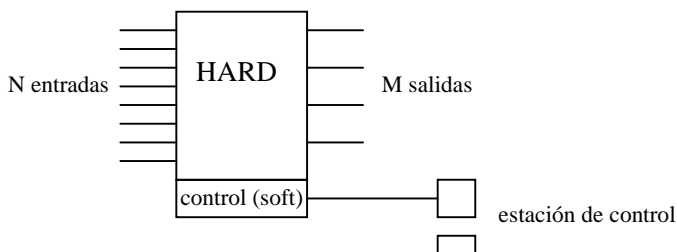
- **Fase de establecimiento de la conexión:** se reserva recursos (física/lógica) y decide el encaminamiento. Aquí la red acepta o rechaza la conexión.
- **Fase de transferencia de la información.**
- **Fase de cierre de la conexión:** libera los recursos reservados.

NO ORIENTADOS A LA CONEXIÓN

En estos servicios se puede transmitir cuando se desee, o puedas, y la WAN hace el mayor esfuerzo para realizar la conexión pero no se asegura que llegue la información. Como no hay una reserva de recursos no sabremos si la red estará congestionada o no, y por lo tanto que llegue la información a su destino.

En este servicio solo existe la fase de transferencia de la información ya que no hay reserva de recursos ni calidad de servicio.

CONMUTACIÓN



Un **nodo** sería un switch/conmutador, es decir, proporciona un camino de salida a las entradas, pero decir por donde sale es trabajo del *encaminamiento*. La **conmutación** consiste en como hacer que una entrada salga por una de las salidas posibles.

El conmutador tiene una estación de control que sirve para configurar dicho conmutador. El conmutador permite multiplexar y por lo tanto tiene que ofrecer velocidades muy altas.

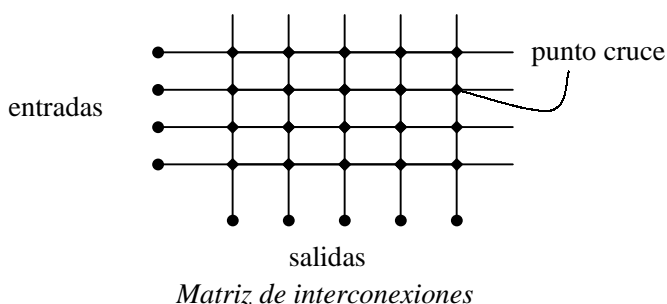
CONMUTACIÓN DE CIRCUITOS

Esta conmutación ofrece servicio orientado a la conexión (establece la conexión, transmite, y finaliza la conexión) con una reserva física de recursos, es decir, es dedicada a cada conexión.

Esta conmutación te asigna un canal dentro del TDM síncrono y este canal será solo tuyo hasta que se desconecte, es decir, tenemos un canal dedicado.

Existen tres tipos de conmutación de circuitos.

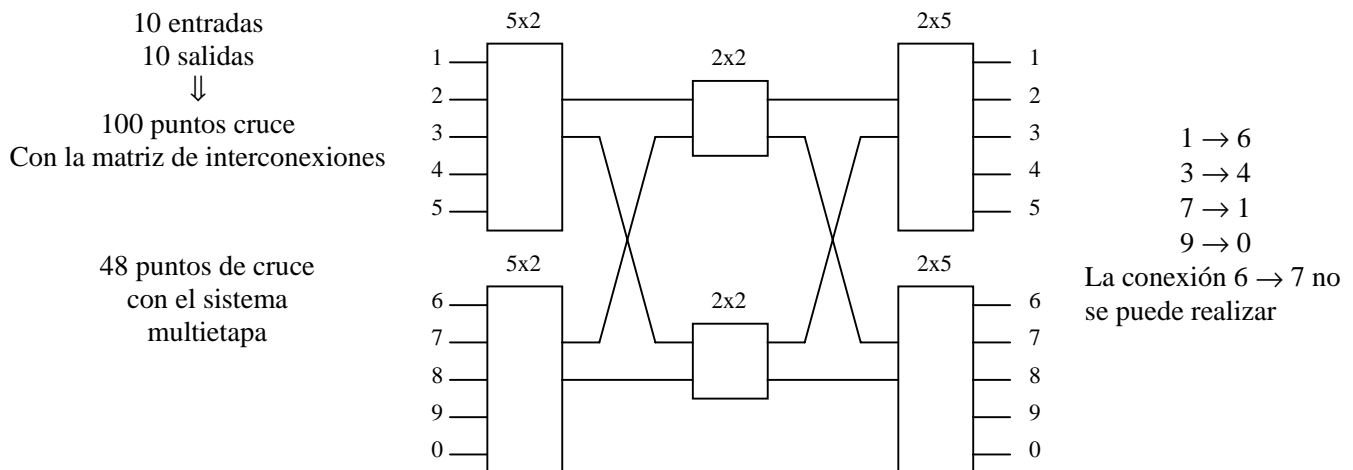
Conmutación espacial



El conmutador tiene N entradas y M salidas, para hacer que un canal de entrada se comunique con uno de salida utiliza una matriz de interconexiones con puntos de cruce. Un **punto de cruce** es una puerta semiconductoras inhibida/habilitada por el control.

