

# Del Modelo OSI: CAPA ENLACE

## Propósito:

La capa de enlace provee medios de procedimientos y funcionales para **establecer, mantener y liberar conexiones a través de la red**, transferir unidades de servicios de datos a través de entidades de red y para transferir unidades de datos del servicio de data link.

Una conexión de data link puede ser construida bajo conexiones físicas.

La capa detecta y posiblemente corrige errores ocurridos en la capa física.

## Funciones :

- 1) Establece y libera las conexiones de enlace de datos
- 2) Data link service unit mapping .
- 3) Data link splitting.
- 4) Delimitación y sincronización .
- 5) Secuencia de control.
- 6) Detección de error.
- 7) Error recovery .
- 8) Identificación y cambio de parámetros.
- 9) Control del data circuit interconnection .
- 10) Management de la capa.

Yo ya tengo mi secuencia de bits ahora me interesa es realizar estas funciones sobre esa cadena de bits, sobre todo de la 4 a la 7

Ahora tenemos que armar un mensaje con esa cadena de bit que tenemos, ese armado de mensaje tiene que tener ciertas cosas, esas con las funciones de la capa 2

## Servicios

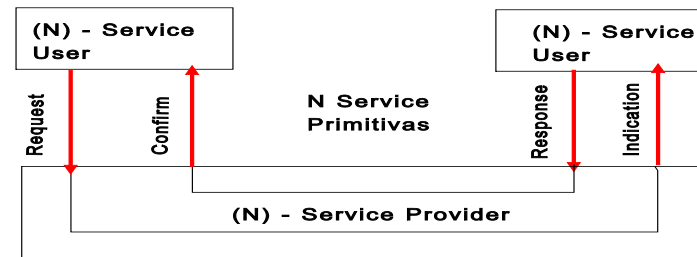
- 1) Conexión data-link: uno o mas data link connection entre dos entidades de red es provista por la capa.
- 2) Data link service data link.
- 3) Data link connection (**DLC**) end point identifiers.
- 4) Sequencing.
- 5) Error notification : a la entidades de red cuando un error es irrecuperable .
- 6) Flow Control.
- 7) Quality of service parameters: opcionalmente selectable . La capa de enlace establece y mantiene una selected quality of service durante la conexión de enlace de datos.

● Primitivas de servicios : Representación conceptual

¿Qué es una primitiva? acciones básicas que tiene un protocolo para establecer comunicación.

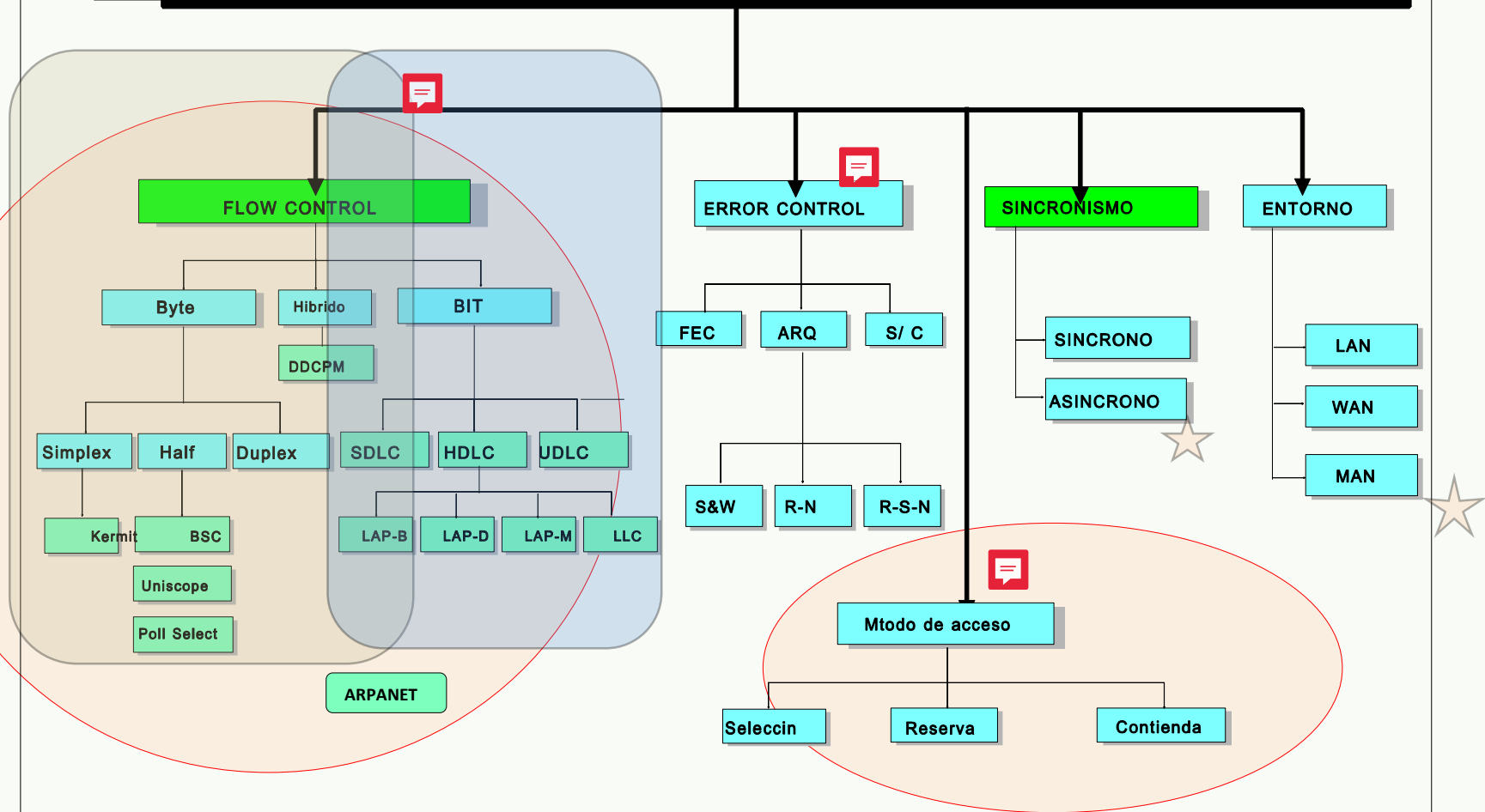
Tenemos 3 fases de esa primitiva:

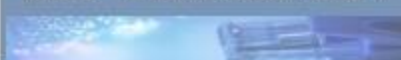
1. Establecer
2. Mantener
3. Liberar la comunicación



| Fase                     | Servicio                      | Primitiva   | Parametro  |
|--------------------------|-------------------------------|---|--|
| <b>DLC establishment</b> | DLC Establishment             | DL connect request<br>DL connect indication<br>DL response<br>DL connect confirm  | Called address calling address , QoS<br>Called address, calling address, QoS<br>Responding Address, QoS<br>Responding Address, QoS |
| <b>Data Transfer</b>     | Normal Data transfer<br>Reset | DL data request<br>DL data indication<br>DL reset request<br>DL reset indication<br>DL reset response<br>DL reset confirm | DLS user data<br>DLS user data<br>Razon<br>Razon , orig.   |
| <b>DLC release</b>       | DLC release                   | DL Disconnet request<br>DL disconnet indication   | Razon<br>Razon   |

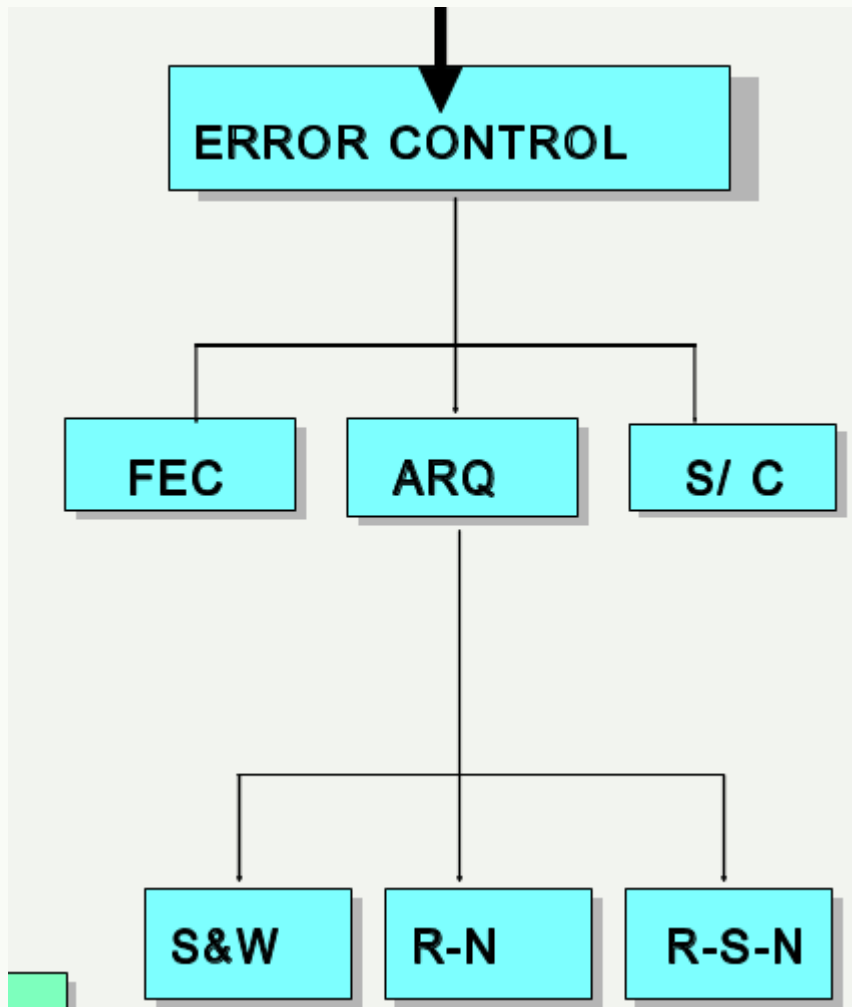
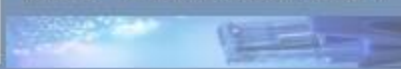
## CLASIFICACION DE LOS PROTOCOLOS DE LA CAPA DE ENLACE





***Que debe contener el mensaje de la capa enlace , para cumplimentar sus funciones:***

- **Delimitación** : quien y como se lo realiza ?    Es lo de bit o byte
- **Sincronismo** : en base a que realiza el sincronismo
- **Direccionamiento** : como direccionar y en base a que...    Tenemos direccionamiento a nivel 2, 3 y 4; nosotros vamos a ver el 2 (a nivel de capa de enlace) donde tenemos 2 tipos de redes: LAN y WAN
- **Control de flujo** : como realizarlo en base a que ?
- **Control de errores** : que sistema de detección /corrección adoptar



**FEC** : *Foward Error Correction*

**ARQ** : *Automatic Repeat Request* :

**Códigos**

- *Stop & Wait*
- *Regreso a N*
- *Regreso a N selectivo*

¿Qué pasa cuando detecta error? Tenemos 3 tipos de protocolos que va a fijar si el canal es half-duplex o full-duplex:

