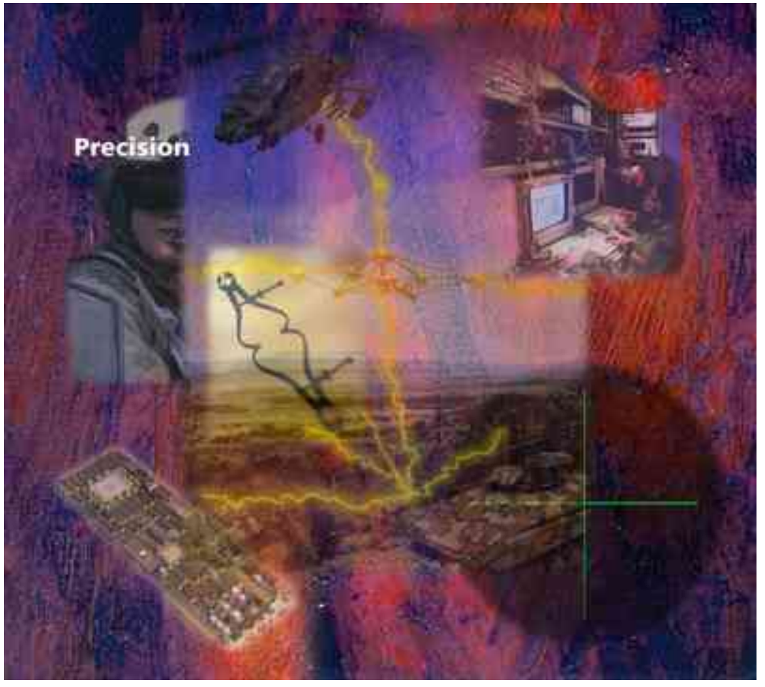
COMUNICACIONES



UT Nº 6
TRATAMIENTO DE ERRORES

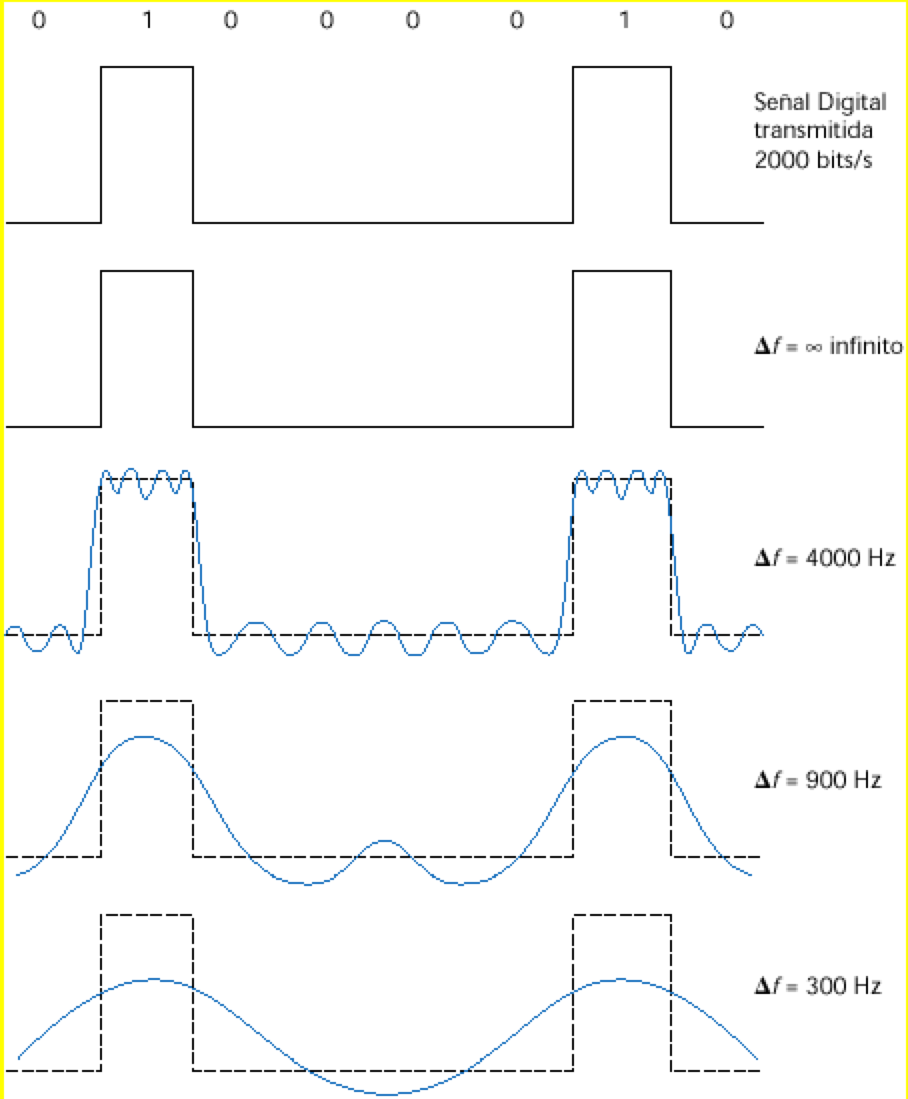
Ingeniero ALEJANDRO ECHAZÚ
aechazu@comunicacionnueva.com.ar

