

Códigos

- Código = representación de un alfabeto (conjunto de símbolos)
- Códigos históricos:
 - Morse (telégrafo)
 - Baudot (teletipo) → 5 bits + bit inicio + bit parada
- Códigos modernos
 - EBCDIC (8 bits) → entornos IBM
 - ASCII (7 bits) → normalizado ANSI e ISO

Códigos correctores de error

Añaden información redundante

Permiten detectar y corregir errores

Fundamento

Distancia de Hamming = n° de bits que cambian entre dos combinaciones

Distancia de Hamming de un código = distancia mínima

Un error de n bits es

 Detectable por un código con distancia n

 Corregible por un código de distancia $2n + 1$

- Códigos m sobre n
 - Son códigos de m bits
 - Sólo son válidas las combinaciones que tienen n bits a 1
 - Distancia de Hamming = 2
- Control de paridad
 - Se añade un bit de paridad
 - Distancia de Hamming = 2
 - Paridad horizontal = para cada dato transmitido
 - Paridad vertical = para todos los bits de una secuencia de datos (columnas)
 - Paridad cruzada = combinación de las dos → distancia de Hamming = 4

Códigos cíclicos (CRC)

Detectan ráfagas de errores

Tratamiento de las series de bits como polinomios

Utilizan un polinomio generador para la comprobación de errores

Proceso:

- Se añaden al dato a transmitir tantos ceros a la derecha como el orden del polinomio generador
- Se divide el polinomio resultante por el polinomio generador y se obtiene el resto
- El resto se suma al dato a transmitir expandido con los ceros

El receptor divide el dato que le llega por el polinomio generador,

- Si el resto es 0 no hay error
- Si el resto no es 0 hay errores

Polinomios cíclicos más usados

- CRC-12 = $x^{12} + x^{11} + x^3 + x^2 + x + 1$
- CRC-16 = $x^{16} + x^{15} + x^2 + 1$
- CRC-CCITT = $x^{16} + x^{12} + x^5 + 1$

• Características de los CRC16

- Detecta 100% errores simples, y dobles
- Detecta 100% errores en un número impar de bits
- Detecta 100% de los paquetes con errores de longitud menor que 18 y 99'998% de los mayores

Protocolos de comunicación

- Protocolo = conjunto de reglas que hacen posible la comunicación
- Funciones de un protocolo
 - Establecimiento y fin de la comunicación
 - Sincronización de la comunicación, a nivel de tramas, de palabras y de bits
 - Control de flujo
 - Detección y corrección de errores

Clasificación de los protocolos

- Según el control sobre el medio
 - Balanceados o simétricos:
 - los dos extremos trabajan igual.
 - Cada uno puede tomar la iniciativa de la comunicación
 - No balanceados a asimétricos
 - Una estación primaria (maestra) y las demás secundarias (esclavas)
 - La estación primaria emite y/o da turnos de palabra para emitir
 - La estación secundaria recibe o espera su turno para emitir
 - Híbridos

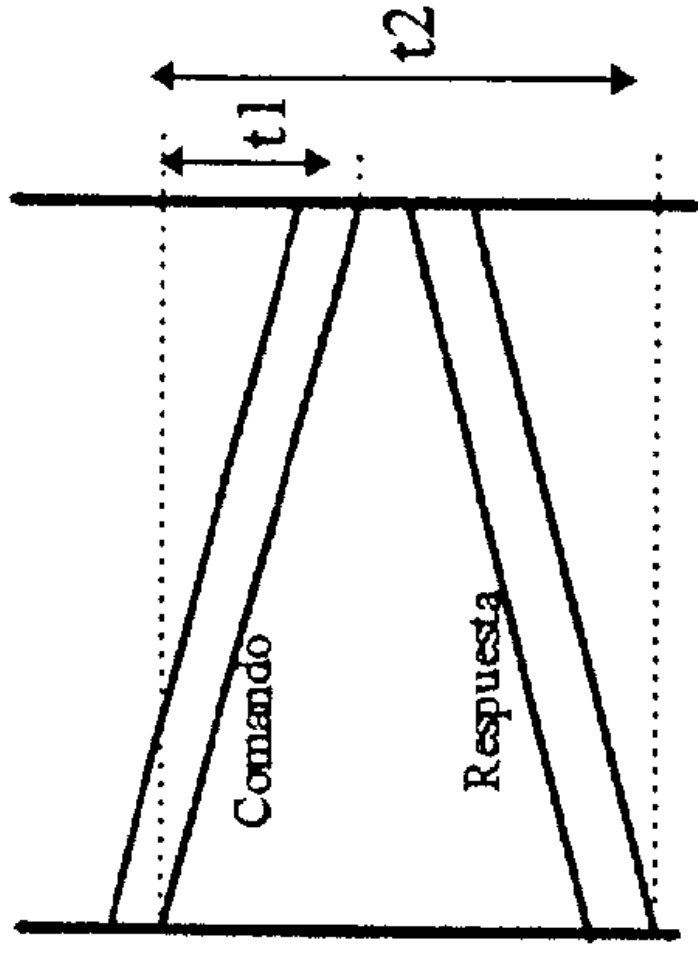
- **Según utilicen o no sondeo**
 - **Protocolos de sondeo–selección**
 - **Sondeo = la estación primaria pide información a la secundaria**
 - **Selección = la estación primaria envía información a la estación secundaria**
 - **El proceso se controla con señales:**
 - **Sondeo = petición de información**
 - **Selección = aviso de envío de información**
 - **ACK = validación**
 - **NAK = no validación**
 - **EOT = fin de transmisión**
- **Protocolos sin sondeo: no realizan sondeo**
 - **Control de flujo hardware: RTS/CTS**
 - **Control de flujo software: XON/XOFF**

