



Tema 1 | Ethernet

1. Medio Compartido

1.1. Half / Full Duplex

2. Half Duplex

2.1. Colisiones

2.1.1. Algoritmos

2.2. CSMA/CD

2.2.1. Tramas

2.2.2. Formulario y Ejercicios

2.3. Trama Ethernet

Tamaño Trama

2.4. Dirección MAC

2.4.1. Tipos de Direcciones MAC

2.4.2. Ejemplo Dirección MAC

3. Full Duplex

Características

3.1. Hubs y Switches

3.2. Control de Flujo

3.2.1. Switch Backpressure

3.2.2. MAC Control

Histéresis

Colas por Prioridad

3.3. Configuración Automática

4. Gigabit Ethernet

4.1. Half-Duplex

4.1.1. El Problema con la Velocidad

4.1.2. Solución: Carrier Extension

4.1.3. Problema: Baja Utilización

4.1.4. Solución: Frame Bursting

4.2. Avances y Contexto

4.3. Full-Duplex

4.4. Tecnologías de Ethernet

4.4.1. Tramas Jumbo

4.4.2. Power Over Ethernet

4.4.3. Cut Through Switches

4.4.4. Link Aggregation

5. 10 Gigabit Ethernet

1. Medio Compartido

A la hora de conectar ordenadores, se podían utilizar dos tipos de enlaces:

- **Enlaces Punto a Punto**

Cada par de nodos debería conectarse mediante un cable, lo que requiere de muchos enlaces si el número de nodos es elevado.

Esta opción es *inviable*.

- Es seguro y rápido.
- No es escalable.

- **Enlaces de Difusión**

Un único enlace es compartido por varios nodos, de forma que no es necesario una unión por cada nodo.

- Permite broadcast.
- Escala bien.
- Solo un nodo puede utilizar el enlace a la vez.

1.1. Half / Full Duplex

Half Duplex

Se transmite en ambas direcciones, pero no de forma simultánea.

Un dispositivo envía los datos mientras otro los recibe.

Como solo un dispositivo puede transmitir a la vez, se pueden producir *colisiones*.

Full Duplex

Se transmite en ambas direcciones de forma simultánea.

Los dispositivos pueden enviar y recibir información a la vez.

Requiere un "cable" para cada sentido.

2. Half Duplex

2.1. Colisiones

!! Colisiones

Estas ocurren cuando en una red **Half Duplex** dos dispositivos intentan transmitir de forma simultánea. La trama debe ser descartada porque ha tenido interferencias.

2.1.1. Algoritmos

A lo largo de la historia han existido diferentes versiones de protocolos para evitar estas colisiones.

Podemos diferenciar estos algoritmos en dos grupos:

- **Algoritmos con Colisión**

Utilizan un método para detectar las colisiones para minimizar su impacto.

ALOHA, CSMA, CSMA/CD (Ethernet), CSMA/CA (Wi-Fi)

- **Algoritmos sin Colisión**

Eliminan completamente la probabilidad de colisiones al pedir un turno a los nodos para poder transmitir por el

medio.

Token-Ring, Token-Bus, Arbitraje

2.2. CSMA/CD

Es un método utilizado en redes Ethernet para minimizar las colisiones de datos que ocurren cuando dos o más dispositivos intentan transmitir datos al mismo tiempo.

1. El nodo transmisor escucha el medio para asegurarse de que no haya señales de otros nodos transmitiendo.
2. Si el medio está libre, el nodo empieza a transmitir.
Si el medio está ocupado, el nodo espera un tiempo aleatorio y vuelve a intentarlo.
3. Si dos o más nodos comienzan a transmitir al mismo tiempo y se produce una colisión, los nodos detienen la transmisión y envían una señal al resto de nodos para avisar de la colisión.
4. Todos los nodos de la red esperan un tiempo aleatorio antes de volver a intentar transmitir.
El nodo que detectó la colisión espera un tiempo mayor que los demás para asegurarse de que los demás han terminado de transmitir.

2.2.1. Tramas

Una trama está compuesta por varias partes, entre las que se incluyen:

- **Preámbulo** 8 Bytes

Un espacio que se utiliza para sincronizar los relojes de transmisión y recepción en los nodos de la red.

- **Espacio entre Tramas** 96 bits

Es un espacio de longitud variable que se utiliza para proporcionar un tiempo de separación entre las tramas que transmiten en la red.

- Da tiempo a los nodos para que procesen la trama antes de que reciban la siguiente.
- Ayuda a reducir la posibilidad de que se produzcan colisiones en la red.

