

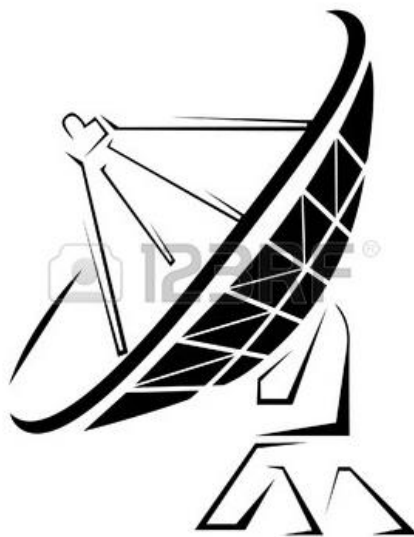


UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE TELECOMUNICACIONES

COMUNICACIÓN DIGITAL



PROFESOR:

Ing. Juan Pablo Pallo Noroña, Mg.

CAPÍTULO IV

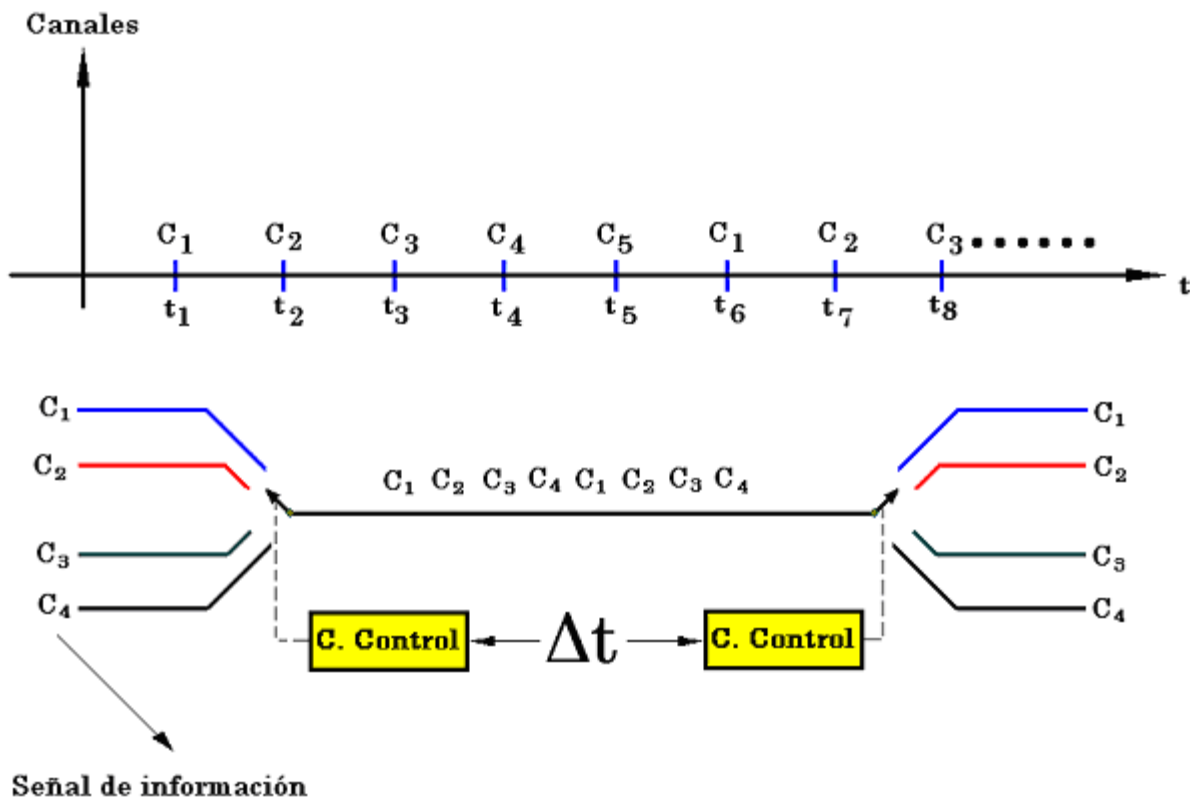
CODIFICACIÓN DE CANAL

- Un canal se utiliza para transmitir información.
 - Un canal \rightarrow I.T.C. (Intervalo de Tiempo de Canal) \rightarrow 8 bits
- Existen dos tipos de codificación por canal:

- TDM (Multiplexación por División de Tiempo).
- PCM (Múltiplexación por División de Pulsos) MIC en español.

a. TDM (Multiplexación por División de Tiempo).

Consiste en asignar todo un ancho de banda a una determinada señal durante un intervalo de tiempo.



La agrupación de las señales \rightarrow escalonada $\left\{ \begin{array}{l} 30 \text{ canales} \rightarrow \text{Europa} \\ 24 \text{ canales} \rightarrow \text{E.U. Japón} \end{array} \right.$

❖ **Multiplexado MIC.**

Hace referencia a que se combina con la señal MIC o PCM, con lo que forma una salida multiplexada digital.

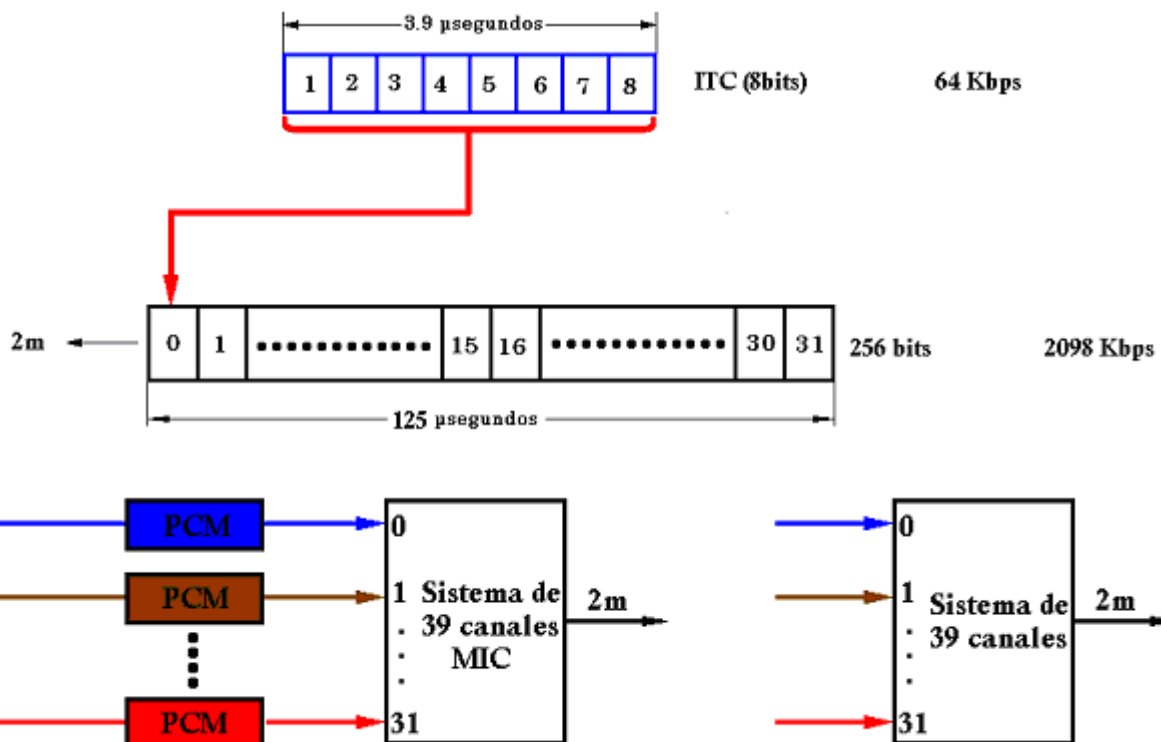
❖ **Multiplexado de la señal digital.**

Agrupar señales de entrada, de baja velocidad binaria en una señal digital con velocidad binaria alta.

1. Sistemas de 30 Canales (E1).

CARACTERÍSTICAS:

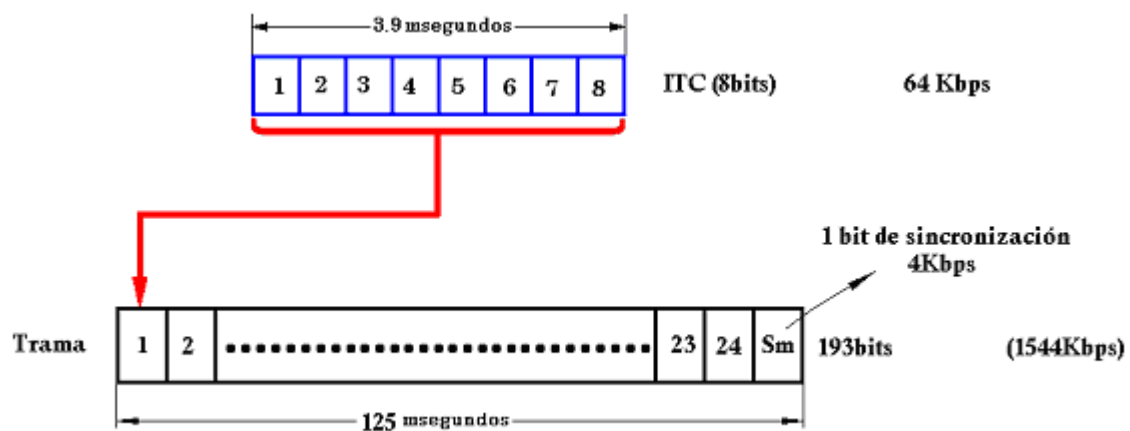
- Cada intervalo de tiempo de canal tiene 8 bits.
- Tiene 30 canales para transmitir información.
- Tiene 1 canal para transmitir señales de sincronización (bit 0).
- Tiene 1 canal para transmitir señalización (bit 16).
- A cada canal se le asigna un intervalo de tiempo de canal (I.T.C.)



