

La fuente original de gran parte de las imágenes presentadas en esta lección son cortesía del texto docente "Redes y Transmisión de Datos" P. Gil, J. Pomares, F. Candelas. Servicio de Publicaciones Universidad de Alicante.

Redes (9359). Curso 2010-11

Ingeniería Técnica en Informática de Sistemas (plan 2001)



Universitat d'Alacant
Universidad de Alicante

Departament de Física, Enginyeria de Sistemes i Teoria del Senyal
Departamento de Física, Ingeniería de Sistemas y Teoría de la Señal

Pablo Gil Vázquez (Pablo.Gil@ua.es)

Grupo de Innovación Educativa en Automática

© 2010GITE – IEA



Redes (9359). Curso 2010-11

Ingeniería Técnica en Informática de Sistemas (plan 2001)

BLOQUE I: INTRODUCCIÓN.

Tema 1: Introducción a las redes de computadores.

BLOQUE II. NIVEL FÍSICO.

Tema 2: Transmisión de señales.

Tema 3: Codificación de la información.

Tema 4: Medios de transmisión.

BLOQUE III. NIVEL DE ENLACE.

Tema 5: Diseño del nivel de enlace y control de errores.

Tema 6: Control de flujo en el nivel de enlace.

Tema 7: Protocolos estandarizados del nivel de enlace.

BLOQUE IV. NIVEL DE RED.

Tema 8: Diseño del nivel de red.

Tema 9: Encaminamiento y control de congestión del nivel de red.



Universitat d'Alacant
Universidad de Alicante

Departament de Física, Enginyeria de Sistemes i Teoria del Senyal
Departamento de Física, Ingeniería de Sistemas y Teoría de la Señal

Pablo Gil Vázquez (Pablo.Gil@ua.es)

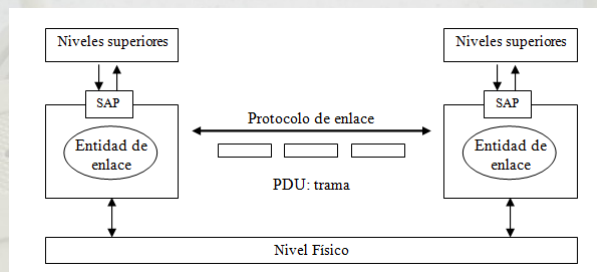
Grupo de Innovación Educativa en Automática

© 2010GITE – IEA



1. Servicios del nivel de enlace.
2. Funciones del nivel de enlace.
3. Iniciación y terminación.
4. Delimitación de tramas.
5. Direccionamiento de tramas.
6. Detección y corrección de errores en tramas.
 - Códigos de redundancia cíclica (CRC).
 - Códigos Hamming.
 - Códigos Reed-Solomon
7. Medición de la tasa de errores.

- El nivel de enlace proporciona al nivel de red diferentes servicios de acuerdo a la calidad que se quiere conseguir en la transmisión.



- Sin conexión y sin reconocimiento:
 - Sin confirmación de recepción correcta de datos.
 - Sin establecimiento de la conexión.
 - Se emplea en líneas de comunicación con tasa de errores bajas.
 - Se emplea en medios físicos donde los retardos perjudican mas que los errores.
- Sin conexión y con reconocimiento.
 - Se confirma la recepción de datos mediante tramas de asentimiento (ACKs).
 - Sin establecimiento de la conexión.
 - Hay reenvío de tramas ante errores.
 - Se usa en redes inalámbricas ya que se busca comunicación fiable.
- Con conexión y con reconocimiento:
 - Hay primitivas de establecimiento y liberación de conexión.
 - Se usan tramas numeradas. Estas serán confirmadas por receptor y si hay errores se producirá reenvío de datos.
 - Se usa en comunicaciones donde prima la fiabilidad.

1. Servicios del nivel de enlace.
2. Funciones del nivel de enlace.
3. Iniciación y terminación.
4. Delimitación de tramas.
5. Direccionamiento de tramas.
6. Detección y corrección de errores en tramas.
 - Códigos de redundancia cíclica (CRC).
 - Códigos Hamming.
 - Códigos Reed-Solomon
7. Medición de la tasa de errores.

