REDES DE COMPUTADORAS
EC5751

USB

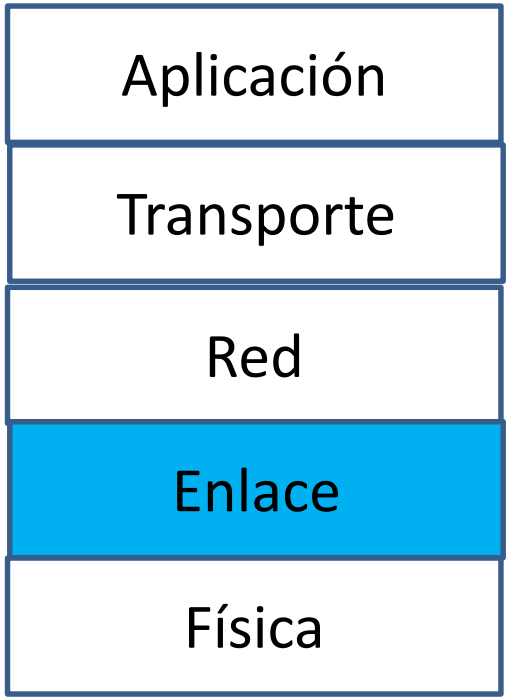
Capa de Enlace I

Prof. Mariano Arias

LA CAPA DE ENLACE

Contenido

-Capa de enlace
-Transmisión Asíncrona y Síncrona
-Reloj
-Tramas (Frames)
-Detección de errores y corrección
-Problema de asignación de canal
-Protocolos de acceso múltiple
-Redes Ethernet
-Redes Inalámbricas
-Otros ejemplos de redes



LA CAPA DE ENLACE

- Es la capa que permite controlar y gestionar el intercambio de información de un equipo a otro en el mismo medio físico.

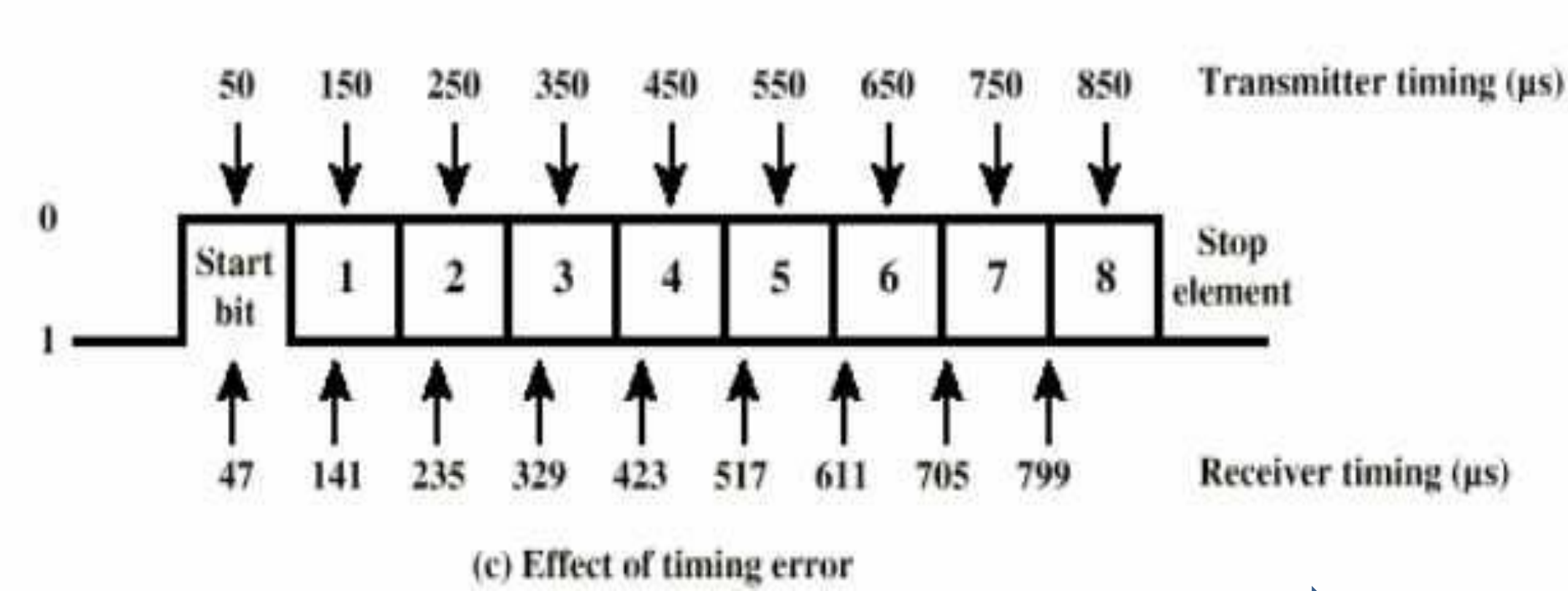
Objetivos:

- Sincronización de la trama.
- Gestión o Control de errores.
- Control del flujo.
- Direccionamiento.
- Datos y control sobre el enlace.
- Gestión del enlace

LA CAPA DE ENLACE

Transmisión Síncrona y Asíncrona

Los problemas de temporización requieren de mecanismos para sincronizar al transmisor y al receptor



LA CAPA DE ENLACE

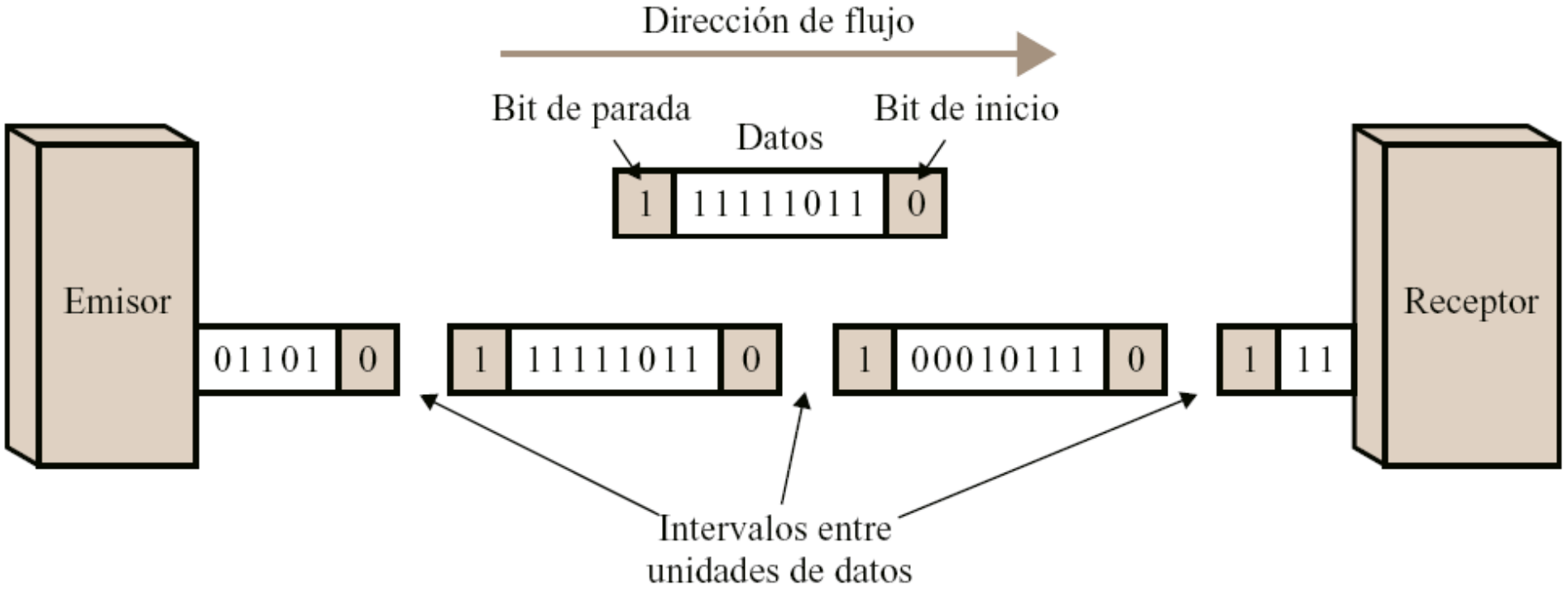
Transmisión Asíncrona

- Elimina el problema de sincronización
- Se evita enviar largas cadenas de bits.
- Se envía un carácter a la vez (5 a 8 bits).
- Se necesita mantener la sincronización dentro de cada carácter
- Se requiere resincronizar al inicio de cada carácter recibido.
- La duración de cada bit la deciden el transmisor y el receptor
- Cada carácter es envuelto por 1 bit de inicio y un bit de parada (alto overhead ~20%)
- Usada cuando los datos a transmitir son generados en forma aleatoria o esporádica. p.e. teclado, sensores.

LA CAPA DE ENLACE

Transmisión Asíncrona

- En estado pasivo, el receptor busca una transición de 1 a 0
- Luego muestrea los próximos 8 intervalos (la longitud del carácter).
- Espera el fin de carácter (típicamente 1)
- Busca el siguiente cambio de 1 a 0.



LA CAPA DE ENLACE

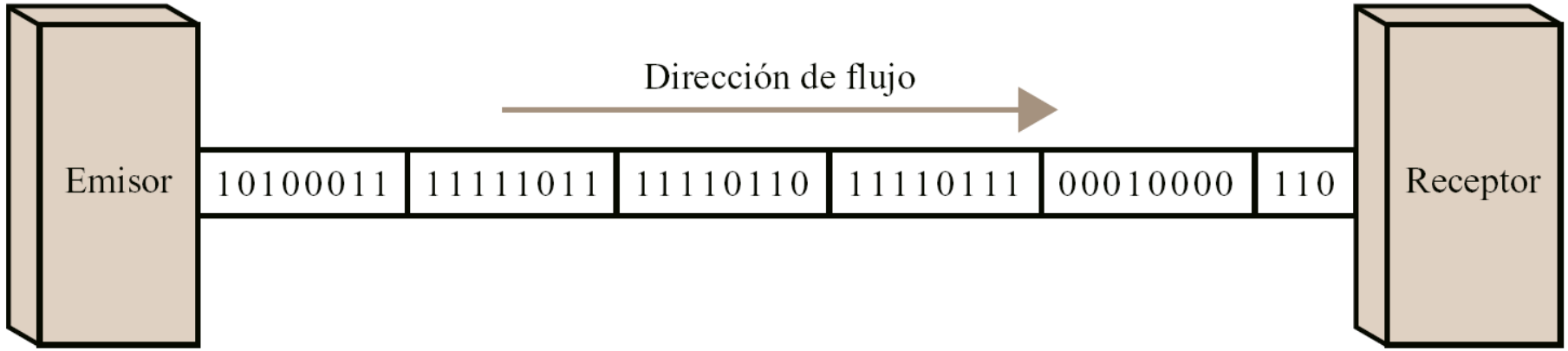
Transmisión Síncrona

- Los datos a transmitir son generados en forma continua y con requerimientos de altas velocidades.
- Se necesita indicar el inicio y el fin de cada bloque de transmisión Por ejemplo: 11111111 inicio y 11111110 para final.
- Más eficiente (menor overhead) que la transmisión asíncrona para cadenas de mas de 64 bits
- La línea está poco tiempo ociosa.
- Los relojes deben sincronizarse

