

BLOQUE IV.

Nivel de enlace de datos

ENTRAMADO.

CORRECCIÓN DE ERRORES.



Contenidos

1. Introducción
 1. Funciones de un protocolo de enlace de datos
2. Entramado
 1. Protocolos orientados a carácter
 2. Protocolos orientados a bit
3. Corrección de errores
 1. Códigos de control de errores
 2. Códigos polinómicos
4. Técnicas de control de flujo y protocolos de control de errores

Contenidos

5. Protocolos de Control de Acceso al Medio
6. Direccionamiento
7. Hubs, Puentes y Conmutadores
8. Protocolo STP
9. Ejemplo de protocolos de nivel de enlace de datos
 1. HDLC
 2. PPP

Contenidos

1. Introducción

1. Funciones de un protocolo de enlace de datos

2. Entramado

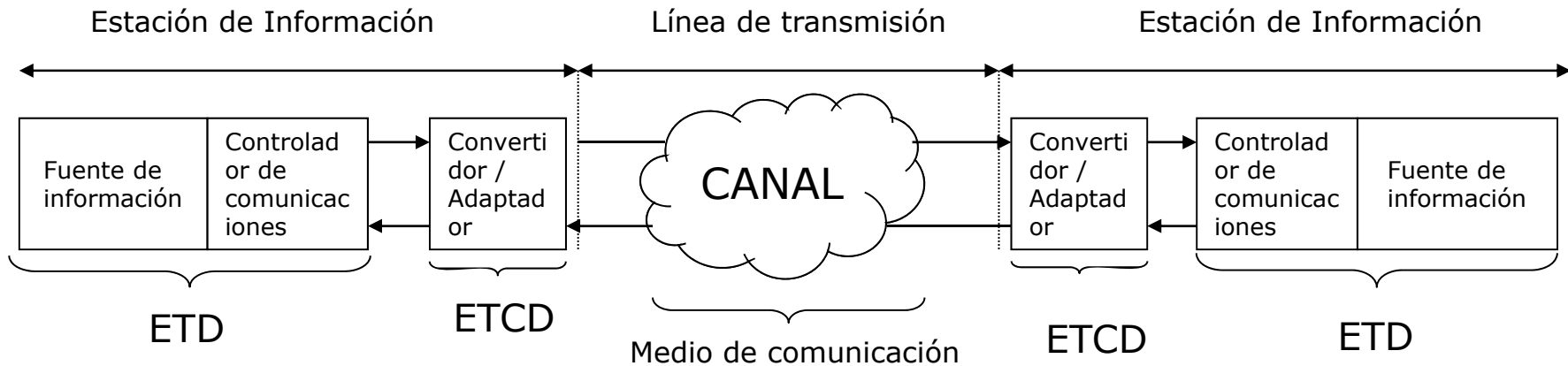
1. Protocolos orientados a carácter
2. Protocolos orientados a bit

3. Corrección de errores

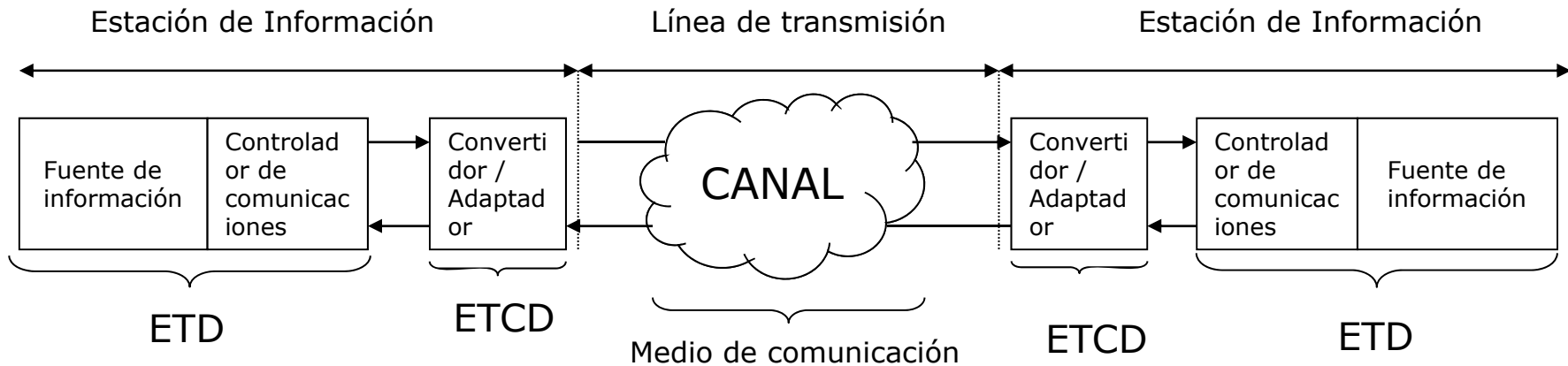
1. Códigos de control de errores
2. Códigos polinómicos

