

PAR - Unidad 4

LA CAPA DE ENLACE DE DATOS: SUBCAPA MAC

Enlace de datos

- Se llama **enlace de datos** a la conexión entre dos o más nodos (o entidades) de red de forma “directa”, mediante el mismo medio físico (capa física)
- Por tanto, los protocolos de enlace de datos deben suministrar los medios funcionales y de procedimiento para la comunicación entre:
 - dos nodos de red adyacentes en una WAN
 - los nodos de un mismo segmento de una LAN
- La unidad de paquete de datos en esta capa se llama **trama**
- Ejemplos de protocolos de la capa de enlace de datos

Capa de Enlace - Diagrama (1)

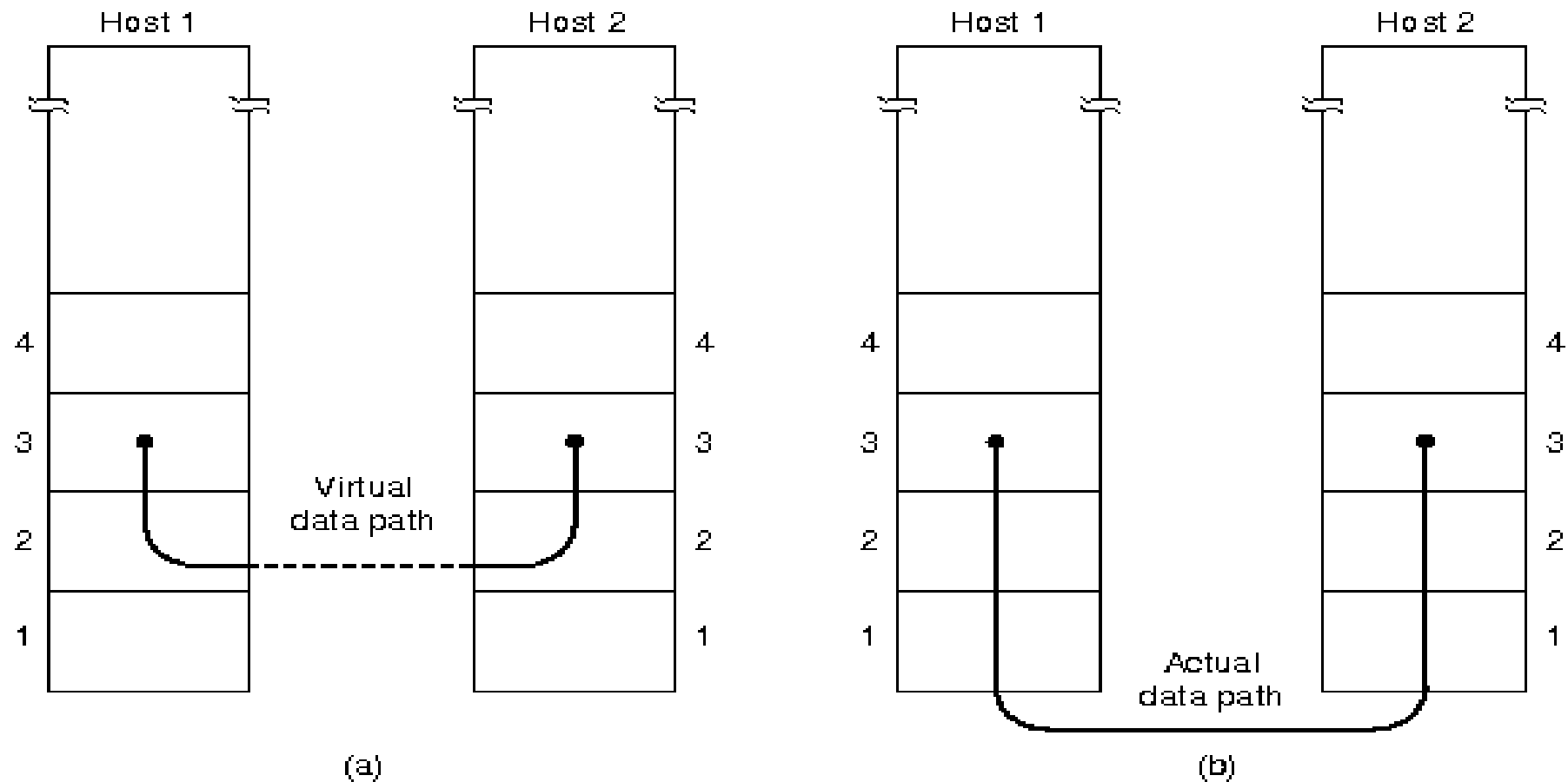


Fig. 3-2. (a) Virtual communication. (b) Actual communication.

Capa de enlace - Diagrama (2)