LA CAPA DE ENLACE

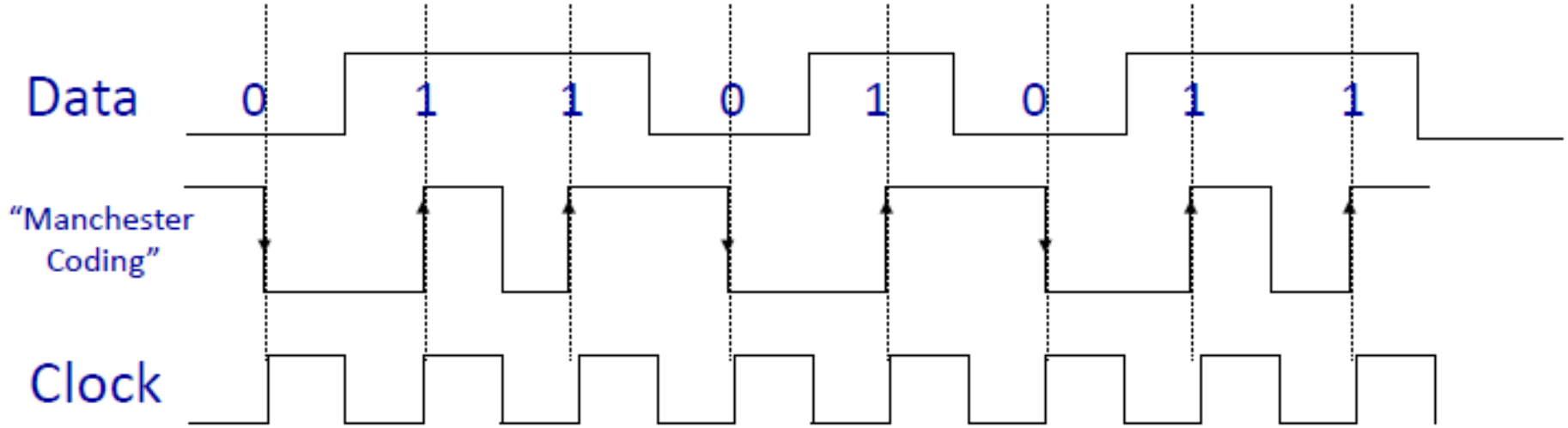
Transmisión Síncrona

1) Pueden usar una línea separada para la señal de reloj
   - Bueno en distancias cortas
2) La señal de reloj esta inmersa en los datos
   - PE: Codificación manchester
   - Frecuencia de la portadora (analógica)



LA CAPA DE ENLACE

Transmisión Síncrona: Reloj



Código Manchester (Ethernet 10 Mbps)

Ventajas: Mantiene balance DC

Cada transición es un bit.

Desventaja: Se requiere el doble del ancho de banda

LA CAPA DE ENLACE

Transmisión Síncrona: Reloj

4-bit data	5-bit code
0000	11110
0001	01001
0010	10100
...	...

Código 4b/5b: Muy usado actualmente (Giga Ethermet), garantiza al menos una transición por cada 4 bits.

Ventajas:

Eficiente en ancho de banda (solo 25% overhead)

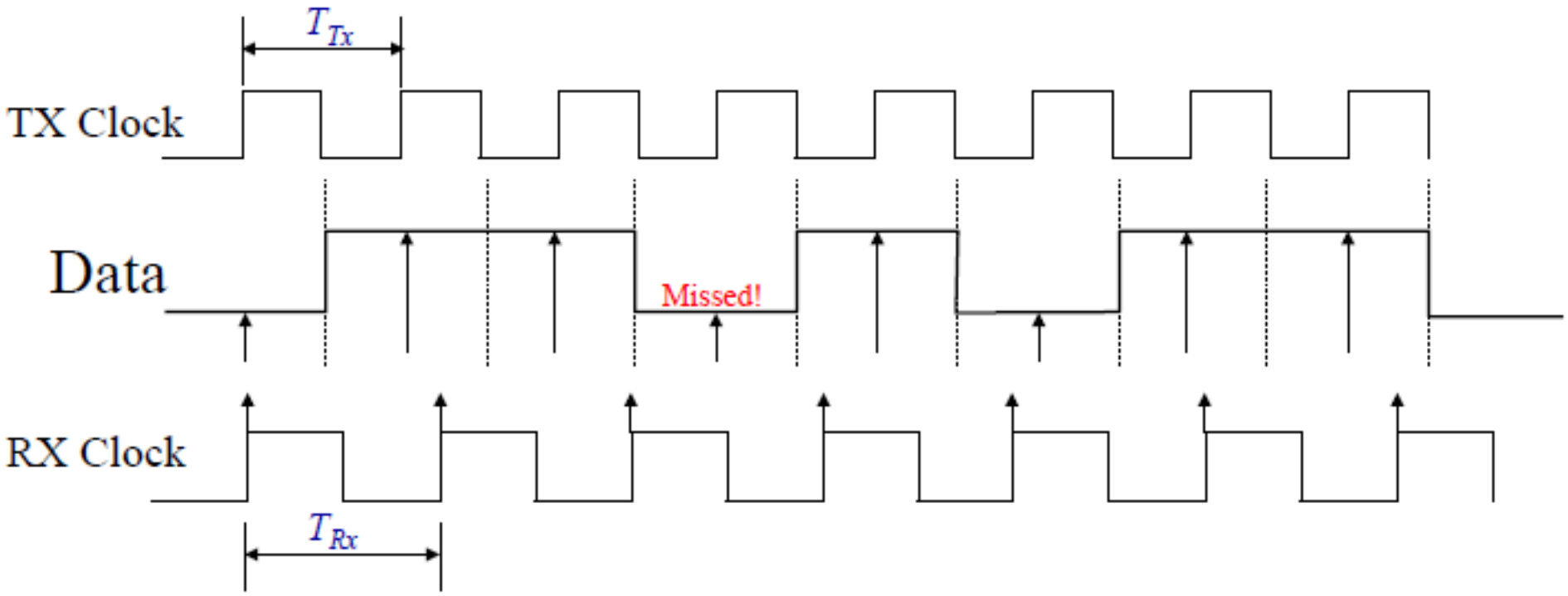
Códigos extra se usan para control.

Desventaja:

Se requiere mayor circuitería para recuperar el reloj

LA CAPA DE ENLACE

Transmisión Síncrona: Reloj

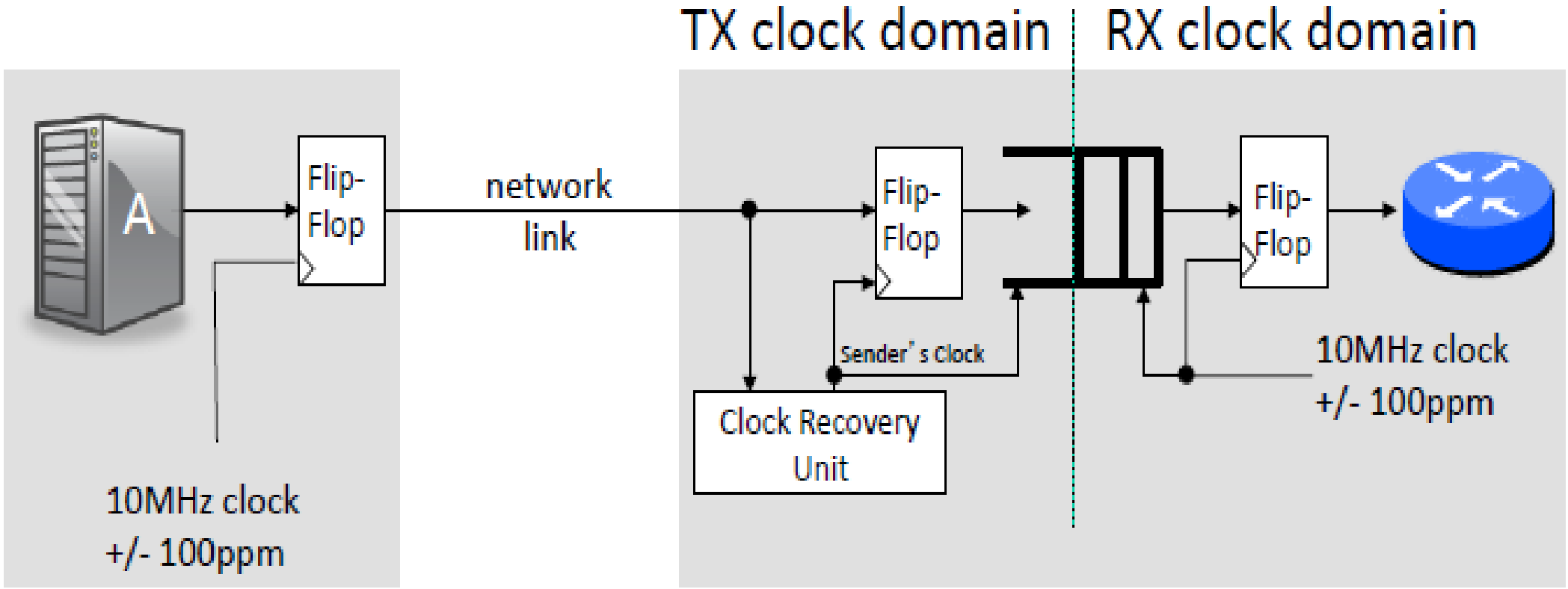


- *Si el reloj Rx es p ppm mas lento que el reloj Tx : TRx = Tx *(1+10^-6 p).*
- *Después de 0.5/(*10^-6p*) bit times, el reloj Rx fallara un bit.*

LA CAPA DE ENLACE

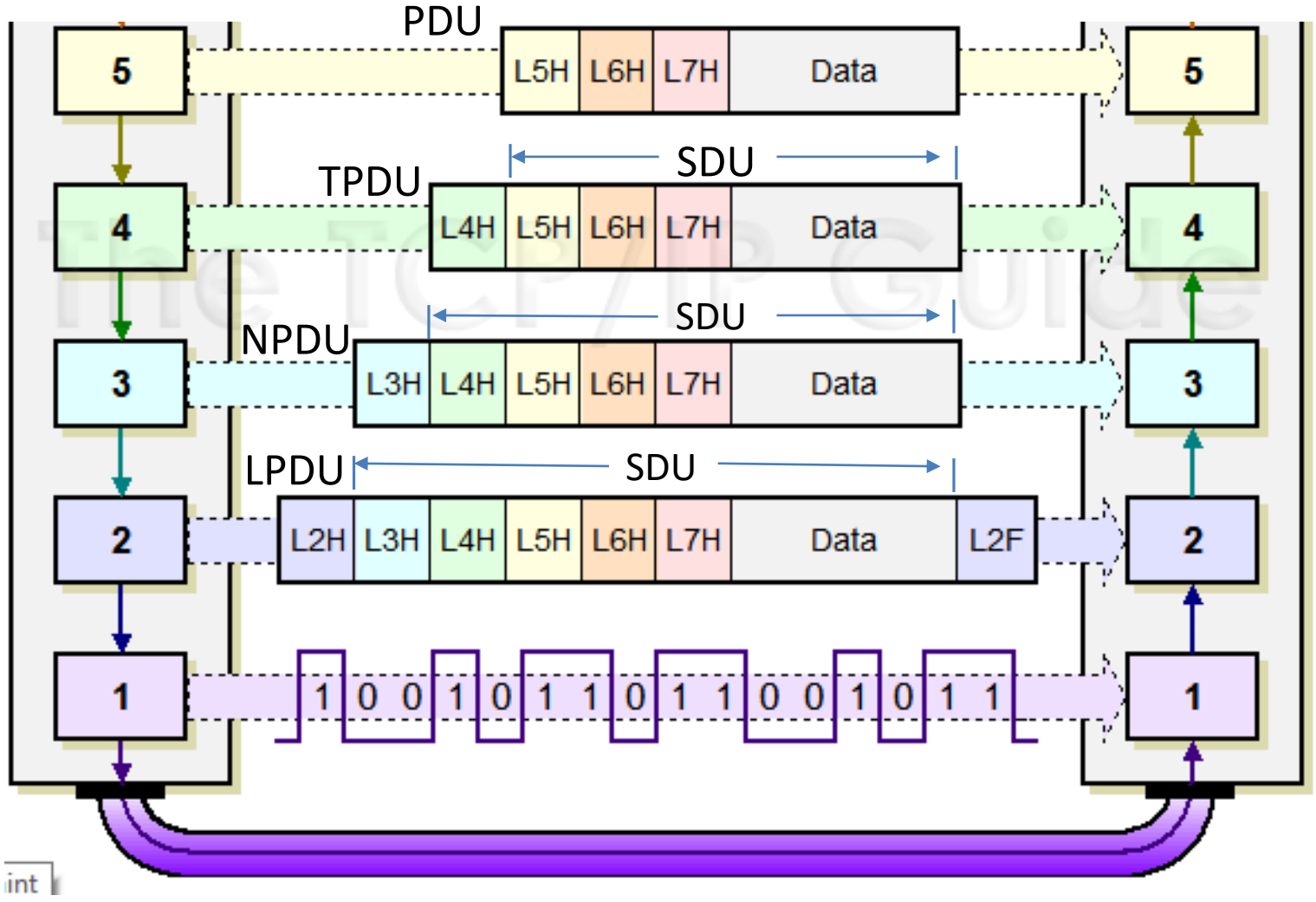
Transmisión Síncrona: Como recupero el Reloj?