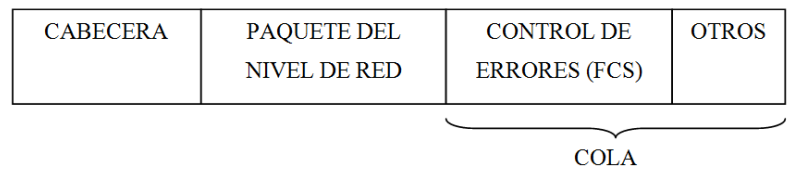
- Para proporcionar ese conjunto de servicios, el nivel de enlace tiene una serie de funciones asignadas.
 - Iniciación, terminación de la comunicación y coordinación.
 - Activar, mantener y liberar el enlace de comunicaciones. Gestión intercambio.
 - Delimitación de tramas.
 - Agrupar datos en tramas. Identificar inicio y fin de la trama de datos.
 - Direccionamiento de tramas.
 - Identificar el origen y el destino de la trama de datos.
 - Control de errores.
 - Asegurar la transmisión sin errores producidos por ruido o atenuaciones del medio físico. Corregir o al menos detectar esos errores.
 - Control de flujo.
 - Evitar saturaciones cuando hay receptores lentos. Adecuar velocidades al destino más lento

1. Servicios del nivel de enlace.
2. Funciones del nivel de enlace.
3. Iniciación y terminación.
4. Delimitación de tramas.
5. Direccionamiento de tramas.
6. Detección y corrección de errores en tramas.
 - Códigos de redundancia cíclica (CRC).
 - Códigos Hamming.
 - Códigos Reed-Solomon
7. Medición de la tasa de errores.

- La función de *iniciación* permite activar el enlace, sincronizar envío-recepción y la *terminación* liberar recursos.
 - Conexión del circuito.
 - Obtener un circuito físico que conecte las estaciones.
 - Establecimiento del enlace lógico.
 - Consiste en determinar las primitivas para poder realizar la transferencia de datos a través del circuito físico establecido (comprobar si se está preparado para recibir, si se dispone de los recursos necesarios, etc.).
 - Transferencia de datos.
 - Comprende todos los procesos necesarios para lograr la transferencia de mensajes de una estación a otra.
 - Terminación.
 - Fase de liberación de los recursos asociados a la transferencia.
 - Desconexión
 - Comprende los procesos de liberación del circuito

1. Servicios del nivel de enlace.
2. Funciones del nivel de enlace.
3. Iniciación y terminación.
4. Delimitación de tramas.
5. Direccionamiento de tramas.
6. Detección y corrección de errores en tramas.
 - Códigos de redundancia cíclica (CRC).
 - Códigos Hamming.
 - Códigos Reed-Solomon
7. Medición de la tasa de errores.

- Formato de trama:



- Cabecera:**

- Identifica el comienzo de la transmisión y delimita el inicio de la trama.
 - Especifican las direcciones de las estaciones origen y destino.
 - Puede especificarse otra información de control como: código identificador de la trama, el tipo de trama (si es de datos, confirmación, u otro tipo de trama de control) o longitud de la trama.

- Paquete red.**

- En este campo se encapsula el paquete generado a nivel de red.
 - Puede ocurrir que el nivel de enlace envíe tramas propias de control (por ejemplo, ACK) que no tienen encapsuladas información del nivel de red.

- Cola:**

- Especifica el final de la trama.
 - FCS: Campo para verificar si la transmisión se ha hecho correctamente.

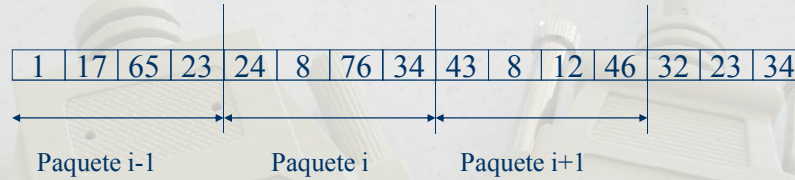
- En terminología OSI, la cabecera, y la cola forma la PCI (Información de Control de Protocolo), el paquete de red es la SDU (Unidad de Datos de Usuario) y todo el conjunto de la trama es la PDU (Unidad de Datos de Protocolo).

- Tipos de delimitación:

- Delimitación temporal.
 - Por numeración de caracteres.
 - Por caracteres especiales
 - Por bits especiales.
 - Por códigos de línea.

