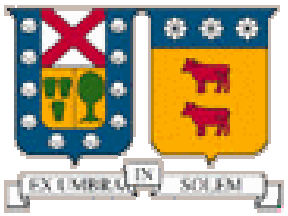
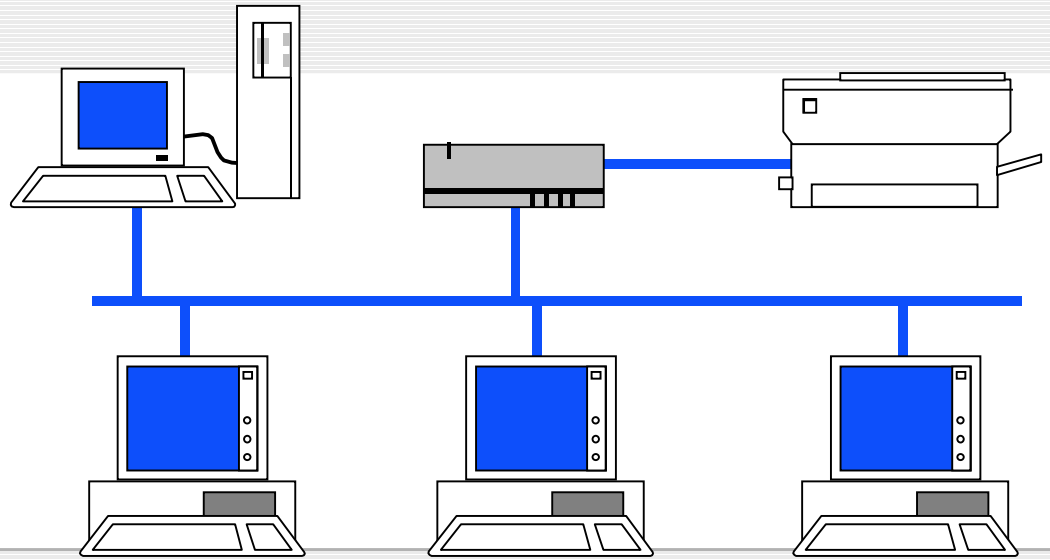


Redes de Computadores

Capa de Enlace de Datos

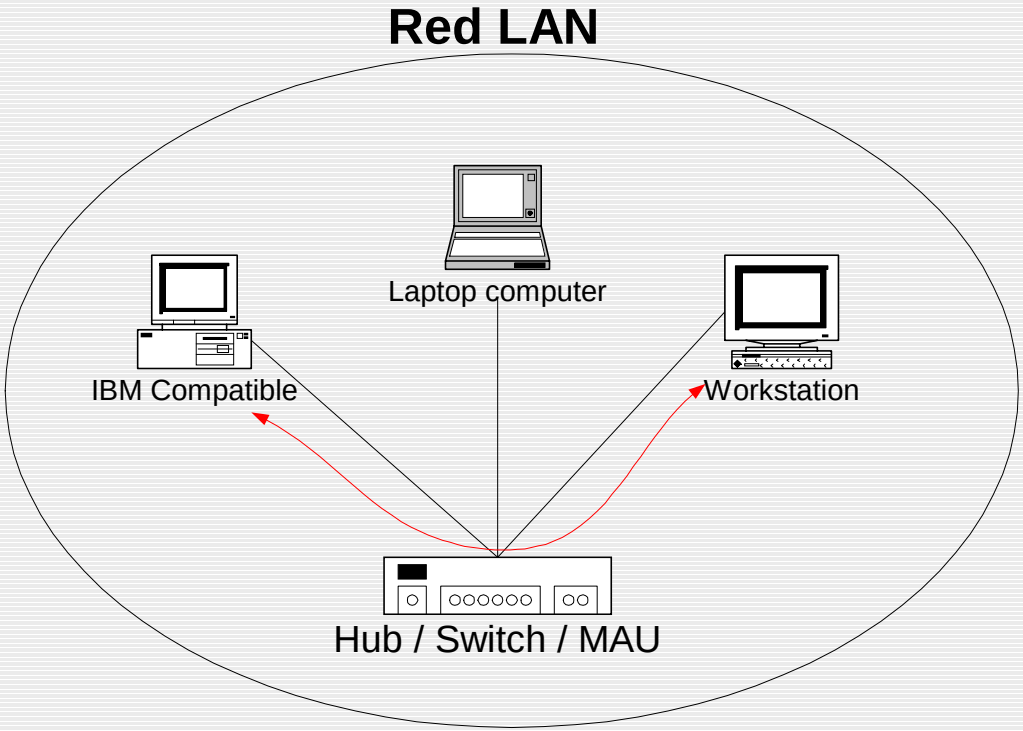


Indice

- Objetivo y Consideraciones
- Funciones
- Enmarcado (Entramado)
- Control de Errores
- Control de Flujo
- Gestión de Enlace
- Errores
  - Detección
  - Corrección

Objetivo

- La capa de enlace debe suministrar, a la capa de red (nivel 3), una comunicación fiable y eficiente entre dos máquinas adyacentes (en la misma red **LAN**).



Consideraciones

¿Porqué es necesaria esta capa?