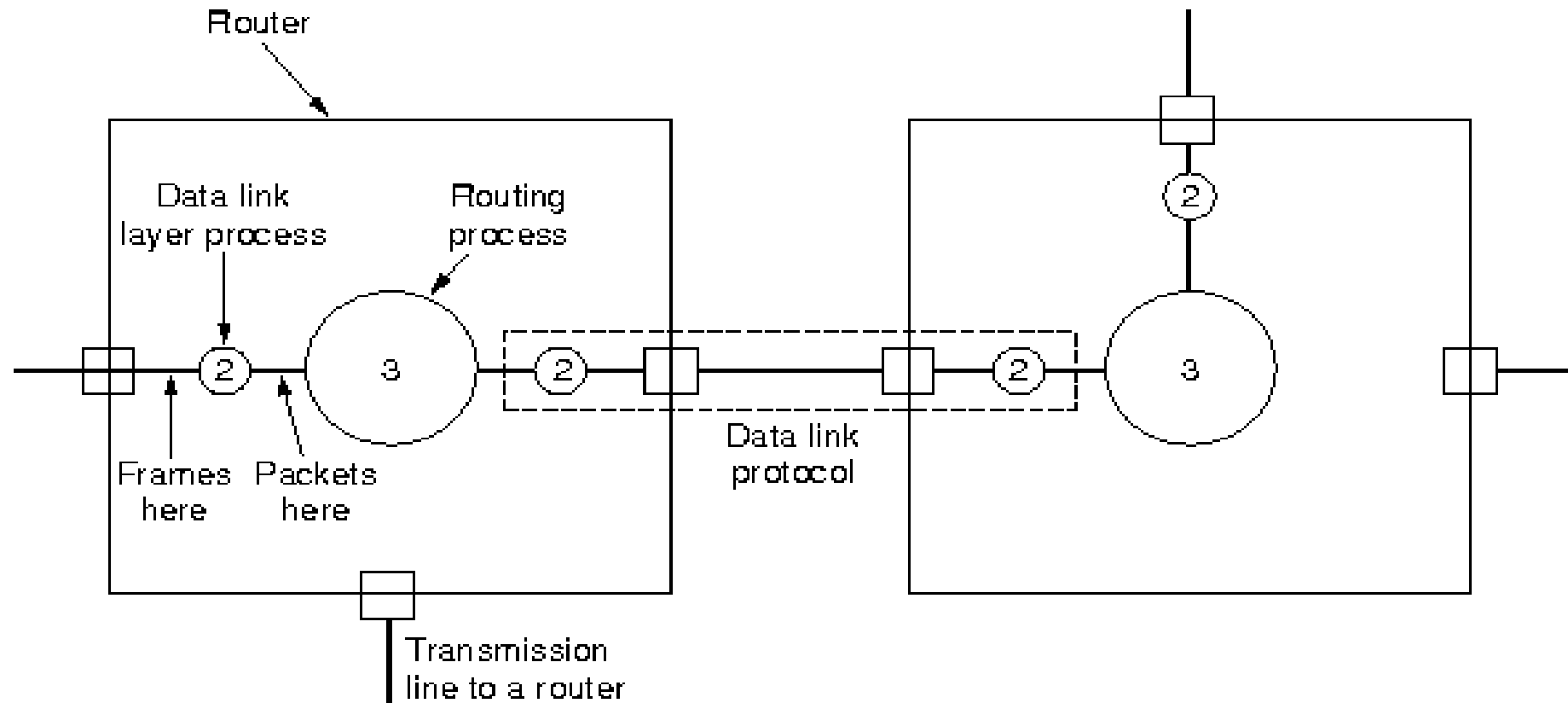


Fig. 3-3. Placement of the data link protocol.

Tipos de Servicios

- Esta clasificación **se aplica también a todas las capas superiores**
- Los protocolos ofrecen servicios a las capas superiores mediante una interfaz
- Estos servicios pueden ser de dos tipos:
 - **No orientados a la conexión:**
 - no se establece conexión de antemano entre las entidades que pone en comunicación el protocolo
 - dos tipos en función del control de errores:
 - No fiable (sin acuse de recibo)
 - Fiable (con acuse de recibo)
 - **Orientados a la conexión:**
 - se establece una conexión antes de comenzar la transferencia de datos entre las entidades

No orientados a la conexión

- Hay dos tipos en función de si se controlan o no los errores:
 - **No fiable:**
 - si se pierde una trama, no se realiza ningún esfuerzo por detectar la pérdida => sin necesidad de control de flujo
 - apropiado cuando la tasa de errores es muy baja, por lo que la detección y corrección de estos se deja a las capas superiores
 - también apropiado para el tráfico en tiempo real
 - se utiliza en las **mayoría de las LANs**
 - **Fiable:**
 - se detectan los daños y pérdidas de tramas, y se tratan de recuperar
 - útil en canales inestables, como los inalámbricos

Orientados a la conexión

- Se establece una conexión antes de transmitir las tramas
- Control de errores y de flujo:
 - Cada trama está numerada y **se garantiza** que llegará a su destino en el **mismo orden y sin duplicados**
 - Ofrece a la capa de red un flujo de bits confiable
- Tres fases:
 - conexión: se negocia y se reserva recursos
 - transmisión
 - desconexión: liberación de recursos
- Se usa típicamente en WANs, p.e. PPP, ATM, ...

Subcapas de la capa de enlace

- Para atender a estas necesidades, se suele dividir esta capa en dos subcapas para ofrecer sus correspondientes **servicios**:
 - **MAC** o de Control de Acceso al Medio, que asume las funciones de:
 - división, delimitación y sincronización de los paquetes en tramas para su entrega a la capa física
 - asignación del canal y direccionamiento inequívoco
 - comprobación de errores y corrección de estos hacia delante o **FEC** (no siempre)
 - **LLC** o de Control de Enlace Lógico, que asume las funciones de:
 - recepción de los paquetes de la capa de red
 - control de flujo (no siempre se controla)
 - corrección de errores hacia atrás o **ARQ** (no siempre)

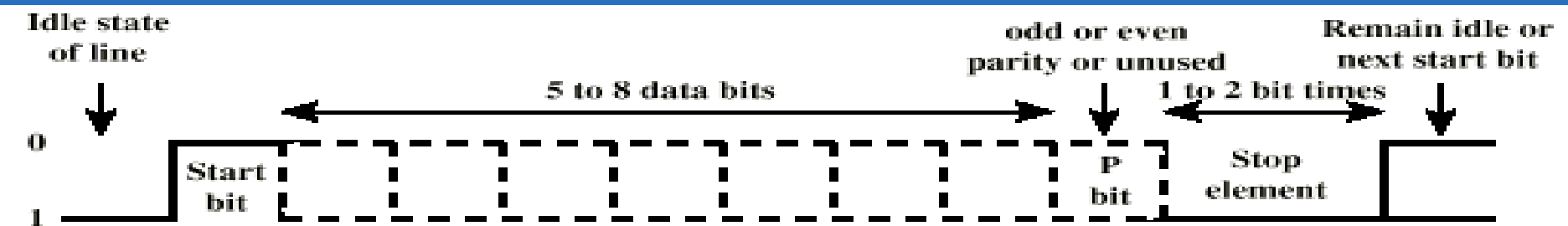
MAC – Sincronización del canal

- La **división en tramas** de los datos facilita:
 - la compartición del canal
 - la corrección de errores
 - la sincronización
- Para determinar correctamente el valor binario en la recepción de los datos digitales hay que realizar un **muestreo de la señal recibida** precisamente en los instantes que comienza cada bit
- P.e., si el emisor transmite a $1\text{Mbps} = 1\text{bit}/\mu\text{s}$ y el reloj del receptor es un 1% más lento:
 - el muestreo 1 estará desplazado $0,01\ \mu\text{s}$
 - el 50 será de $50 \times 0,01\mu\text{s} = 0,5\mu\text{s}$, que producirá un error por muestreo incorrecto
- **Dos soluciones: la asíncrona y la síncrona**