TASA DE ERROR

BER = bits erróneos / bits transmitidos

LIMITANTES FÍSICOS DE LOS SISTEMAS DE COMUNICACIONES

ANCHO DE BANDA
RUIDO



Donde Δf es el ancho de banda del canal de comunicaciones

FIGURA 2.36 EFECTO DEL ANCHO DE BANDA EN UN CANAL DE COMUNICACIONES SOBRE UNA SEÑAL DIGITAL PERIÓDICA QUE SE TRANSMITE A TRAVÉS DEL MISMO.

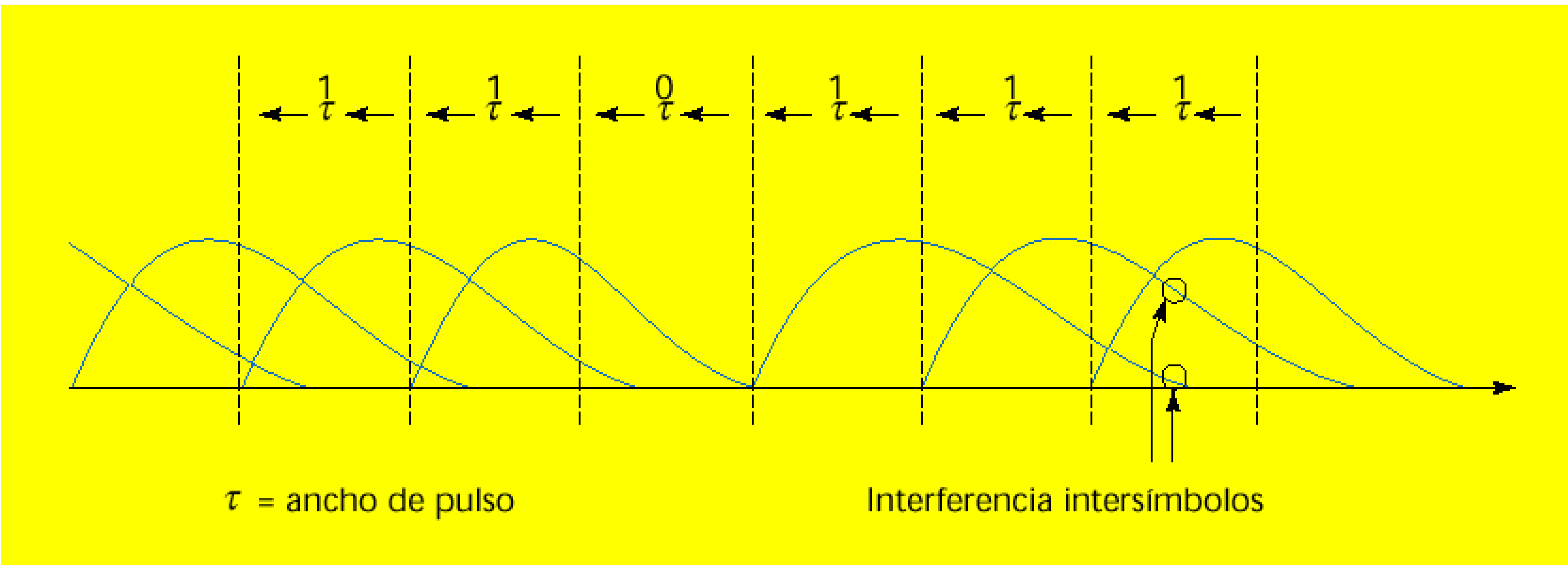


FIGURA 2.41 INTERFERENCIA INTERSÍMBOLOS.