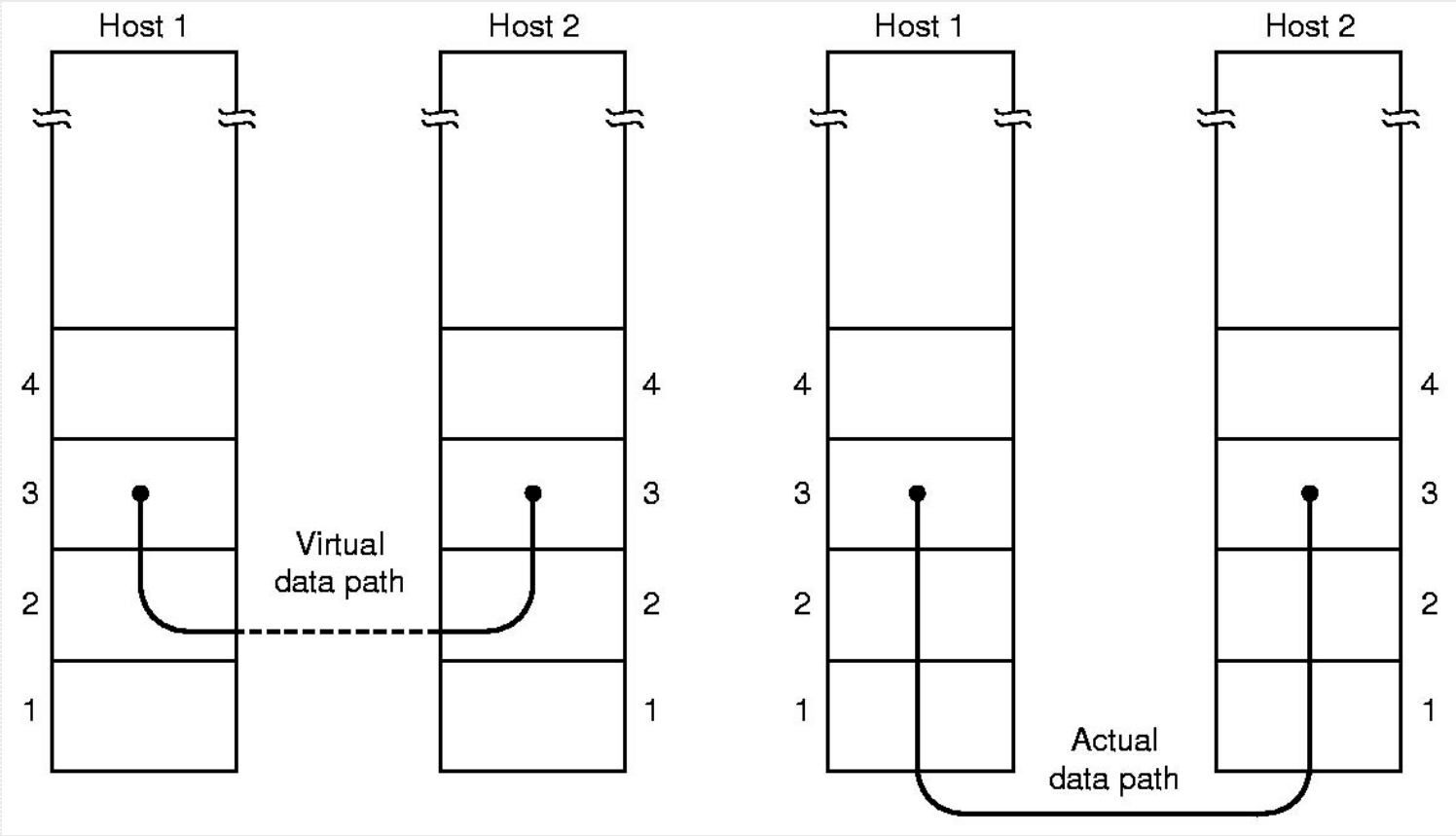
- El canal físico es susceptible a errores
- Las velocidades de transmisión y recepción pueden ser distintas
- Existe retardo en el canal.

Funciones

- Interfaz de servicio para la capa de red
- Agrupar los bits a transmitir en forma de tramas (enmarcar)
- Ocuparse de los errores de transmisión
- Regular el flujo de tramas
- Administrar la capa de enlace (Gestión) (Subcapa LLC, Logical Link Control)
- Traducir tramas de redes heterogéneas.
- Subcapa MAC: Medium Access Control

Servicio proporcionado a la capa de red

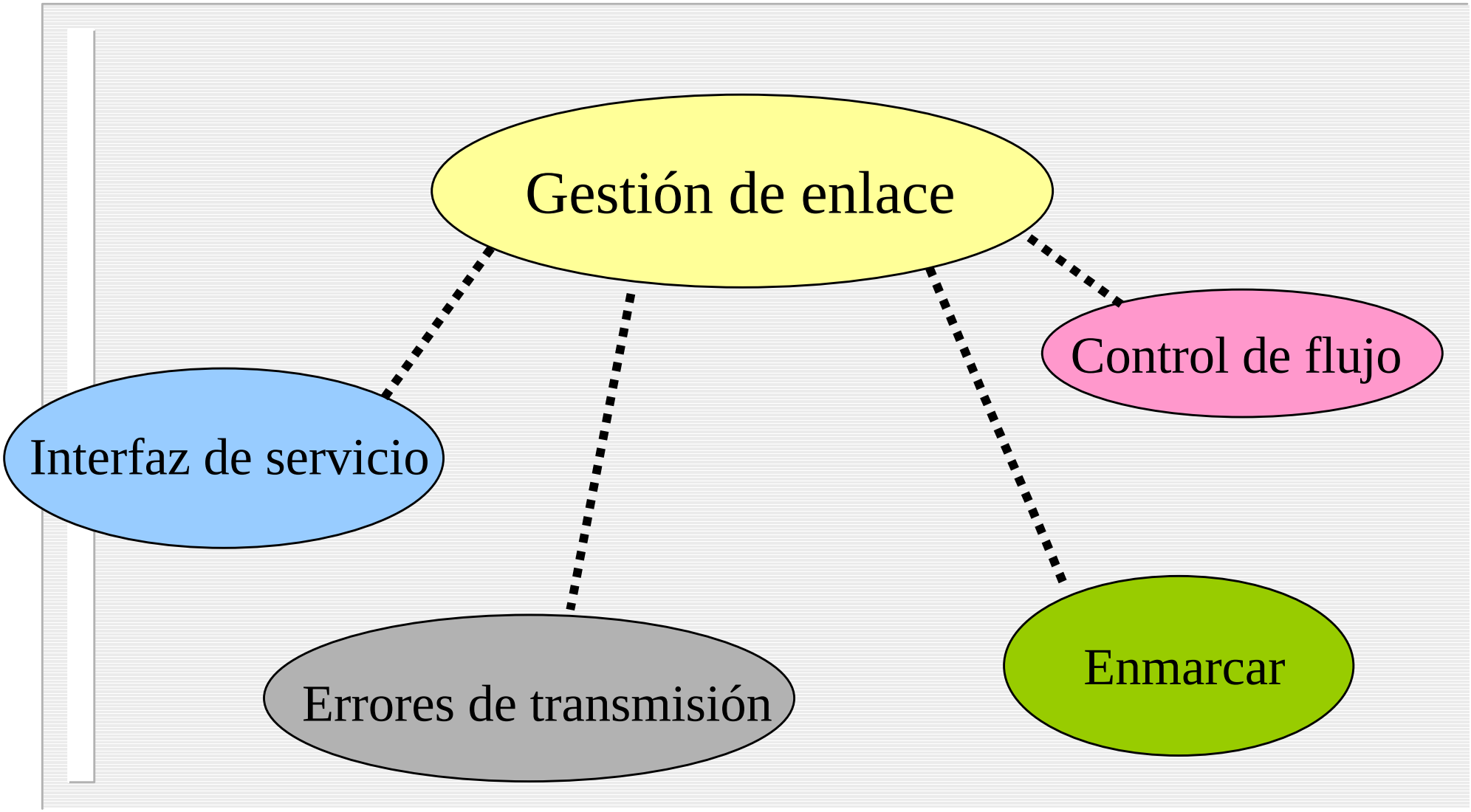
- Transferir datos de la capa de red de la máquina origen, a la capa de red de la máquina destino.



