Curso:
EL271 Codificación de Fuente y de Canal
Unidad 2: CODIFICACIÓN DE BLOQUE
Semana 4, Sesión 4

CARLOS VALDEZ VELÁSQUEZ-LÓPEZ, DR. ING

2023-02

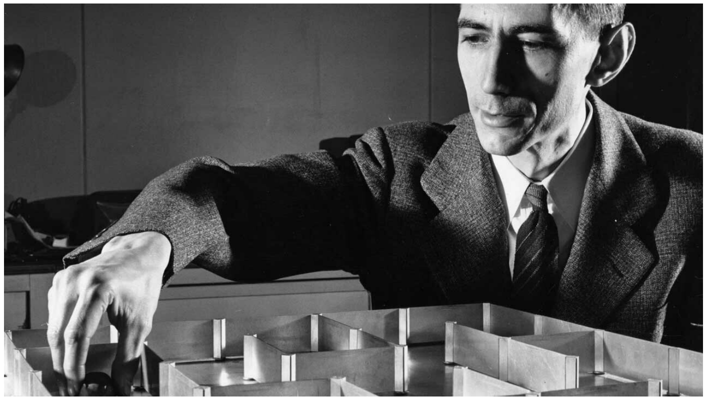
Logro de la sesión

Al finalizar la sesión, el alumno explica los conceptos más importantes sobre el Teorema de Shannon de la capacidad de canal, la codificación de canal, detección y corrección de errores, distancia de Hamming, esquemas FEC y ARQ

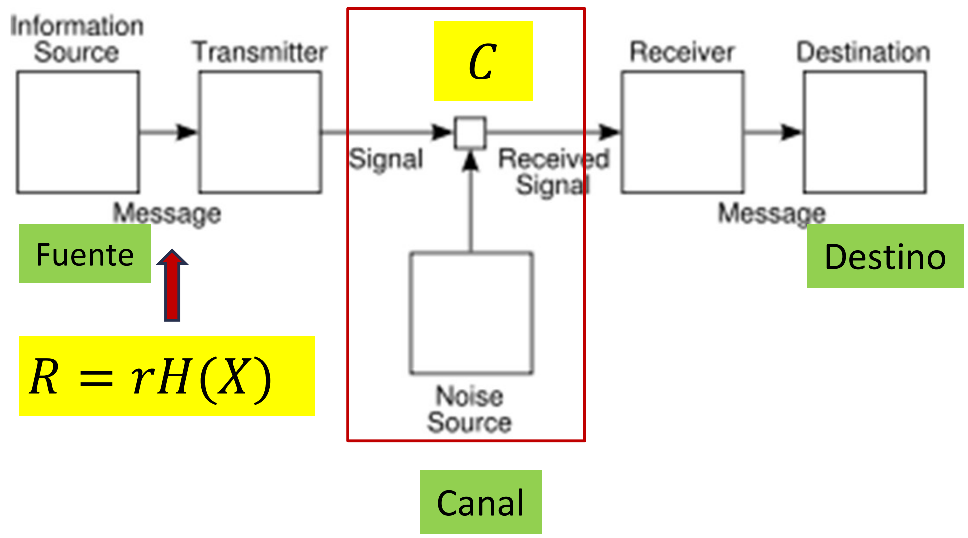
Contenido

1) El Teorema de Shannon de la capacidad del canal

"Si un canal tiene capacidad C y una fuente tiene velocidad de información $R \leq C$, entonces existe un sistema de codificación tal que la salida de la fuente puede transmitirse a través del canal con una probabilidad de error P_e arbitrariamente pequeña. Por el contrario, si $R > C$, no es posible transmitir la información sin errores"



1) El Teorema de Shannon de la capacidad del canal



Fuente: "A Mathematical Theory of Communications", C.E. Shannon, The Bell System Technical Journal, October 1948

2) La capacidad del canal continuo

- Para un canal continuo ideal limitado en banda a B Hz, con relación señal a ruido S/N, la capacidad de canal C:

$$C = B\ \log(1 + S/N)\ \text{bits/segundo}$$

- Expresión compacta que define la capacidad del canal en función de dos parámetros
- S/N y B, son limitaciones físicas inherentes al canal
- S/N: relación señal a ruido disponible
- B: ancho de banda en Hz