!! Longitud de Trama

Es necesario establecer un mínimo de longitud de trama para que el algoritmo funcione correctamente.

Esto es así puesto que el algoritmo de detección de colisiones solo funciona si todas las estaciones en la red tienen suficiente tiempo para detectar la señal de otro nodo antes de que finalice la transmisión.

2.2.2. Formulario y Ejercicios

Tiempo Transmisión

Tiempo que tarda en transmitirse una trama por un medio.

$$T_{trans} = \frac{T_{trama}}{V_{trans}}$$

T_{trama} Tamaño Paquete Byte / bit

V_{trans} Velocidad de transmisión Bps / bps

Ejemplo 1 Se quieren transmitir 1500 Bytes a una velocidad de 100Mbps.

Calcula el tiempo de transmisión.

$$T_{trans} = \frac{1500 \cdot 8 \text{ bit}}{100 \cdot 10^6 \text{ bps}} = 0,00012s = 0,12ms$$

segundo s - milisegundo ms - microsegundo μs - nanosegundo ns - picosegundo ps

Tiempo Propagación

Tiempo que tarda en llegar una señal de **A** a **B**.

$$T_{prop} = \frac{distancia}{V_{prop}}$$

distancia Distancia entre nodos **m**

V_{prop} Velocidad de propagación $\frac{2}{3}c$ **m/s**

Ejemplo 2 Se tiene una señal que viaja a través de un cable de 100m de longitud. Calcula el tiempo de propagación.

$$T_{prop} = \frac{100 \text{ m}}{\frac{2}{3}c \text{ m/s}} = 0,0000003333s = 333,3 \text{ ns}$$

Tasa de Paquetes

La cantidad de paquetes de datos que se envían en una red durante un periodo de tiempo. También se suele denotar como *frecuencia*.

$$f = \frac{paquetes}{tiempo}$$

$$f = \frac{1}{T}$$

T Tiempo paquetes **s**

Ejemplo 3 El nodo **A** transmite paquetes consecutivos a **B** de 1000 Bytes. Estos paquetes llegan cada 20 ms. Calcula la *tasa de paquetes*.

En ese caso no nos importa el tamaño de los paquetes.

$$f = \frac{1}{20 \text{ ms}} = \frac{1}{0,02 \text{ s}} = 50 \text{ paq/s}$$

Velocidad de Transmisión

La cantidad de datos que se pueden transmitir en un tiempo determinado.

$$V_{trans} = \frac{T_{trama}}{T_{trans}}$$

T_{trama} Tamaño Paquete **Byte / bit**

T_{trans} Tiempo Transmisión **s**

Ejemplo 3 El nodo **A** transmite paquetes consecutivos a **B** de 1000 Bytes. Estos paquetes llegan cada 20 ms. Calcula la *velocidad de transmisión*.

$$V_{trans} = \frac{1000 \text{ Bytes}}{20 \text{ ms}} = \frac{1000 \cdot 8 \text{ bit}}{20 \cdot 10^{-3} \text{ s}} = 400 \text{ 000 bps} = 400 \text{ kbps}$$

2.3. Trama Ethernet

La trama *Ethernet* está compuesta por diversos campos.

Preámbulo	Direc. Destino	Direc. Origen	Estándar	Datos	CRC
-----------	----------------	---------------	----------	-------	-----

- **Preámbulo** **8 Bytes**

Es un conjunto de bits que delimitan el inicio de una trama.

...1010101010101011 Siempre termina en **11**.

Start of Frame Delimiter *SFD* – 2

No cuenta a la hora de calcular el tamaño de la trama.

- **Dirección Destino** 6 Bytes

Este campo va el primero porque es lo que más nos interesa. Será de gran utilidad en *switches* para aumentar la velocidad de estos. Contiene la dirección MAC del destino.

- **Dirección Origen** 6 Bytes

Este campo contiene la dirección MAC de quien envía la trama.

- **Estándar** 2 Bytes

Ethernet DIX (**Dell** , **Intel** , **Xerox**) Crearon un primer protocolo.

IEEE 802.3 Estandarizó posteriormente el protocolo de estas empresas.

Esto ha provocado que se deban hacer compatibles 2 versiones del mismo protocolo.

Vamos a poder distinguir cual de las dos versiones se está utilizando gracias a los valores que encontremos:

[0..1500] **IEEE 802.3** Lo nuevo, estandarizado por IEEE.

[1500..65536] **Ethernet DIX** Lo que se ha estado usando de manera no estándar siempre.

Normalmente se usa el **Ethernet DIX** , que soporta protocolos como **IP** .