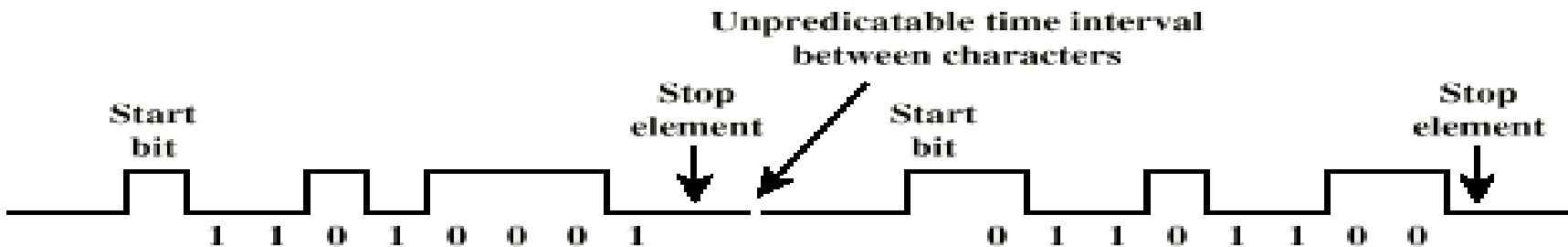
MAC – Transmisión Asíncrona

- En la transmisión asíncrona se evita el problema no enviando largas cadenas o bloques de bits, sino **carácter a carácter** (de 5 a 8 bits)
- Así la sincronización sólo necesita mantenerse dentro de cada carácter
- El receptor puede resincronizarse con cada carácter
- Problemas para transmitir datos binarios
 - deben usarse un *caracter de escape* para evitar confusiones (p.e., para distinguir el salto de línea, código ASCII 10, del simple número 10. ¿Cuál es el carácter de escape en ASCII?)

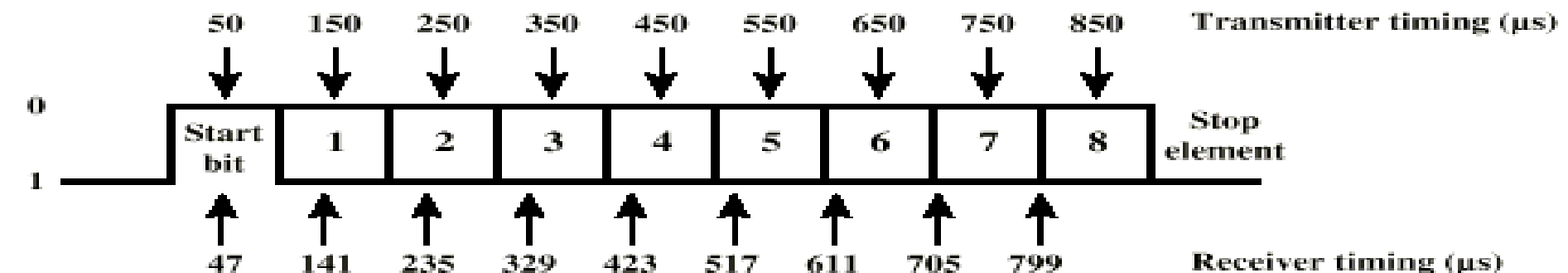
MAC asíncrona (diagrama)



(a) Character format



(b) 8-bit asynchronous character stream



(c) Effect of timing error

MAC asíncrona (funcionamiento)

- El intervalo entre caracteres (longitud entre elementos de parada) es uniforme (reposo):
 - en estado de reposo, el receptor busca transiciones de 1 a 0
 - cuando se detecta una, muestrea los siete intervalos siguientes (la longitud del carácter) hasta que llega el elemento de parada
 - de nuevo en el estado de reposo, busca la siguiente transición de 1 a 0 que marca el inicio del siguiente carácter
- **Simple y barato, pero con sobrecarga** de 2 o 3 bits por carácter (~20%)
- Buena para datos intermitentes (p.e., teclado)

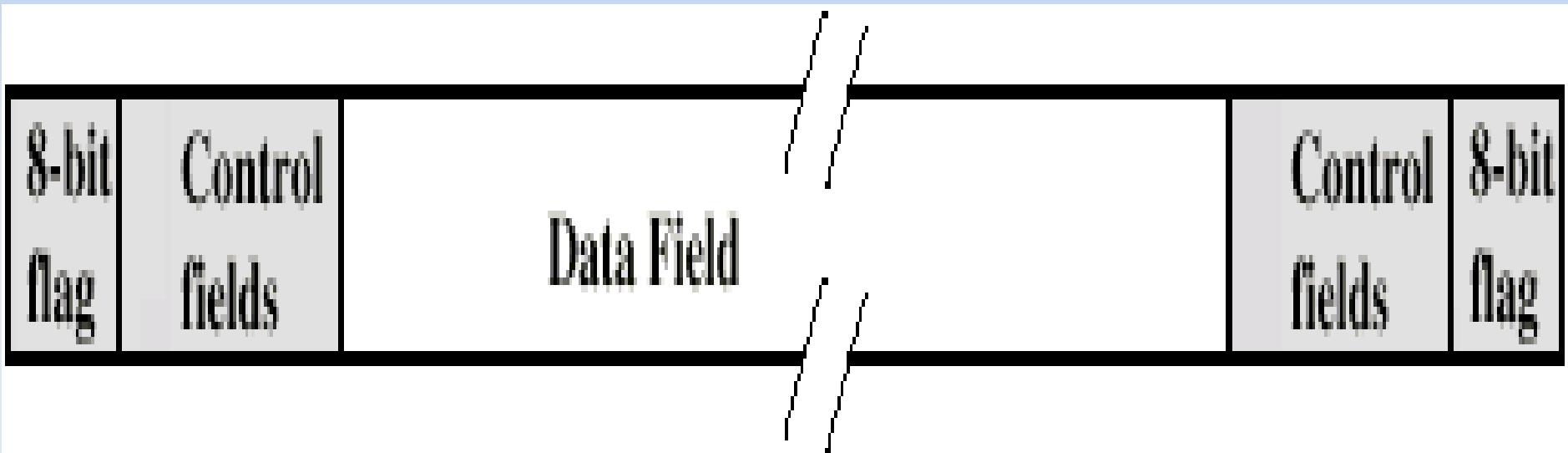
MAC – T. Síncrona

- En la transmisión síncrona se transmiten grandes **bloques de datos**
- Los **relojes deben estar sincronizados**, para ello puede usarse:
 - una línea separada de reloj
 - efectiva en las distancias cortas (p.e. en una placa base)
 - sujeta a los mismos impedimentos que los datos
 - confianza en una buena precisión de los relojes
 - cada vez los relojes son más precisos
 - una *señal virtual de reloj* embebida en la señalización de los datos:
 - como en la señalización bifase (p.e. tipo Manchester) o en el mapeo de bits (p.e. 8b/10b)

