

Corrección de Errores

Profesor: Héctor Abarca A.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

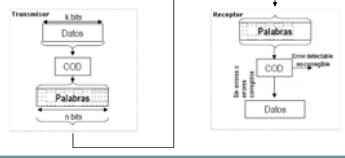
Corrección de Errores

- **FEC (Forward Error Correction)**
 - Significa corrección de errores a posteriori y se utiliza en sistemas sin retorno o sistemas en tiempo real donde no se puede esperar a la retransmisión para mostrar los datos.
 - Básicamente consiste en codificar en el transmisor cada bloque de k bits de la trama en palabras de n bits, siendo $n > k$. El receptor decodifica las palabras en los bloques originales aunque éstos tuviesen algún error.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC.**
 - Los bits añadidos, conocidos como de redundancia, hacen posible detectar errores y deducir el dato que se transmitió.



Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC**
 - Se dispone de los siguientes tipos:
 - **FEC a bloques.**
 - Sus variantes más usadas:
 - BCH.
 - RS (Reed Solomon).
 - **FEC convolucional.**
 - Aplica el algoritmo Viterbi.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC a bloques.**
 - Se denomina *Distancia Hamming* entre dos códigos al número de símbolos en que se diferencian.
 - **Peso de una palabra:** número de unos que tiene.
 - **Distancia de hamming:** Número de bits en que difieren dos palabras.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC a bloques. Distancia Hamming.**

10001110	11100101
00111000	11110111
d=5	d=2

Peso de la suma de las 2 palabras.

10001110	11100101
+00111000	+11110111
=10110110 (peso 5)	=00010010 (peso 2)

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC a bloque. Distancia Hamming.**
 - Cuanto mayor sea la distancia de hamming entre dos palabras, más difícil será que un error en la transmisión convierta una en la otra, ya que será necesario alterar d bits.
 - Un código de distancia hamming d será capaz de detectar errores en $d-1$ bits.
 - Un código de distancia hamming d será capaz de corregir errores en $(d-1)/2$ bits.
 - Para corregir errores en d bits hará falta un código con distancia de hamming $2d+1$.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC a bloques. Distancia Hamming.**
 - **Códigos de control de paridad.**
 - Se añade un bit de paridad al final de la palabra de forma que el número total de unos, incluido el bit de paridad sea par (paridad par) o impar (paridad impar).
 - Paridad par: 1011000 1
 - Paridad impar: 1101011 0
 - Este código tiene una distancia de hamming igual a 2, así que es capaz de detectar errores en 1 bit.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC a bloques. Distancia Hamming.**
 - **Códigos de control de paridad.**
 - Si la transmisión se realiza por bloques, se pueden añadir bits de paridad adicionales.

A	1	0	1	1	0	1
B	0	1	1	0	0	0
C	1	1	0	1	0	1
D	1	0	0	0	1	0
E	1	1	0	1	1	0
F	0	0	1	0	0	1
Paridad del bloque	0	1	1	1	0	1

- Distancia de hamming 4. Puede corregir errores en 1 bit y detectar errores en 1, 2 ó 3 bits.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC a bloques. Distancia Hamming.**
 - **Códigos de hamming.**
 - Son un subconjunto de los códigos de control de paridad, en los cuales se disponen los bits de paridad de forma que permitan localizar la presencia de errores en el mensaje.
 - Su distancia de hamming mínima es 3.
 - Para palabras de L bits, hace falta que R de esos bits sean de paridad para poder corregir un error en un bit, donde:
 - $L \leq 2^R - 1$
 - R será el menor número entero que cumpla esta condición