- **Datos** 46 Bytes - 1500 Bytes

Donde están los datos que se quieren transportar.

- **CRC** 4 Bytes

Es usado para detectar errores. Similar al **CHECKSUM** .

Hoy en día no es necesario, puesto que por *Ethernet* casi no hay errores.

Tamaño Trama

El *preámbulo* no cuenta a la hora de calcular el tamaño de la trama.

Tamaño Mínimo $6 + 6 + 2 + 46 + 4 = 64 \text{ Bytes} = 512 \text{ bits}$

Tamaño Máximo $6 + 6 + 2 + 1500 + 4 = 1518 \text{ Bytes} = 12144 \text{ bits}$

2.4. Dirección MAC

Las direcciones MAC no pueden repetirse. Para que esto no ocurra, la **IEEE** controla el reparto de las direcciones MAC.

Lo que esta institución hace es repartir un *prefijo de direcciones MAC* a cada fabricante.

2.4.1. Tipos de Direcciones MAC

- **Unicast** 3C:07:54:1A:6B:3F

Se utiliza para enviar paquetes de datos a un único dispositivo de red.

- **Multicast** 01:00:23:45:67:89

Se utiliza para enviar paquetes de datos a múltiples dispositivos de red al mismo tiempo.

- **Broadcast** FF:FF:FF:FF:FF:FF

Se utiliza para enviar paquetes de datos a todos los dispositivos de la red al mismo tiempo.

2.4.2. Ejemplo Dirección MAC

0	1	2	3	4	5
0011 0101	0111 1011	0001 0010	0000 0000	0000 0000	0000 0001

La dirección MAC está formada por 6 octetos. De estos octetos, los primeros bits del primer octeto nos dan información acerca del tipo de dirección MAC.

Primer Bit 0 _ 1 _

0 Unicast

1 Multicast

Segundo Bit _ 0 _ 1

0 MAC Universal

1 MAC Localmente Administrada

3. Full Duplex

Puedes estar *transmitiendo* y *recibiendo* a la vez, puesto que tienes un canal para cada cada sentido. Esto permite aumentar mucho la estabilidad y velocidad.

Características

- **No hay mínimo en el tamaño de trama.**
Puesto que lo problemas de colisiones desaparecen.
Sin embargo, se respetan por compatibilidad.
- **No hay máximo en el tamaño de trama.**
Se recomienda tener un tamaño mayor que en half-duplex, aunque no demasiado exagerado para no provocar retrasos.
- **Se aumenta la capacidad del canal.**
Si antes teníamos 10 *Mbps* en half, podemos tener 20 *Mbps* en full.
- **Los switches tienen más trabajo.**
Debido al aumento de paquetes, los switches tienen más trabajo que hacer de forma simultánea.

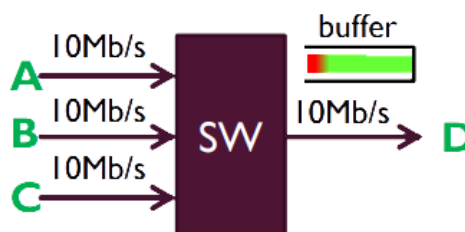
3.1. Hubs y Switches

Ambos dispositivos son utilizados para interconectar dispositivos en la red.

- **Hub**
Es un dispositivo que simplemente repite y envía todos los paquetes de datos que recibe a todos los dispositivos conectados a él. Esto significa que, cuando un paquete de datos llega al hub, este lo envía a todos los dispositivos conectados a él, sin importar si el paquete está destinado a un dispositivo específico o no. Como resultado, los hubs pueden generar tráfico innecesario en la red y disminuir su rendimiento.
- **Switch**
Es un dispositivo que tiene la capacidad de leer la dirección MAC de destino de un paquete de datos y enviarlo solo al dispositivo correspondiente. Esto significa que, cuando un paquete de datos llega al switch, este determina el destino del paquete y lo envía solo al dispositivo que lo solicita. Como resultado, los switches pueden reducir el tráfico innecesario en la red y aumentar su rendimiento.
 - **Wire-Speed**
Decimos que un switch es wire-speed si es capaz de gestionar todos los puertos a la máxima capacidad del cable sin producir retrasos ni saturarse.
 - **No Wire-Speed**
Un switch que no cumple los requisitos para ser wire-speed. Es decir, que tiene dificultades a la hora de soportar el tráfico full duplex en sus puertos.

3.2. Control de Flujo

En algunas ocasiones es fácil que se produzca *congestión*, por lo que se hace necesario implementar un mecanismo que nos permita controlar el flujo que se recibe.



