# CONTROL DE ERRORES

Redes de Datos I

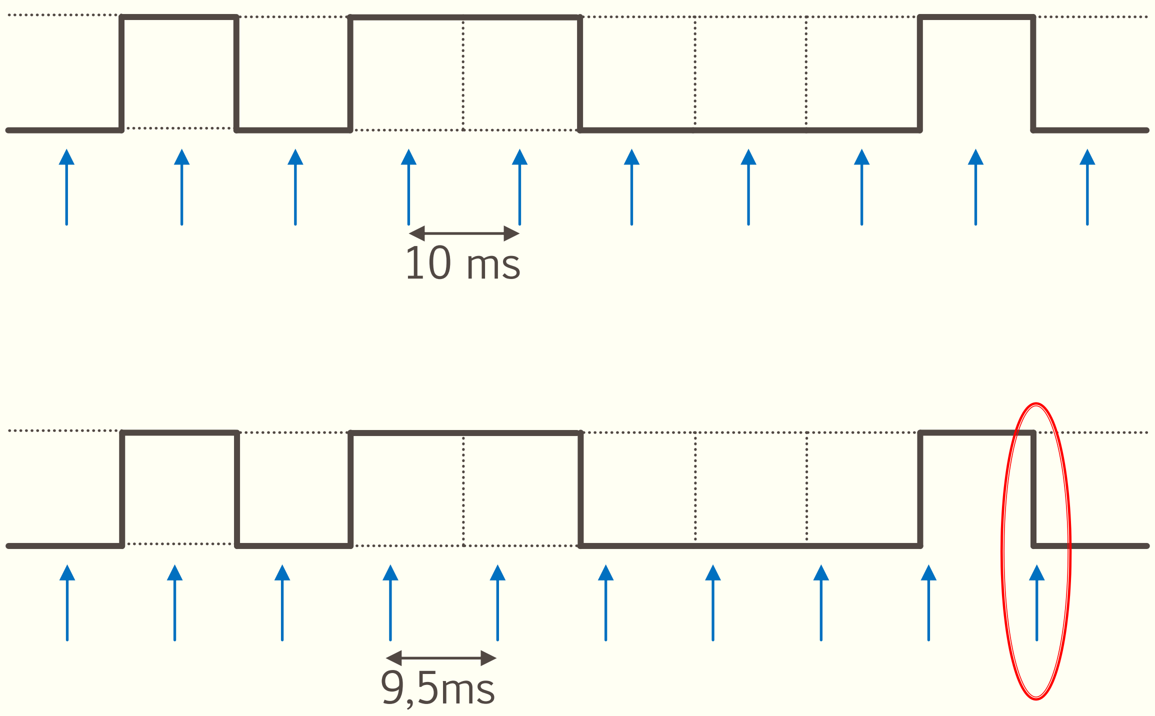
FACULTAD DE INGENIERÍA
UNIVERSIDAD NACIONAL DE LA PLATA

# CONTROL DE ERRORES

- Introducción
- Detección de errores
- Corrección de errores
- Delimitación de tramas

# Introducción

Se debe mantener un sincronismo entre transmisor y receptor




100 bps

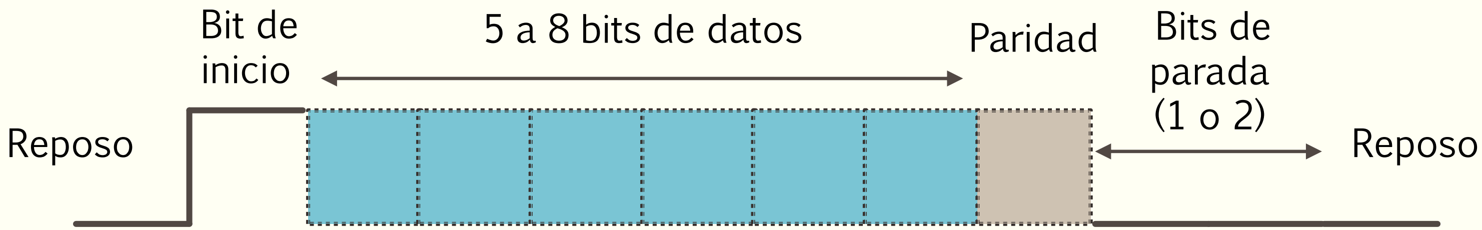
El receptor tiene un reloj sincronizado con el transmisor

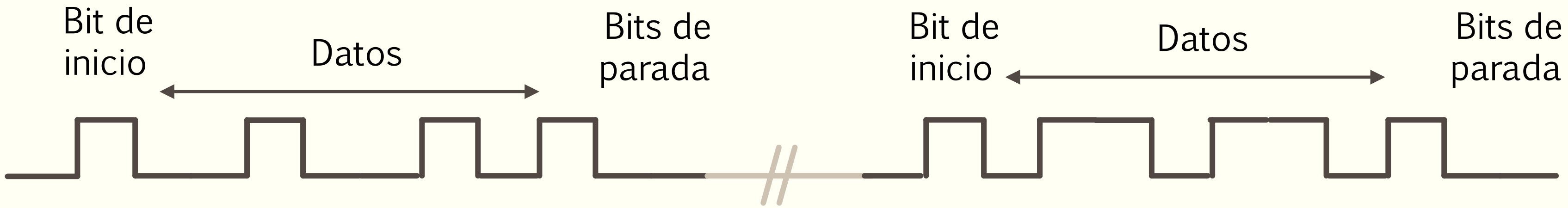
100 bps

El receptor tiene un reloj un 5% más rápido que el del transmisor

# Introducción

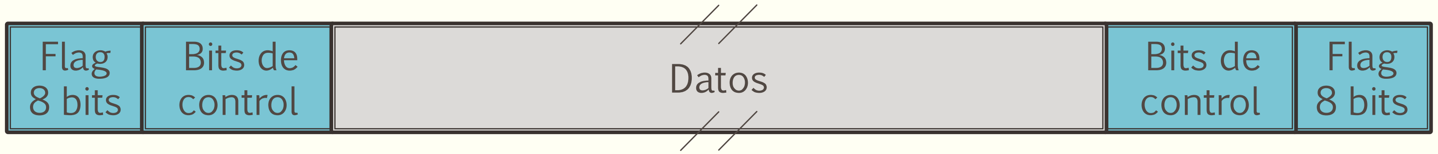
Asincrónica



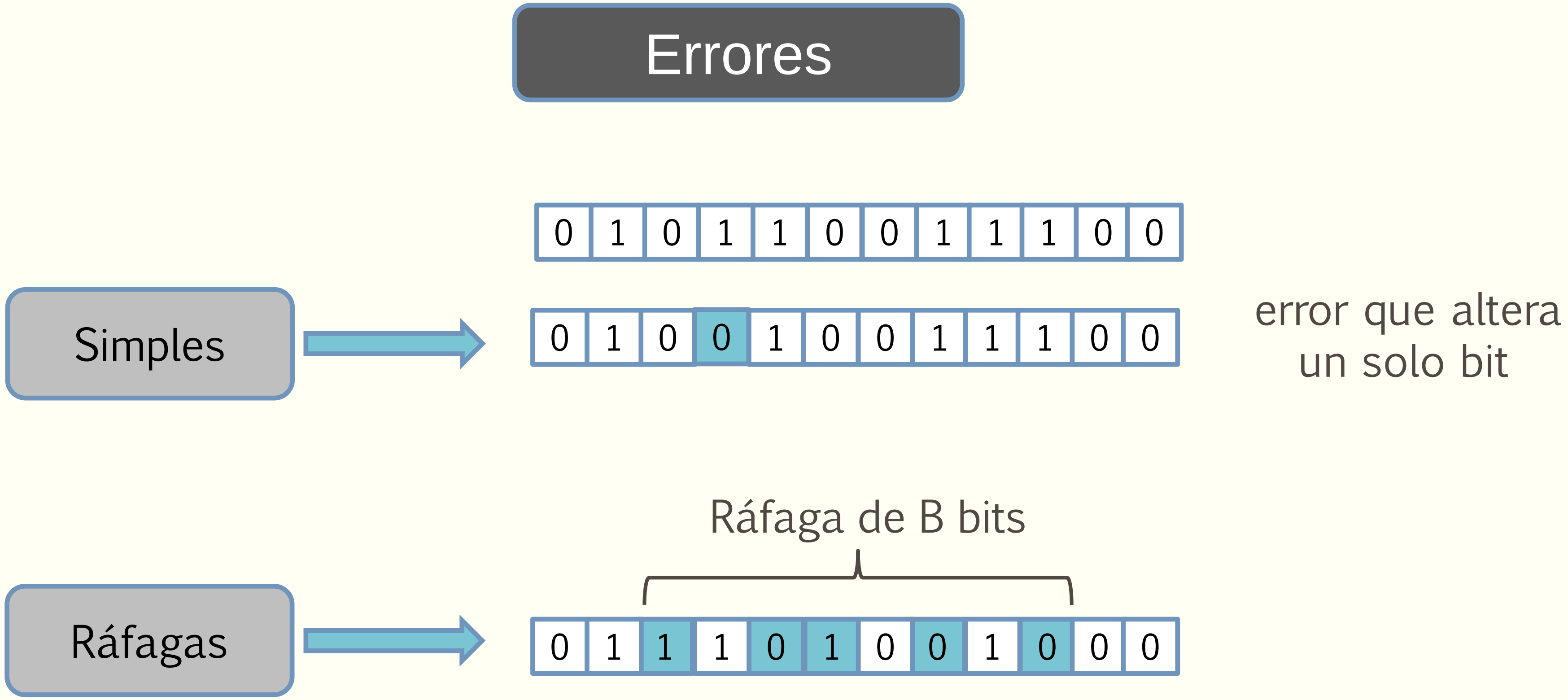
# Técnicas de comunicación

## Sincrónica




- En la transmisión síncrona, cada bloque de bits se transmite como una cadena continua sin utilizar códigos de comienzo o parada.
- Los relojes del transmisor y receptor se deberán sincronizar de alguna manera (reloj independiente o en la misma señal).
- El receptor requiere además que pueda determinar dónde está el comienzo y el final de cada bloque de datos (preámbulo y final).
- Se añaden otros bits que se utilizan en los procedimientos de control del enlace.