Servicios de conexión proporcionados a la capa de red

- Servicios sin conexión y sin reconocimiento
  - TX independiente de tramas hacia la máquina destino, sin esperar reconocimiento. (Ej: Tráfico en Tiempo Real)
- Servicios sin conexión y con reconocimiento
  - Cada trama TX es reconocida por el receptor (TX Inalámbrica)
- Servicio orientado a la conexión
  - Se establece una conexión antes de la transferencia
  - Las tramas son enumeradas
  - Se garantiza la recepción única de cada trama
  - Se garantiza la recepción de las tramas en el orden correcto.

Funciones


Gestión de enlace
Control de flujo
Interfaz de servicio
Errores de transmisión
Enmarcar

Gestión de Enlace

- Las conexiones deben establecerse y liberarse
- La numeración de paquetes debe iniciarse y reiniciarse en caso de errores
- En caso de Control de Acceso al Medio centralizado, la estación primaria deberá sondear a las secundarias por si es que tienen datos que enviar.

Control de Flujo

"Velocidad" de transmisión > "Velocidad" de recepción

El receptor no es capaz de recibir todas las tramas que le llegan ⇒ se requiere control de flujo.

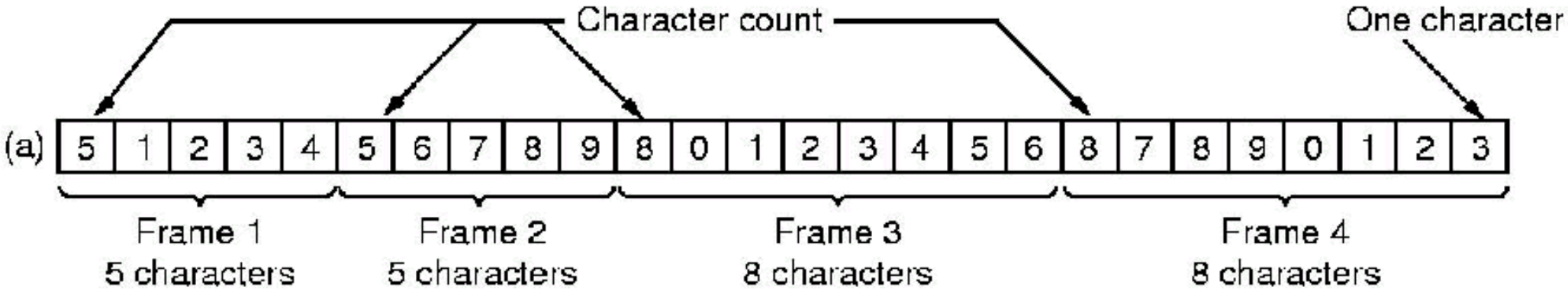
Enmarcado

- Métodos de división del flujo de bits en tramas
  - Cuenta de caracteres
  - Caracteres de inicio y fin, con inserción de carácter (Char Stuffing)
  - Cadena de bits de inicio y fin, con inserción de bit (Bit Stuffing)
  - Violaciones de código en la capa física.

Enmarcado

1.- Cuenta de Caracteres.

- Un caracter indica el número de caracteres de datos del marco y por ende sabe cuando termina.

