

APUNTES DE REDES

1997-1998

REDES:

Niveles OSI (estándar ISO)

Descripción de cada nivel.

Definición del interfaz entre niveles.

Protocolos y interfaz actuales.

2 exámenes

+

1 trabajo que equivale a un parcial (en principio para 2º parcial)

BIBLIOGRAFICA:

Redes de ordenadores

A. Tanenbaum

Ed. Prentice Hall

Data and computer communications.

W. Stallings.

Ed. Mc. Millan Publishing Company.

Redes de telecomunicaciones

M. Schuartz

Ed. Addison Wesley

Open Networking with OSI

A. Tang S. Scoggins

Ed. Prentice Hall

OSI a model for computer communications Standar.

U. Black

Ed. Prentice Hall

Internetworking with TCP/IP Vol I,II,III

Comer, Stevens

Ed. Prentice Hall

Local & Metropolitan Area Networks

W. Stallings

Ed. Prentice Hall

Protocolos de comunicacion para sistemas abiertos

Jose Miguel Alonso

Ed. Addison-Wesley Iberoamericana

¿Que se pretende con una red?

Se pretende compartir información entre 2 máquinas, esta información ha de ser digital.



Normalmente la transmisión de información será vía serie.
Los medios de información serán diversos:

Cables, radio, luz, etc...

El modelo de referencia OSI (Open system Interconexion):

Son una serie de normativas a la hora de realizar una red.

Modelo realizado a medida:

Probablemente mas eficiente que el del sistema OSI
Para cualquier mejora tenemos que modificar todo el software y el hardware diseñado.
Son estructuras de tipo propietario.

Modelo de referencia OSI

Se establecen una serie de niveles de forma que al ajustarnos a dicho modelo, nos permitirá modificar cualquier nivel de la red, sin que se vean afectados los demás niveles.

Un nivel básicamente lo que hace es compartir información con sus niveles contiguos.
Nosotros definimos las primitivas de servicio (información entre niveles); por lo tanto nosotros definimos el formato de dicha primitivas.

Nosotros podemos definir una primitiva que le diga al siguiente nivel que tiene que leer un dato.

Si nosotros mantenemos el lenguaje de comunicación entre niveles, a nosotros nos da igual como este realizado el nivel, siempre y cuando sea compatible con el formato de primitivas.

Un ejemplo:

Nosotros tenemos un nivel de seguridad, y este se queda obsoleto, podemos cambiarlo por otro siempre que maneje las mismas primitivas.

Modelo de referencia OSI:

Este modelo establece 7 niveles:

Capa Aplicación
Capa Presentación
Capa Sesión
Capa Transporte
Capa Red
Capa Enlace
Capa Física

Capa Física:

Es la encargada de la transmisión de información a través de un medio físico.

Transforma la señal digital a otra señal que se transmita de la forma más eficiente a través del medio utilizado.

Capa Enlace:

Lo que hace es ver quien hace uso del medio de transmisión en cada momento. También implementaremos un sistema de control de errores y de flujo.

Uso del medio

Se controla quien hace uso del medio en cada momento.

Control de flujo

Control de errores -> Soluciona los errores producidos en la capa Física.

Capa Red:

Se ocupa de realizar el direccionamiento y del control de congestión.

El encaminamiento dentro de una red es en la capa de enlace.

Esta capa se utiliza para el direccionamiento de maquinas que están en otra red.

Se estudiara el protocolo IPv4 y IPv6

Capa Transporte:

Se encarga de la administración de la conexión.

Se encarga de gestionar la transmisión entre 2 máquinas,

Se suele utilizar con máquinas multiprocesador.

Intercomunicación de proceso de distintas máquinas.

Se encarga del protocolo TCP

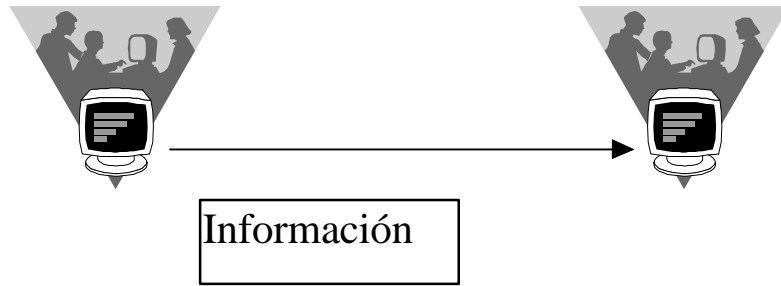
Capa Sesión:

Funcionales para la administración del diálogo.

Controla el flujo de comunicación de procesos también nos dará la forma de recuperar el intercambio de información a partir de un determinado momento si se interrumpe la comunicación entre los sistemas.

Si la información es pequeña se vuelve a enviar entera, ya que no ocupa demasiados recursos a la red.

Pero si la información es muy grande, lo que se hace es subdividir la información en bloques mas pequeños de información; de esta forma lo que se consigue es que es caso de algún corte en la transmisión, no habría que enviar todo la información, sino desde el ultimo bloque enviado correctamente; es decir desde la ultima marca.



Capa Presentación:

Se encarga de la traducción, encriptación y compresión.
Esto permite la conexión entre máquinas de distinto tipo.
Es decir se establecen las normas del formato común de los datos.
Con la encriptación se consigue la confidencialidad de la información.
Las técnicas de compresión nos permiten reducir los bits a transmitir sin perder información.

Capa Aplicación:

Nos permitirá crear aplicaciones que utilicen los recursos de las redes.

Capa Física:

Se encarga de transmitir secuencias de bits entre distintas máquinas a través de un medio de transmisión, normalmente serie; para ello primero habrá que realizar una conversión de la información a un formato recomendado para el medio de transmisión.

Los medios de transmisión más comunes son:

Medio magnético: Constituye un medio en el que se mueve la información es un soporte magnético.
Tiene un ancho de banda muy alto.

Ancho de banda: Viene dada por el número de bits por segundo.

Tiempo de atención: Lo definimos como el tiempo que transcurre desde que se emite la información hasta que la recibe el receptor.

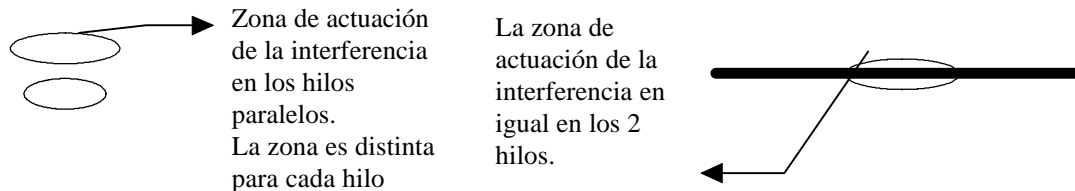
Lo idóneo es un ancho de banda alto y un tiempo de atención bajo.

Par Trenzado: Consiste en 2 hilos de cobre trenzados entre sí.



El trenzado es para evitar lo siguiente:

Si una interferencia puntual actúa sobre 2 hilos paralelos, la interferencia no es la misma en los 2 hilos.



Como normalmente las mangueras no solo van 2 cables, lo que se hace es trenzar cada par de hilos con una relación distinta, para después trenzarlos todos juntos.

Este sistema es barato y fácil de instalar.

Existen varios tipos de *Pares Trenzados*.

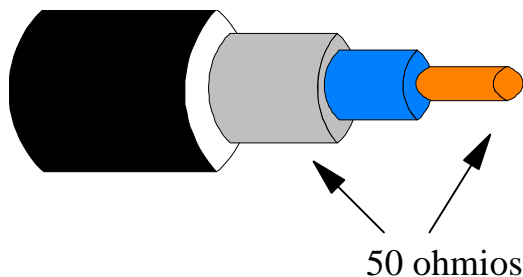
- UTP (Unshield Twister Pair)
Par trenzado sin apantallar.
- STP (Shield Twister Pair)
Par trenzado apantallado, la pantalla esta conectada a masa

Cable apantallado: Cable con un recubrimiento metálico.

El STP protege mas a las interferencias, es mas caro y necesita un buena tierra para el apantallado. Si esta tierra es mala, es mejor el UTP que el STP.

El STP fue ideado por IBM
El ancho de banda es de 100 MHz

Cable Coaxial:



Está compuesto por un conductor con su aislante, los cuales están recubiertos por una malla.

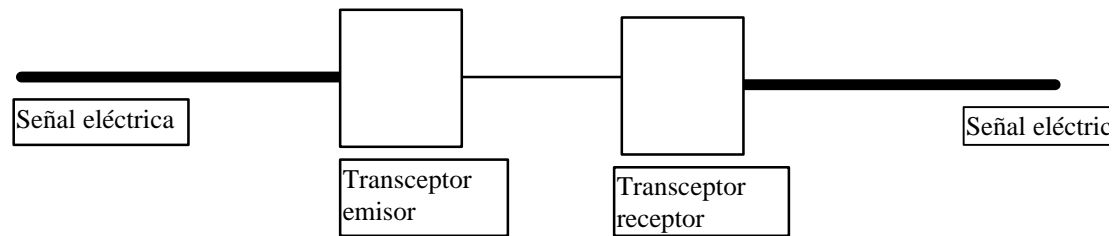
Normalmente los cables coaxiales presentan una resistencia de 50 Ohm o incluso de 75 Ohm (resistencia a la frecuencia de trabajo.).

Este cable se utiliza tanto para señales digitales como para señales analógicas. Normalmente se utiliza el de 50 Ohm con señales digitales y el de 75 Ohm para señales analógicas.

El ancho de banda es normalmente de unos 300 MHz en señales digitales, y algo menos en señales analógicas.

Fibra Óptica:

La transmisión de información a través de fibra óptica se realiza de la siguiente forma:



Normalmente los transceptores emisores son diodos LED o LÁSER; mientras que los transceptores son Fotodiodos o Fototransistores.

Existen 2 tipos de fibras ópticas:

Multimodo:

La señal va rebotando por las paredes de la fibra con unas pérdidas despreciables; esto se debe a que la fibra está hecha por un núcleo de vidrio y forrada por una superficie con un alto nivel de refracción; esto permite transmitir varias señales a la vez según sea su ángulo de ataque en los rebotes

Monomodo:

La señal se difunde por el núcleo de la fibra, con atenuación y anchos de banda mayor que el anterior, por ejemplo para grandes distancias se utiliza el LÁSER.

Las velocidades de transmisión son del orden de 1G/seg. en distancias de 1 Km. sin pérdidas de señal.

Las interferencias electromagnéticas no les afectan.

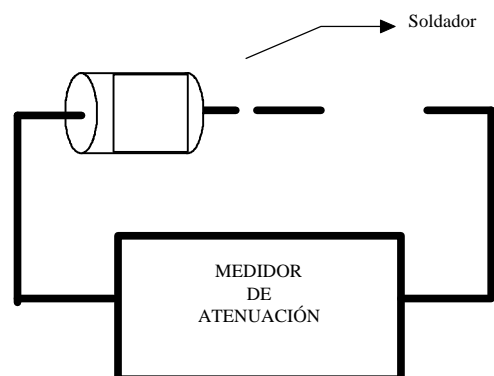
Es difícil de pinchar la información que circula por ellas.

Cualquier corte en la fibra produce atenuaciones en la recepción.

Las velocidades altas implican que los transceptores tienen que ser muy precisos y por lo tanto muy caros.

En cuanto al costo, hay que decir que lo caro no es la fibra en sí, sino que el problema está en el equipo y personal especializado para su montaje; esto se debe a que el límite de curvatura de la fibra es limitado; otro problema está en la soldadura de 2 tramos de fibra.

Para la soldadura, lo que se hace es enfrentarlas en un microscopio, después se mide la atenuación en la unión. cuando esta atenuación es mínima, se procede a soldar las 2 fibras.



Microondas:

Señales de frecuencias por encima de 1GHz.

Son utilizadas para señales punto a punto dada su direccionalidad.

Con señales de "baja frecuencia" del orden de 1 ó 2 Ghz los efectos climáticos las atenúan; con señales del orden de 10 Ghz no tiene problemas con la climatología.

Este medio de transmisión es el utilizado para la comunicación con los satélites.

Radio:

Tiene un ancho de banda muy limitado.

Se comenzó utilizando para redes muy concretas (Salas de ordenadores, etc...).

Infrarrojos:

Son señales luminosas fuera del rango visible. Tiene atenuaciones muy altas y es muy direccionable.

Un tipo de atenuación es la producida por los fluorescentes.

LÁSER:

Se utiliza muy poco y de utilizarse se hace vía aire.

Normalmente se utiliza como generador de la luz en fibra óptica

TOPOLOGIAS:

Los enlaces los podemos dividir en 2 grandes grupos:

Punto a punto:

En los cuales la información es visible solamente por el emisor u el receptor.

Un ejemplo seria el CORREO

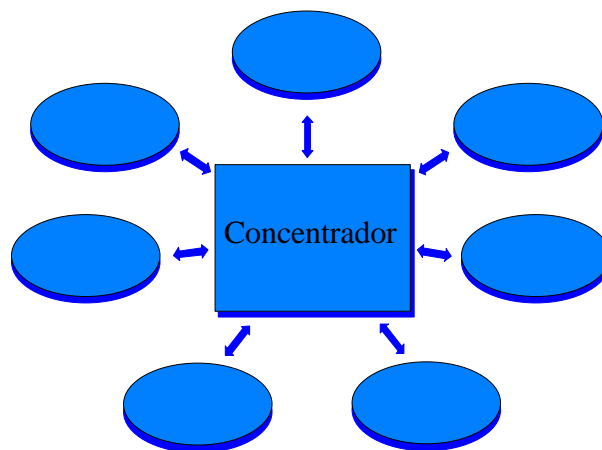
Difusión:

La información generada por un emisor es visible por varios receptores.

Un ejemplo seria una EMISORA DE RADIO.

Formas de Conectar Emisor y Receptor.

Estrella:



Es un enlace punto a punto.

El elemento central es el repartidor y el resto son posibles receptores.

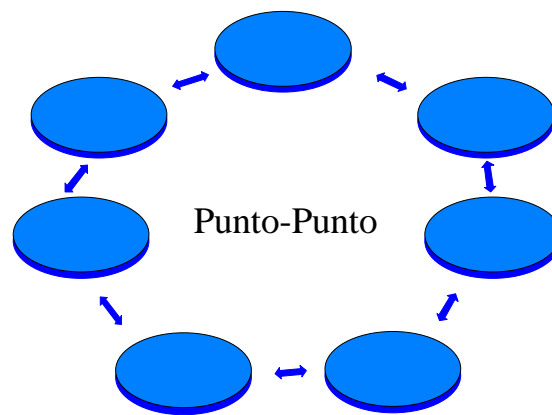
Esta topología tiene el problema de que cualquier problema es el repartidor deja fuera de servicio la red-

Otro problema es que hay que cablear cada receptor al repartidor.

Una ventaja es la de poder mandar información cada elemento independiente.

Otra ventaja es que ante un problema de cableado solo deja de funcionar receptor.

Anillo:



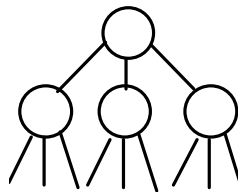
Un interfaz de tipo activo es aquel que recibe una señal, la lee y la vuelve a enviar.

Esta topología no tiene elemento central, por lo tanto es distribuida.

Si hubiera un problema de cableado siempre podemos encontrar un camino alternativo para llevar la información.

Un problema es que se falla un interfaz activo, podemos perder en el toda la información de la red.

Árbol:



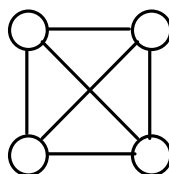
Corresponde con una estructura de tipo jerárquica.

Cada elemento controla a todos los que tiene debajo.

El problema esta es que un fallo en un terminal bloquea a sus hijos en el árbol.

Es un topología punto a punto.

Completa:

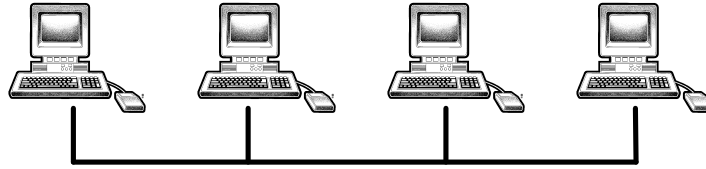


Existe una conexión entre una estación y todas las demás estaciones de la red.

Utiliza enlaces puntos a punto, tiene un alto soporte de fallos.

Tiene el problema del elevado nivel de cableado.

BUS:



Tiene un coste de instalación bajo, ya que tiene poco cableado.
El problema está en que un fallo en el cableado interrumpe toda la red.

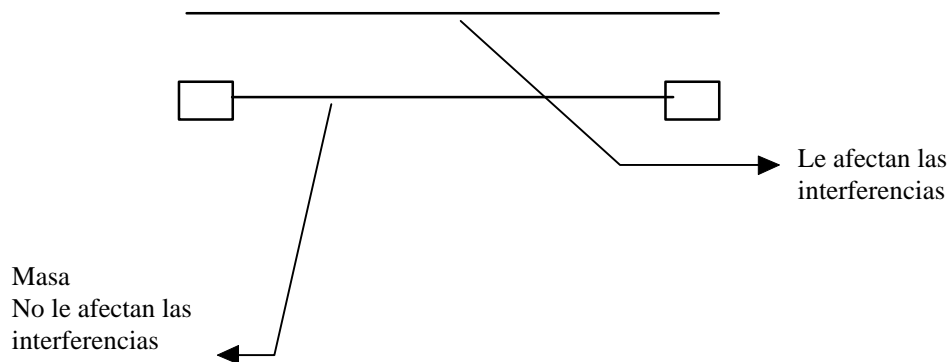
Topología FÍSICA: Es como están conectados las estaciones entre sí.

Topología LÓGICA: Es la estructura que siguen las estaciones para compartir la información.

Tipos de señales:

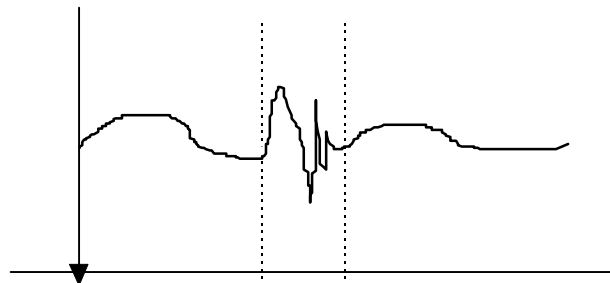
Transmisión no balanceada:

Se transmiten con respecto a un nivel de referencia, por ejemplo a masa. Es muy sensible a las interferencias.



Transmisión balanceada:

Vamos a tener una tensión diferencial.
La referencia se toma de un cable a otro.
Las interferencias afectan tanto a la línea de referencia como a la señal.
Tiene más inmunidad a las interferencias electromagnéticas.



Transmisión de información digital:

Se puede hacer mediante señales analógicas o digitales.

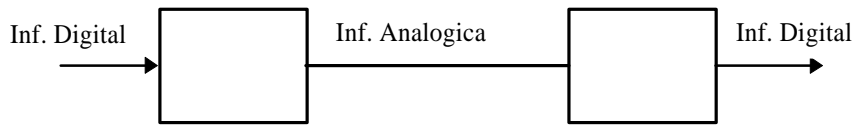
Transmisión mediante señales de tipo analógico:

Transmisión en banda ancha:

Las señales serán de tipo analógico.

El ancho de banda de la transmisión será de 300 a 500 MHz en distancias de hasta 100 mts, si ampliamos la distancia vamos a usar amplificadores.

El esquema de funcionamiento será:



En este caso el ancho de banda ocupado por cada bit entre 1 y 4 MHz dependiendo de la modulación usada.

Multiplexación:

En frecuencia disponemos de un medio de transmisión con un ancho de banda.

Canal1 100 MHz
Canal2 100 MHz
Canal3 100 MHz

En total 300 MHz de forma simultánea

Por cada canal circulara distinta información pero no se puede transmitir a una frecuencia muy cercana a otra ya que se solaparían las 2 señales. Para solucionar este problema de solapamiento, lo que haremos será tener una banda de salva guarda, esto implica una reducción del ancho de banda, aunque elimina las interferencias.

En el tiempo: lo que se hace se dividir el uso del canal en intervalos de tiempo, cada intervalo de tiempo transmite una estación, de esto se encarga la capa de enlace.

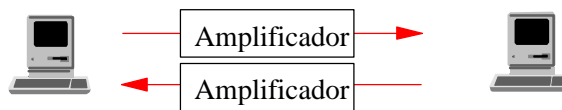
Ninguna de las 2 formas de Multiplexación elimina a la otra.

Los medio de transmisión en banda ancha trabajan generalmente en frecuencia.

Los amplificadores de tipo analógico no son bidireccionales, para solucionar esto, existen 3 posibilidades, estas son las siguientes:

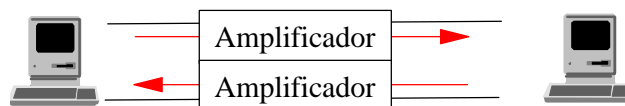
Cable Dual:

Es una estructura de este tipo.



De esta manera no disminuye el ancho de banda, pero es el coste es el problema.

Cable sencillo:



Modulado en frecuencia, se usa un canal para transmitir y otro para recibir, es solo un cable entonces el ancho de banda disminuye, se tiene que usar amplificadores para el cable y que solo amplifique entre el rango de frecuencias del canal.

Para estos amplificadores son caros y el personal muy cualificado.

Las asignaciones de frecuencia se harán:

Baja Frecuencia: Pasa la salida de información de los dispositivos

5 a 30 MHz baja

Con una salvaguarda de 10 MHz

5 a 116 MHz media

Alta Frecuencia: Para recibir información de los dispositivos.

40 a 300 MHz baja

168 a 300 MHz media

Cable coaxial (75 Ohms):

Cable de antena de televisión ancho de banda de 300 MHz

Sistema telefónico:

Para transmisión de voz, ancho de banda muy reducido entre 300Hz y 3Khz, ancho de banda es de 2700Hz.

Frecuencias fuera de este rango se eliminan y no se transmite.

Para transmitir en este sistema se usa un módem que nos transforme la señal digital a analógica para transmitir.

Modulaciones:

ASK Modulación de amplitud.

$$Si(t) = \left(\frac{2Ei(t)}{T} \right)^{\frac{1}{2}} \cdot \cos(\omega t)$$

$$0 \leq t \leq T$$

siendo $i = 1..M$

Se transmiten 2 bits en cada señal.

PSK Modulación en fase.

$$Si(t) = \left(\frac{2Ei(t)}{T} \right)^{\frac{1}{2}} \cdot \cos\left(\omega t + \frac{2\pi i}{M}\right)$$

$$0 \leq t \leq T$$

siendo $i = 1..M$

Se producen saltos bruscos de tensión en el cambio de fase, esto produce un amortiguamiento subcrítico

FSK Modulación en frecuencia

$$Si(t) = \left(\frac{2Ei(t)}{T} \right)^{\frac{1}{2}} \cdot \cos(\omega t + \mathbf{q})$$

siendo $0 \leq t \leq T$
 $i = 1..M$

Modulación Coherente:

Los cambios se producen en los pasos por 0.

Modulación No Coherente:

Los cambios no se producen en los pasos por 0, esto lo que causa es un alto nivel de ruido.

En la modulación coherente los cambios de frecuencia se producen cuando la señal pasa por 0, esto implica que el numero de ciclos es entero. con este sistema no se generan armónicos.

QAM Modulación en fase y amplitud

$$Si(t) = \left(\frac{2Ei(t)}{T} \right)^{\frac{1}{2}} \cdot \cos(\omega t + \phi_i(t))$$

siendo $0 \leq t \leq T$
 $i = 1..M$

Los puntos son los ángulos de desfase 0,90,180,270.

En este tipo de modulación con 4 ángulos de desfase y 2 amplitudes, podremos transmitir 3 bits por unidad de tiempo (t).

Para evitar errores lo que hacemos será alejar los puntos lo máximo posible, aunque sea mas caro.

Los bits se transmiten de la siguiente forma:

(falta dibujo)

Transmisión mediante SEÑALES DIGITALES:

Transmisión en Banda Base

Lo que introduciremos en el canal de transmisión es directamente la señal digital.

Esto produce un alto nivel de ruido, ya que los cambios de tensión son muy bruscos, lo cual también implica la señal ocupa todo el ancho de banda del medio de transmisión, lo que impide que la señal se pueda multiplexar en el tiempo.

Si el medio de transmisión es un cable, el ancho de banda no es infinito, por lo que la señal recibida se vera modificada.

Los límites de velocidad viene dados por el medio de transmisión utilizado.

A más longitud de cable menos velocidad de transmisión.

Este sistema de transmisión funciona de la siguiente manera:

Un 0 es un nivel bajo de la señal y un 1 es un nivel alto de la misma.

Este sistema tiene un fallo, el cual se produce cuando se mandan muchos bit con valor alto conse-

cutivos., ya que por un lado se pierde el sincronismo y por el otro lado con tanto uno, la señal pasa a ser una señal continua, por lo que si después de tantos 1 llega un 0, la señal posiblemente no cambie de estado.

Para evitar este problema se utiliza la codificación Manchester.

Este sistema de transmisión también consigue que el nivel de continua sea siempre el mismo, ya que para cada bit esta igual tiempo en estado alto, como en estado bajo.

Esta codificación funciona de la siguiente forma:

Cuando viene un 0, lo que haremos será pasar de estado alto a estado bajo; mientras que si lo que viene es un 1, lo que haremos será pasar de estado bajo a estado alto.

Esta codificación es normalmente utilizada para redes de área local, mientras que en otros sistemas, lo que se hace es lo contrario, es decir, cuando viene un 0, pasamos a estado alto y si viene un 1 pasamos a estado bajo.



Hasta ahora la información viene contenida en el tiempo de bit, a partir de ahora, la información estará en formato diferencial, es decir el valor de cada bit dependerá del bit anterior.