2. Sistema de 24 Canales T1.

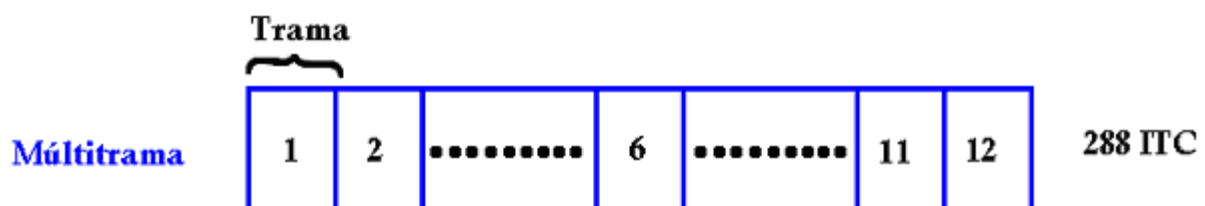
CARACTERÍSTICAS:

- Tiene 24 intervalos de tiempo de canal para transmitir información.
- En el bit número 8 viaja la señal de señalización.



- La trama 6 y 12 en cada ITC, son los encargados de llevar la señalización por lo tanto en una múltitrama la señalización tiene:

$$24 * 2 = 48bits$$



3. SISTEMA DE 32 CANALES

Tenemos 32 intervalos de tiempo de canal (I.T.C), de los cuales 30 se utilizan para transmitir información y 2 intervalos se utilizan para transmitir información del sistema.

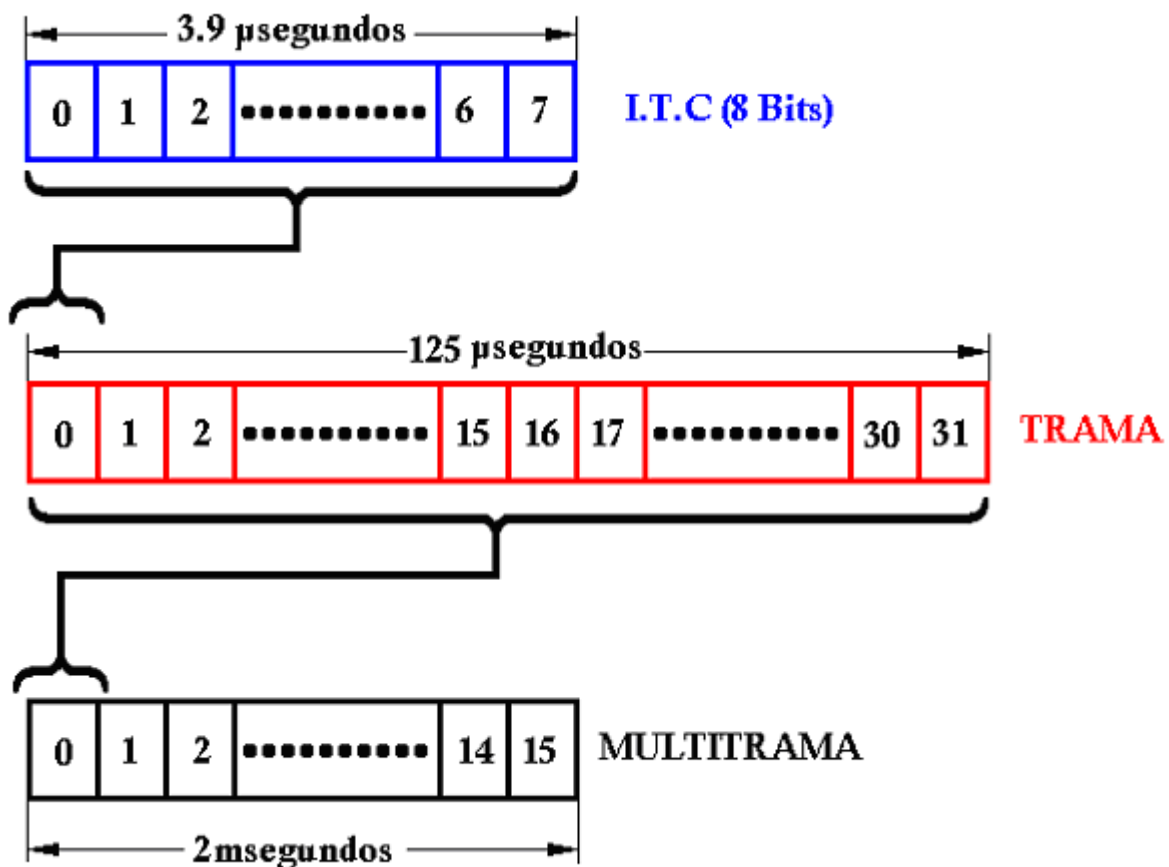
CARACTERÍSTICAS:

- La Multitrama tiene 512 canales y 4.096 bits.
- Tiene 480 canales de información
- Tiene 16 canales de sincronización
- Tiene 16 canales de señalización
- Tiene 3.840 bits de La trama tiene 32 canales y en cada una de ellas 8 bits

$$32 * 8 = 256 \text{ bits}$$

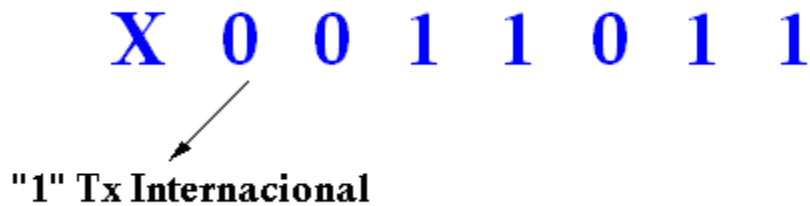
- El canal 0 es el canal de sincronización mientras que el canal 16 es el canal de señalización e información.

La sincronización es para que en los lados de transmisión y recepción, lo que va por el canal 0 llegue a canal 0 y así sucesivamente.



PALABRA DE SINCRONIZACIÓN O PATRÓN.

Se utiliza en ciertos sistemas antiguos.

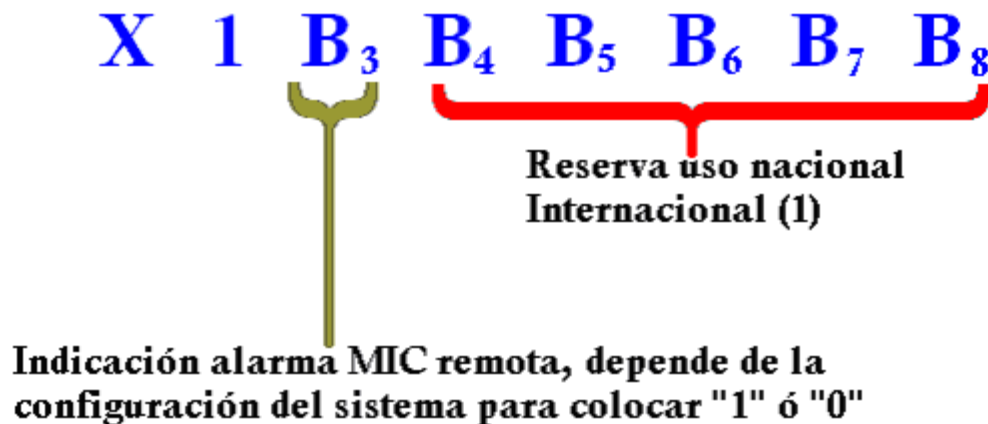


X → Se deja libre para transmisión internacional.

Si se "1" transmisión internacional

Si se "0" transmisión nacional.

Existe otra palabra de sincronización que permite mayor maniobrabilidad.



X → Alarma PCM (equipo central)

El valor "1" es fijo.

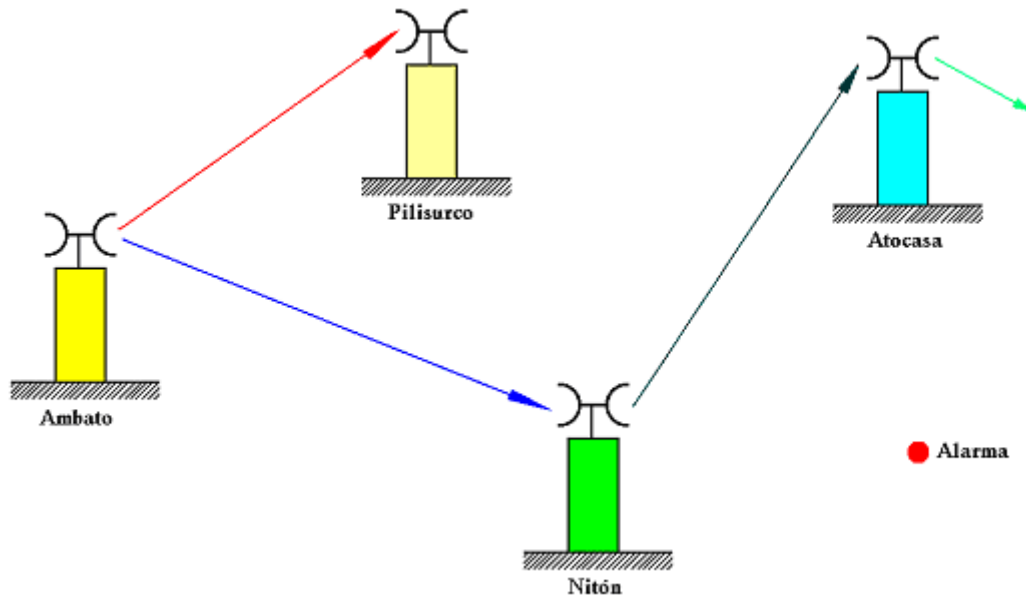
B3 → Indicación de MIC remoto (Lo pone la alarma)

B4, B5, B6, B7 y B8 es reservado para uso nacional así mismo las empresas lo determinan que combinación de bits lo pone la empresa manda en secuencia de bits a la superintendencia, y si es para uso internacional todos los bits deben ser "1".