Elasticity Buffer (FIFO)



LA CAPA DE ENLACE

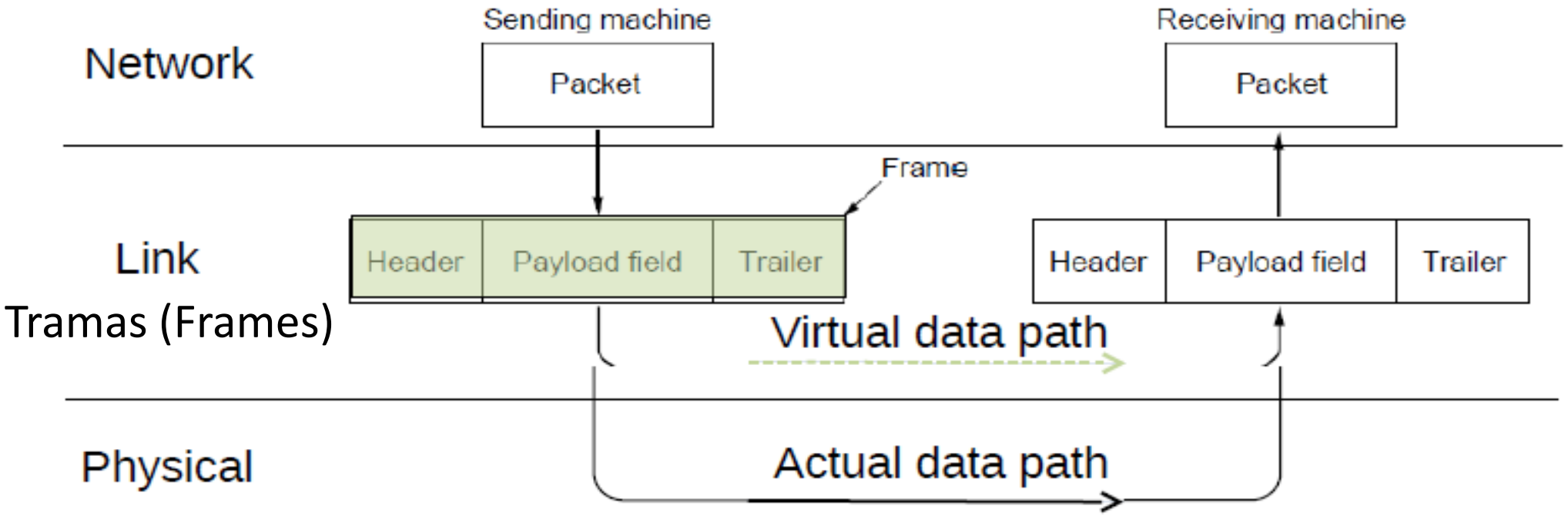
Conceptos: Protocol Data Unit (PDU) y Service Data Unit (SDU)



LA CAPA DE ENLACE

Tramas (Frames)

La capa de enlace acepta paquetes de la capa de red y los encapsula en “tramas” (Frame), que envía o recibe de la capa física. Este proceso se llama “Entramado” (Framing)

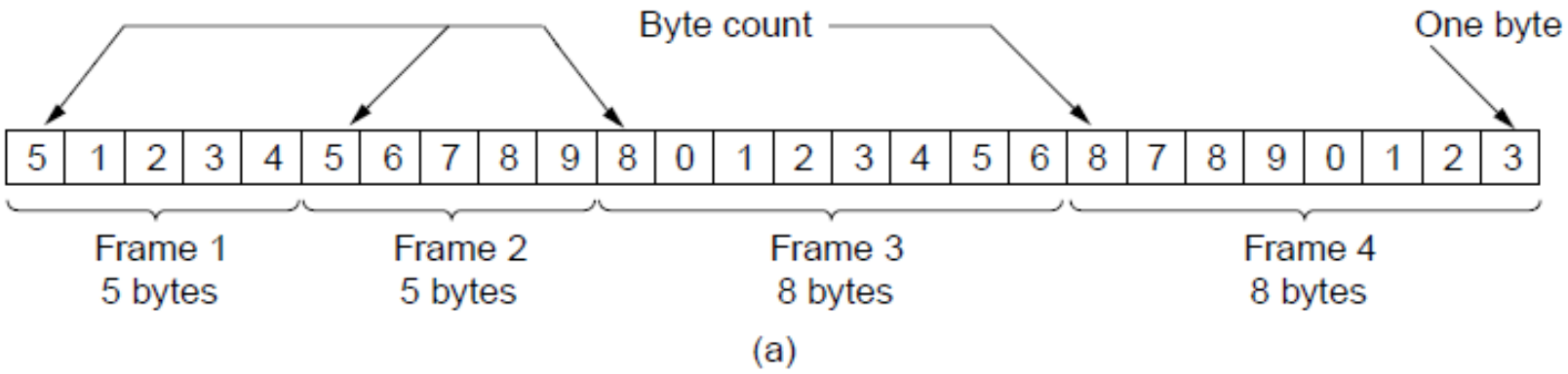


La TRAMA es el PDU de la capa de enlace (LPDU)

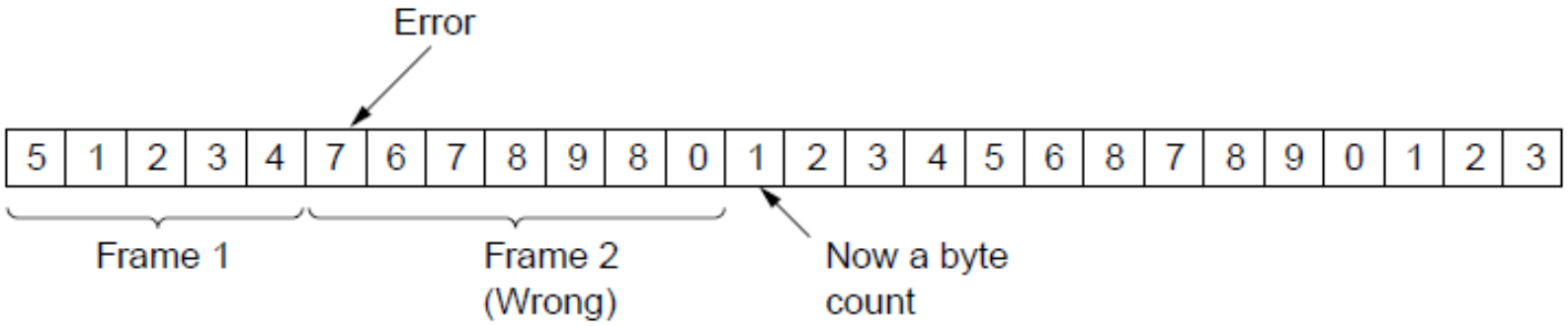
LA CAPA DE ENLACE

Entramado Conteo de Bytes

- La trama comienza con un numero que indica el numero de Bytes a enviar
- Simple, pero difícil de resincronizar si ocurre de un error



(a)

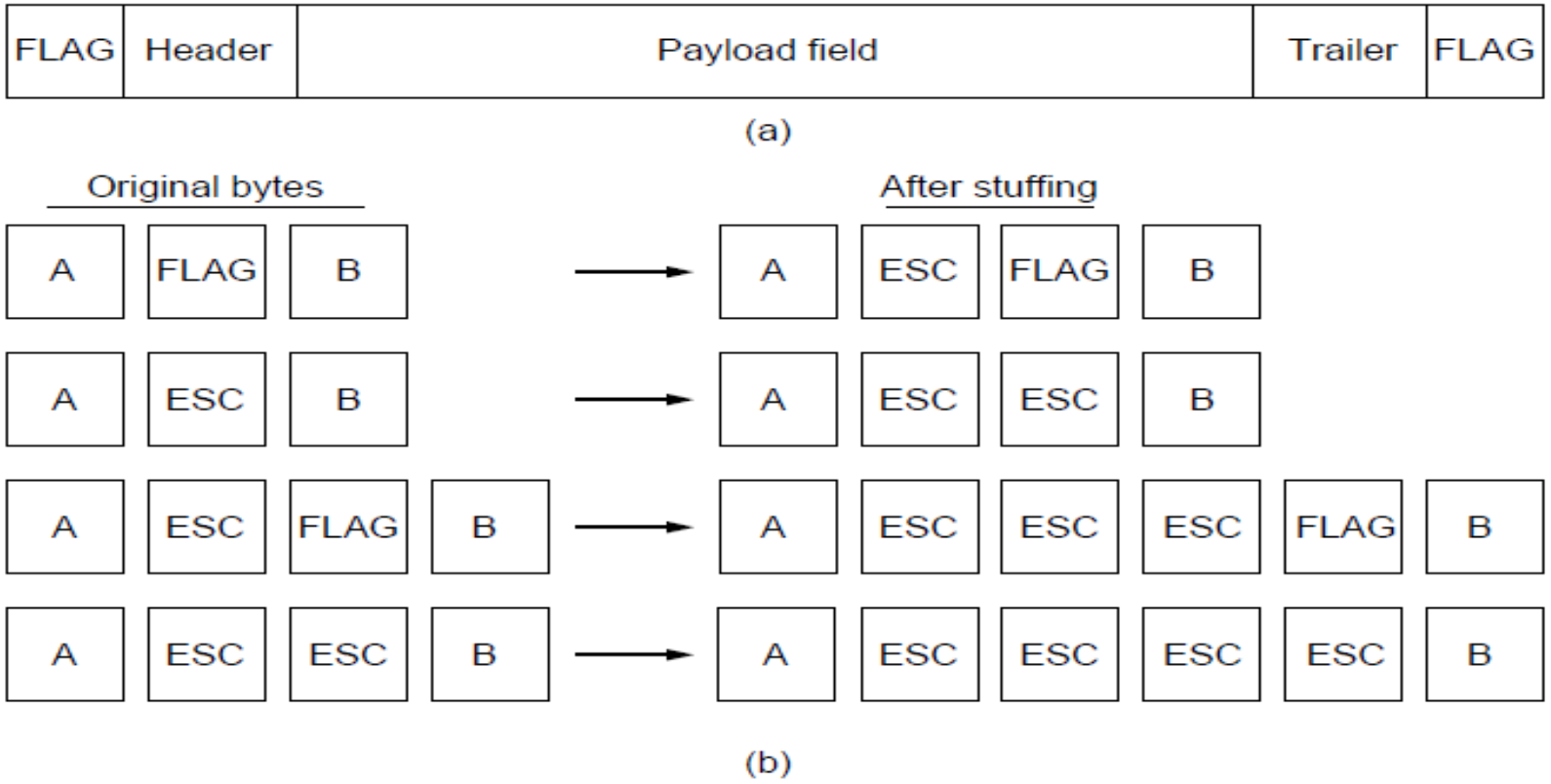


(b)

LA CAPA DE ENLACE

Entramado Relleno de Bytes

- Los bytes bandera (FLAG) especiales limitan las tramas; la ocurrencia de las banderas en los datos debe ser rellenada (secuencia de escape - ESC)
- Mas larga, pero fácil de resincronizar después de un error