4. Técnicas de control de flujo y protocolos de control de errores

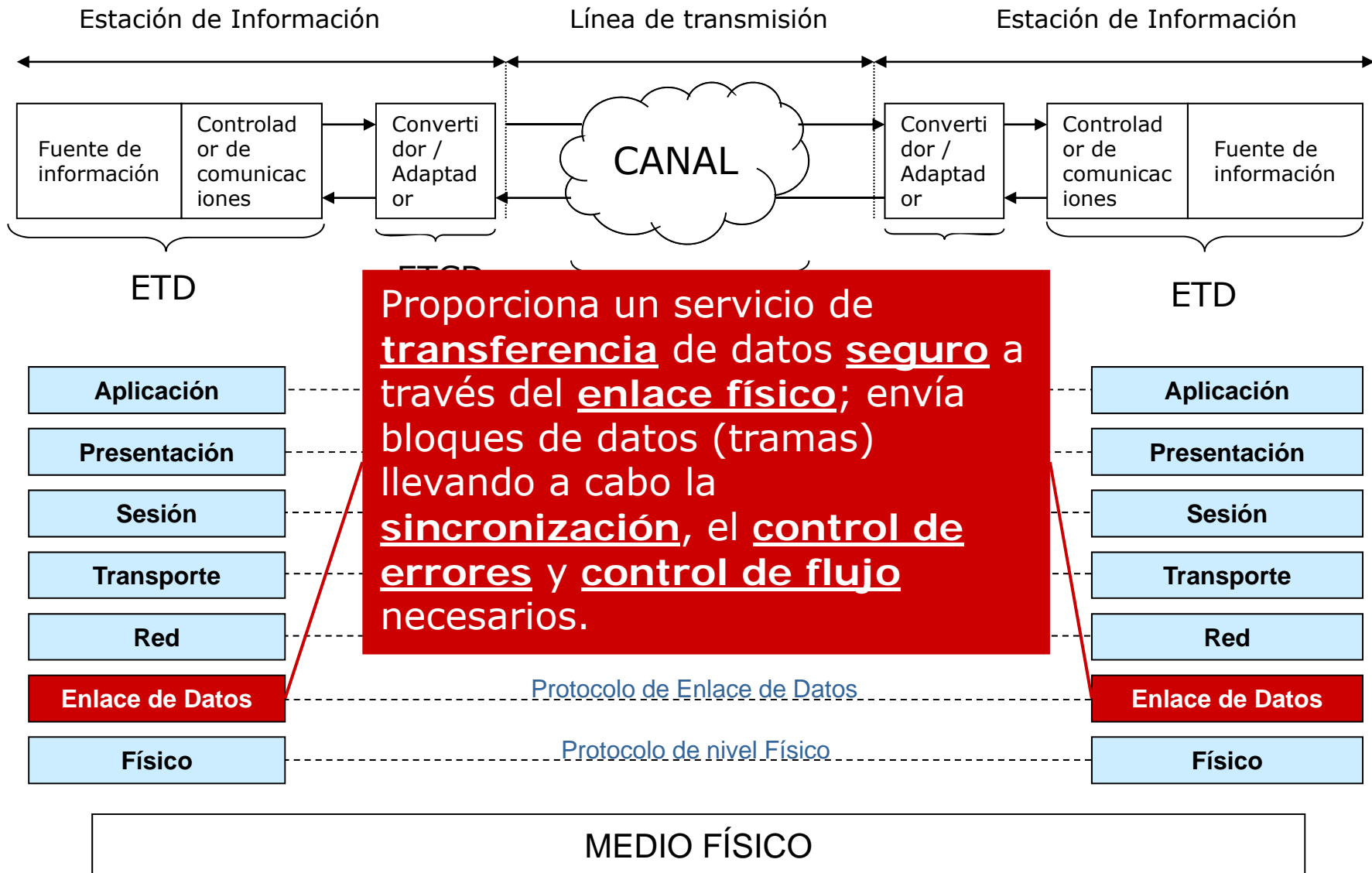
1. Introducción



1. Introducción



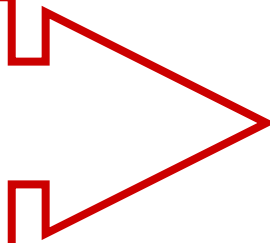
1. Introducción



1. Introducción

Para una comunicación efectiva ¿basta con un enlace físico?