- Stop and wait: chequea si esta bien y manda un reconocimeinto

-R-N y R-S-N lo veo del práctico xd

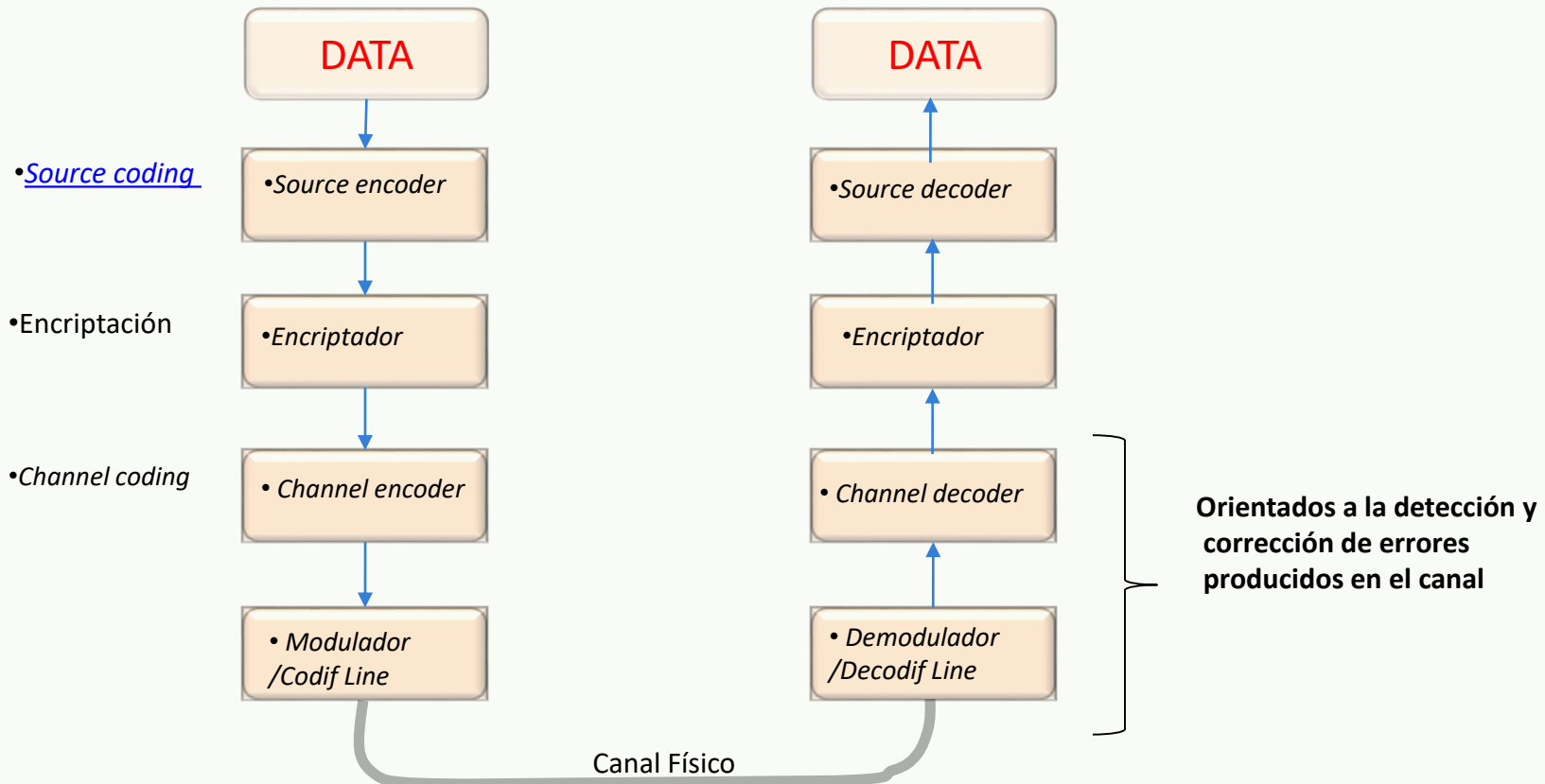
Entonces si tengo SyW, por más que el medio sea full-d el protocolo es half-d.

Los otros 2 si pueden ser full-d

### CODIGOS

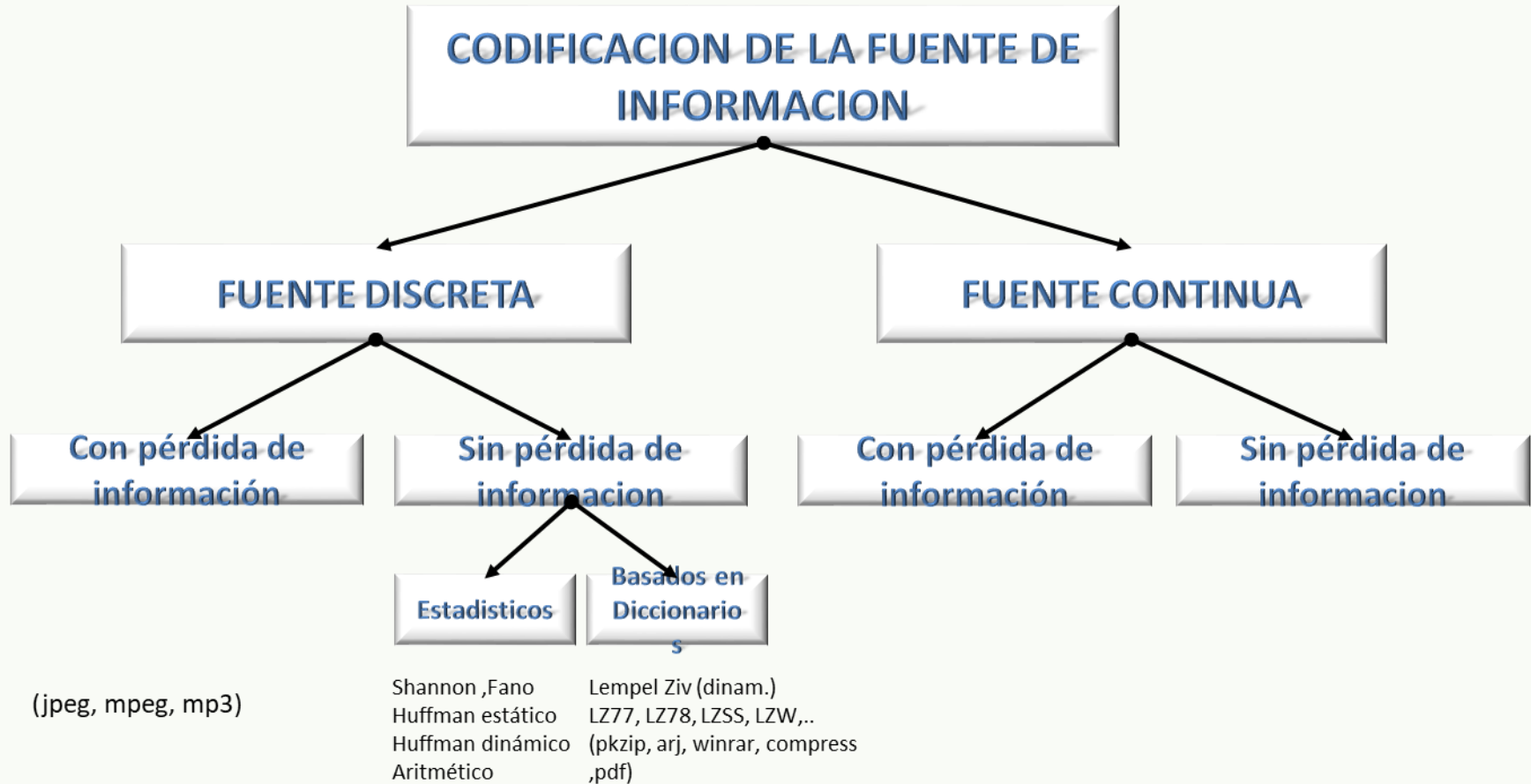
“Sistema de signos y de reglas que permite formular y comprender un mensaje”

-RAE



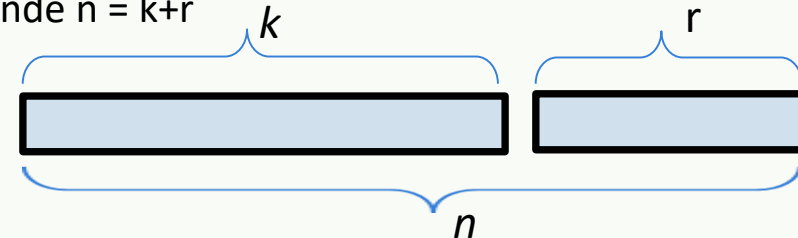
¿Cómo detecto errores? Tengo 2 tipos de códigos: los que miran hacia la fuente de datos (para encriptar, comprimir, específico del dato y no del canal) y los que miran hacia la línea (para mejorar la condición de transmisión dentro del canal)

No le interesa mucho



- **Códigos lineales :** Cuando la suma de dos palabras del código , es otra palabra del código.
- **Códigos de bloque :** k bits información , r bits redundantes

código de bloque  $(n,k)$  ; donde  $n = k+r$



✓ Velocidad del código :  $k/n$

✓ **Ganancia de un código :** La cantidad adicional de S/R, o  $E_b/N_0$  que se requiere para dar con la misma cantidad de BER del mensaje sin codificar.

✓ **Distancia de Hamming :**

- **Códigos Cíclicos:** Un código lineal de bloque  $(n,k)$  donde un corrimiento de una palabra resulta otra palabra del código .-

Un código lineal de bloque C, es cíclico si:

$$Y = y_{n-1} y_{n-2} \dots y_3 y_2 y_1 y_0 \in C$$

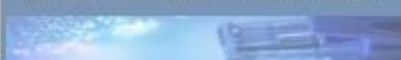
$$Y' = y_{n-2} \dots y_3 y_2 y_1 y_0 y_{n-1} \in C$$

Ej: Si  $\ni$  **01111110** como palabra del código C,  
es cíclico si **00111111** es otra palabra del código

- **Código polinómico**

Es un código lineal en el que cada una de las palabras del código es múltiplo de un determinado polinómico generador. Así un código cíclico es  $(n,k)$  es un código polinómico cuyo polinomio generador es divisor de  $x^n + 1$ .





## ■ **Códigos polinómicos cíclicos :**

Propiedades :

a) Toda palabra de un código cíclico  $(n, k)$  es un múltiplo del polinomio  $g(x)$  de grado  $n-k$  asociado al última línea de  $G$  y llamado polinomio generador .

b) Cualquier múltiplo del polinomio generador es una palabra del código polinomial

Si una secuencia de datos es  $d_1 \dots d_k$  esta se puede escribir como :

$$D(x) = d_1 x^{k-1} + \dots + d_{k-1} x + d_k$$

## ■ **Código convolutivo**

Si  $c(j)$  depende de  $i(j), \dots, i(j-m)$  se dice que el **código** es **convolutivo o recurrente** o secuencial.

## **Hamming estableció :**

### **Para detectar $S$ errores :**

La distancia mínima para detectar  **$S$  errores** en un código de bloque es

$$D = S + 1$$

### **Para corregir errores :**

La distancia mínima para corregir  **$T$  errores** en un código de bloque es

$$D = 2T + 1$$

## Métodos usados para detectar errores en una transmisión

- ☐ Por **paridad** : de una dimensión ,dos o mas ...
- ☐ Por **Checksum**: ya visto en otra materia [Uy no dijo la eficacia de este :\(](#)
- ☐ Por **CRC** (*cyclic redundancy check*) Mas utilizado en la actualidad

HPV

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| V | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| P | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| V | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| P | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| V | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| P | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| V | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| P | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |

Características : De que se requiere proteger ?  
Efectividad de un código ?