LA CAPA DE ENLACE

Entramado por Relleno de Bits (bits stuffing)

Es hecho de la siguiente manera:

- La trama bandera (Flag) tiene seis 1´s consecutivos:

  01111110

- Siempre se coloca al inicio y al final
- Si en la data hay cinco 1´s seguidos, se añade un 0
- Al recibir un 0 después de cinco 1´s es borrado

Data Original

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

Bits incluidos

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

Stuffed bits

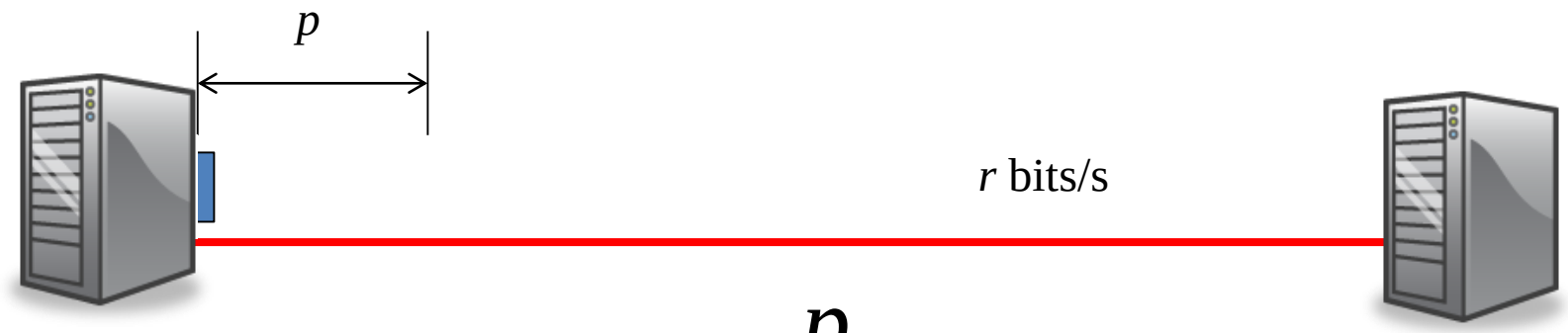
Data Recuperada

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

LA CAPA DE ENLACE

Retardo de Paquetizacion (Packetization Delay)

Tiempo que toma transmitir desde el primero al ultimo bit en un canal.



$$t_p = \frac{p}{r}$$

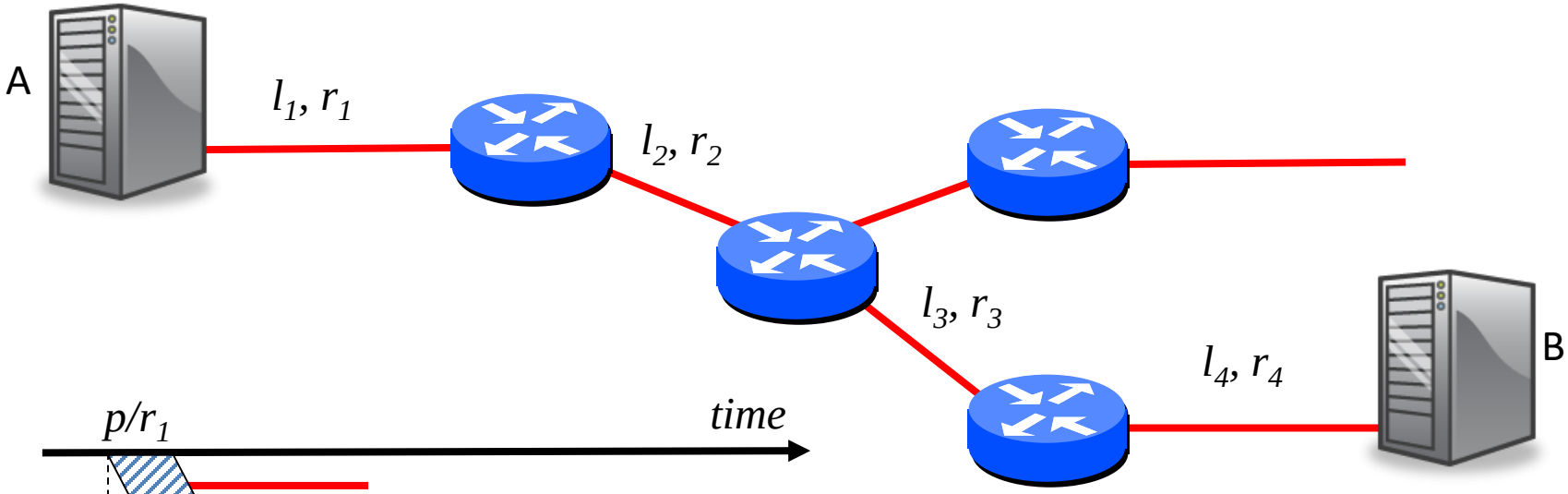
Ejemplos:

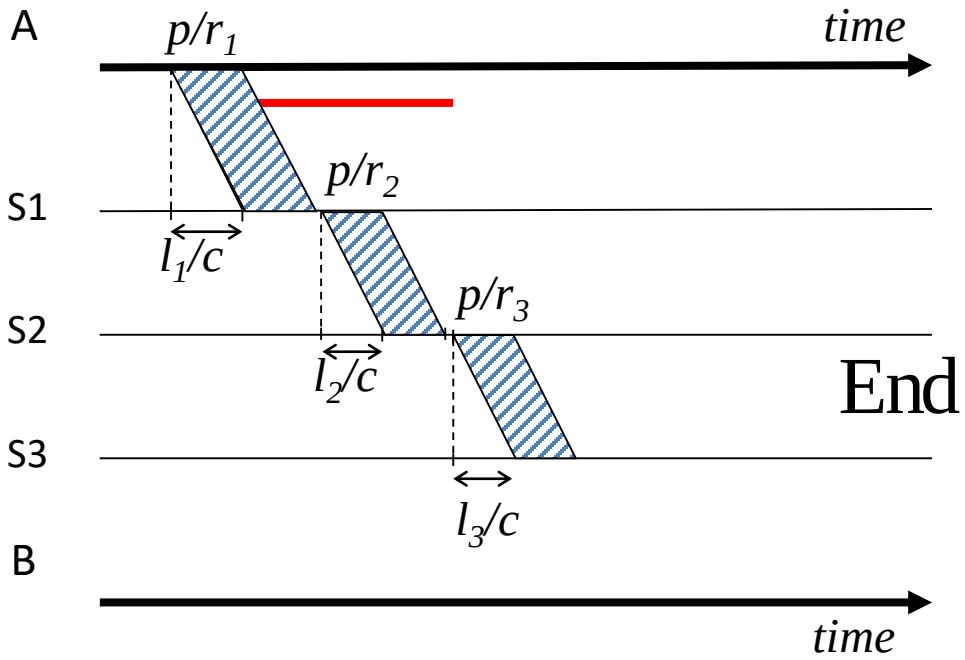
1) Paquete 64bytes tarda 5.12μs ser transmitido en un canal de 100Mb/s
2) Paquete 1 kbit tarda 1,024s ser trasmitido en un canal de 1kb/s.

LA CAPA DE ENLACE

Retardo Extremo-Extremo (End to end Delay)

Tiempo que toma atravesar toda la red.



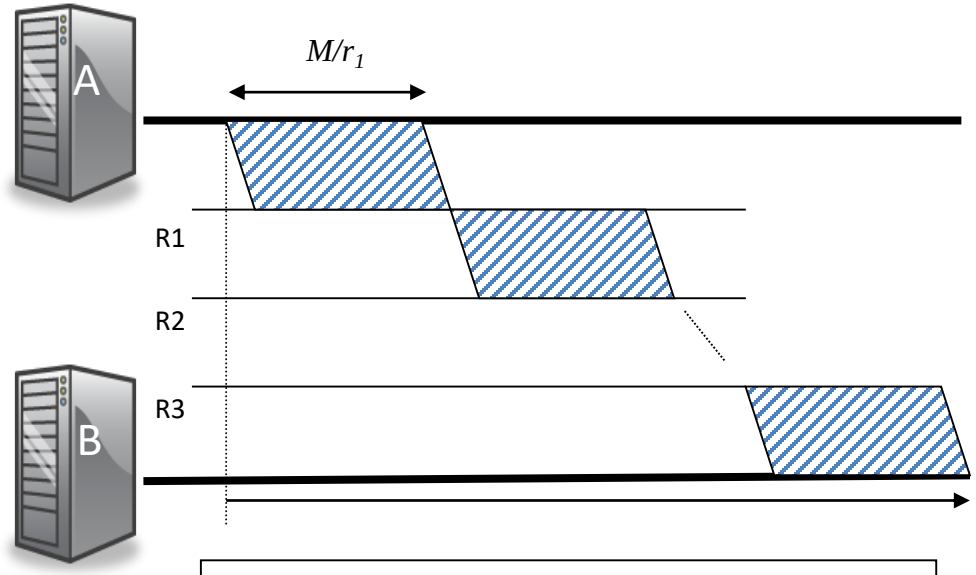


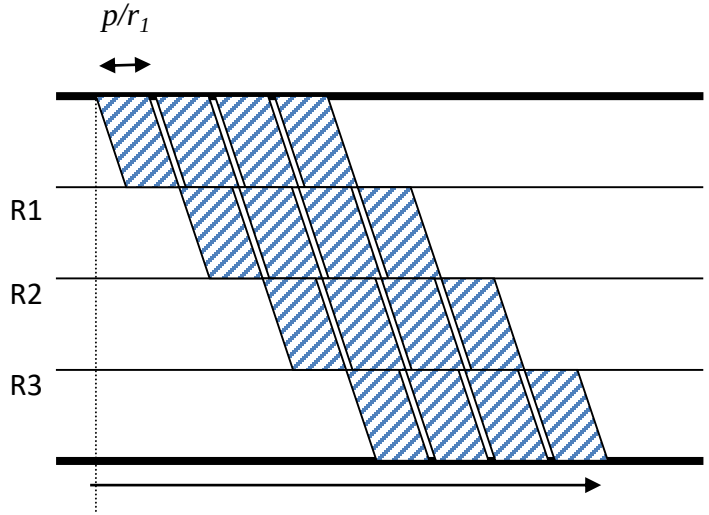
$$\text{End-to-end delay, } t = \sum_i \left(\frac{p}{r_i} + \frac{l_i}{c} \right)$$

LA CAPA DE ENLACE

Retardo Extremo-Extremo (End to end Delay)

Que sucede si dividimos el mensaje en varios paquetes?





$$\text{End-to-end delay, } t = \sum_i \left(\frac{M}{r_i} + \frac{l_i}{c} \right)$$

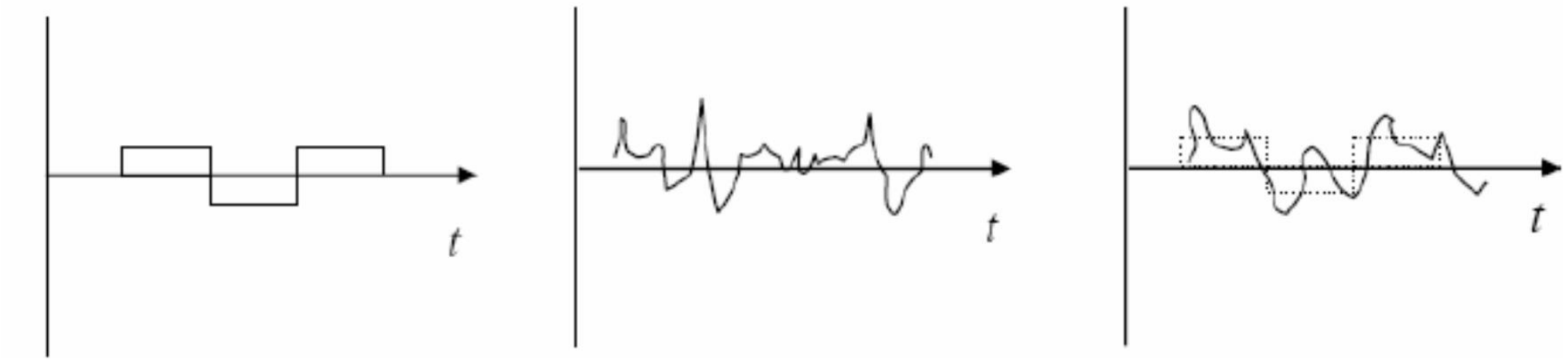
$$\text{End-to-end delay, } t = \sum_i \left(\frac{p}{r_i} + \frac{l_i}{c} \right) + \left(\frac{M}{p} - 1 \right) \frac{p}{r_{\min}}$$