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC a bloques. Distancia Hamming.**
 - **Códigos de hamming.**
 - Cada bit de paridad debe controlar un conjunto distinto de bits de información.
 - Un error en un bits de información debe afectar a 2 o más bits de paridad.
 - Para el cálculo de los bits de paridad sólo se pueden utilizar bits de información \Rightarrow no se pueden incluir los otros bits de paridad.
 - Ejemplo:

i_0	i_1	i_2	p_0	p_1	p_2
1	1	0	0	1	1

Donde:
 $p_0 = i_0 \oplus i_1$
 $p_1 = i_0 \oplus i_2$
 $p_2 = i_1 \oplus i_2$

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC a bloques. Distancia Hamming.**
 - **Códigos de hamming. Código óptimo.**
 - Se numeran los bits de la palabra código empezando por 1. Los bits potencia de 2 serán de redundancia y el resto de datos.
 - Cada bits de datos contribuye a varios bits de redundancia, que se determinan descomponiendo la posición del bit en suma de potencias de 2.
 - Ejemplo: $11 = 1 + 2 + 8$
 - Si se produce un error, el bit erróneo será el situado en la posición que se obtenga sumando las posiciones de los bits de redundancia incorrectos.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC a bloques. Distancia Hamming.**
 - **Códigos de hamming. Código óptimo.**
 - Ejemplo de código óptimo para caracteres de 7 bits.

Carácter	ASCII	Palabra código
T (84)	1 0 1 0 1 0 0	0 0 1 1 0 1 0 1 1 0 0
e (101)	1 1 0 0 1 0 1	0 0 1 1 1 0 0 0 1 0 1
l (108)	1 1 0 1 1 0 0	0 0 1 0 1 0 1 1 1 0 0
e (101)	1 1 0 0 1 0 1	0 0 1 1 1 0 0 0 1 0 1
c (99)	1 1 0 0 0 1 1	1 1 1 1 1 0 0 0 0 1 1
o (111)	1 1 0 1 1 1 1	1 0 1 0 1 0 1 1 1 1 1

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC a bloques. Código Hamming H.**
 - Usado en algunos radioenlaces digitales en la década de los '80. En particular la versión H(555,544).
 - Permite corregir 2 errores en la secuencia de 544 bits de información.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC a bloques. Código BCH.**
 - Bose-Chaudhuri-Hocquenghen.
 - Es el código más conveniente para errores independientes. Los parámetros definidos son:
 - **Longitud del bloque:** $N=2^M-1$ $M \geq 3$
 - **Bits de información:** $l=N-M.t$
 - **Distancia mínima:** $d=2.t+1$

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC a bloques. Código BCH.**
 - Es usado por ejemplo en telefonía celular analógica AMPS en el canal de control bajo la versión BCH(48,36) y BCH(40,28).
 - En codificadores digitales de TV a 34Mb/s se utiliza el codec BCH(511,493) para corregir 2 errores por bloque.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC a bloques. Código RS.**
 - Reed-Solomon.
 - Es una variante del BCH y la más apropiada para ráfagas de errores. Los parámetros definidos:
 - **Bits por símbolo:** m
 - **Longitud del bloque:** $N=m \cdot 2^m - 1$
 - **Bits de información:** $(N-l) = m \cdot 2^l$
 - **Distancia mínima:** $d = m \cdot (2 \cdot t + 1)$

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC a bloques. Código RS.**
 - En general el número de errores corregidos son t ráfagas de m bits en una palabra de código. Por ejemplo:
 - En RS(60,40) se corrigen 2 ráfagas de 4 bits errados.
 - Una aplicación, entre otras, es en radioenlaces digitales de 140Mb/s en la versión RS(65/62) para corregir 4 errores en 4 bloques.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC convolucional.**
 - Se presenta como el método más interesante, teniendo en cuenta la modulación TCM (Trellis Code Modulation).
 - Algoritmo de viterbi.
 - Método denominado decodificación de máxima probabilidad, consiste en computar a cada camino un peso consistente en el número de diferencias acumuladas.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC convolucional. Algoritmo de Viterbi.**
 - Cuando se recibe un error (01 en lugar de 11), se encuentra que dicha secuencia es imposible; sólo 2 errores consecutivos pueden simular un camino correcto. Detectado el error es necesario saber cual es; es decir, si se transmitió 00 o 11. Ambos caminos comienzan con un peso 1 (número de errores acumulados).