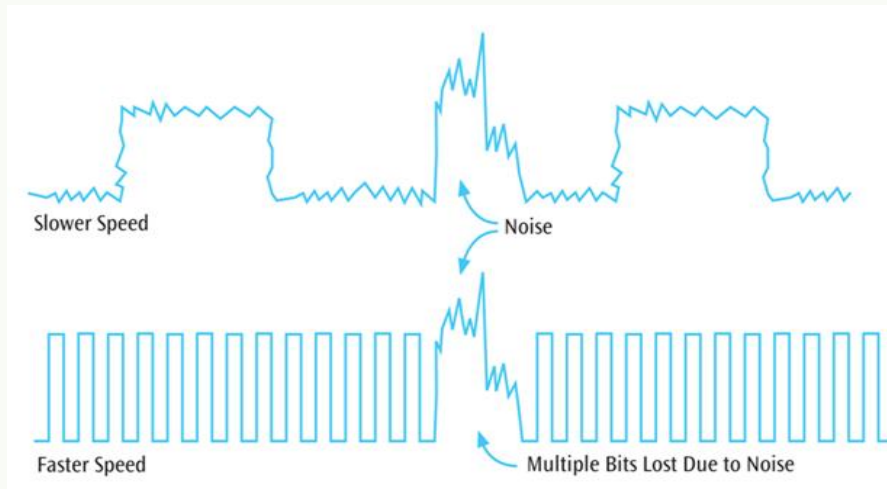
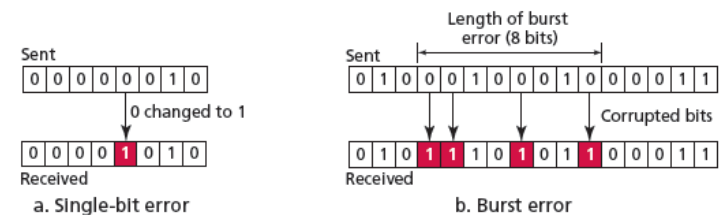
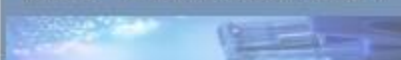


Figure H.1 Single-bit versus burst errors

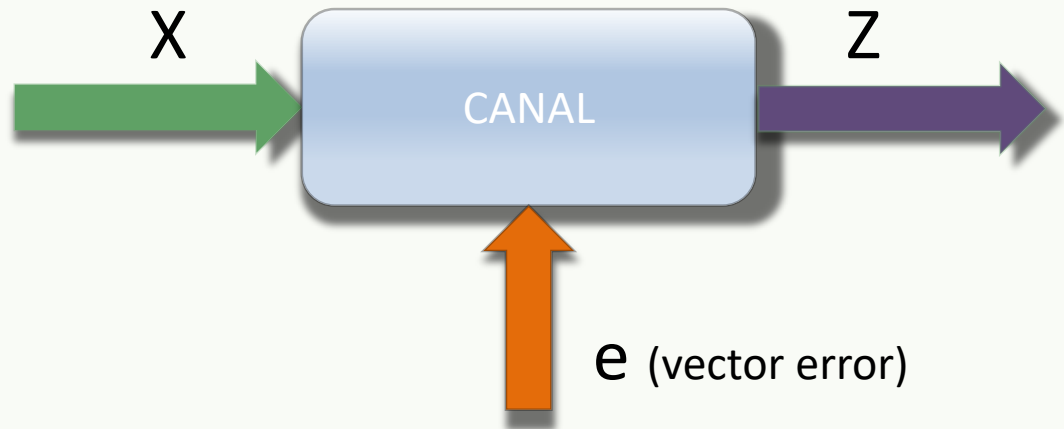




Síndrome :

$$Z = X + e$$

$$e = (e_1, \dots, e_n)$$



Esc

Síndrome de Z:

$$s = Z H^T = (s_1, \dots, s_n)$$

$$s = (X + e) H^T = X H^T + e H^T = e H^T$$

$$s = e H^T$$



## CRC, Cyclic Redundance Check

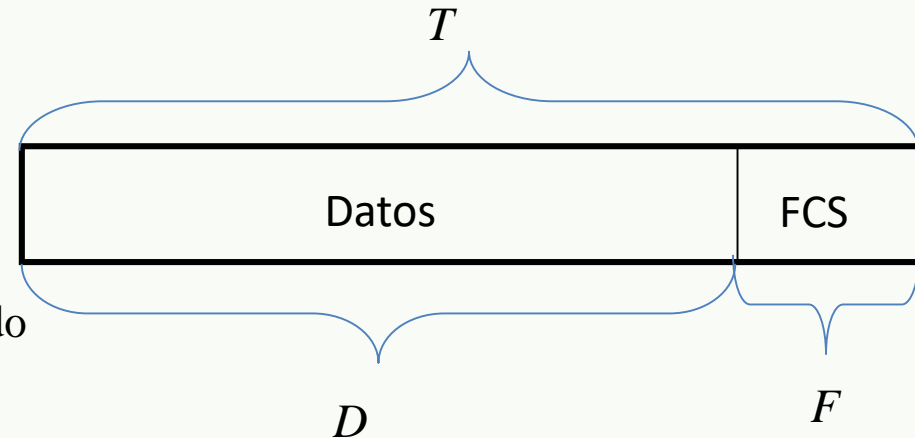
Definiendo :

$T = n$  bits Trama a ser transmitida

$D = k$  bit block de datos, los primeros  $k$  bits de  $T$

$F = (n - k)$  bit FCS, los ultimos  $(n - k)$  bits de  $T$

$P = \text{Polinomio}$   $(n - k + 1)$  bits, el divisor predeterminado



$$T = 2^{n-k} D + F$$

Suponga que :  $\frac{2^{n-k} D}{P} = Q + \frac{R}{P}$  Donde  $Q$  es el cociente y  $R$  es el residuo.

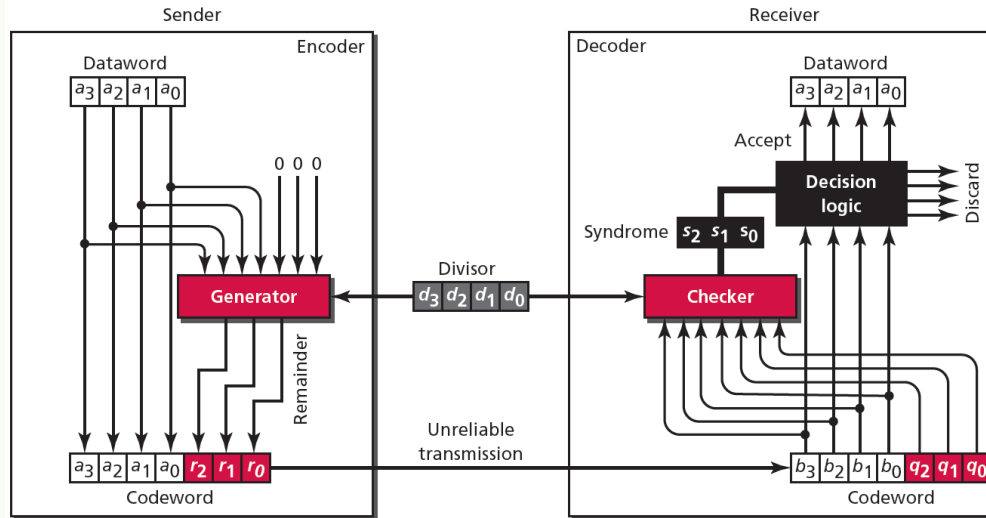
Usando el residuo como FCS, entonces :

$$T = 2^{n-k} D + R$$

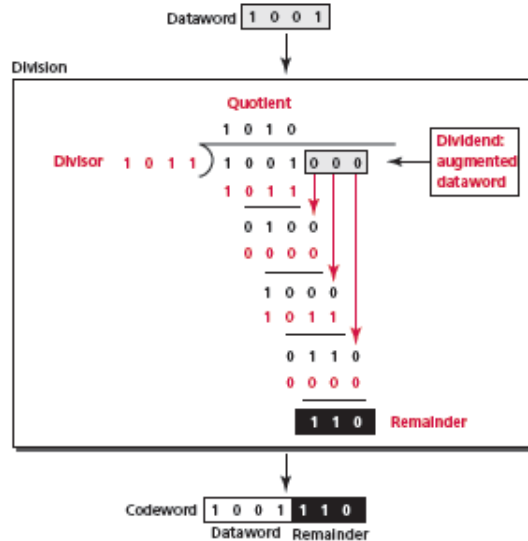
Para chequear que  $T/P$  no tiene residuo :

$$\frac{T}{P} = \frac{2^{n-k} D + R}{P} = \frac{2^{n-k} D}{P} + \frac{R}{P} = Q + \frac{R}{P} + \frac{R}{P} = Q + \frac{R + R}{P} = Q$$

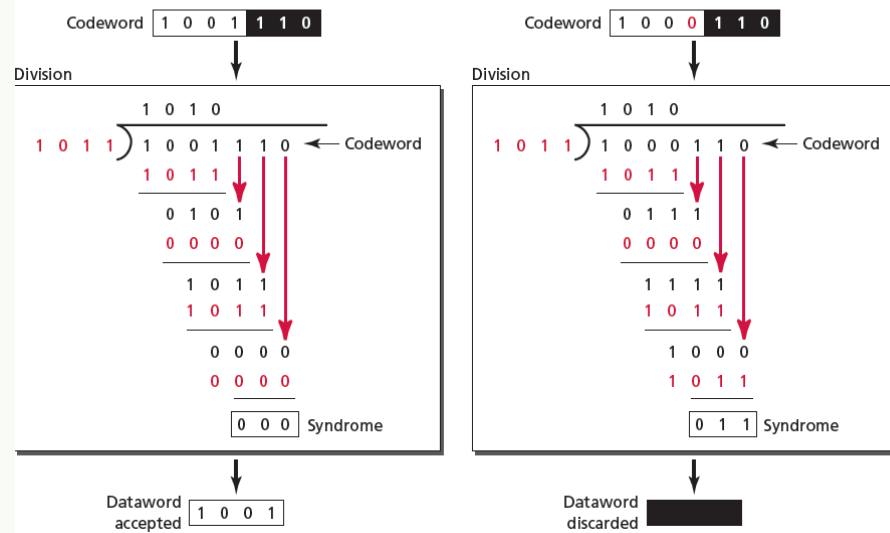
## H.7 CRC encoder and decoder

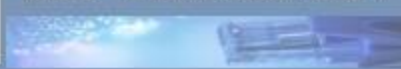


### Division in a CRC encoder



### H.9 Division in the CRC decoder for two cases





### Los CRC son capaz de detectar en los siguientes patrones :

- Cualquier ráfaga de errores de longitud  $n-k$
- Una fracción de ráfaga de longitud  $n-k+1$  , la fracción =  $1-2^{-(n-k-1)}$  .
- Toda combinación de  $d_{\min} - 1$  errores.

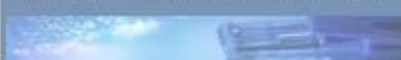
En el caso de 16 CRC detecta :

- Cualquier simple o doble error .
- Cualquier error de un número impar de bit.
- Cualquier burst de longitud menor o igual a 16 bits.
- Efectividad de 99,997% de 17 bit burst error

### Polinomios generadores mas conocidos :

- $\text{CRC-16} = x^{16} + x^{15} + x^2 + 1$  (HDLC)
- $\text{CRC-CCITT} = x^{16} + x^{12} + x^5 + 1$
- $\text{CRC-32} = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$   
(Ethernet )

Cuando son convenientes aplicarlos ?