Un sencillo calculo nos permitirá descubrir que el tiempo de envió se reduce, ya que al haber varios enlaces habrá transmisión paralela

LA CAPA DE ENLACE

Errores

- Un error ocurre cuando un bit es alterado entre su transmisión y su recepción
- Errores de un solo bit
    - Los bits adyacentes no se vieron afectados
    - Ruido Blanco
- Ráfagas de error
    - Longitud B
    - Secuencias contiguas de B bits en las cuales cualquier número de bits presenta errores



LA CAPA DE ENLACE

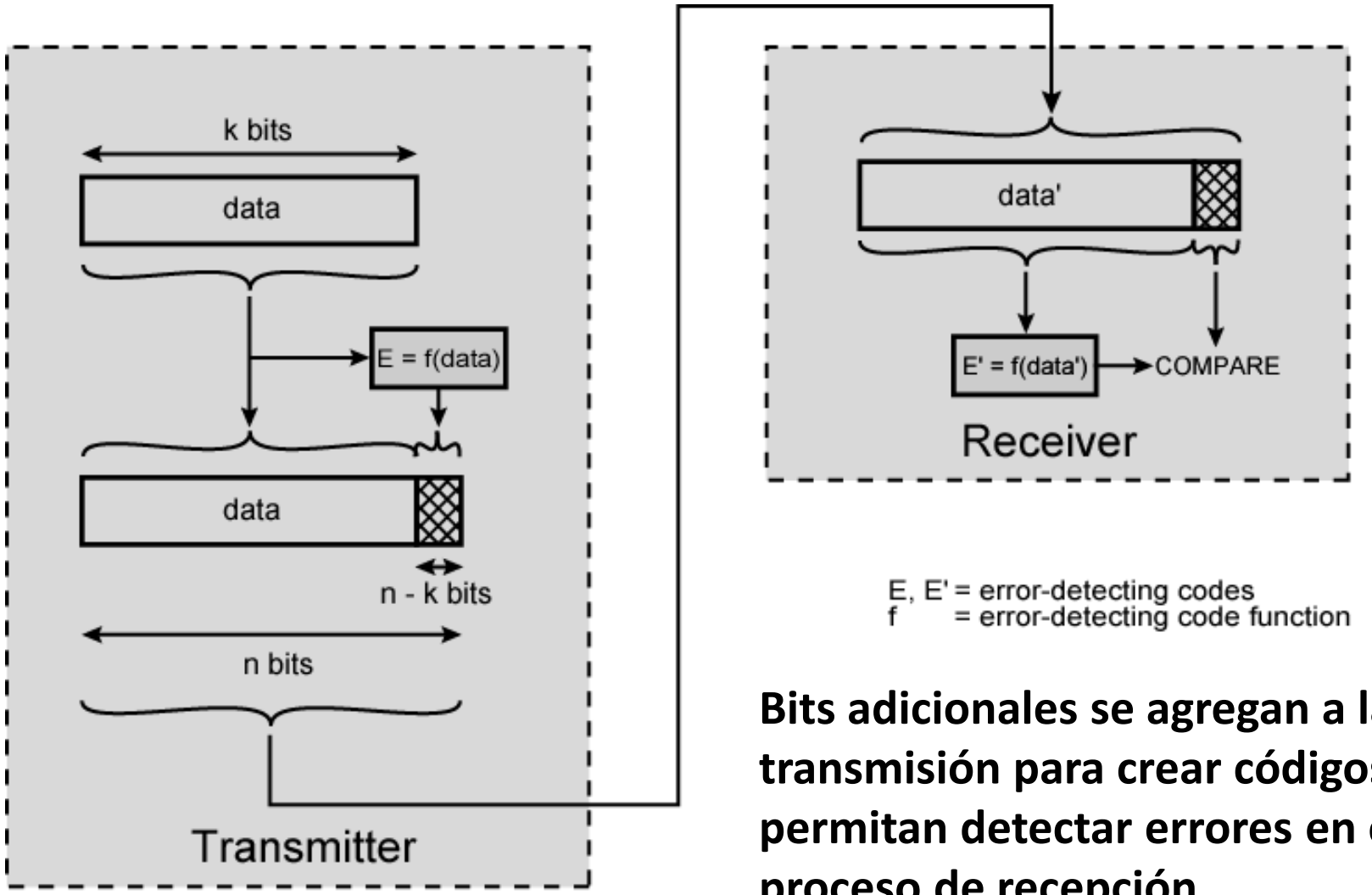
Errores

- **Los códigos de detección de errores** añaden redundancia a los datos de tal forma que puedan ser detectados los errores.
- Si se detecta error las tramas deben ser retransmitidas.
- Se usan en canales con pocos errores (cobre, fibra)

- **Los Códigos de corrección de errores** detectan y corregir errores, son matemáticamente complejos, pero ampliamente usados en los sistemas reales.
- Se usan en canales muy ruidosos o en aquellos que es caro retransmitir la información (radio, satélite).

LA CAPA DE ENLACE

Detección de Errores



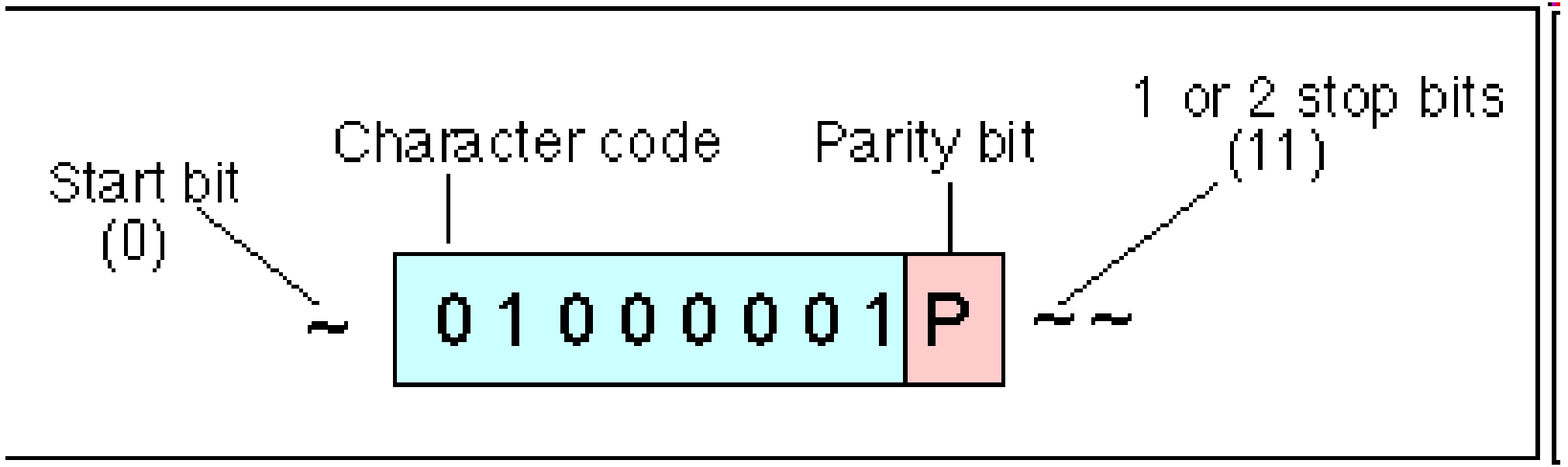
Bits adicionales se agregan a la transmisión para crear códigos que permitan detectar errores en el proceso de recepción

LA CAPA DE ENLACE

Detección de Errores - Paridad

Se inserta un bit a la cadena de datos.

- El valor del bit de paridad es tal que cada carácter poseerá un número par (**paridad par**) o impar (**paridad impar**) de “unos” (XOR, Ejemplo: 1110000 → 11100001).
- La detección chequea si la suma es errada (error), pero solo detecta un **numero impar** de errores
- Errores aleatorios son detectados con probabilidad 1/2



LA CAPA DE ENLACE

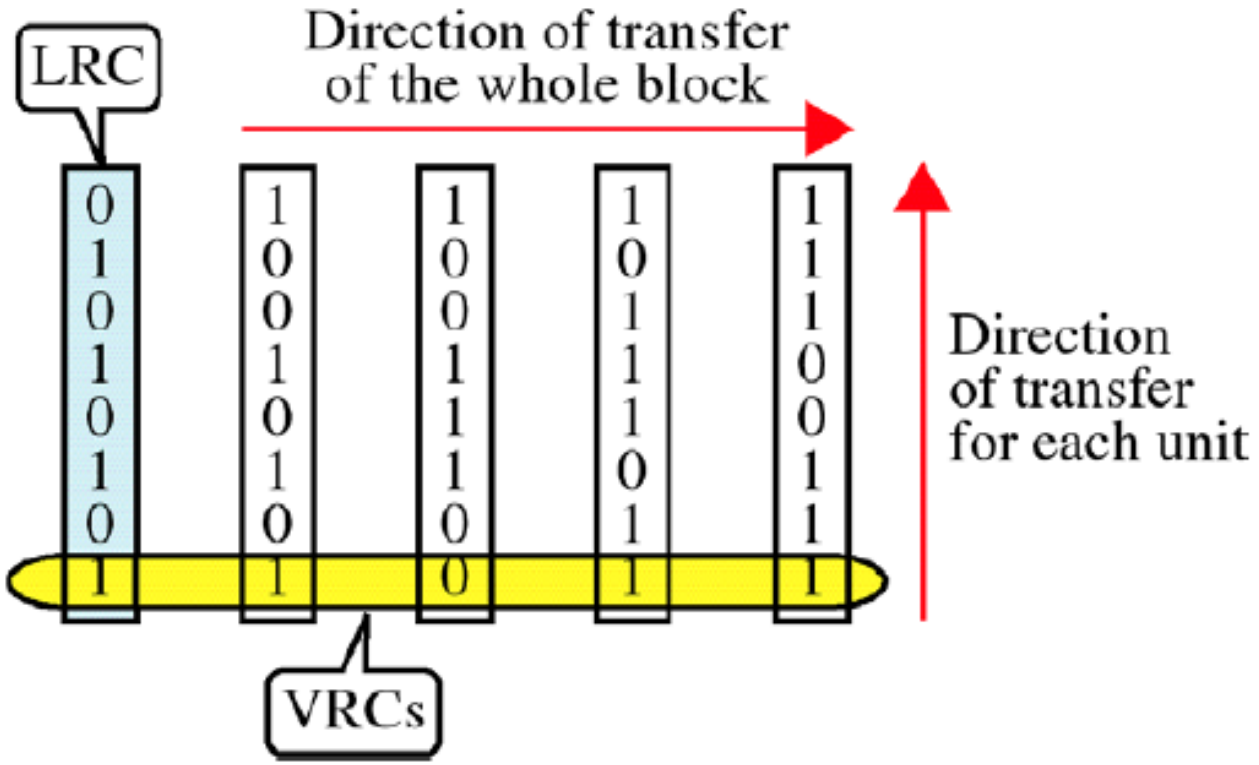
Detección de Errores - Paridad

Paridad Compleja:

LRC = Verificación de redundancia longitudinal

VRC= Verificación de redundancia vertical

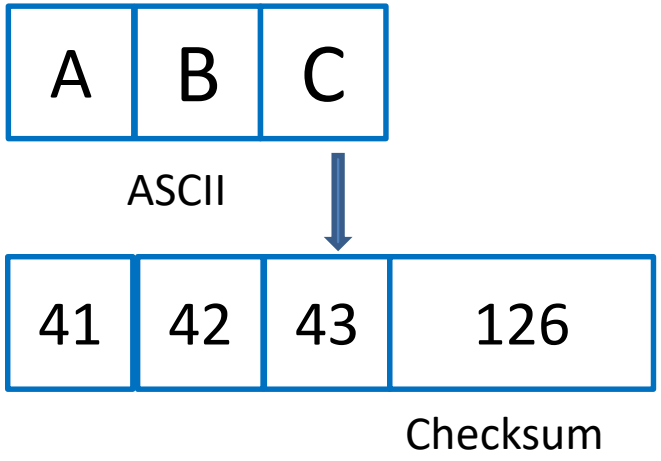
⌘VRC y LRC



LA CAPA DE ENLACE

Detección de Errores - Checksum

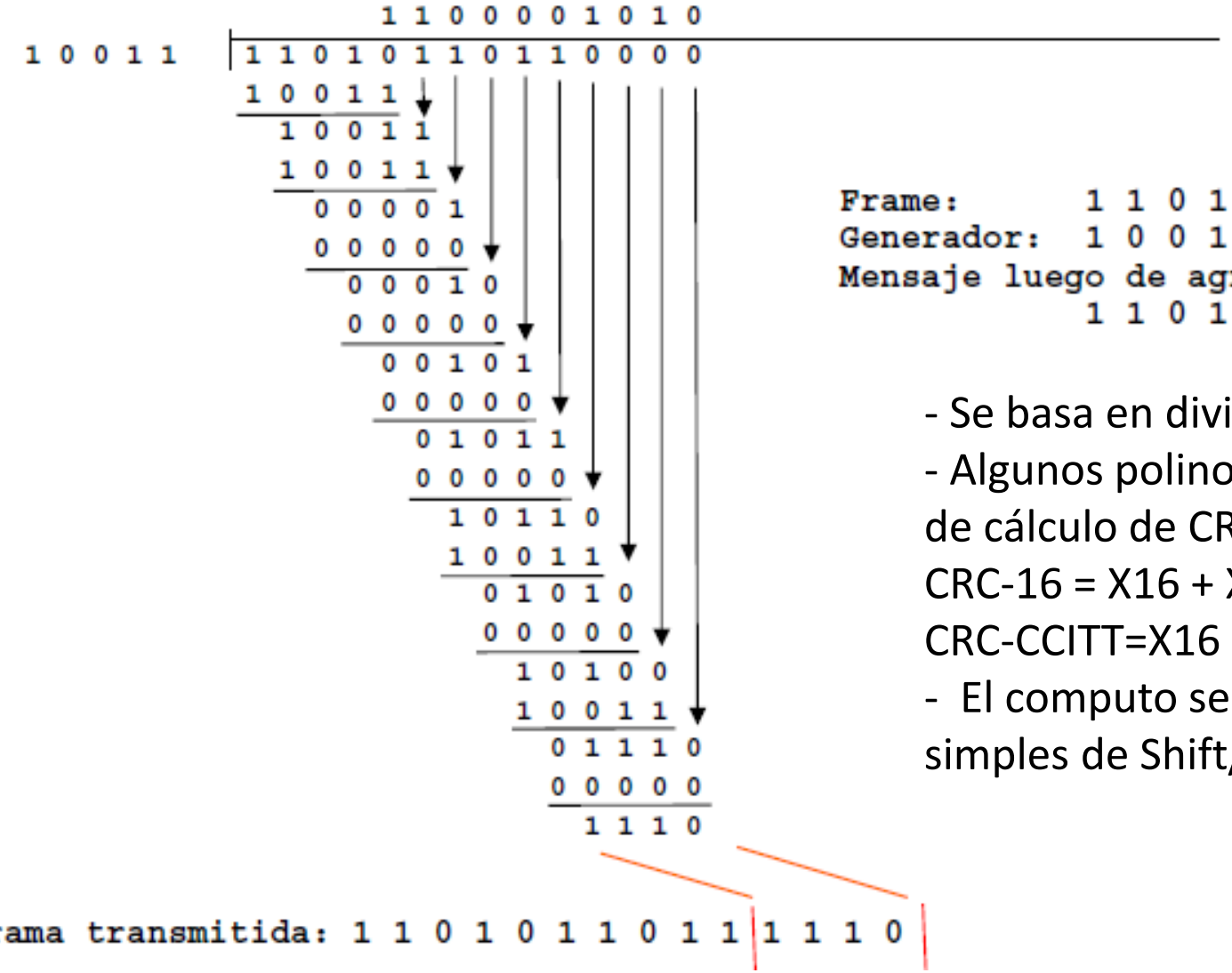
- El emisor de la trama realiza la suma de los bytes transmitidos (su representación como caracteres ASCII) de todos o en determinados campos de la trama.
- Esta suma se realiza módulo 256 o 65536, esto generará 8 o 16 bits, de información para el control de errores,
- Estos bits se añaden al final de la trama o del campo que se supervise
- Mejora la detección de errores en comparación con los bits de paridad



41+42+43=126

LA CAPA DE ENLACE

Detección de Errores - CRC




```
Frame:       1 1 0 1 0 1 1 0 1 1
Generador:   1 0 0 1 1
Mensaje luego de agregar 4 ceros
             1 1 0 1 0 1 1 0 1 1 0 0 0 0
```

- Se basa en división de Polinomios
- Algunos polinomios utilizados en el de cálculo de CRC:

CRC-16 = X16 + X15 + X2 + 1

CRC-CCITT=X16 + X12 + X5 + 1

- El computo se hace con circuitos simples de Shift/XOR

LA CAPA DE ENLACE

Detección de Errores - CRC

Un buen polinomio CRC podrá descubrir:

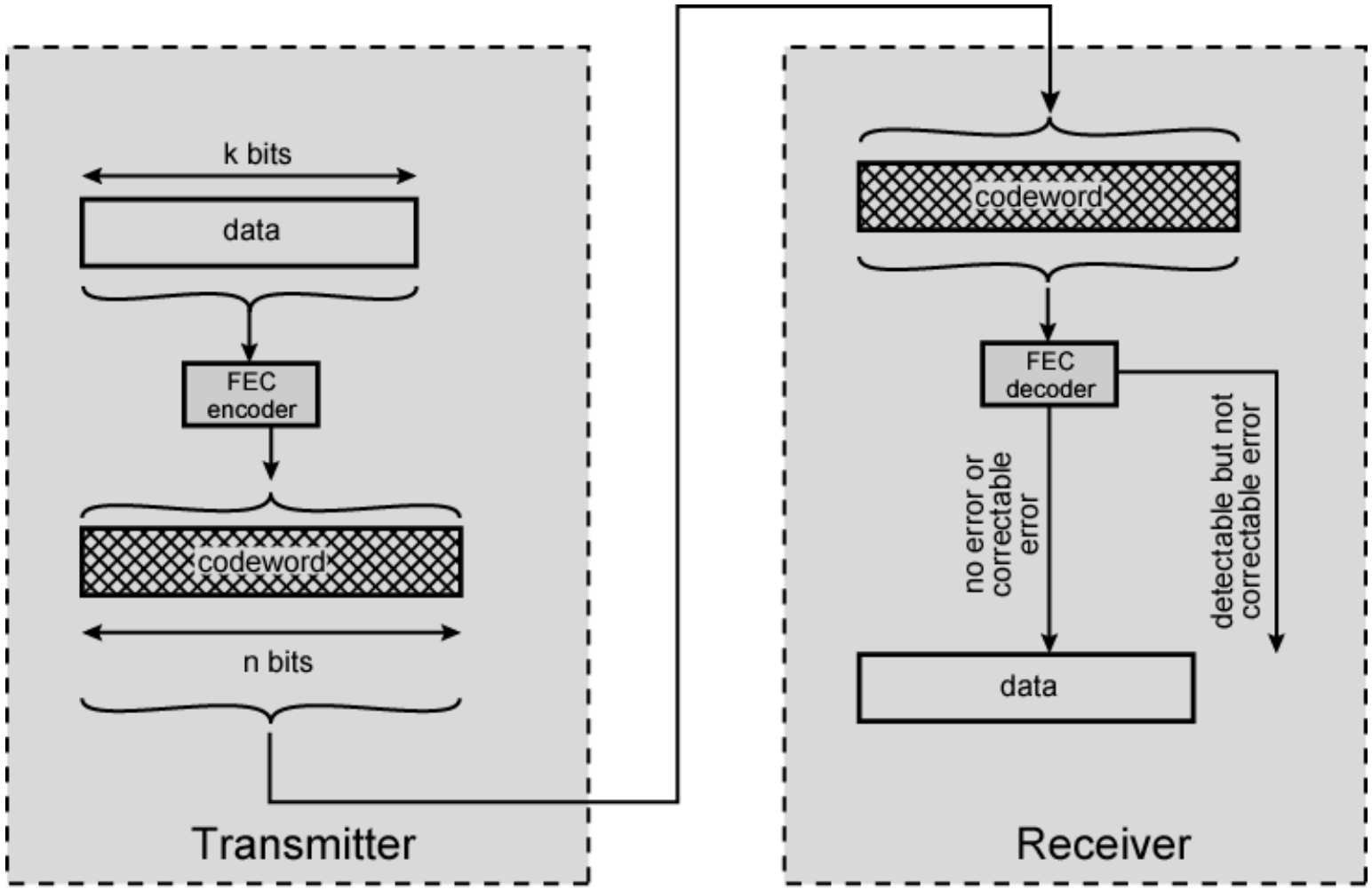
- Cualquier ráfaga contigua de errores más corta que el polinomio.
- Cualquier número impar de errores en el bloque.
- Cualquier error de 2 bits en el bloque.
- Entonces cada arreglo posible de errores de 1, 2, o 3 trozos será descubierto. Pueden existir errores que no sean descubiertos.
- No es vulnerable a los errores sistemáticos (0's continuos).

Ejemplo: CRC de 32-bit usado en redes Ethernet

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

LA CAPA DE ENLACE

Corrección de Errores



Forward **E**rror **C**orrection

LA CAPA DE ENLACE

Corrección de Errores - Hamming

Esta codificación proporciona una manera simple de añadir bits de chequeo y corregir hasta un error de un bit:

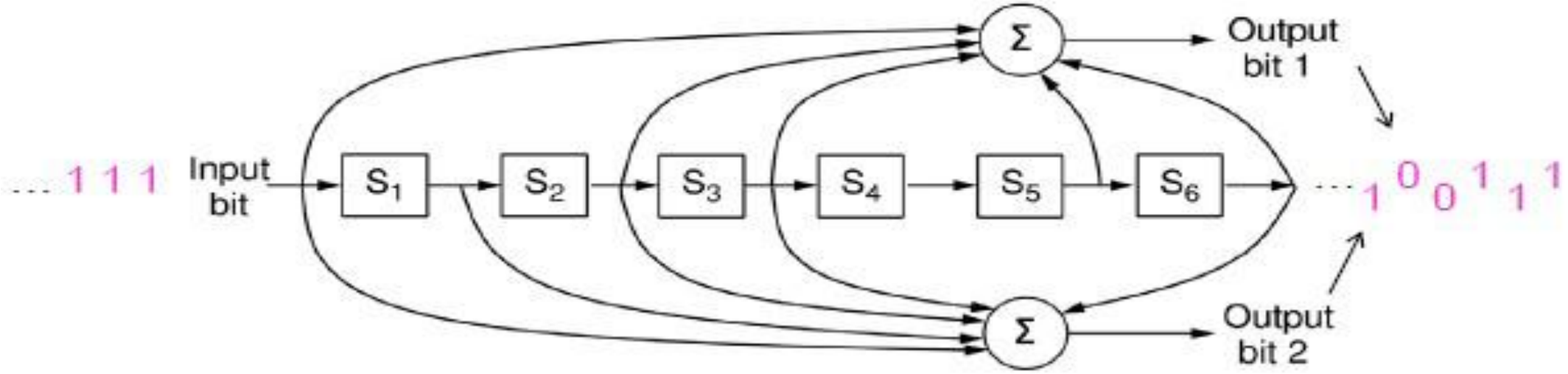
- **La distancia de Hamming** es la cantidad de posiciones de bits en la que difieren dos palabras codificadas.
- Ejemplo con 4 palabras codificadas de 10 bits (n=2, k=8):
  - 0000000000, 0000011111, 1111100000, and 1111111111
  - La distancia de Hamming es 5
- Bits de chequeo son de paridad sobre subconjuntos de palabras codificadas
- El recalculo de sumas de paridad (syndrome) proporciona la posición del error que cambiar, si es cero no hay error.