SEÑALIZACIÓN.

Señal que guía a la señal de información la ruta o camino correcto para llegar desde el transmisor al receptor.

Esta no nos permite entregar mucha información por lo que la CCITT normaliza lo siguiente.



La señalización nos busca la ruta o camino correcto para que lleguen los datos en buen estado.

TIPOS DE SEÑALIZACIÓN:

- ❖ **Canal común.**- Viaja por el mismo canal que la información.
- ❖ **Canal Asociado.**- La señal de señalización viaja por su lado independiente.

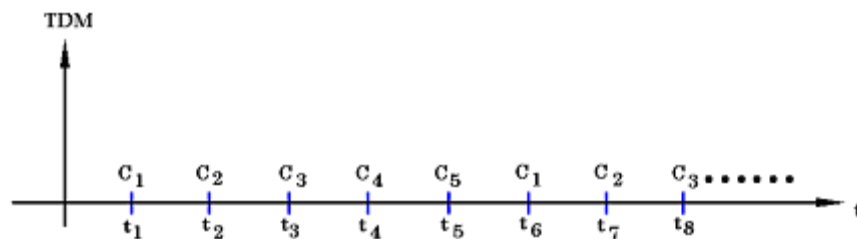
Cuadro Resumido:

Orden	E_1		T_1	
	ITC	$V_{Tx(k)}$	ITC	$V_{Tx(k)}$
1	30	2048	24	1544
2	120	8448	96	6312
3	480	34368	672	44736
4	1920	139264	4032	274176

TÉCNICAS DE MULTIPLEXACIÓN Y ACCESO MÚLTIPLE

1. SISTEMA TDM (MÚLTIPLEXACION POR DIVISIÓN DE TIEMPO)

A cada señal se le asigna un cierto intervalo de tiempo (para transmitir)



Normalizamos los tiempos para a cada señal se le asigne el mismo.
Para un sistema E1 tenemos:

Cuadro Resumido:

Orden		
	# de canales	Velocidad
1	30	2048Kbps => 2Mbps
2	120	8448Kbps => 8Mbps
3	480	34368Kbps => 34Mbps
4	1920	139264Kbps => 40Mbps

MANERAS DE TDM

- **ENTRELAZADO DE PALABRA DE CÓDIGO**

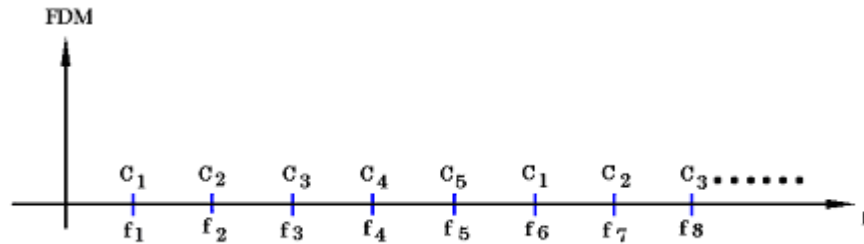
La señal 1 enviamos 8 bits
La señal 2 seguida de 8 bits y así sucesivamente
Grupo de bits de 8 en 8 bits
Toma el grupo de 8 bits de cada señal.

- **ENTRELAZADO DE 8 BITS**

De la señal 1 enviamos 1 bit de la señal 2 enviamos 1bit y así sucesivamente bit a bit.
Tomamos 1 bit de cada señal.
Para el sistema T1 tenemos:

Orden		
	# de canales	Velocidad
1	24	1544Kbps
2	96	6312Kbps
3	672	44736Kbps
4	4032	274176Kbps

2. SEÑAL FDM (MÚLTIPLEXACION POR DIVISIÓN DE LA FRECUENCIA)



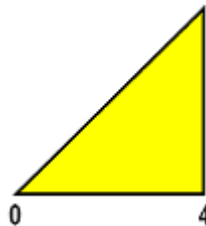
La división se lo hace en base al medio de transmisión.

1. GRUPO BÁSICO PRIMARIO

Significa que podemos multiplexar 12 canales de voz es decir podemos multiplexar 12 llamadas simultáneamente.

a.-12 KHZ----- 60 KHZ

b.-60 KHZ-----108 KHZ



➤ TÉCNICAS

TÉCNICA DEL PREGRUPO

Antes de tener los 12 canales agrupamos de 3 en 3 canales.

Para obtener la banda lateral inferior restamos; o para obtener la banda lateral superior Sumamos. Todas las frecuencias están dadas en Kiloherztz.

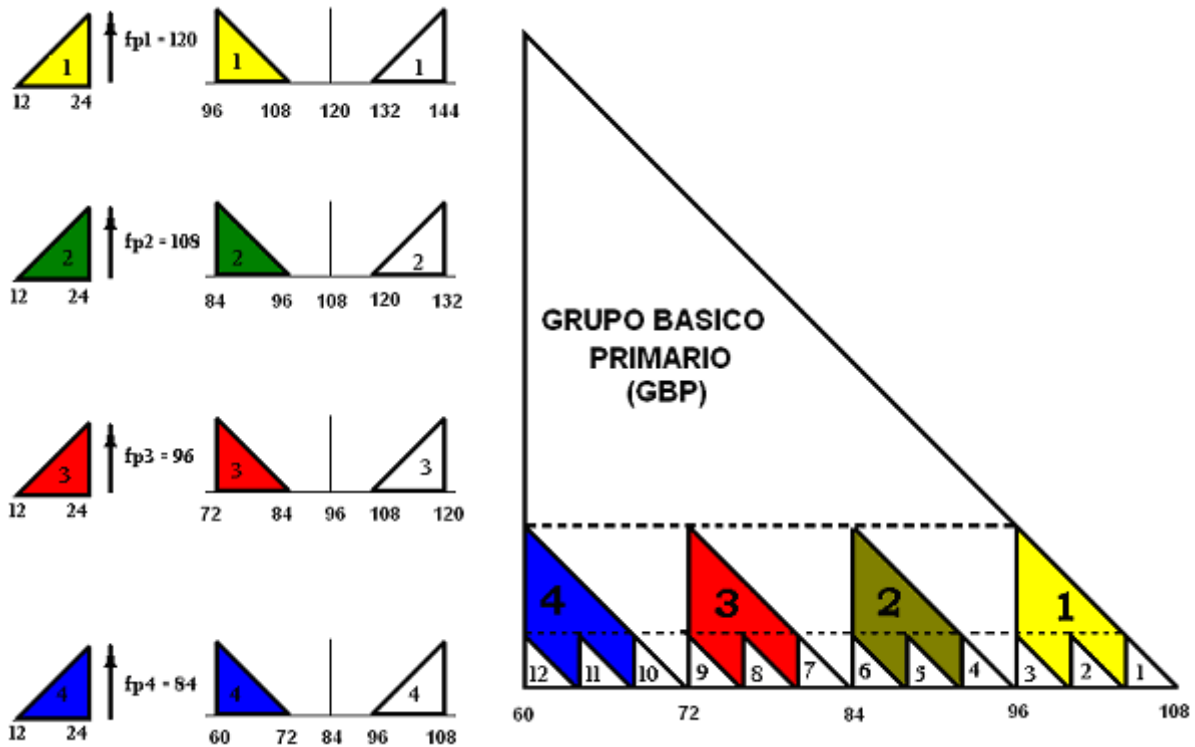
1. GRUPO BÁSICO PRIMARIO

Para obtener el grupo básico primaria agrupamos en 4 pregrupos.

$$\begin{array}{lll}
 \text{CT}_{1, 4, 7, 10} & \uparrow & f_p=12 \\
 \text{CT}_{2, 5, 8, 11} & \uparrow & f_p=16 \\
 \text{CT}_{3, 6, 9, 12} & \uparrow & f_p=20
 \end{array}$$

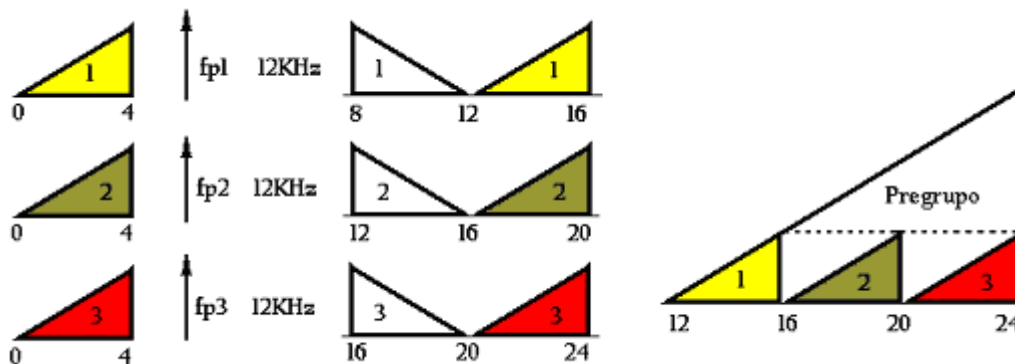
Rango de frecuencia de 60 KHz a 108KHz.