# Errores




Una ráfaga de longitud B es una secuencia de B bits en la que el primero, el último y cualquier número de bits intermedios son erróneos.

# Introducción

Existen dos estrategias básicas para manejar los errores:

- **Códigos de detección de errores**: incluyen sólo suficiente **redundancia** para permitir que el receptor sepa que ha ocurrido un error (pero no qué error) y entonces solicite una retransmisión.

  Se utilizan en conjunto con **protocolos ARQ** (Automatic repeat request) (enlace o transporte).

- **Códigos de corrección de errores**: incluyen suficiente **información redundante** para que el receptor pueda deducir cuáles debieron ser los datos transmitidos.

  FEC: Forward Error Correction (Corrección de Errores hacia Adelante).

# CONTROL DE ERRORES

- Introducción
- **Detección de errores**
- Corrección de errores
- Delimitación de tramas

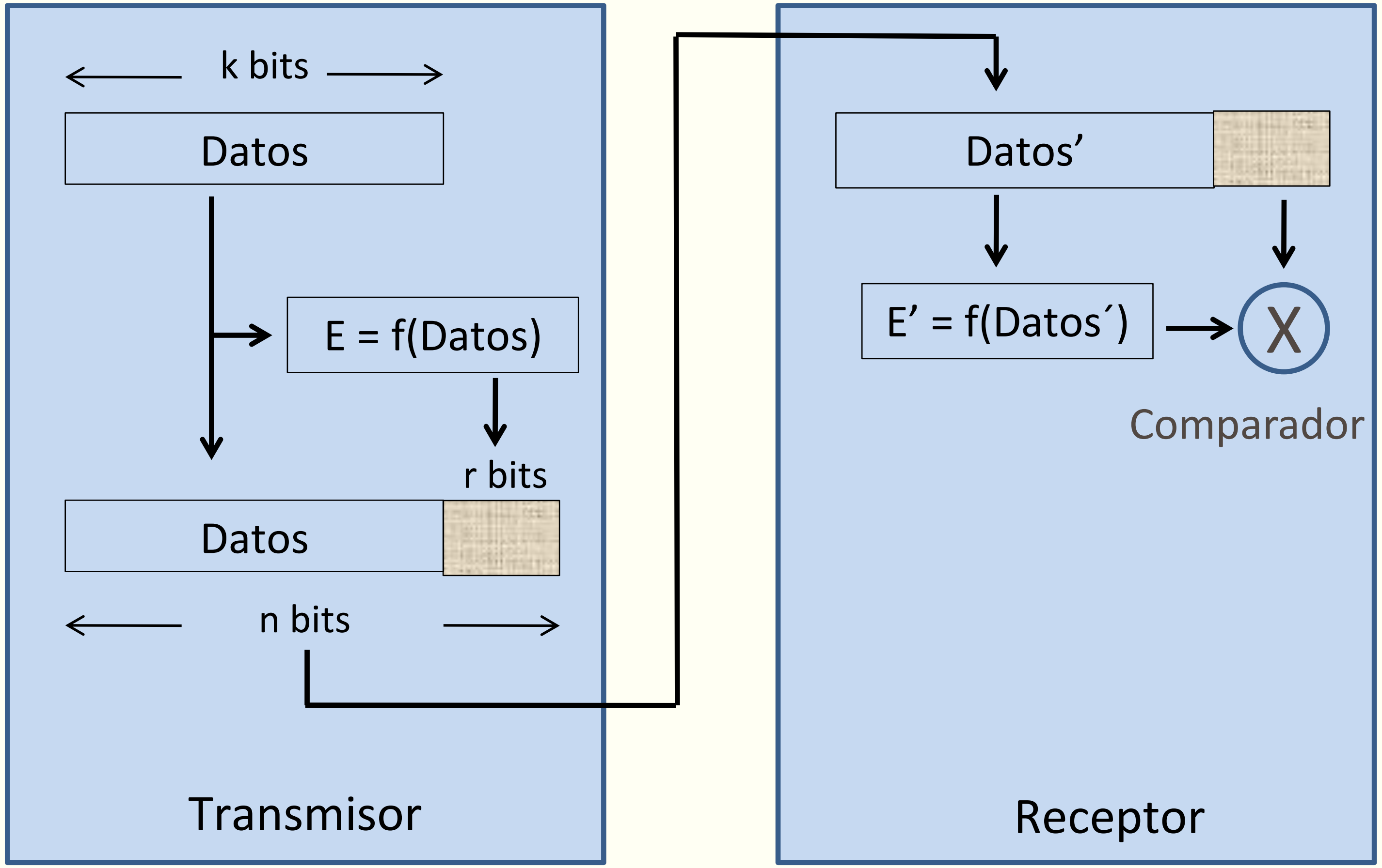
# Detección de errores

## Métodos sencillos de detección de errores

- Redundancia → Envío dos veces el elemento
- Ecoplex → El receptor reenvía el elemento
- Codificación cuenta exacta → Los caracteres tienen número fijo de 1 y 0

# Detección de errores

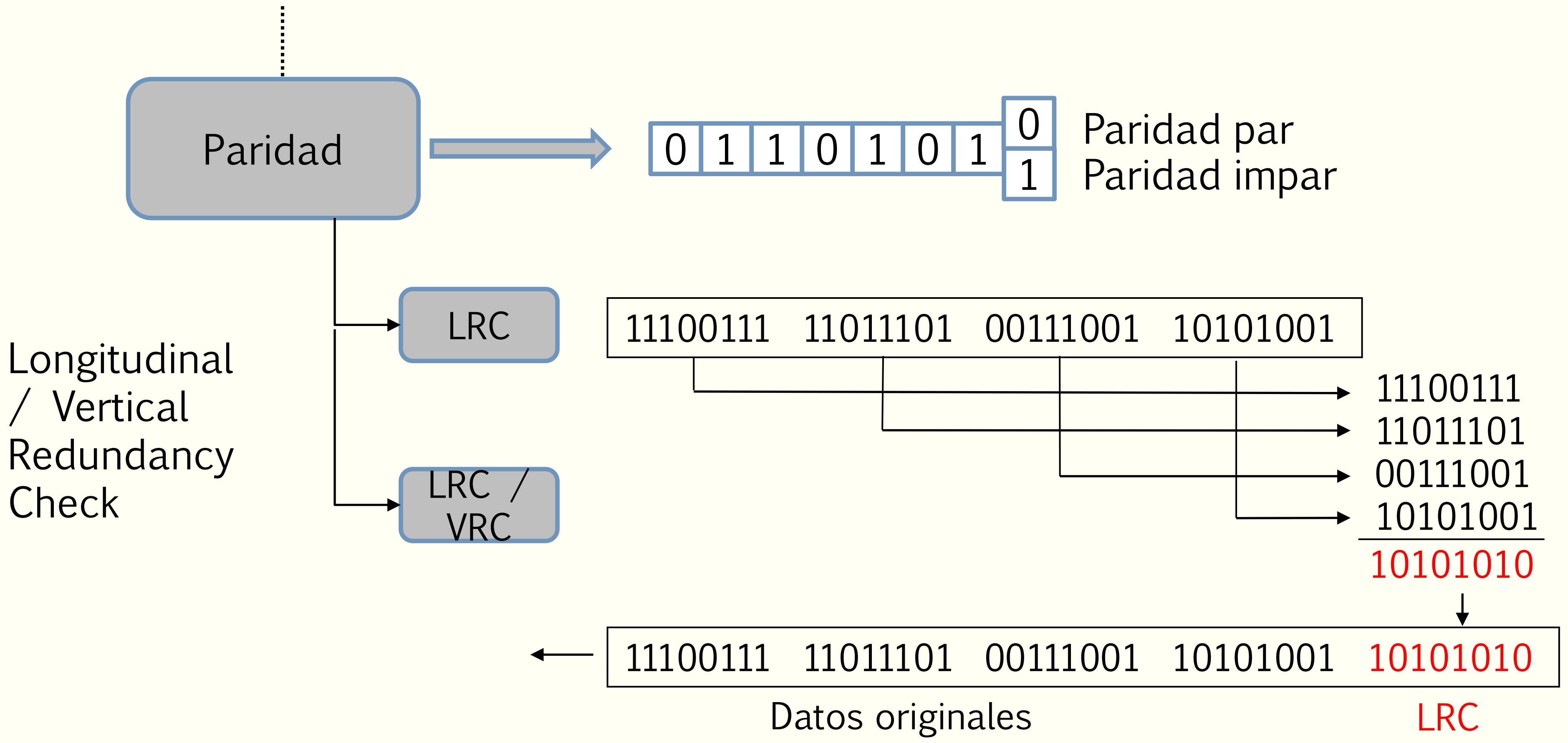
Esquema genérico de un sistema de detección de errores