Ejemplo Tenemos tres nodos **A**, **B** y **C** que quieren enviar paquetes al nodo **D**.

Esto provoca que el switch tenga que mandar todo ese tráfico a **D**.

Por mucho que el switch sea wire-speed, y pueda soportar el tráfico, el cable de **D** tiene una velocidad máxima menor a la necesaria.

Como los *buffers* del switch tienen un límite, el switch debe poder pedir a los nodos **A**, **B** y **C** no transmitir tramas por un tiempo.

3.2.1. Switch Backpressure

Método algo cutre utilizado **solo** en **Half-Duplex**. Encontramos dos posibilidades:

- Provocar una colisión para que el nodo transmisor pare.
- Mandar un preámbulo infinito. Es decir, mandamos `01010101...` sin terminar en `11`.

3.2.2. MAC Control

Método actualizado y utilizado en **Full-Duplex**. Debe poder usarse incluso si uno de los nodos no lo implementa, aunque sea de forma menos eficiente.

Se utiliza una trama normal de *Ethernet*, pero en la parte de datos se incluyen algunos apartados.

Además, se hacen algunos cambios en algunos campos.

Preámbulo	Direc. Destino	Direc. Origen	Estándar	Datos		CRC
Direc. Destino	Direc. Origen	Estándar	MAC Control Cód. Operación	MAC Control Parámetros	Reservado	CRC

- **Direc. Destino** **6 Bytes**
Se establece en *MULTICAST*, de forma que sea compatible con dispositivos que no tienen el protocolo. Un switch que no implemente *MAC CONTROL*, lo retransmitirá por todos los puertos.
De esta forma podrá llegar a un dispositivo que sí lo implemente.
- **Estándar** **2 Bytes**
Se establece en $0x8808_6 = 34824_{10}$.
Por tanto se utiliza *Ethernet DIX*.
- **MAC Control Cód. Operación** **2 Bytes**
Aquí se especificará la operación que se quiere mandar.
En este caso, solo existe *PAUSE* `0x0001`.
- **MAC Control Parámetros** **— Bytes**
Aquí se especifican valores para aplicar el *PAUSE*.
`pause_time = [0x0000 - 0xFFFF] + 512 bit_time`
No va en *segundos*, sino que es un número que se multiplica por un tiempo correspondiente a la transmisión de 512 bits en la tecnología que estás usando.

Tecnología	Tiempo
10 Mb/s	0 - 3.36 s
100 Mb/s	0 - 336 ms
1000 Mb/s	0 - 33.6 ms



Esto es lo que se multiplica por el número que se recibe en la trama. El tiempo varía según la tecnología de Ethernet que se esté utilizando.

- **Reservado** **— Bytes**
- **Check** **4 Bytes**

Parámetros + Reservado = 44 Bytes



A la hora de mandar la trama, se puede realizar de dos formas diferentes:

- **Forma “Inteligente”**

El *host* calcula el tiempo que necesita que el *transmisor* pare y, posteriormente, manda el mensaje. Se consigue usar solo **1 trama**.

- **Forma “Tonta”**

El *host* no calcula nada. El *host* envía un mensaje con el tiempo máximo para que el otro espere. Cuando el *host* está libre, manda otra trama para avisar al *transmisor* de que puede proseguir con el envío. Se utilizan **2 tramas**.



Las tramas **PAUSE** tienen **prioridad** sobre el resto.

Esto permite que estas tramas sean mandadas de forma inmediata y no se encolen en el *buffer* de salida. Así se consigue avisar del problema lo antes posible.

Histéresis



Histéresis

Nos permite definir unos márgenes sobre los que actuar. De esta forma no estamos constantemente mandando mensajes de **start / stop**.



Ejemplo

Con la *calefacción*, no nos interesa estar constantemente apagando y encendiendo los radiadores. Solemos establecer un umbral por encima y debajo para evitar esto.