Ancho de banda 48 KHz.



a. PREGRUPO

Se basa en tener tres canales de 4Khz cada uno, y multiplexarlo con una frecuencia estándar de 12Khz. Esta técnica se llama pregrupo.

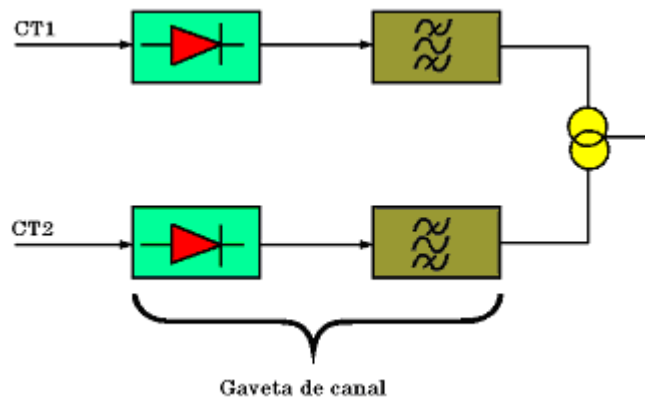


GAVETA DE CANAL

Lugar físico donde se encuentra los circuitos de modulación y de filtrado.

Una de las ventajas es que es una técnica económica.

- ✓ La relación señal ruido disminuye.
- ✓ Confiabilidad del sistema



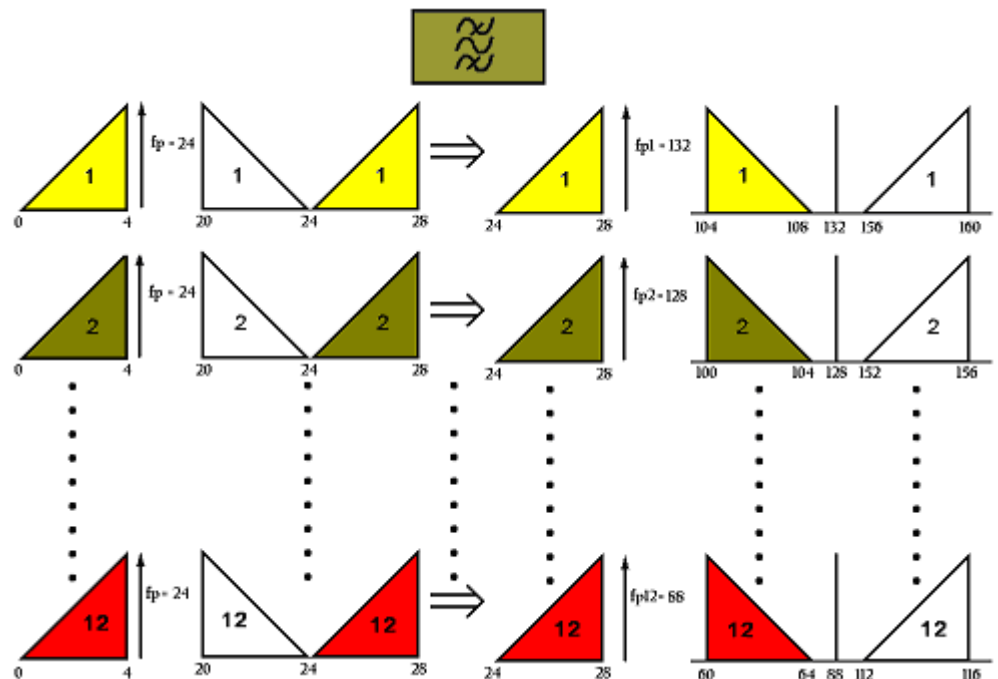
Parte Técnica.

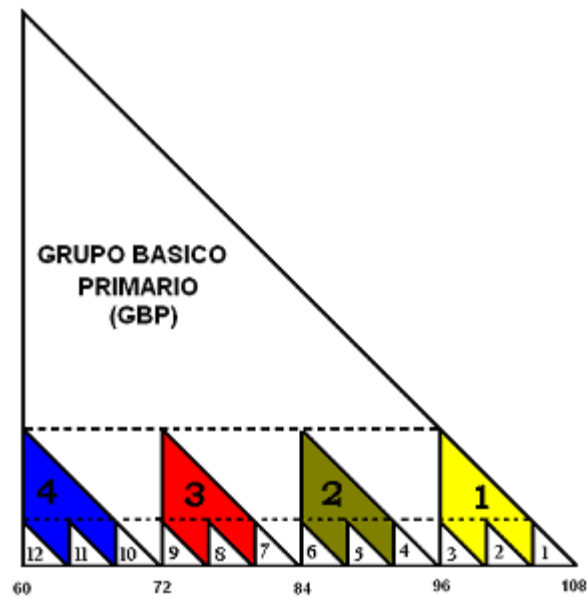
- Hay menor cantidad de repuestos.

b. PREMODULACIÓN

Características

- Tiene dos modulaciones.
- En la primera modulación a todos los canales se la modulan con la misma frecuencia.
- La segunda modulación se le hace a diferentes frecuencias





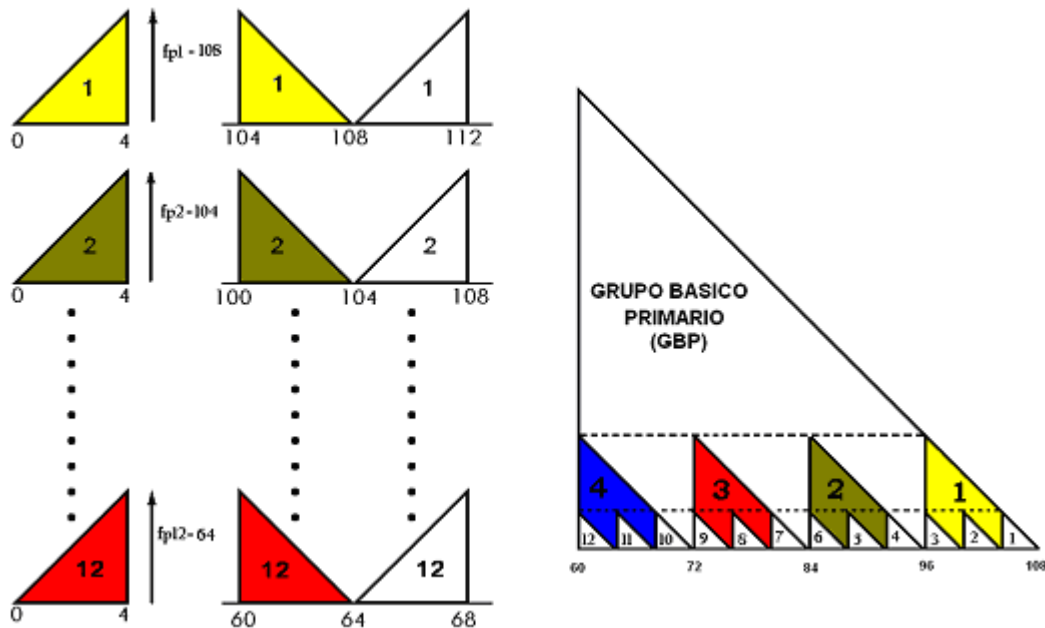
Parte Técnica

- Necesitamos un solo repuesto.

c. MODULACIÓN DIRECTA

Características

- La relación señal ruido comparado con la técnica anterior mejora, pero los costos se incrementa.



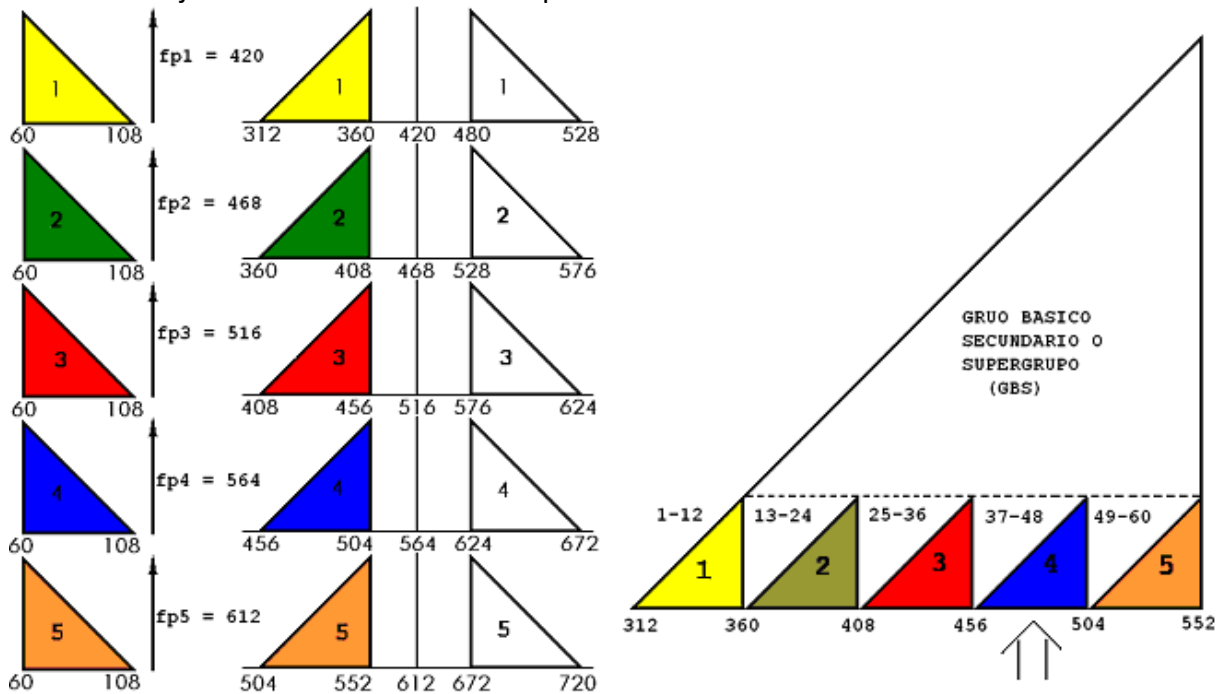
Parte Técnica

- Solo necesita 12 repuestos.
- Tiene una buena relación señal/ruido (S/R)