DETECCIÓN Y CORRECCIÓN DE ERRORES

CONTROL DE ERRORES

ACCIONES

DETECCIÓN

CORRECCIÓN

DATOS ADICIONALES

COMPROMISOS

DETECCIÓN Y CORRECCIÓN DE ERRORES

REDUNDANCIA (R)

EFICIENCIA DE TRANSMISIÓN (E) = b c/info / b totales

> R   > PROTEC ERRORES   < E

COMPROMISO

TAMAÑO DE BLOQUE (Tb) EN TRANSMISIÓN SINCRÓNICA

> Tb   > E   PERO SI ERROR < E

COMPROMISO

DETECCIÓN Y CORRECCIÓN DE ERRORES

VELOCIDAD DE TRANSMISIÓN (Vtx)

> Vtx > P error > BER

COMPROMISO

Métodos de detección y corrección de errores

- Detección y corrección de errores
  - Detección
    - Control de paridad
      - Vertical VCR
      - Longitudinal LRC
      - Bidimensional o de doble paridad
      - Cíclica o entrelazada
    - Código polinomial
  - Corrección
    - Códigos autocorrectivos
      - Hamming
      - Hagelbarger
      - Bose Chaudhurí
    - Técnicas especiales de transmisión
      - Corrección de errores hacia adelante (FEC)
      - Requerimiento automático de repetición (ARQ)

DETECCIÓN

Paridad par e impar

ejemplos

Paridad par será 0, carácter resultante	0	01101101100
Paridad impar será 1, carácter resultante	1	01101101100

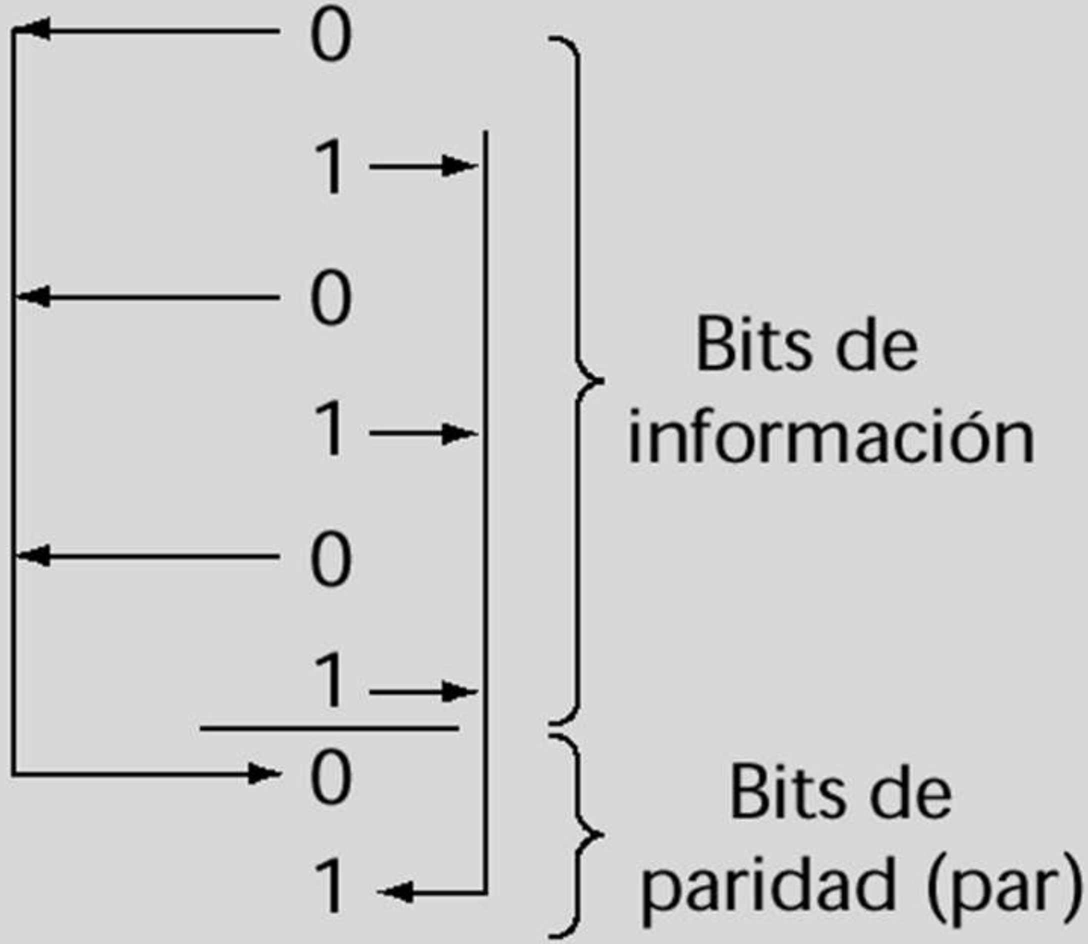
Ejemplo de control de paridad longitudinal