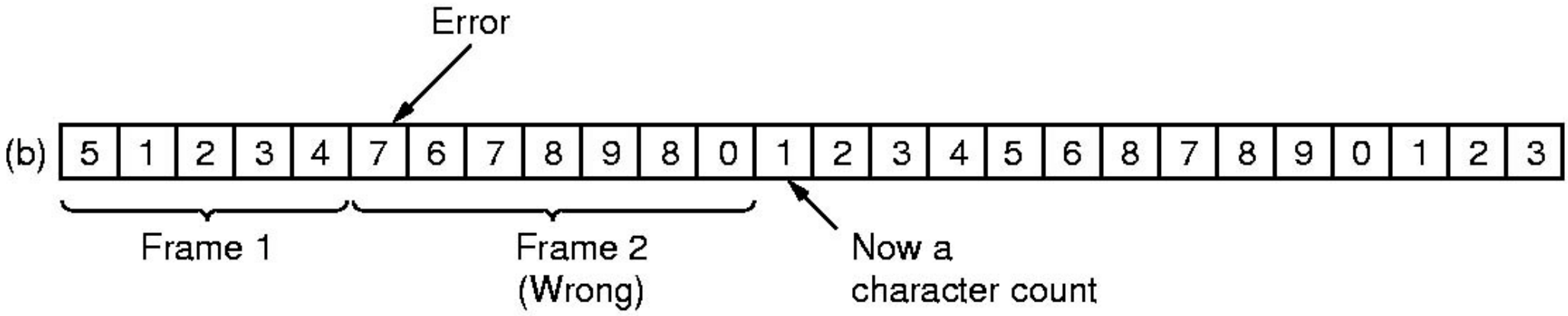


Enmarcado

Cuenta de Caracteres.

- **PROBLEMA:**

la cuenta puede alterarse por un error de transmisión justo en el caracter inicial.

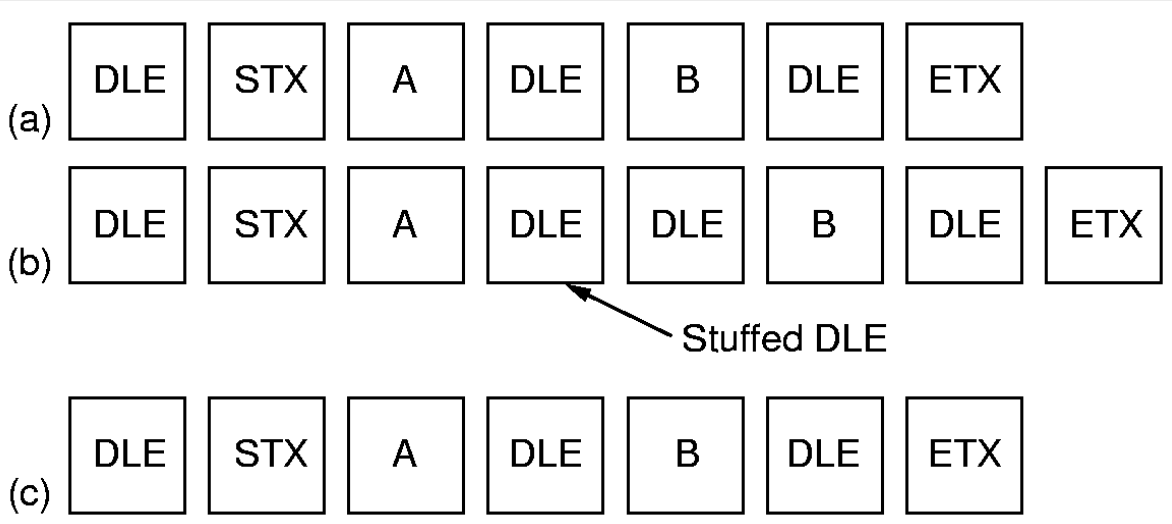


Enmarcado

2.- Inserción de carácter (Char Stuffing)

- DLE: Data Link Escape, STX: Start of Text, ETX: End of Text
- DLE-STX = inicio de trama, DLE-ETX: fin de trama

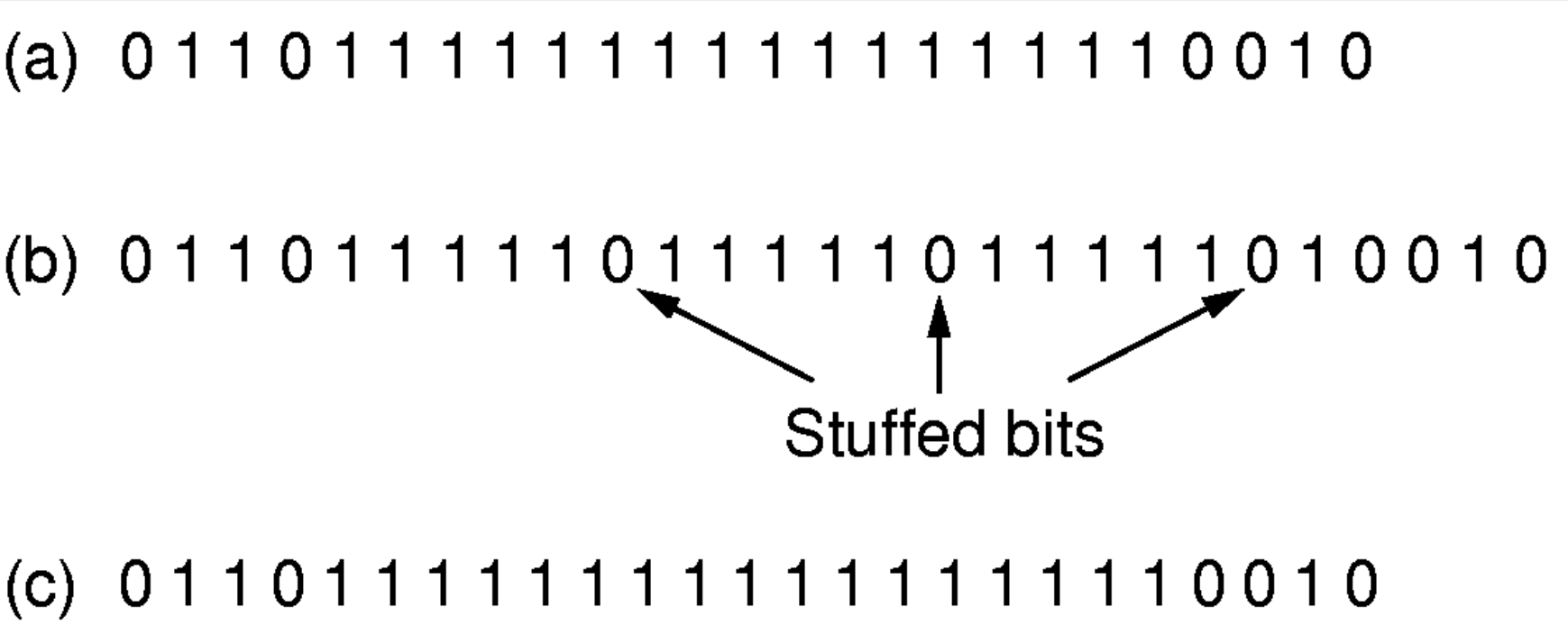
- TX debe insertar DLE extra en caso de existir un DLE en la data
- RX hace proceso inverso al ver un DLE en data