LA CAPA DE ENLACE

Corrección de Errores – Códigos Convolucionales

Opera sobre un flujo de bits, manteniendo un estado interno

- El flujo de salida es una función de todos los bits precedentes de entrada
- Los bits son decodificados con el algoritmo de Viterbi



Ejemplo: código convolucional binario popular de la NASA (rate = 1/2) usado en 802.11

LA CAPA DE ENLACE

Corrección de Errores – Reed Salomon

Es el Código FEC mas utilizado en la actualidad (CD's, DVD's, XDSL, Wimax, Arreglos de disco, etc).

Se basa en polinomios, los valores del código son los "puntos" de una curva.

Se denota Código (X,Y), X símbolos contienen Y datos.

Ejemplo: Código (255,223) es decir 255 símbolos transportan 223 datos (usa 8 bits), por lo que 32 son adicionales.

32 Símbolos adicionales permiten detectar 32 errores y corregir 16.

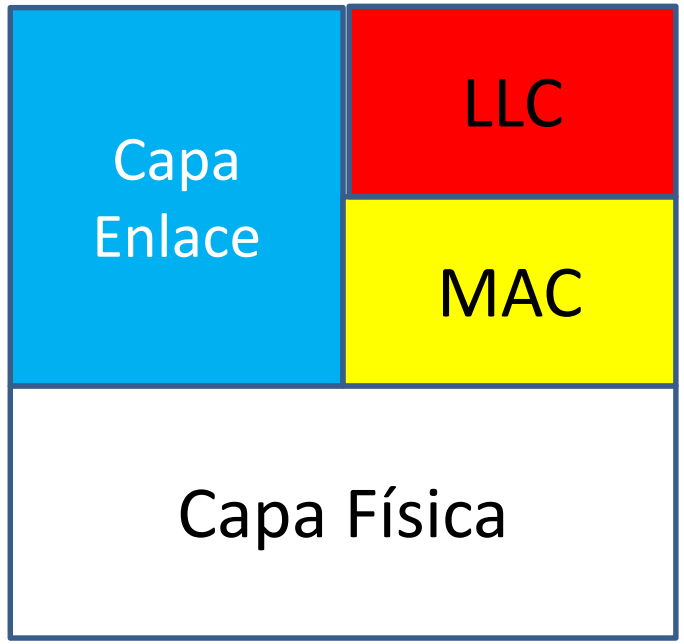
Como regla (X,Y) detectan X-Y errores y corrigen (X-Y)/2

LA CAPA DE ENLACE

Mitad HW/SW, Consta de dos subcapas:

LLC (Logical Link Control) (Drivers): Proporciona versatilidad en los servicios de los protocolos para la capa de red. Brindando algunos servicios de manejo del enlace, como sincronía y control de flujo.

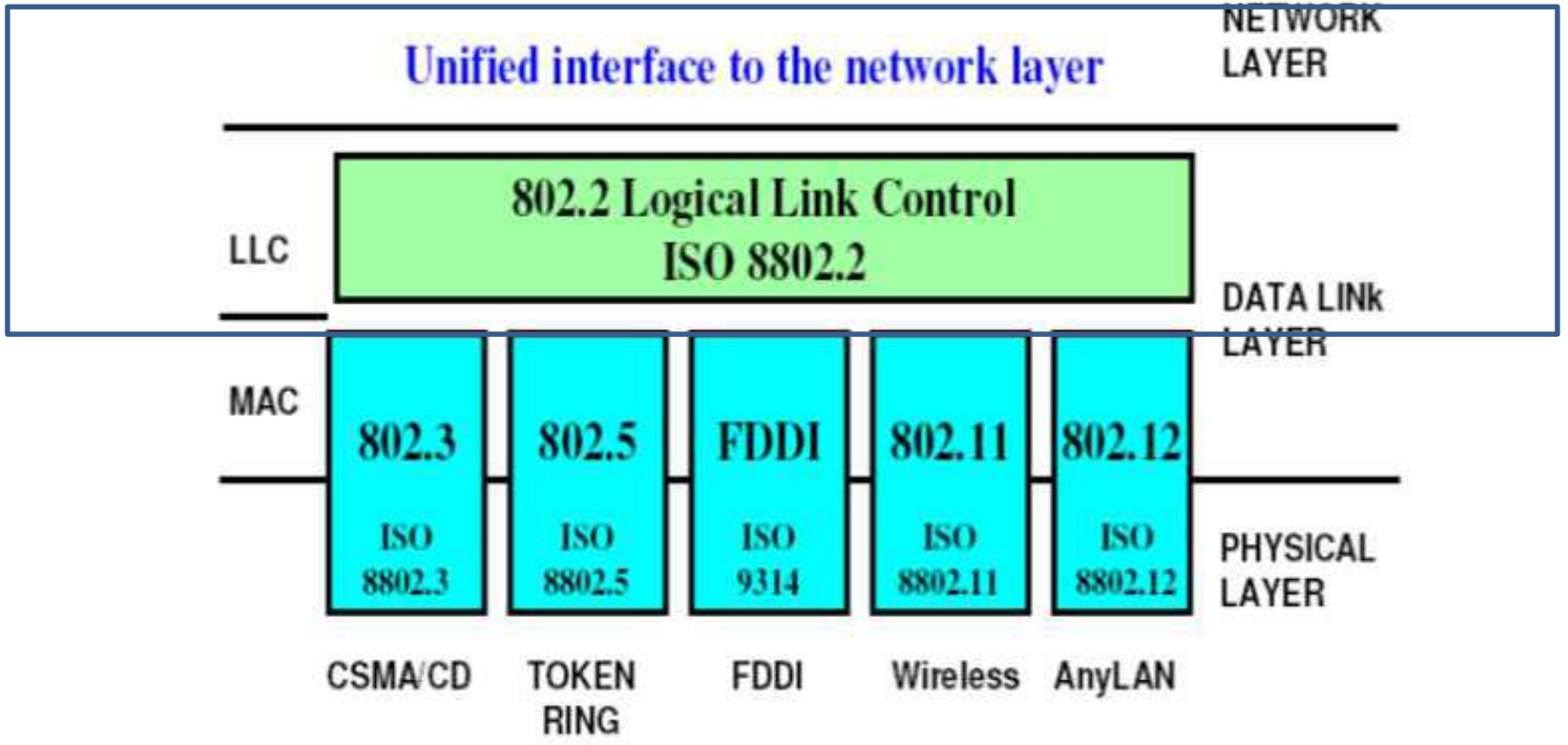
MAC (Media Access Control): (NIC) Es la capa responsable del manejo de los protocolos que un host sigue para poder acceder a los medios físicos.



LA CAPA DE ENLACE

LLC

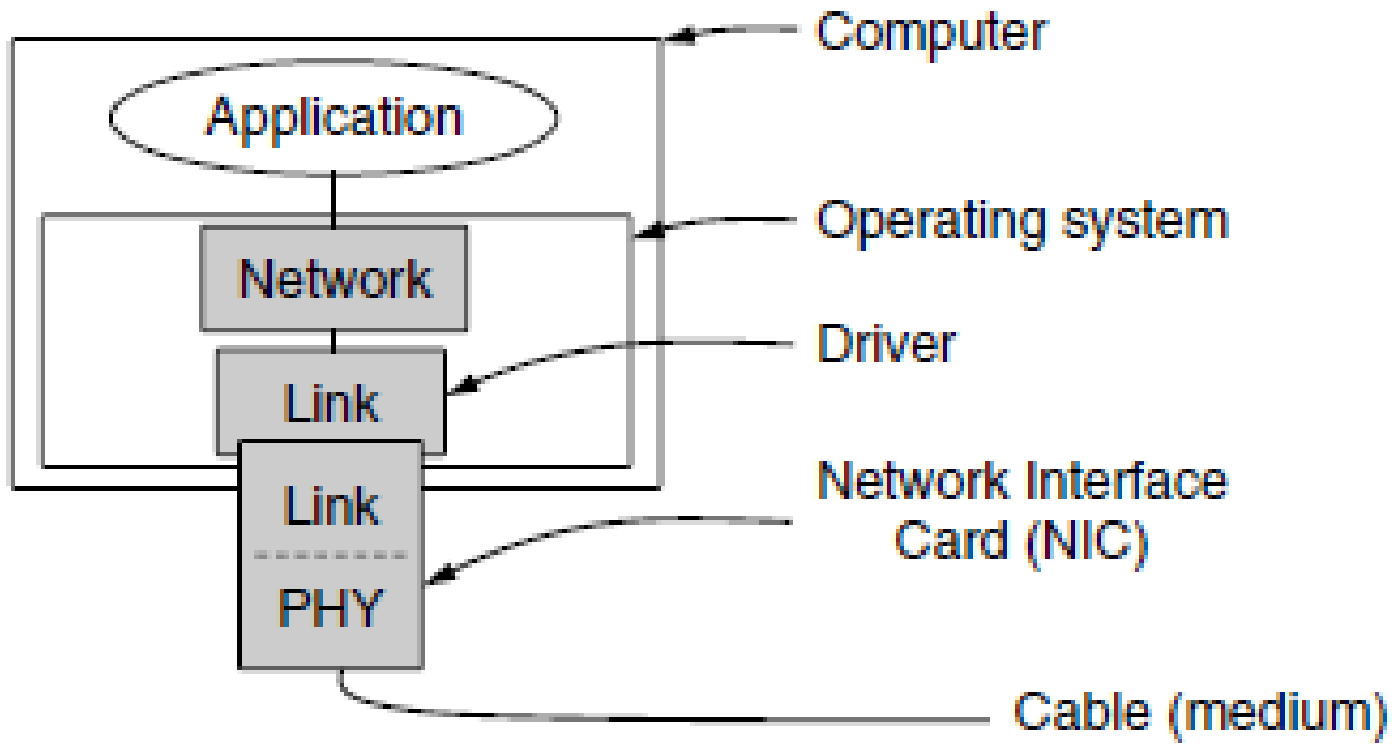
- La sub capa de LLC es común a todas las LANs
- El estándar IEEE 802.2 y el ISO 8802.2 describen servicios y protocolos para esta sub-capa



LA CAPA DE ENLACE

LLC: Protocolos de la capa de enlace

La sub capa LLC contiene los protocolos de ensamblaje de las tramas (Drivers) y su preparación para inserción en la parte fisca



LA CAPA DE ENLACE

LLC: Protocolos de la capa de enlace

Control de Flujo (Stop and Wait) asegura que el emisor no sobrepase la capacidad del receptor:

- Receptor retorna una trama simple frame (ack) cuando esta listo
- Solo se entrega una trama por vez – de allí el nombre

```
#include "protocol.h"

void sender2(void)
{
  frame s;                          /* buffer for an outbound frame */
  packet buffer;                    /* buffer for an outbound packet */
  event_type event;                 /* frame_arrival is the only possibility */

  while (true) {
      from_network_layer(&buffer);  /* go get something to send */
      s.info = buffer;              /* copy it into s for transmission */
      to_physical_layer(&s);        /* bye-bye little frame */
      wait_for_event(&event);       /* do not proceed until given the go ahead */
  }
}
```

LA CAPA DE ENLACE

LLC: Protocolos de la capa de enlace

Ejemplo del lado receptor

```
void receiver3(void)
{
  seq_nr frame_expected;
  frame r, s;
  event_type event;

  frame_expected = 0;
  while (true) {
       wait_for_event(&event);                  /* possibilities: frame_arrival, cksum_err */
       if (event == frame_arrival) {            /* a valid frame has arrived */
            from_physical_layer(&r);            /* go get the newly arrived frame */
            if (r.seq == frame_expected) {      /* this is what we have been waiting for */
                 to_network_layer(&r.info);     /* pass the data to the network layer */
                 inc(frame_expected);           /* next time expect the other sequence nr */
            }
            s.ack = 1 – frame_expected;         /* tell which frame is being acked */
            to_physical_layer(&s);              /* send acknowledgement */
       }
  }
}
```

LA CAPA DE ENLACE

LLC: Protocolos de la capa de enlace

En un Canal Ruidoso

ARQ (Automatic Repeat reQuest) añade control de flujo y de errores

- Receptor confirma (ACK's) tramas correctas. Si están erradas envía NACK.
- Emisor activa un temporizador y reenvía una trama si no llega un ACK en un tiempo predefinido (timeout)
- Para claridad, tramas y ACKs deben numerarse, de lo contrario el receptor no puede pedir retransmisión
- Para parada y espera, 2 números (1 bit) son suficientes, en otros esquemas puede ser hasta una palabra completa.