paridad par

	Dato 1	Dato 2	Dato 3	Dato 4	Dato 5	BCC
Bit nº 1	1	1	0	1	0	1
Bit nº 2	1	1	0	1	1	0
Bit nº 3	1	1	0	1	0	1
Bit nº 4	0	0	0	0	1	1
Bit nº 5	0	0	0	1	0	1
Bit nº 6	1	0	1	1	1	0
Bit nº 7	0	0	1	1	0	0
Bit de paridad vertical	0	1	0	0	1	0

Prueba de paridad cíclica

Carácter transmitido: 010101
Generación de la paridad cíclica



CHEQUEO DE REDUNDANCIA CÍCLICA

RESTO RESULTANTE DEL COCIENTE ENTRE UN POLINOMIO MENSAJE Y UN POLINOMIO GENERADOR

$$M(x).X^{r} / G(x) = C(x) \quad R(x)$$

$$T(x) = M(x).Xr + R(x)$$

soh	block of data with byte stuffing	eot	CRC

ALGUNOS POLINOMIOS GENERADORES NORMALIZADOS

CRC – 16, CRC – 12

EJERCICIO CRC

EMISOR

1) Polinomio Mensaje M = 110101 $(X^5 + X^4 + X^2 + X^0)$
Polinomio Generador G = 11001 $(X^4 + X^3 + 1)$

M tiene 6 bits de datos
G tiene 5 bits por lo que producirá un CRC de 4 bits; en consecuencia: k = 4.

2) Multiplicando el mensaje M por X^k da:
$X^k \times M = X^4(X^5 + X^4 + X^2 + X^0) = X^9 + X^8 + X^6 + X^4$
El equivalente binario de este producto tiene 10 bits y es igual a:
1101010000

3) Se divide por G el producto obtenido

```
Xk x M ——> 1101010000   |11001   <—— G
           11001         100101 <—— C  (Cociente)
           -----
             11100
             11001
             -----
               10100
               11001
               -----
                1101  <———————— R = Resto = CRC
```

4) El resto R se suma a $X^k \times M$ para dar la información a transmitir:
1101011101

RECEPTOR

5) La información recibida se divide por G dando un resto nulo:

```
1101011101   |11001
11001         100101
-----
  11111
  11001
  -----
    11001
    11001
    -----
    00000
```

CHECKSUM

Data Item In Binary	Checksum Value	Data Item In Binary	Checksum Value
00001	1	00011	3
00010	2	00000	0
00011	3	00001	1
00001	1	00011	3
totals	7		7

CORRECCIÓN

CORRECCIÓN HACIA ATRÁS

1RO DETECCIÓN (PARIDAD, CRC U OTROS)

2DO CORRECCIÓN MEDIANTE REPETICIÓN DE BLOQUE DE DATOS (POR EJEMPLO ARQ)

REQUERIMIENTO AUTOMÁTICO DE REPETICIÓN ARQ

- ENTRE DOS ESTACIONES
- VARIANTES → STOP AND WAIT
  → SLIDING WINDOWS

También se lo considera como un método de control de flujo

Confirmación +	ACK
Confirmación -	NAK

ARQ STOP AND WAIT (parar y esperar)