- Según utilicen o no prioridades
 - Sistemas sin prioridad
 - MUX–MDT (Multiplex por división en el tiempo)
 - El canal se divide en intervalos de tiempo
 - Se asigna un intervalo a cada estación
 - CSMA/CD (acceso múltiple por detección de portadora y detección de colisiones)
 - Todas las estaciones pueden utilizar el canal cuando está libre
 - Una estación escucha a ver si el canal está libre, y si está libre transmite
 - Si dos estaciones empiezan a emitir a la vez se produce una colisión. Cada estación corta el envío y espera un tiempo aleatorio antes de empezar a enviar de nuevo
 - El rendimiento se degrada en sistemas con mucho tráfico por el aumento de las colisiones
 - Paso de testigo
 - Se transmite por la red un testigo
 - Sólo la estación que tiene el testigo puede transmitir

- **Sistemas con prioridad**
 - **CSMA/CD con prioridad**
 - El tiempo de espera después de una colisión no es aleatorio sino que se fija para cada estación, menor cuanto mayor sea la prioridad de la estación
 - **Paso de testigo con prioridad**
 - El paso del testigo no se hace por turnos, sino que se puede reservar por las estaciones según su prioridad
- **Protocolos de ventana deslizante**
 - En protocolos normales el canal permanece sin utilizar mientras se espera la validación del receptor
 - Los protocolos de ventana deslizante permiten enviar varias tramas sin esperar validación y validarlas luego todas a la vez
 - Llevan un contador de tramas transmitidas

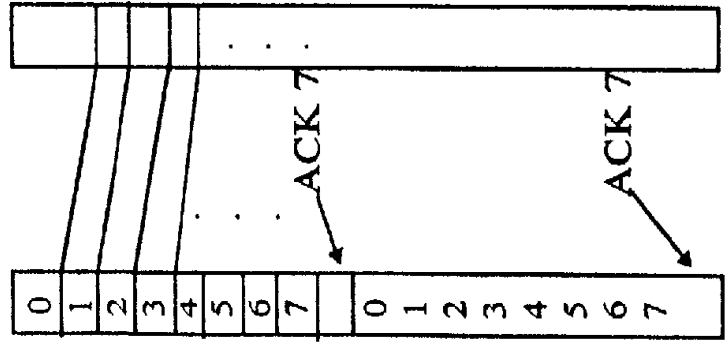
Desperdicio de tiempo de canal en protocolos de parada y espera