El problema es que necesitamos NxM puntos de cruce y estos son caros, y si uno se estropea puede estar inhibiendo otros posibles caminos de salida. Por esta razón las matrices

normalmente son pequeñas. Al ser pequeñas tendremos pocas entradas y pocas salidas, por lo tanto para tener mas entradas y salidas se utilizan **sistemas multietapa** (varias matrices comunicadas entre si).



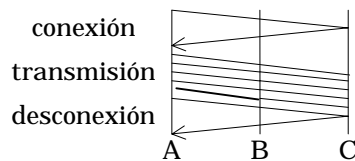
Este sistema es **bloqueante**, es decir, tenemos una entrada y una salida libre pero no podemos utilizarlas porque todos los caminos están ocupados.

Conmutación temporal

El conmutador tiene un bus TDM, internamente, de alta velocidad, en lugar de la matriz. por lo tanto se van asignando entradas a un canal y este canal tendrá su salida. Para que no sea bloqueante la velocidad de transmisión del bus tienen que ser mayor que las velocidades de transmisión de los canales. Por lo tanto será bloqueante si: $v_t < v_{ti}$

A veces se utiliza conmutación mixta entre la espacial y la temporal

De la conmutación de circuitos interesa el tiempo de transferencia de la información (tiempo de enviar la información de un terminal a otro).



En conmutación de circuitos no hay almacenamiento de la información, por lo tanto cuando llega un bit a un nodo este sale por la salida que tenga asociada dicha entrada.

Sea:

E: # de enlaces.

T_p : tiempo de propagación.

$$T_{\text{transf}} = T_{\text{con}} + T_{\text{desc}} + T_{\text{trans}}$$

$$T_{\text{trans}} = E \cdot T_p + T_t$$

Algunas de las redes que utilizan conmutación de circuitos son RTC y RDSI.

CONMUTACIÓN DE PAQUETES POR CIRCUITOS VIRTUALES

Esta conmutación ofrece un servicio orientado a conexión con una reserva lógica de recursos, por lo tanto se reservan buffers en los cuales se copia la información a transmitir.

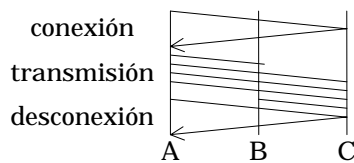
En este caso puede ser que los recursos estén agotados (buffers llenos) y que se pierda la información. Debido a esta posible pérdida de información, se hacen contratos que garantizan una calidad de servicio (QoS – Quality of Service) donde se indica la probabilidad de perder información, los retardos, variación de retardo, etc.

Algunas redes con conmutación de paquetes por circuitos virtuales son X25, FrameRelay, y ATM.

En esta conmutación en lugar de matrices, tenemos líneas de entrada y salida y buffers por cada entrada y salida. El conmutador puede estar formado por:

- Solo buffers de entrada.
- Solo buffers de salida.
- Memoria central compartida.
- Híbridos de los tres.

mas los puntos de cruce, multietapa y el TDM asincrono (hay canales y cada canal es la duración de un paquete).



Suponemos que no hay retardos en los buffers.

$$T_{\text{transf}} = T_{\text{con}} + T_{\text{desc}} + T_{\text{trans}}$$

$$T_{\text{trans}} = E \cdot T_p + n \cdot T_t + (E-1) \cdot T_t$$

E: # enlaces

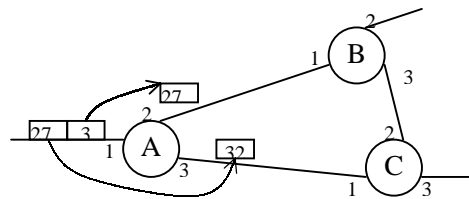
n: # nodos

Un caso particular es cuando $n=1$, en este caso el tiempo de

transito es el de un paquete.