Enmarcado

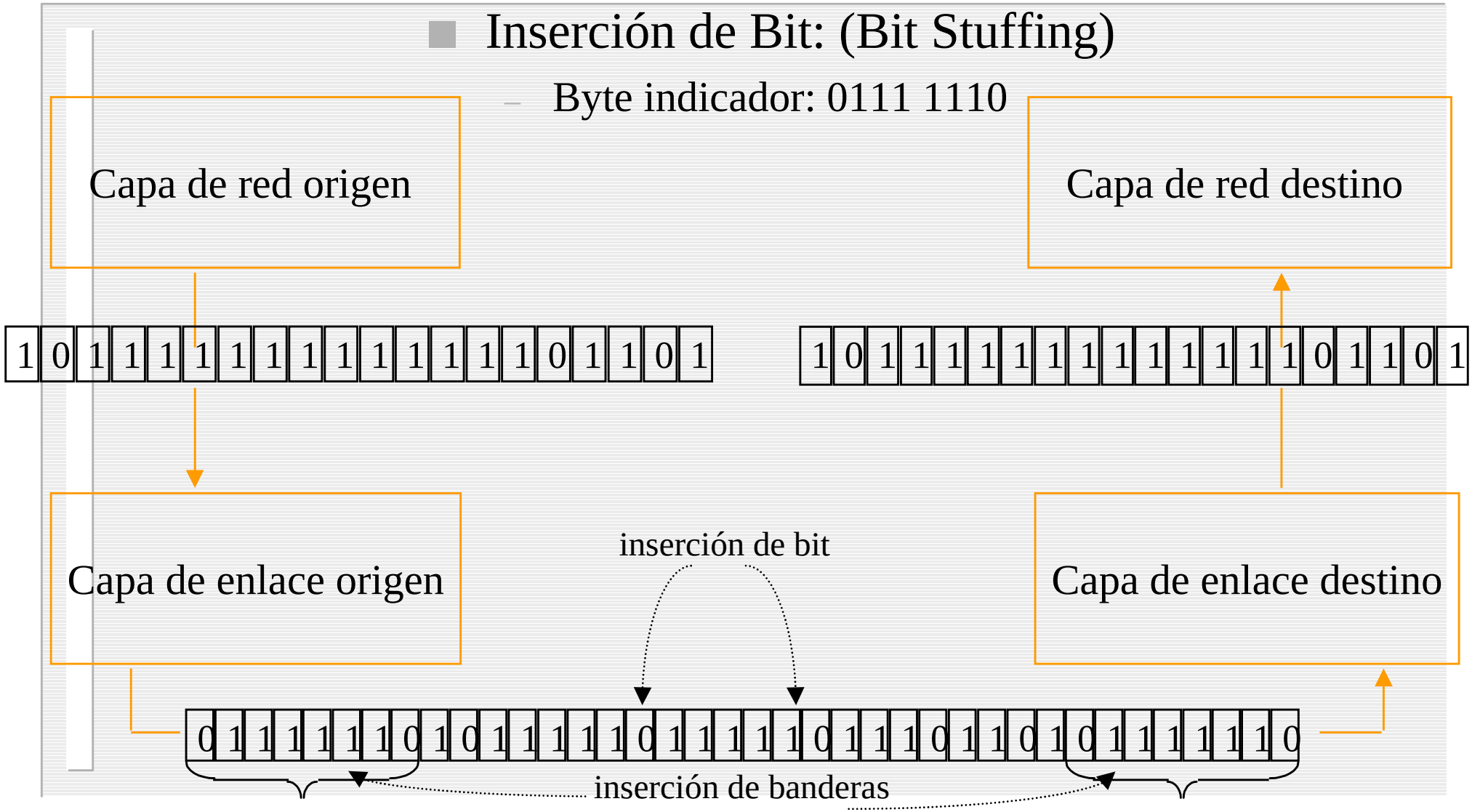
3.- Inserción de Bit: (Bit Stuffing)