Temp. Deseada: 25°C

Temp. Activar: 24.6°C

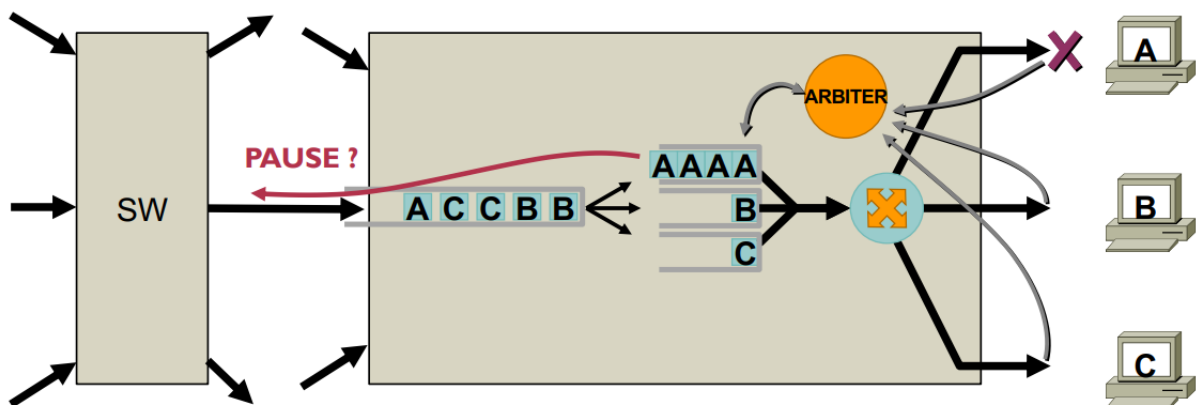
Temp. Desactivar: 25.4

Este valor debe ser mayor que $RTT \cdot V_{trans}$.

RTT Tiempo de ida y vuelta de un mensaje.

V_{trans} Velocidad de transmisión de los mensajes.

Ahora vamos a analizar una situación que puede ocurrir, para saber como debemos actuar.



En esta imagen tenemos un *switch* con 3 colas de salida, cada una para un nodo.
En este caso, la cola de salida hacia el nodo A se ha congestionado.

? ¿Debemos mandar una trama PAUSE?

En esta situación, el mensaje *PAUSE* iría al *switch* de la izquierda. Como consecuencia, este *switch* no nos enviaría nada. Estamos penalizando al resto de nodos (B, C) por un problema con la cola A.
En caso de tener un mayor número de puertos, la situación es aún más clara.



Tipos de Switches

Según su localización, podemos clasificar a los *switches* en dos grupos:

- **Edge**
Son aquellos que están cerca de usuarios finales, en “las afueras” de la red.
- **Core**
Son los que se encuentran en el “corazón” de la red. Interactúa con otros *switches*.

Sabiendo esto, podemos intuir que es útil utilizar el *PAUSE* en la situación presentada cuando estamos conectados con clientes finales.

Si me llegan paquetes de otro *switch*, pidiendo un *PAUSE* estaríamos descartando a infinidad de usuarios.

Por tanto, como norma general, solo nos interesará hacer esto en *switches edge*, pero no en *switches core*.

Colas por Prioridad

Hasta ahora estábamos haciendo una cola por cada uno de los puertos de salida. ¿Y si hacemos colas según la prioridad de cada paquete?

Con esto daríamos a varias colas (con diferentes prioridades) para una misma salida. Esto permite que filtremos tramas según interés.

3.3. Configuración Automática

Un sistema pensado para facilitar la vida al usuario final.

Los dispositivos se encargan de negociar las tecnologías que se van a utilizar. De esta forma el usuario no debe preocuparse por temas de compatibilidad ni de otros aspectos.

Por tanto, negociarán:

1. La Velocidad Máxima

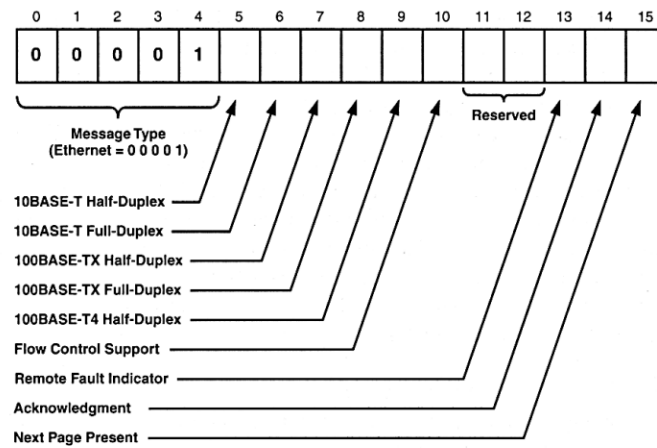
Los dispositivos comparten su velocidad máxima, y de todas, se quedarán con la mínima.
10 Mbps, 100 Mbps, 1000 Mbps.

2. El Modo Duplex

Los dispositivos decidirán si utilizarán Half-Duplex o Full-Duplex, en función de sus capacidades. Se usará Half-Duplex en caso de que alguno no soporte Full-Duplex.

3. El Soporte de Pause

Los dispositivos compartirán si son compatibles con la tecnología *PAUSE*.