Codificaciones DIFERENCIALES:

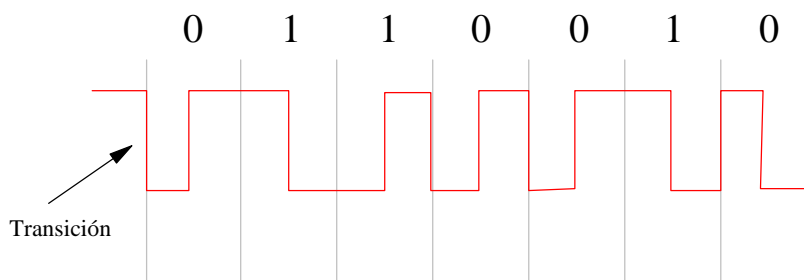
La información ahora se conseguirá comparando el bit anterior con el presente, de forma que $B_{n-1} = B_n$ representara un 0, mientras que si B_{n-1} es distinto de B_n representara un 1.

Ejemplo:

0 1 0 1 1 1

(falta dibujo)

También podemos representar la información en Manchester Diferencial:



El sistema de transmisión de Banda Base se utiliza en redes de área local, longitud de 100 a 300 metros. Se utiliza de codificación Manchester o la codificación Manchester diferencial.

Configuración de una tarjeta de RED:

Primero configuraremos la dirección donde se encuentra la tarjeta, para después configurar la IRQ de la misma. (Normalmente la ADR es la 300h, mientras que la IRQ es la 3).

Otro parámetro a configurar en las tarjetas de red es el tipo de cableado con el que van a trabajar, este tipo de cableado, puede ser por ejemplo el siguiente:

BNC que el para un cable coaxial
RJ45 para cable de par trenzado.
AUI para cable especiales.

Otro parámetro que normalmente esta en la tarjetas de red es el DMA.

Si la tarjeta es una tarjeta clónica, tendremos otra opción mas a configurar, y es la compatibilidad de la tarjeta.

Existen algunas tarjetas en el mercado en las cuales no es necesario seleccionar el tipo de medio de transmisión que van a utilizar, ya que ellas lo detectan directamente.

En algunas tarjetas se le puede poner una memoria EPROM, con la que podremos realizar un arranque remoto.

El funcionamiento de este tipo de arranque es el siguiente:

1º arraque la BIOS del ordenador, para después saltar a la dirección donde se encuentra la EPROM

2º la EPROM se encargara de gestionar el arranque remoto.

La EPROM de las tarjetas esta normalmente situada en la dirección C8000h.

Normalmente las tarjetas de Red llevan un número asociado a ellas, este número lo que hace es indicarnos cual es la dirección física de la tarjeta, esta dirección esta representada con 48bits; este numero no puede ser igual entre 2 tarjetas de la misma RED; en las tarjetas clónicas, sí que podemos modificar este valor, ya que a no ser original, puede darse el caso de que ya exista el número de la tarjeta.

Capa de enlace. (Introducción)

Primitiva de servicio:

Es el formato de los datos entre capas.

Punto de acceso a servicios:

Estos puntos en el sistema mediante el cual se pasan la primitivas.

La primitiva esta formada por:

valor reg. de configuración

Datos

La capa de enlace pasa la primitiva anterior a la física a través de un punto de acceso, en este caso una dir. de memoria.

La respuesta de la capa física sería una primitiva del tipo:

Tipo de datos

Datos

La capa de enlace va a leer a una dirección de memoria (punto de acceso)

Similitud:

En un procedimiento, las primitivas serian los parámetros del mismo o bien los datos devueltos por el procedimiento; mientras que los puntos de acceso serian los registros, las pilas, dirección de memoria, etc...

CAPA DE ENLACE:

La dividimos en 2 subcapas:

La subcapa MAC (Control de acceso al medio)

La subcapa LLC (Control lógico de enlace)

En la subcapa MAC implementa las funcionales que permiten el acceso al medio de transmisión; también implementa la detección de errores.

Existen 3 standard:

IEEE 802.3

IEEE 802.4

IEEE 802.5

Si tenemos un medio de transmisión la información que pasa por dicho medio es vista por todas las máquinas.

El problema que tenemos se produce en el momento que transmiten 2 o mas máquinas al mismo tiempo; ya que las señales se superpondrían; para evitar esto lo que haremos será implementar mecanismos que hagan que solo transmita una sola estación o que si transmiten varias, lo podamos detectar y destruir esos datos.

Estos mecanismos son los siguientes.

Protocolo de contienda:

En el momento que una estación tiene datos que transmitir, va a intentar transmitirla.

Protocolo sin contienda:

en este tipo lo que haremos será establecer un mecanismo que indique en cada momento que estación puede transmitir.

Protocolo de contienda:

Mezcla las 2 técnicas anteriores.

Protocolo de contienda

Se puede dar la situación de que transmitan 2 o mas estaciones al mismo tiempo, esto se llama colisión.

Cuando se produce una colisión, la información es destruida.

La información la transmitiremos mediante tramas que serán de tipo serie. La longitud de las tramas pueden ser fijas o variable, en función de las características de la subcapa MAC

Protocolo ALOHA:

Se crearon en los años 70 en Hawai con la intención de comunicar las estaciones informáticas de las islas.

La principio el medio de transmisión era la radio, para después pasar a ser vía satélite.

Existen 2 variantes:

ALOHA puro:

Cuando una estación tenía que transmitir, se ponía a enviar los datos directamente.

Los datos eran secuencias de bits de longitud fija.

El enviar la trama nos llevara un tiempo que llamaremos tiempo de trama y que a de ser igual a la longitud de la trama.

La velocidad de transmisión en este sistema es:

$$\frac{\text{longitud de trama}}{\text{velocidad de transmisión}}$$

El problema aparece cuando ya tengamos una estación transmitiendo y nos ponemos a transmitir desde otra estación.

El problema de la colisión se soluciona de la siguiente forma:

Las estaciones que están transmitiendo se van a encontrar a la escucha de lo que hay en el medio de transmisión, para así comparar la información que ellos transmiten con la que hay en el medio; si una estación detecta que la información que hay en el medio no es la misma que la que el transmite detecta una colisión.

Entonces lo que hace es retransmitir la trama transcurrido un determinado tiempo, establecido de forma aleatoria.

Tiempo de vulnerabilidad:

Es el tiempo que hay que tener el canal disponible para que una trama se envíe correctamente.

Se necesita 2 veces el tiempo de trama ya que si en el peor de los casos coinciden solo el ultimo bit.

Para evitar que se esto ocurra, es decir un colisión del ultimo bit, el tiempo de transmisión a de ser igual al 2 veces el tiempo de trama.

Para saber si a habido colisión lo que hace el receptor es transmitir la trama y después un silencio de datos.

Es decir que si después del tiempo de trama no hay un silencio el receptor sabe que a habido colisión.

Si S es el numero de tramas transmitidas por unidad de tiempo. S tendrá que ser $0 < S < 1$.

Una estación tiene que transmitir todas las tramas nuevas y además las que halla sufrido colisión.

G es el numero de tramas nuevas mas el numero de tramas que tiene que retransmitir por causa de la colisión.

Si el trafico en el medio de transmisión es bajo G será aproximadamente igual a S

Si el trafico en el medio de transmisión es alto G será mucho mayor que S

Si consideramos la probabilidad de que una transmisión tenga éxito como P, es decir que no sufra colisiones, vamos a tener que $S = G \cdot P$

Si suponemos que las estaciones generan tramas según la distribución de Poason la probabilidad será:

$$P[K] = \frac{G^k \cdot e^{-g}}{K!}$$

La probabilidad de que $P=0$ es $P[0] = e^{-g}$

Si el tiempo de vulnerabilidad es $2T \Rightarrow$ el numero de tramas a transmitir será de $2G$; de forma que solo transmita una estación en este intervalo de tiempo que va a ser e^{-2g} ; por lo tanto las tramas transmitidas con éxito van a ser:

$$S = G \cdot e^{-2g}$$

Suponemos que $G=0.5$, por lo tanto el rendimiento de la red es de 0.184, es decir sustituimos G en $S = G \cdot e^{-2g}$, es decir que el rendimiento de será del 18.4%, lo cual quiere indicar que de cada 100 tramas transmitidas, 18.4 llegan a su destino sin haber sufrido colisión.

Cuando una estación intenta transmitir y encuentra colisión, deja en espera la trama un tiempo aleatorio, por lo tanto en la cola de transmisión están las tramas por colisión mas la tramas nuevas.

rendimiento = (bit transmitido/tiempo) / (complejidad de enlace/tiempo)

Siendo la complejidad de enlace la velocidad de transmisión del enlace.

ALOHA: (continuación)

PURO:

RANURADO:

Si tenemos un tiempo t de transmisión y un tiempo de trata T_t ; lo que hacemos será:

En vez de transmitir cuando se tiene la trama se transmitirá cuando se reciba un tono de aviso de transmisión; este tono, lo que hace es indicar a todas las estaciones que ahora pueden transmitir.

El tiempo que hay entre tono y tono se calcula de la siguiente forma:

Dividimos es tiempo total entre el tiempo de trama.

Con esto lo que conseguimos es que cada estación transmita en cada unidad de tiempo.
Los tonos son de una frecuencia determinada.

Con este sistema se reduce a la mitad el tiempo de vulnerabilidad, por lo que este tiempo se reduce a t .

La probabilidad de que no se tenga colisión es de $P_0 = e^{-G}$; de esta forma las tramas transmitidas con éxito serán :

$$S = G e^{-G}$$

Con $G=1$ tendremos el máximo rendimiento que será de un 36.8%

Ahora las estaciones no transmiten cuando quieren, sino que tiene que esperar a recibir el tono.

En caso de colisión, las estaciones de volverán a transmitir pasado x tiempo, o mejor dicho, después de x tono, siendo x un numero aleatoria de tonos.

Cualidades del ALOHA:

Mayor rendimiento cuando se tiene un numero bajo de estaciones y/o un numero pequeño de tramas.

PROTOCOLO CSMA:

Acceso múltiple por detección de portadora.

Son redes en banda base con los datos en codificación Manchester o Manchester diferencial.

El funcionamiento es el siguiente:

Cuando una estación transmite una trama, lo que hace es transmitir también una señal portadora, la cual desaparece cuando deja de transmitir los datos; por lo tanto cuando una estación quiere transmitir lo que hace es escuchar el canal de transmisión en busca de una señal portadora, si no la encuentra se pone ella a transmitir.

En este sistema se utilizan tramas de longitud variable.

El problema de este sistema aparece cuando 2 estaciones escuchan silencio y las 2 se ponen a transmitir al mismo tiempo, lo que causa una colisión.

Para saber si hay colisión, lo que se hace es escuchar la que hay en el canal con lo que se esta transmitiendo, si lo emitido es distinto a lo escuchado, hay colisión.

Teóricamente la probabilidad de que halla una colisión seria muy baja, pero prácticamente esta probabilidad aumenta considerablemente a causa de los tiempo de retardo del medio de transmisión; ya que desde que una estación comienza a transmitir hasta que la señal se escucha en la totalidad del canal, pasa un tiempo en el cual otra estación puede ponerse a transmitir.

Cuando hay una colisión lo que se hace es terminar de enviar toda la trama, para volver a retransmitirla un tiempo aleatorio después.

Comportamiento de las estaciones si el canal esta ocupado:

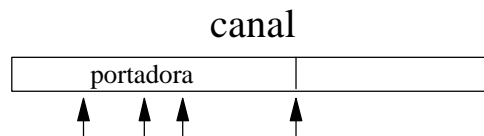
Tendrán que esperarse y iniciar la transmisión en otro momento.

Según en que momento decida volver a transmitir habrá distintos tipos de CSMA.

CSMA 1-persistente:

En el momento que alguien quiere transmitir y el canal está ocupado, lo que hace es escuchar hasta que termine de transmitir el otro.

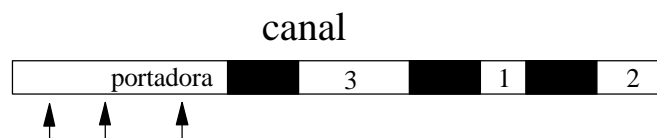
Con tráfico bajo este sistema funciona muy bien, pero con tráfico alto habrá muchas colisiones y por lo tanto nos bajara el rendimiento.



CSMA no persistente:

En este caso cuando una estación está ocupada, lo que hará será esperar un tiempo aleatorio para volver a intentar la transmisión.

Este sistema funciona muy bien si hay tráfico alto, si tenemos tráfico bajo el desperdicio de canal será muy elevado, y por lo tanto tendremos un mal rendimiento.



Siendo T_1, T_1', T_2 tiempos aleatorios.

CSMA p-persistente:

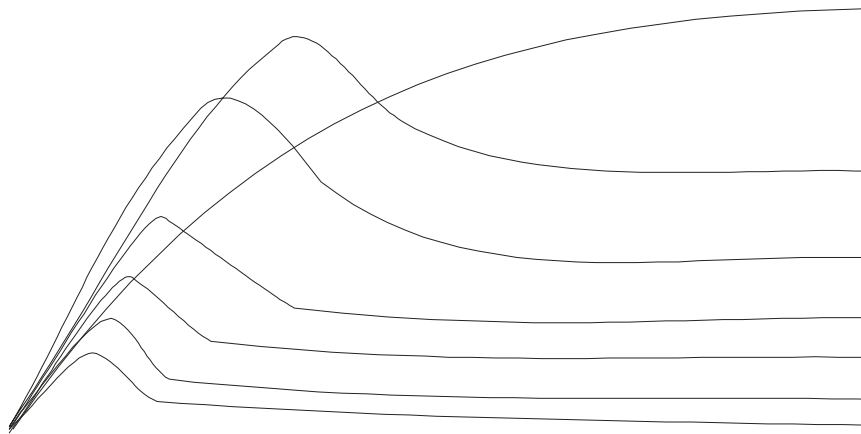
Según el momento tendremos que estaremos trabajando con el sistema CSMA persistente y en otro momento estaremos trabajando como CSMA no-persistente.

La p es la probabilidad de que una estación transmita con el canal libre.

$$0 < p < 1$$

P es el tanto por ciento de veces que se hará el caso de 1 persistente y $q=1-p$ será el tanto por ciento que se hará el caso de no-persistente.

Por ejemplo si $P=0.5$ querrá decir que la mitad de la veces se hará el caso de 1 persistente y la otra mitad será el protocolo CSMA no-persistente.



Otra mejora sobre el protocolo CSMA:

CSMA-CD:

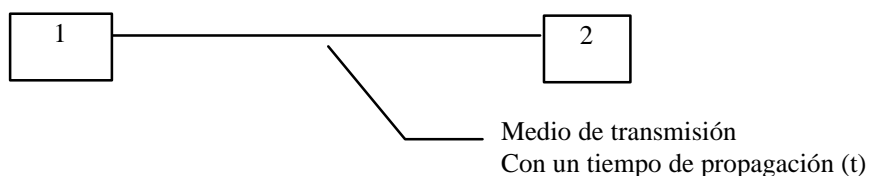
En este sistema lo que hace es lo siguiente:

En el momento de una colisión, lo que hace es abortar la transmisión de la trama, mientras que en los otros CSMA se sigue transmitiendo hasta el fin de la trama.

Esto quiere decir que cuando hay una colisión se deja de transmitir la trama y se ponen a esperar un tiempo aleatorio con esto lo que conseguimos es dejar libre el canal por si otras estaciones quieren transmitir.

El tiempo de detección de la colisión dependerá del tiempo de propagación del medio..

El caso mas desfavorable es 2 veces el tiempo de propagación del medio de transmisión.



Cuando la trama enviada por la estación 1 llega a la estación 2, a pasado un tiempo t que es el tiempo de propagación, si justo antes de llegar la trama la estación 2, esta se pone a transmitir, la estación 1 tardara otro tiempo t en recibir la trama de la estación 2 y por lo tanto en abortar la transmisión de la trama.

Para el estudio de este protocolo se puede modelar como un ALOHA RANURADO con un tiempo de trama $2t$.

Todos los protocolos anteriores son protocolos de contienda, lo que quiere decir que las estaciones luchan por transmitir.

También son llamados protocolos sin tiempo de atención establecido.

Estos protocolos no son validos para trabajar en tiempo real.

Estos protocolos son validos en la transmisión de ficheros.

Protocolos libres de colisión:

Implementan mecanismos para el uso arbitrario del canal.

Mapa de BITS.

Antes de la transmisión de la información, mandaremos una mascara de un bit por cada estación de la red, es decir si tenemos 8 estaciones, mandaremos 8 bits y si tenemos 3 pues 3 bits.

Todas las estaciones conocen cual es su bit de tal forma que cada estación que quiera transmitir activa su bit.

Ejemplo:

Si tenemos 7 estaciones y quieren transmitir las estaciones 2,3 y 6 y después las estaciones 1,3,7

0	0	1	1	0	0	1	0	2	3	6	1	0	1	0	0	0	1	1	3	7
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Los tiempos de transmisión se reparten de la siguiente forma:

0 y 1	No quieren transmitir No se les da tiempo de transmisión
2	Se le da tiempo del canal
3	Se le da tiempo del canal
4 y 5	No se le da tiempo de transmisión
6	Se le da tiempo del canal
7	No se le da tiempo de transmisión

Se transmiten los datos de las estaciones 2,3 y 6.

Se activan los bits para las estaciones 1,3 y 7. Se les asignan los tiempo y se transmiten las tramas.

Con este sistema el canal siempre esta ocupado , ya que cuando no se transmiten los datos se transmiten las mascarar.

El tiempo de trama es variable aunque normalmente esta limitado en longitud.

Este protocolo nos crea dependencia con respecto a las estaciones de mayor numero, ya que desde que quiere transmitir hasta que transmite depende de cuantas estaciones de numero menor quieren transmitir.

Este sistema consigue su máximo rendimiento cuando tenemos pocas estaciones y mucho trafico de datos.

Protocolo BRAP:

Reconocimiento de difusión con prioridades alternas.

Es una variación del sistema anterior.

El funcionamiento es el siguiente:

Se comienza a transmitir la trama, si una estación no quiere transmitir pone un 0, mientras que si quiere transmitir poner un 1 y después los datos que quiere a transmitir.

En los protocolos sin colisión podemos asegurar que una estación transmitirá en X tiempo, mientras que en los protocolos de contienda no sabemos cuando vamos a enviar la trama.

*El tiempo X suele ser numero de estaciones * tiempo de trama.*

Este protocolo tiene mayor rendimiento a menor numero de estaciones; es decir que a bajo numero de estaciones y bajo trafico se desaprovecha el cañal transmitiendo las mascarar; por lo que nos interesa que se un numero alto de datos pero un numero bajo de estaciones.

En estos sistemas el rendimiento se calcula de la siguiente forma:

Información transmitida

Bits transmitidos

Para reducir el numero de bits de la mascar se creo el protocolo MLMA (multiacceso /Multinivel)

MLMA

Supongamos que tenemos 1000 estaciones numeradas del 0 al 999.
Lo que hacemos con este protocolo es lo siguiente:

Creamos un rango de numeraciones por ejemplo decádica.

Primero emitiremos 10 bits, las estaciones que quieran transmitir, lo que harán será marcar su bit de mayor peso, después se enviarán 10 bits para el bit seleccionado más alto y así sucesivamente.

Veamos un ejemplo:

9	8	7	6	5	4	3	2	1	0		
1	0	0	1	0	0	1	0	0	0	xxx	Solicitan transmitir las estaciones 9xx, 6xx y 3xx
0	0	0	0	0	0	1	0	0	1	9xx	Comenzamos con el 9xx, y solicitan transmitir 93x
0	0	0	1	0	0	0	1	0	0	93x	93x y transmiten 936 y 932
0	1	0	0	0	0	0	0	0	1	90x	908, 900
0	0	1	0	0	0	0	0	0	0	6xx	67x
1	0	0	0	1	0	0	1	0	0	67x	679, 675, 672
0	1	0	0	0	1	0	0	1	0	3xx	38x, 34x, 31x
0	0	0	1	0	0	0	0	0	1	38x	386, 380
0	0	0	0	1	0	0	0	0	0	34x	345
0	1	0	1	0	1	0	0	0	1	31x	318, 316, 314, 310

Si quisieran transmitir todas las estaciones hay que enviar 1110 bits.

Ejercicios:

Las estaciones que quieren transmitir son:

871, 868, 860, 835, 631, 627, 600, 105, 10, 4, 2

	9	8	7	6	5	4	3	2	1	0	
xxx	0	1	0	1	0	0	0	0	1	1	
8xx	0	0	1	1	0	0	1	0	0	0	
87x	0	0	0	0	0	0	0	0	1	0	871
86x	0	1	0	0	0	0	0	0	0	1	868, 860
83x	0	0	0	0	1	0	0	0	1	0	835
6xx	0	0	0	0	0	0	1	1	0	1	
63x	0	0	0	0	0	0	0	0	1	0	631
62x	0	0	1	0	0	0	0	0	0	0	627
60x	0	0	0	0	0	0	0	0	0	1	600
1xx	0	0	0	0	0	0	0	0	0	1	
10x	0	0	0	0	1	0	0	0	0	0	105
0xx	0	0	0	0	0	0	0	0	1	1	
01x	0	0	0	0	0	0	0	0	0	1	10
00x	0	0	0	0	0	1	0	1	0	0	4, 2

Descubrir que estaciones quieren transmitir:

9	8	7	6	5	4	3	2	1	0
0	0	1	0	1	0	0	0	1	0
0	0	1	0	1	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	0
1	1	1	0	1	0	1	1	0	0
1	0	0	1	0	0	1	0	0	1
0	1	0	0	0	1	0	1	0	0
0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	1	0	0	0	0
1	1	1	1	0	1	1	0	0	1

Las estaciones que quieren transmitir son:

778, 772, 758, 751, 589, 588, 587, 585, 583, 582, 198, 194, 192, 165, 138, 136, 134, 109, 108, 107, 106, 104, 103, 100

A número más bajo de estación implica que el tiempo de atención es más variable y largo.

PROTOCOLO CUENTA ATRÁS BINARIO:

Cuando una estación quiere transmitir, lo que hará es poner el bit que le toque a 1 y vuelve a lanzar la trama de 10 bits para definir la decenas y centenas, siempre que otra estación de orden superior no quiera transmitir; es decir:

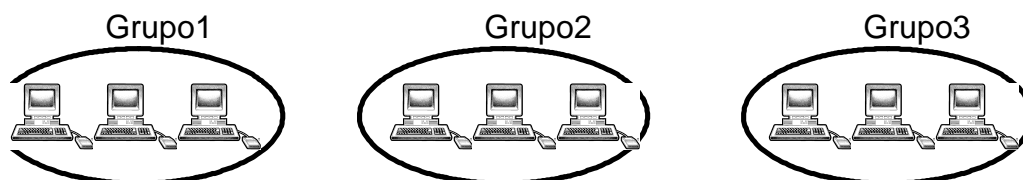
Suponemos que quiere transmitir la estación 239.

9	8	7	6	5	4	3	2	9	8	7	6	5	4	3	9
0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1

Si por ejemplo quieren transmitir las estación 935 y la estación 625, transmitir la 935, ya que es este protocolo la prioridad es mayor cuanto mayor es el número de la estación.

PROTOCOLO DE CONTIENDA LIMITADA:

En estos protocolos básicamente lo que se hace es crear grupos; de tal forma que dentro de cada grupo se trabaja con protocolos de contienda, mientras que entre los grupos se utiliza uno libre de contienda.



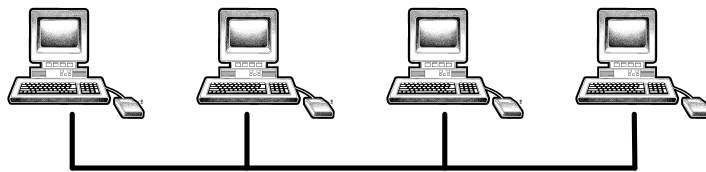
Las estaciones de cada grupo pelearán por transmitir (contienda)
Entre los grupos no pelearán por transmitir (libre de contienda)

IEEE802

(Normas del protocolo IEEE802)

- 802.1 .- Es una introducción a las normas 802; tenemos definido el formato de las primitivas del interface
- Las primitivas nos van a permitir la comunicación entre la subcapa LLC y las capas superiores.
- 802.2 .- Especifica las características de a subcapa LLC
- 802.3 .- Implemetacion de la subcapa MAC para el protocolo CSMA/CD 1p
Esta norma es más conocida por ETHERNET que es ligeramente distinta al CSMA/CD 1p
- 802.4 .- Protocolo libre de contienda con paso de testigo en bus.

Topología en BUS:



Consiste en tener un elemento que hace de testigo, el cual ira pasando de una a otra estación, este testigo es el que indica que estación puede transmitir.

El testigo hace el recorrido de un anillo lógico.

Cuando una estación quiere transmitir, espera a que le llegue el testigo, lo retiene y mientras tanto transmite lo que quiere transmitir; si por el contrario no quisiera transmitir, lo que hará será pasar el testigo.

La implemetacion es mas compleja en la subcapa MAC.

El testigo a de ser una secuencia de bits que no pueda ser un dato.

Las estaciones tienen que conocer a sus vecinos, es decir a su antecesor y a su predecesor.

Los procesos de inserción o dar de alta a una estación, tienen que ser controlados

Si por causa del ruido el testigo se perdiera, las estaciones que podrán transmitir.

- 802.5 .- Paso de testigo en anillo.

La topología física y lógica de la red es en anillo.

El acceso al bus se hará también por testigo igual que antes.

La complejidad es bastante similar al anterior, aunque su implemetacion es mas compleja.

FDDI trabaja con red en anillo y fibra óptica.

Estructura del 802

(falta dibujo)

La subcapa MAC se encarga de la detección y corrección de errores.

- Mecanismos de detección de errores
- Mecanismos de corrección de errores.

Cuando nosotros recibimos una secuencia de bits, los analizamos, si consideramos que es correcta la tratamos; pero si por el contrario hay un error y estamos en detección de errores, la secuencia se desecha; mientras que si estamos en corrección, podremos reparar el error en la secuencia (detecta y corrige los errores)

La implementación de estos sistemas implicaría que hay que transmitir una información adicional.

DATOS	I. ADICIONAL
--------------	---------------------

El receptor con los datos y la información adicional detectada, corregirá el error si lo hay. Normalmente la información adicional es mayor para la corrección de errores y para la detección de los mismos.