NO, porque se producen errores



Es necesario un mecanismo de control que garantice el intercambio de información

Capa de enlace de datos se encarga de proporcionar una comunicación fiable y eficiente entre dos máquinas adyacentes ("físicamente conectadas"), a través de un conjunto de procedimientos para el establecimiento, mantenimiento y desconexión de las comunicaciones

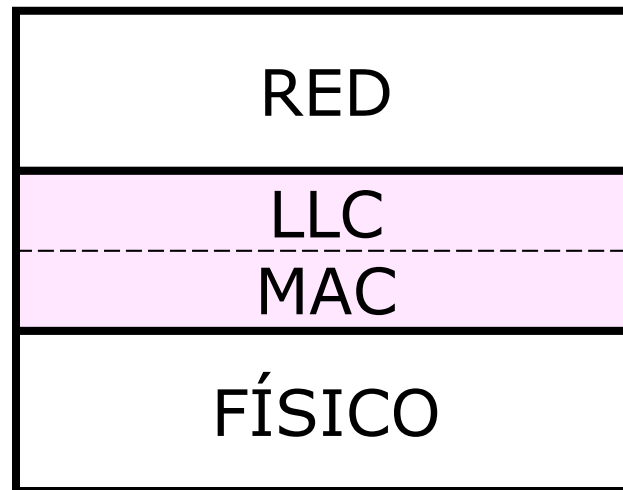
Envío de bloques de información

Control de la transferencia de información

Detección, corrección y recuperación de errores

1. Introducción

- En las redes de área local, el nivel de enlace de datos se divide en dos subcapas:
 - LLC, *Logical Link Control*. Control de flujo y control de errores.
 - MAC, *Medium Access Control*. Control de la transmisión en un medio compartido



Contenidos

1. Introducción

1. Funciones de un protocolo de enlace de datos

2. Entramado

1. Protocolos orientados a carácter

2. Protocolos orientados a bit

3. Corrección de errores

1. Códigos de control de errores

2. Códigos polinómicos

4. Técnicas de control de flujo y protocolos de control de errores

1.1 Funciones de un protocolo de enlace de datos

□ Sincronización de la trama

- El protocolo de enlace de datos tiene que ser capaz de identificar el principio y el final de cada trama:
 1. Utilizar una secuencia de bits que identifiquen el principio de la trama y otra secuencia de bits que identifiquen el final de la misma.
 2. Utilizar una secuencia de bits que identifiquen el principio de la trama, seguida de otro indicador o campo que especifique la longitud de la trama

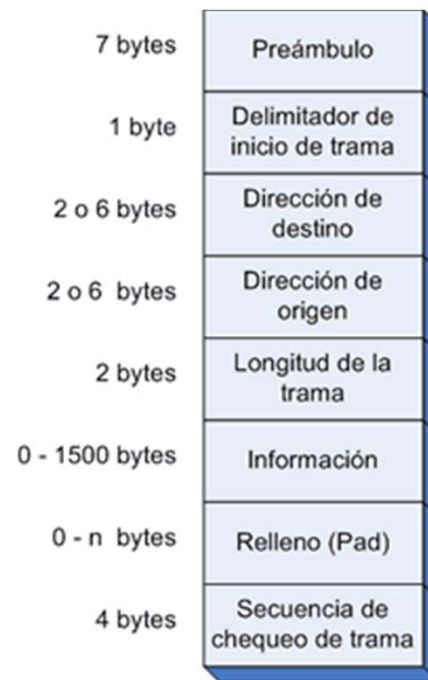
Principio de trama	Trama	Fin de trama
--------------------	-------	--------------

Principio de trama	Longitud	Trama
--------------------	----------	-------

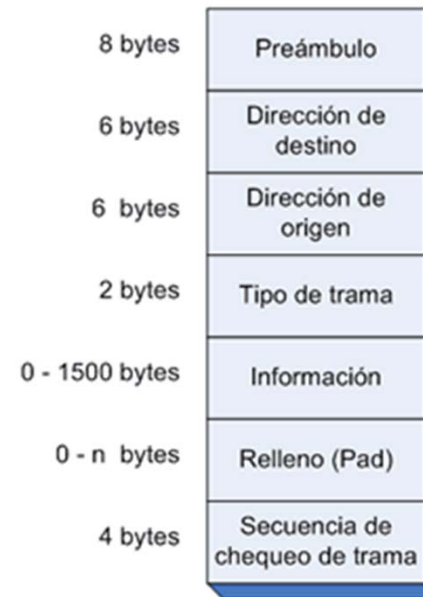
1.1 Funciones de un protocolo de enlace de datos

□ Entramado:

- Definir el formato y tamaño de cada uno de los campos en que se divide la trama, así como el significado de esos campos