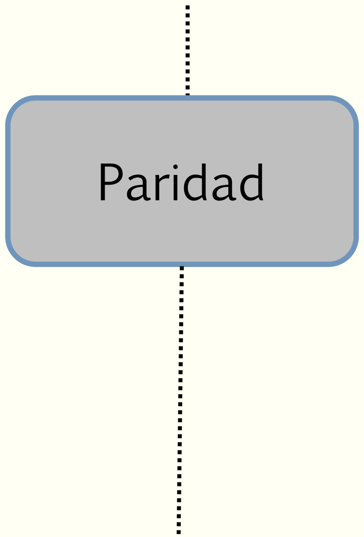

k bits de datos
r bits de verificación
n palabra codificada
$n = k + r$
Tasa de código: $k/n$

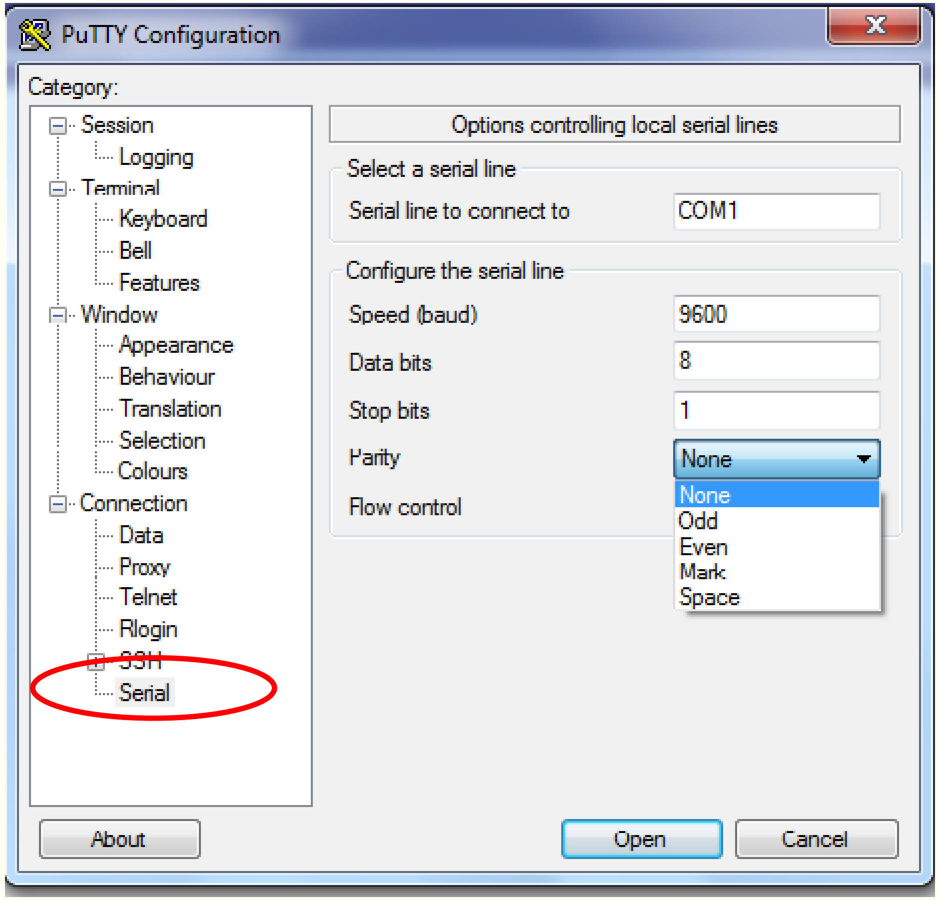
E, E’: códigos de detección de errores
f : función de código de detección de errores

# Detección de errores



# Detección de errores



# Detección de errores

Checksum

Compara la suma de los valores de los caracteres

Se utiliza aritmética complemento a uno, con acarreo

Al resultado, se lo complementa a uno

El receptor realiza la suma, incluido el checksum: el resultado debería ser todos 1s

En Internet se lo utiliza en protocolos como IP, TCP y UDP, como suma de palabras de 16 bits

# Detección de errores

Checksum

| D1 | D2 | D3 | D1+ D2+D3 |
|---|---|---|---|

D1 1001 D2 0001 D3 1001

D1+D2

$$\begin{array}{r} 1001 \\ 0001 \\ \hline 1010 \end{array}$$

D1+D2+D3

$$\begin{array}{r} 1010 \\ 1001 \\ \hline 10011 \end{array} \qquad \begin{array}{r} 0011 \\ 1 \\ \hline 0100 \end{array}$$

(complemento a 1)
1011

| 1001 | 0001 | 1001 | 1011 |
|---|---|---|---|

# Detección de errores

CRC

- División de polinomios
- Aritmética módulo 2, sin acarreo

| | |
|---|---|
| 0+0 (mod 2) = 0 | 0-0 (mod 2) = 0 |
| 0+1 (mod 2) = 1 | 0-1 (mod 2) = 1 |
| 1+0 (mod 2) = 1 | 1-0 (mod 2) = 1 |
| 1+1 (mod 2) = 0 | 1-1 (mod 2) = 0 |

- Representación polinomial

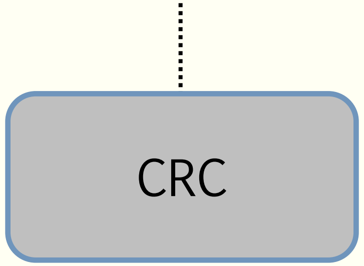
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|

$$0x^6+1x^5+1x^4+0x^3+1x^2+0x^1+1x^0$$

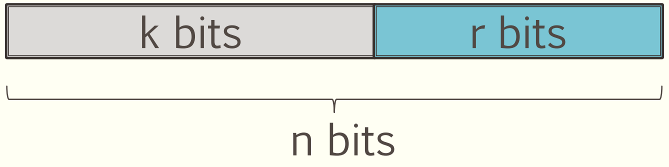
$$\Downarrow$$

$$x^5 + x^4 + 1x^2 + 1x^0$$

# Detección de errores

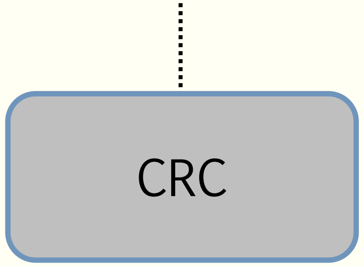


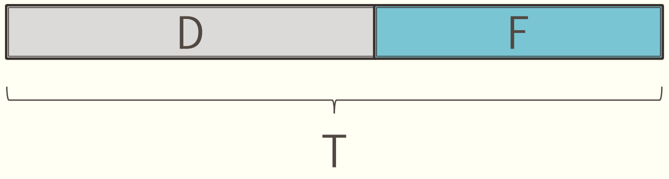

Dado un bloque o mensaje de k bits, el transmisor genera una secuencia de r bits, denominada secuencia de comprobación de la trama (FCS, Frame Check Sequence), de tal manera que la trama resultante, con n bits, sea divisible por algún número predeterminado




El receptor dividirá la trama recibida entre ese número y si no hay resto en la división, supondrá que no ha habido errores.