2. GRUPO BÁSICO SECUNDARIO O SUPERGRUPO

Características.

- Vamos a tener 60 canales telefónicos por lo tanto tenemos 5 grupo básicos primarios (GBP).
- Ancho de banda 240 KHz.
$$GBS = 5 * GBP = 5 * 48 = 240$$
- Rango de frecuencia de 312 KHz a 552 KHz
$$240 + 312 = 552$$
- Trabajamos en la banda lateral superior.



¿En que canal esta el canal 40 y en que frecuencia?.

Cada canal tiene 12 canales

$$12 * 4 = 48$$

Por lo tanto esta en el canal 4.

37 → 456 – 460

38 → 460 – 464

39 → 464 – 468

40 → 468 – 472

Por lo tanto está en la frecuencia es de. 468 → 472

GRUPO BASICO PRIMARIO	# De canales
1	1-12
2	13-24
3	25-36
4	37-48
5	49-60

3. GRUPO BÁSICO TERCEARIO

Características

Vamos a tener 300 canales.

$$5 * 60 = 300$$

Utilizamos 5 grupos básicos secundarios.

$$GBT = 5 * GBS = 5 * 5GBP = 5 * 5 * 48 = 1200KHz$$

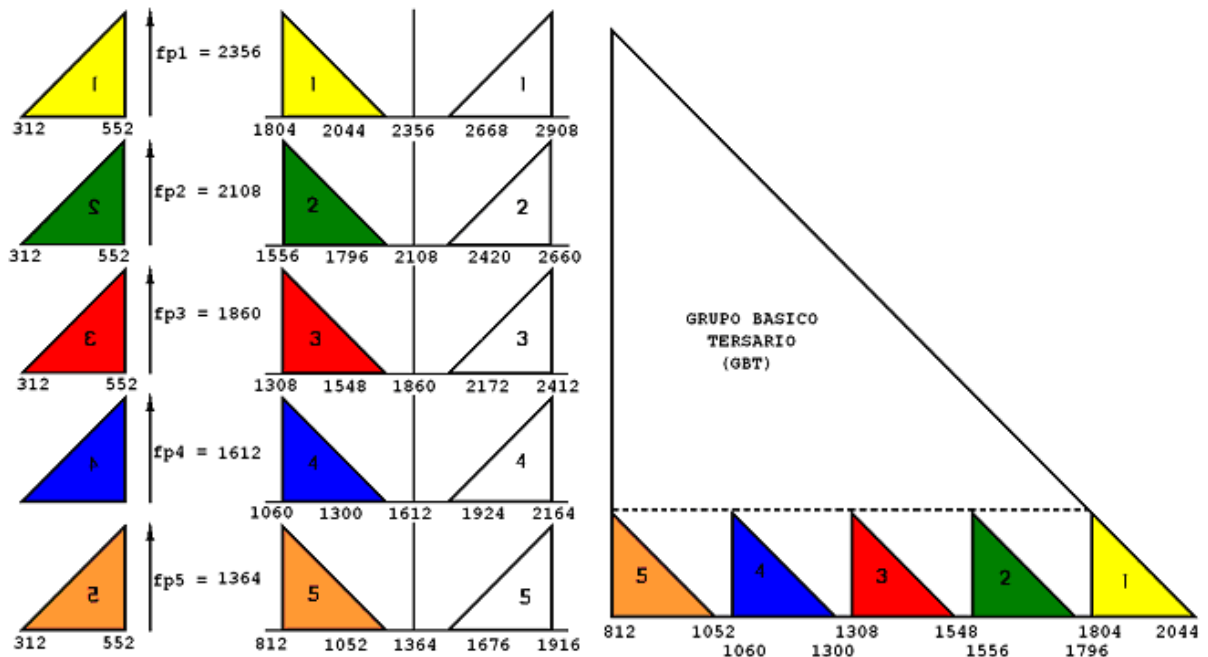
Con un ancho de banda de 1200 KHz $AB = 1200 + 4(8) = 1.232$ KHz

Cuando se trabaja con bandas anchas muy grandes se habla que se tiene Guarda Bandas.

Guarda Bandas.- Guarda Banda es el espacio que hay entre las bandas para seguridad de transmisión.

Hay Guarda Bandas de 8, 16, 32 KHz,

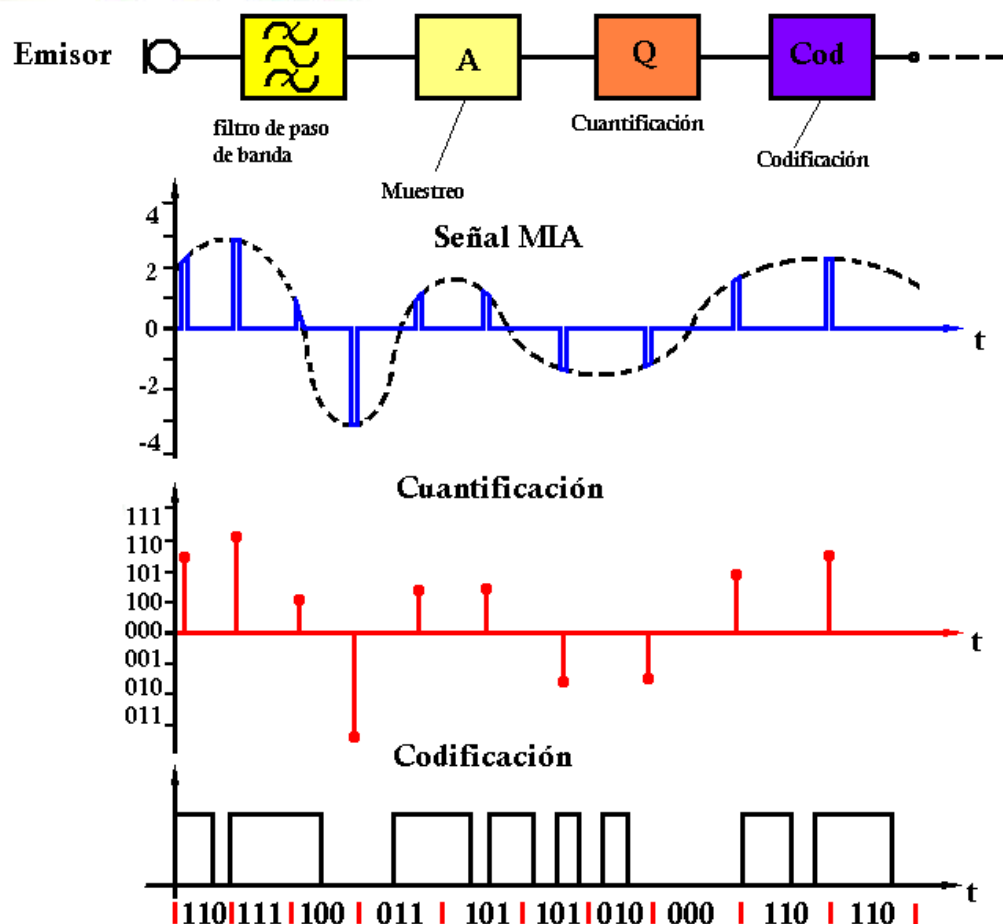
Utilizamos la banda lateral izquierda.



GRUPO BASICO SECUNDARIO	# De canales
1	60- 1
2	120- 61
3	180-121
4	240-181
5	300-241

SISTEMA MIC

La señal MIC modulada en amplitud y considerada hasta este momento es todavía una señal analógica; precisamente con un proceso de cuantificación se convertirá en una señal digital. Durante el proceso de cuantificación se asignan a los impulsos de amplitud de la señal MIA un número limitado de intervalos de cuantificación discretos. Para ello se dividen el margen de amplitudes de la señal en una cantidad igual de intervalos.



En el caso de que un valor de amplitud se encuentra situado entre los límites de un intervalo, se le asigna a este valor mencionado. En el punto de recepción y para que el error de cuantificación permanezca lo menor posible, se reproduce este valor con un valor analógico que esté situado en la mitad (en el centro) de este intervalo de cuantificación.