Ocurre que la detección y/o corrección no es 100% segura, ya que esta normalmente entono al 99.996%

Métodos de corrección:

Códigos de Hamming

Métodos de detección:

CRC (código de redundancia cíclica)

¿Como elegir entre uno y otro?

La elección entre uno y otro depende del ancho de banda y del trafico en el medio de transmisión.

Con ancho de bits altos y/o trafico bajo es recomendable utilizar métodos de detección.
Con ancho bajo y/o excesivo trafico es recomendado utilizar los métodos de corrección.

CÓDIGO DE HAMMING:

Suponemos m bits de datos y vamos a utilizare r bits de redundancia; por lo tanto los bits transmitidos serán:
 $m+r=n$

Entenderemos como distancia Hamming como el numero de bits en que difieren 2 secuencias correctas (la menor de todas)

Los bits de redundancia son siempre los mismos par ala misma secuencia de datos.

La distancia de Hamming nos da idea del nivel de protección del sistema; si por ejemplo la distancia es 2, si se generan 2 errores puede ser que el código recibido es válido.

emisor	receptor
0110 00	0111 01

Secuencias válidas:

0110 00
0111 01

Tanto los datos emitidos como los recibidos son correctos, por lo que podría pasar el error totalmente inadvertido.

Para la detección de errores se precisa un código con una distancia $e+1$; mientras que para la corrección de errores lo que precisáramos será un código de $2e+1$.

Algoritmo para los códigos de Hamming:

Los bits de redundancia serán los que ocupen las posiciones potencias de 2; mientras que el resto de los bits serán de datos.

Los bits de redundancia serán bits de paridad, en este caso paridad par.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	1	1	1	1	1	0	0	0	0	0	1	0	1	0	0	1	1	0

BIT REDUNDANCIA	BITS CHEQUEADOS	
1	1,3,5,7,9,11,13,15,17,19	Los impares
2	2,3,6,7,10,11,14,15,18,19	El y el siguiente; 2 libres, 2 si, 2 no, 2 si ...
4	4,5,6,7,12,13,14,15,18,19	El y 3 mas; 4 libres, 4 si, 4 no, 4 si, 4 no ...
8	8,9,10,11,12,13,14,15	el y 7 mas, 8 libres, 8 si 8 no, 8 si, 8 no ...
16	16,17,18,19,20	el y 15 mas, 16 libres, 16 si, 16 no, 16 si, 16 no

La distancia de este código es de 3; por lo que nos permitirá detectar 2 errores y corregir un error.

La distancia es 3 por que si cambiamos el bit 6 cambiaría el $6 + 2$ de verificación (2,4), lo que implica una distancia de 3.

Para saber cual es el erróneo y corregirlo bastara con sumar los pesos de los bits de paridad que resultan incorrectos.

Suponemos que tenemos un error en el bit 11.

Miramos la paridad y vemos que hay error en los bits de verificación 1,2,8; por lo tanto:

$$1+2+8=11; \text{ el 11 es el bit erróneo.}$$

DETECCIÓN DE ERRORES:

CRC (códigos de redundancia cíclica)

No nos permite corregir pero si detectar errores en la transmisión.

Para conseguir los bits de redundancia tendremos un polinomio generador, dicho polinomio tienen que conocerlo tanto el emisor como el receptor.

Nº de bit del polinomio-1=nº de bits de redundancia.

Las secuencia a transmitir esta compuesta por los bits de datos mas el resto.

Forma de conseguir la trama a enviar:

Se ponen los bits de resto a 0.

Se divide el trama compuesta por los datos mas los 0 del resto por el polinomio.

Se sustituyen los 0 de la trama por el resto obtenido.

Se transmite la trama.

En el receptor se utilizar la formula $\text{dividendo} = \text{divisor} + \text{resto}$.

Ejemplo de CRC:

10001101101001011 000

1011

<u>1011</u>	
1101	01100111000001000
<u>1011</u>	
10011	
<u>1011</u>	
10000	
<u>1011</u>	
1011	
<u>1011</u>	
0000001011	
<u>1011</u>	
0000000	

10001101101001011 000	1011
<u>1011</u>	
1101	01100111000001000
<u>1011</u>	
10011	
<u>1011</u>	
10000	
<u>1011</u>	
1011	
<u>1011</u>	
0000001011	
<u>1011</u>	
0000000	

Polinomios generadores:

CRC-16 ($X^{16}+X^{15}+X^2+1$) Tiene las bits 16, 15, 2 y 0 a 1.

CRC-CCITT($X^{16}+X^{12}+X^5+1$) Tiene los bits 16, 12, 5 y 0 a 1.

Estos polinomios cubre el 100% de los errores simples, los dobles, los errores con numero impar de bits, errores de ráfagas con longitudes de 16 o menos bits.

El 99.997% de las ráfagas de 17bits

El 99.998% de las ráfagas de mas 18 bits.

Se utilizan en redes de área local en la subcapa MAC.

Especificación subcapa MAC

IEEE802.3 corresponde con un protocolo CSMA/CD

acceso al medio por detección de portadora con detección de colisión.

Nos cubre características sobre la capa física y la subcapa MAC.

Capa Física:

La notación a emplear va a ser la siguiente:

Estará compuesta por 3 valores:

1º.- Nos indica la velocidad de transmisión expresada en Mb/seg.

2º.- Indica el tipo de banda:

Banda base: es indicada con la palabra BASE.

Banda ancha: es indicada con la palabra BROAD.

3º.- Longitud del segmento expresado en 10^2 (valor aproximado)

El primer medio en IEEE802.3 era de 10base5, lo que indica una velocidad de transmisión de 10Mb/seg. con banda base con 500 metros.

El IEEE802.3 se implementa sobre cables coaxiales, pares trenzados y fibra óptica.

Con fibra óptica aparece el problema de la detección de la portadora.

Se denomina comercialmente como coaxial grueso.

Esta compuesto por un cable coaxial de 0.404 pulgadas y una impedancia generalizada de 50.

La designación comercial es RG8 o cable amarillo.

La conexión entre estos cables se realiza con un transceiver.

Normalmente son conectores de tipo vampiro, con respecto al cable coaxial, mientras que la conexión al ordenador se realiza por medio de cable AUI.

La conexión por medio de AUI es de 50 metros y el conector que junta el cable AUI con el ordenador es del tipo del puerto serie.

La distancia mínima entre 2 transceiver es de 2.5 metros, ya que si están mas juntos pueden dar problemas.

En un mismo segmento (500 metros de cable) puede haber un máximo de 100 estaciones (transceiver).

Existen transceiver con mas de una salida AUI, lo que permitiría conectar mas estaciones por metro.

Cuando las estaciones están a mas de 500 metros o hay mas de 100 estaciones se tiene que utilizar estaciones de repetición que permiten conectar varios segmentos, simplemente lo que hacen es regenerar la señal.

El máximo numero de segmentos es de 5 con una longitud máximo de 500 metros; o lo que es lo mismo 4 repetidores, lo que nos permitiría conectar 500 ordenadores o estaciones en una distancia de 2500 metros.

Las terminaciones no están al aire, sino que se pone una resistencia (en esta caso una resistencia de 50.

Si se dejan los extremos al aire la señal rebota y vuelve hasta que se amortigua, lo que causa un alto nivel de ruido y de posibles errores.

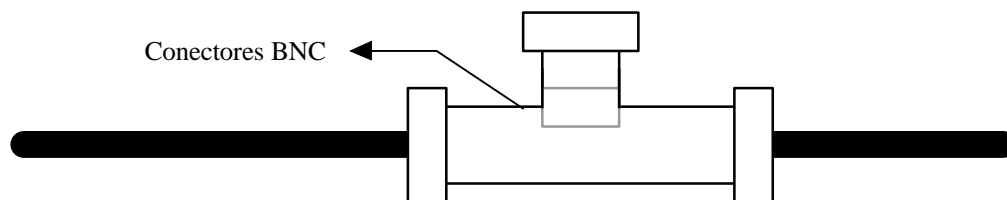
Al poner la resistencia la señal desaparece, lo que es decir, que no rebota.

Especificación del 10Base2

Topología en bus con cable coaxial igual que el 10Base5, con una velocidad de 10Mb/s en banda base, con una impedancia generalizada de 50, y un diámetro de 0.2 pulgadas y su nomenclatura es la de RG58.

Se utilizan conectores BNC, los transceiver están en las tarjetas de red.

Modo de conexión.



Es un cableado barato, la longitud máxima de segmento es de 185 metros.

Nos permite un máximo de 30 estaciones por segmento.

Para ampliar el numero de estaciones o la longitud del segmento se deben utilizar repetidores, que nos limitan a 5 segmentos, pero solo 3 tiene que tener estaciones.

Esto implica una distancia máxima de 925 metros con 90 ordenadores.

Ejemplo de conexión:

(falta dibujo)

La problemática de este sistema de conexión es que cuando se crea un corto circuito o se corta el cable de conexión, la red se va a pique.

Cada segmento tiene que tener 2 terminaciones.

Especificación 10BaseT

Cable de par trenzado, que esta compuesto por 8 cables trenzados 2 a 2 siendo STP o UTP.

Tenemos varias categorías de par trenzado:

Categoría 3: Frecuencia de trabajo del orden de 16 MHz.

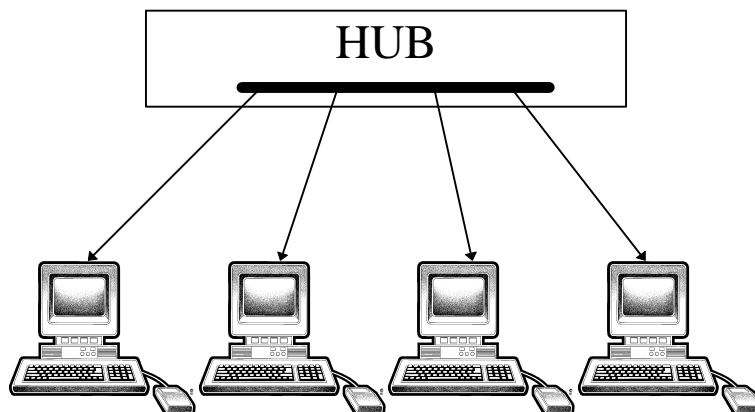
Categoría 4: Frecuencia de trabajo del orden de 20 MHz.

Categoría 5: Frecuencia de trabajo del orden de 100 MHz.

Actualmente los mas habituales son los de categoría 3 y 5; ahora especialmente son los de categoría 5.

La topología de este cableado es en estrella, aunque la topología lógica es en bus.

Para que esto funciones lo que se hace es utilizar un concentrados llamado HUB; a dicho concentrador llega un cable por cada estación conectada a el; para que internamente realice la función de un bus.

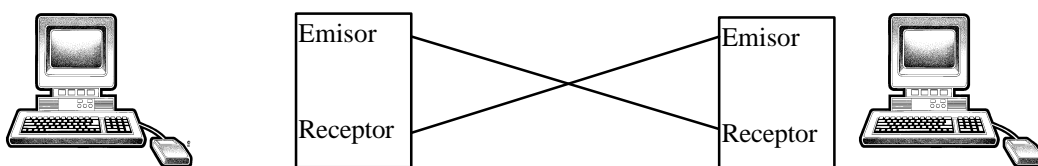


De los 4 pares se utilizan 2 pares uno para emisión y otro para recepción.

Este tipo de cableado nos da una distancia máxima de 100 metros entre el ordenador y el HUB, por lo tanto la distancia que recorre la información es de 200 metros.

A la hora de detectar problemas solo falla un ordenador; si fallan todas las estaciones, lo que falla es la HUB.

Para conectar 2 ordenadores con esta especificación, solo hay que cruzar los cables de emisor con los de recepción.



10BaseF

El medio de transmisión de esta especificación es la fibra óptica.

Para solucionar el problema del 802.3 con fibra óptica lo que haremos será utilizar una topología en estrella.

La configuración en estrella la realizaremos de la siguiente manera:

El centro de la estrella estará compuesto por un HUB de fibra óptica, y las puntas de la estrella serán las estaciones.

Para cada estación utilizaremos un cable para transmitir y otro para recibir.

Para detectar la colisión, lo que se hace es primero pasar la señal luminosa a señal eléctrica, cosa que hace el HUB; un vez que tenemos la señal eléctrica, introducimos la portadora.

A partir de ahora el funcionamiento en la detección de colisión por portadora es el mismo que en los casos anteriores.

Como es multidifusión, la señal que llega al HUB se reparte por todas las salidas del mismo, es decir por las líneas Rx.

En 802.3 se utilizan normalmente el 10Base5, 10Base2, 10BaseT y el 10BaseF.

El 10BaseF se utiliza solo en situaciones con un alto nivel de ruido eléctrico, o bien en distancias muy largas, donde la poca atenuación de la fibra hace que esta sea rentable.

Otros estándares del 802.3 son:

10Broad36:

Impedancia 75. (en cable coaxial)

Distancia de hasta 3.6Km.

Se modula la información digital en analógica.

Los amplificadores analógicos son monodireccionales, por lo que o bien utilizamos 2 cables uno para transmitir y otro para recibir; o ampliamos el ancho de banda; para que así utilizar una trozo del ancho para emitir o otro trozo del ancho para recibir.

10Base5 (ATT estándar)

Par trenzado

Se utiliza un HUB 250 metros entre estación y HUB

Velocidad de transmisión del orden del 1M/seg.

100BaseT y 100BaseVG

Par trenzado o fibra óptica.

El par trenzado de categoría 3 y 5.

Una velocidad de transmisión es del orden de las 100Mb/seg.

La distancia de transmisión entre el HUB y la estación es de 100 metros.

La más utilizada es la 100BaseT

Señalización de la especificación 802.3

Utiliza una codificación Manchester.

Cuando queremos transmitir un 0 mandamos un flanco de bajada; mientras que si queremos transmitir un 1 el flanco será de subida.

Este tipo de codificación nos garantiza que por cada bit hay un cambio de estado; por lo que no se pierde el sincronismo entre las estaciones; esta codificación también consigue que el nivel de tensión de continua en el medio de transmisión sea constante.

El problema de esta codificación es que requiere un ancho de banda el doble de los bits a transmitir.

Ejemplo:

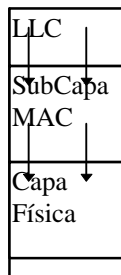
Si queremos transmitir 10Mb/seg. nos hace falta un ancho de banda de 20Mb/seg.

El caso del ejemplo se consigue con secuencias de 0 o 1 muy largas, ya que la frecuencia es mas elevada que cuando se transmiten 0 y 1 de forma aleatoria.

SubCapa MAC del 802.3

Funcionalidad de esta subcapa:

Da servicio a las capas superiores. Existe 3 tipos de servicios estos servicios los asociaremos a 3 primitivas (1 servicio una primitiva)



Da servicio a las capas superiores.
Existe 3 tipos de servicios estos servicios los asociaremos a 3 primitivas (1 servicio una primitiva)

Las primitivas de esta especificación son las siguientes:

MA_DATA.request
MA_DATA.confirm
MA_DATA.indication

MA_DATA.request

Va de la capa superior a inferior; es decir va de la capa LLC a la subcapa MAC

Solicitar el inicio de un servicio.

Solicita que la subcapa MAC realiza una operación determinada.

Solicita el estado de la red (numero de colisión, si funciona bien, etc...

MA_DATA.confirm

Va de la subcapa MAC a las capas superiores.

Da el resultado del servicio asociado solicitado anteriormente.

Ejemplo:

Nos indica que va a poder o no poder transmitir.

Van asociadas a MA_DATA.request

Es la contestación a MA_DATA.request.

MA_DATA.indication

Va de la subcapa MAC a las capas superiores.

Evento significativo para la capa superior.

Ejemplo:

Cuando recibimos información que va a la estación, hay que indicar a la capa superior que va a recibir datos.

No va asociado a ninguna directiva de MA_DATA.

Su utilidad es la de mandar información importante para la capa superior.

Parámetros de las primitivas: MA_DATA.

MA_DATA.request:

Va a constar de 3 partes:

- 1.- Dirección destino (DA) *Dirección de longitud fija a donde va a ir la información.*
- 2.- Unidad de servicio de datos (SDU) *Información que se quiere enviar.*
- 3.- Clase de servicio *En 802.3 solo se implementa un servicio.*

La longitud de SDU es variable, por lo que por ejemplo se podría indicar la longitud con los 2 primeros bytes del SDU.

MA_DATA.confirm:

Debe contener la suficiente información para asociar la respuesta con la solicitud correspondiente.
Tendremos tantas MA_DATA.confirm como MA_DATA.request.

La constentacion (confirm) no tiene que estar en el mismo orden que se solicitan /request); por lo tanto tenemos que buscar una forma de asociar una request con una confirm; una forma seria con un identificador de request, el cual lo devolvería confirm.

MA_DATA.indication:

Va indicada a la recepción de datos.

Consta de 4 elementos:

Dirección destino (DA)

Dirección origen(SA)-> *De donde viene la información.*

Esta porque la SUBCAPAMAC puede filtrar información por lo que tiene que decir a donde va a ir la información después del filtrado.

Si la información va destinada a todas las estaciones, estará indica ahí con un BRO-ADAST.

Unidad de servicio de datos(SDV)->*Datos+informacin de longitud*

Estado de recepción.->*Informacion de si hay o no hay error o si esta segmentada o no, etc...*

Estructura de trama MAC en 802.3

La unidad de transmisión será la trama (octeto o byte).

Estructura de la trama:

(falta dibujo)

Estructura Básica:

Cabecera:

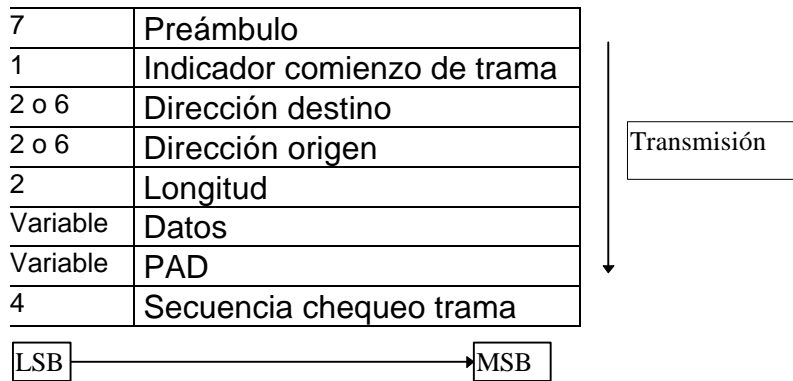
En la cabecera tendremos información del protocolo, secuencia de octetos para que indica su cometido en cada capa.

Datos:



→ Todas las capas (como las muñecas rusas)

Formato de la C.MAC EN 802.3



El tamaño de la dirección a de ser igual, es decir o las 2 tiene una longitud de 6 o 2.
 El orden de transmisión va de LSB a MSD empezando por el preámbulo hasta la secuencia de chequeo de trama.

Los 7 primeros octetos sirven como bits de sincronismo, los cuales están formado por secuencias de 1 y 0 alternados.

10101010....

Si dibujamos esto en Codificación Manchester:

(falta dibujo)

El indicador de comienzo de trama es un valor fijo el cual indica el comienzo de la trama; el la siguiente secuencia de bits: 10101011

Dirección destino/origen:

En origen nos indica quien esta generando la trama.

En destino nos indica a donde va destina la información.

La longitud de ambas a de ser la misma.

El bits de menor peso nos indica:

0 si la dirección es individual.

1 si la dirección es de grupo.

Esto solo hay que mirarlo en la dir de destino, ya que la de origen es siempre 0

En dirección de 2 y 6 es común todo lo anterior.

En la dirección de 6 octetos el LSB+1 indica lo siguiente:

0 Dirección administrada globalmente.

Dirección visibles para toda la red.

1 Dirección administrada localmente.

Dirección que va a ver en parte de la red (solo en el grupo al que pertene-

ce).

Longitud (2 octetos// 16 bit)

Nos indica el numero de octetos de la zona de datos.

El primero octeto transmitido es el de mayor peso, es decir que cuando recibimos un 2 y un 5 tendremos una

longitud de datos de 25 octetos.

Con este indicador podremos conocer el tamaño de datos y por lo tanto el de la trama.

Datos (longitud variable):

Contiene la información a transmitir, la cual viene de capas superiores.

Los datos de MA_DATA.request va a ser la información de la zona de datos.

Esta zona de datos presenta una longitud máxima de 1500 octetos.

Lo que hará un tamaño máximo de trama de 1526 octetos; suponiendo las direcciones de 6 octetos.

Nosotros podemos enviar cualquier secuencia de bit en esta zona, ya que es transparente, esto hace que no tengamos restricciones en la información; Pero hay protocolos donde esto no pasa, en estos protocolos hay que evitar una serie de secuencias de bits, ya que dichas secuencias indicarian inicio o fin de trama, etc..

PAD (longitud variable):

Nos asegura un tamaño mínimo de trama, por lo tanto no tiene porque estar es todas las tramas.

El tamaño mínimo de trama, es por que si la trama es muy pequeña y el tiempo de propagación es grande, la trama se termina de enviar sin detectar colisión, pero si puede existir colisión, por lo tanto la estación que lo envía lo da por bueno..

La forma de que la estación que envió primero sepa que hay colisión es si en el tiempo de propagación después de enviar su trama recibe algo.

Con esto solo sabe si hay o no hay colisión, pero no sabe si dicha colisión es en su trama o en otra.

El campo PAD, lo que hace es que el tiempo de trama sea mayor y por lo tanto llegue a todas las estaciones antes de que se termine de enviar a todas las estaciones.

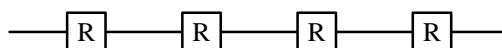
Ejemplo:

Si el numero de octetos mínimos es de 72 y nuestra trama es de 32, el PAD sea de 72-32 octetos.

Normalmente el campo PAD se llena de el valor 20h (código ASCII del espacio).

En el 802.3 el tamaño mínimo de trama es de 72 octetos.

Este tamaño de trama tiene encuentra la longitud máxima entre 2 estaciones en cualquiera de sus configuraciones.



El tamaño del PAD viene dado por la siguiente expresión:

Valor máximo $[0, \text{tamaño mínimo de trama} - (8N + 2 \text{ tamaño ADR} + 48 + 64)]$ expresado en bits.

Si tenemos menos de 72 octetos el numero será mayor, por lo tanto este será el tamaño del PAD y tenemos mas de 72 octetos el resultado será menos y por lo tanto cogemos 0.

El campo long solo nos indica el tamaño del campo de datos, no es de PAD, para así no crean problemas a la hora de leer los datos.

A la subcapa MAC los 2 primeros campos no llegan por lo que en algunos sitios se toma como longitud de trama el valor 64.

CRC (longitud de 4 octetos):

Normalmente se utilizan 33 bits para el polinomio generador.

Este nos cubre desde la longitud hasta el PAD ambos incluidos.

¿Cuando es una trama incorrecta?

Un trama es incorrecta cuando se cumplen almenos 1 de las siguientes condiciones:

- No hay un n° entero de octetos.
- Inconsistencia entre la longitud de datos y la longitud de trama.
- CRC incorrecto.

Detección de colisión:

Normalmente cuando una estación detecta que hay colisión lo que hace es dejar de transmitir; lo que podría llegar a causar el siguiente problema:

Si el producirse la colisión la estación deja de emitir, existirán algunas estaciones que no se enteraran de la colisión; para solucionar este problema, lo que se hace es lo siguiente:

Cuando hay colisión la estación sigue transmitiendo durante $2t$ siendo t el tiempo de propagación del medio de transmisión.

Optimizacion del rendimiento del protocolo:

Cuando hay colisión se dejar de transmitir durante un tiempo aleatorio, el cual esta situado entre un máximo y un mínimo.

Si el tiempo es muy alto, lo que pasa es que se pierde ancho de banda; si por el contrario el tiempo es muy bajo, lo que pasa es que el numero de colisiones aumenta.

Para optimizar el rendimiento del protocolo lo que se hace es ajustar el tiempo de forma dinámica; de la siguiente forma:

Consideraremos el tiempo aleatorio en ranuras, las cuales son espacios de tiempo del orden de $2t$.

Si una trama que una estación quiere enviar es la primera colisión que se produce el valor aleatorio esta comprendido entre 0 y 1, si es la segunda colisión, entonces el valor aleatorio estará entre 0 y 3 ranuras.

El rango de los números aleatorias va definido por la siguiente expresión:

n° ranuras $\rightarrow 0..2^Y - 1$ siendo Y el numero de colisiones consecutivas para una trama.

El numero máximo de ranuras por trama es de 1023.

El numero máximo de reintentos por trama es de 16; cuando es da este caso lo que pasa es que la subcapa MAC se lo comunica a las capas superiores, para que estas soluciones el problema.

Si una estación tiene muchas tramas que enviar podría darse el caso de que la estación monopolizara el canal, ya que cuando terminara una trama, inmediatamente intentaría mandar la siguiente, por lo que las demás estaciones tendría siempre el canal ocupado; para solucionar este problema lo que se hace es poner un tiempo de retardo entre cada trama, para que así de esta forma, las demás estaciones encuentran el canal libre en algún momento y puedan transmitir.

RESUMEN DE FUNCIONES DE LA CAPA MAC

1.- Transmisión de tramas:

- a) Recopilación de datos de la capa superior y construcción de la trama.
- b) Pasa la secuencia de bits a la capa física para su transmisión.

2.- Recepción de tramas:

- a) Recepción de una cadena de bits de la capa física.
- b) Descapsulado de la trama y envío de la información a la capa superior de la tramas con la dirección de la estación o la dirección BROADCAST (Dirección que indica que es para todo la red (todos los bits a 1)).
- c) Descartar las tramas que no van dirigidas a la estación.

3.- Esperar a que el canal quede libre cuando se desea transmitir.

4.- Añadir secuencias de control de trama y verificación lineal de octetos(transmisión)

- 5.- Chequeo de las tramas para la detección de errores y verificar el tamaño de octetos (recepción)
- 6.- Retrasar la transmisión un determinado tiempo entre tramas.
- 7.- Detener la transmisión cuando se detecta colisión.
- 8.- Gestión de las retransmisiones tras detectar la colisión.
- 9.- Generar mensajes de atascos cuando se superan un determinado nº de colisiones.
- 10.- Descartar tramas que no tiene longitud mínima.
- 11.- Conformar la trama en transmisión. (Llenar todos los campos de la trama).
- 12.- Extraer o quitar los campos de la trama recibida.