Formato de la trama IEEE 802.3

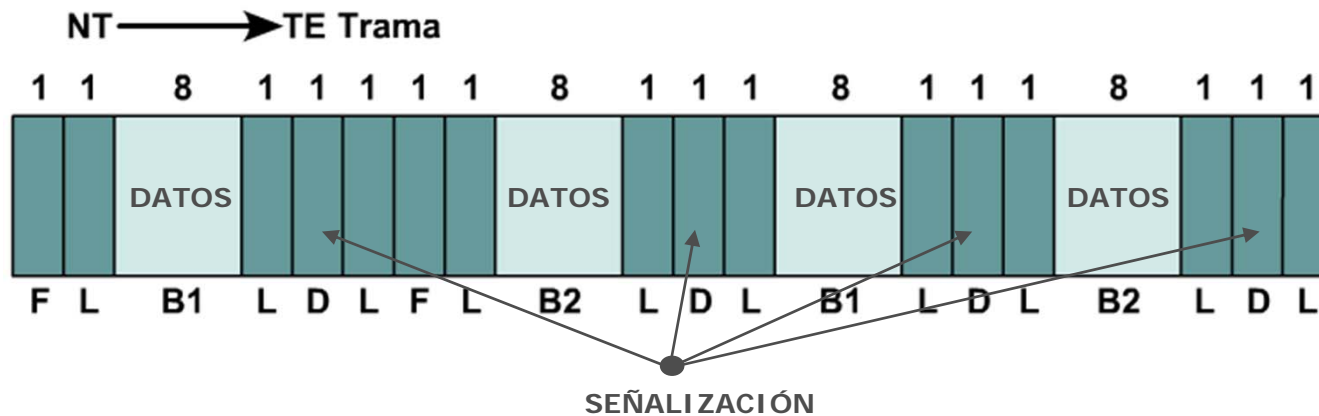


Formato de la trama Ethernet

1.1 Funciones de un protocolo de enlace de datos

□ Control y datos sobre el mismo enlace:

- En ocasiones, no se tiene un enlace físicamente separado para señales de control
- Se necesita un receptor capaz de diferenciar entre información de control y datos

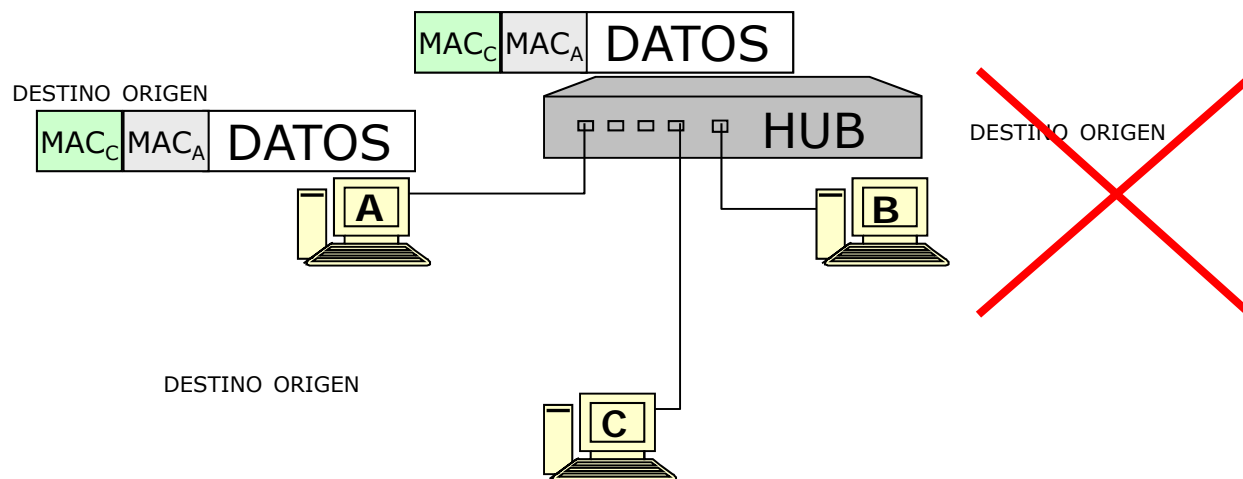


ENVÍO DE DATOS Y
SEÑALIZACIÓN EN RDSI

1.1 Funciones de un protocolo de enlace de datos

□ Direccionamiento:

- Identificación del origen y destino de las tramas, para permitir la transferencia de la información (direcciones físicas)
- **Dirección física** ⇒ **dirección MAC**
 - Ejemplo: 00-21-2F-67-BA-AA
- Dirección lógica ⇒ dirección IP (dirección de nivel de red)
 - Ejemplo: 212.124.40.25



1.1 Funciones de un protocolo de enlace de datos

□ Gestión del enlace:

- Inicio, mantenimiento y terminación del intercambio de información:

1. **Coordinación centralizada**: la estación de enlace primaria se encarga de sondear a las demás para ver si tienen datos que enviar.

El sondeo (*polling*) se puede hacer en función de ciertos pesos, para organizarlo todo mejor en cuanto a retardos y caudales de información.

2. **Coordinación por contienda**: es este caso todas las estaciones pueden transmitir en cualquier momento, la coordinación es distribuida.

Las distintas estaciones tienen que competir por conseguir el enlace.

1.1 Funciones de un protocolo de enlace de datos

□ Recuperación de anomalías:

- Incluye el control de todas las situaciones imprevistas que pueden presentarse a lo largo de una transmisión
- Se suele controlar mediante plazos de espera y limitación del número de reintentos

1.1 Funciones de un protocolo de enlace de datos

□ Control de errores:

- Detección, corrección y recuperación de errores
- Las técnicas más utilizadas en la recuperación de errores son:
 1. Control de errores FEC, en cuyo caso se intenta recuperar el error en destino
 2. Petición automática de retransmisión (ARQ): el error no puede corregirse en destino, lo que se hace es pedir al emisor que vuelva a enviar la trama errónea

1.1 Funciones de un protocolo de enlace de datos

□ Control de flujo

- Se pretende asegurar que, cuando el emisor envía información, el receptor tiene recursos suficientes para recibirla ≡ **adaptación de velocidades**

1. **Parada y espera** (*Stop & Wait*): se usan señales de confirmación (ACK).