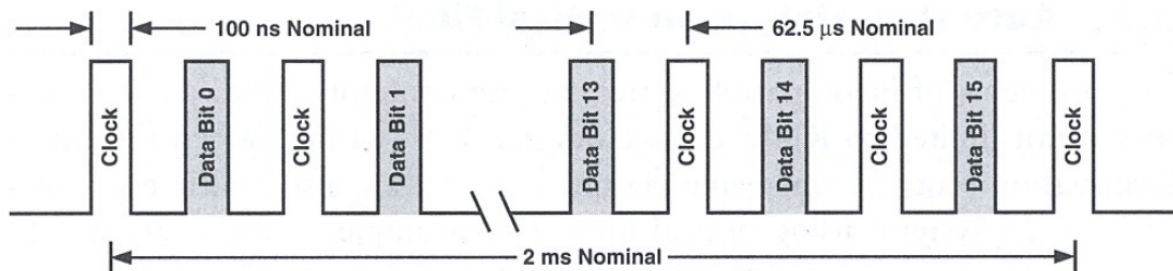


Formato de una trama de negociación.

Estos mensajes son tramas de 16 *bits* que se agrupan en páginas.

Si el bit **Next Page Present** está a **1**, el receptor espera otro mensaje de 16 *bits*.

Es un mensaje que se transmite de forma lenta, para evitar errores.



i = 0

1. Señal de Reloj

2. Bit **i+1**

Se repiten los pasos **1.** y **2.** hasta haber transmitido la página completa.

Esto hace que sea lento, pero muy seguro.

Ejemplo Calcula la velocidad de transmisión (V_{trans}) de estos mensajes.

Sabemos que tarda 2 *ms* porque lo pone en la viñeta, y que son 16 *bits* porque lo hemos estudiado.

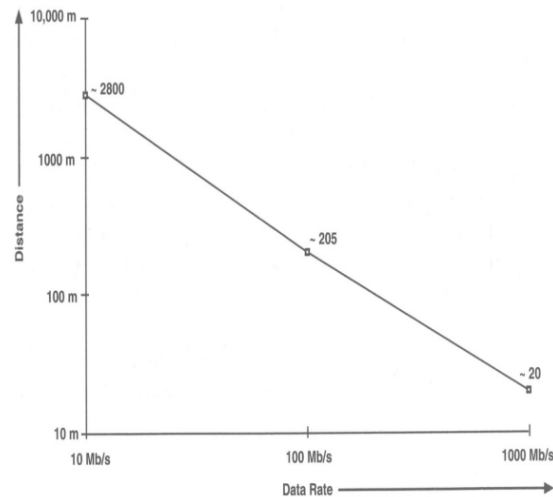
$$T_{trans} = \frac{T_{trama}}{V_{trans}} = \frac{16 \text{ bits}}{2 \text{ ms}} = \frac{16 \text{ bits}}{0,002 \text{ s}} = 8000 \text{ bps} = 8 \text{ Kbps}$$

4. Gigabit Ethernet

Es un protocolo Ethernet especial, puesto que surgió en un momento de transición. Por tanto, este estándar debía seguir dando soporte a *Half-Duplex* y a la vez avanzar en tecnología.

4.1. Half-Duplex

4.1.1. El Problema con la Velocidad



Este gráfico nos muestra como en *Half-Duplex* la distancia a la que podemos detectar colisiones se reduce con la velocidad.

Fast Ethernet (100 Mbps) 205 m

Gigabit Ethernet (1000 Mbps) 20 m

4.1.2. Solución: Carrier Extension

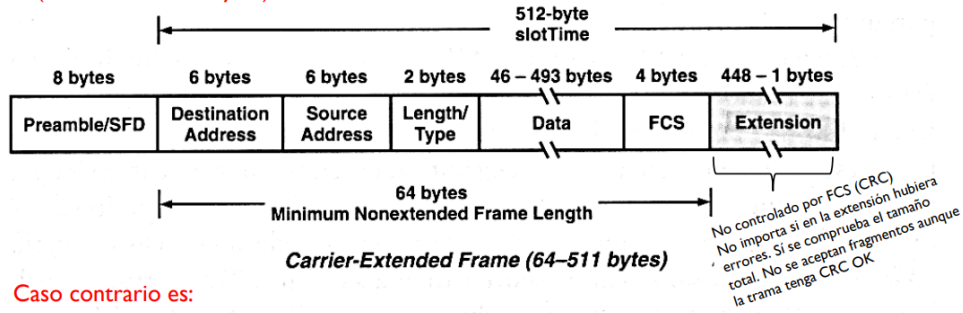
Con este mecanismo conseguimos implementar cambios y solucionar el problema sin realmente tener que cambiar nada en los dispositivos existentes.