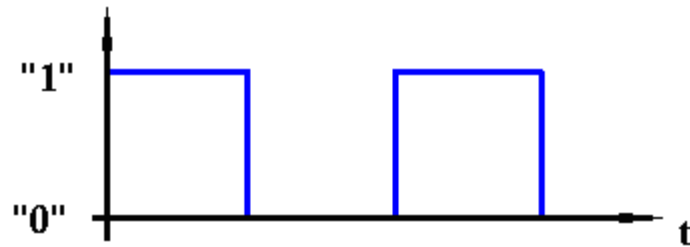
Por codificación se sobre entiende un convenio por el cual cada uno de los intervalos individuales se enumeran correlativamente con el código binario apropiado.

La secuencia de muestreo, cuantificación y codificación se resume bajo el concepto general de conversión analógica-digital (A/D).

CARACTERÍSTICAS

1. CONVERSIÓN DE POLARIDAD.

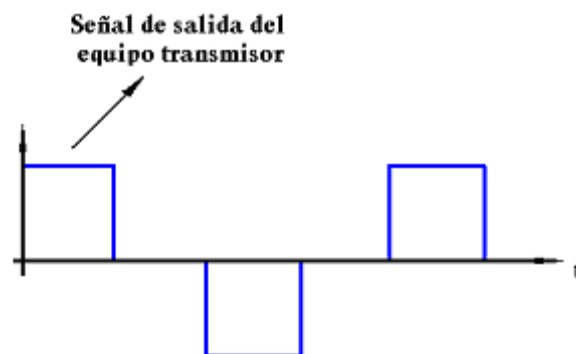
Para transmitir necesitamos pulsos bipolares (positivo y negativo), por eso los pulsos binarios los pasamos a un código binario de transmisión.



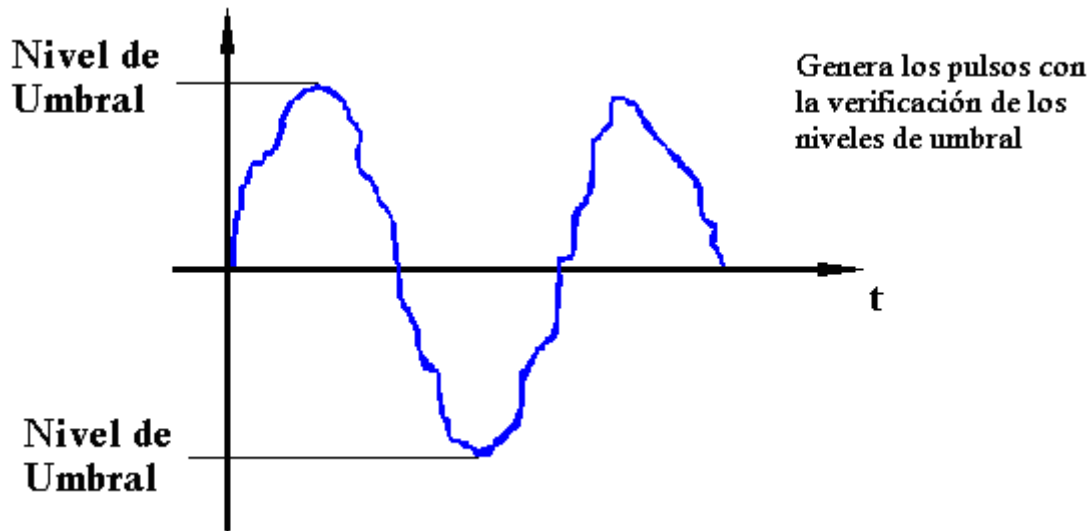
2. REGENERACIÓN.

En los equipos MIC, la señal que se transmite sufre distorsiones, variaciones por el ruido a lo largo del medio de transmisión, en el trayecto se ubican repetidores o regeneradores.

Existen dos tipos de regeneradores: repetidor pasivo y repetidor activo.

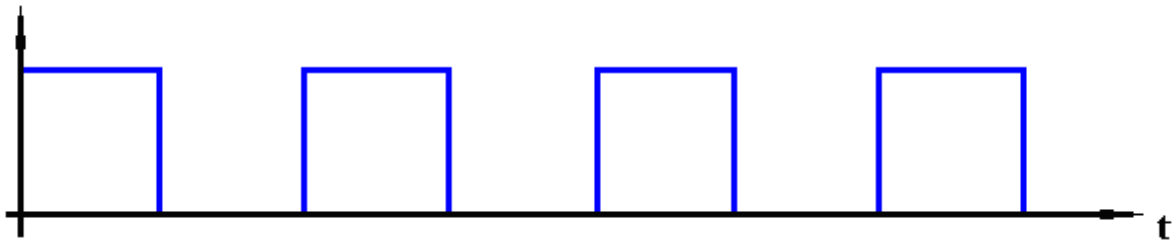


- **Repetidor Pasivo.** - Capta la señal, no verifica, solamente amplifica la señal para transmisión en una distancia determinada.
- **Repetidor Activo.** - Analiza la señal, si puede eliminar el ruido, no retransmite si tiene demasiados errores, envía un mensaje al transmisor luego de analizar retransmite amplificando, también no puede transmitir cuando existe demasiados bits de error.



3. **SINCRONIZACIÓN.**- Al hablar de sistema MIC es importante hablar de sincronización. En este sistema se trabaja con circuitos digitales que están sincronizados, o sea tienen una señal de reloj. Existen tres tipos de sincronización:

- a) **Sincronización de dígito o bit.**- Se refiere a lo que es la sincronización del generador de la señal de reloj necesario para el codificador tanto en el transmisor como en el receptor.



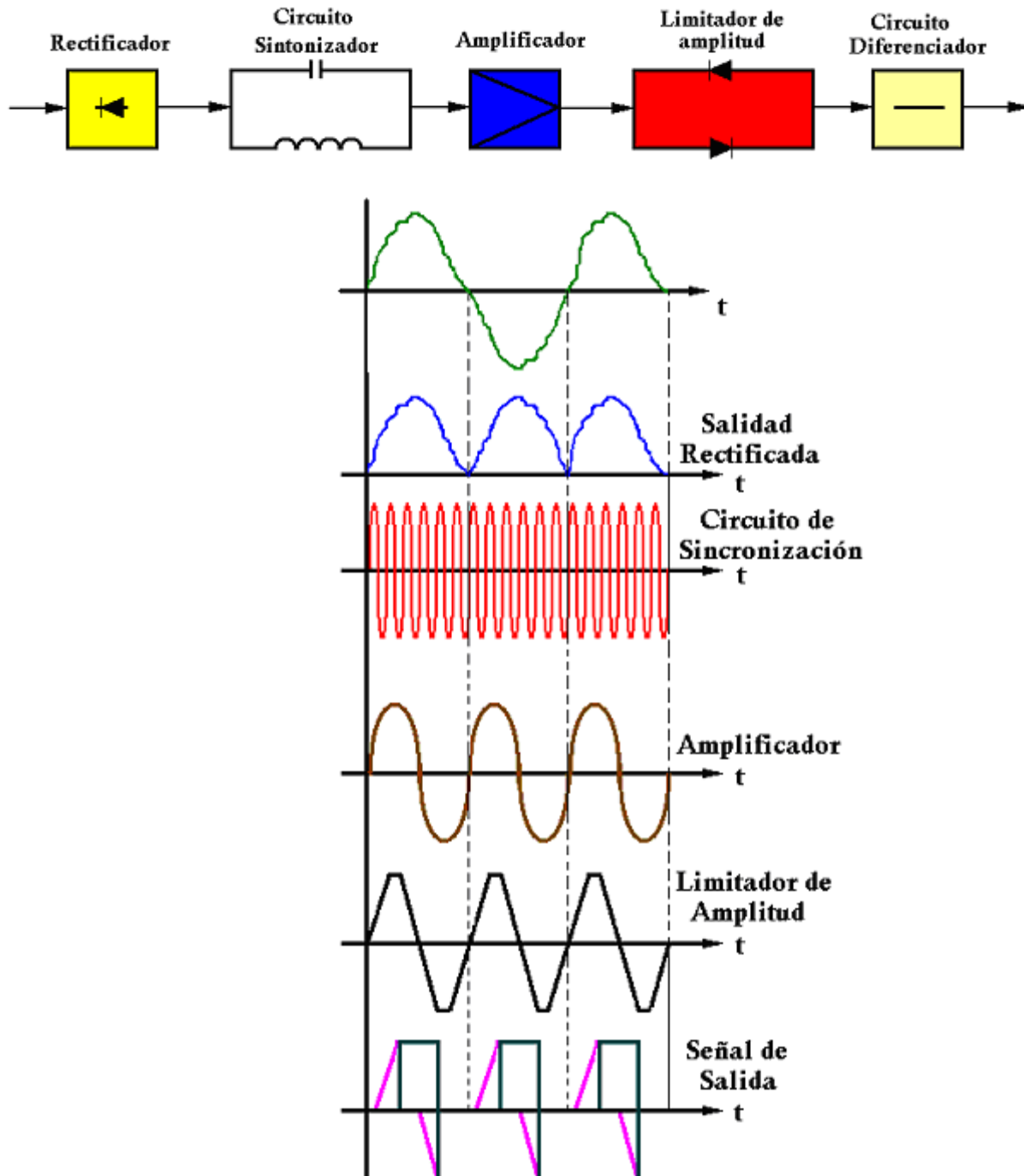
Se puede extraer la frecuencia y la fase de la señal de reloj, también cuando se envía mayor información se puede extraer lo antes mencionado.