Necesita mayor información para corregir. En transmisión se utiliza tanto el ARQ como el FEC. Se sigue utilizando FEC porque tiene una mejora relacionada con la ganancia del código que es de lo que hablamos en la fibra óptica

Los FEC pueden ser clasificados dentro de dos tipos de código : códigos de bloques o convolucionales .

Los códigos de bloques::

1) **Hamming** (*corrige 1 error*)

2) **Bose - Chaudhuri - Hocquenqhem (BCH)** estos son los códigos de mejor construcción para errores de ráfaga

2) **Reed - Salomom** . Son mas complejos .es una subclase del BCH

3) **Golay código** . Es un código (23,12).

Reed-Solomon codes are the most used codes today, due to their use in compact disc digital audio systems. They are also well known for their role in providing pictures of Saturn and Neptune during space missions such as the *Voyager*. In fact, RS codes are incorporated in the NASA Standard. These and several other applications of RS codes are described in [9].

Introduction to Algebraic Coding Theory

- Spence Adams

An RS code  $C_{RS}(n, k)$  able to correct any error pattern of **size**  $t$  or less is defined over the Galois field  $GF(q)$ , and it has as parameters:

- **Code length**  $n = q - 1$
- Number of parity check elements  $n - k = 2t$
- Minimum distance  $d_{min} = 2t + 1$
- Error-correction capability  $t$  element errors per code vector

## Detection *versus* correction

Detección :



Corrección :



## Forward error correction *versus* retransmission

ARQ

- En enlaces con elevado BER/ retardo
- Dado un determinado BER de enlace
  - Reducir potencia de transmisión
    - Tamaño de antena
  - Reducir el BW requerido
  - Etc....

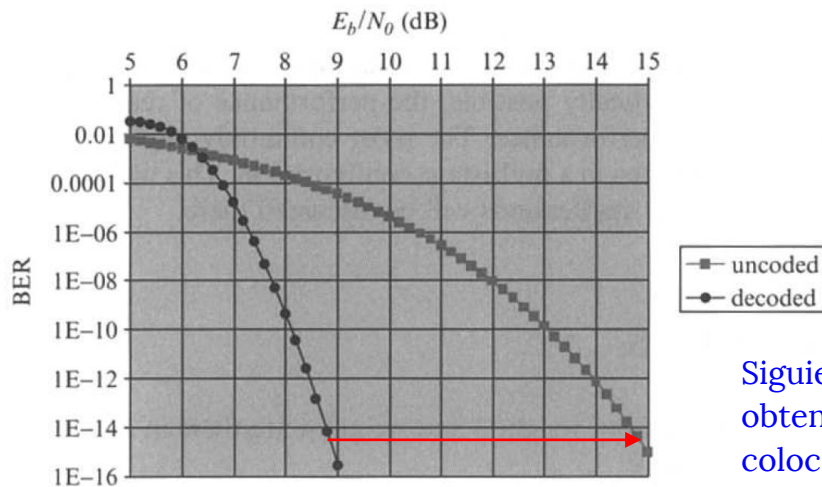


Figure 8.4 Performance of (255, 239) RS code

**Pregunta :**

Si se tiene una ganancia de 3db por utilizar FEC, en cuanto puedo reducir el ancho de banda ?

Siguiendo con la explicación de ganancia de un código, acá dice que obtengo un BER de  $10^{-14}$  con una relación señal ruido de 14. Pero si le coloco un FEC necesito una relación S/R mucho menor (9); entonces puedo manejarme con un canal ruidoso y tener la misma cantidad de errores pero con un código de corrección de errores, es lo que usa la fo





## Automatic repeat request -Métodos de ARQ

¿Como se recupera errores en la transmisión ?

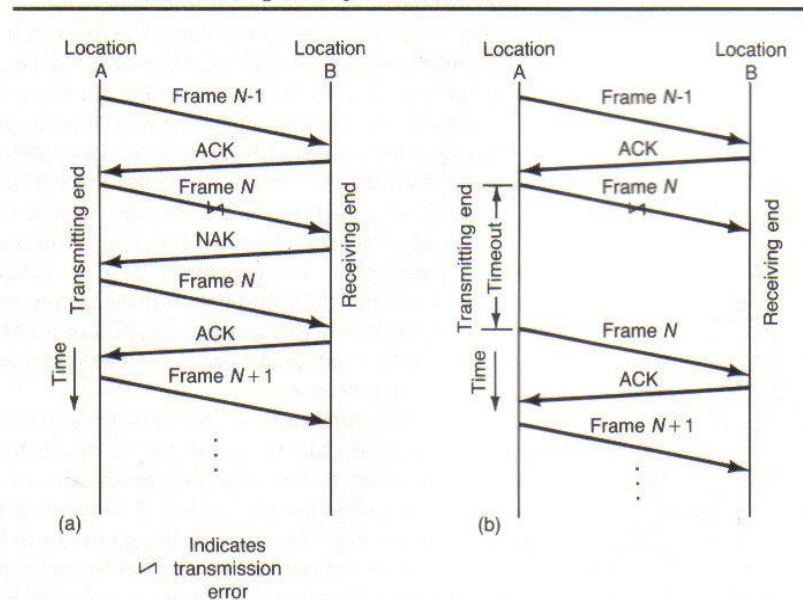
Dentro del método de ARQ existen varia posibilidades de tratamiento del ARQ, como :

**a) Protocolo de parar y esperar** : básicamente consiste en transmitir una trama y esperar su confirmación a travez de un ACK, de lo contrario se retransmite la trama .

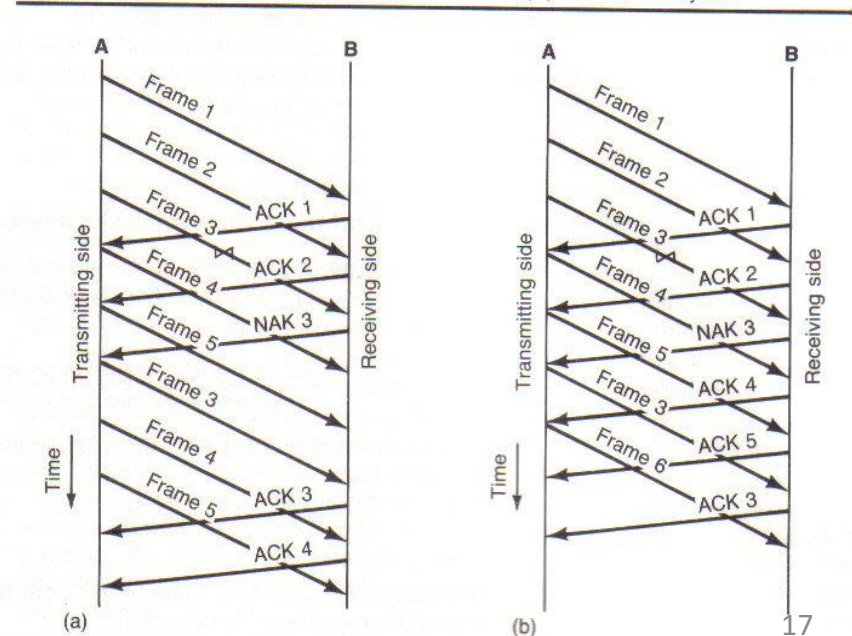
**b) Protocolo Regresa a N** , o *transmisión continua*,: El método consiste en enviar tramas sin esperar un ack , cuando se recibe un NACK o expira un determinado tiempo , se transmite la trama por el cual se recibió un nack y todas las que siguen.

**c) Protocolo de retransmisión selectiva** : consiste en retransmitir solamente la trama en cual se recibió un NACK.

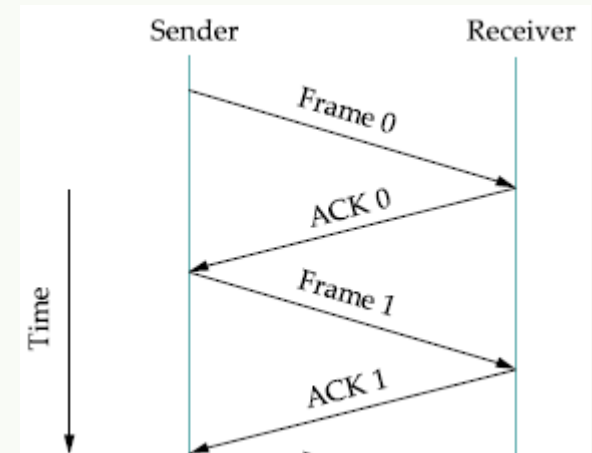
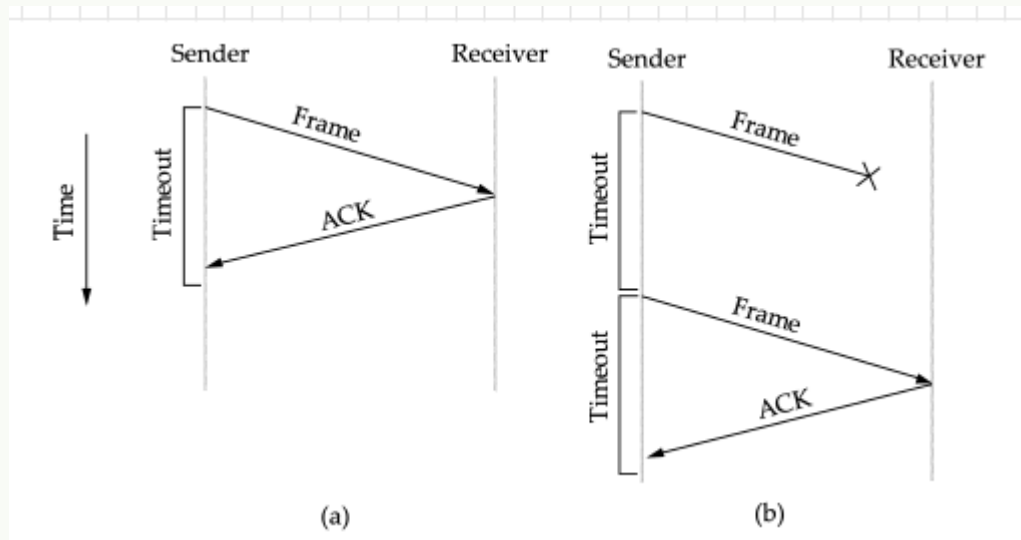
**Figure 6.4** Stop-and-Wait ARQ. (a) With both positive and negative acknowledgments. (b) With positive acknowledgments plus timeouts.



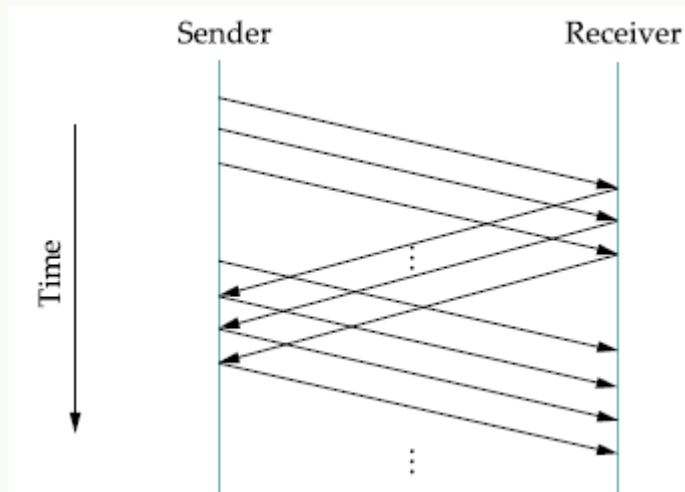
**Figure 6.8** Continuous ARQ. (a) Go-back-N. (b) Selective reject.