2. **Ventana Deslizante** (*Sliding Window*): el emisor envía varias tramas sin esperar confirmación, tantas como quepan en la ventana de transmisión.

El receptor autoriza en cada instante el envío de un determinado número de tramas (vaciando la ventana).

El receptor renovará las autorizaciones según tenga disponibilidad para su recepción.

Contenidos

1. Introducción

1. Funciones de un protocolo de enlace de datos

2. Entramado

1. Protocolos orientados a carácter
2. Protocolos orientados a bit

3. Corrección de errores

1. Códigos de control de errores
2. Códigos polinómicos

4. Técnicas de control de flujo y protocolos de control de errores

2. Entramado

- Protocolos orientados a carácter (años 60):
 - BSC (Comunicación síncrona binaria, de IBM)

- Protocolos orientados a bit (recientes):
 - HDLC (*High Level Data Link Control*, ISO)
 - ADCCP (*Advanced Data Communications Control Procedure*, ANSI)
 - LAP-B (*Link Access Procedure Control Balanced*, X.25)
 - LAP-D (*Link Access Procedure D-Chanel*, UIT, RDSI)

Contenidos

1. Introducción

1. Funciones de un protocolo de enlace de datos

2. Entramado

1. Protocolos orientados a carácter

2. Protocolos orientados a bit

3. Corrección de errores

1. Códigos de control de errores

2. Códigos polinómicos

4. Técnicas de control de flujo y protocolos de control de errores

2.1 Protocolos orientados a carácter

- Un protocolo orientado a carácter es aquel en el que **los mensajes se componen de un conjunto de caracteres de un determinado código** (p.e., ASCII)
 - Cada carácter, tanto de información como de control, tiene un significado específico y único
 - Los caracteres de control se clasifican según tres categorías:
 - ① Delimitadores de bloque
 - ② Controladores de diálogo
 - ③ Transmisión/Recepción transparente

2.1 Protocolos orientados a carácter

1 Delimitadores de bloque:

- SYN (*Synchronous Idle*), mantiene el sincronismo en la transmisión de los caracteres, alerta al receptor de la llegada de una trama
- SOH (*Start Of Heading*), indica el principio del mensaje
- STX (*Start Of Text*), indica el comienzo de un bloque de información dentro del mensaje
- ETX (*End Of Text*), indica el final de un bloque de información y el final de un mensaje
- ETB (*End Of Transmisión Block*) Bloque de final de transmisión

2.1 Protocolos orientados a carácter

② Controladores de dialogo:

- EOT (*End Of Transmisión*): la transmisión se ha terminado y se puede liberar el enlace
- ENQ (*Enquery*): se solicita una respuesta urgente de la estación adyacente
- ACK (*Affirmative acknowledge*): indicación de que se ha recibido bien un bloque de información
- NAK (*Negative Acknowledge*): indicación de que se ha recibido de forma incorrecta un bloque de información

2.1 Protocolos orientados a carácter

③ Transmisión/Recepción transparente:

- DEL (*Data Linkescape*): cambia el significado de los caracteres de control que le siguen, para que las estaciones puedan enviar información coincidente con los propios caracteres de control