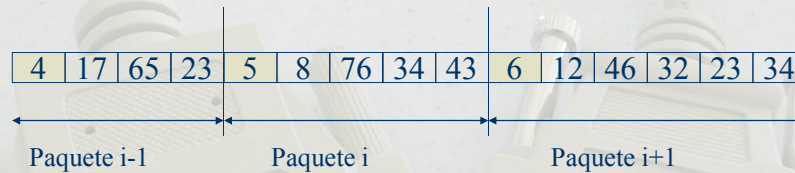
▪ Delimitación temporal.

- No se emplean cabeceras ya que se supone todas las tramas del mismo tamaño.
- El tamaño se asigna en función de una estimación de tiempo que tarda la trama en llegar al receptor.
- Este método es sensible a los retardos que hace que se introduzcan errores.



▪ Delimitación por numeración de caracteres.

- Se emplean cabeceras.
- La cabecera indica el número de bytes que posee la trama.
- Este método es sensible al ruido, si hay variaciones en los datos de la cabecera se pierde la sincronización.



▪ Delimitación por caracteres especiales.

- Se emplean cabeceras y colas.
- La cabecera/cola almacena caracteres especiales que no se emplean en los datos transmitidos.
- Este método se emplea cuando los datos transmitidos son caracteres de texto.
- Si algún carácter especial se usa como dato, se usa un carácter de relleno que no es más que el carácter duplicado

DLE|STX| A | B | C | D | F | G | DLE|DLE| C | F | H | DLE|ETX

▪ Delimitación por bits especiales

- Se emplean cabeceras y colas.
- La cabecera/cola almacena una secuencia de bits única.
- Para evitar que la secuencia de bits especial pueda aparecer como datos se usan 'bits de relleno'.

▪ Ejemplo:

*Datos: '1111111111101111101111110'

Transmitir:

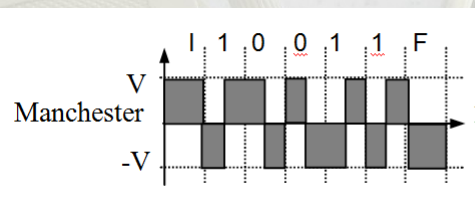
'01111110111110111110111101111010111110100111110'

Cabecera

Bit de relleno

Cola

- Delimitación por códigos de línea
 - Se delimita el inicio y final de la trama empleando codificaciones distintas a las empleadas para codificar el resto de bits de la trama.
 - Ejemplo: Trama de datos cuyos bits se codifican en Manchester y el inicio y fin se presentan con una codificación distinta al resto de bits ya que no presentan una transición sino que el bit de inicio se codifica con un estado alto y el bit de final con un estado bajo.



1. Servicios del nivel de enlace.
2. Funciones del nivel de enlace.
3. Iniciación y terminación.
4. Delimitación de tramas.
5. Direccionamiento de tramas.
6. Detección y corrección de errores en tramas.
 - Códigos de redundancia cíclica (CRC).
 - Códigos Hamming.
 - Códigos Reed-Solomon
7. Medición de la tasa de errores.

- Consiste en identificar las estaciones o extremos de la comunicación que intercambian la información.
- En función del origen y el destino
 - Implícito:
 - No es necesario especificar las direcciones de las estaciones origen y destino.
 - Se emplea en conexiones punto a punto.
 - Explícito:
 - Se indica la dirección origen y la destino en la trama. Cada equipo conectado a la red presenta una dirección única.
 - Se emplea en redes de difusión y en redes multipunto donde existen varias estaciones en la misma red que pueden ser el destino.

- En función de cómo están conectadas las estaciones al canal.
 - Preselección
 - Existe un dispositivo central que se encarga de seleccionar secuencialmente cada una de las estaciones de destino accesibles desde una estación
 - Master único
 - Una estación central denominada master gestiona las comunicaciones.
 - Sólo el equipo master tiene iniciativa de enviar datos.
 - El resto de estaciones se le denominan esclavos y se encargan únicamente de responder a las peticiones del master.
 - Master múltiple
 - Este caso es similar al anterior pero pueden existir varios equipos master en la red.
- Tipos de direcciones especiales.
 - Multicast
 - El destino son un subconjunto de los equipos que componen la red.
 - Broadcast
 - El destino son todos los equipos de la red.