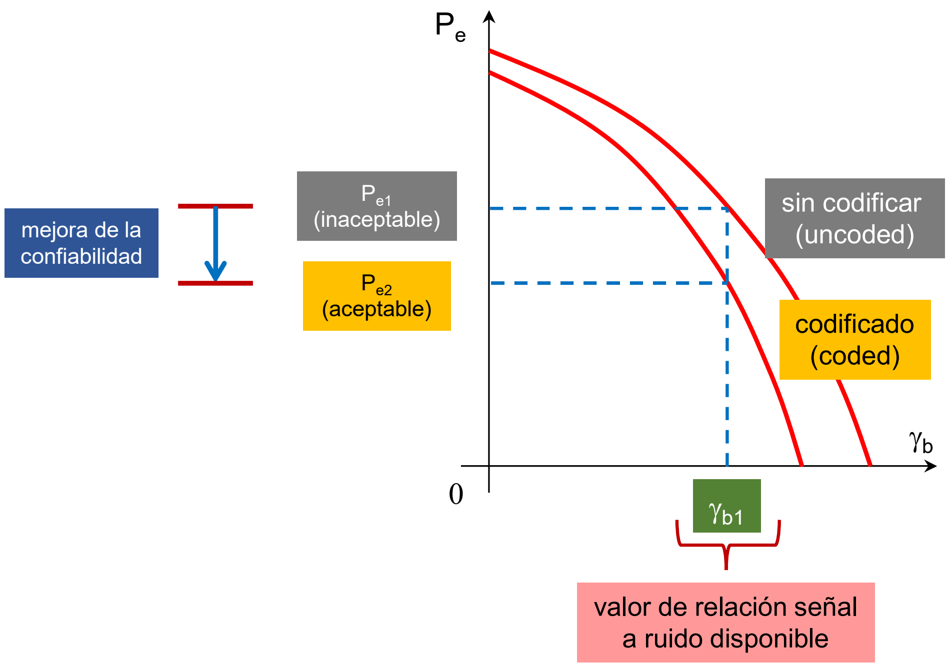
Formas de alcanzar la capacidad del canal

- Dicho teorema es no constructivo pues no indica cómo deben diseñarse los códigos
- Sin embargo, existe una deducción matemática sobre la probabilidad de error, que demuestra que $P_e \rightarrow 0$
- Mediante señales ortogonales, por ejemplo M-FSK: tienen la gran desventaja del ancho de banda que ocupan
- Mediante técnicas de codificación de canal, de acuerdo a lo indicado por C.E. Shannon en su Teorema

3) Objetivo de la codificación de canal

- Los errores de transmisión en comunicaciones digitales dependen de la relación señal a ruido $\gamma_b = E_b/\eta$
- Para una γ_b dada, con una probabilidad de error P_e inaceptable, deben considerarse formas **alternativas para mejorar la confiabilidad** de la comunicación
- Opción 1: aumentar la potencia de la señal (solución de “fuerza bruta”, costosa)
- Opción 2: codificar la señal (solución elegante)

3) Objetivo de la codificación de canal

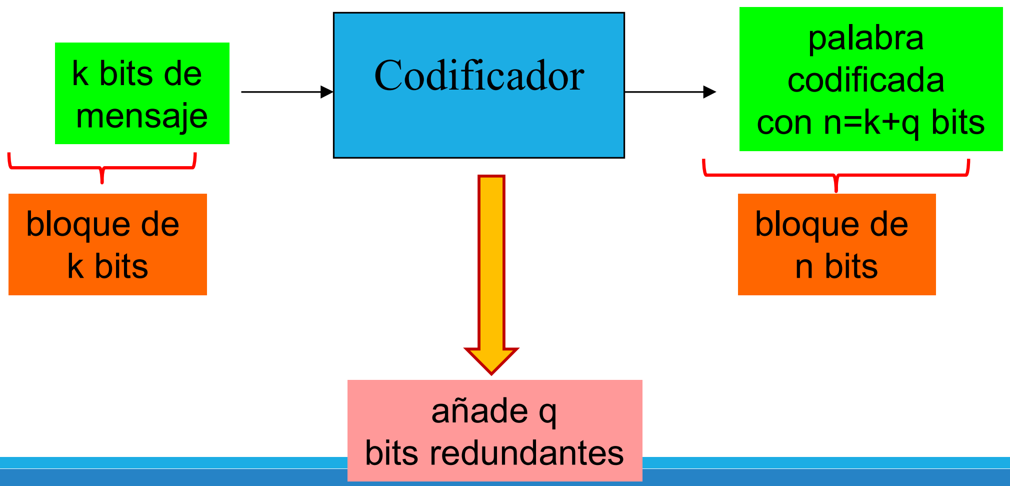


4) Detección y corrección de errores, distancia de Hamming

- A través de la codificación de canal, es posible la detección y la corrección de errores
- Algunos esquemas permiten únicamente la detección de errores, otros además la corrección
- Ello se logra mediante la introducción de bits denominados redundantes que no transmiten información pero sirven para protegerla contra los efectos nocivos ocasionados por el canal
- Existen 2 grandes familias: códigos de bloque y códigos convolucionales