Contenidos

1. Introducción

1. Funciones de un protocolo de enlace de datos

2. Entramado

1. Protocolos orientados a carácter

2. Protocolos orientados a bit

3. Corrección de errores

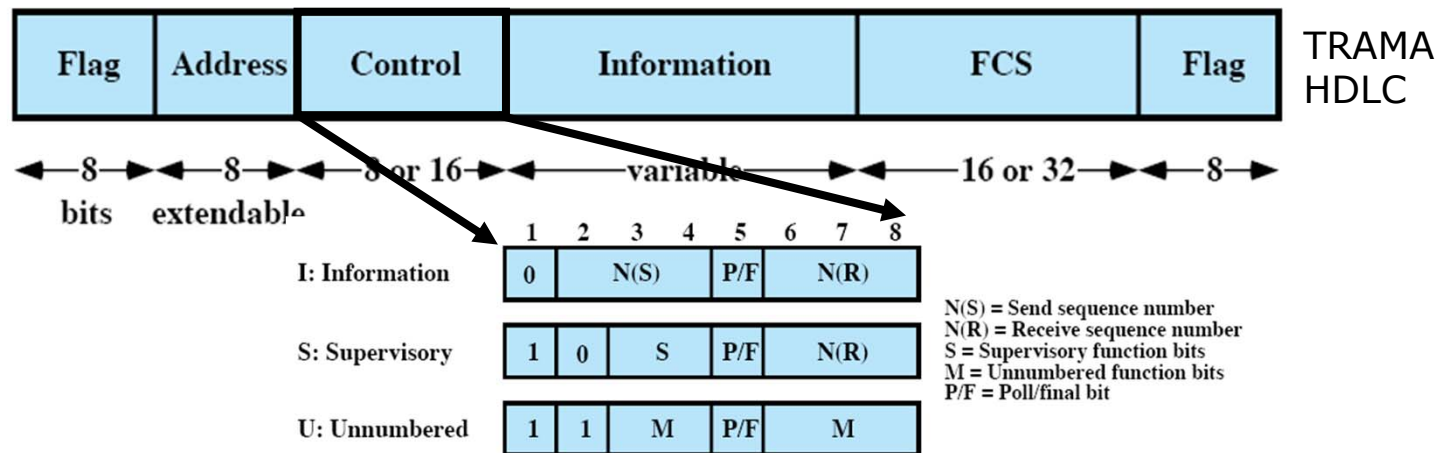
1. Códigos de control de errores

2. Códigos polinómicos

4. Técnicas de control de flujo y protocolos de control de errores

2.2 Protocolos orientados a bit

- ❑ Son protocolos más recientes que los anteriores
- ❑ No están asociados a ningún código en particular ⇒ independientes del código
- ❑ Muy eficientes: un único bit (o un grupo reducido de bits) puede proporcionar información de control
- ❑ Se pueden usar tramas más cortas
- ❑ No existe el problema de transparencia de los protocolos orientados a carácter



Contenidos

1. Introducción

1. Funciones de un protocolo de enlace de datos

2. Entramado

1. Protocolos orientados a carácter
2. Protocolos orientados a bit

3. Corrección de errores

1. Códigos de control de errores
2. Códigos polinómicos

3. Corrección de errores



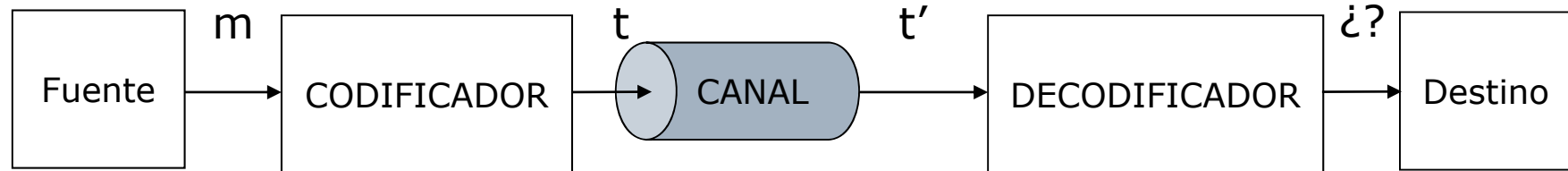
$M \neq M'$ → ¡ERRORES!

- ❑ Una forma de proteger la información que se quiere transmitir, frente a los errores que introduce el canal, es añadir redundancia
- ❑ **La redundancia de un código** \equiv la diferencia entre la longitud de la información útil y la de la secuencia que se envía realmente
- ❑ Un código más redundante:
 - Es menos eficiente
 - Es más fiable

3. Corrección de errores

- Los dígitos que no transmiten información se emplean como detectores ó correctores de errores
- Sus estructuras son distintas, depende del tipo de errores que se quieran tratar
- En la actualidad
 - La generación de códigos se realiza mediante procesos algebraicos que permiten la implementación práctica de codificadores y decodificadores mediante autómatas lineales y lógica combinacional

3. Corrección de errores



- Cuando el mensaje llega al decodificador:
 - Que t' coincida con una palabra código y por tanto se decodificará como m'
 - Que t' no sea una palabra código:
 1. Descartar el mensaje sin más
 2. Pedir retransmisión
 3. Corregir si dispone de la lógica adicional para hacerlo
 - Normalmente se utiliza una combinación de las tres opciones

Contenidos

1. Introducción

1. Funciones de un protocolo de enlace de datos

2. Entramado

1. Protocolos orientados a carácter
2. Protocolos orientados a bit

3. Corrección de errores

1. Códigos de control de errores

2. Códigos polinómicos

4. Técnicas de control de flujo y protocolos de control de errores

3.1 Códigos de control de errores

- Ejemplos de códigos redundantes:
 - Códigos de control de paridad
 - Códigos polinómicos
 - Códigos Hamming

3.1 Códigos de control de errores

□ Códigos de control de paridad:

- Añaden un bit adicional a cada carácter que se transmite (bit de paridad)
- Paridad par ó paridad impar
- Pueden detectar todos los errores en un único bit
 - Detectar errores de ráfaga si el número total de errores en cada unidad de datos es impar

3.1 Códigos de control de errores

□ Códigos Hamming:

- Distancia Hamming \equiv número mínimo de bits en que se diferencian dos palabras código válidas
 - Para detectar n errores \Rightarrow un esquema de codificación con distancia Hamming de $n+1$
 - Para corregir n errores \Rightarrow un esquema de codificación con distancia Hamming de $2n+1$
- Los códigos Hamming son, por tanto, códigos detectores y correctores
- Matemáticamente son interesantes porque permiten hacer un estudio de los errores en función de la distancia entre palabras código
- No son muy utilizados