1. Servicios del nivel de enlace.
2. Funciones del nivel de enlace.
3. Iniciación y terminación.
4. Delimitación de tramas.
5. Direccionamiento de tramas.
6. Detección y corrección de errores en tramas.
 - Códigos de redundancia cíclica (CRC).
 - Códigos Hamming.
 - Códigos Reed-Solomon
7. Medición de la tasa de errores.

- **Qué es un Error**
 - Cuando el valor de un bit enviado por el receptor es alterado durante la transmisión.
- **Causas de errores en medios de transmisión**
 - Ruidos e interferencias externas que reduzcan la relación señal-ruido.
- **Tipos de errores**
 - **Aislados**
 - Cuando los bits erróneos son independientes de los demás bits de la trama (vecinos).
 - **A ráfagas**
 - Los bits erróneos se encuentran secuencialmente en el tiempo, además de estar agrupados. Es lo que ocurre habitualmente cuando la señal con los datos se deteriora.

▪ **Técnicas para tratamiento de errores:**

- **Detección: (sólo detectar errores)**
 - Comprobar que los datos recibidos son realmente los que se quiso enviar.
 - Se implementa mediante la técnica del reenvío de tramas.
 - Se emplean cuando el tiempo de reenvío es inferior al tiempo de chequeo y corrección.
- **Corrección: (detectar y corregir errores)**
 - Consiste en modificar los datos que han sido alterados durante la transmisión.
 - Se emplea en medios físicos dónde el reenvío produce retardos elevados.

▪ **El control de errores se lleva a cabo tanto en emisor como en receptor.**

- Emisor: Añade en la cola de la trama información para el control de errores (campo FCS-> Secuencia de verificación de trama).
- Receptor: Maneja e interpreta la información recibida.

▪ **Para medir errores en un medio físico se usa el BER.**

- Sus valores típicos están comprendidos entre 10^{-3} y 10^{-6}

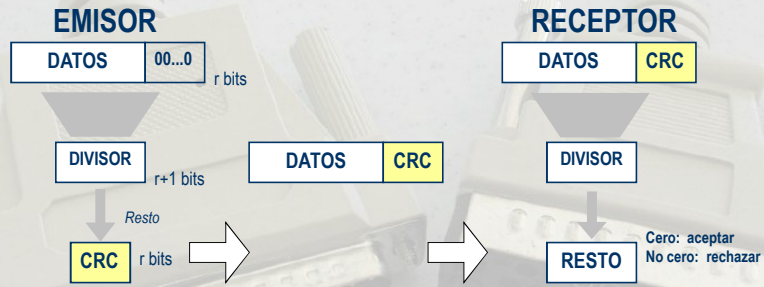
▪ **Factores que afectan a la *tasa de error* del medio.**

- Tipo del medio físico empleado.
- Entorno del medio.
- Velocidad de transmisión.
- Calidad del servicio del medio físico.
- Horario en que se realiza la comunicación.

$$T_{error} = BER = \frac{n^{\circ} errores}{n^{\circ} bits transmitidos}$$

▪ Códigos de redundancia cíclica (C.R.C.)

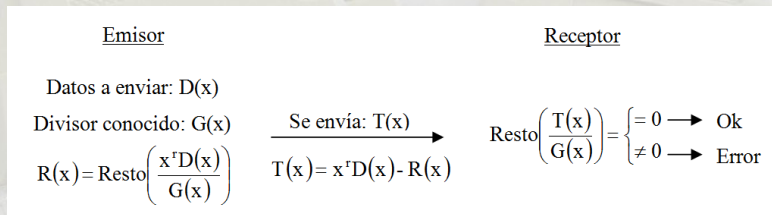
- Se basa en la división binaria e inclusión de un resto en la cola de la trama de datos.
- Un CRC es una secuencia de bits que se colocan en la cola de la trama, concretamente en el campo FCS de una trama del nivel de enlace.



▪ Códigos de redundancia cíclica (C.R.C.)

▪ Proceso de CRC

- Para crear el resto CRC es necesario disponer de un divisor CRC, $G(x)$.
- A los datos, $D(x)$, que se quieren transmitir se le añaden r ceros por la derecha, siendo $r+1$ la longitud del divisor CRC.
- Se dividen los datos + 'ceros', $D(x) \cdot x^r$, por el divisor, $G(x)$. Se busca el resto, $R(x)$. Dónde r es el grado del polinomio del generador.
- El resto de la división $R(x)$ es el CRC.
- El CRC sustituye a los 'ceros' y viaja con los datos $T(x)$, hasta llegar al receptor.
- En el receptor se dividen los datos + CRC, $T(x)$, y se calcula el nuevo resto, $R'(x)$.
- Si resto, $R'(x)$, cero: trama sin error. Si resto, $R'(x)$, no cero: trama con error.