ETD canal ETD

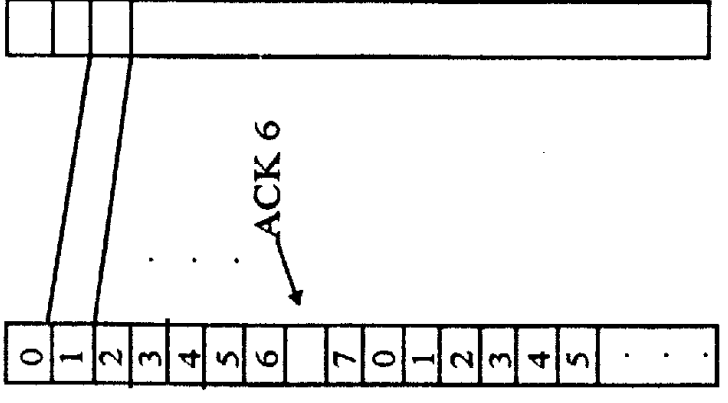


Protocolos con ventana deslizante

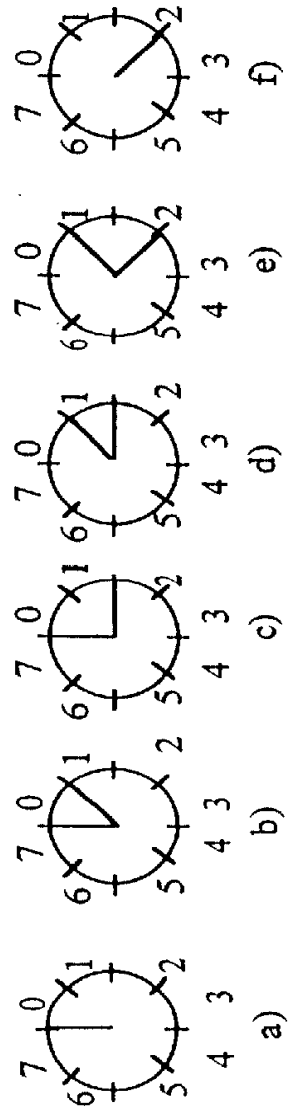
EMISOR RECEPTOR



EMISOR RECEPTOR

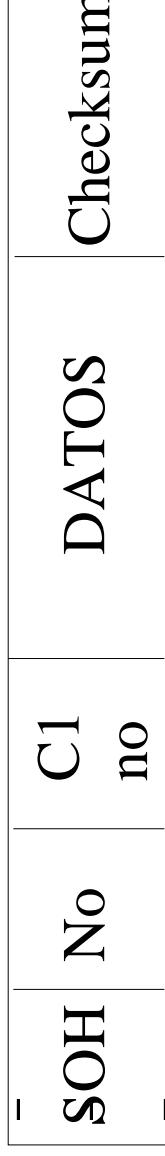


E



Protocolo XMODEM

- Protocolo para transferencia de archivos entre PC's a través del módem
- Protocolo de parada y espera, serie, asíncrono
- Tramas de longitud fija:

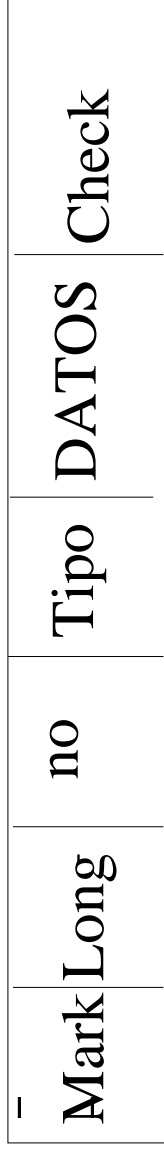


- SOH = cabecera (carácter 1 ASCII)
- no (1byte)= número de secuencia del paquete
- C1 no (1byte)= n° secuencia en complemento a 1
- DATOS (128 bytes)
- Checsum (1 byte) = suma de todos los bytes de datos

- Inicio de la transmisión → receptor envía ACK indicando que está preparado para recibir
- Transmisión:
 - El emisor envía un dato
 - Si el receptor lo recibe bien envía ACK
 - Si hay error de secuencia en vía CAN → corta la transmisión
 - Si hay otro error envía NACK → el emisor reenvía la trama
- Fín de la transmisión → el emisor envía EOT

Protocolo kermi

- Protocolo para transferencia de archivos entre ordenadores (no PCs) a través del módem
- Protocolo de parada y espera, serie, asíncrono
- Tramas de longitud variable:



- mark (1 byte) = cabecera (secuencia irrepitible)
- long (1 byte) = longitud de la trama
- no (1byte)= número de secuencia de la trama
- tipo = tipo de trama
- DATOS (longitud variable)
- Check (1,2,3 byte) = puede ser check o CRC

- Permite la transferencia de archivos entre diferentes sistemas
- Sólo presupone que los sistemas son capaces de enviar caracteres imprimibles (20h–7Fh ASCII)
- Los "códigos de control" son tramas en vez de caracteres
- Tramas de longitud variable
- El protocolo incluye el nombre del fichero
- Permite negociar parámetros de la comunicación
- Permite versiones de ventana deslizante (nº de secuencia en tramas ACK y NACK)
- Permite transferir múltiples ficheros

Protocolo BSC

- Protocolo orientado a caracter, síncrono, punto a punto o multipunto
- Utilizado en entornos IBM
- Permite el empleo de ASCII, EBCDIC, SBC o modo transparente (cualquier combinación binaria)
- Protección contra errores: paridad, CRC-12, CRC-16

Protocolo HDLC

- Protocolo orientado a bit, síncrono, punto a punto o multipunto, de ventana deslizante.
- Estandar ISO.
- Permite explotación duplex del enlace.
- Permite la transmisión de cualquier tipo de datos.
- Permite enlaces equilibrados y no equilibrados.
- Trama:

- Trama

Band	Dir	Cntrl	DATOS	FCS	Band
------	-----	-------	-------	-----	------

- Bandera = 01111110
- Dirección (8bits) = identifica estación (multipunto)
- Control (8bits) = tipo de trama, etc
- DATOS = cualquier número de bits
- FCS (16 bits) = control de errores