- Byte indicador: 0111 1110



Enmarcado

- Inserción de Bit: (Bit Stuffing)
  - Byte indicador: 0111 1110



Enmarcado

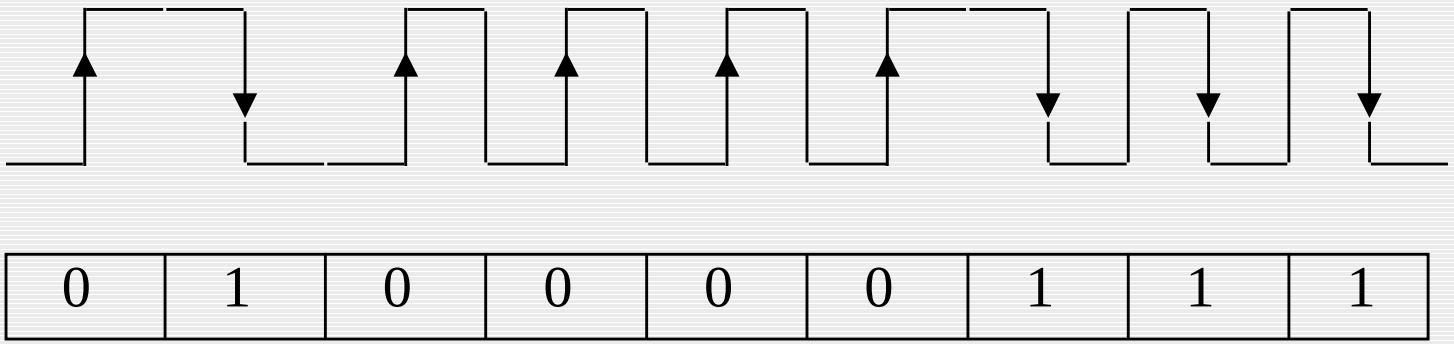
- 4.- Violación de Códigos

Sólo se aplica cuando la codificación, efectuada por la capa física, contiene alguna redundancia o característica especial.

Enmarcado

Violación de Códigos

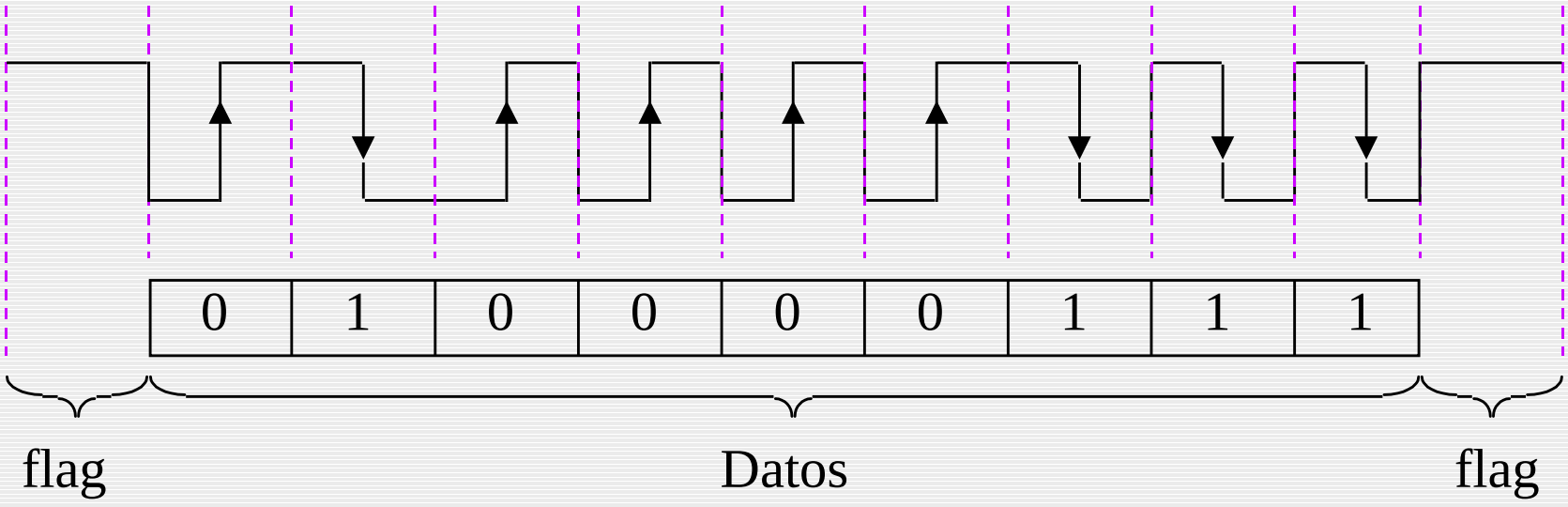
- Por ejemplo, el código Manchester codifica cada 1 como un par alto-bajo, y cada 0 como par bajo-alto
- Es decir que las combinaciones alto-alto y bajo-bajo no se utilizan.
- Este hecho es aprovechado para marcar el inicio y fin de cada trama.



Enmarcado

Violación de Códigos

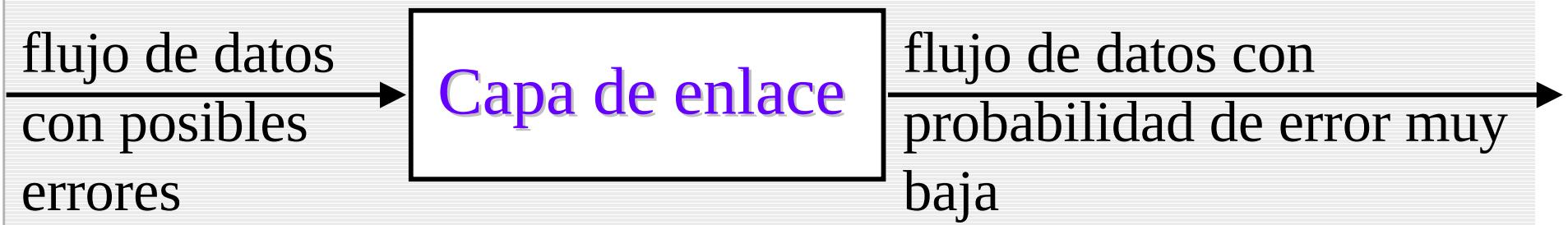
Por ejemplo, si utilizamos la combinación alto-alto, como flag de inicio y fin de trama, tenemos el siguiente caso:



Control de Errores

Función Principal

- Asegurar que todas las tramas sean entregadas sin error a la capa de red del extremo receptor y en el orden correcto.