Códigos de bloque

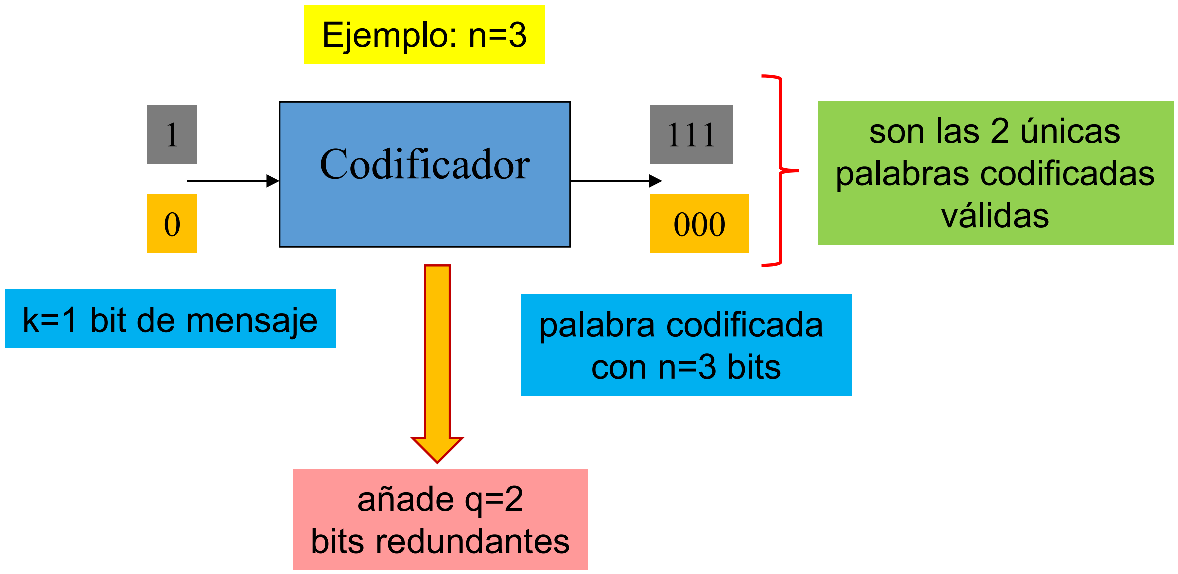
- Codifican la secuencia binaria en bloques
- Notación
  - k: número de bits de mensaje
  - q: número de bits redundantes
  - n=k+q: número de bits codificados



Ejemplos de 2 esquemas simples: repetición y chequeo de paridad

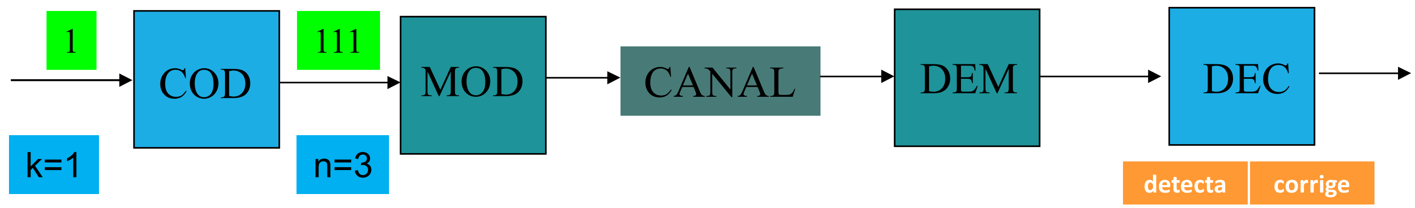
Código de repetición

- Consiste en repetir cada bit de mensaje, de modo que la palabra codificada (codeword) tiene n bits idénticos



¿Cómo funciona el código de repetición?

Ejemplo: asumimos que se transmitió un 1



si el canal introduce un error (error simple), el DEC recibe cualquiera de las siguientes palabras

	detecta	corrige	
011	si	si	1
101	si	si	1
110	si	si	1

si el canal introduce dos errores (error doble), el DEC recibe cualquiera de las siguientes palabras

	detecta	corrige	
001	si	no	0
100	si	no	0
010	si	no	0

si el canal introduce tres errores (error triple), el DEC recibe la siguiente palabra