Los *sistemas operativos* siguen viendo *Ethernet* con tamaño mínimo de 64 *Bytes*.

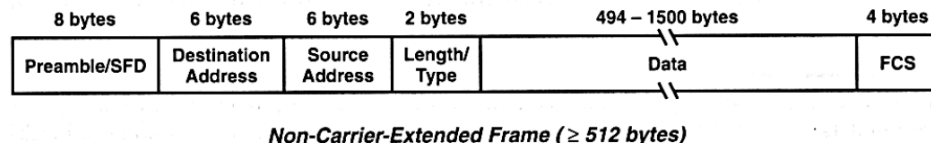
A la hora de transmitir, la trama se hace artificialmente más grande para evitar fallos. Por tanto, si se mandan menos de 512 *Bytes*, la trama se ampliará para alcanzar esos 512 *Bytes*.

Con esto se consigue cambiar solamente las tarjetas de red, que deben conocer esto para ser compatibles con *Gigabit Ethernet*, pero los *niveles superiores* no conocen esta estrategia.

Si (frame size < 512 bytes) entonces en formato es:



Caso contrario es:



4.1.3. Problema: Baja Utilización

Carrier Extension soluciona el problema de detectar colisiones, pero provoca que la utilización de la red pueda ser muy baja.

Si transmitimos solamente 64 Bytes de información en 512 Bytes de trama, realmente estamos teniendo una utilización muy baja.

$$Utilización = \frac{64}{512} = 12.5\%$$

4.1.4. Solución: Frame Bursting

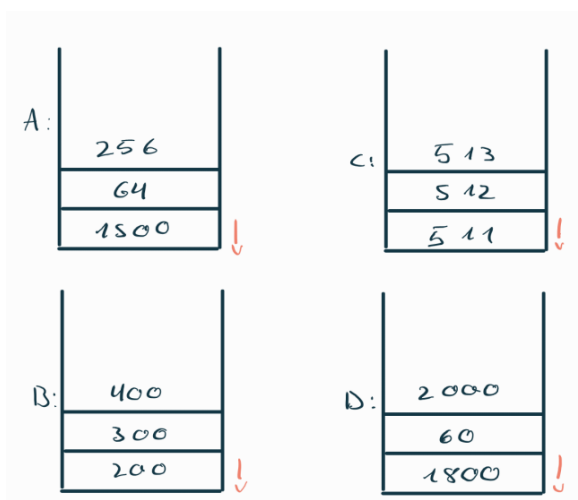
Cuando un nodo envía una trama, suele necesitar enviar otras tramas después.

Trama Ethernet

Preámbulo	Direc. Destino	Direc. Origen	Estándar	Datos	CRC
-----------	----------------	---------------	----------	-------	-----

Es en la zona de **Datos** donde se pueden llegar a encadenar otras tramas. Como mucho, se alargará hasta 8192 Bytes, para no acaparar todo el ancho de banda todo el rato.

Ejercicio Organiza las tramas de estos transmisores utilizando *Frame Bursting*.



Notas

Solo podemos encadenar las tramas si no superamos los 8192 Bytes en total.

Solo se encadena si la primera trama es menor a 512 Bytes.

Una trama no puede tener menos de 64 Bytes ni más de 1518 Bytes.

▼ **Nodo A**



No necesita ningún tipo de extensión, así que todo puede enviarse de forma conjunta directamente.

▼ **Nodo B**



Vamos a añadir una extensión de 312 a la primera trama. Con esto, podemos encadenar el resto de tramas para enviarlas de forma conjunta.

▼ **Nodo C**



Como 511 es menor que 512, añadimos una extensión de 1. Ahora podemos encadenar el resto de tramas para enviarlas de forma conjunta.

▼ **Nodo D**



Este caso no podría darse, puesto que una trama de 2000 no puede existir. Nunca podría enviarse una trama mayor a 1518.

4.2. Avances y Contexto

Los *hubs* han sido completamente sustituidos por *switches*, que experimentaron una bajada de precios muy notoria. Hoy en día es casi imposible encontrar un *hub*.

En 10GE no hay soporte para *Half-Duplex*, de ahí viene que su creador dijese:

| *Ethernet no es Ethernet*

No es necesario tener CS, ni MA, ni CD.

4.3. Full-Duplex

Encontramos diferentes cables para *Gigabit Ethernet*, entre los que destacan:

- **1000 Base-X**

Solamente usado en empresas, debido a su elevado precio.