MAC síncrona (funcionamiento)

- Además se necesita **indicar el comienzo y el final de cada bloque**
- Para ello se usan patrones de bits, preámbulo y final:
 - p.e. serie de caracteres SYN (¿cuál es su código ASCII?)
 - p.e. inicio con 11111111 que acaban con 11111110
 - se tiene que evitar secuencias de bits que coincidan con dichos patrones mediante el **relleno de bits**, que será eliminado por el receptor
- **Más eficiente** (menor sobrecarga) que la asíncrona, es el sistema más usado hoy en día
- Al conjunto de bits formado por los datos más el preámbulo más el final se le denomina **trama**

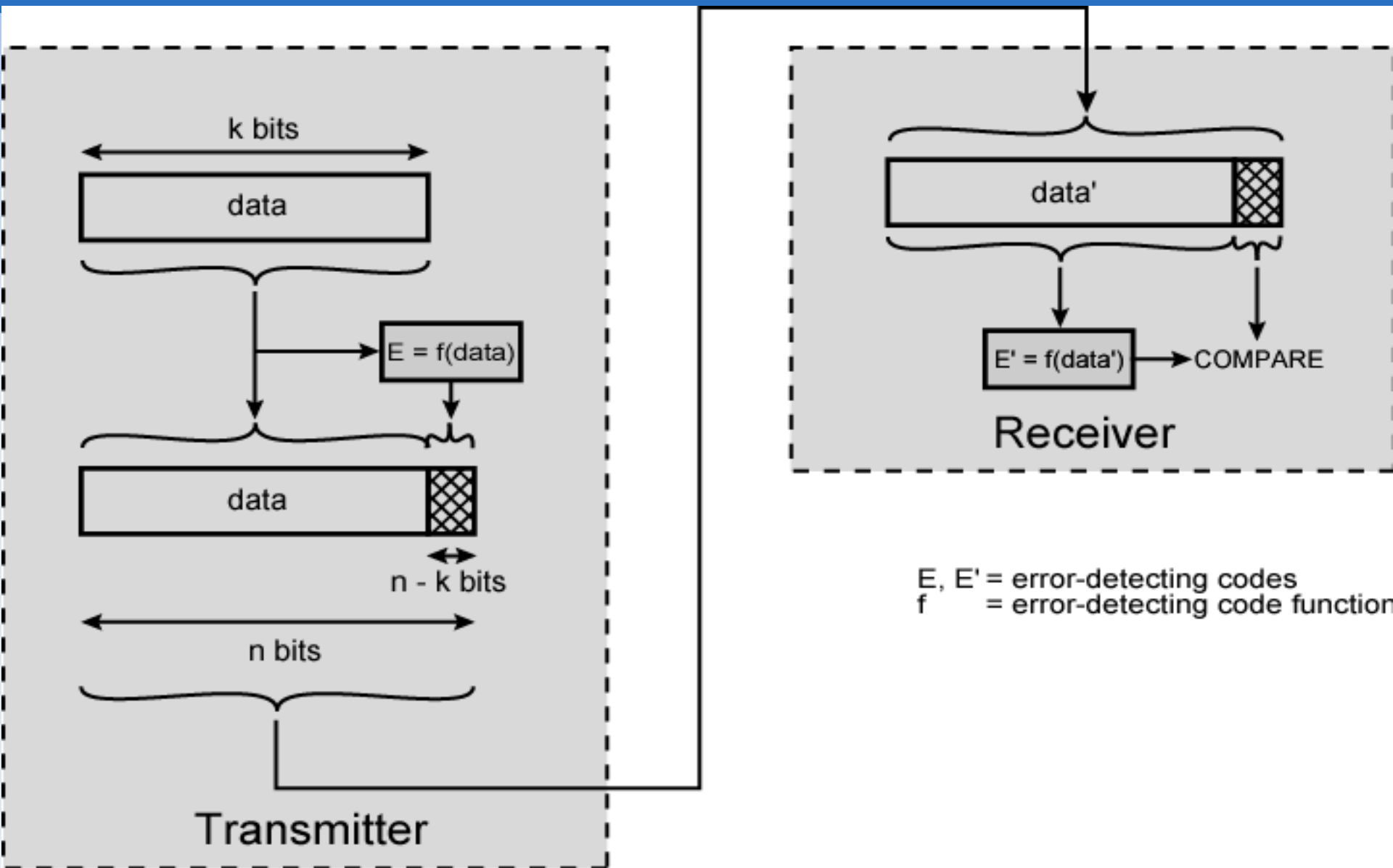
MAC – T. Síncrona (diagrama)



MAC – Control de Errores

- Un error ocurre cuando **un bit es alterado** entre la transmisión y la recepción
- **Errores de un solo bit**
 - un bit alterado
 - los bits adyacentes no son afectados
 - típico del ruido blanco
- **Errores en ráfagas**
 - secuencia contigua de B bits en la que el primer, el último y cualquier número de bits intermedios son erróneos
 - típico del ruido impulsivo o del desvanecimiento por multitrayectoria en la comunicación inalámbrica
 - efecto mayor a mayores tasas de datos

MAC - Detección de Error



MAC – D. Errores de Paridad

- **Bits añadidos por el transmisor** para formar un código con capacidad de detectar errores:
 - de un bloque de n bits: k bits de datos y $n-k$ bits de comprobación
- **Comprobación de Paridad**
 - se añade a los datos un bit de paridad, que toma su valor de forma que mantenga un número par de unos (paridad par) o impar (paridad impar)
 - un número par de errores pasará inadvertida
 - se emplea la transmisión de tramas por columnas (en vez de por filas) para intentar evitar los errores en ráfaga
- Es la base de otros mecanismos de detección y corrección de errores
 - para aumentar el número de errores detectados y/o corregirlos se emplea el **bloque de paridad**, organizado por filas y columnas

MAC – D. Errores de CRC

- **CRC** es un tipo de función que recibe un flujo de datos de cualquier longitud como entrada y devuelve un valor de longitud fija como salida (*checksum*).
- Para un **bloque de k bits**, el *transmisor* genera una secuencia de **r bits**, tal que el número formado por los $k+r$ bits sea divisible por un **número G** de $r+1$ bits convenido en el protocolo (el llamado CRC-...)
 - para ello se divide por G el número formado por los k bits más r 0's a la derecha ($\times 2^r$), siendo el resto de esta división la secuencia de r bits redundante
- El *receptor* divide la trama por el mismo **número G** :
 - si el resto es cero, se asume que no hay error
 - si el resto es distinto de cero, se asume que se ha producido algún error
- La **paridad par** es un caso especial del control de redundancia cíclica (CRC), donde el bit de CRC se genera por el polinomio $x+1$