	detecta	corrige	
000	no	no	0

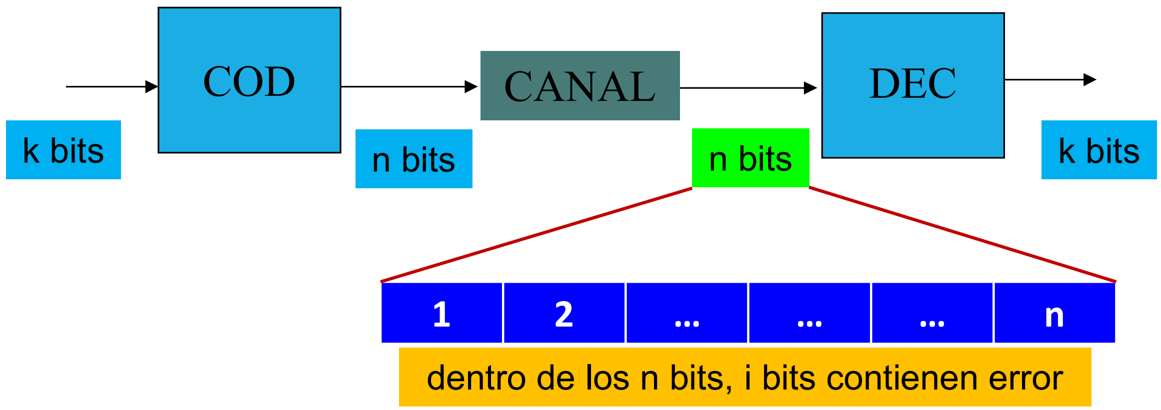
1. El DEC detecta errores si recibe una palabra no válida
2. Para corregir el DEC aplica mayoría de bits

¿Cómo funciona el código de repetición?

- El DEC detecta errores si recibe una palabra que “no está en su diccionario”
- El DEC detecta errores simples y dobles
- El DEC no detecta errores triples porque la palabra recibida es válida
- El DEC solamente puede corregir errores simples, si aplica la regla de mayoría de bits (“mayoría manda”)
- La situación es mejor que en el esquema sin codificar, en donde no hay forma de detectar que un 1 cambió a 0 o viceversa, y menos aún corregir

¿Cómo funciona el código de repetición?

- A la salida del DEC, la situación debe ser mejor que el caso sin codificar
- En su entrada, el DEC recibe n bits (del DEM), de los cuales i bits pueden estar errados (no se muestra el MODEM)



- El DEC cumplirá su rol aplicando una regla de decodificación al bloque de n bits recibidos (tal como la de mayoría de bits)

A la entrada del DEC (salida del DEM), ¿Cuál es la probabilidad de tener i bits errados en el bloque de n bits?

- Los errores de bit son aleatorios (es decir, los i errores pueden ubicarse en cualquiera de los n bits) y ocurren con probabilidad α, los no errores (bits correctos) tienen probabilidad $(1-\alpha)$. Ejemplo:

bit N°	1	2	...	...	...	n
ocurrencia (ejemplo)	error 1	no error 1	error 2	no error 2	no error (n-i)	error i
Probabilidad de error/no error	α	$(1-\alpha)$	α	$(1-\alpha)$	$(1-\alpha)$	α
Probabilidad de i errores en n bits	$\underbrace{\alpha.\alpha.\ldots\ \alpha.}_{\text{i veces}}\ \underbrace{(1-\alpha).\ (1-\alpha)\ldots.\ (1-\alpha)}_{\text{(n-i) veces}}$					

- Además, el número de combinaciones diferentes de las ocurrencias de errores/no errores (es decir, en ubicaciones distintas al ejemplo) está dado por el número combinatorio $\binom{n}{i}$, y todas las contribuciones deben incluirse

A la entrada del DEC (salida del DEM), ¿Cuál es la probabilidad de tener i bits errados en el bloque de n bits?

- Entonces, la probabilidad de i errores en una palabra codificada de n bits se puede expresar mediante la función binomial:

$$P(i,n) = \binom{n}{i} \alpha^{i} (1-\alpha)^{n-i} \approx \binom{n}{i} \alpha^{i}, \quad \alpha \langle\langle 1$$

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} = \frac{n(n-1)\cdots(n-i+1)}{i!}$$

- α viene a ser la probabilidad de error de bit individual (como la P_b estudiada en modulación digital), y α<<1 (ejemplo: $\alpha = 10^{-6}$)

Ejemplo

- La probabilidad de error simple $(i = 1)$, con α=10^{-6}

$$P(1,3) = \binom{3}{1}\alpha^{1}(1-\alpha)^{3-1} \approx \binom{3}{1}\alpha^{1} = 3\times 10^{-6}$$

