

Unidad 4: Transmisión en bloques de datos, detección de errores. Capa de Interfaz de red.

(capa1)

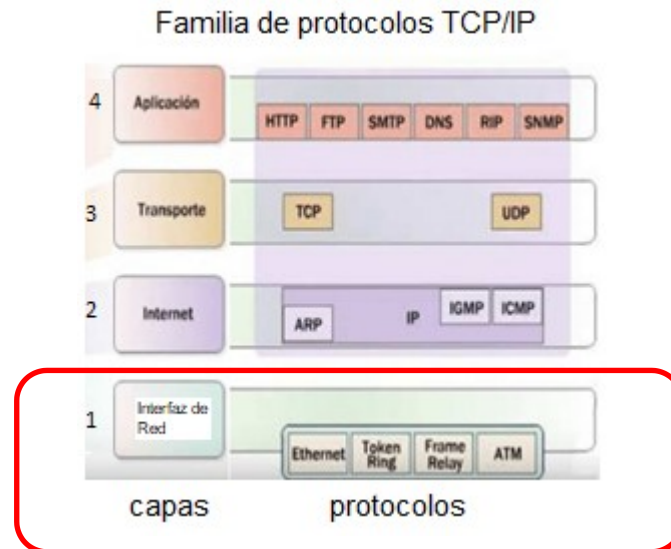
1. Empaquetamiento de datos en bloques para el envío por la red.

- a) Concepto de Bloque.
- b) Concepto de Anidamiento.

2. Transmisión de bloques por multiplexión por división de tiempo.

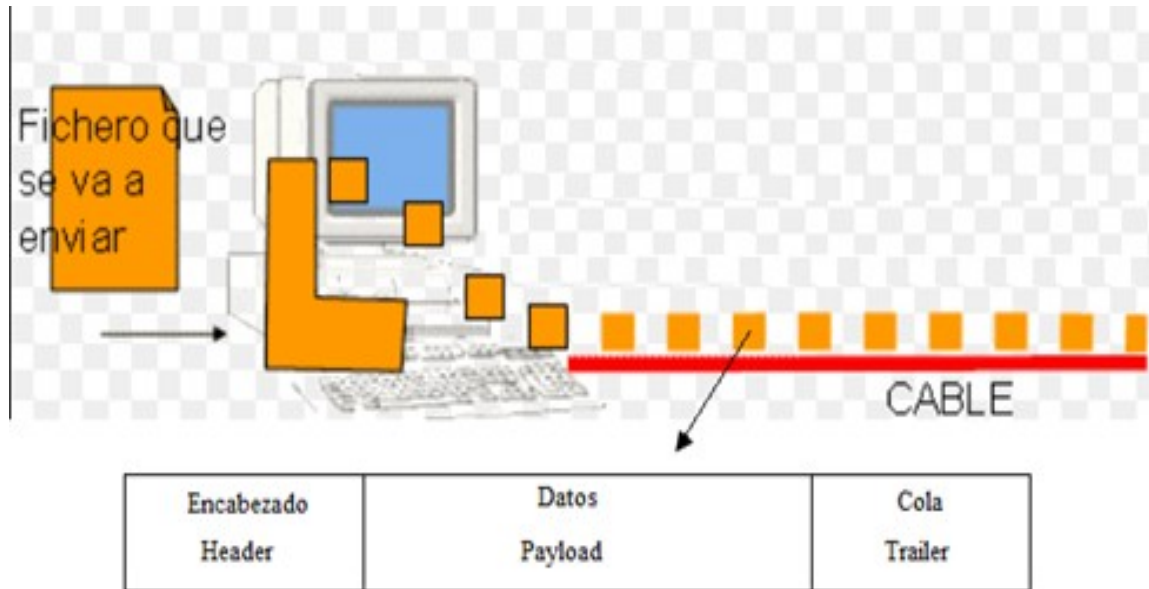
3. Errores de transmisión y Detección de errores por los siguientes sistemas:

- a) Por bit de paridad.
- b) Por paridad bidimensional.
- c) Por Cifra de comprobación.
- d) Por Comprobación de Redundancia Cíclica. (CRC).