CRC - Ejemplo de cálculo

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

Generator: 1 0 0 1 1

Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0 0 0

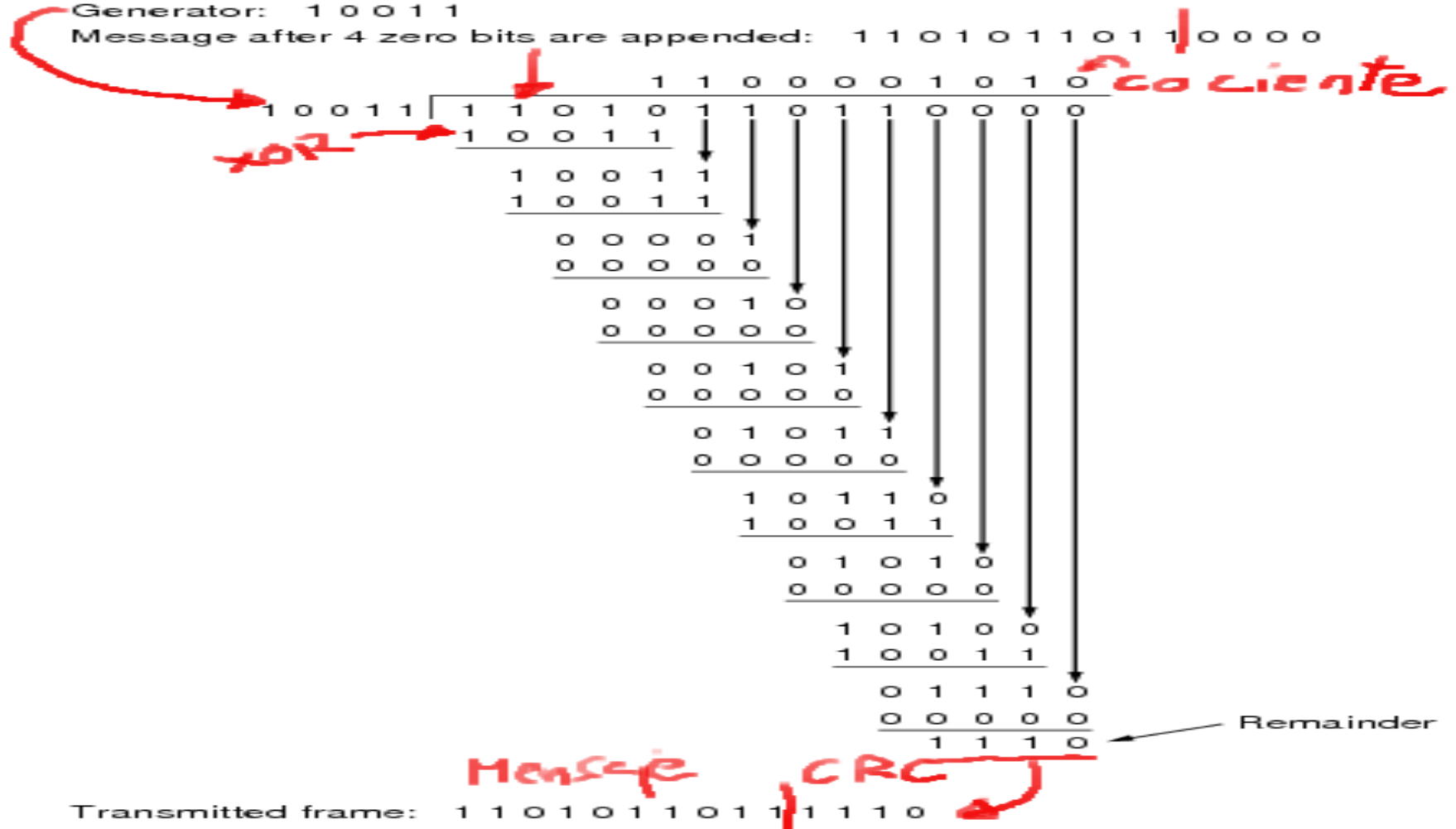
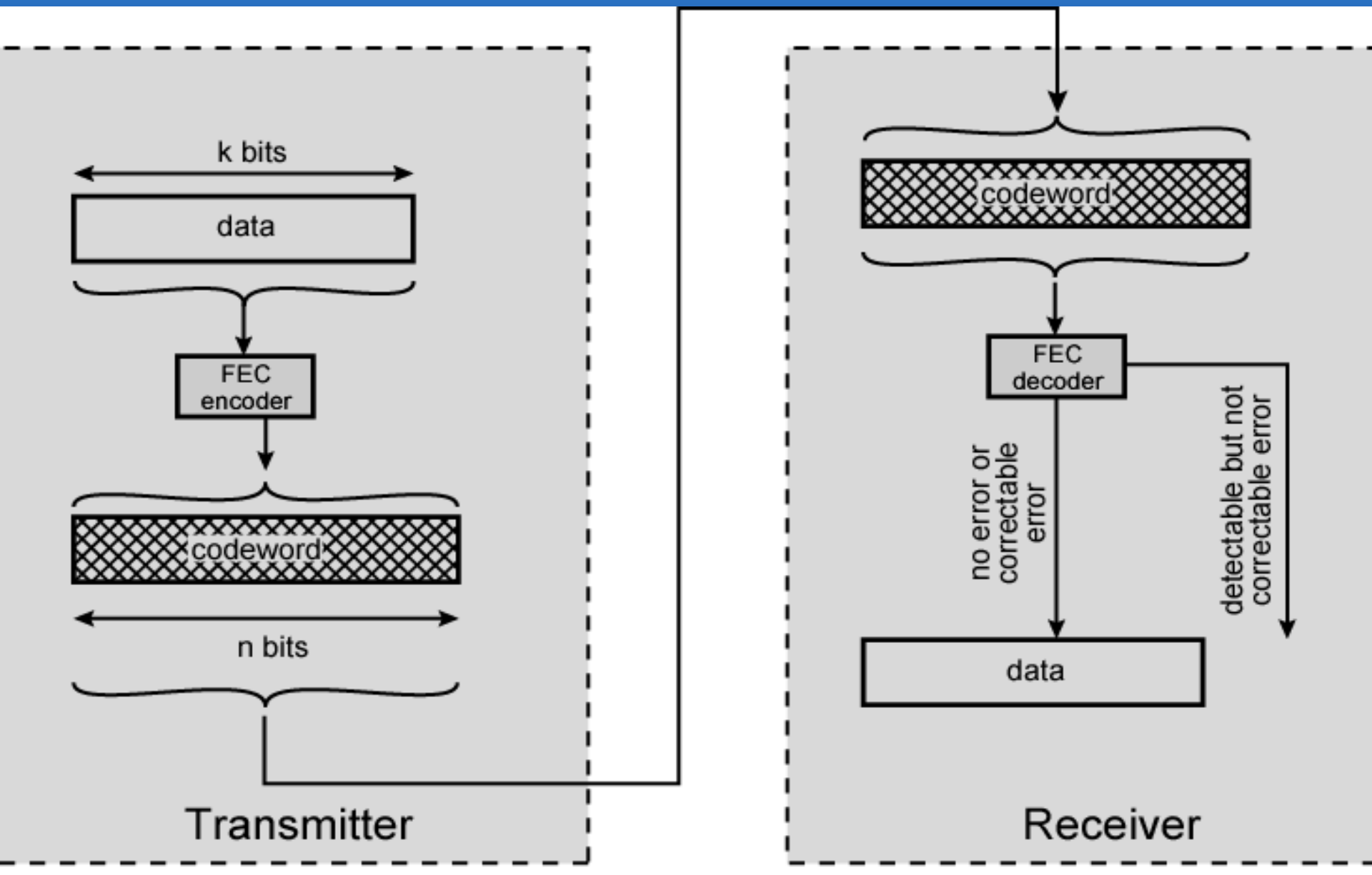