Si suponemos que hay retardos: $T_{\text{transf}} = T_{\text{con}} + T_{\text{desc}} + E \cdot T_p + n \cdot T_t + (E-1) \cdot (T_t + T_{\text{retardo medio buffers}})$

En la conmutación de paquetes por circuito virtual tenemos el **CVI** que es el identificador de circuito virtual. Este identificador lo lleva cada paquete por conexión, y el circuito virtual se asigna a un puerto.



Nodo A				Nodo B			
Puerto In	CVIin	Puerto Out	CVIout	Puerto In	CVIin	Puerto Out	CVIout
1	3	2	27	1	27	2	14
1	27	3	32	3	12	1	22
2	14	1	23				

Si tenemos 8 bits para el CVI, el puerto de entrada (puerto IN) y el de salida (puerto OUT) puede ser de 256 CVI diferentes. Lo que sale por un puerto tiene que tener el mismo CVI que el puerto al que está conectado el puerto.

CONMUTACIÓN DE PAQUETES POR DATAGRAMA

Esta conmutación ofrece un servicio no orientado a la conexión (Best Effort) y por lo tanto no hay reserva de recursos ni calidad de servicio (QoS), es decir, sería Internet. Esta conmutación ofrece buffers y los **datagramas** (unidades de transferencia) llevan la dirección origen y la dirección destino.

Por encima de estas redes se necesita un protocolo de control de errores, flujo, etc.

El camino que sigue un datagrama no es seguro, y puede ser que uno siga un determinado camino y otro datagrama de la misma transmisión vaya por otro camino. Por esta razón el tiempo de transferencia de un datagrama se calcula según el tiempo de retardo medio más el tiempo del datagrama.

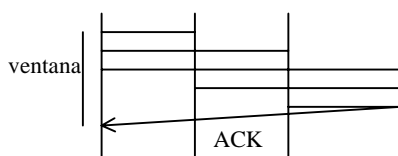
CONTROL DE FLUJO Y DE LA CONGESTIÓN

En redes de paquetes por circuito virtual.

los buffers de los nodos se pueden llenar y entonces descartaría la información. Por lo tanto hay que hacer un control de flujo por ventana deslizante y un control de errores.

El control de flujo se puede hacer entre nodos, entre los extremos, de 2 en 2 nodos, etc.

Extremo a extremo



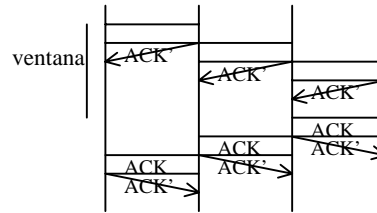
Si hay congestión (los buffers de los nodos) el ACK llegará más tarde y por lo tanto el transmisor se bloqueará.

$$K_{\text{opt}} = \left\lceil \frac{T_c}{T_t} \right\rceil ; T_c = T_{\text{transito 1 paquete}} + T_{\text{transito ACK}}$$

Entre nodos

ACK' → confirmación entre nodos.

ACK → confirmación entre terminales.



Actualmente las redes no hacen control de flujo, ni de errores, solo lo hacen los terminales.

El **Throughput** nos indica el máximo flujo, que sería el número de paquetes por segundo. Este valor depende del control de flujo.

Si no hay control de flujo: $Throughput = \frac{1}{T_{paquete}}$

Con control de flujo, depende de la ventana: $Throughput = \frac{K}{T_c}$

El contrato de tráfico existe para poder evitar las congestiones y se especifica la $v_{t(max)}$, $v_{t(media)}$, calidad de servicio (pérdidas, retardos entre extremos, retado por varianza, etc.) La red calcula si se puede asegurar este contrato.

ENCAMINAMIENTO

El conmutador no solo le asigna una salida a una entrada, sino que selecciona la salida más adérente para llegar a su destino (es decir, mira por donde sacar la información y decide de esta manera el camino). EL **encaminamiento** busca el camino óptimo. Estos algoritmos tienen que ser:

- Robusto: que encuentre el camino.
- Fiable: escoger el mejor camino.
- Justo: equilibrar las rutas, es decir, no asignar el mejor camino a una conexión y dejar las otras conexiones con el peor camino.

La **métrica** es el criterio de decisión. Esta puede ser según:

- **velocidad de transmisión**, la velocidad más alta.
- **hops**: número de saltos o número de nodos que atravesamos hasta llegar al destino.
- **retardo**: escoge el camino que tarda menos desde el origen al destino.
- **ocupación de buffers**.
- etc.

A la métrica se le aplica un algoritmo de mínimo coste como podría ser un Dijkstra o el Bell Man Ford.