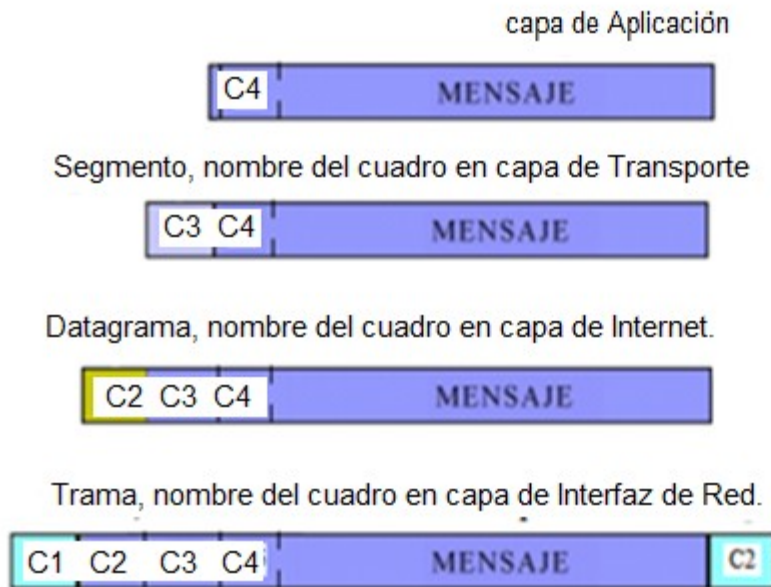
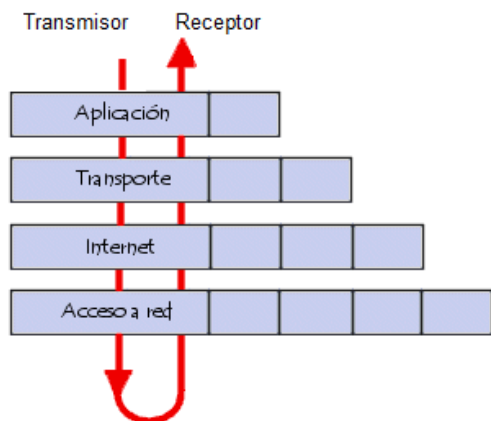
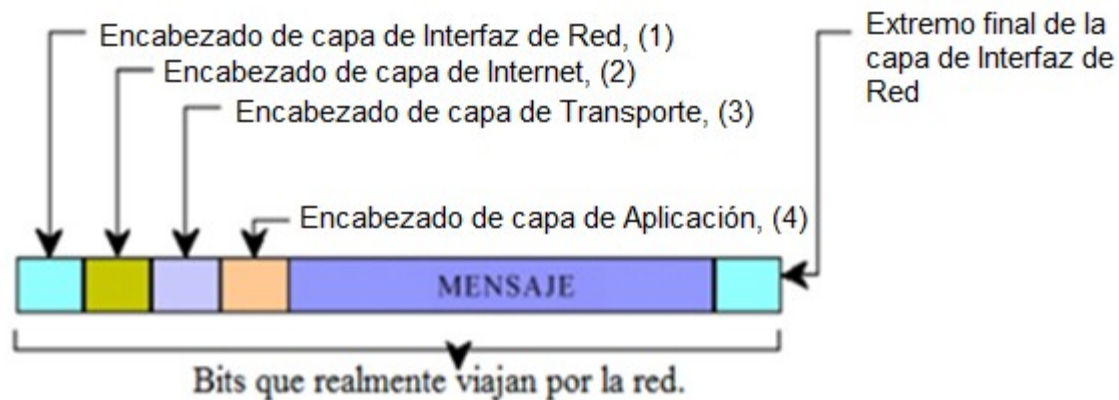
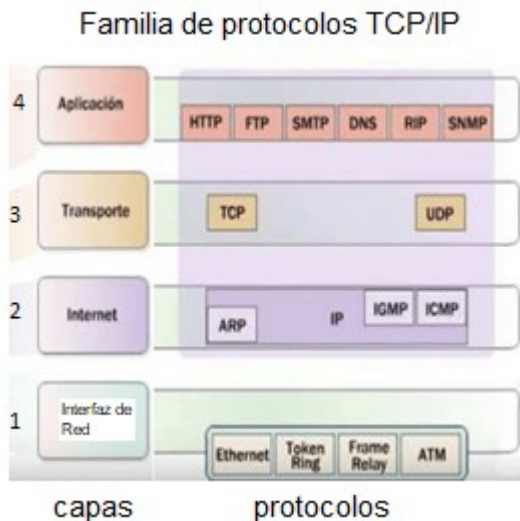
1. Empaquetamiento de datos en bloques para el envío por la red.