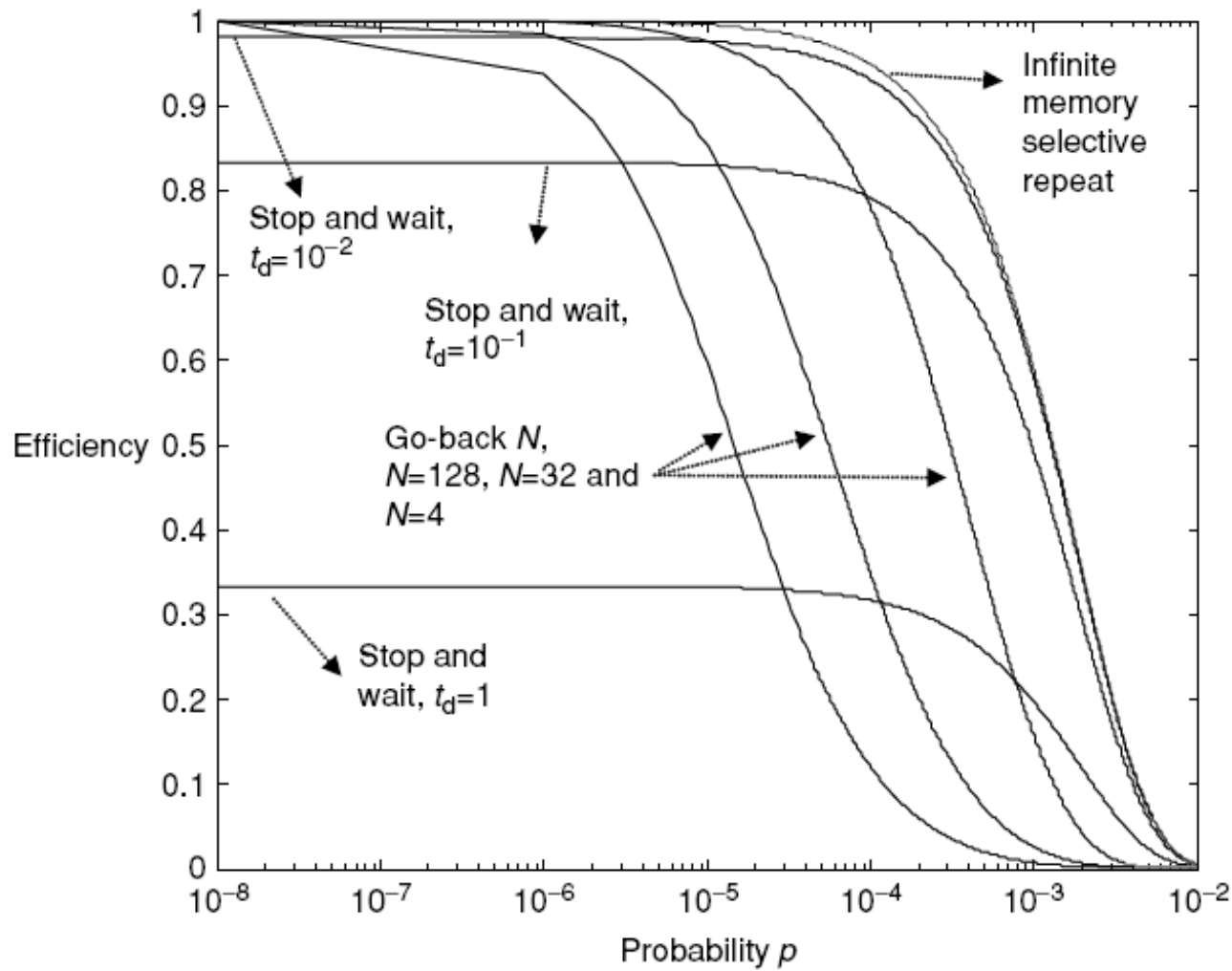


## Stop and wait



## Regreso a N





| Line                           | Error Rate  |
|--------------------------------|---|
| Floppy disk                    | $10^{-9}$<br>at 5 MB/s, 3 erroneous bits per minute           |
| Optical CD-ROM                 | $10^{-5}$<br>7 kB erroneous on a 700 MB CD-ROM                |
| DAT audio                      | $10^{-5}$<br>at 48 kHz, two errors per second                 |
| Blu-ray disk                   | $< 2 \cdot 10^{-4}$<br>about 1.6 MB erroneous on a 32 GB disk |
| Hard Disk Drives (HDD)         | $> 10^{-4}$<br>$\sim 10^6$ errors in 10 GB                    |
| Flash technologies             | $\approx 10^{-5}$   |
| Solid-State Drives (SSD)       | $< 10^{-5}$   |
| Semiconductor memories         | $< 10^{-9}$   |
| Phone line                     | Between $10^{-4}$ and $10^{-7}$                               |
| Infrared remote control        | $10^{-12}$  |
| Communication on optical fiber | $10^{-9}$   |
| Spacecraft                     | $10^{-6}$ (Voyager), $10^{-11}$ (TDMA)                        |
| ADSL                           | $10^{-3}$ to $10^{-9}$  |
| Computer network               | $10^{-12}$  |

Probabilidad de error en una trama de longitud  $F(P_1)$  :  $P_1 = (1 - P_b)^F$

$$P_{nc} = 1 - P_1$$

$P_b$  Probabilidad de error 1 bit

$P_{nc}$  Probabilidad de error en una trama no detectado

## Que factores Intervienen en el rendimiento de un protocolo , si se entiende por rendimiento

Tenemos que entender por qué existen distintos protocolos y no hay uno supremo y listo, esto va a depender de donde se lo va a utilizar y como.

¿A que le llamamos rendimiento de un protocolo y que factores intervienen? A la cantidad de bits de info que queremos transmitir vs los transmitidos. Siempre bits de info < bits transmitidos por que hay que transmitir el comienzo, el final, el control, la sincro, elementos que no están dentro de la info propiamente dicha. Esta es una forma de verlo.

$$R = \frac{\text{Bits de informacion}}{\text{Bits transmitidos}}$$

Otra forma de verlo es el tiempo que tengo para transmitir información vs el tiempo total en transmitir dicha info

$$R = \frac{\text{tiempo de informacion}}{\text{tiempo total en transmitir dicha info}}$$

¿Cuales son los factores que van a influir en el R?

- ✓ Velocidad de transmisión
- ✓ Velocidad de propagación
- ✓ Distancia entre T/R
- ✓ Longitud de bits a transmitir
- ✓ Tasa de errores producidos en el canal

Como inciden ?

¿Entonces me conviene hacer tramas largas o cortas? Si yo hago una trama muy grande y el canal es ruidoso siempre va a detectar errores y me va a decir retransmitime pq tenes errores; si hago una muy chica le tengo que poner encabezados y demás. Hay un punto medio que es lo óptimo dentro de mi sistema particular.



## Protocolo de parar y esperar

Para realizar un análisis del rendimiento de este protocolo considérese la fig cap-en01. Para ello se considera siguiente :

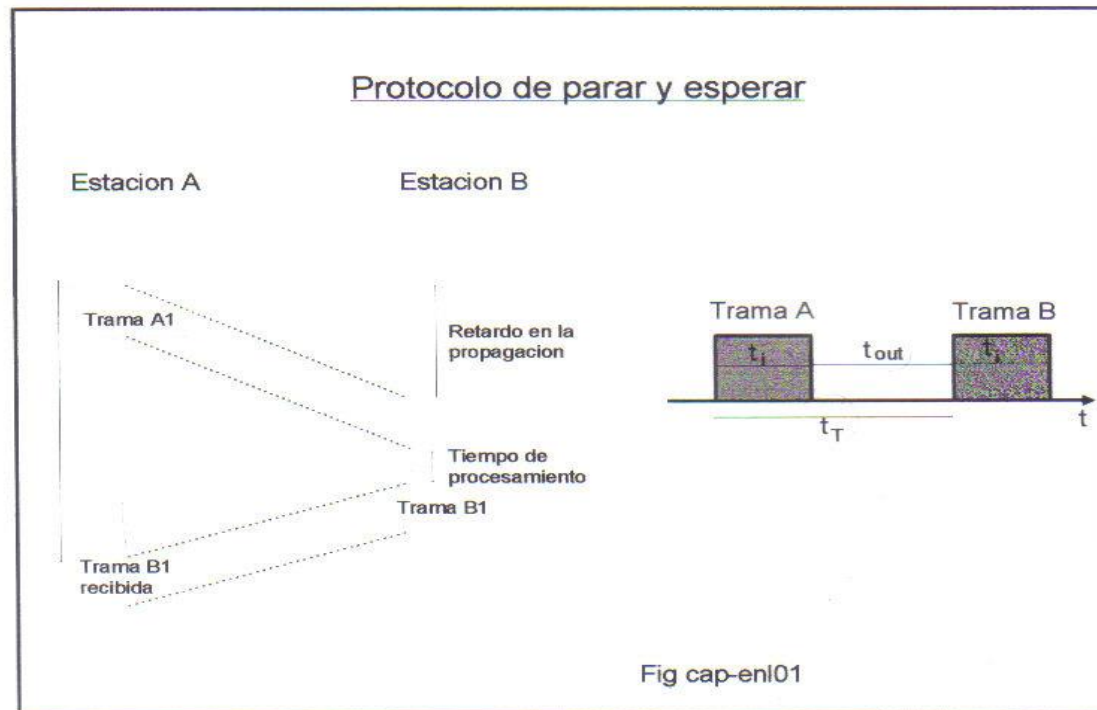
El tiempo  $t_T = t_i + t_{out}$ .

El transmisor opera de forma que siempre tiene información para transmitir .