# Detección de errores



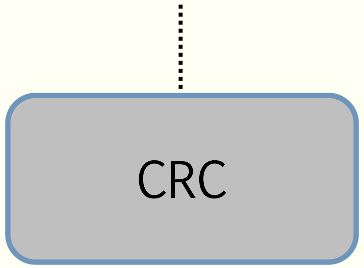

T: trama de n bits
D: mensaje con k bits
F: (n-k) bits de FCS
P: patrón de n-k+1 bits (divisor)

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

- El objetivo es que la división T/P no tenga resto (T divisible entre P).
- Si se divide $2^{n-k}D$ entre $P$ $\Longrightarrow$ $\dfrac{2^{n-k}D}{P} = Q + \dfrac{R}{P}$
- Transmisor: $FCS = F = R$ $\Longrightarrow$ $T = 2^{n-k}D + R$
- Receptor: $\dfrac{T}{P} = \dfrac{2^{n-k}D + \mathrm{R}}{P} = \dfrac{2^{n-k}D}{P} + \dfrac{\mathrm{R}}{P}$ $\Longrightarrow$ $\dfrac{T}{P} = Q + \dfrac{\mathrm{R}}{P} + \dfrac{\mathrm{R}}{P} = Q$

Resto nulo

# Detección de errores




Generación

- $M(x)$ : Polinomio con mensaje a transmitir (grado k)
- $P(x)$ : Polinomio generador (grado n)

- Multiplicar $M(x)$ por $x^n$
- Dividir $x^n M(x)$ por $P(x)$
- Obtener el resto de la división $C(x)$
- Transmito: $F(x) = x^n M(x) + C(x)$

Recepción

- Recibo $F'(x)$
- Divido $F'(x)$ por $P(x)$
- Si el resto es igual a cero, sin errores

# Detección de errores

CRC

Transmisor

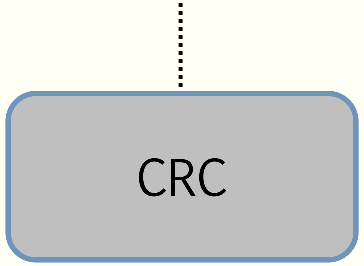
```
101000110100000 | 110101
110101          |----------
 111011           1101010110
 110101
   111010
   110101
     111110
     110101
       101100
       110101
        110010
        110101
         01110
```

T= 101000110101110

mensaje D = 1010001101 (10 bits)
patrón P = 110101 (6 bits)
FCS R = a calcular (5 bits)

El mensaje se multiplica por $2^5$ :
101000110100000
y luego se divide por P

# Detección de errores

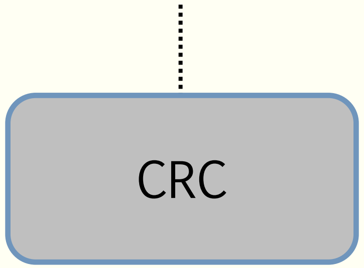



Receptor

```
1 0 1 0 0 0 1 1 0 1 0 1 1 1 0 | 1 1 0 1 0 1
1 1 0 1 0 1                   |--------------
-----------                     1 1 0 1 0 1 0 1 1 0
  1 1 1 0 1 1
  1 1 0 1 0 1
  -----------
    1 1 1 0 1 0
    1 1 0 1 0 1
    -----------
      1 1 1 1 1 0
      1 1 0 1 0 1
      -----------
        1 0 1 1 1 1
        1 1 0 1 0 1
        -----------
          1 1 0 1 0 1
          1 1 0 1 0 1
          -----------
            0 0 0 0 0
```

Divido el mensaje recibido por P
Si el resto es nulo, no hubo errores

# Detección de errores




- Detecta todos los errores simples, si P(x) tiene mas de un término y $x^0 = 1$
- Detecta todos los errores dobles, si P(x) no divide a $x^t + 1$ (t entre 0 y n – 1)
- Detecta cualquier número de errores impares, si P(x) contiene (x + 1)
- Detecta todas las ráfagas de error de longitud $<=$n-k
- Detecta todas las ráfagas de error de longitud n-k+1, con probabilidad $1- 2^{-(n-k-1)}$
- Detecta todas las ráfagas de error de longitud superior a n-k+1, con probabilidad $1- 2^{-(n-k)}$

# Detección de errores

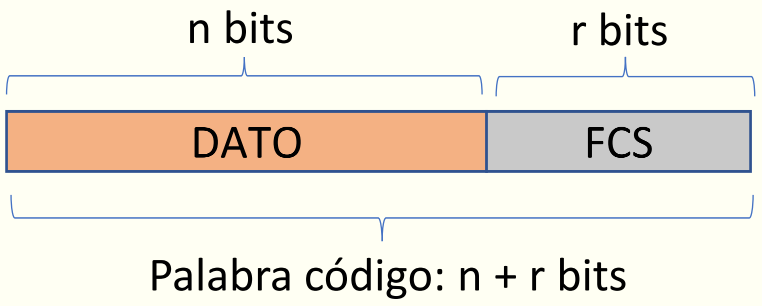
CRC

- Características deseables de un polinomio generador:
  - Contener al menos dos términos
  - El coeficiente de $x^0$ debería ser 1
  - No ser divisible por $x^t + 1$ (t entre 2 y n – 1)
  - Contener el factor x + 1
- Algunos CRCs característicos:

- CRC-16 = $X^{16}+X^{15}+X^2+1$ (HDLC)
- CRC-32 = $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$ (LANs)


Facultad de Ingeniería
Universidad Nacional de La Plata

# Detección de errores




1-a) Dada el siguiente dato 110100 y el polinomio generador 101, calcular la palabra código transmitida

Dato: 6 bits
Polinomio: 3 bits
El FCS debe ser de un grado menor que el polinomio generador: 2 bits
Palabra código: 6 + 2 bits

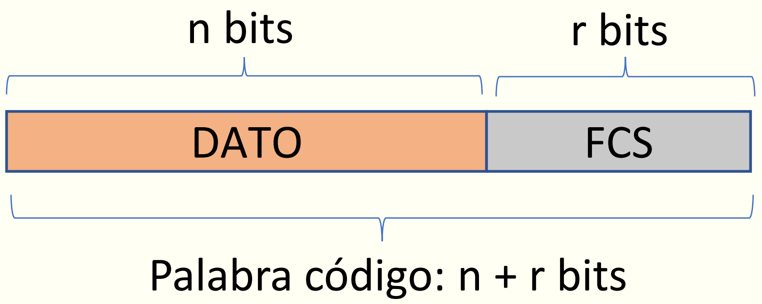
Agrego la misma cantidad de bits que el FCS

```
1 1 0 1 0 0 0 0  | 1 0 1
1 0 1            |------
-----              111010
  1 1 1
  1 0 1
  -----
    1 0 0
    1 0 1
    -----
    0 1 0 0
      1 0 1
      -----
      0 1 0
      0 0 0
      -----
        1 0
```

Palabra código transmitida:

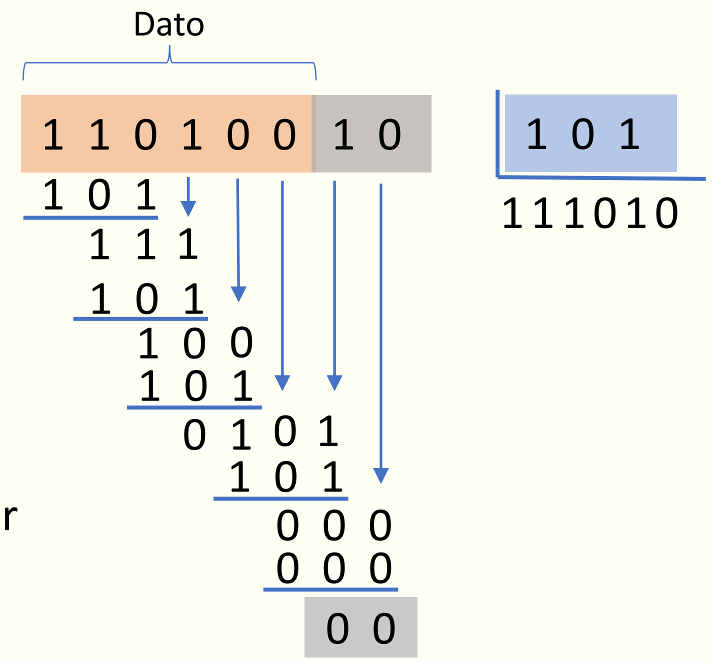
| 1 1 0 1 0 0 | 1 0 |
|---|---|

# Detección de errores




1-b) Verificar si la palabra código recibida 11010010 tiene errores o no.

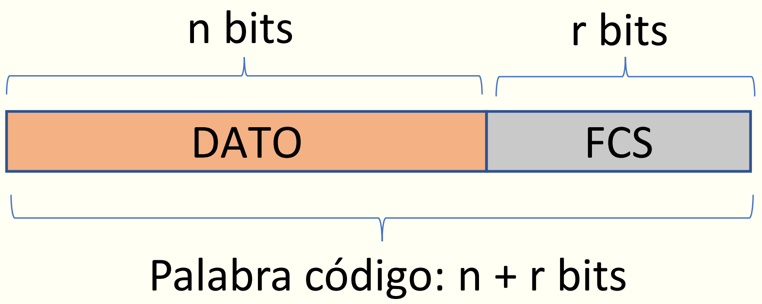
Dato: 6 bits
Polinomio: 3 bits
El FCS debe ser de un grado menor que el polinomio generador: 2 bits
Palabra código: 6 + 2 bits




Palabra código recibida sin error

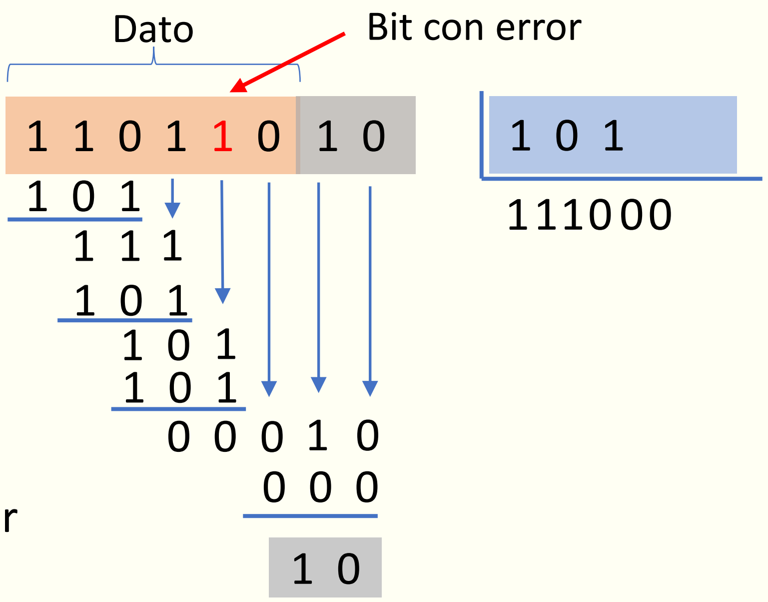
1 1 0 1 0 0 ✓

# Detección de errores




1-c) Verificar si la palabra código recibida 11011010 tiene errores o no.