a) Concepto de Bloque.



1. Empaquetamiento de datos en bloques para el envío por la red.

b) Concepto de Anidamiento.



2. Transmisión de bloques por multiplexión por división de tiempo.

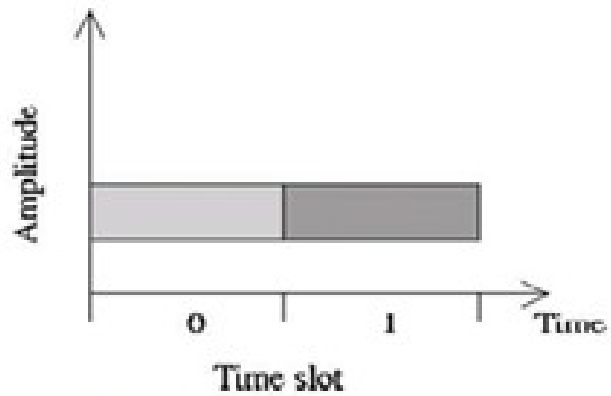


Figura 6. Multiplexación por División de Tiempo (TDM).

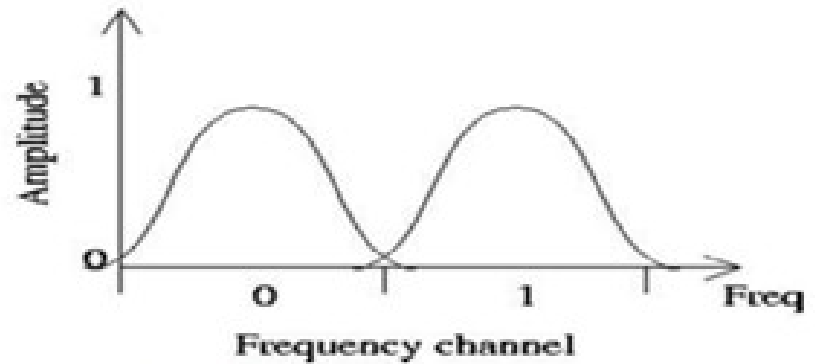
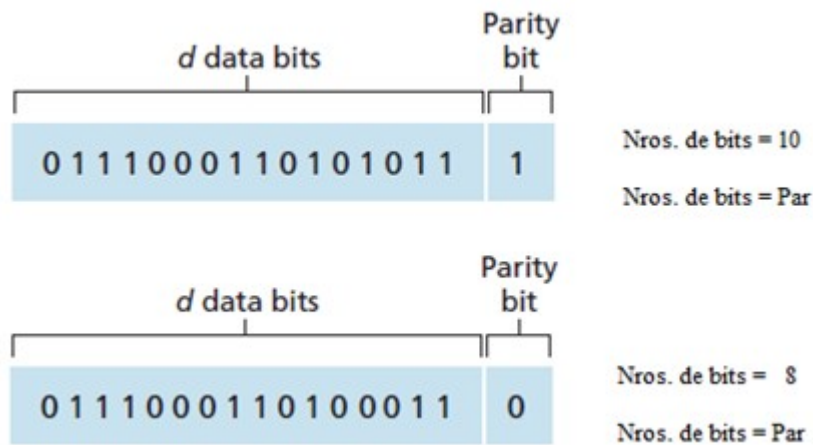


Figura 5. Multiplexación por División de Frecuencia (FDM).