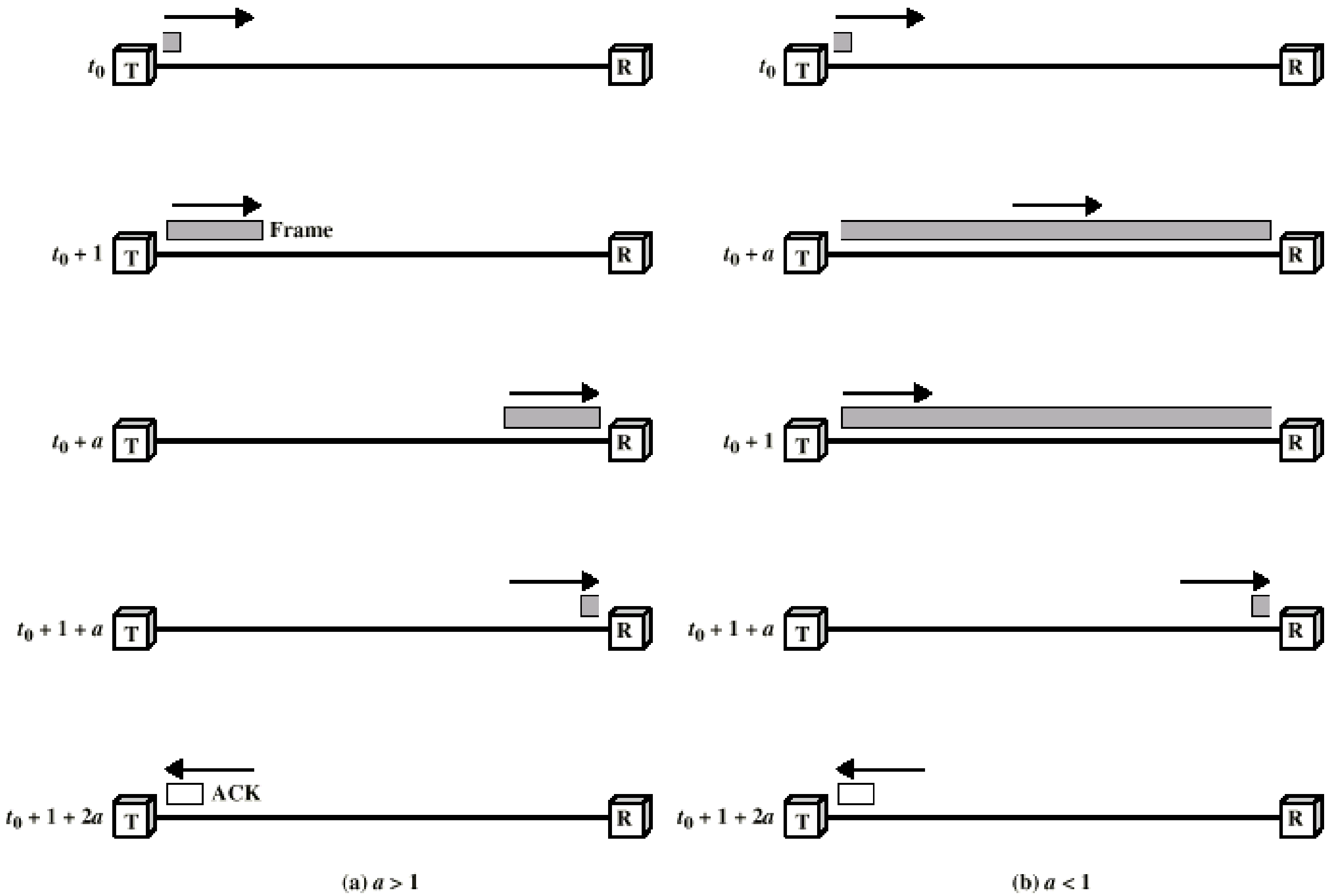