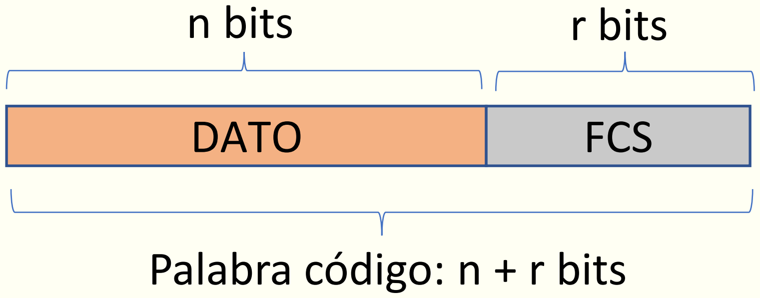
Dato: 6 bits
Polinomio: 3 bits
El FCS debe ser de un grado menor que el polinomio generador: 2 bits
Palabra código: 6 + 2 bits




Palabra código recibida con error

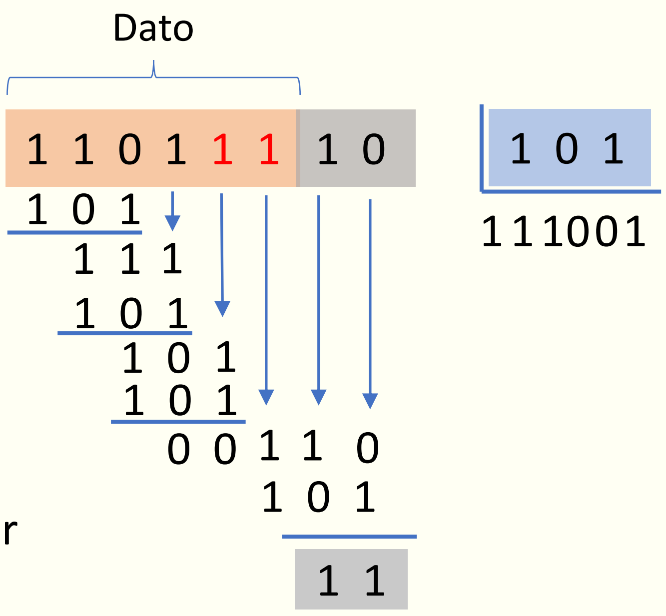
1 1 0 1 0 0

# Detección de errores




1-d) Verificar si la palabra código recibida 11011110 tiene errores o no.

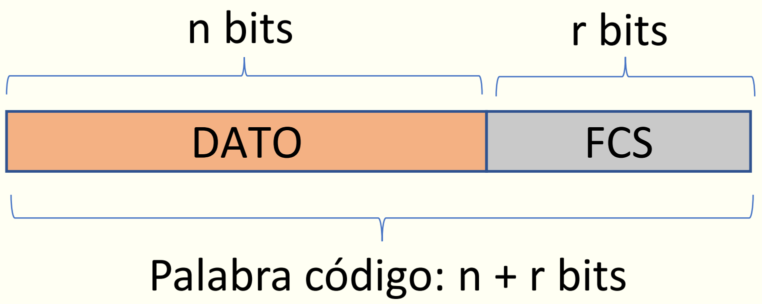
Dato: 6 bits
Polinomio: 3 bits
El FCS debe ser de un grado menor que el polinomio generador: 2 bits
Palabra código: 6 + 2 bits




Palabra código recibida con error

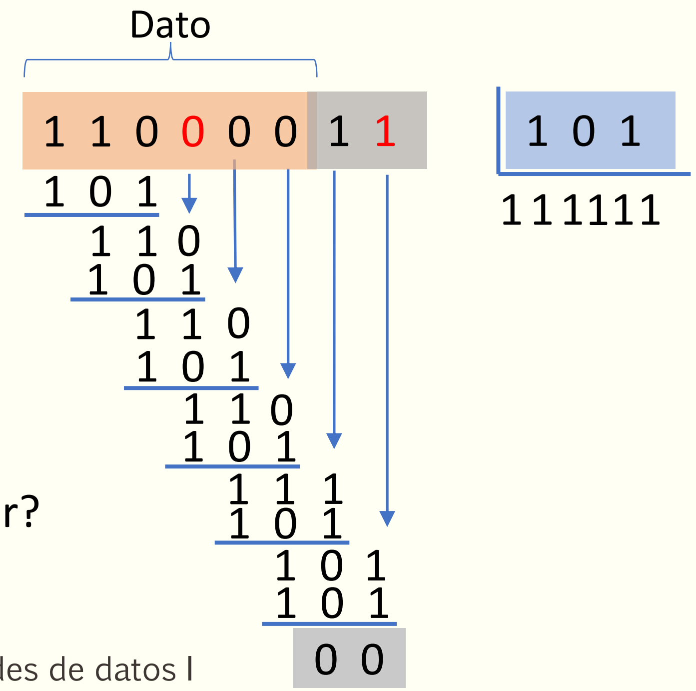
1 1 0 1 0 0

# Detección de errores




1-e) Verificar si la palabra código recibida 11000011 tiene errores o no.

Dato: 6 bits
Polinomio: 3 bits
El FCS debe ser de un grado menor que el polinomio generador: 2 bits
Palabra código: 6 + 2 bits

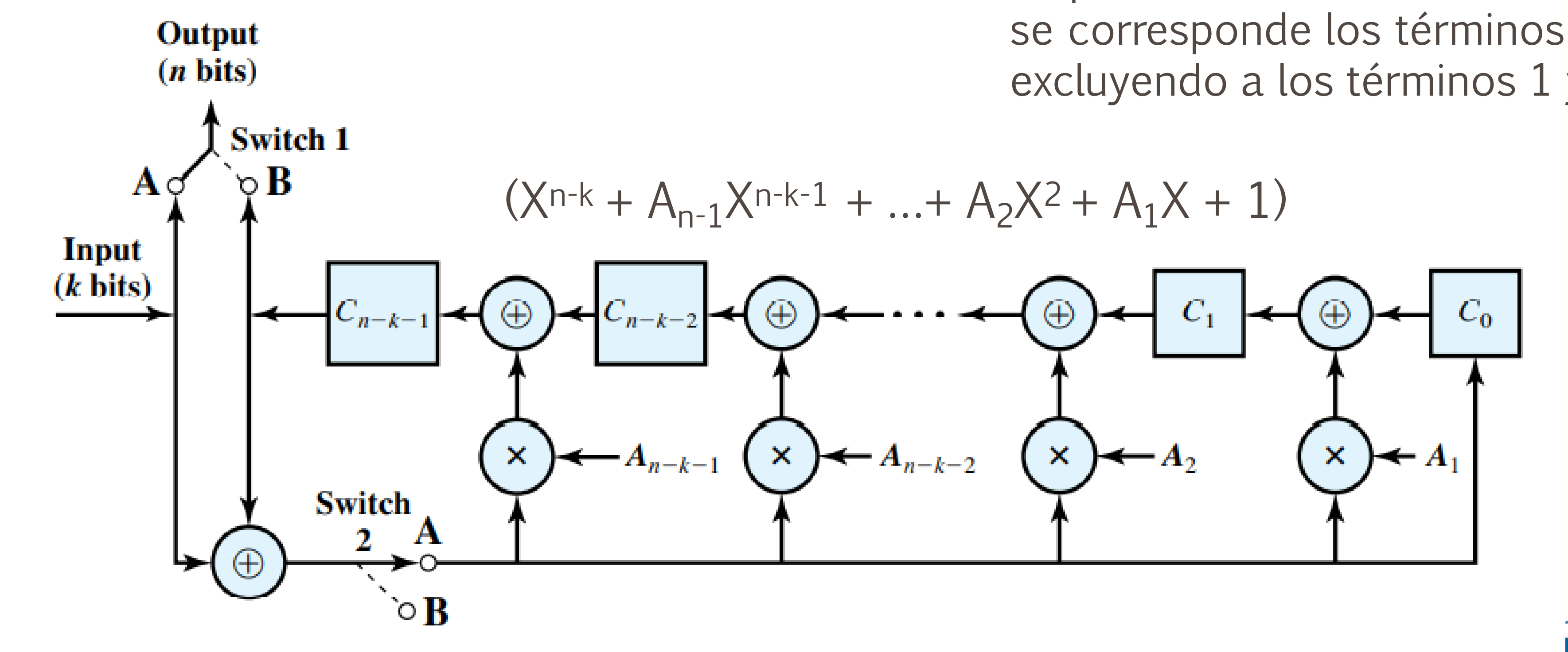



¿Palabra código recibida sin error?