▪ Problemas de cálculo de CRC

- EJEMPLO 1:
 - Calcular el marco de datos a enviar $T(x)$ si el emisor quiere enviar la cadena de datos '10100111' usando como polinomio generador $G(x) = x^2 + x + 1$. Y a continuación comprobar si se han producido errores en los datos recibidos por el receptor
- EJEMPLO 2:
 - Calcular el marco de datos a enviar $T(x)$ si el emisor quiere enviar la cadena de datos '1101011011' usando como polinomio generador $G(x) = x^4 + x + 1$. Y a continuación comprobar si se han producido errores en los datos recibidos por el receptor.

▪ Ejemplo 1

- Paso 1: Calcular $D(x)x^r$, para ello añadir tantos ceros como grado r tiene el polinomio generador $G(x)$.

$$G(x) = x^2 + x + 1 \rightarrow '111' \Rightarrow r = 2^\circ \text{ grado}$$

$$D(x) = '10100111'$$

$$D(x) x^r$$

$$1010011100$$

▪ Ejemplo 1

- Paso 4: Comprobación en el receptor. Para realizar dicha comprobación se divide $T(x)$ por $G(x)$ y se observa el resto.



$R'(x)$
No se han producido errores.

▪ Ejemplo 2

- Paso 1: Calcular $D(x)x^r$, para ello añadir tantos ceros como grado r tiene el polinomio generador $G(x)$.

$G(x) = x^4 + x + 1 \rightarrow '10011' \Rightarrow r = 4^\circ \text{ grado}$

$D(x) = '1101011011'$

$1101011011\underbrace{0000}$

$D(x) x^r$

▪ Ejemplo 2

- Paso 2: Calcular $R(x)$, para ello dividir la cadena resultante del paso 1 por $G(x)$ usando la división en módulo 2.

$$\begin{array}{r}
 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0 \quad | \quad 1\ 0\ 0\ 1\ 1 \\
 \underline{1\ 0\ 0\ 1\ 1} \\
 0\ 1\ 0\ 0\ 1\ 1 \\
 \underline{1\ 0\ 0\ 1\ 1} \\
 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0 \\
 \underline{1\ 0\ 0\ 1\ 1} \\
 0\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
 \underline{1\ 0\ 0\ 1\ 1} \\
 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0
 \end{array}$$

$R(x)$

▪ Ejemplo 2

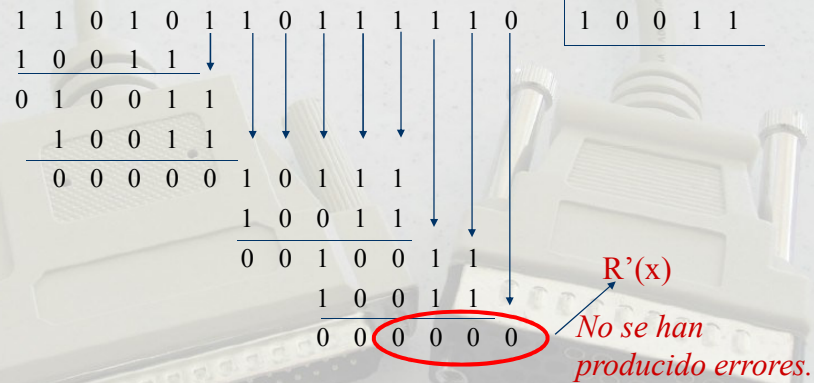
- Paso 3: Calcular $T(x)$, para ello, a la cadena resultante del paso 1 se le subtrae el resto $R(x)$ obtenido de calcular la división del paso 2.

$$\begin{array}{r}
 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0 \\
 \underline{} \\
 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0
 \end{array}$$

$T(x)$

▪ Ejemplo 2

- Paso 4: Comprobación en el receptor. Para realizar dicha comprobación se divide $T(x)$ por $G(x)$ y se observa el resto.