El tiempo de propagación es  $t_p$ . Sea  $p$  es la probabilidad que una trama se reciba con error.

$$t_v = t_T + (1-p) \sum_{i=1}^{\infty} i p^i t_T \Rightarrow \text{considerando que la serie } x + 2x^2 + 3x^3 + \dots = \frac{x}{(1-x)^2}$$

$$t_v = \frac{t_T}{(1-p)}$$



El rendimiento máximo entregado en paquetes por segundo, es recíproco al  $t_v$

$$\lambda_{\max} = 1/t_v = (1-p)/t_T \quad t_T = t_i + 2t_p$$

para tener un rendimiento normalizado, tenemos:

$$\rho = \lambda_{\max} \cdot t_i = (1-p) / (t_i + 2t_p) / t_i = (1-p)/(1 + 2.a)$$

donde  $a = t_p / t_i = d_{\text{ist}} \cdot v_{\text{enlace}} / l_{\text{ong.trama}} \cdot V_{\text{prop}}$

$$T_{\text{out}} = t_{\text{inf}} + 2\tau = t_{\text{inf}} + 2 \cdot d_{\text{istanica}} / v_{\text{pro}}$$

$$T_{\text{inf}} = l_{\text{ong trama}} / v_{\text{enlace}}$$

$\rho$  será mayor si  $a < 1$  y  $p$  cercano a 0

## PROTOCOLO REGRESA A N

En este caso el  $t_v$  es, teniendo en cuenta la supocisiones de transmisión saturada y el  $t_{\text{out}}$ :

$$t_v = t_i + (1-p) \sum_{i=1}^{\infty} i p^i t_T = t_i \cdot \frac{[1 + (a-1)p]}{(1-p)}$$

El rendimiento máximo esta dado por:

$$\lambda_{\max} = \frac{1}{t_v} = \frac{(1-p)}{t_i [1 + (a-1)p]}$$

Y el rendimiento normalizado

$$\rho = \lambda_{\max} \cdot t_i = \frac{(1-p)}{(1 + (a-1)p)}$$

Coupling ratio  $\alpha = \tau / T$

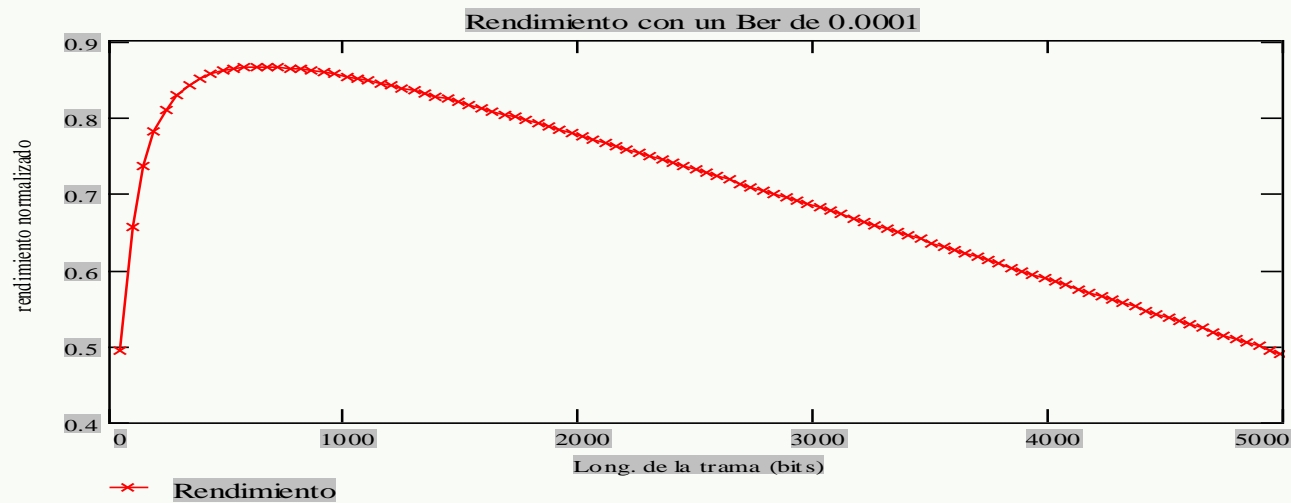
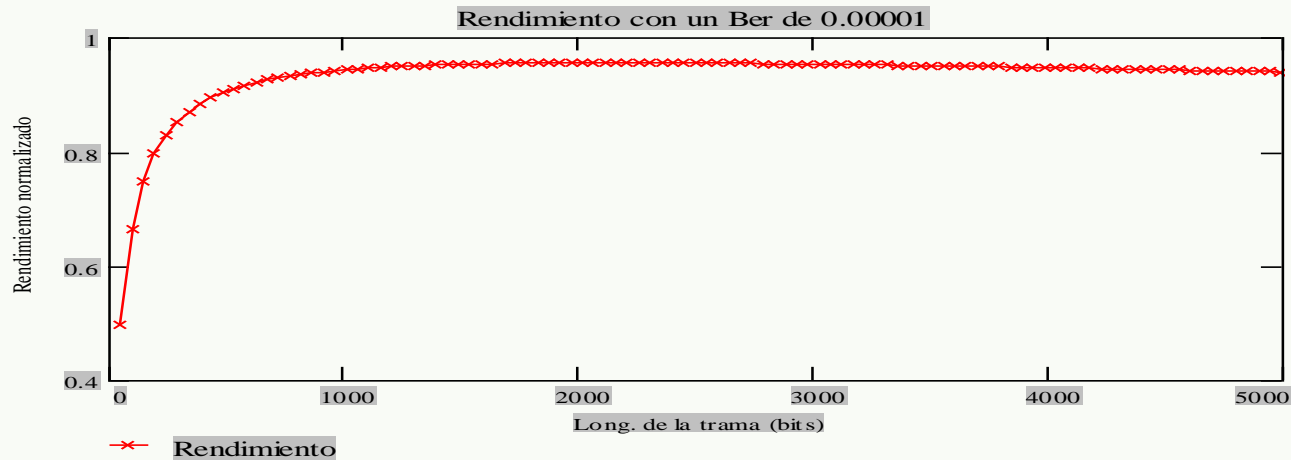
Where

- $\tau$  Propagation delay for packet
- $T$  Average packet transmission time
- $\alpha \ll 1$  indicates a LAN
- $\alpha = 1$  indicates a MAN
- $\alpha \gg 1$  indicates a WAN



## Rendimiento según tasa de errores en el canal de comunicación

$x := 48,96..5000$





## PROTOCOLOS ORIENTADOS AL BYTE.....

ASCII tiene caracteres especiales referidos al control con SYN, IBM fue el primero en hacerlo, son los bytes que usan los protocolos

### ASCII CONTROL CODE CHART

| BITS<br>B7 B6 B5 B4 B3 B2 B1 | 0 0 0 0 | 0 0 0 1 | 0 1 0 0            | 0 1 0 1 | 1 0 0 0    | 1 0 0 1 | 1 1 0 0    | 1 1 0 1         |
|------------------------------|---------|---------|--------------------|---------|------------|---------|------------|-----------------|
|                              | CONTROL |         | NUMBERS<br>SYMBOLS |         | UPPER CASE |         | LOWER CASE |                 |
| 0 0 0 0                      | NUL     | DLE     | SP                 | 0       | @          | P       | '          | p               |
| 0 0 0 1                      | SOH     | DC1     | !                  | 1       | A          | Q       | a          | q               |
| 0 0 1 0                      | STX     | DC2     | "                  | 2       | B          | R       | b          | r               |
| 0 0 1 1                      | ETX     | DC3     | #                  | 3       | C          | S       | c          | s               |
| 0 1 0 0                      | EOT     | DC4     | \$                 | 4       | D          | T       | d          | t               |
| 0 1 0 1                      | ENQ     | NAK     | %                  | 5       | E          | U       | e          | u               |
| 0 1 1 0                      | ACK     | SYN     | &                  | 6       | F          | V       | f          | v               |
| 0 1 1 1                      | BEL     | ETB     | '                  | 7       | G          | W       | g          | w               |
| 1 0 0 0                      | BS      | CAN     | (                  | 8       | H          | X       | h          | x               |
| 1 0 0 1                      | HT      | EM      | )                  | 9       | I          | Y       | i          | y               |
| 1 0 1 0                      | LF      | SUB     | *                  | :       | J          | Z       | j          | z               |
| 1 0 1 1                      | VT      | ESC     | +                  | ;       | K          | [       | k          | {               |
| 1 1 0 0                      | FF      | FS      | ,                  | <       | L          | \       | l          |                 |
| 1 1 0 1                      | CR      | GS      | -                  | =       | M          | ]       | m          | }               |
| 1 1 1 0                      | SO      | RS      | .                  | >       | N          | ^       | n          | ~               |
| 1 1 1 1                      | SI      | US      | /                  | ?       | O          | _       | o          | RUBOUT<br>(DEL) |

#### LEGEND

|       |    |   |                         |
|-------|----|---|-------------------------|
| Octal | 10 | ← | LSI CURSOR CONTROL CODE |
| hex   | BS |   | ASCII character         |
|       | 8  |   | decimal                 |



### 4.4.1.- PROTOCOLO BISINCRONO (BSC)

Protocolo propietario de IBM (1960) , con las siguientes características:

- Protocolo orientado al carácter.
- Tipo de acceso : por sondeo , sistema polling.
- Tipo de explotación : half duplex.
- Detección de errores : según versión del código :
  - Vertical redundancy checking (VRC)
  - Longitudinal redundancy checking (LRC)
  - Cyclic redundancy checking (CRC)

#### Formatos típicos del protocolo BSC

|     |     |     |         |     |      |         |     |
|-----|-----|-----|---------|-----|------|---------|-----|
| SYN | SYN | SOH | HEADING | STX | DATA | ETX/ETB | BCC |
|-----|-----|-----|---------|-----|------|---------|-----|

|       |       |       |       |                    |   |       |     |
|-------|-------|-------|-------|--------------------|---|-------|-----|
| SYN - | SYN - | DLE - | STX - | TEXTO TRANSPARENTE | - | DLE - | ETX |
|-------|-------|-------|-------|--------------------|---|-------|-----|

|       |       |       |         |   |     |   |     |
|-------|-------|-------|---------|---|-----|---|-----|
| PAD - | SYN - | SYN - | POLLING | - | ENQ | - | PAD |
|-------|-------|-------|---------|---|-----|---|-----|

## Formato típico

|    |    |     |        |     |            |     |     |     |
|----|----|-----|--------|-----|------------|-----|-----|-----|
| Sy | Sy | SOH | Header | STX | DATA ..... | ETX | EOT | BCC |
|----|----|-----|--------|-----|------------|-----|-----|-----|

## PROTOCOLO UNISCOPE

Protocolo propietario de UNISYS, con las siguientes características:

- Protocolo orientado al carácter.
- Tipo de acceso : por selección , sistema de polling.
- Tipo de explotación : Half duplex ( protocolo de parar y esperar).
- Corrección de errores : VRC y LRC.

| Función                 | cht a la terminal     | cht desde la terminal |
|-------------------------|-----------------------|-----------------------|
| Start of heading        | SOH                   | SOH                   |
| Start of text           | STX                   | STX                   |
| End of text             | ETX                   | ETX                   |
| Status Poll             | ENQ                   | ----                  |
| Replay Request          | ----                  | DEL ENQ               |
| Acknowledge             | DEL 1                 | DEL 1                 |
| Retransmit Request      | DEL NACK              |                       |
| Busy                    | ----                  | DEL?                  |
| Sincronismo             | SY                    | SY                    |
| Device Ready            |                       | DEL >                 |
| Device error            |                       | DEL <                 |
| Controles de secuencia: | Caracteres combinados |                       |
| Erase to end of display |                       | Esc a                 |
| Erase to end of line    |                       | Esc b                 |
| Cursor to home          | Esc e                 |                       |
| Insert line             | Esc d                 |                       |
| Print transparent       | Esc Dc2               |                       |
| Delete line             | Esc k                 |                       |

## Formato sin datos con reconocimiento

|     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|
| SOH | RID | SID | DID | DL1 | ETX | BCC |
|-----|-----|-----|-----|-----|-----|-----|

## Reply Request(Poll)

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| SOH | RID | SID | DID | DEL | ENQ | ETX | BCC |
|-----|-----|-----|-----|-----|-----|-----|-----|

## Busy

|     |     |     |     |      |     |     |     |
|-----|-----|-----|-----|------|-----|-----|-----|
| SOH | RID | SID | DID | DEL? | ENQ | ETX | BCC |
|-----|-----|-----|-----|------|-----|-----|-----|

## Text sin ACK

|     |     |     |     |     |       |     |     |
|-----|-----|-----|-----|-----|-------|-----|-----|
| SOH | RID | SID | DID | STX | TEXTO | ETX | BCC |
|-----|-----|-----|-----|-----|-------|-----|-----|

## Text con ACK

|     |     |     |     |      |     |       |     |     |
|-----|-----|-----|-----|------|-----|-------|-----|-----|
| SOH | RID | SID | DID | DEL1 | STX | TEXTO | ETX | BCC |
|-----|-----|-----|-----|------|-----|-------|-----|-----|

## Requesta al poll sin ACK

|     |     |     |     |      |     |     |
|-----|-----|-----|-----|------|-----|-----|
| SOH | RID | SID | DID | DEL0 | ETX | BCC |
|-----|-----|-----|-----|------|-----|-----|

## Repuesta al poll con ack

|     |     |     |     |      |      |     |     |
|-----|-----|-----|-----|------|------|-----|-----|
| SOH | RID | SID | DID | DEL1 | DEL0 | ETX | BCC |
|-----|-----|-----|-----|------|------|-----|-----|



Protocolo propietario de **D**igital **E**quipment **C**o, **D**igital **D**ata **C**ommunications **M**essage Protocol, básicamente orientado al byte , con las siguientes características :

- Orientado al byte .
- Tipo de encabezamiento de long. fija.
- Unidades de control menor que el byte.