Fig. 3-8. Calculation of the polynomial code checksum.

MAC - Corrección de Errores

- La **corrección de un error** detectado puede requerir que un bloque de datos sea retransmitido: que se haga **hacia atrás (ARQ, Automatic Repeat-reQuest)**. Esta corrección se hace en la **LLC**
- Esto no es muy apropiado cuando:
 - la tasa de error de bit es alta
 - el retraso de propagación puede ser largo (p.e. satélite) comparado con la transmisión
 - resultaría en la retransmisión de la trama errónea y de todas las tramas subsecuentes
- Entonces hay la necesidad de corregir errores a partir de los bits recibidos: que se haga **hacia delante (FEC, Forward Error Correction)**. Esta corrección se hace en la **MAC**

MAC - FEC - Diagrama



MAC - FEC - Proceso

- Cada **bloque de k bits** de datos se mapea a un **bloque de n bits ($n > k$)**. Hay dos formas:
 - **código de bloques**: bloques de longitud fija
 - **código convolucional**: secuencia de bits según se van generando
- Lo recibido se pasa al decodificador FEC:
 - si no hay errores, el bloque original va a la salida
 - algunos patrones de error son detectados y corregidos
 - otros son detectados, pero no corregidos
 - otros pocos patrones de error ni siquiera se detectan
 - resulta en salidas incorrectas de datos del FEC

MAC - FEC - Funcionamiento

- Añade **redundancia** al mensaje transmitido
- **Puede deducirse el original** ante la presencia de cierto nivel de tasa de error
- **Códigos de bloque**
 - añade $(n - k)$ bits al final del bloque
 - da un bloque de n bits (dato-código)
 - todos de los k bits originales incluidos en el dato-código
 - **código Hamming**
 - código por mayoría
 - el mismo bit se envía un número impar de veces
- **Códigos de convolución**
 - transforman símbolos de m bits en símbolos de n bit en función de los k símbolos anteriores, permitiendo la codificación según se generan los datos

Código de Hamming - ejemplo

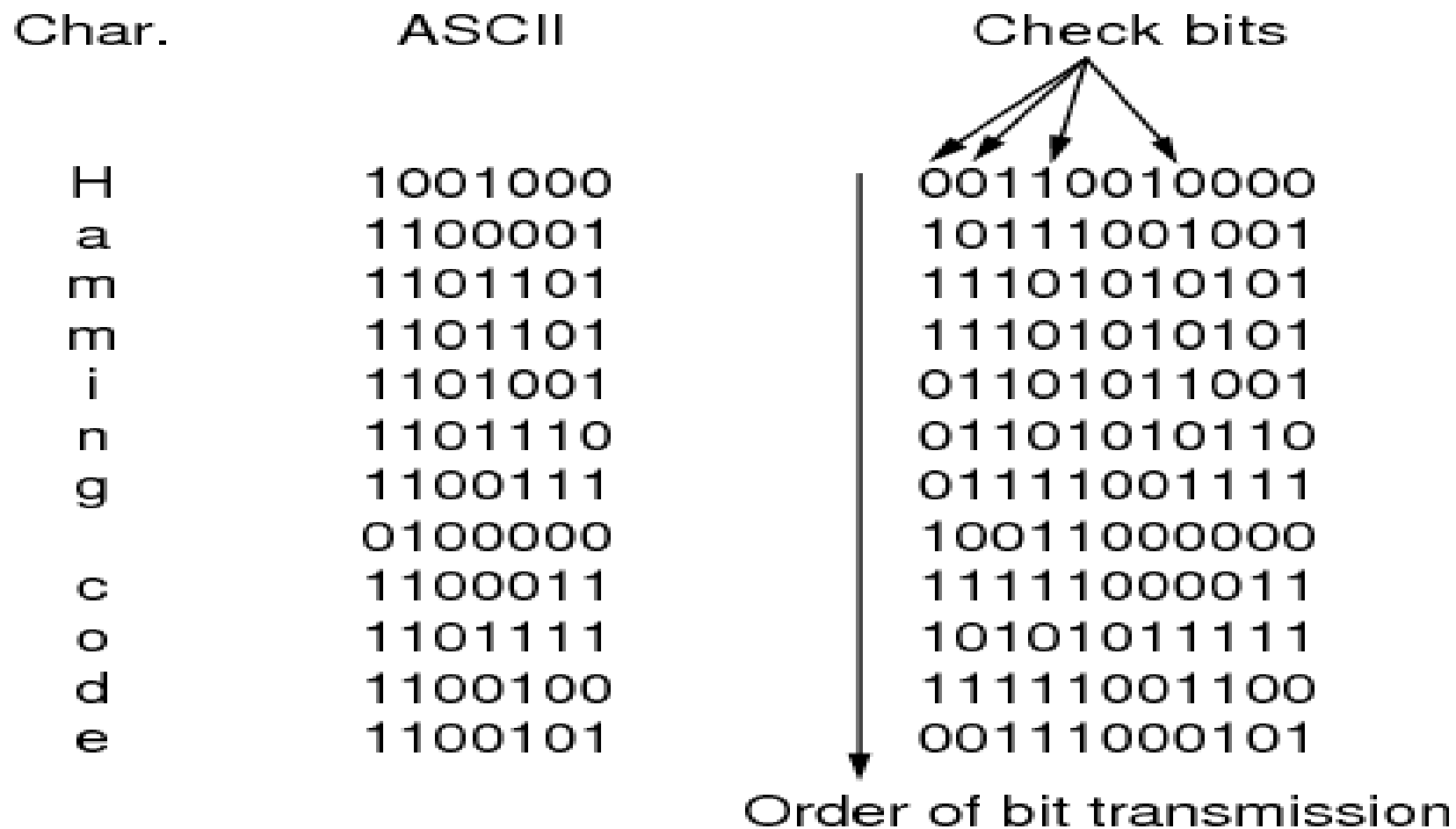


Fig. 3-7. Use of a Hamming code to correct burst errors.