LA CAPA DE ENLACE

LLC: Protocolos de la capa de enlace

Ventanas deslizantes

- Emisor y Receptor mantienen una “ventana” de tramas para el intercambio de información
    - Ambos necesita colas para las tramas en caso de que se necesite retransmitir
    - La ventana avanza a medida que llegan las tramas y son enviados los acuses de recibos (ACK)
    - Si se recibe un NACK o se vence el timeout, se retransmite todo.

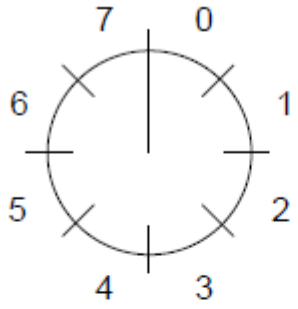
LA CAPA DE ENLACE

LLC: Protocolos de la capa de enlace

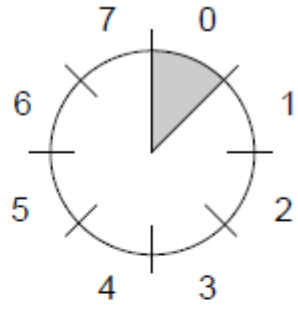
Ventanas deslizantes avanzando en el emisor y receptor

- Ejemplo: Tamaño de ventana es 1, con un numero de secuencia de 3 bit.

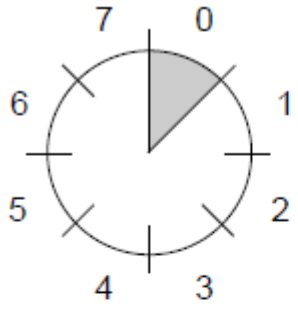
Sender



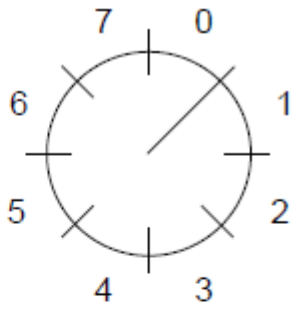
Inicio



Envía 1 Tx

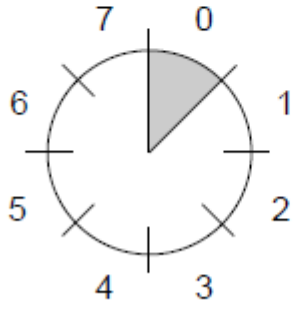


Espera ACK0

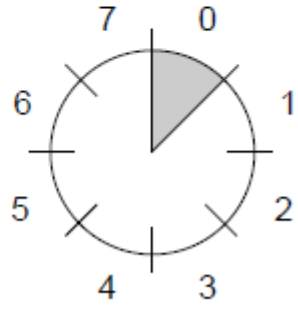


Ack recibido

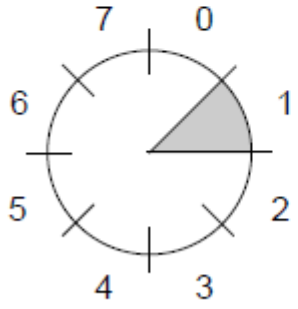
Receiver



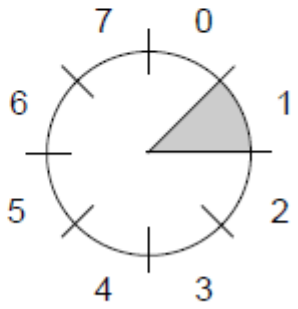
Inicio



Espera 0



Recibe 1 Tx



Espera 1

LA CAPA DE ENLACE

LLC: Protocolos de la capa de enlace

Ventanas deslizantes

- Ventanas mas grandes permiten “canalización” (pipelining) para un uso mas eficiente del enlace
    - Parada y espera (w=1) es ineficiente para enlaces a grandes distancias
    - La mejor ventana (w) depende en el ancho de banda y el retardo (BD)
    - Se quiere $w \geq 2BD+1$ para asegurar una alta utilización del enlace
- Pipelining implica diferentes elecciones para errores/buffering

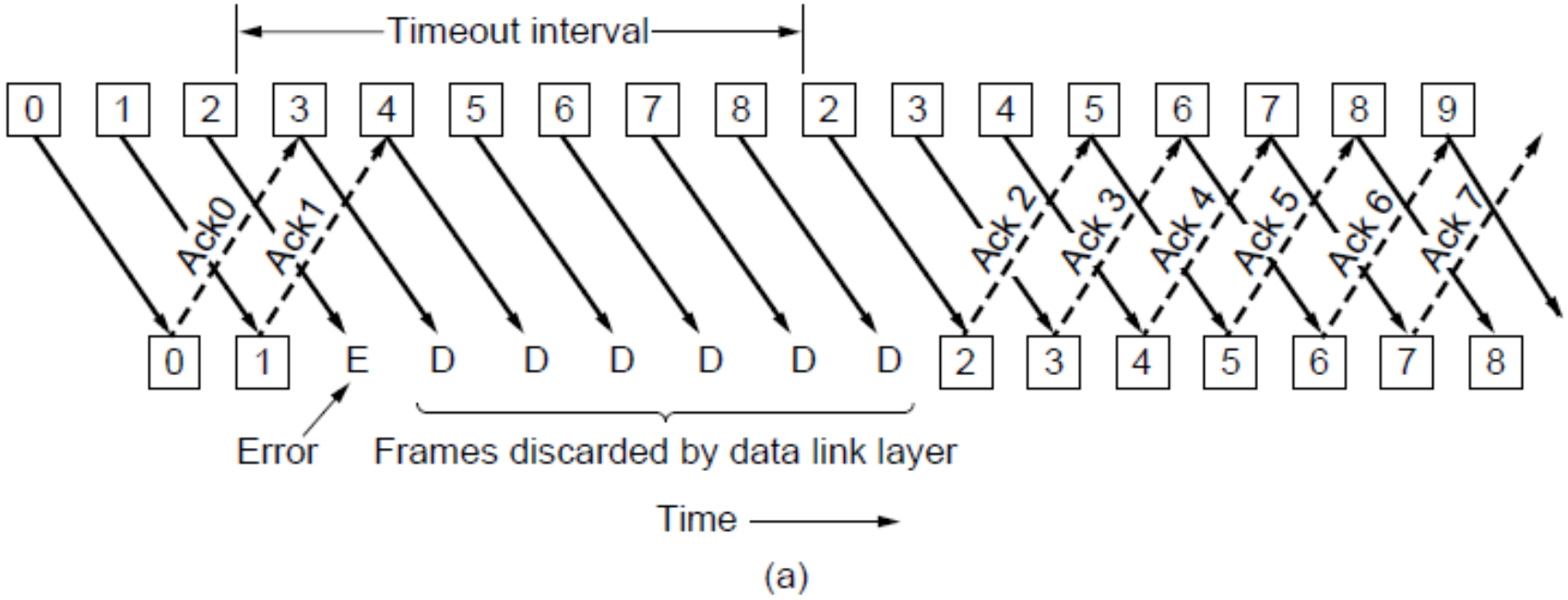
LA CAPA DE ENLACE

LLC: Protocolos de la capa de enlace

Go-Back-N

El receptor solo acepta/confirma tramas en el orden de llegada:

- Se descartan las tramas posteriores a una trama con errores o perdidas
- El emisor envía todas las tramas relevantes si ocurre una terminación del tiempo

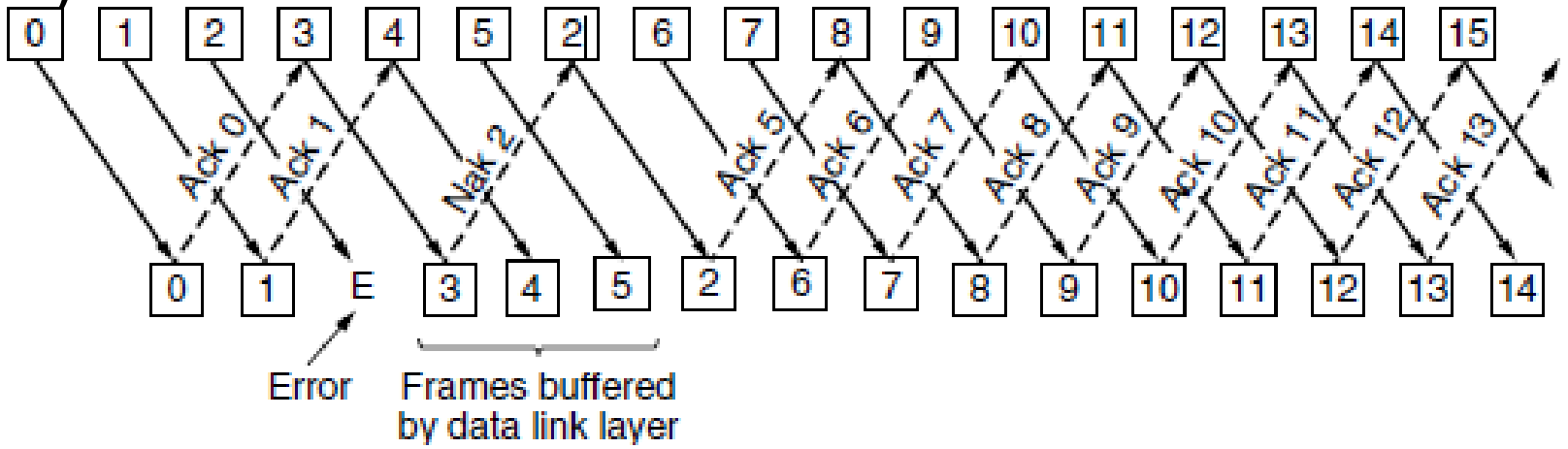


LA CAPA DE ENLACE

LLC: Protocolos de la capa de enlace

Repetición Selectiva

- Similar al anterior pero La ventana en el receptor es mayor a 1.
- Mas complejo que Go-Back-N debido al encolado en el receptor y a los múltiples temporizadores en el emisor
- Uso mas eficiente del ancho de banda del enlace, debido a que solo las tramas perdidas son enviadas (con tasas de fallas bajas)
- Para precisión se requiere números de secuencia (s) de al menos el doble de la ventana (w)



USB