Es fibra óptica, con **2 cables**, uno para transmitir, y otro para recibir.

Existen *Multi-Mode* y *Single-Mode*, siendo este último el mejor y más caro por, sobre todo, aumentar la distancia a la que puede usarse.

- **1000 Base-SX** **Multi-Mode**

- **1000 Base-LX** **Multi-Mode** **Single-Mode**

- **1000 Base-CX**

- **1000 Base-T** **Cat-5** **Cat-5e** **Cat-6** **Cat-7** **Muy Usado**

Usado por el usuario normal, puesto que es mucho más barato.

Necesita utilizar **8 cables** para alcanzar **1Gbps**.

- **1000 Base-TX** **Cat-6** **Cat-7**

Tan solo necesita **4 cables** para alcanzar **1Gbps**.

Cable Tipo T **RJ-45**

Different Ethernet Categories

	Category 3	Category 5	Category 5e	Category 6	Category 6a	Category 7
Cable Type	UTP	UTP	UTP	UTP or STP	STP	S/FTP
Max. Data Transmission Speed	10 Mbps	10/100/1000 Mbps	10/100/1000 Mbps	10/100/1000 Mbps	10,000 Mbps	10,000 Mbps
Max. Bandwidth	16 MHz	100 MHz	100 MHz	250 MHz	500 MHz	600 MHz

4.4. Tecnologías de Ethernet

4.4.1. Tramas Jumbo

Las tramas jumbo son tramas Ethernet con una **carga de datos mayor** que el tamaño máximo permitido en las tramas Ethernet estándar. Mientras que las tramas Ethernet estándar tienen un tamaño máximo de carga útil de **1500 Bytes** ($1518 B - 18 B$), las tramas jumbo aceptan tramas de hasta **9000 Bytes**.

Las *tramas jumbo* se utilizan en **redes de alta velocidad y con alto tráfico de datos** para:

- Reducir la sobrecarga del procesamiento de la trama.
- Mejorar la eficiencia de la red.

Al transmitir grandes cantidades de datos a través de tramas jumbo, se reduce el número de tramas que se necesitan para enviar los mismos datos, lo que a su vez reduce el procesamiento de las cabeceras de la trama y mejora la eficiencia de la red.

4.4.2. Power Over Ethernet



Gracias a **1000 Base-TX**, que solo necesita utilizar **4 cables** para conseguir su máxima velocidad, podemos destinar los otros **4 cables** libres para alimentar el dispositivo.

Esto es útil en algunos casos cuando nos es más cómodo alimentarlo con un solo cable.

4.4.3. Cut Through Switches

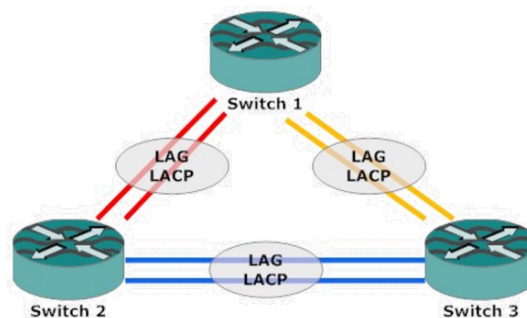
Este tipo de *switches* en lugar de esperar a que se complete la recepción de la trama (como hacen todos), comienzan a retransmitir la trama en cuanto se recibe la información de la dirección de destino.

Ya sabemos que en una trama *Ethernet*, el primer campo que recibimos es la dirección de destino, y estos *switches* se apoyan en ello.

4.4.4. Link Aggregation

En algunas situaciones tenemos demasiado tráfico entre *switches*, lo que resulta en altos tiempos de espera. Una de las opciones que tenemos es usar *link aggregation*, que nos permite conectar varios cables entre *switches* para tener un mayor ancho de banda.

Dentro del *switch* debemos encargarnos de configurar el enlace como único. De esta forma el *switch* no ve varios cables, sino que ve un enlace, y repartirá entre todos los cables los paquetes que se deban enviar.



5. 10 Gigabit Ethernet

- La capa MAC y el formato de trama se mantiene.
 - El tamaño de las tramas no varía. **64 B - 1518 B**
 - Las Jumbo-Frames no son un estándar.
- **No hay soporte para *Half-Duplex*.**
- No hay algoritmos de control de colisiones.
- Soporta velocidades de **10 Gbps**.
- Se utiliza para conectar:
 - Espacios alejados.
 - Servidores de base de datos, u otros servicios importantes.