- La probabilidad de error doble $(i = 2)$, con α=10^{-6}

$$P(2,3) = \binom{3}{2}\alpha^{2}(1-\alpha)^{3-2} \approx \binom{3}{2}\alpha^{2} = 3\times 10^{-12}$$

- La probabilidad de error triple $(i = 3)$, con α=10^{-6}

$$P(3,3) = \binom{3}{3}\alpha^{3}(1-\alpha)^{3-3} = \binom{3}{3}\alpha^{3} = 1\times 10^{-18}$$

- Es claro que para $\alpha<<1$, $P(3,3) << P(2,3) << P(1,3)$
- En consecuencia, es mucho más probable la ocurrencia de errores simples que la de dobles y la de triples (todo ello en la entrada del DEC)

Las limitaciones del DEC

- Las limitaciones del DEC, se deben a los casos que no puede detectar y los que no puede corregir

Errores no detectables

- Cualquier palabra recibida, diferente de **000** y **111**, constituye una palabra con error
- Se detectan errores de 1 bit (simple) y 2 bits (doble), no así de 3 bits (triple)
- Entonces, la probabilidad de una palabra errada **no detectada** es $P_{we} = P(3,3) = \quad 3$

Las limitaciones del DEC

Errores no corregibles

- Puede usarse la regla de decodificación de **mayoría de bits**, asumiendo que al menos 2 de los 3 bits son correctos
- Así **001** se decodifica como **0** y **101** como **1**
- Esta regla corrige errores simples (1 solo bit errado), pero no los dobles ni triples
- La probabilidad de una palabra decodificada erróneamente **por no ser corregible** es $P_{we}=P(2,3)+P(3,3)=3\alpha^2-2\alpha^3$

¿Cuál es el precio a pagar por la mejora?

- El precio a pagar es la ineficiencia de los bits transmitidos, pues los redundantes no transmiten información (en el ejemplo, 1 de 3 bits transmite información)
- Se mide mediante la razón del código R_c

$$\boxed{R_c = \frac{k}{n}}$$

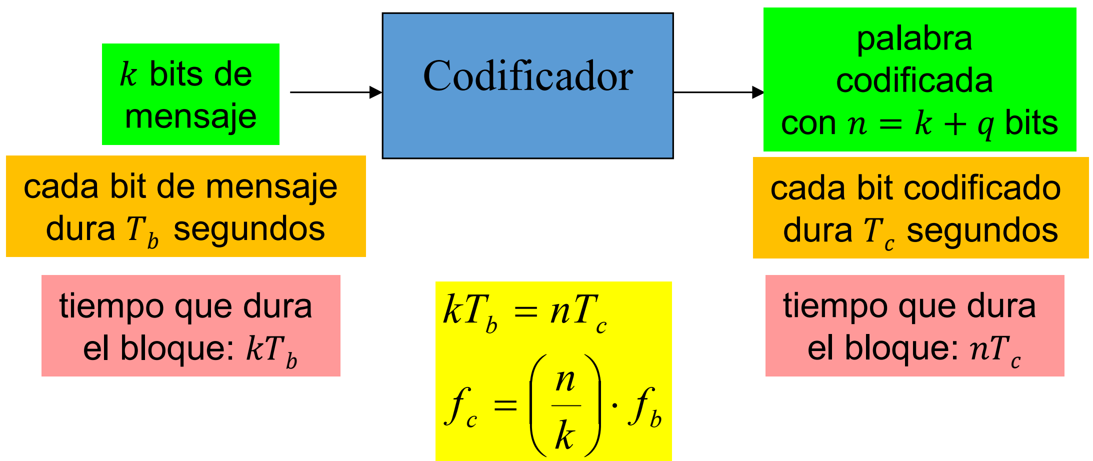
- En el código de repetición

$$\boxed{R_c = \frac{1}{n}}$$

- Lo que significa que es muy ineficiente, ya que si n aumenta, R_c disminuye
- Se busca que R_c tienda a la unidad

¿Cuál es el precio a pagar por la mejora?

- Además, se produce un aumento del ancho de banda ocupado por la señal modulada, lo cual es una gran desventaja en los canales de comunicación limitados en banda



- El ancho de banda de la señal modulada por bits codificados (B_c) es mayor que el caso sin codificar (B_u), porque la frecuencia binaria codificada f_c es mayor que f_b (note en la expresión que n/k es mayor que la unidad)

¿Cuál es el precio a pagar por la mejora?