802.4 Paso de testigo en BUS (Token Bus)

Topología en BUS.

Medio de transmisión en banda ancha.

Protocolo libre de colisión gracias a un testigo.

La estación que posea el testigo será la que pueda transmitir.

El testigo ira pasando de estación en estación permitiendo que transmitan todas las estaciones.

Para evitar que una estación monopolice el medio de transmisión, lo que se hace es limitar el tiempo máximo de posesión del testigo por cada estación.

Esto hace que tengamos un tiempo a atención establecido, el cual será el tiempo máximo de posesión del testigo por el numero de estaciones.

Con esto, hace que el protocolo pueda trabajar en tiempo real.

Al tener un tiempo máximo de atención podremos detectar perdidas de testigo y por lo tanto podremos solucionarlo.

Esta red no sigue un esquema centralizado, o lo que es lo mismo una estación no controla el testigo.

Es topología física en bus, pero el testigo funciona en anillo.

Este protocolo tiene buenos rendimientos con trafico alto; mientras que con trafico bajo el rendimiento es bajo.

$$\text{Rendimiento} = \frac{\text{Bits transmitidos de información (bits útiles)}}{\text{Bits totales Tx}}$$

Tráficos altos:

Se transmiten tantos bloques de testigos como bloques de datos, o incluso mas de datos que de testigos.

TESTIGO	DATOS
---------	-------

Tráficos bajos:

En tráfico bajos el numero de testigos es mayor que el numero de datos.

TESTIGO	TESTIGO	DATOS	TESTIGO	TESTIGO
---------	---------	-------	---------	---------

Este protocolo lo adopto la GM con la intención de conectar no solo ordenadores, sino también otro dispositivos.

Características del control de acceso al medio por paso de testigo.

Tendremos un testigo o Token, gracias al cual, el que posea el testigo será el que tenga acceso al medio.

El testigo va a ir pasando de estación en estación formando un anillo lógico, esto implica que tendremos un tiempo máximo de posesión del testigo para cada estación.

En este tipo de protocolo vamos a tener 2 estados; ya que el medio va a estar siempre ocupado, o bien transmite información, o bien transmite el testigo.

El mantenimiento del anillo se lleva a cabo en cada estación; lo que hace que el sistema no sea centralizado, y por lo tanto que si tenemos el problema de que una estación falla, la red no se estropea.

Al ser un protocolo de banda ancha normalmente será un cable coaxial de 75Ohm, al ser de banda ancha la información va a ser de tipo analógica.

Este protocolo también está implementado en banda base aun que no es lo normal.

Frecuencia alta-> 1

Frecuencia baja-> 0

Fase continua

Se utiliza una modulación en frecuencia con codificación Manchester.

Existen 2 tipos de modulación en frecuencia una en la que el número de ciclo no es entero por tiempo de bit y otro es con número entero de ciclo por tiempo de bit.

Existen de 2 tipos de modulación:

Modulación en frecuencia coherente => número entero de ciclos.

Modulación en frecuencia no coherente => el número de ciclos no tiene por que ser un número entero.

Otra forma es la de multinivel, o lo que es lo mismo un rango de frecuencias para cada cosa; un rango para emitir y otro para recibir mas un tramo de frecuencias de salvaguarda.

Modulación QAM -> Mod. en amplitud y fase (esto también se utiliza en la transmisión)

Cuando se utiliza fibra óptica, lo que se hace es poner una fibra para emitir o otra para recibir.

SCRAMBLING

Cuando se transmite una serie de bits, por ejemplo el testigo, se pueden crear problemas de sincronismo entre estaciones, para ello lo que se hace es crear una mayor aleatoriedad en la secuencia a transmitir (secuencia de 0 y 1).

El scrambling lo que hace es añadir 0 o 1 de forma aleatoria en la secuencia a transmitir, esto lo hace el emisor, mientras que en el receptor se hace el trabajo contrario, es decir eliminar los 0 y 1 añadidos en el emisor.

EMISOR:

(falta dibujo)

An

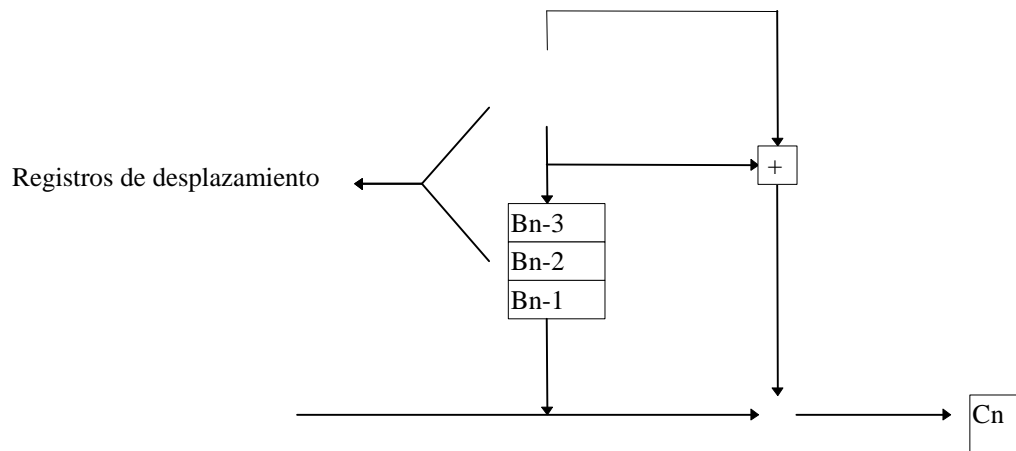
Bn

Bn=An

Bn-3

Bn-5

RECEPTOR:



Bn

Cn=Bn

Bn-3

Bn-5

Ejemplo:

An=00011111000

Bn=11000111000

Cn=00011111000

802.4 SubCapa MAC:

Funciones:

- 1.- Temporizar la perdida de testigo (tiempo de atención).
- 2.- Inicialización distribuida: Cualquier estación a de ser capaz de inicial el anillo lógico.
- 3.- Temporizar la retención del testigo: tiempo limitado de posición del testigo.
- 4.- Buffer de datos: Guardar los datos pendientes de transmitir o de ser atendidos por las capas superiores.
- 5.- Reconocimiento de la ADR de nodo: Verificar que las direcciones destino son las de la maquina.
- 6.- Encapsulado / Desencapsulado de la trama: Construcción de la trama en el formato establecido.
- 7.- Gestión de chequeos de trama: Corrección y detección de errores con CRC.
- 8.- Comprobación testigo válido.
- 9.- Añadir nuevos miembros al anillo: Cuando se quiere añadir una nueva estación, lo que hay que hacer es indicar a las estaciones donde esta, es decir darla de alta.
- 10.- Recuperación de errores de nodo: En caso de un error en un nodo hay que darle de baja para que el testigo no se pierda.

Primitivas del 802.4

Son las mismas que para 802.3 (tienen la misma estructura)

MA_DATA.request
MA_DATA.indication
MA_DATA.confirm

VISIÓN DE LA SUBCAPA MAC 802.4

Como máquinas lógicas asincronas.

IMF:

Gestiona la comunicación con la capa superior, se encarga de la gestión de primitivas.
Tendremos implementado un buffer para los datos que llegan de la capa LLC y los de la capa física.

ACM:

Funcionamiento del paso en testigo en BUS.
Gestiona el testigo.
Tendrá implementado las funciones del anillo lógico, al igual que la detección y corrección de errores de la gestión del anillo lógico.

TxM:

Gestiona las tramas a transmitir.
Tendrá implementada el encapsulado de la información de trama en 802.4
Añade el CRC a la trama.

RxM:

Hace lo contrario que TxM.
Verifica la integridad de datos de la trama.

RRM:

Es opcional se encarga de generar las tramas.

Cuando la capa superior quiere enviar un dato, la primitiva llegara a (IMF), lo cual se lo comunica a ACM, la cual da la orden a TxM de generar la trama, la cual es o bien enviada a la capa física o bien a RRM, la cual lo manda a la capa física.

En recepción la RRM, lo que hace es modular la señal analógica(opcional) a señal digital, para así pasarla a RxM, la cual extrae los datos de la trama, ahora tenemos 2 opciones según sea el tipo de trama recibido (datos o control)

Si la trama es de control los datos van a ACM; si por el contrario la trama es de datos, la RxM se conecta a IMF, mandando también algo de información a ACM, ya que ahora hay información de control que puede interesarnos.

Con los datos que nos llegan a IMF, los mandaremos a las capas superiores.
Si el RRM no existe la modulación y demodulación lo harán la RxM y la TxM.

FORMATO DE TRAMA EN 802.4

Nº octetos	
>= 10	PREÁMBULO
1	DELIMITADOR DE COMIENZO (SD)
1	CONTROL DE TRAMA (FC)
2 ó 6	ADR DESTINO (DA)
2 ó 6	ADR ORIGEN (SA)
>= 0	UNIDAD DE DATOS
4	SECUENCIA CHEQUEO TRAMA (FDS)
1	DELIMITADOR FINAL (ED)

→

↓

LSB

MSB

El orden de transmisión es de LSB a MSB.

Preámbulo:

Permite ajustar el reloj de recepción y los niveles de la señal (los niveles de recepción son por la atenuación de la señal en el medio de transmisión).

El medio de transmisión siempre esáa ocupado o bien por el testigo o bien o por datos.

El preámbulo lo que hace es asegurarnos un tiempo mínimo entre el delimitador final de la trama recibida y el inicio de la siguiente para permitir a las estaciones procesar la trama recibida; este tiempo es del orden de las 2 seg.

El numero de octetos depende de la velocidad del medio de transmisión a mas rápida mas octetos, y a mas lenta menos octetos.

$$\text{Velocidad de transmision} = \frac{\text{Mb}}{\text{Seg.}}$$

El tiempo de transmisión vendrá dado por:

$$T = \frac{\text{nº de bits}}{\text{Velocidad de Tx (b/seg.)}}$$

$$2 \cdot 10^{-6} = \frac{\text{nº de bits}}{\text{Velocidad de Tx} \cdot 10^6}$$

nº bits = 2 · vel Tx (Mb/s) numero de bits en el preámbulo.

nº de bits

nº de octetos = $\frac{\text{longitud}}{8}$ con redondeo hacia arriba

Ejemplo:

2,1 => 3 octetos.

DELIMITADOR DE COMIENZO:

Este formado por una secuencia que se pueda diferenciar siempre de los datos.

Normalmente la técnica para esto es emplear una violación en la codificación (así no impedimos la transmisión de secuencias de datos).

Ejemplo:

Manchester:

Con violación

Esto nos permite disponer de todas las combinaciones de datos, sin posibilidad de mala interpretación.

Campo de control de trama:

Nos indica el tipo de trama de que se trata.

Existen los siguientes tipos de clases de trama.

1.- MAC control

Solución de errores.

Añadir estaciones, etc...

Los bits de 3 al 8 -> Nos indica el tipo de control de trama del que se trata.

2.- Datos LLC

Información para la capa de control lógico de enlace.

3.- Dato de mantenimiento de estación.

Información para que intercambiaría la capa física con la capa MAC

Ejemplo:

Estado de la electrónica.

Información que no va por la red

4.- Propósitos especiales.

Usos especiales.

Para funcionalidades no contempladas anteriormente.

Del octeto cada bit indica.

1,2 -> tipo de trama.

Para los tipos de trama 2,3,4.

3,4,5 -> Servicio MAC solicitado.

6,7,8-> Información de la prioridad de la trama.

ADR origen y destino:

Las direcciones de origen y destino tienen un tamaño de 2 ó 6 octetos en el mismo formato que en 802.3

El tamaño a de ser el mismo para las 2 direcciones.

Gracias a que el formato es el mismo (802.3 y 802.4); se pueden pasar tramas de una red 802.3 a una 802.4

o viceversa.

Para juntar una red 802.4 y otra 802.3 lo que se pone es un switch el cual desmonta la trama que le llega y la reconstruye para el otro formato.

Unidad de datos:

Contendrá la información que viene de capas superiores.

Chequeo de trama:

Tiene una longitud de 4 octetos.

Se hace cargo de chequear la información de la trama desde el control de trama hasta chequeo de trama.

Delimitador final:

Tiene un tamaño de 1 octeto.

Es una secuencia de bits que no puede estar en la unidad de datos; para eso lo que se hace es lo mismo que para la trama de inicio de trama, y es violar el código (Mirar delimitador de inicio de trama).

Formato del testigo en 802.4:

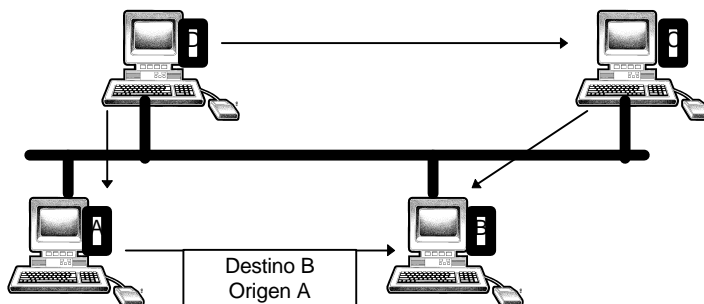
Será una trama de control de la subcapa MAC.

Existen 2 tipos de testigos:

Testigo normal: Cuando la red funciona correctamente.

≥ 1	Preámbulo	
1	Delimitador de comienzo de trama	
1	FC 00001000	→ <i>Identificador del testigo en 802.4</i>
2 ó 6	ADR destino	
2 ó 6	ADR origen	
4	FCS	
1	Delimitador de final de trama	

En dirección destino nosotros tendremos la dirección física de donde va a ir el testigo y en origen nuestra dirección.



Testigo inicio: Se usa para poner la red en marcha.

>=1	Preámbulo	
1	Delimitador de comienzo de trama	
1	FC 00000000	Identificador de testigo de inicio
2 ó 6	ADR destino	
2 ó 6	ADR origen	
variable	Unidad de datos	
4	FCS	
1	Delimitador de final de trama	

Si solo hay una estación en la red, el testigo se lo mandara a si mismo.

Problemas en 802.4

Añadir estaciones al anillo.

Cada estación llega su identificación, que lo sitúa es una posición en el anillo lógico.

Ahora suponemos que la estación 40 quiere incorporarse al anillo; en un principio la estación 40 no podrá nunca transmitir, ya que nunca le llegara el testigo.

Cada estación tiene incorporada la gestión de añadir o eliminar estaciones a la red, este programa se activa cada X veces que recibe el testigo.

El programa lo que hace es buscar nuevas estaciones, cuando la encuentra, mira su identificador y hace una cosa u otra según las siguientes condiciones:

Estación con ID > del ID mas bajo.

Genera la trama:

solicit_successor:

Con el campo FC 00000001

Esto permite incorporar estaciones donde el identificador este entre la emisora y la siguiente.

Ejemplo:

Si la trama la genera la estación 108, se podrán conectar las estaciones de identificación desde la 109 a la 120.

La zona de datos de esta trama esta vacía con la intención de que en ella, las estaciones nuevas, puedan transmitir una señal de aviso comunicando que se quieran incorporar a la red.

P	D	F	A	A		F	E
R	C	C	D	D		C	D
E						S	

Después de transmitir dicha trama pueden darse los siguientes casos:

1.- No hay respuesta.

Pasar el testigo a la siguiente estación.

2.- Una respuesta.

Emitir una trama del tipo set_successor después lanzar un proceso de enlace.

Se encarga de comunicarle a la nueva cual se su sucesora y antecesor y también modificarse su sucesora y comunicar a su exsucesora cual es su antecesora.

La secuencia para insertar una estación será:

Mandar trama Solicit_Successor.

Mandar la trama Set_Successor.

Comunicar a la nueva cual es su sucesor y su antecesor.

La vieja sucesora ve que su antecesora o mandado un Set_Successor y se actualiza.

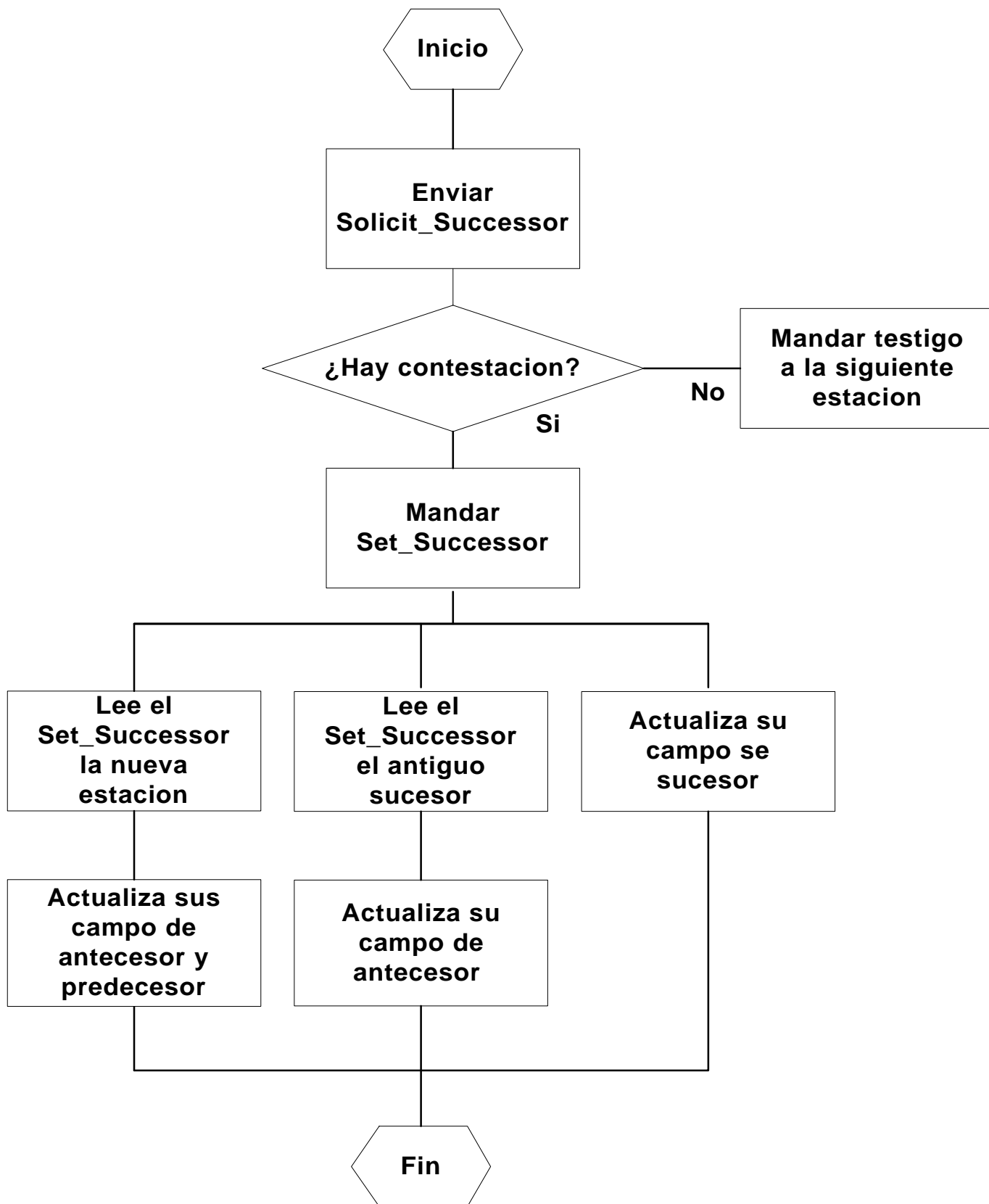
Actualizarse a si mismo cual es su sucesora.

Una trama Set_Successor será por ejemplo:

ADR destino: 40

ADR Origen: 37

Sucesor: 37:43



3.- Varias Respuestas:

Cuando la estación 23 tenga el testigo emitirá la trama Solicit_Successor entonces contestaran la 48 y 53 generando una colisión.

Por lo que 23 detectara que hay varias estaciones y generara una trama de resolución de contienda, la cual en la zona de datos tendrá 4 zonas de respuesta; para que así cada estación contestara en la posición que indiquen sus 2 bits de menor peso.

Trama de resolución de contienda

Con esta trama de contienda gana la estación que conteste antes.

La que conteste primero será la estación que se añada a la red, y las demás estación tendrán que esperar a que se vuelva a buscar nuevas estaciones.

Si se diera el caso de que coincidieran en la ventana de contienda, la estación repetiría el proceso de resolución de contienda con las que tengan colisión pero con los 2 siguientes bits.

Si la colisión por ejemplo se produce en 01 y hay una estación en 00, solo se hace caso a la de 00.

Si fuera al contrario, es decir colisión en 00 y una estación en 01, cuando se repita el proceso de colisión, solo entrarán las de 00 dejando para la siguiente vuelta las de 01; en esta segunda caso se tomarán los bits 3 y 4 de menor peso.

La trama de contestación se llama Resolve_contention (FC:00000100)

Cuando la dirección de la estación a incorporar es menor que el menor de los actuales.

Lo que ocurre es que la de menor ID mandará una trama Solicit_Successor_2 (FC:00000010) en vez de Solicit_Successor que la mandarán las demás estaciones.

La peculiaridad de esta trama es la de tener 2 ventanas de respuesta, ya que tiene 2 casos posibles:

Incorporar una estación con ID menor que el menor.

Incorporar una estación con ID mayor que el menor.

Por lo tanto las estaciones con menor ID contestarán en la primera ventana, y el resto en la segunda.

Si tenemos solo una estación, la estación se añadirá sin problemas.

Si por el contrario tenemos más de una estación, existirá colisión, y se solucionará como hemos indicado anteriormente.

En la segunda ventana, como ya hemos dicho, contestarán las de ID mayor que el ID menor, siempre y cuando no existan estación con un ID menor que el menor.

La incorporación de estaciones solo se realiza de 1 en 1.

Cuando se ha incorporado una estación el testigo pasa a la siguiente estación y por lo tanto la incorporación de otras estaciones, para cuando se vuelva a generar la trama Solicit_Successor.

ELIMINAR ESTACIONES DEL ANILLO:

Tenemos 2 formas de que una estación se de baja de la red, y son las siguientes:

Darse de baja de forma controlada dando las informaciones pertinentes.

Darse de baja de forma descontrolada: Por avería o apagado incorrecto, por ejemplo.

Baja controlada:

En el momento que coge el testigo, lo que hace es emitir una trama Set_Successor a su estación antecesora, indicando cuál es su nueva sucesora, gracias a la cual actualizará su campo de sucesora.

Si se da de baja la 53 la trama Set_Sucesor va destinada a la 23 indicando que su sucesora a partir de ahora será la 61.

Ahora bien para que se entere la nueva sucesora (61) tenemos 2 casos:

Que haya escuchado la trama Set_Successor y actualice su campo de anterior.

O bien espera a que la nueva antecesora suya (53) le mande el testigo para actualizarse.

Baja descontrolada:

Ahora suponemos que la 53 se da de baja de forma descontrolada. Cuando la 23 pasa el testigo a la 53, este se pierde y por lo tanto el canal se queda libre; gracias a lo cual se detecta que una estación ha caído. En vista de que el canal ha quedado libre la estación 23 vuelve a emitir el testigo, por si se da el caso de que el testigo

se halla estropeado por el camino.

Si el testigo vuelve a desaparecer, es decir sigue sin escuchar información, asume que su estación sucesora a fallado y emite una trama de control del tipo Who_Follows (FC:00000011).

Esta trama contiene información de su sucesora, en este caso la 53, y para que así la 63 vea que la que a caído es su antecesora y por lo tanto actualiza sus datos y se lo comunica a la estación 23, la cual mandara la trama Set_Successor a la 63.

En el caso de que no se reciba información la segunda vez que se transmita la trama Who_Follows, se retransmite otra la trama Who_Follows, pero esta vez preguntando si hay cual es la siguiente de ¿?, es decir preguntando si hay alguien en la red. Cuando conteste una estación, lo que hace es generar una red de 2 estaciones, a la cual se irán añadiendo el resto de las estaciones que no han caído.

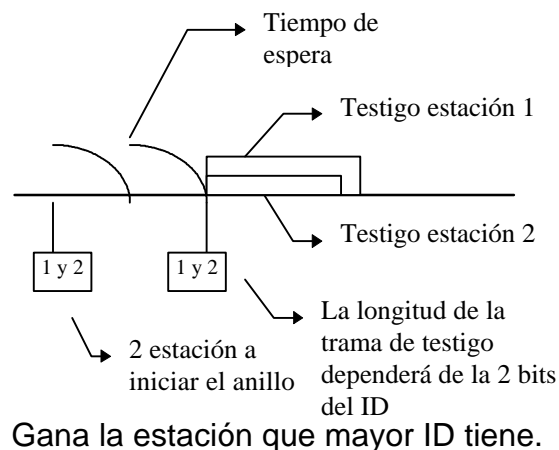
INICIALIZACIÓN DEL ANILLO LÓGICO:

Cuando una estación quiere incorporarse al anillo, antes escucha si el medio de transmisión esta libre o ocupado, si se da el segundo caso, la estación esperara al proceso de inserción de una estación en el anillo; si el caso fuera el primero, es decir, el canal libre, esperara un tiempo determinado para volver a escuchar el canal, ya que puede darse el caso, de que estuviera escuchando una de las ventanas de las tramas de control.

Si después de escuchar 2 vez, sigue sin escuchar nada en el medio de transmisión, pasara a iniciar el proceso de inicialización del anillo.

Este proceso consiste en enviar una trama de testigo de inicio (Clain_Token (FC:00000000)); la zona de datos de este testigo tendrá una longitud determinado, dependiendo de 2 bits del ID, si en la primera vez que se transmite, será las 2 bits de menor peso. Las longitudes posibles de la zona de datos será 0 para los 2 bits a 0, 2 para los bits a 01, 4 para los bits de 10 y 6 con los 2 bits a 1.