Formatos del protocolo

Mensaje de control

- Resp : response number , número del mensaje del último mensaje recibido, modulo 256.
- Fill : no usado.
- FL : dos bits contiendo flag usados en líneas de control.
- Num : Número del mensaje.
- Count : 14 bits especificando el número de caracteres en campo de datos.

|     |     |                  |        |     |      |     |
|-----|-----|------------------|--------|-----|------|-----|
| SYN | SYN | Especialcaracter | Header | BCC | Data | BCC |
|-----|-----|------------------|--------|-----|------|-----|

|     |     |     |      |         |    |      |      |      |     |
|-----|-----|-----|------|---------|----|------|------|------|-----|
| SYN | SYN | ENQ | Type | Subtype | FL | Resp | Fill | Addr | CRC |
|-----|-----|-----|------|---------|----|------|------|------|-----|

|     |     |     |       |    |      |     |      |     |
|-----|-----|-----|-------|----|------|-----|------|-----|
| SYN | SYN | SOH | Count | FL | Resp | Num | Addr | CRC |
|-----|-----|-----|-------|----|------|-----|------|-----|



## Problemas con protocolos orientados al byte:

- Dependientes del código
- No encapsulables
- Propietarios

Uno de los primeros protocolos orientado al bit es este, de acá depende el HLC, el CLC, ethernet, X25, todos tienen formato similar:

- Tiene una sola bandera que le fija el inicio y la finalización de la trama (ver adjunto) (1 byte)
- Un campo de dirección (1 o 2 bytes)
- Un campo de control (1 o 2 bytes)
- Un campo de información (n bytes)
- FCS (2 bytes, el CRC)

Lo bueno es que el campo de control es específico, en este caso tiene 8 bits y está dividido en 3 tipos de tramas o controles:

- Información
- Supervisión
- No numerado (no seq): sirve para establecer y liberar la comunicación

Si el primer bit es 0 lo que viene es info, cuando el primero es 10 lo que viene es supervisión, si es 11 es no seq

Establece con no secuencial, mantener lo hace con supervisión e información

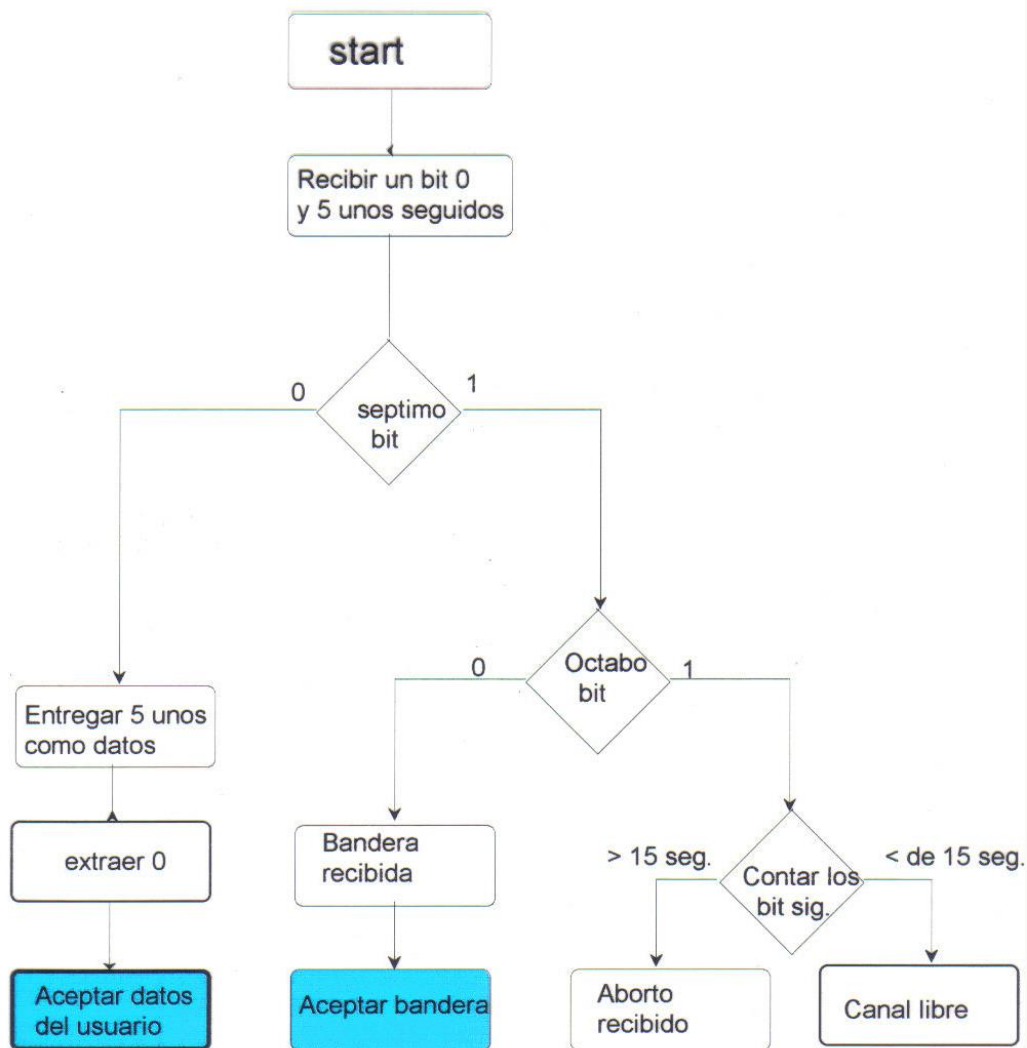
N(S): Número de secuencia enviada. N(R): Número de secuencia recibida.

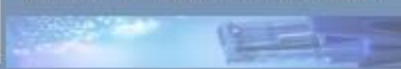
P/F es para decir si es una orden de comando o respuesta

M si lo que está haciendo es un ¿llamado? o no

CAPA ENLACE

## Condición de una línea HDLC





- Flag pattern: 01111110
- Sender **stuff**: add 0 after 11111
- Receiver **destuff**: delete 0 after 11111

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

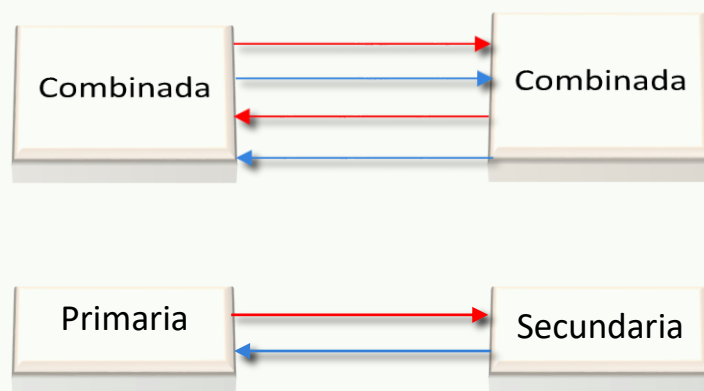
(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits



(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

## Configuración de estaciones en HDLC



Comando →

Repuesta →

En HLC podemos tener una estación combinada (puede enviar o recibir comandos), una estación primaria cuando solo emite comando y secundaria cuando solo emite respuesta

## Estados de una estación

- *Estado de desconexión lógica*: (no transmite ni recibe) Si la estación secundaria se encuentra en desconexión lógica, solo podrá emitir cuando reciba una autorización expresa.
- *Estado de inicialización*: depende del fabricante.
- *Estado de transferencia de información*: estaciones principal y secundaria en condiciones de transmitir y recibir información

## Configuración del canal

- *No equilibrada* (UN) Se llama no equilibrada porque existe una estación encargada de gobernar a cada una de las secundarias y establecer los comandos de los distintos modos que pueden trabajar: half / full duplex, p. a p., multipunto.
- *Simétrica*: (UA) utilizada originalmente, proporciona dos configuraciones: punto a punto independientes y no equilibradas, cada estación tiene su estado principal y su estado secundaria.
- *No simétrica*: (BA) Consta de dos estaciones combinadas por un solo enlace punto a punto, half o full duplex, cada una posee derecho y repuesta del canal.

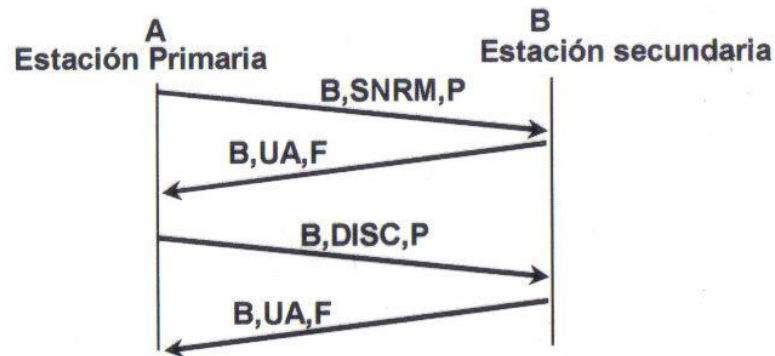
## MODOS DE TRANSFERENCIAS DE INFORMACIÓN

- *De repuesta normal* (NRM, *normal response mode*). Utilizada en ambiente de control centralizado. Por consiguiente la configuración típica es multipunto, donde la estación primaria (master, en la nomenclatura de IBM) obliga a la estación secundaria (slave) a esperar autorización explícita antes de transmitir.
- *Modo de repuesta asincrónica* (ARM, *asynchronous response mode*), similar al NRM excepto que una estación secundaria puede transmitir sin autorización previa de la estación principal. Opera en la modalidad de rechazo selectivo de tramas.
- *Modo asincrono equilibrado* (ABM, *asynchronous mode*) (LAP-B, *link access procedure-balanced*). Este modo solo opera en configuración punto a punto. Estaciones combinadas las cuales pueden iniciar sus transmisiones sin autorización de las otras estaciones combinadas. Además como característica diferente de los otros dos modos, en que opera en la modalidad de retransmisión de errores de regresar a N.