- Es decir, mientras que el ancho de banda sin codificar es:

$$B_u = \frac{f_b}{\log_2 M}$$

en el caso codificado es:

$$B_c = \frac{f_c}{\log_2 M}$$

- De las expresiones anteriores:

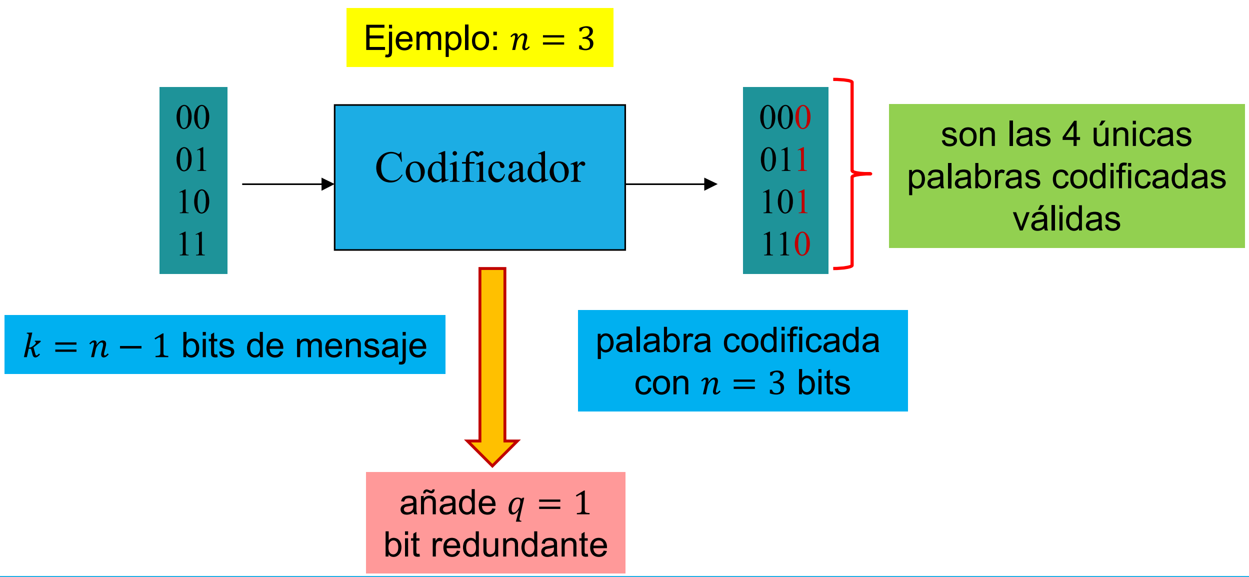
$$B_c = \left(\frac{n}{k}\right) \cdot B_u$$

- En el caso del ejemplo del código de repetición $(k = 1, n = 3)$:

$$B_c = 3 \cdot B_u$$

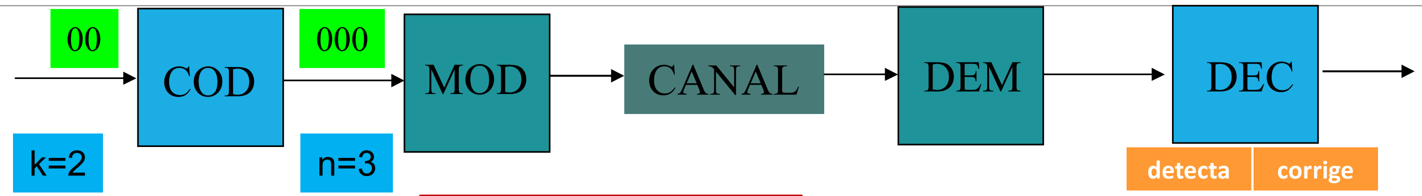
Código de chequeo de paridad

- Consiste en **añadir 1 bit a cada bloque de (n-1) bits de mensaje**, de modo que la **palabra codificada** tenga un número par de **1s** (paridad par)



¿Cómo funciona el código de chequeo de paridad?

Ejemplo: asumimos que se transmitió un 00



si el canal introduce un error (error simple), el DEC recibe cualquiera de las siguientes palabras

	detecta	corrige	
100	si	no	¿?
010	si	no	¿?
001	si	no	¿?

si el canal introduce dos errores (error doble), el DEC recibe cualquiera de las siguientes palabras

	detecta	corrige	
110	no	no	¿?
011	no	no	¿?
101	no	no	¿?

si el canal introduce tres errores (error triple), el DEC recibe la siguiente palabra

	detecta	corrige	
111	si	no	¿?

1. El DEC detecta errores impares, no detecta errores pares
2. El DEC no puede corregir porque no se sabe la ubicación del bit errado
3. La salida del DEC depende de su estructura (puede ser mediante la técnica del Síndrome)
4. En conclusión, este esquema solamente sirve para detectar errores

¿Cómo funciona el código de chequeo de paridad?