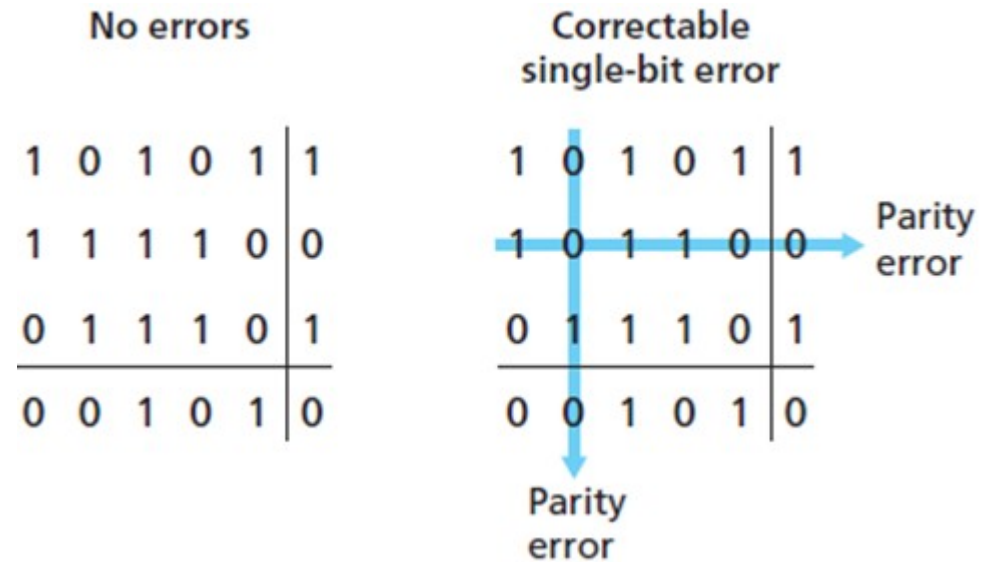
3. Errores de transmisión y Detección de errores por los siguientes sistemas:

a) Bits de paridad par.



b) Bits de paridad par bidimensional.

Este sistema permite la corrección de un bit.



3. Detección de errores por el siguiente sistema:

$$\begin{array}{r}
 \text{FOA1} \\
 + 1\text{C12} \\
 \hline
 10\text{CB3} \\
 + \quad \quad \quad 1 \\
 \hline
 0\text{CB4}
 \end{array}$$

H	e	l	l	o		w	o	r	l	d	.
48	65	6C	6C	6F	20	77	6F	72	6C	64	2E

$4865 + 6C6C + 6F20 + 776F + 726C + 642E + \text{carry} = 71FC$

Puede ser suma simple o OR-Exclusive

Sistemas numéricos, relaciones.

Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Hexadecimal	0	F	1	4
Binario	0000	1111	0001	0100

Este sistema no puede detectar errores si cambian cuatro o más bits en el grupo transmitido. En el siguiente ejemplo se cambiaron los cuatro bits de la segunda columna.

0001	1	0011	3
0010	2	0000	0
0011	3	0001	1
0001	1	0010	2
0111	7	0111	7

3. Errores de transmisión y detección de errores.

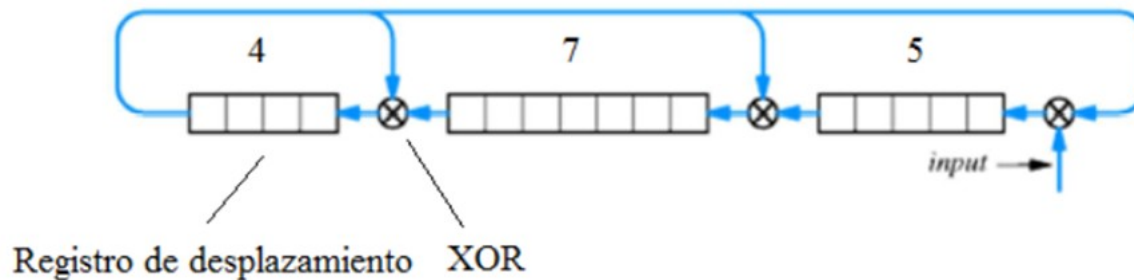
d) Chequeos de Redundancia Cíclicas, CRC o FSC.