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC convolucional. Algoritmo de Viterbi.**
 - Al paso siguiente las distintas posibilidades incrementan el valor del peso con excepción de un camino (el de máxima probabilidad) que mantiene el peso en (1). Al cabo de una determinada longitud de análisis se decide el camino de máxima probabilidad (mínimo de errores) y se determina el bit con error, corrigiéndolo. Si existen muchos errores puede ser que varios caminos tengan igual peso; en tal caso se selecciona uno en forma aleatoria.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **FEC convolucional. Representación.**

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **ARQ.**
- La corrección de los posibles errores en la transmisión se consigue retransmitiendo de nuevo la trama problemática.
- Se basa en:
 - Confirmaciones positivas (ACK, ACKnowledgement, Reconocimiento).
 - Retransmisión tras expiración de un temporizador (time_out). El origen retransmite las tramas sin ACK después de un periodo de tiempo.
 - Confirmación negativa (NACK, NoACKnowledgement, No Reconocimiento) y retransmisión. El destino detecta un error en la trama y envía un ACK negativo.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **ARQ.**
- Las variantes de ARQ son:
 - ARQ con parada y espera.
 - ARQ con vuelta atrás N.
 - ARQ con rechazo selectivo.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **ARQ. ARQ con parada y espera**
 - Basado en el control de flujo parada y espera.
 - El transmisor envía una única trama y espera la llegada del ACK.
 - No se puede enviar otra trama hasta haber recibido el ACK.
 - Sencillo e ineficiente.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **ARQ. ARQ con parada y espera**
 - Si se recibe una trama dañada el receptor la descarta.
 - El transmisor asocia un timeout a cada trama enviada. Si no recibe el ACK antes de que expire, la retransmite.
 - Si se pierde o daña el ACK, el transmisor retransmitirá la trama.
 - El receptor debe descartar la trama duplicada que recibe.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **ARQ. ARQ con vuelta atrás N.**
 - Basado en ventana deslizante.
 - El transmisor puede enviar una ventana de tramas.
 - El receptor confirma las tramas recibidas correctamente (ACK).
 - Si el receptor detecta un error, envía un ACK negativo (N-ACK).
 - Descarta esa trama y las futuras tramas recibidas hasta recibir la trama errónea.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **ARQ. ARQ con vuelta atrás N.**
 - El transmisor si recibe un N-ACK debe retransmitir la trama errónea y todas las posteriores.
 - Es bastante utilizado (aunque con variantes)

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

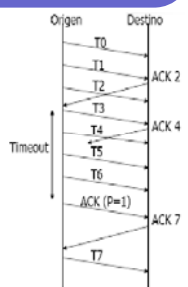
- **ARQ. ARQ con vuelta atrás N.**
 - Si el receptor recibe la trama X errónea, la descarta y:
 - Si el origen envía la trama X+1, el receptor envía N-ACK X, el transmisor retransmite la trama X.
 - Si el origen no envía más tramas, el receptor no contesta.
 - El origen enviará un ACK con un bit $P=1$
 - El receptor al recibir el ACK con $P=1$, contesta con un ACK X.
 - El transmisor retransmitirá la trama X.



Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

- **ARQ. ARQ con vuelta atrás N.**
 - Si se pierde el ACK de la trama X (ACK X+1):
 - Si el transmisor recibe algún ACK posterior (ACK X+2,...) antes de que expire el timeout, la trama X está confirmada.
 - Si el timeout expira, se envía un ACK con $P=1$, y se inicia otro temporizador.
 - Si el receptor no responde antes del temporizador, el transmisor repite el ACK con $P=1$.
 - Se reintenta varias veces.



Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

● **ARQ. ARQ con rechazo selectivo.**

- Sólo se retransmiten las tramas con ACK negativo (SN-ACK) o cuando expira el timeout.
- Se minimiza el número de retransmisiones.
- El receptor necesita un buffer suficientemente grande para almacenar las tramas que va recibiendo después de un SN-ACK.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

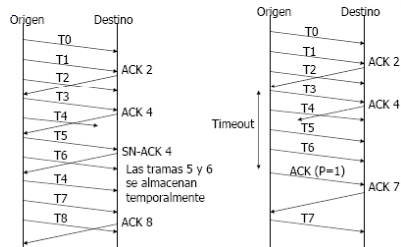
● **ARQ. ARQ con rechazo selectivo.**

- También necesita una lógica especial para insertar la trama retransmitida en su posición correcta.
- El transmisor también tiene una lógica más compleja para enviar tramas fuera de orden.
- Se utiliza bastante menos que el ARQ con vuelta atrás N.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Corrección de Errores

● **ARQ. ARQ con rechazo selectivo.**



Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Control de Flujo

Profesor: Héctor Abarca A.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Control de Flujo

- **Control de Flujo**
 - Consiste en que el receptor tenga la capacidad de regular el ritmo de envío del transmisor para no verse saturado.
 - Define el conjunto de procedimientos utilizados para restringir la cantidad de datos que el emisor puede enviar antes de esperar un reconocimiento (ACK)
 - Evitar que el flujo de datos desborde al receptor.
 - Básicamente existen dos técnicas.
 - Parada y espera
 - Ventana deslizante

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Control de Flujo

- **Parada y espera.**
 - La fuente transmite una trama
 - El destino recibe la trama y responde enviando una confirmación.
 - La fuente espera dicha confirmación antes de transmitir la siguiente trama.
 - El destino puede parar el flujo reteniendo las confirmaciones.
 - Buen funcionamiento usando un número reducido de tramas de gran tamaño.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Control de Flujo

- **Parada y espera.**

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Control de Flujo

- **Parada y espera. Fragmentación.**
- Es frecuente que la fuente rompa un gran bloque de datos en bloques pequeños:
 - Tamaño limitado de la memoria temporal.
 - Los errores se detectarán antes, una vez recibida toda la trama.
 - Si hay errores, se necesitará retransmitir pequeñas tramas.
 - Evita que una estación ocupe el medio durante un largo período.
- El procedimiento de parada y espera puede ser inadecuado.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Control de Flujo

- **Parada y espera. Utilización del enlace.**

Utilización del enlace mediante parada y espera
(tiempo de transmisión = t ; tiempo de propagación = a).

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Control de Flujo

● **Ventana deslizante. Ejemplo.**

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Control de Flujo

● **Ventana deslizante. Mejoras.**

- Mediante un mensaje "receptor no preparado" (**RNR**, receiver not ready), el receptor puede confirmar las tramas prohibiendo la transmisión de tramas adicionales.
- Se debe transmitir una confirmación normal para reabrir la ventana.
- Si los dos dispositivos necesitan enviar datos, se debe utilizar un procedimiento denominado incorporación de confirmación.
 - Si un dispositivo no tiene datos que enviar, enviará únicamente una trama de confirmación.
 - Si hay datos, pero nada que confirmar, repetirá la última confirmación enviada con anterioridad.

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Control de Flujo

● **Ventana deslizante . Mejoras.**

- $F(n,m) = f(n) + RR(m)$
 - n: número de secuencia enviada ó N(S).
 - m: número de secuencia recibida ó N(R)

Sistemas de Comunicación de Datos II.
Héctor Abarca A.

Control de Flujo

• Ventana deslizante . Prestaciones.

- Se pueden dar dos casos:
 - $W \geq 1+2a$. Los acuses de recibo del receptor llegan al transmisor antes de que agote su ventana de emisión. En este caso:
$$U = 1$$
 - $W < 1+2a$. El transmisor agota su ventana de emisión en $t=W$ y no podrá seguir enviando tramas hasta recibir la primera confirmación en $t=1+2a$. La utilización de la línea será:

$$U = \frac{W}{1+2a}$$

Sistemas de Comunicación de Datos II.
Héctor Abarca A.