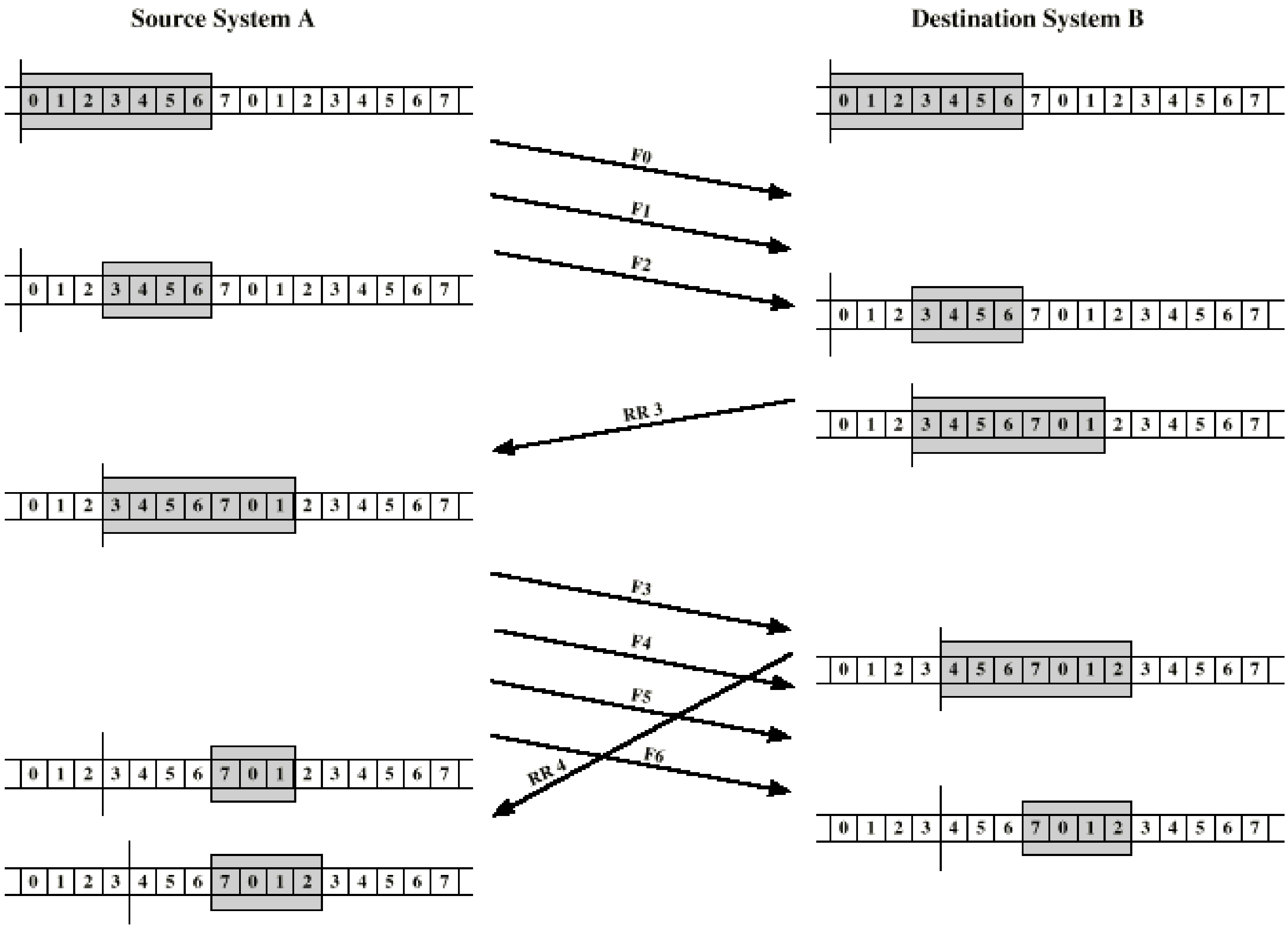