- El DEC detecta errores si recibe una palabra que “no está en su diccionario”
- El DEC detecta errores simples y triples
- El DEC no detecta errores dobles porque la palabra recibida es válida
- El DEC no puede corregir errores simples porque no se sabe la ubicación del bit errado
- La situación es mejor que en el esquema sin codificar, en donde no hay forma de detectar que un 1 cambió a 0 o viceversa

Las limitaciones del DEC

- Las limitaciones del DEC, se deben a los casos que no puede detectar y los que no puede corregir

Errores no detectables

- Cualquier palabra recibida, diferente de **000, 011, 101** y **110**, constituye una palabra con error
- Se detectan errores de 1 bit (simples) y de 3 bits (triples), no así de 2 bits (dobles)
- Entonces, la probabilidad de una palabra errada **no detectada** es $P_{we}=P(2,3)=3\alpha^2(1-\alpha)\approx 3\alpha^2$

Las limitaciones del DEC

Errores no corregibles

- El DEC no puede corregir errores simples, ni dobles, ni triples
- La probabilidad de una palabra decodificada erróneamente **por no ser corregible** es $P_{we}=P(1,3)+P(2,3)+P(3,3)\approx P(1,3)\approx 3\alpha$

¿Cuál es el precio a pagar por la mejora?

- En el presente caso

$$R_c = \frac{k}{n} = \frac{n-1}{n} = 1 - \frac{1}{n}$$

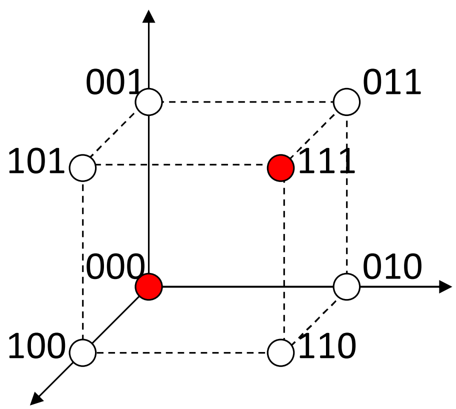
- Lo que significa que es muy eficiente, ya que si n aumenta, R_c aumenta
- R_c tiende a la unidad si n es mucho mayor que la unidad
- Tal como en el caso de repetición, el ancho de banda de la señal modulada por bits codificados (B_c) es mayor que el caso sin codificar (B_u), porque la frecuencia binaria codificada f_c es mayor que f_b (note en la expresión que n/k es mayor que la unidad)
- En el caso del ejemplo del código de chequeo de paridad:

$$B_c = \left(\frac{n}{k}\right) \cdot B_u = \left(\frac{3}{2}\right) \cdot B_u$$

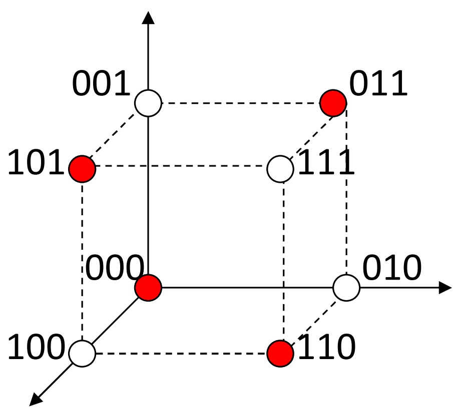
Distancia de Hamming

- Las palabras codificadas de *n* bits pueden representarse en un espacio *n*-dimensional como vectores con coordenadas de *n* bits

Ejemplo:



a) Código de repetición, *n*=3 bits



b) Código de chequeo de paridad, *n*=3 bits

- Con puntos rojos se muestran las palabras codificadas válidas

- Se observa que los vectores del código de repetición están más separados que los vectores del código de chequeo de paridad. Tal separación se mide en términos de la **distancia de Hamming**
- La distancia de Hamming, $d(X,Y)$ entre 2 vectores X e Y, se define como el número de elementos diferentes
- Si X=(101) e Y=(110), $d(X,Y)=2$, ya que hay 2 elementos diferentes
- La **distancia mínima** d_{min} de un código particular, es la mínima distancia de Hamming entre vectores de código válidos

- La distancia mínima d_{min} en el código de repetición es 3, y en el de chequeo de paridad es 2
- La capacidad de control de errores del código depende de su d_{min}:

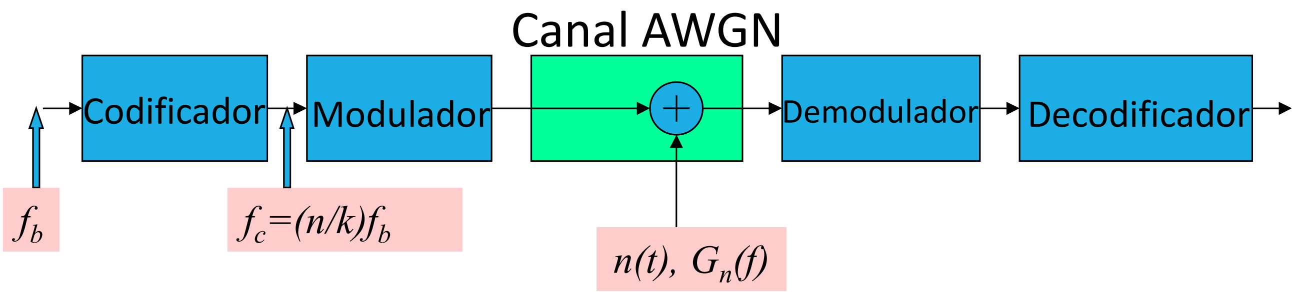
  Detección de hasta *l* errores: $d_{\min} \geq l+1$

  Corrección de hasta *t* errores: $d_{\min} \geq 2t+1$
- La **potencia del código** depende del número de bits que han sido añadidos con el propósito de controlar los errores

5) Esquemas FEC y ARQ

Forward Error Correction (FEC)

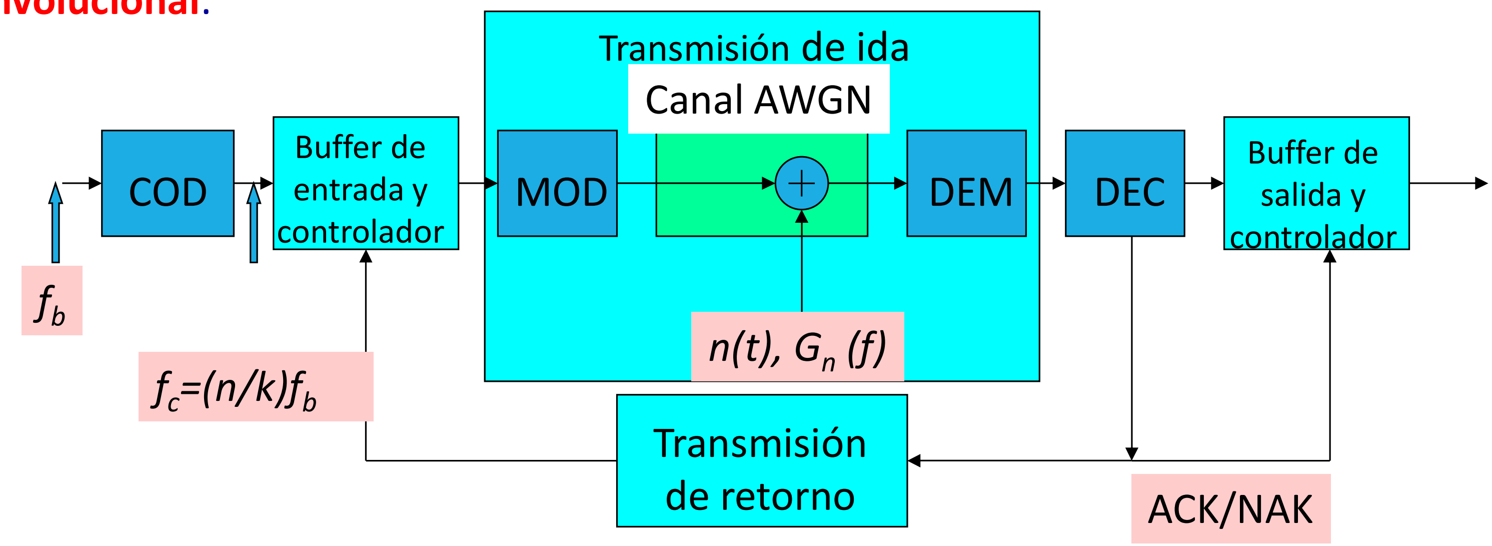
- Diagrama de bloques con un esquema **FEC** (**F**orward **E**rror **C**orrection) con **código de bloque o convolucional**:



- Se detectan y corrigen errores en forma contínua, conforme se recibe la secuencia

Automatic Repeat Request (ARQ)

- Diagrama de bloques con un esquema **ARQ** (**A**utomatic **R**epeat Re**Q**uest) con **código de bloque o convolucional**:



ACK: Positive acknowledgement
NAK: Negative acknowledgement

- En base a detección de errores, el buffer de salida/controlador envía mensajes de reconocimiento positivo (ACK) o negativo (NAK), solicitando la retransmisión del mensaje