1 1 0 0 0 0 ✗

# Detección de errores

- Implementación de un CRC
  - El registro contendrá n-k bits, igual a la longitud de la FCS.
  - Hay hasta n-k puertas exclusive-or.
  - La presencia o ausencia de una compuerta se corresponde los términos de P(X), excluyendo a los términos 1 y $X^{n-k}$.




$$(X^{n-k} + A_{n-1}X^{n-k-1} + \ldots + A_2X^2 + A_1X + 1)$$

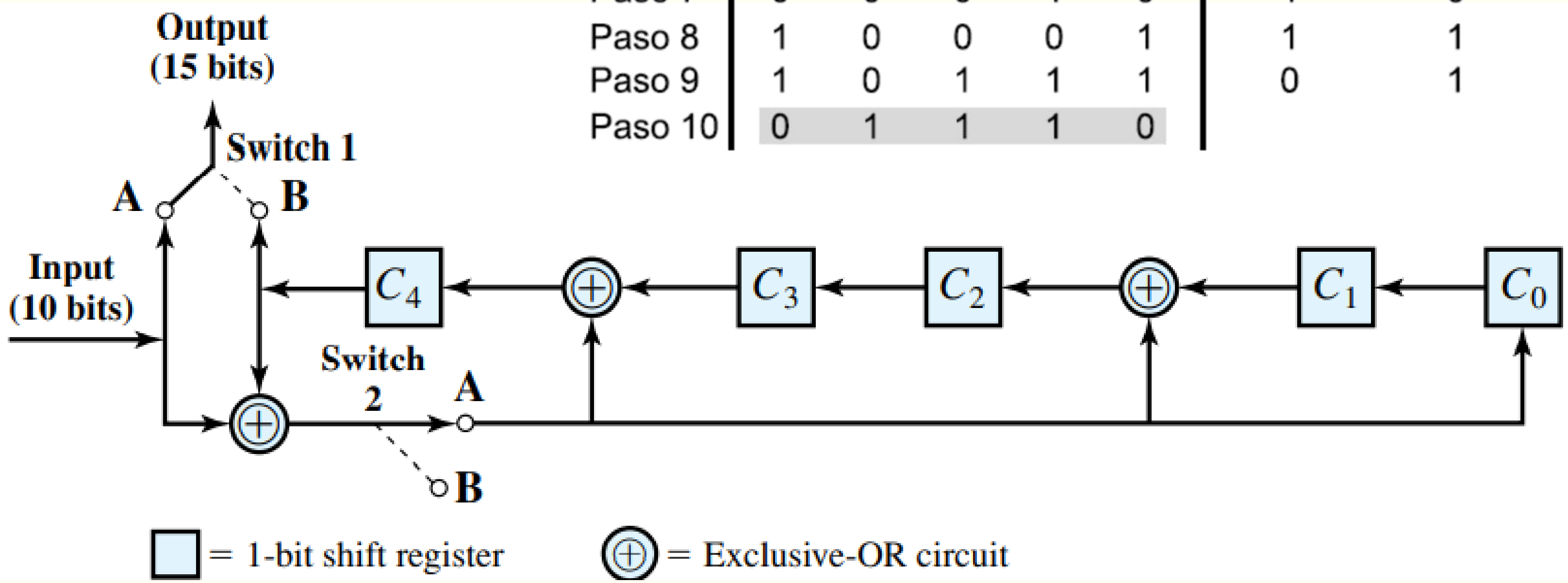
# Detección de errores

Datos D=1010001101
$D(X)=X^9+X^7+X^3+X^2+1$

Divisor P=110101
$P(X)=X^5+X^4+X^2+1$

| | $C_4$ | $C_3$ | $C_2$ | $C_1$ | $C_0$ | $C_4 \oplus C_3 \oplus I$ | $C_4 \oplus C_1 \oplus I$ | $C_4 \oplus I$ | $I$ = entrada |
|---|---|---|---|---|---|---|---|---|---|
| Inicial | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Paso 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| Paso 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| Paso 3 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| Paso 4 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| Paso 5 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| Paso 6 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| Paso 7 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| Paso 8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| Paso 9 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| Paso 10 | 0 | 1 | 1 | 1 | 0 | | | | |

Mensaje a enviar



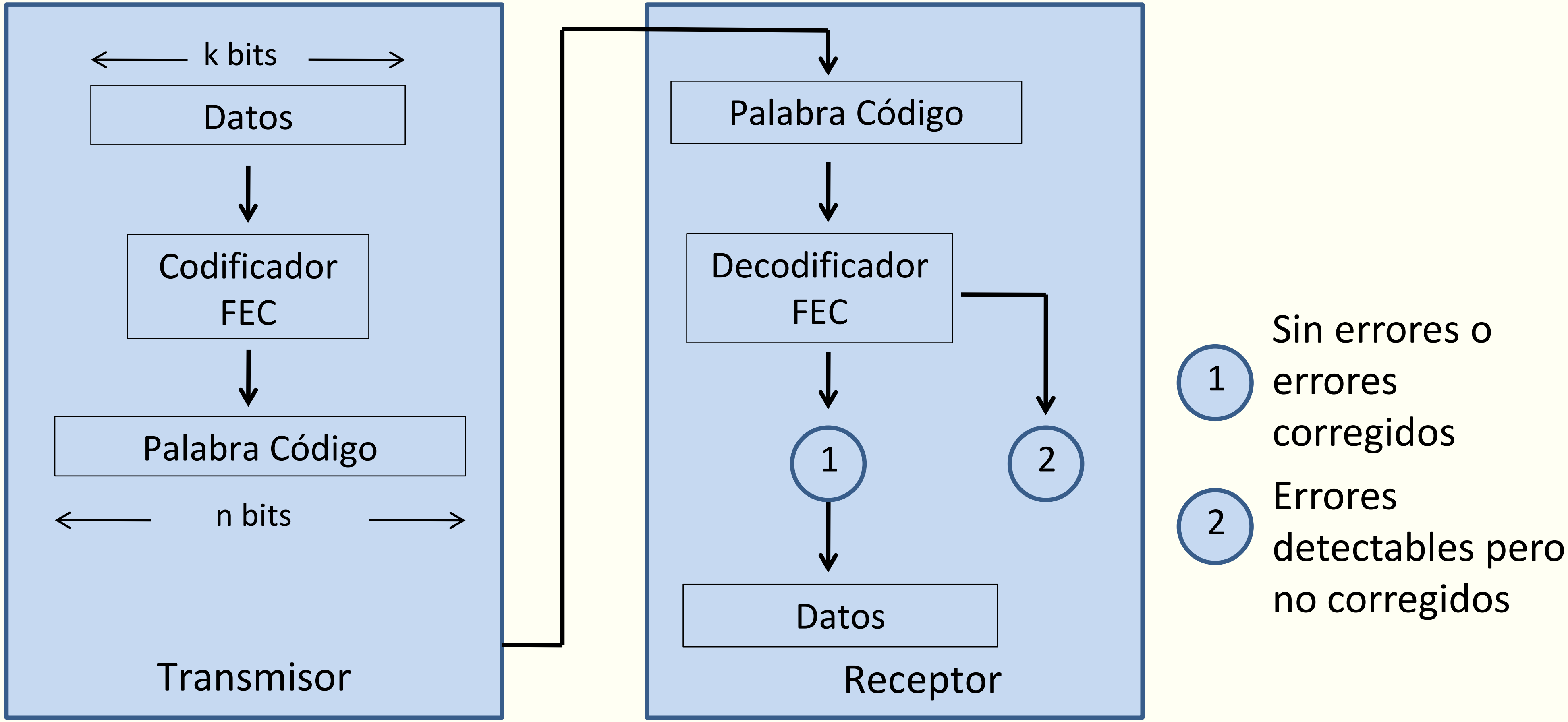


Facultad de Ingeniería
Universidad Nacional de La Plata

# CONTROL DE ERRORES

- Introducción
- Detección de errores
- Corrección de errores
- Delimitación de tramas

# Corrección de errores




La idea del corrector de errores es que el receptor sea capaz de corregir errores usando exclusivamente los bits recibidos en la transmisión.