3.1 Códigos de control de errores

□ Códigos Polinómicos:

- Códigos de redundancia cíclica (CRC, *Cyclic Redundancy Check*)
- La cadena de bits que se envía se ve como un polinomio cuyos coeficientes son los valores 0 y 1 en la cadena de bits, con operaciones en la cadena de bits interpretadas como aritmética polinómica
- Son códigos detectores de errores, no correctores, si bien, pueden detectar un número considerable de patrones de errores

Contenidos

1. Introducción

1. Funciones de un protocolo de enlace de datos

2. Entramado

1. Protocolos orientados a carácter
2. Protocolos orientados a bit

3. Corrección de errores

1. Códigos de control de errores

2. Códigos polinómicos

4. Técnicas de control de flujo y protocolos de control de errores

3.2 Codigos polinómicos

$M(x) \equiv$ Bloque de información de k bits
 $G(x) \equiv$ Polinomio Generador

GENERAMOS

n bits FCS, Frame Check Sequence

TALES QUE

$M(x) || \text{FCS (n+k bits)} \equiv T(x)$ Divisible por $G(x)$

Canal de
Comunicación

$T'(x)$

¿ $T'(x)$ es divisible
por $G(x)$?

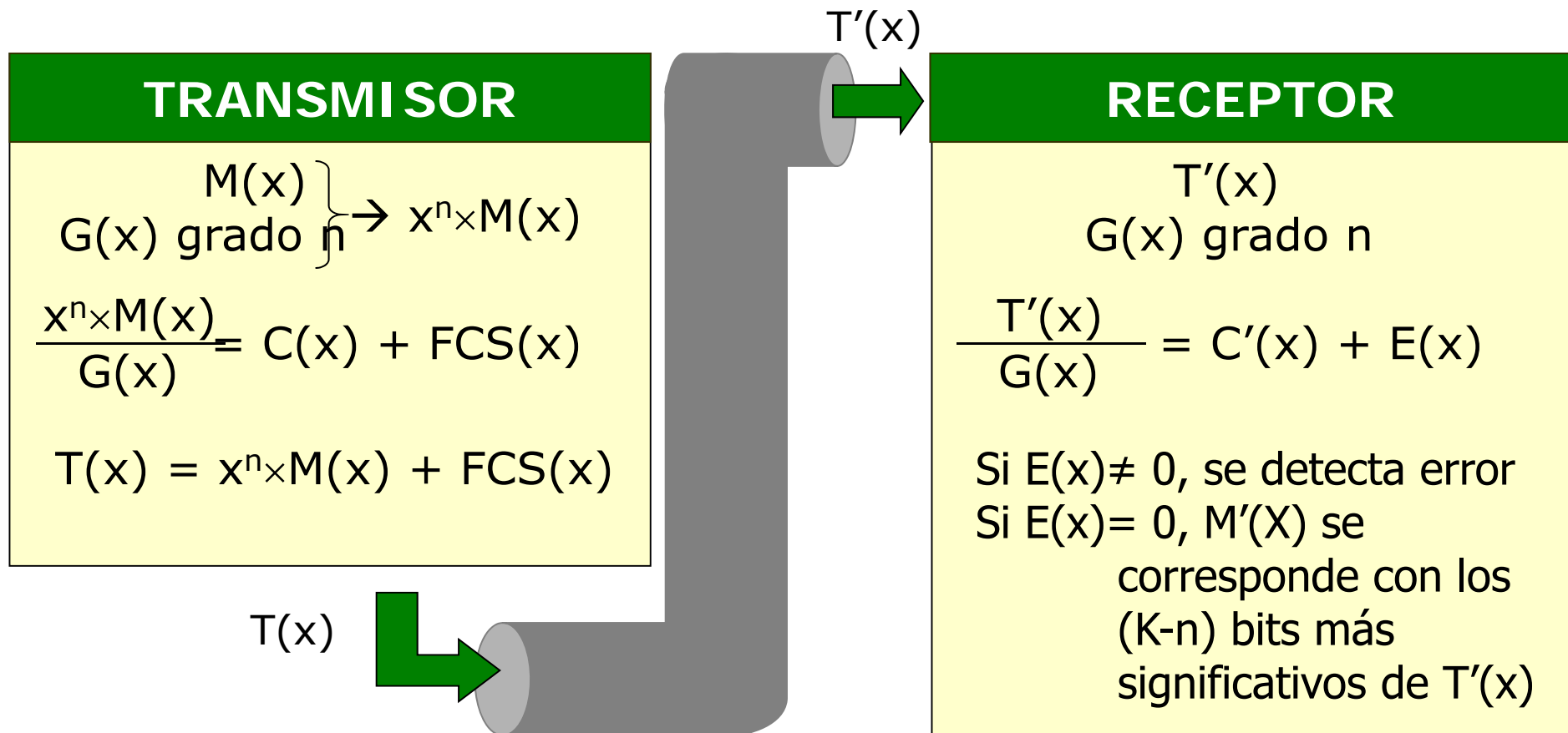
SI \Rightarrow No hay ERROR
No \Rightarrow ERROR