▪ Corrección de errores empleando códigos de bloque:

- Consiste en dada una secuencia de bits datos llamada bloque de información, dividir éste en bloques y a cada uno de ellos añadirle un cierto número de símbolos de redundancia.
- Estos símbolos adicionales contienen la información necesaria y suficiente para poder localizar la posición de los símbolos de datos erróneos y corregirlos.
- No emplean campo FCS
- Los símbolos de redundancia se intercalan entre los de datos.
- El receptor tendrá que separar los bits de datos de la información adicional intercalada para enviar sólo al nivel de red la parte de datos.

▪ Tipos:

- Códigos Hamming.
- Códigos Reed-Solomon (RS)

▪ Códigos de Hamming

▪ Conceptos previos

- Distancia Hamming.
 - La **distancia Hamming entre dos palabras** de código Hamming se define como el número de bits en que difieren las dos palabras.
 - La **distancia Hamming de un código Hamming** es la menor de todas las distancias de Hamming entre todas las palabras del código.

Regla:

Dado un código binario de distancia Hamming d es posible detectar errores de $d-1$ bits y corregir errores de $(d-1)/2$ bits

▪ Códigos de Hamming

▪ Proceso para la obtención del código Hamming

- Si se quiere transmitir m bits de datos se necesitarán transmitir 2^m palabras de datos.
- A cada una de las 2^m palabras de datos hay que añadirles r bits redundantes como información adicional para la detección y corrección. Así, se enviarán 2^{m+r} palabras de nuevos datos, que formaran lo que se llama código de palabras.

$$m+r+1 \leq 2^r$$

- La posición que debe ocupar cada bit de redundancia vendrá determinada por las potencias de 2.

Problemas de cálculo de Códigos Hamming

- EJEMPLO 1:
 - Obtener el código Hamming con paridad par asociado a un conjunto de palabras de datos de 2 bits.
- EJEMPLO 2:
 - Dado un determinado código Hamming verificar si un determinado subconjunto de palabras son o no son correctas y en caso de que no lo sean, corregir el error que se ha producido durante la transmisión.

Ejemplo 1

- Paso 1: Número de bits redundantes necesarios ($m+r+1 \leq 2^r$)
 - Si $m=2 \rightarrow$ supóngase $r=1 \rightarrow 2+1+1 \leq 2$ falso.
 - $r=2 \rightarrow 2+2+1 \leq 4$ falso
 - $r=3 \rightarrow 2+3+1 \leq 8$ cierto
- Paso 2: La longitud de las palabras a enviar ($m+r$).
 - Si $m=2$ y $r=3 \rightarrow$ La longitud de las palabras a enviar será de $3+2=5$ bits.
- Paso 3: El número de palabras de datos (2^m).
 - El número de palabras de datos a enviar es de $2^2=4$ palabras
- Paso 4: Localizar posiciones de bits de datos y bits redundantes.
 - Los bits redundantes se intercalan con los datos y se colocan en posiciones que son potencia de 2. Por ejemplo, $2^0, 2^1, 2^2, 2^3, \dots$

	r	r	d	r	d	d	d
Posición	1	2	3	4	5	6	7
Potencias	2^0	2^1	2^1+2^0	2^2	2^2+2^0	2^2+2^1	$2^2+2^1+2^0$

Ejemplo 1

Paso 5: Códigos de palabras de datos.

- A partir de los bits de datos se hacen todas las combinaciones posibles de $m=2$ bits de datos. Y obtendremos las 4 palabras de datos que se quieren enviar.

Posición Potencias	1 2^0	2 2^1	3 2^1+2^0	4 2^2	5 2^0+2^2
	r	r	d	r	d
			0		0
			0		1
			1		0
			1		1

Ejemplo 1

Paso 6: Códigos de palabras de Hamming.

- A partir de los bits de datos $m=2$ y de los bits de redundancias $r=3$. Se pueden formar $2^{2+3}=32$ palabras de código, de las cuales sólo $2^2=4$ palabras de dicho código serán correctas que serán las asociadas a los datos a transmitir y que cumplirán las relaciones de paridad entre los bits de datos 'd' y los bits de redundancia 'r'.

Posición Potencias	1 2^0	2 2^1	3 2^1+2^0	4 2^2	5 2^0+2^2
	r	r	d	r	d
	0	0	0	0	0
	1	0	0	1	1
	1	1	1	0	0
	0	1	1	1	1

▪ Ejemplo 2

- Paso 1: Numerar los bits de izquierda a derecha ascendentemente empezando en k=1
- Paso 2: Inicializar una variable contador a 0.
- Paso 3: Comprobar para cada bit redundante si la paridad es correcta.