# Corrección de errores

## Distancia de Hamming:

Dadas dos palabras de n bits $v_1$ y $v_2$ se define como:
d($v_1; v_2$) el número de bits en que $v_1$ y $v_2$ difieren

$v_1$ = 0 1 1 0 0 1
$v_2$ = 1 1 1 0 1 1

$\Longrightarrow$ d($v_1; v_2$) = 2

Su significado es que, si dos palabras codificadas están separadas una distancia de Hamming d, se requerirán d errores de un solo bit para convertir una en la otra.

# Corrección de errores

## Códigos de bloque

Por ejemplo:

| Bloque de datos | Código de palabra |
|---|---|
| 00 | 00000 |
| 01 | 00111 |
| 10 | 11001 |
| 11 | 11110 |

k=2 bits de bloque de datos

n=5 bits de código de palabra

Si recibo 11101 (inválida)

d (11101, 00000) = 4
d (11101, 00111) = 3
d (11101, 11001) = 1
d (11101, 11110) = 2

distancia mínima entre todas las posibles: elijo esa palabra código

Si recibo 01010 (inválida)

d (01010, 00000) = 2
d (01010, 00111) = 3
d (01010, 11001) = 3
d (01010, 11110) = 2

¿Cuál elijo?

# Corrección de errores

## Códigos de bloques

- Un código de bloque (n, k) codifica k bits de datos en palabras-código de n bits.
- Generalmente, cada palabra-código válida incluye a los k bits de datos originales y les añade (n-k) bits de comprobación para constituir la palabra-código de n bits.
- En un código de bloque (n, k), habrá $2^k$ palabras-código válidas de un total de $2^n$ posibles.
- Se define la **redundancia del código** como el cociente del número de bits redundantes entre el número de bits de datos (n-k)/k.
- Se define **la tasa del código** como el cociente del número de bits de datos entre el número de bits totales, k/n. (medida del ancho de banda adicional)

# Corrección de errores

- Para un código constituido por las palabras-código $w_1$, $w_2$, ..., $w_s$, en el que $s = 2^n$, se define la distancia mínima del código como:

$$d_{min} = \min d(w_i, w_j) \text{ para } i \neq j$$

- Se puede demostrar que (con t entero positivo):
  - El número de errores t que se pueden detectar es: $t = d_{min} - 1$
  - Para corregir errores de hasta t bits, se requiere que $d_{min} \geq 2t + 1$

# Corrección de errores

## Códigos de bloque

Por ejemplo:

| Bloque de datos | Código de palabra |
|---|---|
| 00 | 00000 |
| 01 | 00111 |
| 10 | 11001 |
| 11 | 11110 |

k=2 bits de bloque de datos

n=5 bits de código de palabra

Código (5,2)

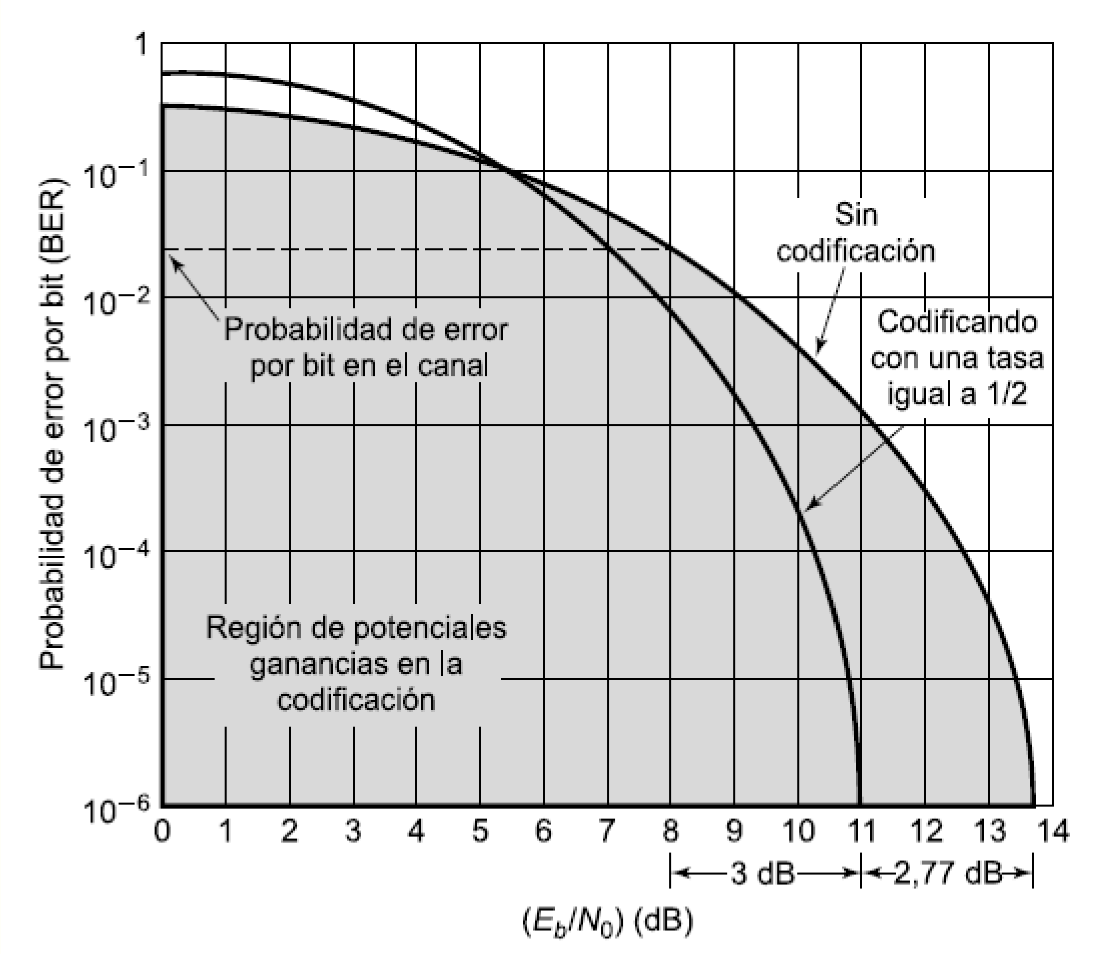
Tasa del código: 2/5

d (00000, 00111) = 3
d (00000, 11001) = 3
d (00000, 11110) = 4
d (00111, 11001) = 4
d (00111, 11110) = 3
d (11001, 11110) = 3

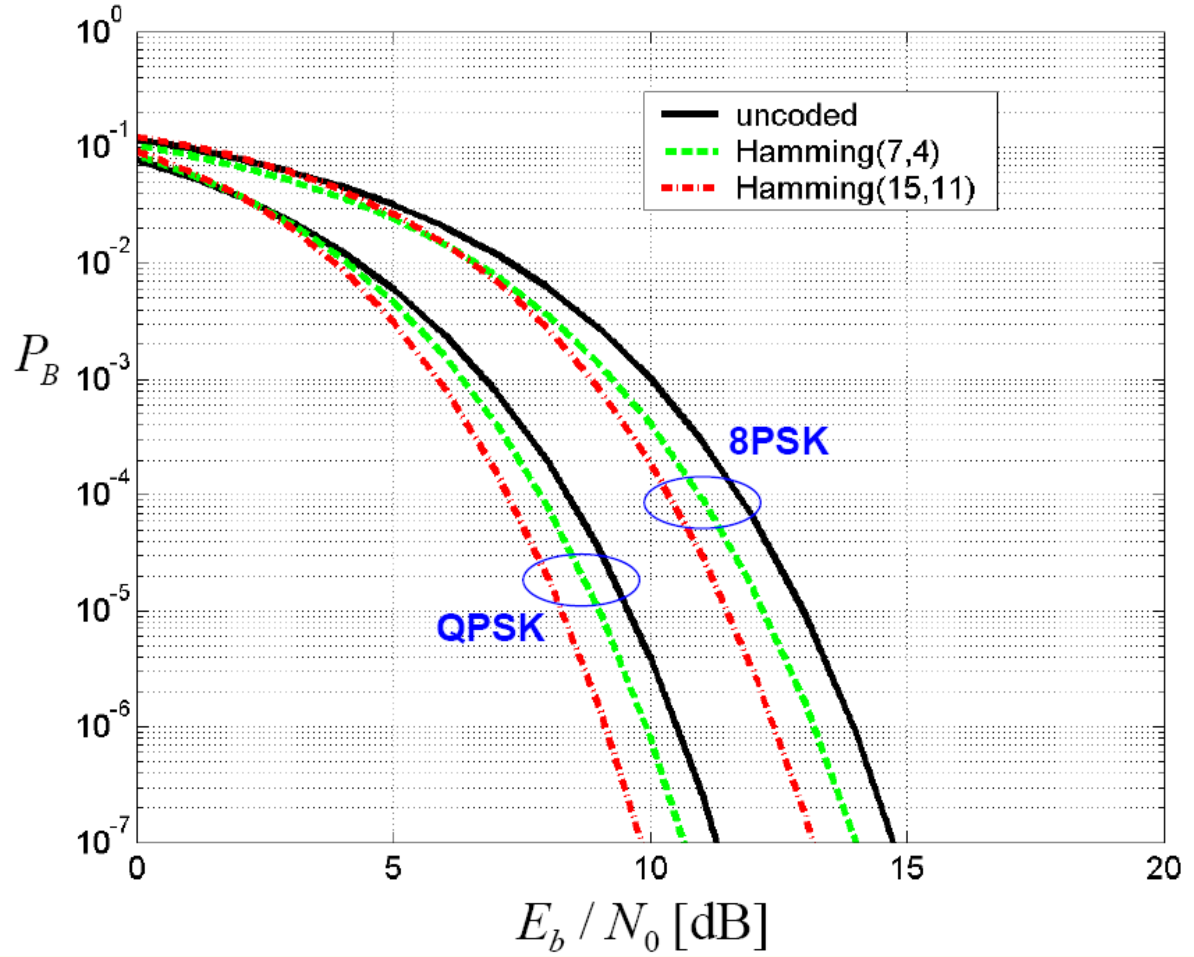
$d_{min}$ = 3

Puede detectar 2 errores
y corregir 1 error

# Corrección de errores



# Corrección de errores



# CONTROL DE ERRORES

- Introducción
- Detección de errores
- Corrección de errores
- **Delimitación de tramas**