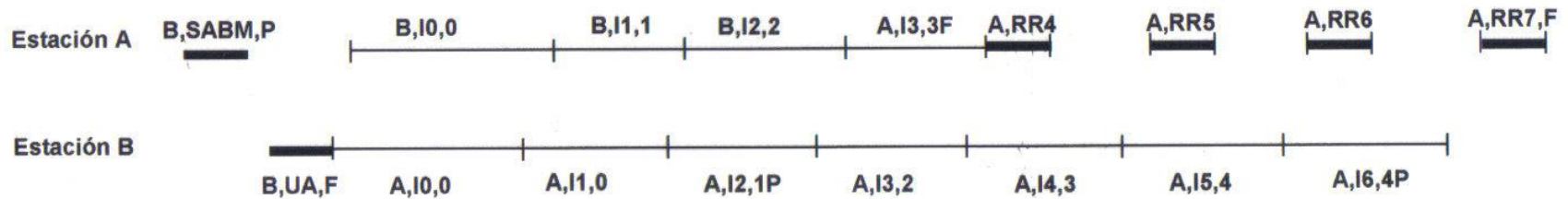
El uso del bit P/F es diferente según los modos. Un 1 en comando se toma como P, lo que la repuesta debe ser con F al final de la repuesta. En un modo de respuesta normal el bit P/F se utiliza para escrutinio. En modo ABM con procedimientos BAC (balanced asynchronous mode) 2,8 (uso de REJ y que la trama I sean solo comandos) trama I enviada con P = 1 requiere de una trama S como repuesta con F=1, ya que la trama I no puede ser repuesta. Este procedimiento se conoce como punto de control.



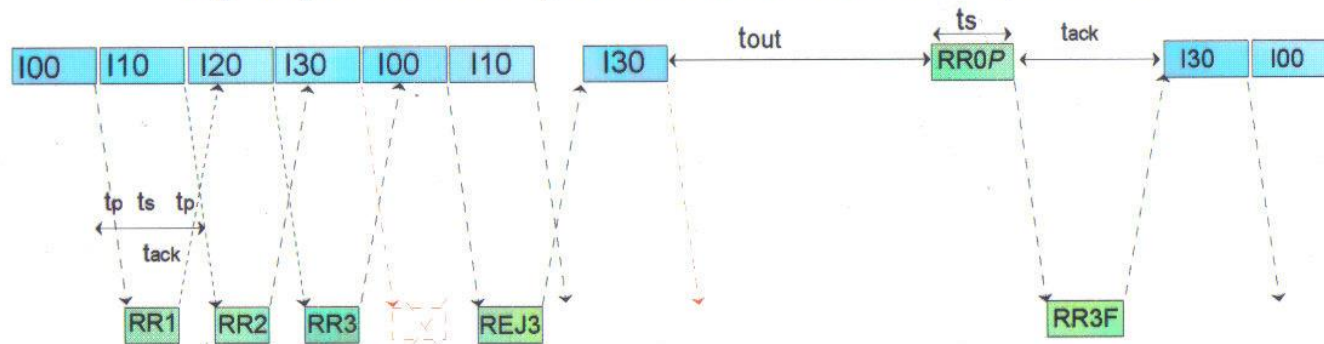
## Inicialización y desconexión en HDLC



## LAP-B - Transmisión continua de tramas



## Ejemplo de recuperación de errores en HDLC





## LINK ACCESS PROCEDURE -BALANCED ( LAP-B)

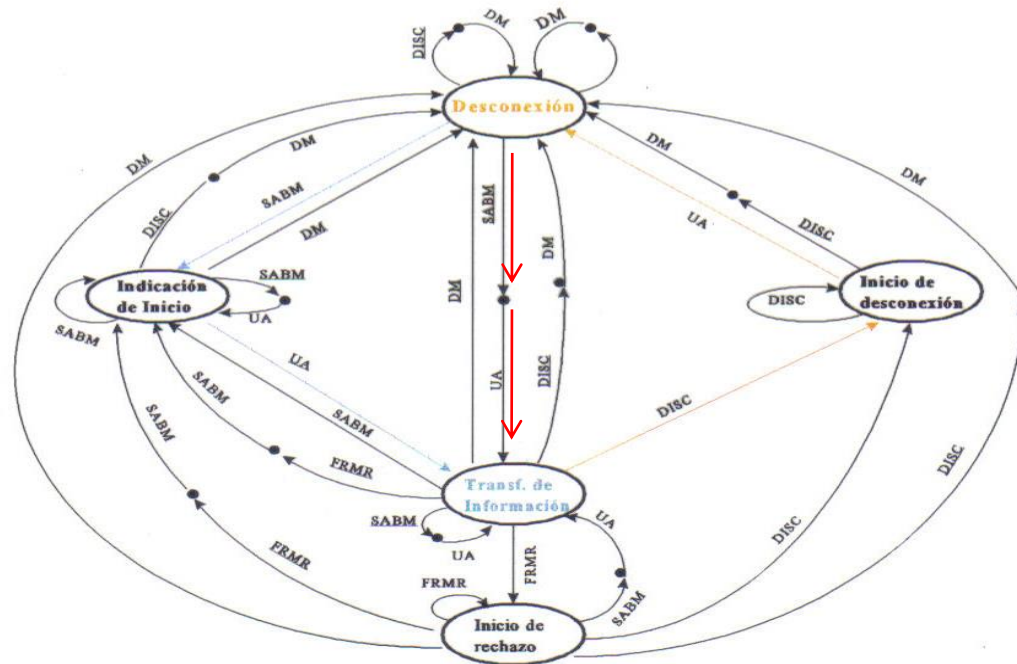
CAPA ENLACE

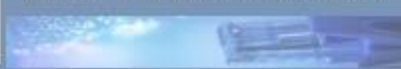
### Un subproducto del HDLC : LAP-B

EL campo de control en el LAP-B es:

| Formatos           | Comandos                         | Respuestas   | Bits |      |   |   |     |      |   |   |
|--------------------|----------------------------------|--|------|------|---|---|-----|------|---|---|
|                    |                                  |  | 1    | 2    | 3 | 4 | 5   | 6    | 7 | 8 |
| <b>Información</b> | <b>I</b>                         |  | 0    | N(S) |   |   | P   | N(R) |   |   |
| <b>Supervisión</b> | <b>RR</b> - Receptor listo       | <b>RR</b>  | 1    | 0    | 0 | 0 | P/F | N(R) |   |   |
|                    | <b>RNR</b> Recp. no listo        | <b>RNR</b>   | 1    | 0    | 1 | 0 | P/F | N(R) |   |   |
|                    | <b>REJ</b> Rechazo               | <b>REJ</b>   | 1    | 0    | 0 | 1 | P/F | N(R) |   |   |
| <b>Sin Numerar</b> | <b>SABM</b> ModoAsinc.Balanceado | <b>DM</b> modo desconexión<br><b>UA</b> ACK sin Numerar<br><b>FRMR</b> Trama rechazada | 1    | 1    | 1 | 1 | P   | 1    | 0 | 0 |
|                    | <b>DISC</b> Desconexión          |  | 1    | 1    | 0 | 0 | P   | 0    | 1 | 0 |
|                    |                                  |  | 1    | 1    | 1 | 1 | F   | 0    | 0 | 0 |
|                    |                                  |  | 1    | 1    | 0 | 0 | F   | 1    | 1 | 0 |
|                    |                                  |  | 1    | 1    | 1 | 0 | F   | 0    | 0 | 1 |

DIAGRAMA DE ESTADOS DEL LAP-B





## REFERENCE MODEL of OPEN SYSTEMS INTERCONNECTION

( O . S . I . )

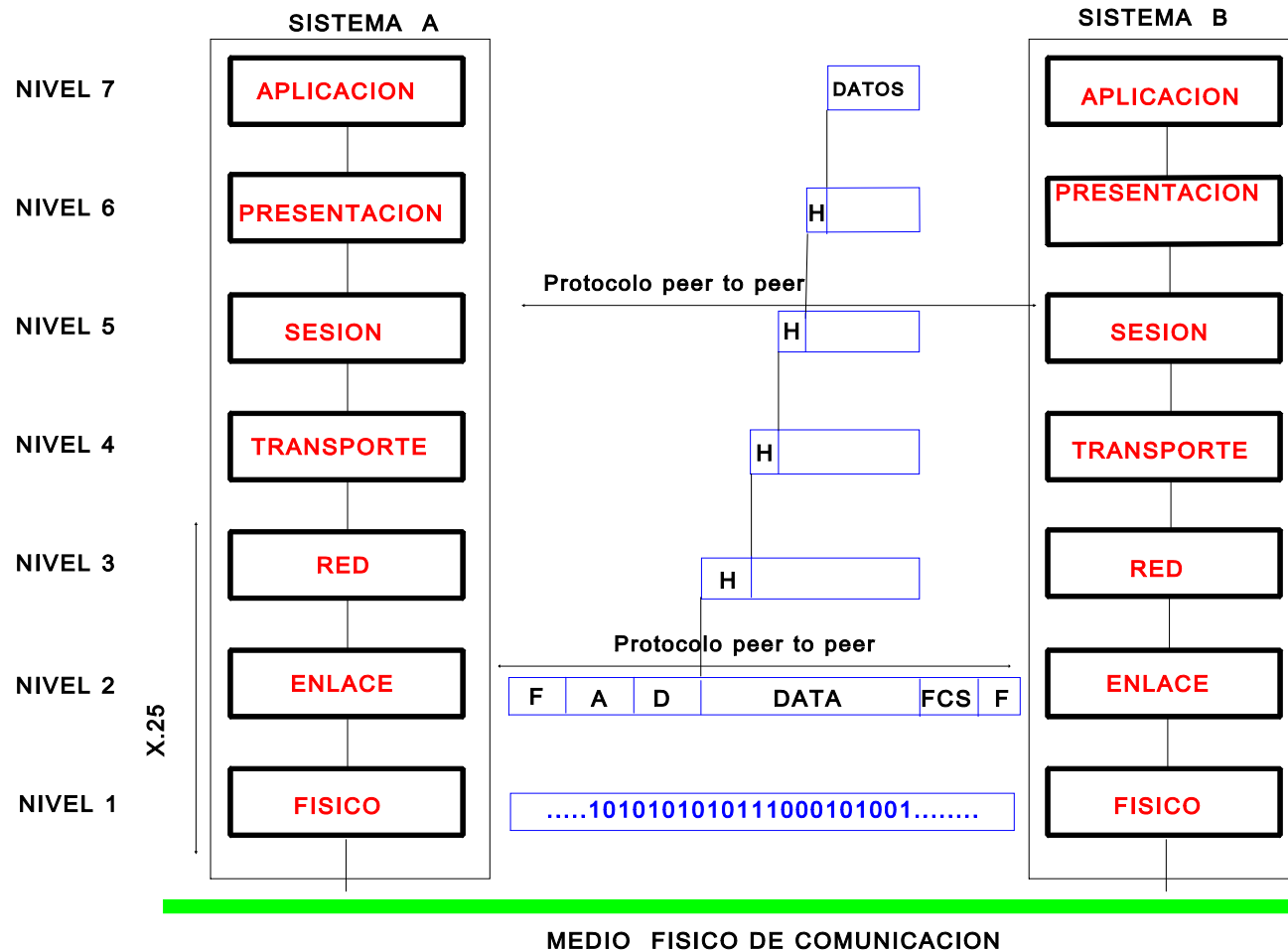
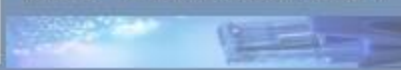


Fig OSI06



# HDLC- LAP-B