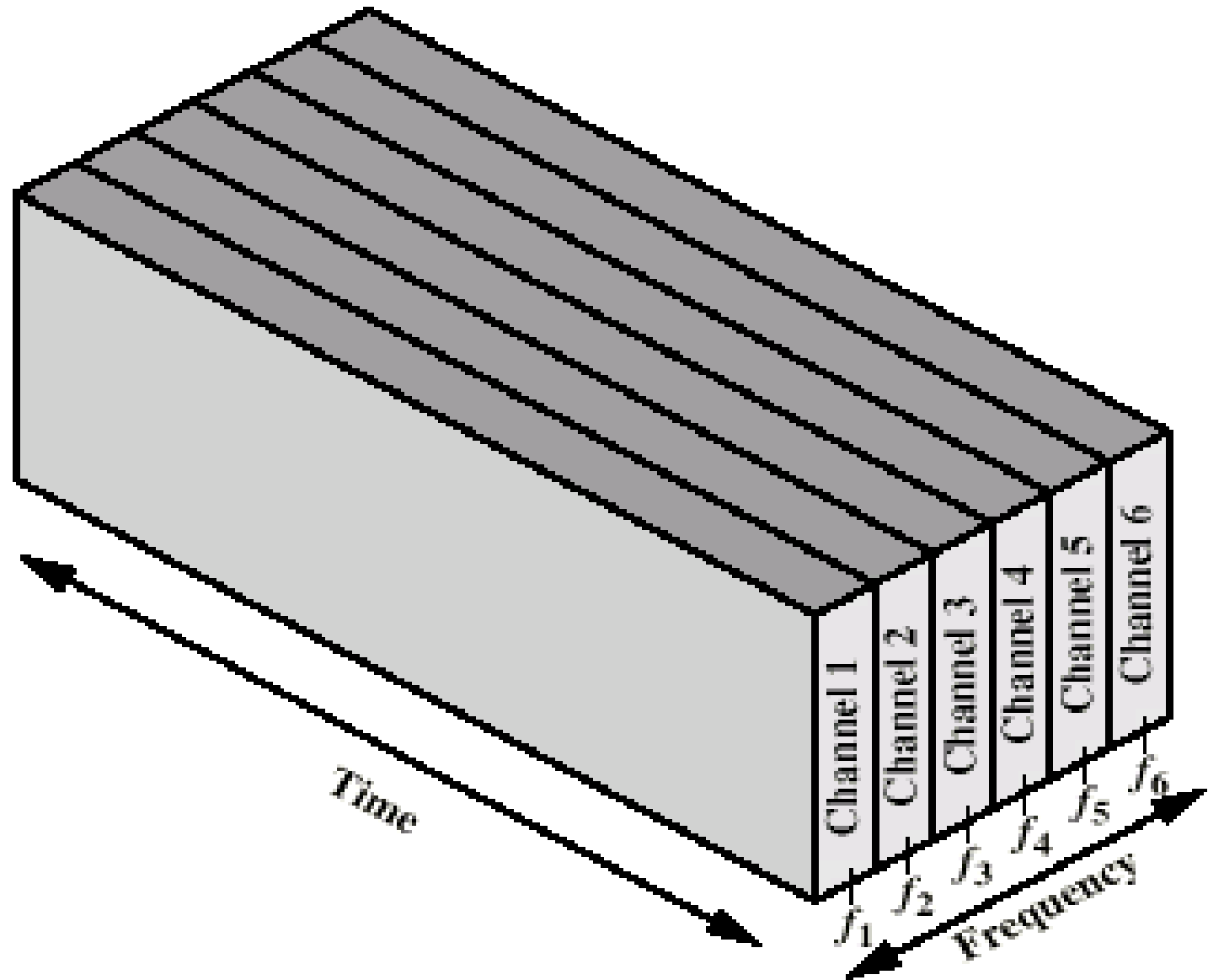
MAC – Asignación del canal

- Determina quién ocupa qué parte de un canal compartido durante cuánto tiempo: **Multiplexión**
- Para compartirlo entre varios usuarios o/y convertirlo en full-duplex o/y adaptarse dinámicamente a las condiciones de transmisión en el canal
- Se puede hacer de dos formas, pudiendo combinarse:
 - **estática**, mediante FDM o TDM fija, presentando problemas de eficiencia con altos y variables números de usuarios (siempre **sin contienda** entre los nodos)
 - **dinámica**,
 - **sin contienda** (p.ej. con reserva), utilizando un dispositivo organizador
 - **con contienda**, no necesitando dicho dispositivo
 - **espectro expandido**

MAC – Estática: FDM

- Multiplexión por División de Frecuencia
- Es posible si el ancho de banda útil del medio excede al requerido por las señales a transmitir
- Sólo puede usarse con señales analógicas para transmitir **datos analógicos o digitales**
 - mediante la **modulación** de los datos en portadoras de diferente frecuencia llamadas **subportadoras**
 - las frecuencias de las subportadoras se separan para que las señales no se solapen (bandas de guardia)
- Ejemplo: radiodifusión