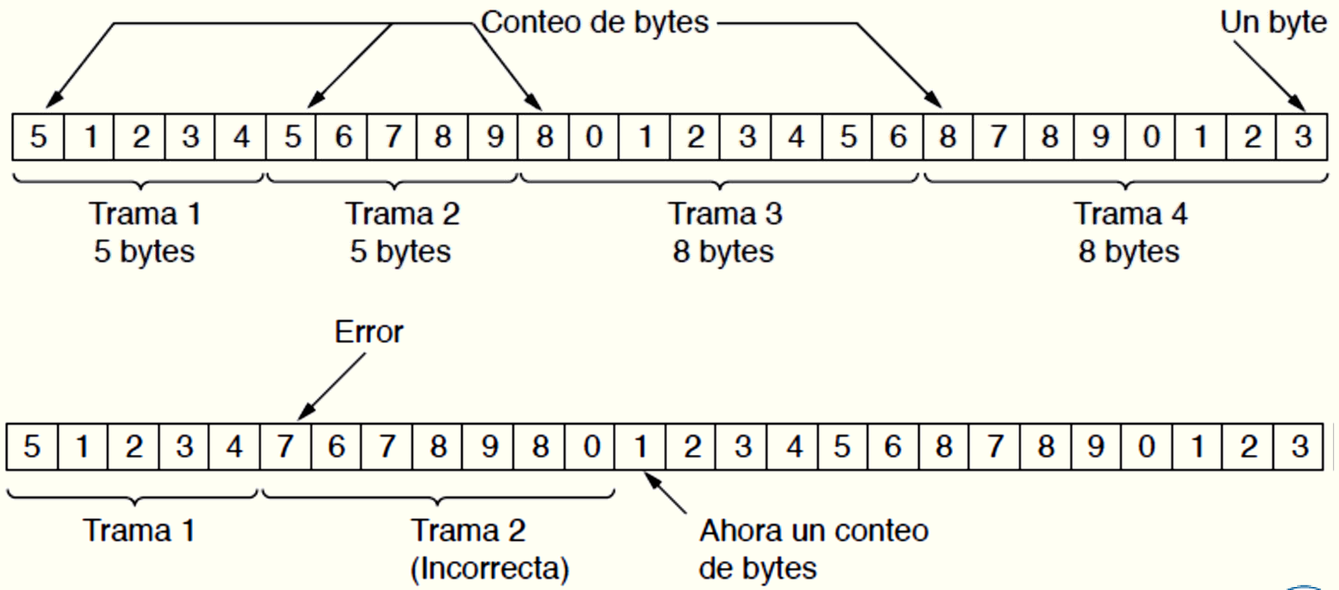
# Delimitación de tramas

- La capa física proporciona la sincronización de bits para garantizar que el emisor y el receptor utilicen las mismas duraciones de bits y la misma temporización.
- La capa de enlace de datos, por otro lado, necesita empaquetar los bits en tramas, de modo que cada trama sea distinguible de otra
- Si las tramas son de longitud variable, es necesario delimitarlas: entramado
- Dos opciones:
  - Entramado orientado a carácter
  - Entramado orientado a bit

# Entramado orientado a carácter

## Conteo de bytes

- Un campo en el encabezado especifica el número de bytes en la trama.
- El conteo se puede alterar debido a un error de transmisión.
- Raras veces se utiliza el método de conteo de bytes por sí solo.



# Entramado orientado a carácter

## Bytes bandera con relleno de bytes

- Cada trama inicia y termina con bytes especiales.
- Con frecuencia se utiliza el mismo byte, (flag o byte bandera) como delimitador inicial y final.
- Generalmente se utiliza el carácter 0x7E (01111110 en binario)

| | | | | |
|---|---|---|---|---|
| Flag 8 bits | Bits de control | Datos | Bits de control | Flag 8 bits |

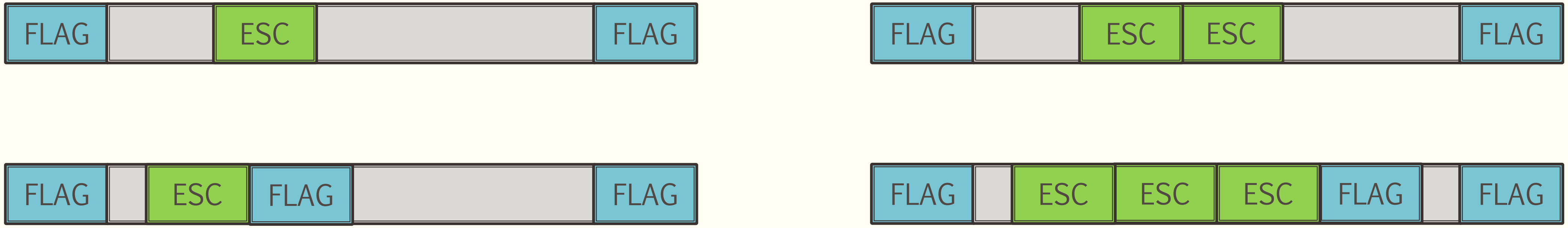
- El problema surge si el flag se repite dentro de los datos.

# Entramado orientado a carácter

## Bytes bandera con relleno de bytes

- Si el flag aparece dentro de los datos, se le añade el carácter de escape ESC (0x7D)


- ¿y si aparece el carácter ESC en los datos? Se le añade otro carácter ESC

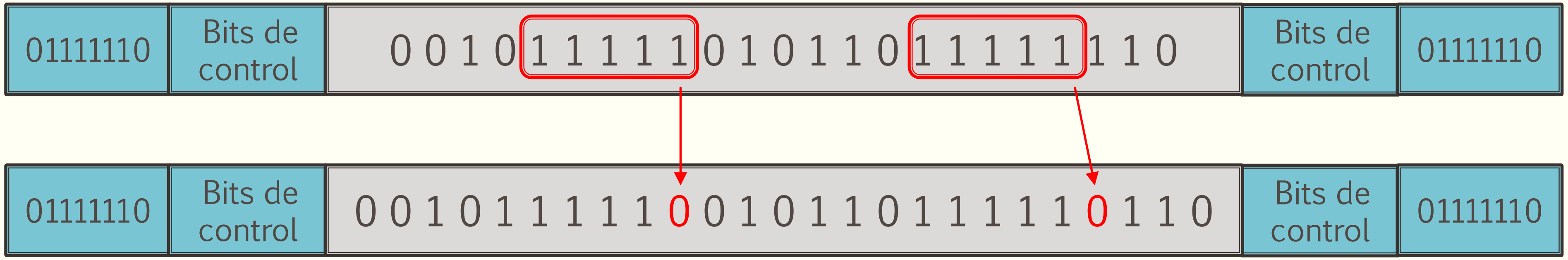



Byte stuffing

# Entramado orientado a bit

- Se utiliza un flag delimitador (generalmente 01111110) para delimitar el inicio y fin de la trama, que contiene un número variable de bits.


- ¿y si aparece el flag en los datos? Si en los datos se encuentran cinco bits 1 consecutivos, se inserta un 0 como relleno en el flujo de bits de salida.




Bit stuffing

# Violaciones de codificación de la capa física

- La codificación de bits como señales incluye a menudo redundancia para ayudar al receptor.
- Esta redundancia significa que algunas señales no ocurrirán en los datos regulares.
- Por ejemplo, en el código de línea 4B/5B se asignan 4 bits de datos a 5 bits de señal para asegurar suficientes transiciones de bits. Esto significa que no se utilizan 16 de las 32 posibles señales.
- Podemos usar algunas señales reservadas para indicar el inicio y el fin de las tramas. Se usan "violaciones de código" para delimitar tramas.
- Como hay señales reservadas, es fácil encontrar el inicio y final de las tramas y no hay necesidad de rellenar los datos.