Se usa en **transmisiones en las redes**, y las **transferencia de archivos entre discos** de las computadoras, y permite detectar los **errores en ráfagas**, (burst), que se producen cuando un **efecto electromagnético** afecta la transmisión. Se usan **en conjunción** el sistema de **detección de paridad** y el sistema de **comprobación de redundancia cíclica**.

Implementación en hardware:

- Este ejemplo está compuesto de tres Registros de desplazamiento, de cuatro, siete y cinco bits cada uno. Lo que conforma una salida de dieciséis bits. ($4+7+5=16$).
- Se hacen ingresar por la entrada, (input), todos los bits de la trama, lo que va modificando la salida de dieciséis bits. Al ingresar el último bit de la trama, queda conformada la salida con una palabra de dieciséis bits.

Ejemplo de Salida: $\begin{matrix} 4 & 7 & 5 & = & 16 \\ S=1001 & 0000001 & 00001 & & S=1001000000100001 \end{matrix}$





A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Explicación en decimal: (Esta es una explicación para comprender el proceso aproximado, pero no es cierto en todos los casos en el sistema decimal).

D = Datos

P = Polinomio que se multiplica por $D = 10^1 = 10$

G = Generador

R = Resto

D = 13

P = 10

G = 4

----- Transmisor -----

$$\begin{array}{r|l} 130 & 4 \\ 10 & \\ \hline 2 & 32 \end{array}$$

$$D \times P = 13 \times 10 = 130 = DP$$

$$DP / G = 130 / 4 = 32 \text{ con } R = 2$$

Se envía al Receptor la suma $S = DP + R = 130 + 2 = 132$

----- Receptor -----

$$\begin{array}{r|l} 132 & 4 \\ 12 & \\ \hline 0 & 33 \end{array}$$

$$S / G = 132 / 4 = 33 \text{ con Resto } 0$$

Explicación en binario:

Proceso realizado por el Transmisor:

Se interpretan los datos binarios como polinomios y se opera con aritmética binaria módulo 2.

Ejemplo: $D=101110$ $P(X)=X^5+X^3+X^2+X^1$

El polinomio de los Datos se multiplica por 2^3 . Esto sirve solamente para agregar ceros a la derecha de los Datos para incorporar allí el Resto.

Ejemplo: $D=101110$ $2^3 = 1000$ \gggg $D \times 2^3 = 101110 \times 1000 = 1011101000$

Luego se divide el resultado anterior por un polinomio predeterminado denominado Generador, (G),

Ejemplo: $G=1001$

Lo que nos permite obtener el Resto de la división, que es lo que importa en este proceso.

Finalmente se envía $(D \times 2^3 + R)$, donde R es el resto de la división, que se incorpora justo en los tres bits de la derecha.

Proceso realizado por el Receptor:

El receptor divide lo enviado por el transmisor por el polinomio predeterminado llamado generador, G.

$(D \times 2^3 + R) / G \ggggggg$ Resto = 0

Ejemplo de cálculo de CRC.