Después de transmitir la trama de testigo, la estación se pone a la escucha, si detecta que hay alguien transmitiendo interpreta que hay una contienda por iniciar el anillo, es decir que hay varias estaciones que quieren iniciar el anillo.



Cuando termina de transmitir el testigo, escucha el canal, si este esta libre y continua transmitido todas las tramas que le falten hasta que transmita tantas tramas con pares de bits tiene su ID; si por el contrario el canal no esta libre, deja de transmitir, ya que la otra estación tiene mas prioridad que ella.

La estación tiene que transmitir tantas tramas de testigo como pares de bits tiene la longitud del ID.

Ejemplo:

Suponemos que las direcciones son de 2 octetos y suponemos también que hay 3 estaciones que quieren iniciar el anillo al mismo tiempo.

Los ID de las estaciones son los siguientes:

+

-

```

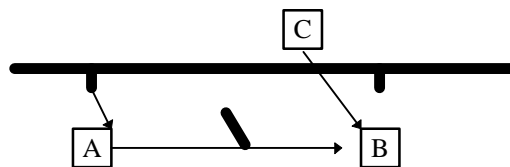
11 11 11 11 11 11 11 11
10 11 11 11 11 11 11 11
11 11 01 11 11 11 11 11

```

Las 3 estaciones comenzaran a transmitir un testigo de longitud 6, ya que los 2 bits de menor peso su ID es 11. Después de transmitir las tramas, las 3 se pondrán a la escucha, como ninguna escuchara nada, continuaran con los siguientes 2 bits de menor peso, en este caso implican en las 3 estaciones una longitud de 6; esto se repetirá en todos los pares de bits hasta el par 6, en el que la estación 3 tendrá una trama de testigo con una longitud de 2 y las otras 2 estaciones tendrán una longitud de datos en sus testigo de 6; por este hecho, la estación 3 dejara contienda dejando a las estación 1 y 2 que vuelvan a transmitir.

Ahora la estaciones 1 y 2 solamente serán las que transmiten las tramas correspondientes a los bits de par 7, lo que implica que su longitud de datos en sus tramas de testigo será de 6; después de escuchar a ver si hay alguien mas en la red y no escuchar nada, continuaran con los bits de par 8, en los cuales, la estación 1 transmite con una longitud de datos de 6 y la estación 2 con una longitud de datos de 4, lo que hace que la estación 1 inicie el testigo estándar.

RECUPERACIÓN DE TESTIGO (Por perdida o daño)



La estación A transmite el testigo a la B si el testigo no llega a su destino o llega erróneo, la estación A vuelve a transmitirlo, si este tampoco llega a su destino, el procedimiento será el de caída de una estación de forma no controlada.

Otro problema con el testigo, es que tengamos mas de 1 en la red:

Si una estación que tiene el testigo escucha que alguien esta transmitiendo, quiere decir que hay mas de un testigo, con lo que decide desechar el suyo y quedarse a la escucha.

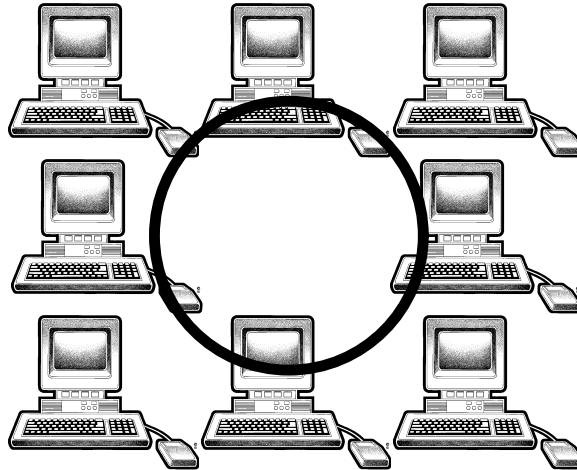
La existencia de mas de un testigo, puede ser causada por el ruido, ya que si por ejemplo, una trama con un FC determinado, por causa del ruido, pasa a tener un FC igual que el del testigo tendremos 2 testigo.

802.5:

Este protocolo es un protocolo libre de colisión, ya que funciona por paso de testigo (Token Ring).

La filosofía del 802.5 es similar al 802.4.

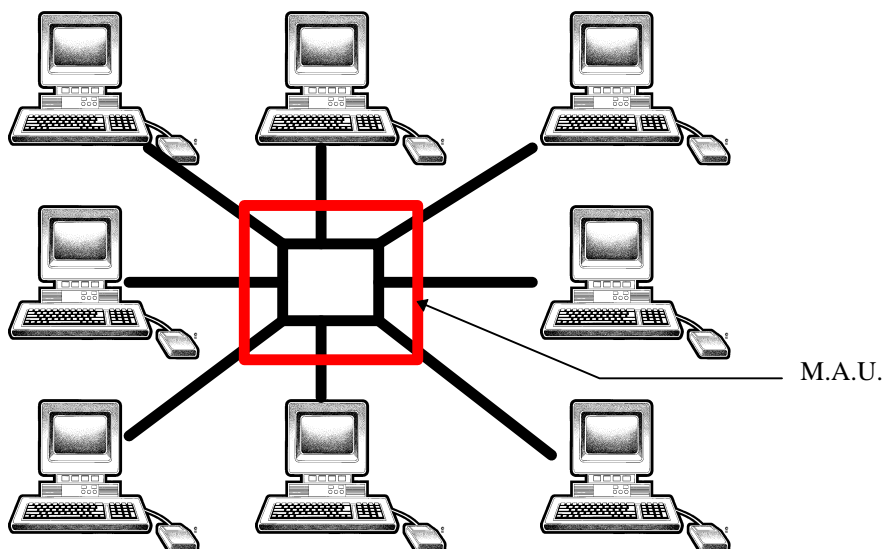
La principal característica es que aquí si que existe un anillo físico con las estaciones, es decir la topología física es en anillo.



Este protocolo es un protocolo más abierto que los anteriores, es decir deja a los fabricantes mas libertad de movimiento.

Este protocolo es el más utilizado por I.B.M.

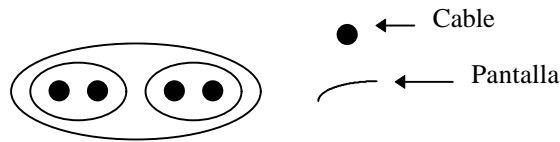
La conexión de esta red se realiza entorno a una M.A.U. (Unidad de acceso múltiple); por lo que la topología de la red es como se muestra en la figura:



La M.A.U. lo que hace es que si una estación se cae, esta reconstruye el anillo. El cableado utilizado es el STP (par trenzado apantallado por pares y en total).

Cables de I.B.M.

Tipo 1: Consiste en 2 pares trenzados con cable rígido y los pares se encuentran apantallados cada par.



Presenta una impedancia característica de 150 Ohm y nos permite distancias de 200 metros funcionando a 4 Mb/s o bien 100 metros funcionando a 16 Mb/s.

Tipo 2: Es el tipo 1 pero añadiendo 2 pares sin apantallar.
Estos 2 nuevos pares sin apantallar se utilizan para telefonía.

Tipo 3: Esta compuesto por 4 pares sin apantallar.
Se corresponde con los cables de categoría 3 y 5 del 802.3 (Ethernet).

Tipo 5: Esta compuesto por 2 fibras ópticas.

Tipo 6: 2 pares apantallados con cable flexible nos permite una longitud máxima. de 30m.

Tipo 8: 2 pares apantallados (STP), estos cables tiene la característica de ser planos.

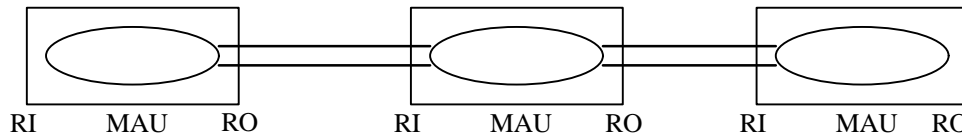
Tipo 9: 2 pares apantallados (STP) con un recubrimiento resistente al fuego.

(Siempre son 2 pares porque uno se utiliza para emitir y otro para recibir.)

Si nosotros queremos tener mas estaciones que las que admite la M.A.U., solo tenemos que poner otra M.A.U. juntando las entradas RI y RO de cada una de ellas.

La forma de conexión es la siguiente:

RI con RO.



La velocidad de transmisión de este estándar es de 4 Mb/s o de 16Mb/s.

En este estándar se utiliza una codificación Manchester diferencial en banda base.

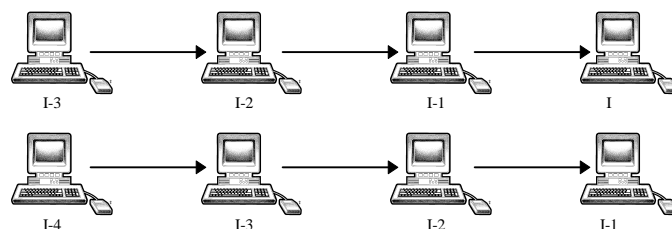
Tenemos un tiempo máximo de atención a la estación, ya que tenemos un tiempo máximo de posesión del testigo y un numero máximo de estaciones.

El testigo estará compuesto por una serie de bits determinados. Lo que hacemos será violar la codificación Manchester diferencial.

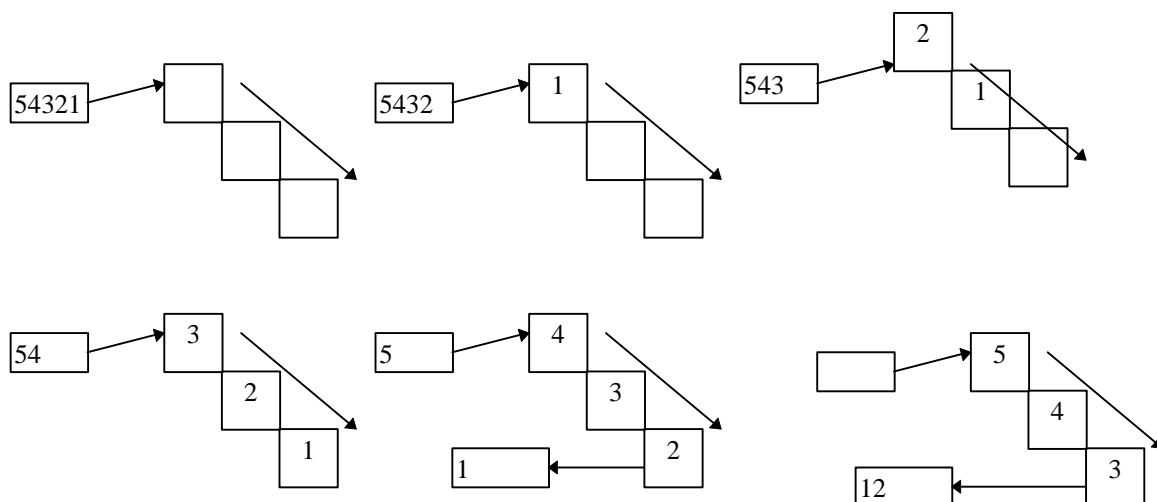
Otra cualidad de este protocolo es que primero se transmiten los bits de mayor peso y después las de menor peso.

No tenemos long. máximo de trama, por lo que una estación puede estar transmitiendo hasta que se le termine el tiempo de posesión del testigo. Esto permite distintas velocidades de transmisión.

La información va en serie bit a bit, cada bit ira de estación en estación; este bit se ira almacenando en una memoria temporal de 1 bit.



Cada estación puede analizar e incluso modificar un bits determinado, esto hace que tengamos un bit de retraso por cada estación; este mecanismo permite regenerar la señal para cada estación.



Todas las estaciones tiene que sacar una copia de la trama hasta que completa la dirección destino, cuando ya tiene dicha dirección sigue retransmitiendo, pero sin sacar una copia para sí.

El anillo debe tener un numero de retardos suficientes para poder componer la secuencia de bits del testigo completamente dentro del anillo.

Es decir si el testigo esta compuesto por 4 bits, nos harán falta o bien 4 estaciones que harían 4 retardos, o menos estación con lo que hará falta tener una estación monitor, la cual creará los retardos que faltan.

Ejemplo:

Bits de testigo = 8
Numero de estaciones = 5

Como el testigo no cabe en las 5 estaciones una de ellas, denominada estación monitor, generará los 3 retardos que faltan para que quepa el testigo.

Para evitar que una estación monopolice el testigo, las estaciones tiene un contador de veces consecutivas de posesión del testigo, cuando el contador se rebasa, la estación tiene que esperar.

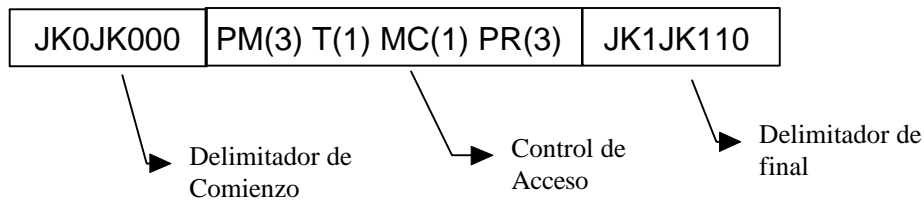
También se la puede poner prioridad a las estaciones, gracias a la cual varias estaciones tendrán mas prioridad sobre otras a la hora de tener el testigo.

Funciones de la SUBCAPA MAC en 802.5

- 1.- Identificación de ADR propia, grupo o broadcast.
- 2.- Copia de la trama y reconocimiento de esta.
- 3.- Generación de tramas en el formato establecido.
- 4.- Manejo de prioridades.

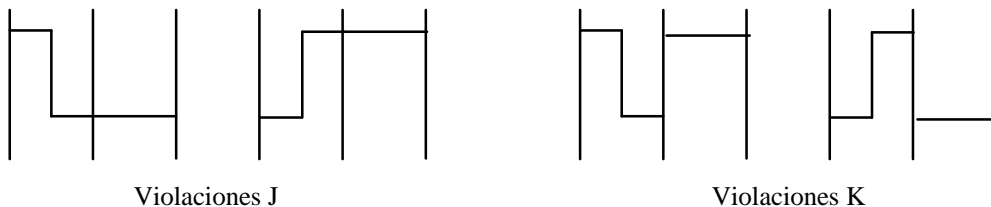
- 5.- Encaminamiento de trama.
- 6.- Temporizaciones (Tiempo máximo de posesión del testigo // Tiempo máximo de atención (por pérdida de testigo).
- 7.- Manejo del testigo

Formato del testigo en 802.5



JK son violaciones de código.

J nos mantiene la polaridad del signo anterior y con k nos la cambia.



Campo de control de acceso:

Esta compuesto por:

3 bits llamados PM: Nos indica la prioridad del testigo.

Solo podrá transmitir una estación si le llega un testigo de prioridad igual o inferior que el que tiene determinada la estación.

Ejemplo:

Prioridad de la estación = 4

Solo podrá coger el testigo para transmitir si la prioridad del testigo es ≤ 4 .

1 bit llamado T: Es el bit de testigo. Nos indica si la trama es el testigo o es una trama de datos.

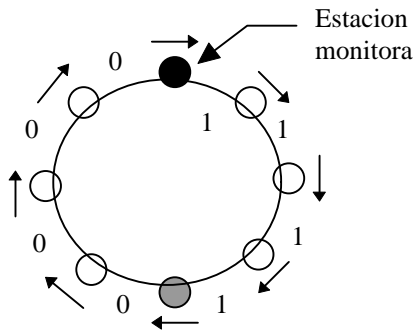
0 -> Es una trama de testigo

1 -> Es una trama de dato

1 bit llamado MC: Bit de monitor.

Tiene 2 cometidos según sean tramas de testigo o de datos.

Suponemos la red de la siguiente forma:



En el momento que una estación cualquiera se pone a transmitir el bit MC esta a 0, y cuando llega el testigo a la estación monitor esta cambia este bit a 1.

Si ninguna estación transmite el testigo llegara a la estación monitora con un 1 en MC.

Si llega a la estación monitor el bit MC a 1 lo que hace es bajar la prioridad del testigo, ya que lo que indica es que ninguna estación a podido apoderarse del testigo.

La prioridad del testigo pone la estación que la transmite.

A cada vuelta la prioridad baja en una unidad.

Si esta en prioridad mínima sigue dando vueltas sin tocar el prioridad.

3 bits (PR):

Reserva de testigo.

Se utilizan en las tramas de datos, en el testigo están solo para mantener el formato de trama.

Modo de funcionamiento en 802.5

Existen 2 modos de funcionamiento que son:

-Modo repetición:

Tenemos que los bits recibidos son retransmitidos con 1 bit de retraso.

En esta situación están las estaciones que están a la escucha.



- Modo transmisión:

Se activa una vez tiene datos que transmitir y a capturado el testigo.

Cuando esta en este caso, lo que hace es abrir el anillo, y ponerse a transmitir.

Cuando a terminado de transmitir, lo que hace es esperar a que le retornen todos los datos para cerrar de nuevo el anillo.

Cuando a retirado todos los bits de su trama pasa a modo repetición.

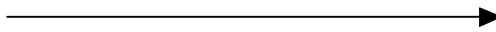
Cuando a terminado la trama manda el testigo y pasa a repetición.

Proceso de captura del testigo:

Si la estación tiene una prioridad inferior pasa el testigo, sino se lo queda y se pone a transmitir.

FORMATO DE TRAMA DE DATOS EN 802.5

1	Delimitador de comienzo	(SD)
1	Control de acceso	(AC)
1	Control de trama	(FD)
2 ó 6	Dirección destino	(DA)
2 ó 6	Dirección origen	(SA)
>0	Información	
4	Secuencia de chequeo de trama	(FCS)
1	Delimitador final	(DE)
1	Estado de trama	(FS)

MSB  LSB

El Delimitador de comienzo de trama es la misma secuencia que en el testigo.

El control de acceso tiene el mismo formato que en el testigo:

JKOJK000 PM(3) T(1) MC(1) PR(3) JK1JK110

PM es el mismo que en el testigo (copia de la prioridad del testigo)

T se pone a 1 por ser datos.

Nos evita tener tramas huérfanas.

Esto es que si una estación transmite una trama y se cae, la única que podrá sacarla será la monitor si este esta a 1

Si una trama de datos con un 1 en este bit pasa por la monitor, es retirada y se reemite un testigo.

Si se diera el caso de que una trama estuviera partido, la monitor se da cuenta de que la trama no esta completa y la retira.

MC tiene otro cometido, evita que existan tramas huérfanas.

Si una estación esta en modo transmisión y se cae antes retirar su trama de datos, entonces no se ha podido emitir el testigo con lo que se quedaría dando vueltas y no podría transmitir otra. La estación emite con el bit MC a 0, cuando la trama pase por la estación monitora, esta pondrá MC a 1 si esta trama vuelve a la estación monitor, la estación monitora la retirara.

PR: Permiten reservar prioridad, de modo que cuando la trama circula por el anillo se pueda reservar el uso del anillo poniendo su prioridad en ese campo, poniendo ahí su prioridad siempre y cuando no haya un valor de prioridad mas alta puesta en PR, para que esto funcione correctamente es por lo que se transmite de MSB y LSB ya que así le llegara el bit de mas peso primero.

CONTROL DE TRAMA

FF(2) R(3) CONTROL(3)

FF: Formato de trama de 2 bits y tiene siguientes valores:

- 00 información relativa a la capa MAC
- 01 trama dirigida a capas superiores (LLC PDV)
- 10 Reservado
- 11 Reservado

Control: si estamos con FF a 01 nos indica la prioridad de la primitiva.

Los campos ADR origen/destino puede tener un tamaño de 2 ó 6 pero para los 2 igual.

El formato es el mismo en 802.3, 802.4 y 802.5.

La única diferencia es el orden de transmisión de los bits en 802.3, 802.4 es de menor a mayor y en 802.5 es de mayor a menor.

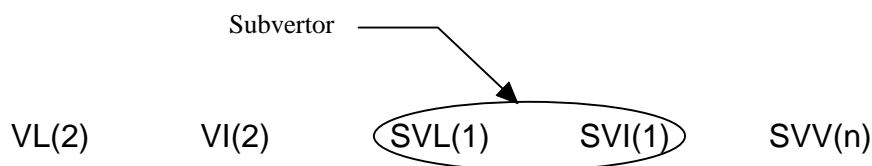
INFORMACIÓN:

Tendrá un formato específico llamado “vector”

La intención de este formato es mandar en una sola trama varios mensajes.

Se podrán mandar tantos mensajes como nos de tiempo en el tiempo de transmisión.

La estructura del vector es la siguiente:



El valor representado entre paréntesis son el número de octetos que lo forman.

VL tiene un tamaño de 2 octetos.

Nos indica la longitud del vector.

El valor de VL estará comprendido entre 4 y 65535 octetos.

VI Identificación del vector.

Nos permite distinguir distintos vectores.

Le permite ordenar las tramas según dicho identificador.

Estos 2 campos son fijos en todas las tramas.

Si ahora queremos transmitir datos añadiremos los siguientes campos por cada bloque de datos.

SVL -> Longitud del subvector.

SVI -> Identificación del subvector.

Si nosotros queremos mandar más de 254 octetos, lo que haremos será poner SVL a FF y a continuación pondremos 2 octetos para poner la longitud, si SVL no es FF no existirán estos 2 octetos.

Si el subvector es de 254 SVL será FE y si el subvector es de 255 octetos de largo, SVL será FF+00+FF (siendo estos 2 últimos pares la ampliación).

SVV es la información.

Ejemplo:

Mensaje 1 de longitud 125d.

Mensaje 2 de longitud 300d.

435		127		Mensaje 1	FF	304		Mensaje 2
VL	VI	SVL	SVI	SVV	SVL		SVI	

El SVI se pone a FF si el tamaño es de más de 1 octeto.

ED (dirección destino) se produce mediante una violación de código.

Se hace mediante la trama JK1JK11E, siendo E el bit de error y el resto son bits de violación.

Si alguna estación detecta que ha habido un error en la trama activa el bit E y todas las estaciones que reciben la trama con E activado, la desechan, si está a 0 la trama está bien.

CRC:

Verifica desde el control de acceso hasta la secuencia de chequeo de trama, no incluye el delimitador de comienzo, el delimitador de final y el estado de trama.

ESTADO DE TRAMA:

Presenta los siguientes valores ACrrACrr el cometido es realizar operaciones de asentamiento de la trama. La información esta contenida en los bits A y C, siendo las r bits reservados, y los segundos AC son copia de los primeros.

Se realiza una copia ya que al estar fuera de la secuencia de chequeo, no se pueden detectar errores, y es por ello que se duplican, para tener seguridad de que el octeto es valido mediante una comprobación de redundancia.

A: Reconocimiento de dirección:

Este bit lo activa la estación destinataria de la información en el momento que reconoce como propia la dirección de destino.

Si el bit esta a 0 cuando llega a la estación de origen, esta sabe que no ha llegado bien la información.

C: Bit de copia

Se activa en el momento que la estación de destino copia la trama, cuando va leyendo va copiando en un buffer, y así la estación origen constancia de que la destino ha recibido la información.

Se puede el caso de que una estación este activa, pero tenga el buffer lleno y ya no pueda recibir mas información, entonces la estación pondrá A a 1 y C a 0, entonces la emisor retendrá el envío de información, evitando el colapso de las estaciones.

COMETIDO DE LA ESTACIÓN MONITORA:

Puede ser cualquier estación de la red, de modo que solo habrá uno, pero todas las demás quedan como monitoras de reserva, y si se desconecta la monitora, cualquier otra puede tomar el rango de monitora.

Los cometidos van a ser los siguientes:

-Mantenimiento del reloj:

Se ocupa de mantener una señal de reloj que permita mantener el sincronismo con las demás estaciones y además se asegura que todas las estaciones que forman parte del anillo se encuentren sincronizadas con esta señal.

La estructura de sincronismo vendrá dada por la implementación Manchester.

Asegurar el tiempo de retardo en el anillo necesario, el anillo debe de presentar un retardo suficiente para que el testigo este completo en el anillo, cada estación tiene un retardo de 1 bits, si tenemos un bit y una V de valor:

$$V = \frac{xMb}{sg} = \frac{x \cdot 10^6}{sg} = \frac{1bit}{\frac{x \cdot 10^6}{seg}} = \frac{10^{-6}}{v} seg$$

El retardo introducido por una cada estación es:

$$t_{bit} = \frac{10^{-6}}{V} seg.$$

Si no tenemos un n° de estaciones suficiente la estación monitora introducirá el numero de retardo míni-

mos para que el testigo entre completo en el anillo.

Notificación de vecindad: Cada cierto tiempo la estación monitora va a generar una trama que va a ir dirigida a todas las estaciones del anillo y mediante esta trama cada estación del anillo va recibiendo la notificación de la estación mas próxima que le precede, estación predecesora (M.A.U.), cada estación manda su dirección a la siguiente. La propia monitora ya manda su propia dirección, con esto en un solo giro de la trama todas las estaciones saben cual es su estación predecesora. La estación monitora después de enviar esta trama especial enviará el testigo.

Verificación del testigo y la transmisión de tramas, esto esta relacionado con el cometido del bit monitor.

Cuando una estación envía una trama pone el bit monitor a 0, cuando esta trama pasa por la estación monitora el bit es puesto a 1, de forma que si estas con una trama de datos y esta vuelve a pasar por la estación monitora, (con el bit monitor a 1) la estación sabrá que esta ante una trama huérfana (no ha sido retirada por la estación emisora), entonces la estación monitora eliminara la trama del anillo y genera un nuevo testigo; si la trama es un testigo indica que no ha emitido nadie, y por lo tanto la estación monitora bajara un nivel de prioridad al testigo, si dicho nivel es 0 el nivel lo dejará como esta.

Eliminar tramas mutiladas o erróneas (por las estaciones)

Aquellas tramas incompletas o erróneas la estación monitora se da cuenta y las elimina del anillo y cuando las retira emite un nuevo testigo.

Detectar testigo perdido: La estación monitora lo repone, esta situación la estación monitora la detecta por medio del contador de tiempo máximo, si llega a su máximo valor sin que haya recibido el testigo esta estación la regenera.

Purgar el anillo: En este proceso antes de emitir el nuevo testigo envía una trama para indicarles que se reinicializa el anillo, todas las estación que se halla quedado sin terminar de recibir una trama al recibir esta trama la estación desprecia lo que tiene guardado en el buffer.