Existen dos tipos:

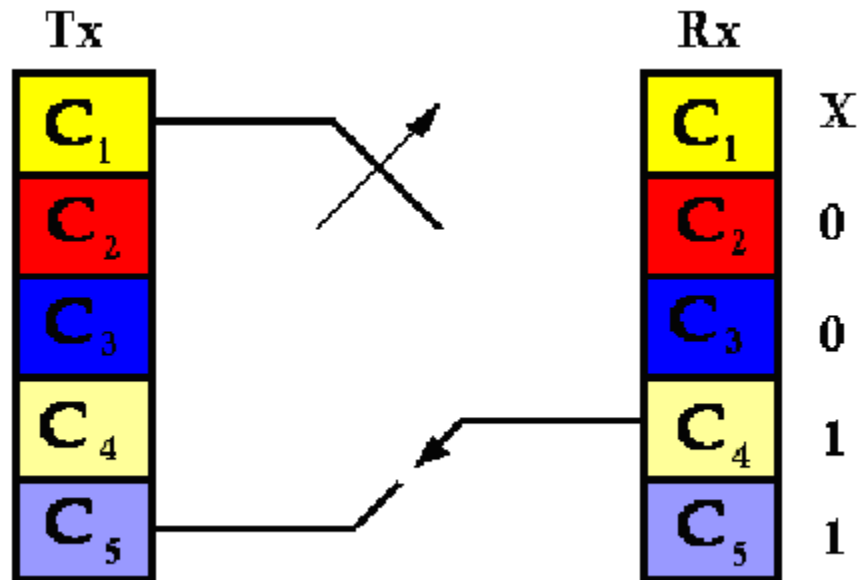
- I. **Sincronización Interna.**- La información de la señal de reloj se transmite en los pulsos de la señal MIC.

II. Sincronización Externa.- La señal de reloj se transmite vía independiente de la señal MIC.

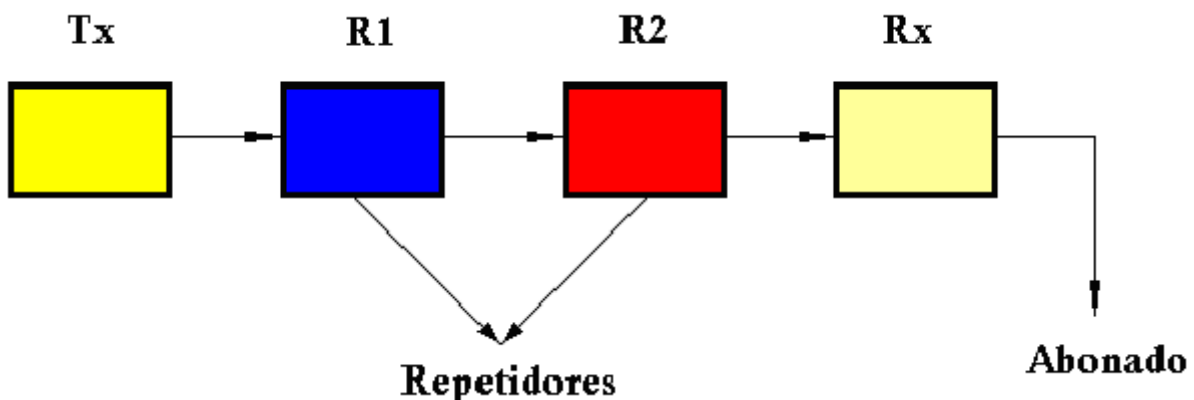
CIRCUITO DE TEMPORIZACIÓN



- b) **Sincronización de trama.**- Se refiere a la operación para encontrar una fase correcta que discrimina el principio y fin de una trama; de tal manera que sin transmisión se tiene el canal uno conectado al medio de transmisión en el terminal de recepción de debe tener la conexión al mismo canal.



- c) **Sincronización de la red.**- Debe tener una sincronización entre los equipos para que la calidad de la comunicación sea aceptable. La precisión del reloj debe ser adecuada, caso contrario habrá deslizamiento de bits (esto genera ruido dependiendo del caso), la frecuencia de operación de los relojes debe ser la misma. Los equipos son diseñados para trabajar con cierta tolerancia en la frecuencia.



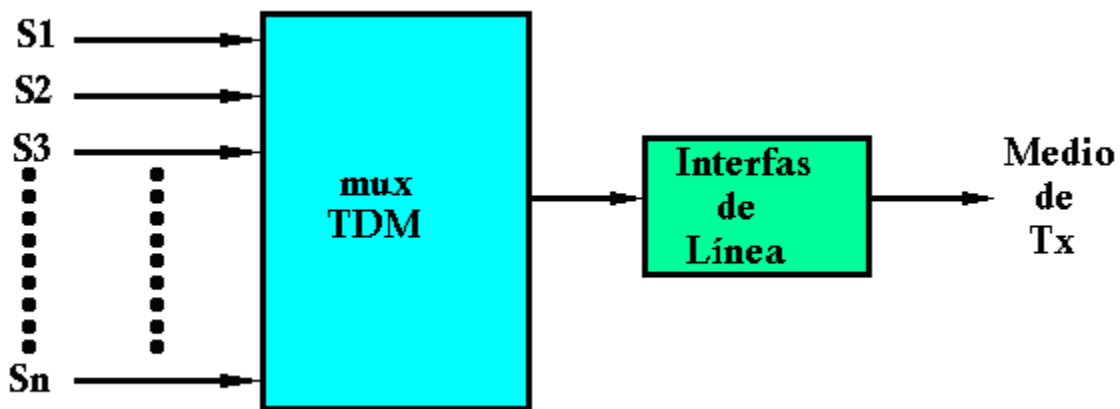
Precisión.- Es el grado de error cuando se lo compara después de un intervalo de tiempo con un reloj patrón.

Estabilidad.- Se refiere a que intervalo de tiempo el reloj puede seguir generando la misma frecuencia. Existen dos tipos:

- a. **Estabilidad a corto plazo.** - Se refiere a la variación aleatoria de la frecuencia de la señal de reloj.
- b. **Estabilidad a largo plazo.** - Depende del cambio sistemático de la frecuencia.

INTERFACE DE LÍNEA

Es un circuito que adapta la señal digital a las características del medio de transmisión



Características:

- No debe tener componentes continuos
- El esquema del código debe ser transparente para todas las señales binarias.
- Las condiciones de señal de invalidez deben ser fácilmente detectables.
- La señal codificada debe ser decodificada de manera cíclica en el receptor.

PROTECCIÓN DE LOS DATOS: CONTROL DE ERRORES.

PROTECCIÓN DE LOS DATOS.

Los datos, cuya utilidad radica en su **integridad** y en su **confidencialidad**, están sujetos a dos tipos de amenazas:

ERRORES

Las causas por las que la señal eléctrica se deteriora al viajar por el canal de comunicación, ellas son: distorsión, atenuación, limitación del ancho de banda, ruido, interferencia y diafonía. Esta degradación de la señal puede hacer que recibamos en el receptor un caracter distinto al que fue emitido por el extremo transmisor, diremos entonces que se ha producido un error.

Si bien es imposible evitar que ocurran errores, un buen diseño los minimizará. Hecho esto la tarea es en primer lugar determinar la presencia de los errores, aquí es donde aparecen las técnicas de **detección de errores**, y luego tratar de corregirlos, lo que da lugar a la **corrección de errores**, la denominación genérica de estas técnicas es **Control de Errores**.

ACCIONES NO AUTORIZADAS

Ya se ha hablado extensamente sobre el valor de **la información**, asegurar que esta información:

- ✓ **No sea vista o copiada por personas no autorizadas ó no calificadas para ello.**
- ✓ **No sea alterada en ningún sentido (modificada, destruida, alterados los equipos donde reposa, etc) por personas ó máquinas no autorizadas.**
- ✓ **No sea creada y/o difundida engañosamente simulando fuentes reales ó inexistentes.**

Esto de **vital** importancia dado el auge de Internet y de las conexiones en red y entre redes.

De ello se ocupa la **Seguridad en Redes**.

CONTROL DE ERRORES

DETECCIÓN DE ERRORES.

La **detección de errores** consiste en monitorear la información recibida y a través de técnicas implementadas en el Codificador de Canal ya descrito, determinar si un carácter, caso asincrónico, ó un grupo de datos, caso sincrónico, presentan algún ó algunos errores.

Las técnicas más comunes son:

- ✓ **Redundancia.**
- ✓ **Codificación de cuenta exacta.**
- ✓ **Chequeo de paridad vertical (VRC).**
- ✓ **Chequeo de paridad horizontal (LRC).**
- ✓ **Chequeo de paridad bidimensional (VRC/LRC).**
- ✓ **Checksum**
- ✓ **Chequeo de redundancia cíclica (CRC).**

REDUNDANCIA.

La **redundancia** significa transmitir cada carácter dos o tres veces, o si se emplea a nivel de mensaje repetir el mensaje dos o tres veces, en caso que las versiones difieran habrá

error ó errores. Obviamente la **eficiencia** con este método se reduce a $\frac{1}{2}$ ó $\frac{1}{3}$ según corresponda.

CODIFICACIÓN DE CUENTA EXACTA.