Errores

- Los errores de los datos pueden deberse a:
  - número de bits recibidos > = < número de bits que se transmitieron
  - Los bits recibidos pueden estar errados

Control de Errores

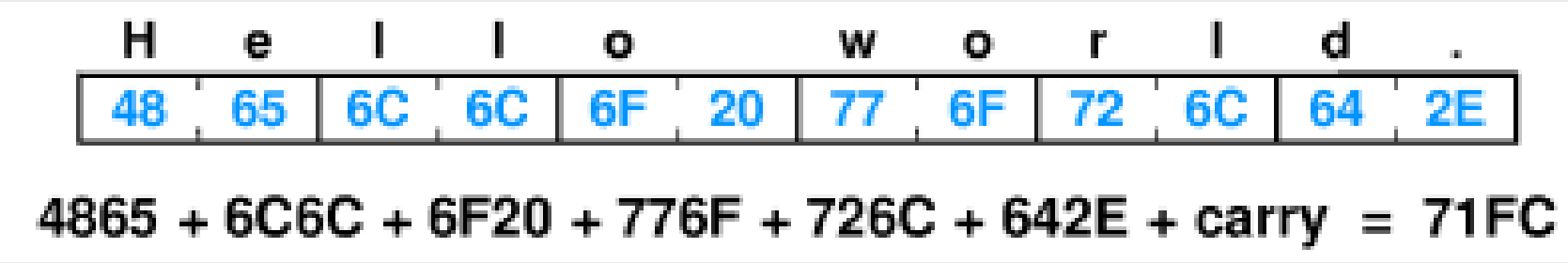
- El RX debe poder determinar si la trama recibida está correcta o posee errores de transmisión
- Códigos de Detección
  - incluye información adicional en la trama
  - sólo indica si ha ocurrido un error
- Códigos de Corrección
  - incluye mayor información adicional en la trama
  - indica dónde se encuentra el(los) error(es)

Códigos de Detección ***bit de Paridad***

- Muy usado en TX asincrónicas orientadas a caracter
- El TX agrega un bit de paridad por cada grupo de bits (7 u 8) para que la suma binaria de todos los bits (8 u 9) resulte:
  - cero: bit de paridad par
  - uno: bit de paridad impar
- El RX realiza el mismo cálculo para ver si hubo error
- Sistema sólo detecta un # de errores impares

Códigos de Detección ***Checksum***

- El frame es tratado como secuencia de caracteres
- se suman los char y se envía esta suma



- RX comprueba el checksum

Códigos de Detección ***Checksum***

- Ventaja:
  - operación muy sencilla
  - se envían menos bits que en “bit de paridad”
- Desventaja:
  - errores múltiples no son detectados.

Data Item In Binary	Checksum Value	Data Item In Binary	Checksum Value
0001	1	0011	3
0010	2	0000	0
0011	3	0001	1
0001	1	0011	3
totals	7		7

Códigos de Detección ***CRC***

- Se ocupa el código **CRC** (Código de Redundancia Cíclica)
  - Representación polinómica de una cadena de bits
  - 110 001 equivale a x^5+x^4+1 (grado=5)
  - Se emplea un polinomio Generador G(x) en el TX y RX
  - La idea es dividir una cadena de bits (data a enviar = M(x)) por G(x).

Códigos de Detección

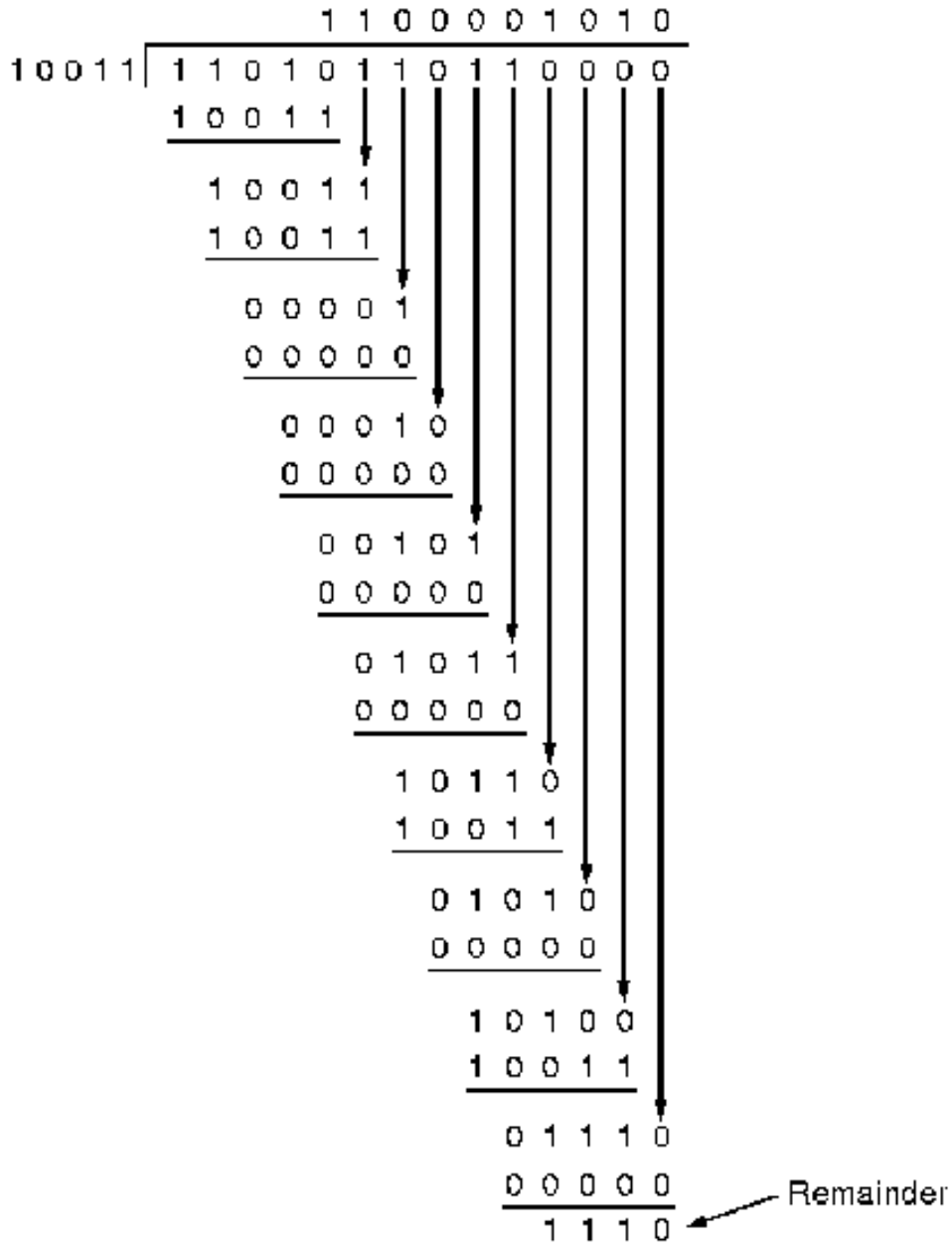
- Procedimiento en el transmisor
  - Sea “r”el grado de G(x)
  - Agregue “r” bits 0 al final de M(x): $x^r M(x)$
  - Divida $x^r M(x)$ por G(x)
  - Reste el residuo a $x^r M(x)$ y obtendrá el polinomio T(x) que deberá transmitirse.