PROCESO DE INCORPORACIÓN DE UNA ESTACIÓN AL ANILLO.

Fase 0: Se realiza un chequeo del medio físico que va de la estación a la M.A.U.

Fase 1: La estación que se quiere incorporar pone en marcha un temporizador para detectar si hay estación monitora en el anillo; para detectar si la hay lo que hace es mira si hay un testigo, si el resultado de esta escucha es que no hay monitora, lo que hace es ponerse como monitora y generar un testigo.

Fase 2: Verifica que no hay ninguna estación en la red con su misma dirección.

Fase 3: Participa en el proceso de notificación de vecindad.

La estación tomara nota de su estación antecesora y se identificara ante su sucesora.

Fase 4: solicita los parámetros de la red, esto se hace para dar solidez a la red.

Estos parámetros será:

La localización física.

Valor de temporizaron de notificación de errores recuperables.

SUBCAPA LLC (802.2) (Control Lógico de enlace)

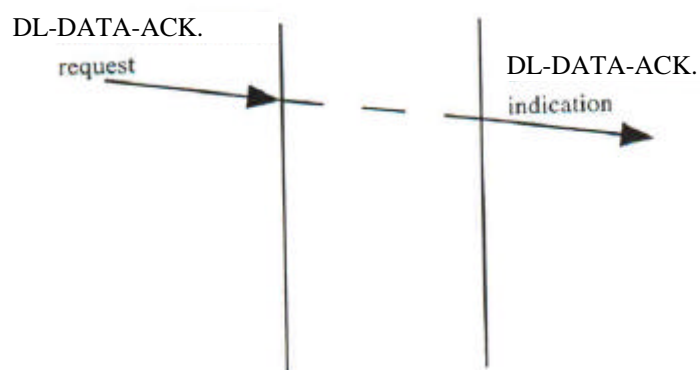
Mecanismo para la certificación de la transmisión y recepción de las tramas dañadas.

Protocolo de transmisión sin conexión:

Servicio sin conexión y sin asentamiento:

La estación origen transmite una trama a la destino cuando la trama se termina de recibir

se termina la comunicación entre ellas.



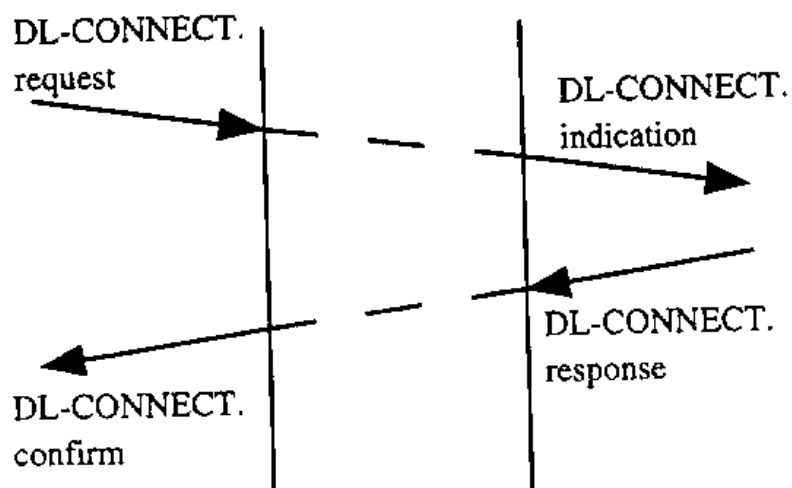
Servicio con conexión:

Establecemos un enlace entre emisor y receptor, se transmite la información, se interrumpe la conexión entre emisor y receptor.

Paso 1:

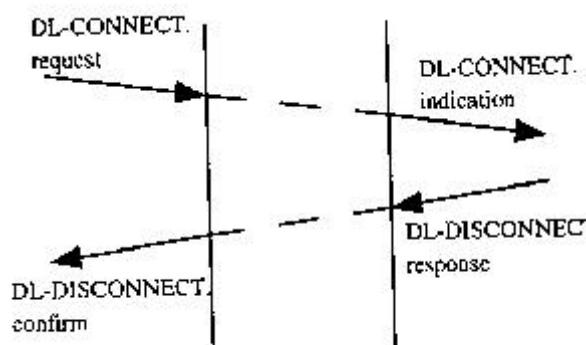
Establecer la conexión

Generar una primitiva DL_CONNECT



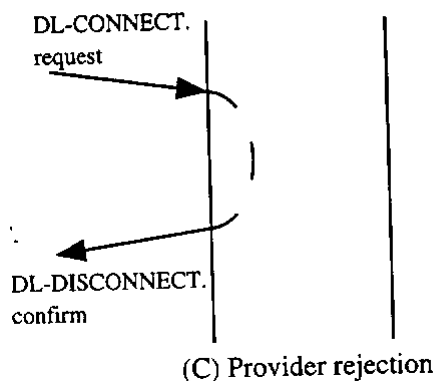
(a) Successful connection establishment

Si la estación destino no puede atender la petición, se lo comunica.

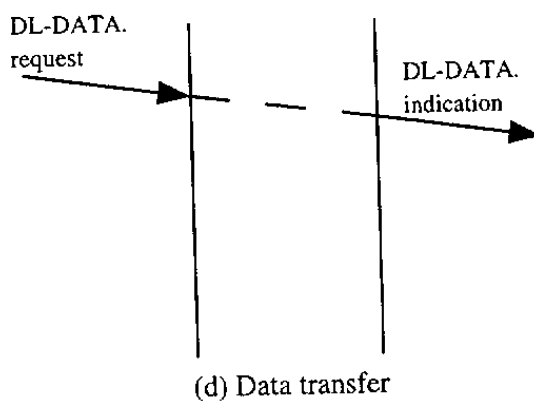


(b) Remote rejection

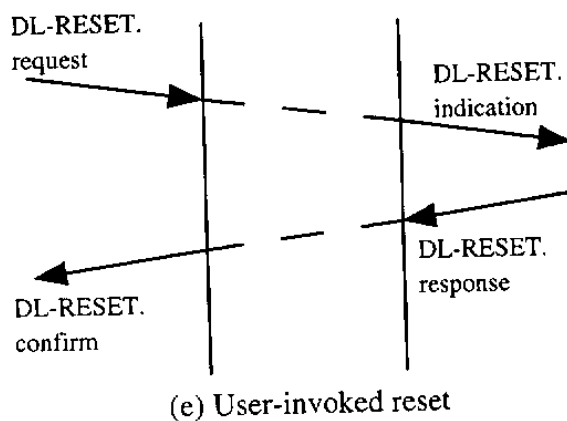
En caso de que la conexión no pueda llevarse a cabo por culpa del enlace, o la estación receptora no esta activa o disponible, la desconexión la haría el propio enlace.



Paso 2: Transmisión de la información con DL_DATA.request y DL_DATA.indication



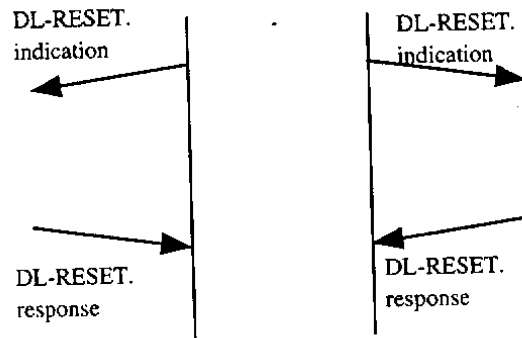
Reset de la conexión por parte de las estaciones



Reset por parte del proveedor

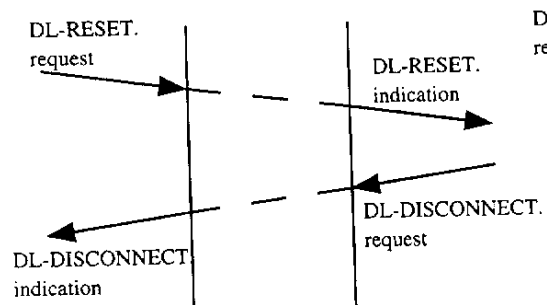
El proveedor generara la primitiva DL_RESET.request.

La cual llegara a las estaciones como una DL_RESET. indication y las estaciones contestaran con DL_RESET.response



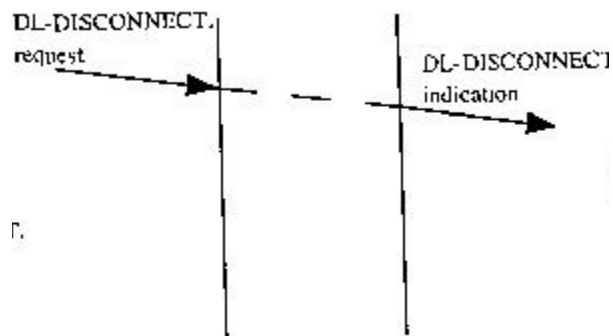
(f) Provider-invoked reset

Si el reset es rechazado.



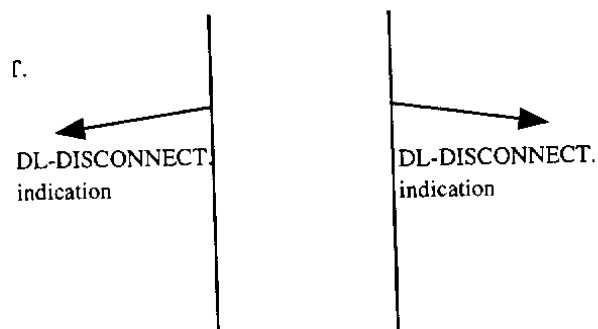
(g) Remote reset refusal

Paso 3:
Desconexión del enlace.



(h) User-invoked disconnect

Desconexión por parte del proveedor.



(i) Provider-invoked disconnect

Las primitivas de la hoja anexa son las primitivas a las que se refieren los gráficos anteriores.

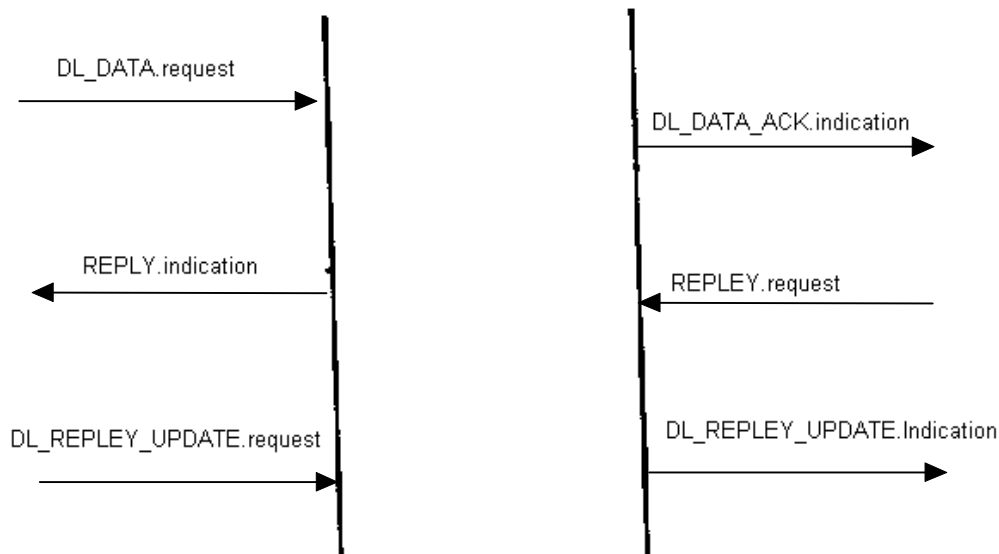
Las primitivas DL_CONNECTION_FLOWCONTROL nos permiten controlar la cantidad de información. el campo amounts de dichas primitivas es el que nos indica el ancho de banda.

Este servicio de conexión no permite ni el modo broad (todas las estaciones a la vez) ni el multi (varias estaciones a la vez), ya que solo admite la conexión entre 2 estaciones.

Servicio sin conexión y con confirmación:

No se va a tener conexión donde se delimita donde empieza y acaba la información.

Vamos a enviar una trama y la estación destinataria, si la a recibido correctamente enviara una confirmación.

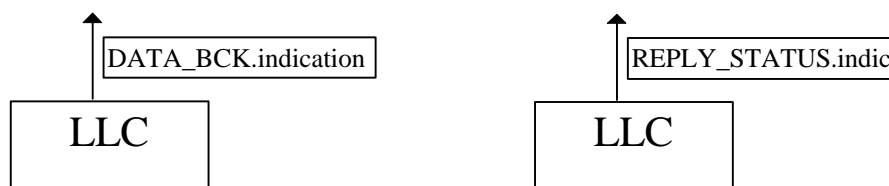


Las DL_REPLY van desde la capa LLC a capas superiores.

Las primitivas REPLY_UPDATE nos permitirán mirar el estado de una confirmación.

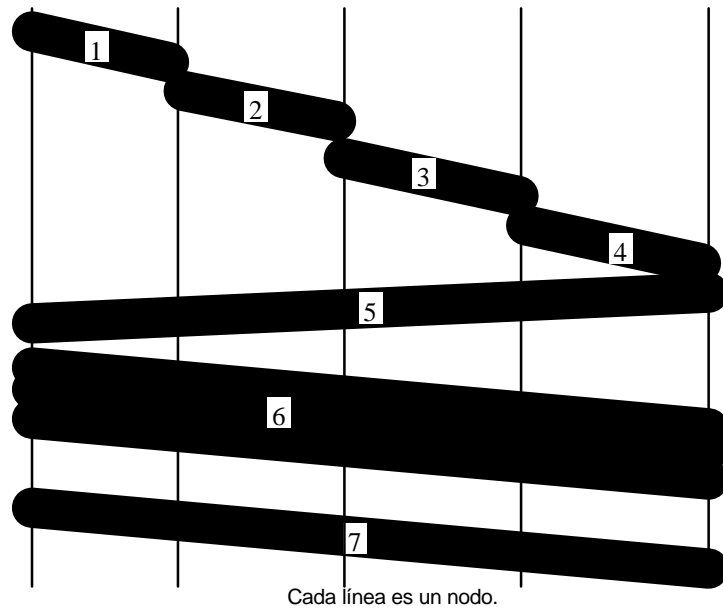
Solicita una actualización de una confirmación.

Las 2 primeras STATUS son internas a cada maquina y va hacia capas superiores.



Conmutación de circuitos. (Llamada de teléfono)

Esquema de servicio de conexión.



Los pasos del 1 al 4 son la búsqueda del receptor.

El paso 5 es la confirmación.

El paso 6 es el envío de información.

El paso 7 es el indicador de fin de transmisión.

El tiempo de respuesta será:

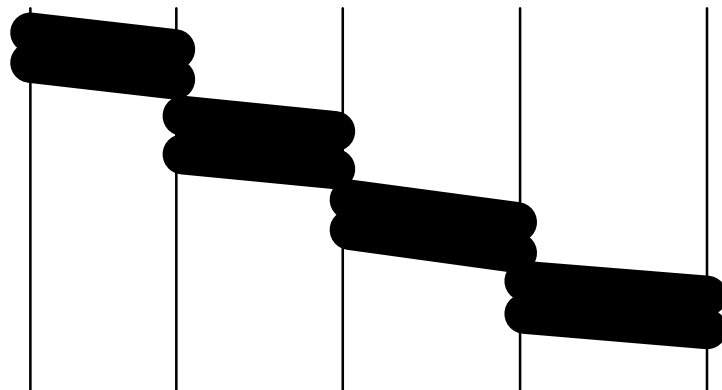
$$Tr = t \text{ de establecimiento de conexión (A)} +$$

En este tiempo no se tiene en cuenta el tiempo de conexión.

Conmutación de mensajes:

Se coge todo el mensaje y se manda todo de vez.

Los nodos intermedios leen todo el mensaje y lo retransmite al siguiente nodo hasta que llega al destino.



Tiempo de respuesta:

$$Tr = \text{numero de nodos} \cdot \left(\frac{\text{longitud}}{\text{velocidad}} + \text{retardo} \right)$$

Suponemos que todos los tiempo de retardo son iguales.

Gracias a este sistema nos ahorramos el tiempo de conexión.

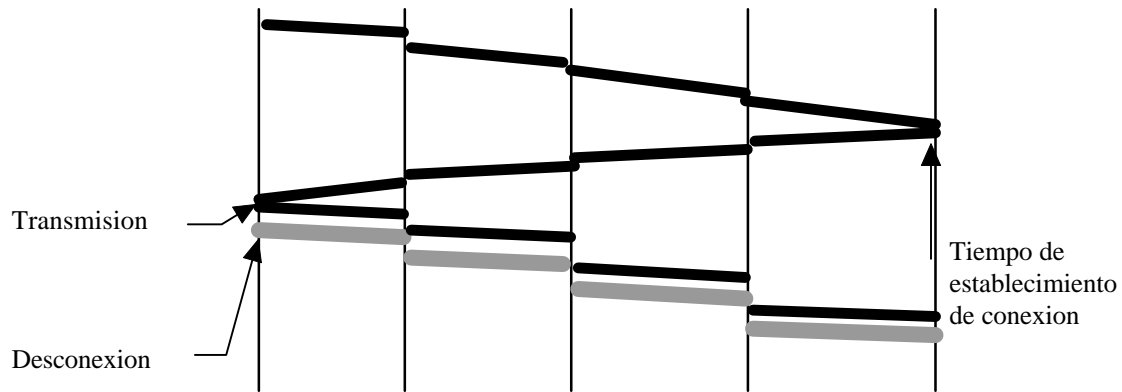
Este sistema es un sistema mas tolerante a los fallos que el anterior, ya que en el anterior si algún nodo falla, la conexión falla, mientras que en este sistema, si un nodo falla, se puede buscar una ruta alternativa.

Como inconveniente tiene que no tiene acotado un tamaño de mensaje, lo cual crea el problema de dimensionar los nodos intermedios para el almacenamiento de la información.

Conmutación de paquetes.

El sistema es muy similar al anterior, pero con la diferencia de que aquí la información no se transmite toda de golpe sino por paquetes de un tamaño máximo.

Circuito virtual:(Servicio con conexión)



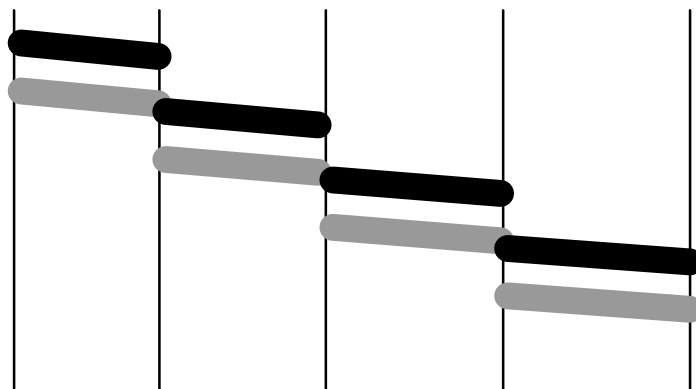
Enviaremos las direcciones origen y destino durante el tiempo de conexión para que así con los paquetes no se envíen dichas direcciones, sino que solo se envía el identificador.

Un error en alguno de los elementos intermedios, supone la desconexión de la transmisión.

Una vez finalizada la conexión se libera todos los nodos intermedios.

Datagrama:

Conmutación de paquetes.



El tiempo de conexión es:

$$Tr = \text{numero de paquetes} \cdot \left(\frac{\text{Longitud}}{\text{Velocidad}} + \text{retardo} \right)$$

Cada paquete tiene que tener información de su encaminamiento.

Este sistema es muy flexible, en caso de error en un nodo se pueden buscar alternativas.

Si por ejemplo una estación está muy saturada, se busca una ruta alternativa, esto implica que podemos

recibir los paquetes desordenados; por lo tanto habrá que implementar un sistema para ordenarlos.
Cada paquete tendrá un identificador de mensaje y un identificador de orden dentro del mensaje.

PROTOCOLO DE ENLACE:

Recuperación de informaciones dañadas.

· Protocolo unilateral restringido:

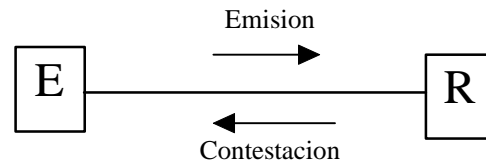
No hay pérdida de paquetes.
El emisor y el receptor no pueden sufrir saturación.
La línea funciona perfectamente.
No repara ni arregla nada

· Protocolo unilateral de parada y espera:

Soluciona problemas de saturación de información en el destinatario.
Suponemos un medio de transmisión libre de errores.

El emisor genera mas información que el receptor puede asimilar con su buffer.

El sistema para solucionar esto es que el emisor una vez enviada una trama espera para que el receptor lo emite una trama de asentimiento.



Si intentamos utilizar este protocolo sobre un sistema con errores se puede dar el caso de que se emitan varios veces la misma trama.

· Protocolo unilateral para canal ruidoso:

Es el anterior pero con el añadido de que la información se dañe o se pierda.

El funcionamiento es:

El emisor emite su trama.

El receptor le contesta si la a asimilado (igual que antes)

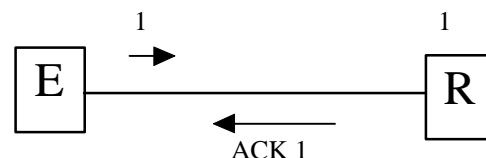
Si el emisor no recibe la contestación en un plazo determinado de tiempo vuelve a reemitir.

Para evitar repetir tramas, se enumeran tanto las tramas de datos como las tramas de confirmación. Dicho enumeración la consideraremos como un identificador de tramas.

El identificador es un bit.

Si el bit es igual al anterior la desecha sino la acepta.

El reconocimiento tiene ese bit al mismo valor.

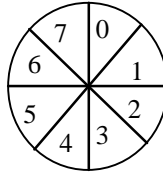


· Protocolo ventana deslizante:

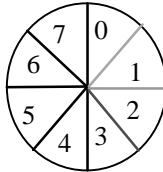
Está pensado para Tx en los 2 sentidos, nos permite enviar varias tramas antes de recibir las confirmaciones. En cada uno de los extremos vamos a tener una ventana de emisión y otra de recepción.

Cada trama va a llegar con un nº para su identificación, tendremos un rango de identificación (Por ejemplo: Para 3 bits -> 8 nº de identificación distintos)

Ventana de emisión:



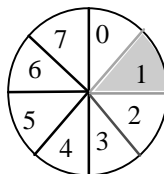
Estado inicial de la ventana.



- Tamaño 0
- Tamaño 1
- Tamaño 2

Como máximo vamos a poder abrir la mitad del rango máximo de trama (en este caso hasta 3). Lo único que tiene son tramas no confirmadas y enviadas.

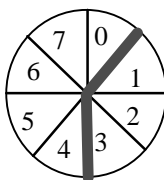
Confirmación de la trama 0.



Si por ejemplo se confirmase antes la trama 1 esto va a depender la implementación del protocolo sea correcto o no. Normalmente cuando se confirma la 1 automáticamente se confirma también la 0. Si no llega la confirmación se le deja un tiempo y se vuelve a enviar.

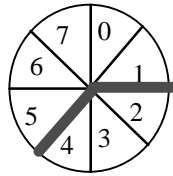
Ventana de recepción:

Es de tamaño fijo

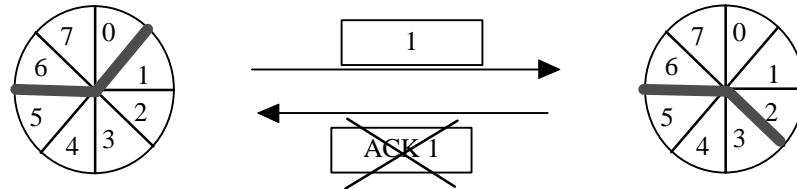


El tamaño de la ventana influye en el nº de tramos de transmisión, si es muy grande se necesitan mas recursos por que se usa casi todo el ancho de banda.

Aquí tenemos los identificadores de las tramas que podemos recibir correctamente en este caso t1, t2 y t3, cuando se recibe una trama lo que se hace es trasladar una posición la ventana de recepción.



Cuando se recibe una trama se envía la confirmación de trama, aquellas tramas que están fuera de la ventana de recepción serán descartadas puesto que son tramas que se envían repetidas. Las tramas que se envían repetidas significa que no le ha llegado el ACK al emisor, entonces tantas veces como llegue una trama repetida se enviara un ACK para confirmarla, este es el motivo por el cual no se puede aumentar el tamaño de la ventana.



Entonces se vuelve a enviar la trama 1 y se va a recibir como una trama nueva cuando en realidad es una copia de la trama 1 porque ya se tiene en ventana de recepción.

Capa de Red:

Esta capa el cometido fundamental va a ser posibilitarnos la comunicación entre distintas redes. Permite realizar las operaciones de encaminamiento de información entre distintas redes.

Si estas redes tienen el mismo formato del identificador nos va a suponer muchos más recursos (se usaría MAC)

RED
LLC
MAC
FISICA

Cada estación de las distintas redes va a tener un identificador físico distinto.

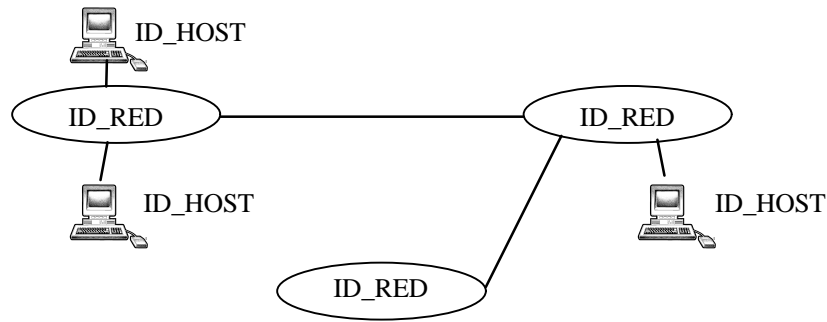
El encaminador debe conocer toda información de todos los dispositivos de identificación, esto funciona bien pero si hay muchas estaciones el encaminador sería inviable, una solución es implementar otra capa para la conexión con otras redes.