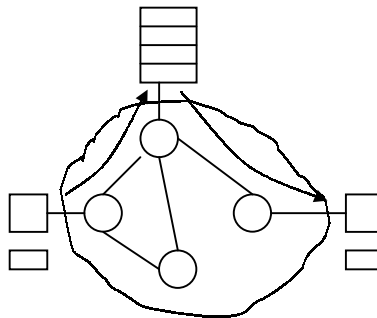
Una vez escogida la métrica y el algoritmo hay que pensar como se realiza la comunicación y cuando se busca el camino:

- Por **circuito virtual**: cuando establecemos la comunicación, en la fase de conexión es cuando se busca la ruta. una vez tomada una decisión (concretar el camino) este será el camino para el resto de paquetes.
- Por **datagrama**: se toma la decisión por cada datagrama, es decir, para cada paquete que se transmite se busca el camino.

La busca del camino se toma en:

- **Origen**, es decir, es el nodo origen quien toma la decisión del camino (**estático**). Nos comunicamos directamente con un nodo determinado y lo que haya entre medio no nos importa.

- **Centralizado**, implica que la red tiene un servidor al que se le pregunta por rutas. Los servidores también envían información que queremos transmitir mientras se busca el camino.



- **Distribuido**, cada nodo toma sus propias decisiones y se utiliza por datagramas en Internet. Cada Router o equipo intermedio toma sus propias decisiones independientemente de las conexiones.

La información para las métricas pueden provenir de:

- **Ninguna información**: se toman decisiones sin tener en cuenta las métricas (**flooding**).
- **Información local al nodo**: solo toma decisiones en función de la velocidad de transmisión y buffers del propio nodo.
- **Información de los nodos adyacentes**.

Estos dos últimos para sistemas distribuidos.

- **Información de toda la red**: sabemos todas las métricas de todos los nodos de la red. Esta forma es ideal para sistemas centralizados.

TÉCNICAS DE ENCAMINAMIENTO

- **Encaminamiento estático**: implica que al principio, al establecer/crear la red se decide el encaminamiento, es decir, la ruta de salida. Esta ruta no cambia a no ser que la cambie el administrador. Esta técnica no reacciona a la congestión ya que siempre se seguirá la misma ruta. Es muy eficiente si tenemos unos cuantos terminales y un solo router. Si tenemos más de un router tenemos un problema de diseño y sería mejor utilizar una técnica dinámica.
- **Flooding**: no tiene en cuenta ninguna información de la red (métrica) al igual que el anterior. Lo que hace es que cuando llega un paquete este sale por todas las salidas excepto por donde ha entrado y esto ocurre en todos los nodos. Al final llegará al destino pero tiene problemas de:
 - Congestión.
 - Alta carga en la red.
 - El destino recibe múltiples copias.
 - Los paquetes pueden estar viajando infinitamente por la red sin llegar a su destino.

Para evitar que lleguen múltiples copias al destino se asignan identificadores para cada conexión (número de secuencias, circuitos virtuales).

Para evitar la congestión y la carga de la red, los paquetes salen por todas las salidas por donde no ha pasado todavía, si ya había pasado no se envía por esa salida.

Para evitar que los paquetes viajen por la red infinitamente se utiliza la técnica TTL (**Time To Live**) la cual limita el tiempo que puede estar ese paquete en la red.

- **Centralizado**: el nodo central obtiene la información de todos los nodos de la red y este calcula el mejor camino, mediante un algoritmo el cual llena una tabla y por lo tanto para ir a un destino primero hay que mirar la tabla que es quien nos indica el mejor camino.
- **Distribuido**: hace lo mismo que hace el nodo centralizado pero ahora para cada nodo de la red y no solo en el router o nodo central.

Ejemplo en las transparencias.

TEMA 6.- INTERCONEXIÓN DE REDES

Todas las redes tienen un nivel orientado a la red. Los nodos intermedios solo tienen implementados los niveles físico, enlace y red:

- Red ATM: Físico, enlace (ATM).
- Red X25: Físico, enlace, red.
- Red Ethernet: Físico, enlace (MAC+LLC o MAC)

En nivel IP conecta las diferentes redes, es decir, los nodos intermedios implementa hasta el nivel IP, estos nodos son los Routers.

El **Router** adapta la información a las diferentes redes a las que este conectado.

La dirección IP identifica redes.

Ningún Router hace control de errores, ni control de flujo.

Hay diferentes jerarquías de encaminamiento, es decir, habría tablas de encaminamiento en las WAN y otra tablas de encaminamiento entre routers.