Códigos de Detección

Frame : 1 1 0 1 0 1 1 0 1 1

Generator: 1 0 0 1 1

Message after appending 4 zero bits: 1 1 0 1 0 1 1 0 1 1 0 0 0 0



Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 0

Códigos de Detección

- Procedimiento en el receptor
  - El receptor recibe [T(x)+E(x)], E(x) es el error.
  - Se divide por G(x)
  - Dado que el residuo de T(x)/G(x) = 0, se está calculando el residuo de E(x)/G(x)
  - Si el residuo es 0, no hubo errores
  - Si $E(x) = x^i$, se detectarán los errores de 1 bit.

Códigos de Detección

- G(x) estándares
  - CRC-12: $x^{12}+x^{11}+x^3+x^2+x+1$ (char=6 bits)
  - CRC-16: $x^{16}+x^{15}+x^2+1$ (char=8 bits)
  - CRC-CCITT: $x^{16}+x^{12}+x^5+1$ (char=8 bits)
  - CRC-32: (Ethernet, FDDI)

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

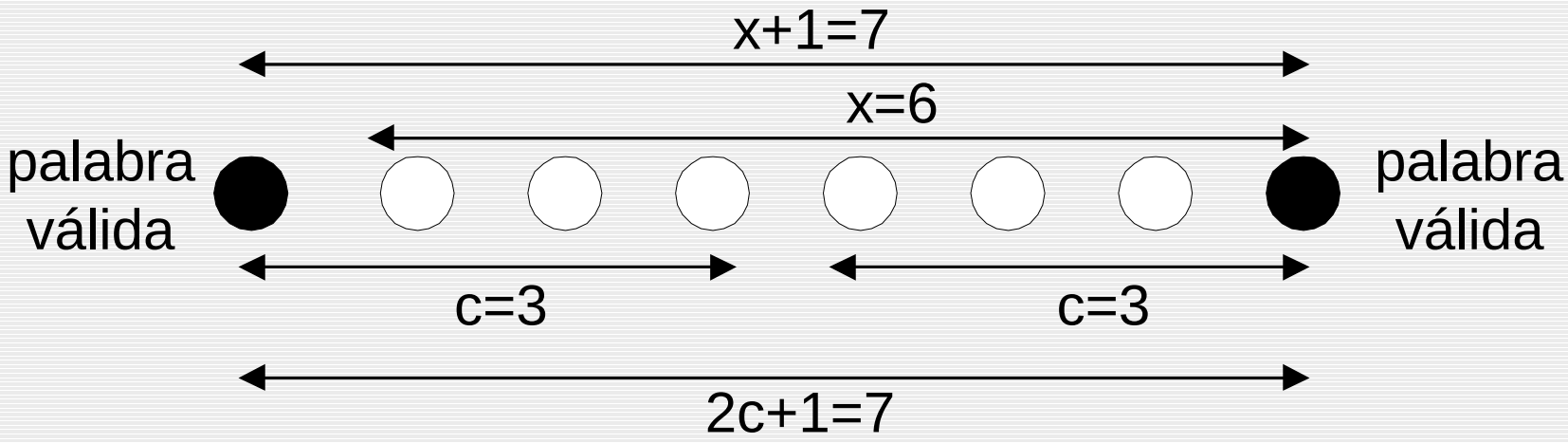
  - El CRC-16 o CRC-CCITT
    - detecta todos los errores simples y dobles
    - todos los errores con número impar de bits
    - todos los errores en ráfaga de longitud 16 o menos
    - 99,997% de ráfagas de errores de 17 bits
    - 99,998% de ráfagas de 18 bits o mayores.

Código de Corrección ***Hamming***

- Distancia Hamming entre 2 palabras (secuencia de bits)
  - se aplica la operación OR EXCLUSIVO
  - EJ: 1000 1001 y 1011 0001 tienen distancia=3
- Existen Palabras válidas y Palabras no-válidas
- Distancia del código (palabras válidas)
  - distancia **mínima** entre todas las palabras válidas.

Códigos de Corrección

- Detección de “x” errores
  - requiere de un código de distancia mínima “x+1”
- Corregir “c” errores
  - requiere de un código de distancia mínima “2c+1”



Códigos de Corrección

- Ejemplo:
  - palabras válidas:
    - 00000 00000
    - 00000 11111
    - 11111 00000
    - 11111 11111
  - Es código de distancia=5, corrige errores dobles
  - Si se recibe un “00000 00111” se supondrá que debió ser un “00000 11111”
  - Si hubo error triple (se envió “00000 00000”), entonces no se corrige adecuadamente. (supuso mal).