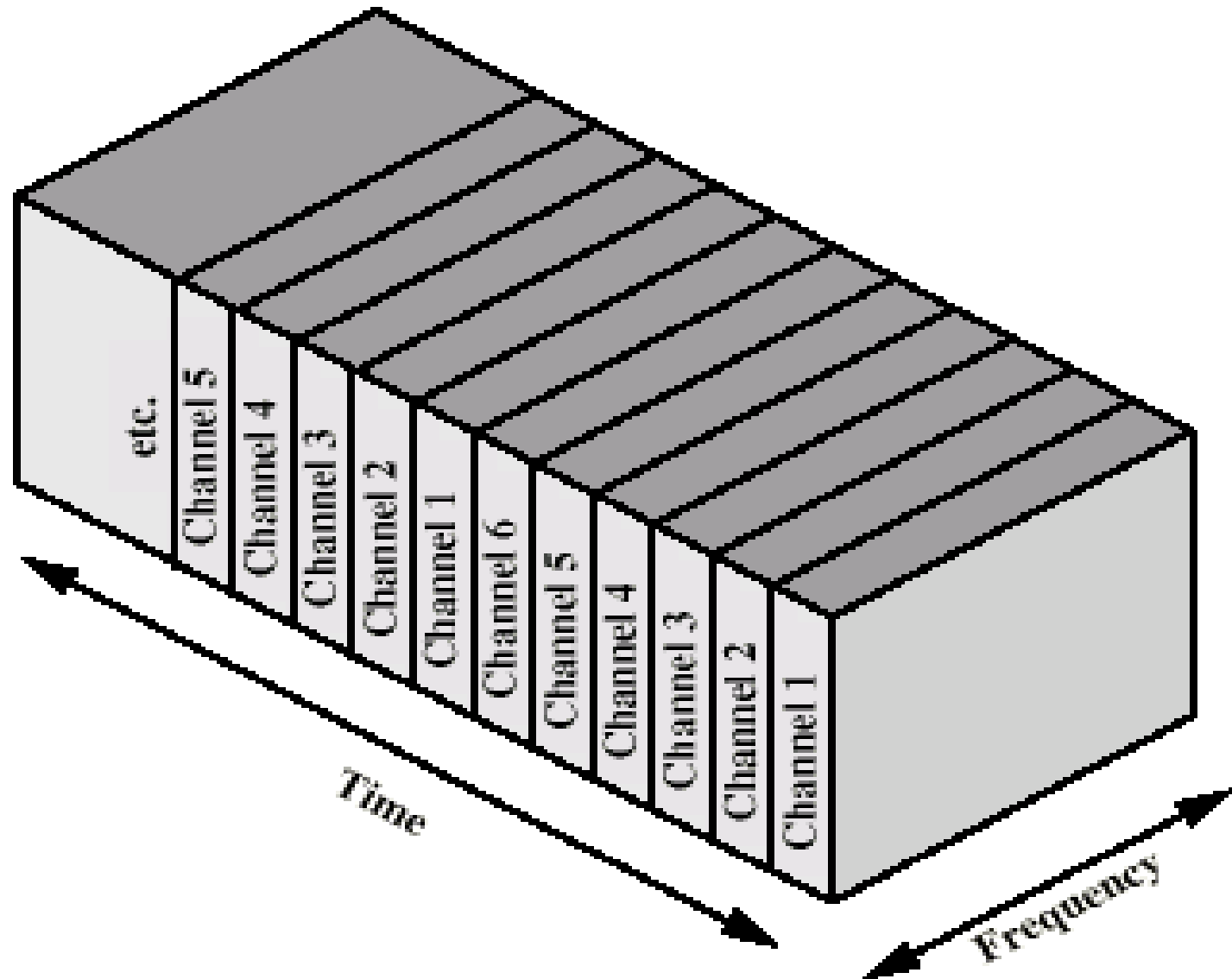
MAC - FDM Diagrama



MAC – Estática: TDM Síncrona

- Multiplexión por División en el Tiempo Síncrona
- Es posible si la tasa de datos del medio excede la tasa de datos de la señal digital a transmitir
- Sólo se puede usar para transmitir **datos digitales** de distintas fuentes entremezcladas en el tiempo
- La mezcla puede ser a nivel de bit o de bloques
 - se necesitan **buffers** para almacenar los datos
- Se preasignan **ranuras** (*slots*) de igual duración a las distintas fuentes incluso si no hay datos
 - aunque se puede asignar distinto número de ranuras
- Las ranuras de tiempo no tienen que ser distribuidas a partes iguales entre fuentes

MAC - TDM Síncrona Diagrama



MAC – Dinámica: Con contienda

- TDM descentralizada y asíncrona
- Cuando se permite que las estaciones transmitan en cuanto tienen datos, de forma que:
 - se pueden producir colisiones de datos
 - las estaciones compiten entre ellas por el acceso al medio de forma directa (sin dispositivos extras)
- Se buscan los siguientes objetivos:
 - añadir nuevos nodos de manera simple
 - evitar complicaciones en la infraestructura de red
 - permitir una razonable tasa de datos
- Todas las estaciones ejecutan el mismo algoritmo para intentar resolver las colisiones

MAC – Dinámica: Sin contienda

- Un sistema rector se encarga de dirigir el tráfico en el medio compartido
- Se buscan los siguientes objetivos:
 - permitir un alto aprovechamiento del medio en condiciones difíciles
 - conseguir un alto control del acceso al medio
- MAC sin contienda:
 - con asignación fija
 - con rotación circular (p.e. *Token ring* o paso de testigo)
 - con conmutación (*switched ethernet*)
 - con reserva

MAC - TDM con reserva

- Reserva previa de un intervalo de tiempo o/y una banda de frecuencia para la estación que quiere transmitir (TDM)
- La reserva puede ser estática o dinámica
- Retardo alto en canales con baja carga de tráfico y viceversa

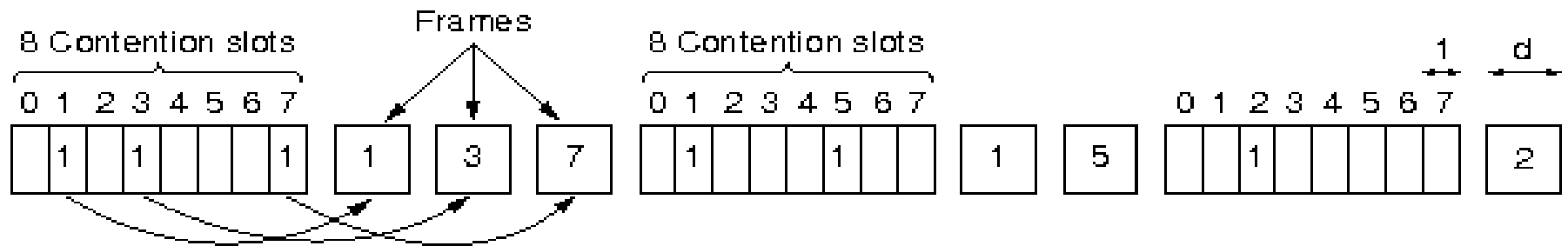


Fig. 4-6. The basic bit-map protocol.