Figure 7.2 Stop-and-Wait Link Utilization (transmission time = 1; propagation time = a)

Relación velocidad de transmisión y tiempo de propagación, ineficiencia si vel altas y grandes distancias.

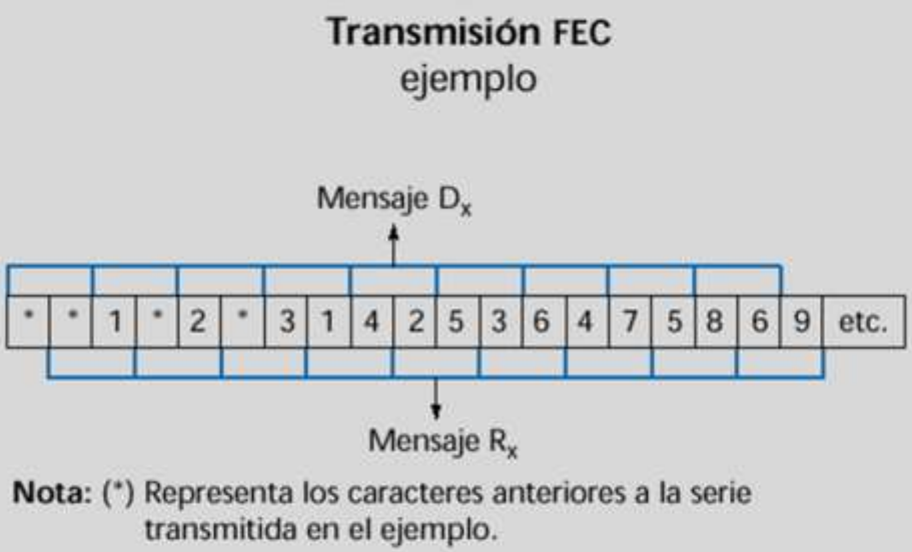
ARQ SLIDING WINDOWS (ventana deslizante)



Tamaño de ventana (fija o variable), eficiencia, necesidad de buffer, confirmación en full duplex (piggyback).

CORRECCIÓN HACIA ADELANTE FEC

- ENTRE DOS O MÁS ESTACIONES
- DOBLE ENVÍO DE MENSAJE EN TIEMPO DIFERIDO



Posibles operaciones del FEC

Caracteres		Operación del FEC
Serie D_x	Serie R_x	
Sin error	Sin error	Se ordena imprimir el carácter
Con error	Sin error	Se ordena imprimir el carácter de la serie R_x
Sin error	Con error	Se ordena imprimir el carácter de la serie D_x
Con error	Con error	Se ordena imprimir un carácter especial que indica error en la recepción

CÓDIGOS AUTOCORRECTORES

CÓDIGO HAMMING

d_H **Es el número de bits en los que difieren dos secuencias**

$d_{H\,min}$ **Es la menor distancia H en un código determinado**

Distancias de Hamming
tomadas para la secuencia
correspondiente al símbolo B

Conjunto	Representa	Secuencia binaria	Distancia de Hamming
S_1	B	0100001	–
S_2	C	1100001	1
S_3	D	0010001	2
S_4	E	1010001	3
S_5	U	1010101	4

CAPACIDAD DE DETECCIÓN O CORRECCIÓN HASTA …

E detec = $(d_{H\,min} - 1)$

E correc = $< (d_{H\,min} - 1) / 2$

Detección y corrección de errores
en función del valor de H

Distancia de Hamming	Errores	
	Detección	Corrección
1	no	no
2	uno	no
3	dos	uno
4	tres	uno

Código 1

000
111

$d_{H\,min}$ 3

D2 C1

Código 2

000
011
110
101

$d_{H\,min}$ 2

D1

Código 3

000
011
110
101
001
010
100
111

$d_{H\,min}$ 1

Ninguno

Formación del código Hamming
para un carácter de 4 bits

Bits de información			I_3		I_5	I_6	I_7
Bits de paridad	P_1	P_2		P_4			
Carácter resultante	P_1	P_2	I_3	P_4	I_5	I_6	I_7

FIGURA 6.48

$$d_{H\,min}\quad 3$$

Bits de información y paridad
relación entre ambos

Bits de paridad	Bits de información		
P_1	I_3	I_5	I_7
P_2	I_3	I_6	I_7
P_4	I_5	I_6	I_7

Ejemplo de aplicaciones del código Hamming

	I_3	I_5	I_6	I_7
Carácter original	0	0	1	1

Cálculo de bits de paridad (PAR)	Bits de información asociados			Bits de paridad (PAR)
P_1	0	0	1	1
P_2	0	1	1	0
P_2	0	1	1	0

	P_1	P_2	I_3	P_4	I_5	I_6	I_7
Código de Hamming formado	1	0	0	0	0	1	1