- Considere:

- Mensaje: $D = 101110$ (6 bits) $\Rightarrow 2^6$
- Polinomio codificador o Generador: $G = 1001$ (4 bits) $\Rightarrow 2^4$
- Resto por ser calculado. $R = ???$ (3 bits) $\Rightarrow 2^3$

- Pasos:

En el transmisor:

- El mensaje se multiplica por 2^3 : $D * 2^3 = 101110 * 1000 = 101110000$

Porque elegimos el Resto con 3 bits $\Rightarrow 2^3$ (Siempre es uno menos que el grado del polinomio generador que es de 2^4)

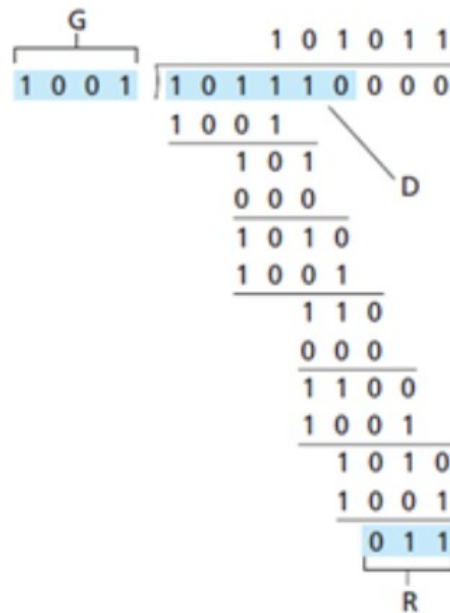
Esto equivale a agregar tres ceros a la derecha de D, en aritmética binaria módulo 2.

$$\begin{array}{r} 101110 \\ * \quad 1000 \\ \hline 000000 \\ 000000 \\ 000000 \\ 101110 \\ \hline 101110000 \end{array}$$

- El producto se divide por G: $(D * 2^3) / G = 101110000 / 1001$

– El producto se divide por G:

$$(D * 2^3) / G = 101110000 / 1001$$



Se utiliza operaciones XOR para las sumas.

es decir si son iguales es cero
si son distintos es uno.

$$1 + 1 = 0 \quad \text{y} \quad 0 + 0 = 0$$

$$1 + 0 = 1 \quad \text{y} \quad 0 + 1 = 1$$

es decir $\begin{matrix} == 0 \\ \neq 1 \end{matrix}$

– El resto es sumado al producto $D * 2^3$:

$$D * 2^3 + R$$

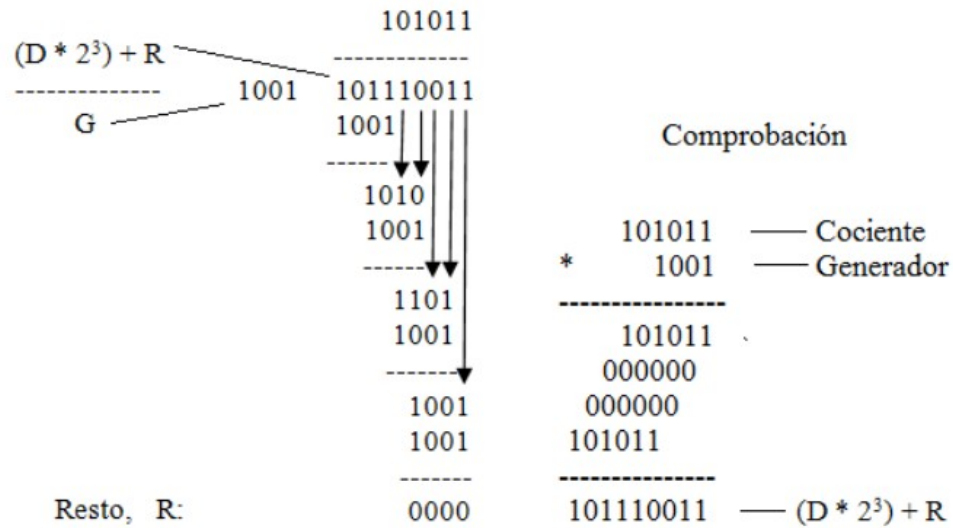
$$\begin{array}{r}
 101110000 \\
 + \quad \quad 011 \\
 \hline
 101110011
 \end{array}$$

Esto se envía al transmisor



En el receptor:

- El receptor efectúa la división de lo que recibió: $(D * 2^3 + R)$ por G



Y detectará la presencia de errores cuando al hacer esta división el resto resulta **no nulo**.