PROTOCOLO IP (Internet Protocol)

- Nos facilita el encaminamiento entre distintas redes.
- Para conseguir un encaminamiento correcto nos hace falta un identificador de máquina de red.
- Encima de ID puede haber cualquier otro protocolo, por ejemplo: TCP, UDP, UDD.
- Podemos utilizar identificadores de capa MAC, pero lo normal es utilizar sus propios identificadores.

Los identificadores del protocolo IP v.4

Los identificadores del protocolo IP no nos identifican a una estación de la red, sino que primero nos identifica la red donde se encuentra la estación y después la estación que es en dicha red.



Es decir el identificador de IP será el ID_RED+ID_HOST.

Estos identificadores están compuestos por 32 bits (ID_RED+ID_HOST), y la disposición de los bits es de la siguiente forma:

$\frac{8 \text{ BITS}}{\text{---}} \quad \circ \quad \frac{8 \text{ BITS}}{\text{---}} \quad \circ \quad \frac{8 \text{ BITS}}{\text{---}} \quad \circ \quad \frac{8 \text{ BITS}}{\text{---}}$

Estos bits se pueden repartir entre RED y HOST de las siguientes formas:

Clase A:

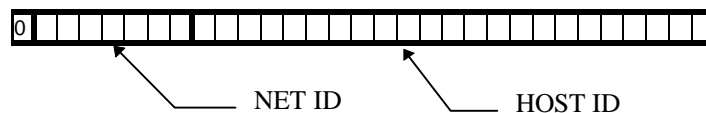
El bit de mayor peso es a 0.

Los 7 siguientes bits nos identifican la red.

Y los 24 bits restantes nos identifican el host.

El rango de los 8 primeros bits (identificador de red) va desde el 1 al 127d ya que el valor 0 y el 127 son restringidos.

Esto quiere decir que con este sistema de identificación IP podemos direccionar 27 redes, y en cada red 224 estaciones.



Clase B:

Los 2 bits de mayor peso estarán a 10, identificador de clase B, después vienen 14 bits de identificador de red, lo cual implica un rango desde 80h (128d) a BFh (191d) en los 8 primeros bits del identificador IP.

Con este tipo de identificadores podremos direccionar 214 redes distintas y en cada red 216 estaciones.

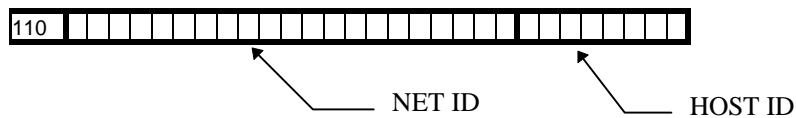


Clase C:

Esta clase está compuesta por:

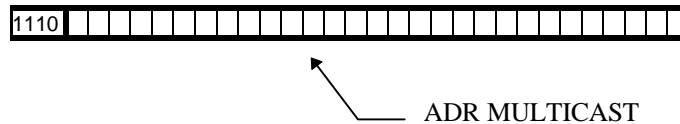
21 bits de ID_RED, 8 bits de ID_HOST, los 3 bits de mayor peso con el valor 110.

En esta clase el rango de los 8 primeros bits va del C0h (192d) al Dfh (223d)



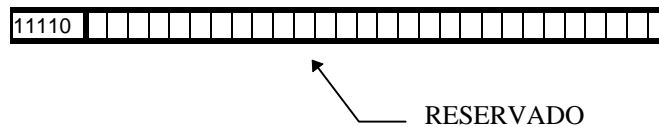
Clase D:

Se identifica por tener los 4 bits de mayor peso puestos a 1110.
Esta clase se utiliza para direcciones en grupo.



Clase E:

Se identifica por tener los 5 bits de mayor peso puestos a 11110.



Normalmente se trabaja en las clases A, B, C.

Ahora veremos algunos ID especiales:

0.0.0.0 -> Este ID hace referencia a la misma máquina, es decir la estación que manda los datos se los manda a si misma.

Si el ID_RED es 0 y el ID_host es distinto de 0 hacemos referencia a un ordenador que se encuentra en nuestra misma red.

255.255.255.255 -> Este ID es un identificador de BroadCast. Este BroadCast normalmente esta limitado solo a nuestra red.

Si nosotros queremos hacer un broadcast sobre otra red, lo que haremos será, poner el ID_RED, y en el campo de HOST poner 255.

Si fuera con un identificador de clase A sería:

____.255.255.255

con uno de clase B:

____.____.255.255

y uno de clase C

____.____.____.255

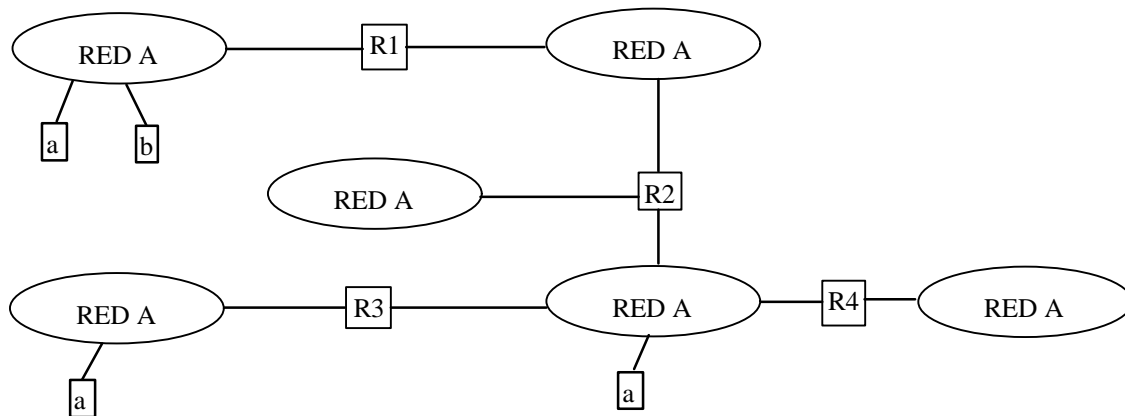
Si no estuviera limitado a nuestra red y deseáramos hacer un broadcast sobre nuestra red, lo que tendríamos que poner en el ID, esta la siguiente dirección:

0.0.0.255 Esta ID esta en clase C, si quisiéramos ponerlo en clase A, lo que tendríamos que enviar seria 0.255.255.255.

Si el ID_RED es distinto de 0 y el ID_HOST=0 hacemos referencia a una red y no a una estación en concreto.

Si mandamos algo a esta dirección nos contestara el router.

127.0.0.0 -> Dirección Loop Back, este Id tiene un comportamiento parecido al ID 0.0.0.0 pero sin que la trama salga a red.



Para transmitir un dato entre la estación A de la red A y la estación B de la red A, los router no se enteran.

Si la información va de la estación A de la red A a la estación A de la red D.

Son los encargados de realizar las operaciones de encaminamiento de información que vaya fuera de la red.

Para poder encaminar la información, los router tiene una tabla de encaminamiento, en dicha tabla esta la ruta que a de seguir una trama para alcanzar su destino.

Para optimizar el funcionamiento de los routers, lo que se guarda en las tablas es la información del siguiente routers para encaminar la información.

Suponemos una trama que va de Aa a Ea.

El routers 1 solo sabe que la trama la tiene que mandar a R2 y el R2 solo sabe de los router R3 y R4.

Cada R conoce solo los router de su entorno directo.

¿La filosofía que se sigue es utilizar unos bits para identifican la subred y también identificar el HOST?

ADR IP

Máscara ->Indicará si la red esta dividida en subredes y cuantos bits del identificador del host se usan para identificar la subred.

ID_Subred -> Los bits de mayor peso del ID_HOST.

La dirección de la subred.

Adv subred: IDred+Bits ID_host utilizadas para la subred resto de bits a 0.

Las máscaras por defecto (sino tenemos dividida en la subred):

Clase A:	255.0.0.0
Clase B:	255.255.0.0
Clase C:	255.255.255.0

El valor hexadecimal del 255 es FF esto con un AND de dir. clase A = ID:RED con clase B= ID_RED; con clase C=ID_RED.

Clase A:	255.0.0.0
	24 bits para ID_host, 8 bits de ID_red

Mascara clase A: 255.255.0.0

Con Subred: El primer 255 es el identificador de red, el según ID es el identificador de la subred, y los 2 últimos 0 son el identificador de Host

La utilización de la mascara permite identificar la información.

Red de clase B y se accede a través de un router.

RED:	158.220.0.0
SUBRED:	158.220.43.0
	158.220.125.0
	158.220.220.0
	158.220.234.0

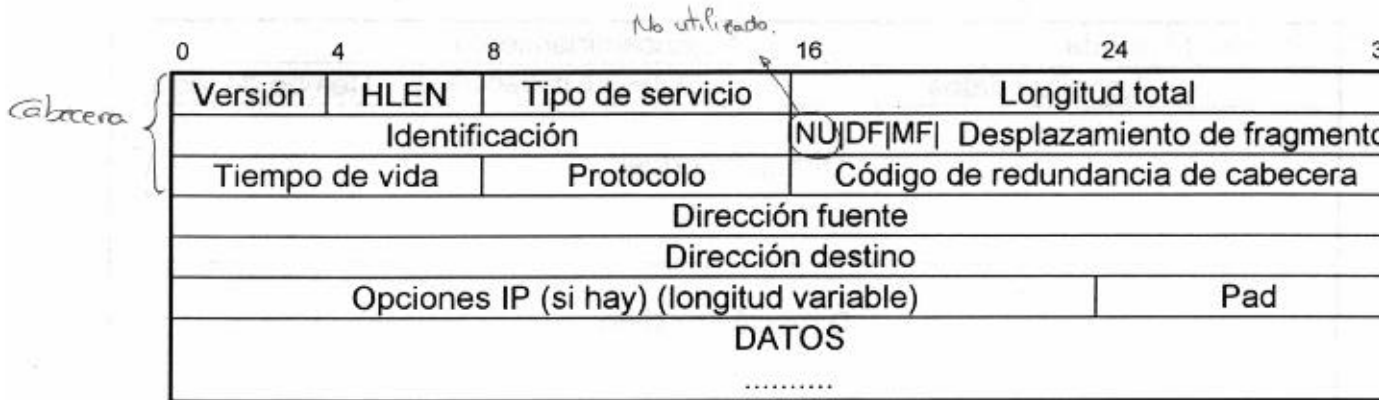
MASCARA:	255.255.255.0
----------	---------------

Cuando el router hace la AND con la mascara sabe que subred corresponde. La mascara se tendría también en los otros interfaces.

La mascara con AND_RED el resultado es la identificador de cada red si p.e. de 158.220.125.0 o sabe que subred es.

Sin la máscara el router solo identificaría 128.220 y no podría alcanzar la dirección.

Cabecera IP v4



Descripción de los campos de la cabecera IP :

- Versión : Indica el número de versión del protocolo.
- HLEN : Longitud de la cabecera en palabras de 32 bits. (Múltiplos de 32 bits)
- Tipo de servicio : Está formado por varios subcampos con la siguiente estructura :

0	3	4	5	7
Preferencia	D	T	R	Sin uso

Preferencia del tipo de servicio : Indica la importancia de cada datagrama. 0 corresponde con la preferencia mas baja y 7 con la mas alta.

D, T, R : Tipo de transporte para el datagrama. Si el bit correspondiente está activo, indica :

D : Bajo retraso (busca una ruta con poco retraso).

T : Alto rendimiento (ancho de banda ancho)

R : Alta fiabilidad (estaciones fiables)

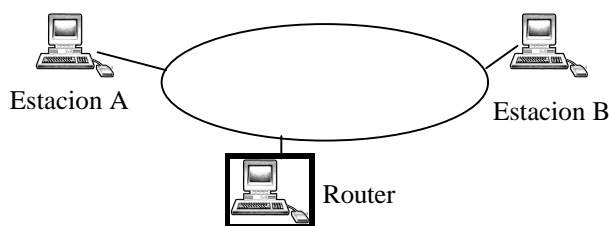
- Longitud total : Longitud total del datagrama expresada en octetos. Incluye cabecera y datos.
- Identificación : Indicador del conjunto de información al que pertenece el datagrama. Todos los fragmentos de un mensaje tienen el mismo.
- Bit DF : Si está activo, indica que el datagrama no puede ser fragmentado.
- Bit MF : Si está activo indica que no se trata del último fragmento de un conjunto de información, desactivándose en el último fragmento.
- Desplazamiento del fragmento : Lugar que ocupa el datagrama dentro del conjunto de información fragmentada.
- Tiempo de vida : Contador para limitar el tiempo de vida de los paquetes. Al llegar a cero, el datagrama se destruye. (valor máximo 255) (No en segundos, sino routers que atraviesa)
- Protocolo : Tipo de protocolo implementado en la capa superior. Especifica el formato que presenta el área de datos.
- Código de redundancia de cabecera : Bits para verificar la integridad de los bits de la cabecera.
- Dirección fuente : Dirección de nivel de red de la estación emisora del datagrama.
- Dirección destino : Dirección de nivel de red de la estación destinataria del datagrama.
- Opciones IP : Campo de longitud variable para incorporar información adicional. Este campo no está presente siempre.
- Pad : campo de relleno para ajustar el tamaño de la cabecera IP a un múltiplo de 32 bits, si se utiliza un campo de opciones que no cumpla esta condición.

Encaminamiento Directo e Indirecto.

Encaminamiento: Proceso que sigue la información para ir de unas redes a otra.

Encaminamiento Directo:

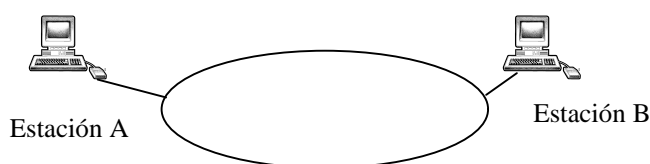
Cuando tenemos una trama que va de una estación a otra dentro de la misma red, estaremos hablando de encaminamiento directo, es decir con encaminamiento directo la trama no tiene por que pasar un ningún router.



De la estación A a la estación B existe un encaminamiento directo, al igual que también hay un encaminamiento directo desde el Router hasta cualquiera de las 2 estaciones (A y B)

Este sistema de encaminamiento tiene un problema, ya que la trama enviada será:

ADR Destino Capa MAC	ADR Origen Capa MAC			
		ADR Destino Capa IP	ADR Origen Capa IP	



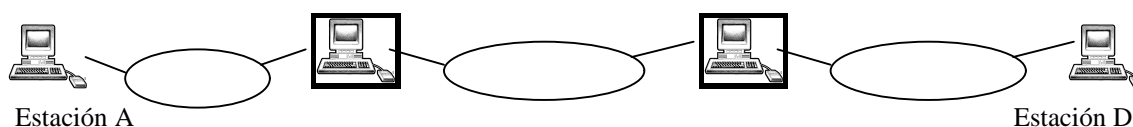
La estación A manda a la estación B una trama, la cual tendrá en su zona de datos la trama IP, y en las dirección tendremos que poner la dirección MAC.

Cuando la estación B recibe la trama, mira la dirección MAC si es la buena la coge y la desmonta para obtener la trama IP (Zona de datos); después mira la dirección IP, para ver si tiene que leerla o no.

El problema esta en que hay que hacer coincidir la dirección MAC con la dirección IP de la estación destino.

Encaminamiento Indirecto:

Es aquel que se realiza en distintas redes, es decir la información pasa por al menos un Router.



Las estación A y D se quieren enviar información.

Proceso a seguir:

Nivel IP: La estación origen tiene que ver si es un encaminamiento directo o indirecto. La forma de descubrir si es un encaminamiento u otro es a través de las direcciones IP (origen y destino) y de una mascara.

Pasamos la IP destino por la mascara y nos da la ADR red/subred destino.

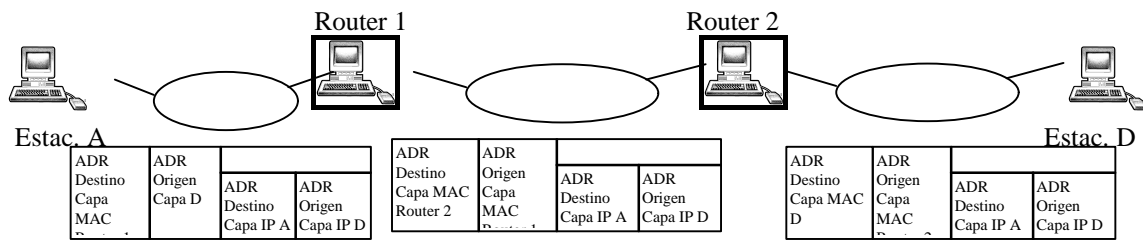
Pasamos la IP origen por la mascara y nos da la ADR red/subred origen.

Si las 2 direcciones son iguales el encaminamiento será directo y si no será indirecto.

Si el encaminamiento indirecto, lo que haremos será enviar la trama al router configurado.

Es decir se hará un encaminamiento directo entre A y el router y entre el ultimo router y la estación D.

Durante todo el proceso de encaminamiento las direcciones IP no se verán modificadas,



La estación A encapsula la trama IP en una trama MAC poniendo como destino la dirección del Router.

El router saca la trama IP (Zona de Datos), lee la dirección IP destino para decidir el nodo de encaminamiento, para saber cual utiliza la IP destino y la IP del interface siguiente, como no son iguales hace un encaminamiento indirecto al siguiente router.

En el router 2 decide por que interface va a enviar la información, comprueba la IP destino con la del interface gracias a la mascara y ve que el encaminamiento es directo.

TABLA DE ENCAMINAMIENTO

Para solucionar el problema que se presenta en los routers sobre la ruta a seguir para alcanzar el destino, el software de encaminamiento utiliza las tablas de encaminamiento.

La información que han de contener las tablas de encaminamiento ha de ser la mínima para poder realizar las operaciones de encaminamiento. El disponer de una tabla con todas los posibles destinos es inviable. Como los routers trabajan entre redes, en sus tablas, solo es necesario que contengan la parte de la dirección IP que identifica la red.

Las tablas contienen pares (N,R) con la dirección IP de la red destino (N) y del siguiente router por el que se ha de pasar para poder alcanzar el destino (R). Estas tablas no contiene el camino completo hasta el destino, solamente el del siguiente router que se puede alcanzar a través de una red simple o conexión directa.

Dentro de las tablas se pueden especificar rutas por defecto, de forma que aquellas direcciones que no pertenecen a la misma red y que no están en la tabla se dirijan a un determinado router.

En casos especiales (administración y alta seguridad), dentro de las tablas se puede especificar rutas para un identificador de máquina.

ALGORITMO DE ENCAMINAMIENTO

```
Route_IP_Datagrama (datagrama,tabla_routing)

Obtener direccion IP destino,  $I_D$ , del datagrama
Procesar dirección IP de la red destino  $I_N$ 
if  $I_N$  pertenece a la misma red
    Enviar datagrama a estación destino;
    (Incluye resolución de la dirección  $I_D$  y
    encapsulado del datagrama en una trama)
else if  $I_D$  está en la tabla como identificador de maquina
    encaminar el datagrama como se especifica en la tabla;
else if  $I_N$  aparece en la tabla de encaminamiento
    encaminar el datagrama como se especifica en la tabla;
else if hay una ruta por defecto
    encaminar el datagrama al router por defecto;
else indicar error de encaminamiento;
```

Si en el campo R aparece un DIRECT, nos quiere decir que es el ultimo router, por lo que va directamente a la estación. Otra posibilidad es que en vez de DIRECT, aparezca directamente la dirección IP de la estación.

Mapeo ADR RED con ADR FÍSICA:

Mapeo Directo:

Emparejamos las direcciones de red y la dir. física en una tabla.

Otra forma es que a través de una dirección de red obtener la dirección física.

$$AF=F(AR)$$

Esta 2ª forma no es viable en 802.3, ya que el tamaño de la Af es de 48 y la Ar es de 32 bits.

PROBLEMAS:

En el sistema de tablas, cuando se modifica algo en la red, hay que modificar todas las tablas de los routes.

La otra forma presenta el problema de que un cambio en la tarjeta de red implica tener que cambiar la dirección de red, ya que cambia la dirección física.

Enlace dinámico:

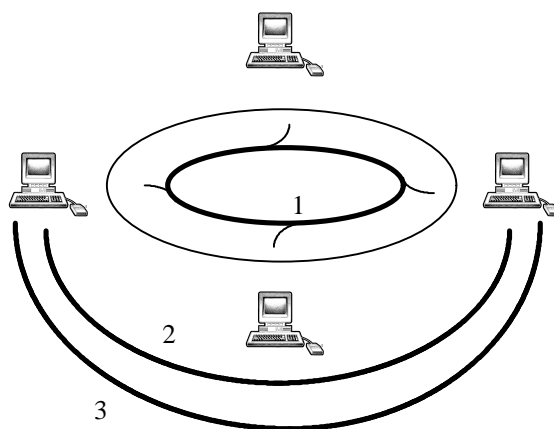
Cada vez que una estación tenga que enviar una trama IP a otra estación, lo primero que hará será consultar su dirección física.

Problema:

Consultar una dirección que no conocemos, la cual nos hace falta para poder preguntarla.

Para ello se utiliza un protocolo de resolución de direcciones (ARP (Address Resolution Protocol))

Funcionamiento del ARP:



La estación A quiere enviar una trama IP a la estación B.

La trama IP la tiene que encapsular en una trama MAC, para la cual le hace falta la dirección MAC de B.

Para descubrir la dirección MAC de B lo que hará será manda una trama BROADCAST a nivel de capa MAC, dicha trama estará compuesta por:

1) ARP

MAC A	IP A	IP B
Broad		

Esta trama va de la estación A a todas las estaciones.

Por ser una trama Broadcast, todas las estaciones la lee y comparan la IP destino (IP B) con la suya si la comparación dice que son distintas, pasa de ella, y si las IP son iguales, la estación contesta con la trama siguiente:

2) ARP

Esta trama va de la estación B a la estación A.

MAC B	MAC A	IP B	IP A
-------	-------	------	------

Gracias a esta trama la estación A ya sabe cual es la dirección MAC de B y le manda la trama IP que quería mandarle en un principio.

Formas de Optimizar el ARP:

Una de las formas de optimizar el protocolo ARP es utilizar una cache donde se almacena las ADR MAC y ADR RED de la estación destino.

En este caso normalmente la estación B también cogería la dirección MAC y de RED de A. También es posible que el resto de las estaciones cojan la dirección de la estación A gracias a la red BROADCAST.

La cache no es permanente, ya que una estación puede sufrir un cambio de dirección MAC.

Una forma de darle tiempo de máximo de vida a cada entrada de la caché.

Las entradas se pueden actualizar cuando le llega información de dicha dirección que ha sido modificada, bien sea por una trama de datos o por cualquier otra trama.

Cabecera IP v6

0	4	8	16	24	31
Versión	Prioridad	Encaminamiento			
Longitud datos			Siguiente cabecera	Tiempo de vida	
Dirección origen (128 bits)					
Dirección destino (128 bits)					
DATOS					

Descripción de los campos de la cabecera IP :

- Versión : Indica el número de versión del protocolo.
- Prioridad : Identifica el nivel de prioridad asignado a cada paquete. Se divide en dos rangos :
Valores de 0 a 7 identifican la prioridad de tráfico que puede ser regulado por el emisor en caso de congestión. Valores de 8 a 15 identifican la prioridad de tráfico que no puede ser descartado en caso de congestión.
- Encaminamiento : Este campo puede ser utilizado para marcar aquellos paquetes que precisan un tratamiento especial por los routers. En aquellos dispositivos en los que no este implementada esta funcionalidad, este campo tomará valor 0.
- Siguiete cabecera : Identifica el tipo de cabecera que sigue inmediatamente a la IPv6. *(después de la dirección)*
- Tiempo de vida : Contador para limitar el tiempo de vida de los paquetes. Al llegar a cero, el datagrama se destruye.
- Dirección fuente : Dirección de nivel de red de la estación emisora del datagrama, con un tamaño de 128 bits.
- Dirección destino : Dirección de nivel de red de la estación destinataria del datagrama, con un tamaño de 128 bits.

La cabecera IPv6 puede ir seguida por una o varias extensiones de cabecera, que permiten implementar distintas funcionalidades no soportadas por la cabecera principal :

- Hop-by-Hop : Incluye información adicional que debe ser examinada por cada uno de los nodos por los que circula el paquete.
- Cabecera de Routing : Indica una lista de nodos por los que debe pasar el paquete en la trayectoria seguida desde su origen al destino.
- Cabecera de fragmentado : Permite el envío de paquetes de un tamaño mayor del que pueden gestionar los nodos intermedios, permitiendo el fragmentado de la información y su posterior reensamblado.
- Cabecera de opciones de destino : Permite añadir información adicional que solo deba ser examinada por el nodo destino.

El formato de las direcciones esta compuesto por 4 campos pero de 32 bits cada uno en vez de los 8 bits del IP v.4

En este protocolo no existen las clase A, B y C.

En IP v.4 tenemos UNICAST, MULTICAST y BROADCAST ahora con IP v.6 añadiremos también ANYCAST.

Es similar al multicast pero con la condición de que la trama solo se envía a la primera estación que se encuentre de dicho grupo y no al resto.

En multicast se envía una copia a cada elemento del grupo.

Para aligerar un poco el protocolo se han reducido los campos de cabecera, esto se a hecho por el aumento

de bits en las direcciones $128 + 1287$ bits.

Capa Transporte:



El problema de ordenador destino y origen están solucionados en capa de red pero tenemos un problema en capa de transporte para identificar procesos destino y origen -> Tramas perdidas y dañadas -> Protocolos.

Protocolos: UDP, TCP están asociados a IP, el protocolo UDP no es fiable por no tener control de flujo.

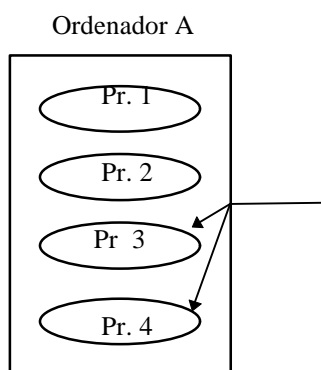
UDP:

Tiene mecanismos necesarios para identificar aplicaciones destino (que se esta ejecutando) es la destinataria del datagrama.