3.2 Codigos polinómicos

- Este proceso se puede explicar de tres formas distintas:
 1. Mediante aritmética módulo 2 que hace uso de sumas binarias sin acarreo, equivalente a la operación OR-exclusiva
 2. Usando lógica digital, implementando un circuito divisor formado por puertas OR-exclusivas y un registro de desplazamiento
 3. **Mediante polinomios, haciendo la correspondencia entre los bits de los números binarios y los coeficientes del polinomio**

3.2 Codigos polinómicos

- Cálculo de FCS



3.2 Codigos polinómicos

- Con una elección adecuada del polinomio generador $G(x)$ se pueden **detectar** los siguientes errores:
 - Todos los errores simples (1 bit)
 - Todos los errores dobles si $G(x)$ tiene al menos tres unos
 - Cualquier número impar de errores si $G(x)$ tiene el factor $(x+1)$
 - Cualquier error a ráfagas en la que la longitud de la ráfaga sea menor que la longitud del polinomio generador (menor o igual que la longitud de FCS)
 - La mayoría de las ráfagas de mayor longitud
 - Para patrones de error equiprobables y ráfagas de longitud $n+1$ (n longitud de FCS), la probabilidad de que el resto $R(x)$ sea divisible por $G(x)$ es $1/2^{n-1}$ y para ráfagas mayores es $1/2^n$

3.2 Codigos polinómicos

□ Polinomios estandarizados:

■ CRC-12 = $x^{12} + x^{11} + x^3 + x + 1$

■ CRC-16 = $x^{16} + x^{15} + x^2 + 1$

■ CRC-ITU-T = $x^{16} + x^{12} + x^5 + 1$

■ CRC-32 =
 $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

□ La elección de uno u otro dependerá de las características que se quieran potenciar

□ El CRC se implementa en *hardware* y se incluye en la tarjeta de red

Resumen

Capa de enlace de datos se encarga de proporcionar una comunicación fiable y eficiente entre dos máquinas adyacentes ("físicamente conectadas"), a través de un conjunto de procedimientos para el establecimiento, mantenimiento y desconexión de las comunicaciones

Sincronización de la trama

Direccionamiento

Control-Datos sobre mismo enlace

Control de flujo

Recuperación de anomalías

Gestión del enlace

Entramado

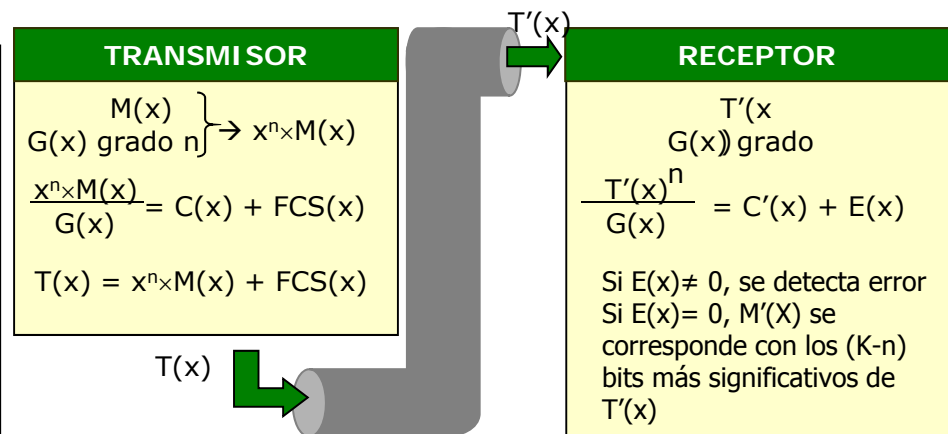
Protocolos orientados a carácter: mensajes compuestos por conjunto de caracteres de un determinado código (delimitadores de bloque, control de diálogo, tx/rx transparente)

Protocolos orientados a bit: no asociados a código, un bit puede proporcionar información.

Control de errores

Redundancia de código (+ redundante \Rightarrow - eficiente + fiable): paridad, hamming, polinómicos

Códigos polinómicos: detectores, no correctores. Cadena de bits que se envía vista como polinomio, coeficientes son valores 0 y 1 en la cadena de bits, con operaciones en la cadena de bits interpretadas como aritmética polinómica



Próximo día

1. Introducción

1. Funciones de un protocolo de enlace de datos

2. Entramado

1. Protocolos orientados a carácter
2. Protocolos orientados a bit

3. Corrección de errores

1. Códigos de control de errores
2. Códigos polinómicos

4. Técnicas de control de flujo y protocolos de control de errores

Próximo día

5. Protocolos de Control de Acceso al Medio
6. Direccionamiento
7. Hubs, Puentes y Conmutadores
8. Protocolo STP
9. Ejemplo de protocolos de nivel de enlace de datos
 1. HDLC
 2. PPP