Posición	k=1	k=2	k=3	k=4	k=5
Potencias	2^0	2^1	2^1+2^0	2^2	2^0+2^2
	r	r	d	r	d
	1	1	1	0	1



▪ Ejemplo 2

- Paso 5: Mirar el contador:
 - Si cont=0 entonces no hay errores.
 - Si cont≠0 entonces hay errores y el contador indica el bit erróneo.
 - cont=5 -> el bit erróneo es el quinto. Si se corrige la palabra correcta será 111100, que si está en código Hamming de palabras correctas obtenido en el ejemplo 1.

Posición	1	2	3	4	5
Potencias	2^0	2^1	2^1+2^0	2^2	2^0+2^2
	r	r	d	r	d
	1	1	1	0	0

▪ Códigos Reed-Solomon (RS)

▪ Conceptos previos:

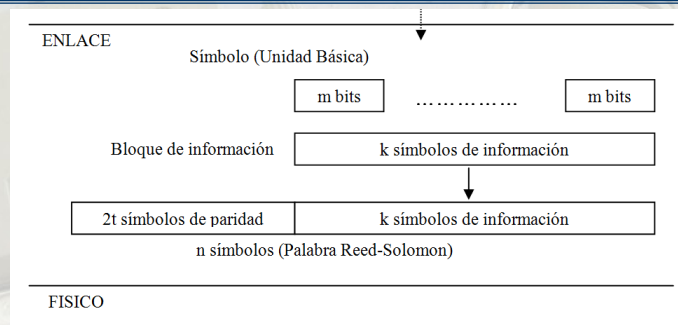
- **Símbolo** de un código Reed-Solomon. Un símbolo es un conjunto de m bits.
- **Borrado**: Símbolo erróneo cuya localización se conoce.
- **Error de símbolo**: Cuando al menos 1 bit del símbolo es erróneo.
- **Código Reed-Solomon**. Se puede describir un código por la tupla $RS(n,k)$ donde k es el número de símbolos del bloque de información y n es el tamaño del mensaje completo una vez añadidos los símbolos de redundancia.
- **Palabras Reed-Solomon**. Son cada uno de los bloques individuales de información más los símbolos de redundancia añadidos a cada bloque. Así una palabra estará formada por $n=k+r$ símbolos y donde $r=n-k$ siendo r la cantidad de símbolos redundantes añadidos.

▪ Códigos Reed-Solomon (RS)

▪ Proceso para la obtención de palabras de un código RS:

Regla:

Los códigos Reed-Solomon permiten corregir hasta un máximo de $t=r/2$ símbolos, donde t es como se denota a los símbolos erróneos cuya localización se desconoce. Además, Reed-Solomon es capaz de corregir hasta $2t$ borrados



1. Servicios del nivel de enlace.
2. Funciones del nivel de enlace.
3. Delimitación de tramas.
4. Direccionamiento de tramas.
5. Detección y corrección de errores en tramas.
 - Códigos de redundancia cíclica (CRC).
 - Códigos Hamming.
 - Códigos Reed-Solomon
6. Medición de la tasa de errores.

- En todo sistema de transmisión habrá ruido y éste dará lugar a errores que modificarán uno o varios bits de la trama que se envíe.
- Las probabilidades que definen los posibles errores en las tramas son:
 - P_0 : Probabilidad de que un bit recibido sea erróneo.
 - P_1 : Probabilidad de que una trama llegue sin errores.
 - P_2 : Probabilidad de que una trama llegue con uno o más errores no detectados.
 - P_3 : Probabilidad de que una trama llegue con todos los errores que se han producido, detectados.

- Si no se usa ningún método para detectar errores, entonces será imposible detectar algún error en uno o más bits. En tal caso, si se supone que la probabilidad de que un bit cualquiera de la trama enviada tiene una probabilidad de error de P_b :

$$P_1 = (1 - P_b)^F \quad \text{donde F es el número de bits}$$
$$P_2 = 1 - P_1 \quad \text{por trama}$$
$$P_3 = 0$$

- La probabilidad de que una trama llegue sin ningún bit erróneo disminuye al aumentar la probabilidad de que un bit sea erróneo.
- La probabilidad de que una trama llegue sin errores disminuye al aumentar la longitud de la misma.