Esto plantea un problema:

Los procesos son creados y destruidos dinámicamente motivo por el cual el remitente del datagrama rara vez tiene información para identificar un proceso en otra maquina.

Poder reemplazar el proceso que recibe el datagrama sin tener que notificarlo a todos los posibles remitentes.



Con un reset en el ordenador los ID se borran, entonces tendremos el mismo problema , ya que tenemos que comunicar el ID del proceso que recibe a todas las estaciones.

Vamos a poder tener un interface común a todas las estaciones y que solo conociendo el formato de las tramas nos de igual la función que lo implemente.

SOLUCIÓN:

Establecer un conjunto abstracto de puntos destino y se identificara como un n° entero.

El S.O. nos suministrara un interface para que los procesos intercambien información especificando un puerto para recibir información o accediendo a un determinado puerto para obtener información.

Filosofía de los puertos:

Se establecen como una serie de buzones y cuando queremos enviar información el S.O. se encarga de procesarla y enviarla. Si la información se quiere enviar a un proceso la capa de transporte le dice donde tiene que ir a buscar su información. La información se deposita en las puertas correspondiente, en su nombre el ID del proceso, nos da lo mismo por que los ID del puerto serán siempre iguales.

Y al mismo tiempo conseguiremos abstraer del funcionamiento a cada estación.

Para que 2 puertas se comuniquen los mensajes estarán identificados como:

ADR destino (proceso)	Puerto destino
-----------------------	----------------

El protocolo UDP es un protocolo primario para la comunicación.

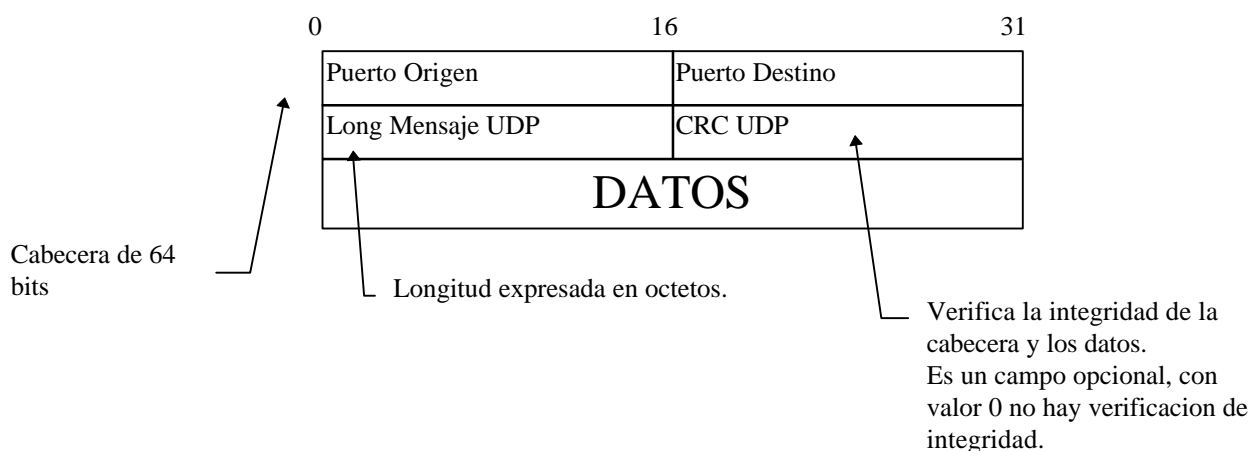
Formato del mensaje UDP:

ID puerto ORIGEN	ID puerto destino
------------------	-------------------

No establece comunicación entre el puerto destino y el puerto origen, nos podemos encontrar con mensajes repetidos que lleguen fuera de orden o se pierdan. La aplicación que este usando este protocolo será la responsable de solventar los problemas de fiabilidad del intercambio de información.

Es un protocolo ligero (no tendrá excesivo transporte) pero no es fiable.

Formato del DATAGRAMA UDP:



TCP (Transmisión Control Protocol)

Propiedades que debe de cumplir un servidos de transporte fiable.

Ordenación de paquetes.

Gran cantidad de datos se dividen en octetos que según la implementación de capas superiores pueden llegar desordenados (La ordenación en capa de transporte aunque la información incluida para ello se pone en la capa de RED).

Conexión de circuito virtual:

(Establecen: Conexión, transporte de información entre estaciones y circuitería de conexión.).

Para establecer y cerrar comunicación necesita operaciones tanto la aplicación del emisor como la aplicación del receptor, ambas aplicaciones deben notificarlo al S.O. acerca de la necesidad de reciclar una transmisión - > el S.O. verifica que existen los permisos necesarios para reciclar la conexión y va a disponer de los recursos

necesarios para que esta se lleve a cabo una vez establecida la conexión, el protocolo se asegurara de la comunicación con el otro protocolo de la otra máquina, para verificar que la información es recibida correctamente.

TCP (Transmission Control Protocol)

INTRODUCCIÓN

En las capas inferiores, las redes de comunicaciones no incorporan mecanismos para asegurar una transmisión fiable de la información. Los paquetes pueden ser perdidos cuando los errores de transmisión afectan a los datos, el hardware falla, o la carga de la red es mas alta de la que soporta. Las redes que encaminan paquetes dinámicamente, pueden entregar los paquetes fuera de orden, con retrasos importantes o duplicados. Además en las distintas implementaciones de redes se establece un tamaño óptimo de paquete, o se establecen otro tipo de restricciones para conseguir adecuadas velocidades de transmisión.

En los niveles altos, los programas de aplicación necesitan enviar extensos volúmenes de datos entre distintas máquinas. El usar sistemas de transmisión no fiables para transmitir grandes cantidades de datos, obligan al programador a implementar detección y recuperación de errores en cada programa de aplicación. El realizar este tipo de control, es una tarea difícil, por lo que sería recomendable ofrecer al programador de aplicaciones soluciones para conseguir una transmisión fiable mediante funciones de control de flujo que abstraigan al software de aplicación de este tipo de problemas, definiendo un interface uniforme para los servicios de control de flujo.

PROPIEDADES DE LOS SERVICIOS COMUNICACIÓN FIABLE

El interface entre los programas de aplicación y los servicios que aseguran una transmisión fiable han de presentar las siguientes 5 funciones:

- Ordenación de paquetes: Cuando dos programas de aplicación transfieren grandes cantidades de datos, estos se dividen en grupos de octetos. El servicio de control de flujo en la máquina destino, debe encargarse de recibir la misma secuencia de octetos que generó la máquina origen.
- Conexión de Circuito Virtual: Antes que la transmisión pueda comenzar, tanto el programa de aplicación emisor, como el receptor, deben informar a sus respectivos sistemas operativos sobre la necesidad de realizar una transmisión, de forma que estos verifiquen que la transmisión que se desea realizar está autorizada y que ambos extremos están dispuestos. Si todas las condiciones se cumplen, el protocolo informa al programa de aplicación que se ha establecido una conexión y puede comenzar la transferencia. Durante la transferencia los protocolos siguen comunicándose para verificar que los datos son recibidos correctamente. Si la comunicación falla por cualquier razón, ambas máquinas lo detectan, comunicándolo a los programas de aplicación. El uso del término Circuito Virtual se debe a que el programa de aplicación ve la conexión como un circuito físico dedicado.
- Buffer de transferencia: Los programas de aplicación envían flujos de datos a través del enlace virtual pasando octetos al software del protocolo. El tamaño de la información pasada depende de la aplicación. El software del protocolo es libre de dividir la información pasada en paquetes. Para conseguir una transmisión mas eficiente y minimizar el trafico de la red, se puede agrupar información en un paquete, o en caso que la información pasada sea extensa, se puede dividir en varios paquetes. En ambos casos, la información debe ser pasada a la aplicación destino en el mismo

formato que la genere la aplicación origen. Para esto es necesario disponer de un buffer en donde a partir de los paquetes recibidos se reconstruya la información original.

- Flujo de datos sin estructurar: El protocolo no tiene un formato definido de los datos a transmitir. Los programas de aplicación deben conocer el contenido y acomodarlo al flujo de datos antes de iniciar la transmisión.
- Conexión Full Duplex: Permite la transmisión en los dos sentidos.

TCP (Protocolo Control Transmisión)

Por TCP se entiende las especificaciones de un protocolo de comunicaciones, no un programa (TCP se puede implementar de muchas maneras).

El protocolo tiene los siguientes cometidos:

Especifica el formato de los datos y confirmaciones que dos máquinas intercambian para conseguir una transferencia fiable.

Los medios que usan los ordenadores para asegurar que los datos llegan correctamente.

Distinguir entre los múltiples destinos en una misma máquina.

Recuperación de errores como paquetes perdidos o duplicados.

Establecimiento y finalización de una transferencia de datos.

Con el objeto de dotar al protocolo de la suficiente flexibilidad, no se especifican los detalles del interface entre los programas de aplicación y TCP, indicándose únicamente los servicios que ha de ofrecer el protocolo.

PUERTOS, CONEXIONES Y PUNTOS DESTINO

Uno de los cometidos de TCP es identificar, dentro de una máquina, la aplicación a que va destinada el datagrama. Para realizar esta operación se asigna un número entero para identificar el puerto destinatario dentro de una máquina, al igual que ocurre en UDP.

Los puertos en TCP son mas complejos que en UDP, puesto que un número de puerto no corresponde con un único objeto, sino que identifica una conexión a un circuito virtual, no a un puerto específico.

Como *punto destino* se entiende un par de enteros (host, port) en donde host es la dirección IP de la máquina y port identifica el puerto TCP en esa máquina.

Como *conexión* se entiende un circuito virtual entre dos programas de aplicación, definida por dos puntos destino. Esto permite que varias conexiones compartan un mismo punto destino sin incurrir en ambigüedades, puesto que TCP asocia los nuevos mensajes con una conexión en lugar de un puerto, usando los dos puntos destino para identificar la conexión adecuada. Esto posibilita que un programa ofrezca acceso concurrente a múltiples conexiones simultáneas, sin necesidad de asignar un número de puerto para cada conexión.

INICIALIZACIÓN ACTIVA Y PASIVA

En TCP antes de establecer una comunicación, los puntos destinos deben de estar conformes en realizar la transmisión. El proceso seguido es el siguiente: el programa de aplicación de uno de los extremos realiza una función de inicialización pasiva, indicando a su sistema operativo que aceptará una nueva conexión; este asignará un numero de puerto TCP para su punto de destino. El programa de aplicación del otro extremo debe entonces contactar

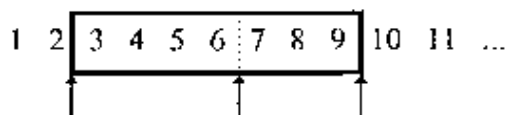
con su sistema operativo usando una solicitud de inicialización activa para establecer la conexión. Los dos módulos de TCP se comunican para establecer y verificar la conexión. Una vez que la conexión se ha creado, los programas de aplicación pueden comenzar a transferir datos. El software TCP de ambos extremos intercambia mensajes durante la transferencia de datos para garantizar la fiabilidad de la transmisión.

CONTROL DE FLUJO

TCP ve los mensajes a transmitir como una secuencia de octetos que se dividen en segmentos para su transmisión. Generalmente cada segmento se asocia a un único datagrama IP.

TCP usa un mecanismo de ventana deslizante para conseguir una transmisión eficiente y un control del flujo. Permite enviar múltiples segmentos antes que lleguen los asentimientos, logrando incrementar el rendimiento puesto que mantiene mas tiempo la red ocupada. Mediante el control de flujo el destinatario puede detener la transmisión hasta que disponga de suficiente espacio en el buffer para aceptar mas datos.

El mecanismo de ventana deslizante usado por TCP opera a nivel de octetos. Estos son numerados secuencialmente, de forma que el emisor guarda tres punteros asociados a cada conexión, de forma que definen la ventana deslizante:



El primer puntero marca el extremo izquierdo de la ventana, separando los octetos que han sido enviados y confirmados. Un segundo puntero marca el extremo derecho de la ventana, delimitando el octeto mas alto que se puede enviar antes de recibir mas confirmaciones. El tercer puntero marca la separación entre los octetos que han sido enviados y los que no.

El receptor mantiene una ventana similar, con el objeto de solucionar el problema de datos duplicados. El tamaño de la ventana es el mismo que el del emisor, de forma que contiene aquellos octetos que pueden llegar duplicados. Cuando se recibe el octeto siguiente al de mas a la derecha de la ventana, el remitente ha recibido confirmación del octeto de la izquierda de la ventana, con lo cual no lo retransmitirá, y se puede desplazar la ventana una posición a la derecha.

TCP permite variar el tamaño de la ventana a lo largo del tiempo. Cada asentimiento, en el que se especifican cuantos octetos se ha recibido, contiene una notificación de ventana que especifica cuantos octetos mas de datos esta preparado para aceptar el receptor (espacio del buffer disponible). Como respuesta a una notificación de incremento de ventana, el emisor aumenta el tamaño de su ventana deslizante y procede a enviar los octetos. La respuesta a una notificación de disminución de ventana, el emisor disminuye el tamaño de esta y detiene el envío de octetos hasta el límite. No se genera ningún problema con los asentimientos pendientes, puesto que en el segmento que contiene la notificación llegan también asentimientos, y el tamaño de la ventana cambia a la vez que se desplaza. Este mecanismo permite un buen control de flujo a la vez que una transferencia fiable, al poder adaptar la transmisión a la capacidad del receptor, permitiendo la coexistencia de máquinas de distinta velocidad y capacidad.

Se pueden crear problemas de congestión al sobrecargarse máquinas intermedias de la red (routers). TCP no dispone de mecanismos explícitos de control de congestión, resuelve los

problemas de control de flujo entre extremos, pero algunas implementaciones incorporan este tipo de mecanismos.

FORMATO SEGMENTO TCP

Los segmentos son intercambiados entre máquinas para establecer conexiones, transferir datos, enviar confirmaciones, notificar el tamaño de las ventanas y para cerrar conexiones. Para enviar asentimientos de datos enviados de una máquina A a B, se puede aprovechar un segmento en el que B envíe datos a A.

El formato de segmento TCP es el siguiente:

0	4	10	16	31
PUERTO ORIGEN				PUERTO DESTINO
NÚMERO DE SECUENCIA				
NÚMERO DE ASENTIMIENTO				
HLEN	RESERV.	CODE	VENTANA	
CHECKSUM			PUNTERO URGENTE	
OPCIONES (SI HAY)				PAD
DATOS				
.....				

Cada segmento está dividido en dos partes: una cabecera TCP y una zona de datos. La cabecera consta de los siguientes campos:

- Puerto Origen y Puerto Destino:** Contienen los números de puerto TCP que identifican los programas de aplicación en el extremo de la conexión.
- Número de secuencia:** Identifica la posición que ocupa dentro del mensaje los datos que contiene el segmento. Permite recomponer el mensaje original ordenando correctamente los distintos fragmentos.
- Número de asentimiento:** Identifica el número del siguiente octeto que se espera recibir. Esto indica que el emisor del segmento ha recibido correctamente el octeto anterior al referenciado, en una anterior transmisión (en sentido opuesto) con el receptor.
- Hlen:** Este campo contiene la longitud de la cabecera del segmento expresada en múltiplos de 32 bits.
- Reserv.:** Campo de 6 bits reservado para futuros usos.
- Code:** Campo de 6 bits utilizado para indicar la función y contenido del segmento. Los valores para cada uno de los bits son los siguientes:

URG	ACK	PSH	RST	SYN	FIN
-----	-----	-----	-----	-----	-----

- URG:** El campo de puntero urgente es válido.
- ACK:** El campo ACK es válido.
- PSH:** Solicitud de atención sin pasar por buffer.
- RST:** Reset de la conexión.
- SYN:** Sincronización de los números de secuencia.
- FIN:** Final del mensaje.

Ventana: Campo de 16 bits indicando el tamaño de la ventana.

Checksum: Código de validación para verificar la integridad de los datos.
Puntero Urgente: Posición en la ventana donde se finalizan los datos prioritarios.
Opciones: Campo opcional de longitud variable.
Pad: campo de relleno para ajustar las opciones a múltiplos de 32 bits.

DATOS PREFERENTES

Aunque TCP es un protocolo orientado a mensajes, muchas veces es importante para el programa de uno de los extremos de la conexión, enviar datos fuera de secuencia, sin esperar a que el programa del otro extremo procese los octetos del mensaje aún pendientes.

Para implementar el mecanismo de datos preferentes, TCP permite al remitente marcar los datos como urgentes. El programa destinatario recibe una notificación de la llegada de este tipo de datos, independientemente de su posición en la secuencia de datos, pasando la aplicación, por indicación de TCP a *modo urgente*, volviendo a *modo normal de operación* una vez tratados los datos urgentes.

La implementación sobre como TCP informa a la aplicación de la existencia de datos urgentes, depende del sistema operativo del ordenador. El mecanismo usado para marcar datos urgentes consiste en el bit URG y el campo Puntero Urgente, de forma que cuando el bit URG está activo, el campo Puntero Urgente especifica la posición en la ventana donde los datos urgentes finalizan.

TAMAÑO MÁXIMO DEL SEGMENTO

No todos los segmentos enviados a través de la conexión tienen la misma longitud. Sin embargo ambos extremos necesitan estar de acuerdo con la máxima longitud del segmento que transferirán. Las implementaciones de TCP de ambos extremos utilizan el campo de opciones para especificar el tamaño máximo de segmento (MSS). Este aspecto es especialmente importante en redes de alta velocidad, en las que interesa utilizar segmentos lo mas grandes posibles para lograr un buen aprovechamiento del ancho de banda.

Si el destino reside en la misma red, TCP utiliza el máximo tamaño que permite la implementación física de la red. Si el destino se encuentra en otra red, la especificación actual recomienda utilizar un tamaño máximo de segmento de 536 octetos (tamaño por defecto de un datagrama IP, 576, menos el tamaño estandar de las cabeceras TCP e IP).

El uso de tamaños pequeños de segmento ocasiona un bajo aprovechamiento de la red, puesto que la relación entre datos y octetos enviados es baja. Por otro lado, utilizar tamaños de segmento elevados puede dar lugar a unas prestaciones bajas, debido a que un segmento, al tener mayor longitud, tendrá mas posibilidades de sufrir un error y tener que ser retransmitido.

En teoría, el tamaño óptimo de segmento, es aquel en el que el datagrama IP que lleva el segmento, es lo mas largo posible para que no precise ser fragmentado a lo largo del camino seguido desde el origen al destino. En la práctica obtener el tamaño óptimo resulta difícil por los siguientes motivos: TCP no incorpora mecanismos para este fin; los routers cambian las rutas dinámicamente, con lo cual cambia el tamaño en que deben ser fragmentados los datagramas; el tamaño óptimo depende de los protocolos de bajo nivel.

CONTROL DE VALIDACIÓN EN TCP

El campo de Checksum en la cabecera TCP contiene un código de validación consistente en un entero de 16 bits, usado para verificar la integridad de los datos, así como la cabecera.

Para el cálculo del código de validación, se precede el segmento TCP con una pseudo cabecera (que no se transmite) que permite al destinatario verificar que el segmento ha llegado a su correcto destino (máquina y puerto correctos). El formato de la pseudo cabecera es el siguiente:

0	8	16	31
DIRECCIÓN IP ORIGEN			
DIRECCIÓN IP DESTINO			
CERO	PROTOCOLO	LONGITUD TCP	

Cuando llega un datagrama portando un segmento TCP, IP pasa a TCP tanto el segmento como las direcciones IP, necesarias para identificar la conexión, y la demás información para confeccionar la pseudo cabecera. El campo Protocolo contiene el valor que la capa inferior contiene en el campo protocolo (para datagramas IP que contienen un segmento TCP, este campo toma valor 6). El campo Longitud TCP indica la longitud del segmento TCP, incluyendo la cabecera TCP. El campo Cero es de relleno.

INICIALIZACIÓN ACTIVA Y PASIVA

Cuando tenemos una aplicación o proceso con capacidad de recibir datos, esta aplicación se lo notificara al S.O., el cual le asignara un puerto, a través del cual se comunicara con otros procesos y se quedara a la espera de recibir información. Este proceso se corresponde con la inicialización pasiva.

La aplicación que desea enviar información va a realizar una inicialización activa, para ello se lo comunica al S.O.

Se establece una comunicación entre los protocolos TCP de las 2 maquinas, con la intención de verificar si la conexión es posible y establecer la conexión.

Una vez establecida la conexión los programas de comunicación pueden esperar a transmitir, mientras tanto los protocolos TCP de cada maquina siguen comunicándose, para saber si la conexión funciona correctamente.

El TCP es un protocolo fiable, por lo tanto a de tener implementado el control de flujo, para así solucionar la perdida , duplicados de paquetes, al igual que los problemas de saturación del receptor.

Los mensajes son secuencias de octetos, los cuales formara un datagrama.

El sistema de control de flujo es el de ventanas deslizantes, ya visto en su día.

El sistema de ventanas deslizantes en TCP funciona a nivel de octeto, es decir se confirmaran los octetos recibidos; esto no complica que se tenga que mandar una confirmación para cada octetos, sino que se manda hasta que numero de octeto a llegado correctamente.

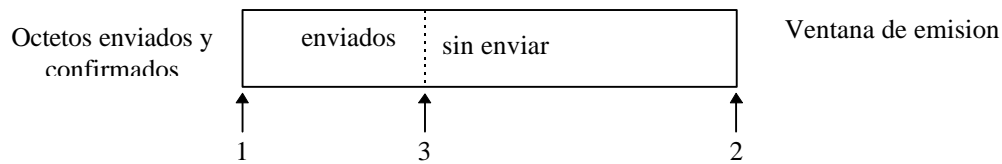
Para el control de la ventana TCP implementa 3 punteros que son:

1.- El primer puntero nos marca el limite izquierdo de la ventana y nos indica los octetos enviados y confirmados.

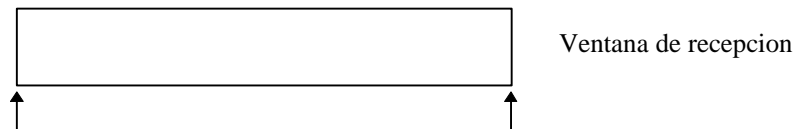
2.- Nos indica la parte derecha de la ventana y nos marca el octetos mas alta que se puede enviar antes de recibir mas confirmaciones.

Es decir no se podrá transmitir mas hasta que no llegue la confirmación.

3.- Nos separa los octetos que se pueden enviar de los octetos que ya han sido enviados, es decir que este puntero nos marca el ultimo octeto que se a enviado.



En recepción disponemos de una ventana del mismo tamaño que la ventana de emisión, la cual solo esta limitado por 2 punteros



La ventana contendrá a los octetos que se pueden recibir, es decir los que tendrán confirmación.

En el momento que se recibe un octetos, lo que haremos será desplazar la ventana. Si ahora volvemos a recibir el mismo octeto, y este estará fuera de la ventana no lo recogerá.

Veamos un ejemplo:



Llegan los octetos 5 6 y 7, y la ventana de recepción se mueve.

Se envía la confirmación para las octetos 5 6 y 7, pero la confirmación no llega, con lo que el emisor vuelve a emitir las trama, las cuales cuando llegan se desechan por no estar en la ventana de recepción.

El funcionamiento del asentamiento es:

Se realiza una confirmación por defecto.

Cuando nos llega la confirmación de un octeto se dan por confirmados los octetos anteriores que estaban pendientes de confirmación.

Cuando se reciben confirmaciones en el emisor se desplazan los punteros izq. y derecho tantos octetos como octetos se confirmen.

En TCP el tamaño de la ventana puede ser variable en función del tiempo, según las necesidades de la red.

¿Como se modifica el tamaño de la ventana en emisor y receptor?

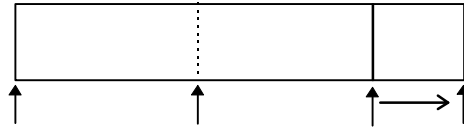
Aumento del tamaño de la ventana:

Emisor:

Desplazar el puntero derecho tantas posiciones, tantos octetos como se incrementa el tamaño de la ventana.

Sumar al puntero derecho el nº de octetos que se incrementa.

Este suma no afecta al funcionamiento del protocolo, ya que simplemente aumentamos el nº de octetos que se pueden recibir.



Receptor:

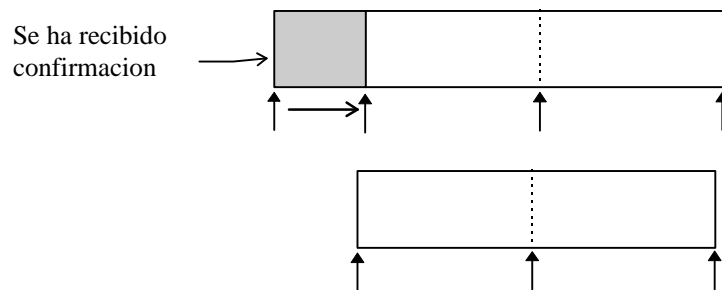
Incrementamos el puntero derecho de la ventana de recepción.
Añadimos mas octetos que se pueden recibir.

Reducción del tamaño de ventana:

Emisor:

Cuando recibimos confirmaciones movemos toda la ventana, así que cuando tenemos que reducir la ventana aprovechamos la confirmaciones de la siguiente forma:

Mantenemos fijo el puntero derecho hasta que se alcance el tamaño de ventana deseado. Solo se mueve el puntero derecho.



Cuando ya tenemos el tamaño deseado, el funcionamiento de la ventana será el mismo de siempre. (Desplazando los 2 punteros).

Si queremos reducir la ventana en 5 y nos llegan 10 confirmaciones, lo que haremos será utilizar la 5 primeras para la reducción, mientras que con las otras 5 el funcionamiento de la ventana será el mismo de siempre.

Receptor:

El puntero izquierdo no se puede mover, ya que podríamos dar por recibidos octetos que no han llegado.

El puntero derecho lo dejaremos fijo y solo desplazaremos el puntero derecho, modificando solo el puntero izquierdo.

TCP solo soluciona problemas entre el emisor y el receptor no en los pasos intermedio.