En esta técnica de **codificación de cuenta exacta**, lo que se hace es configurar el código de manera que cada carácter esté representado por una secuencia de unos y ceros que contiene un número fijo de unos, por ejemplo, tres de ellos. Tal es el caso del Código de cuenta exacta ARQ (Requerimiento automático de repetición) que se muestra en la **Tabla 4.1**, en caso de recibirse un carácter cuyo número de unos no sea tres, se tratará de un error. Es claro que este método, al igual que los demás, tiene limitaciones: cuando se recibe un 0 en vez de un 1 y un 1 en vez de un 0 dentro del mismo carácter los errores no serán detectados. Ésta no es la única posibilidad de errores detectados el lector podrá imaginar varias más.

Código Binario								Carácter	
Bit:	1	2	3	4	5	6	7	Letra	Cifra
	0	0	0	1	1	1	0	Cambio a letras	
	0	1	0	0	1	1	0	Cambio a cifras	
	0	0	1	1	0	1	0	A	-
	0	0	1	1	0	0	1	B	?
	1	0	0	1	1	0	0	C	:
	0	0	1	1	1	0	0	D	(WRU)
	0	1	1	1	0	0	0	E	3
	0	0	1	0	0	1	1	F	%
	1	1	0	0	0	0	1	G	@
	1	0	1	0	0	1	0	H	£
	1	1	1	0	0	0	0	I	8
	0	1	0	0	0	1	1	J	(campana)
	0	0	0	1	0	1	1	K	(
	1	1	0	0	0	1	0	L)
	1	0	1	0	0	0	1	M	.
	1	0	1	0	1	0	0	N	,
	1	0	0	0	1	1	0	O	9
	1	0	0	1	0	1	0	P	0
	0	0	0	1	1	0	1	Q	1
	1	1	0	0	1	0	0	R	4
	0	1	0	1	0	1	0	S	'
	1	0	0	0	1	0	1	T	5
	0	1	1	0	0	1	0	U	7
	1	0	0	1	0	0	1	V	=
	0	1	0	0	1	0	1	W	2
	0	0	1	0	1	1	0	X	/
	0	0	1	0	1	0	1	Z	+
	0	0	0	0	1	1	1		(blanco)
	1	1	0	1	0	0	0		(espacio)
	1	0	1	1	0	0	0		Alimentar línea
	1	0	0	0	0	1	1		Regreso de línea

Tabla 4.1. Código de cuenta exacta ARQ

CHEQUEO DE PARIDAD VERTICAL Ó PARIDAD DE CARÁCTER (VRC).

Este método, como todos los que siguen, hace uso del agregado de **bits de control**. Se trata de la técnica más simple usada en los sistemas de comunicación digitales (Redes Digitales, Comunicaciones de Datos) y es aplicable a nivel de byte ya que su uso está directamente relacionado con el código ASCII.

Como se recordará, el código ASCII utiliza 7 bits para representar los datos, lo que da lugar a 128 combinaciones distintas. Si definimos un carácter con 8 bits (un byte) quedará un bit libre para **control**, ese bit se denomina **bit de paridad** y se puede escoger de dos formas:

- ✓ Paridad par
- ✓ Paridad impar

Según que el número total de unos en esos 8 bits, incluyendo el octavo bit (el de paridad), sea par ó impar, tal como se muestra en la Figura 4.1. Por sus características la técnica se denomina también paridad de carácter. El uso de un bit adicional para paridad disminuye la eficiencia, y por lo tanto la velocidad en el canal, el cálculo es sencillo pasamos de 7 bits de datos a 7+1, ello conduce de acuerdo a la expresión a un overhead de: $(1 - 7/8)100\% = 12.5\%$ de disminución en la eficiencia.

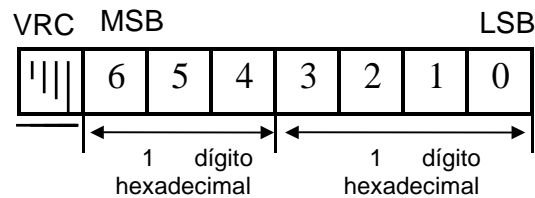


Figura 4.1. Bit de paridad del Código ASCII.

En el extremo de transmisión el Codificador de Canal calcula el bit de paridad y lo adosa a los 7 bits de datos. El Decodificador de Canal recibe los 8 bits de datos calcula la paridad y la compara con el criterio utilizado, tal como describe la **Figura 4.2**.

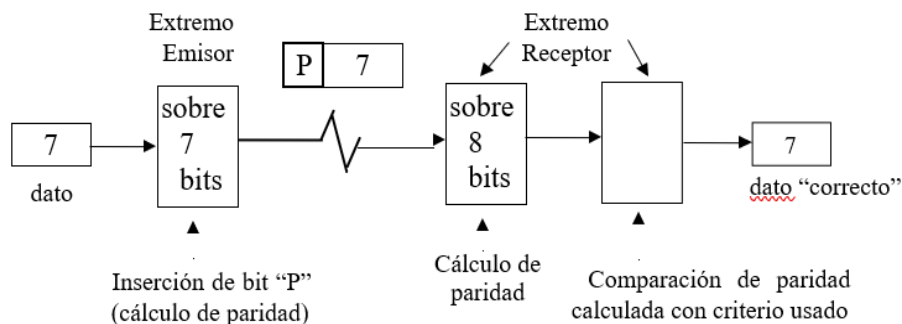
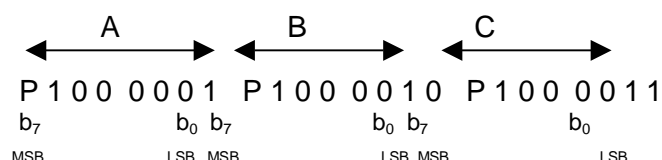


Figura 4.2. Método de chequeo de paridad vertical (VRC).

Este método tampoco asegura inmunidad a errores, basta con que dos bits cambien su valor simultáneamente para que el error no sea detectado pues la paridad será correcta y el dato no. Sin embargo, este sencillo sistema permite que en una línea telefónica discada que transmite entre 10^3 y 10^4 bps con una tasa de error (BER) de 10^{-5} mejore a 10^{-7} .

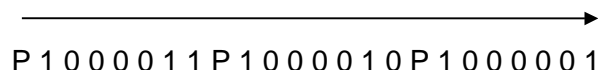
Debe mencionarse que en transmisión serial la interpretación de una secuencia 0 y 1 presenta un problema, **pues el bit menos significativo (LSB) se transmite primero y el más significativo (MSB) de último.**

En una secuencia de tres letras ABC no hay duda de identificar a la A como la primera letra, sin embargo, si escribimos sus códigos ASCII (debe hacerse notar que no todos los autores numeran de b_0 a b_7 algunos lo hacen de b_1 a b_8) con el bit de paridad tendremos:



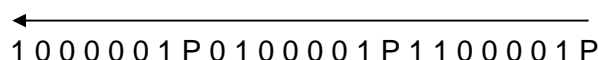
Esta presentación induce a confusión pues no es la de transmisión serial. Tenemos dos alternativas para mejorarla:

Caso 1: flecha a la derecha



el bit del extremo derecho del primer carácter es el primero en ser transmitido (b_0 de A) y el último corresponderá al extremo izquierdo del último carácter (b_7 de C, bit de paridad P de C), la flecha indica el sentido en que fluyen los bits. **Los datos para ser interpretados deben tomarse en grupos de 8 y ser leídos de derecha a izquierda.**

Caso 2: flecha a la izquierda



este caso, común cuando se utiliza un osciloscopio para monitorear líneas de datos ya que el primer bit recibido queda en el extremo izquierdo de la pantalla, la flecha indicará aquí también el sentido en que fluyen los bits, **requiere para interpretar correctamente los caracteres tomar grupos de 8 bits y leerlos de izquierda a derecha.**

Chequeo de paridad horizontal (LRC),longitudinal ó de columna.

Este chequeo de paridad horizontal ó longitudinal (HRC ó LRC) en vez de estar orientado al carácter lo está al **mensaje**, y consiste en que cada posición de bit de un mensaje tiene bit de paridad, así por ejemplo se toman todos los bits b_0 de los caracteres que componen el mensaje y se calcula un bit de paridad par o impar, según el criterio definido, este bit de paridad es el bit b_0 de un carácter adicional que se transmite al final del mensaje, y se procede luego sucesivamente con los demás bits incluyendo el de paridad. El carácter así construido se denomina BCC (Block Check Character), también se le denomina BCS (Block Character Sequence), ver **Figura 4.3.**

COMUNICACIÓN DIGITAL

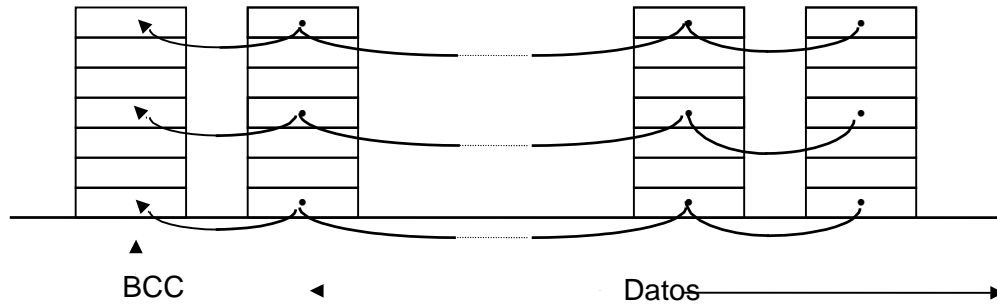


Figura 4.3. Chequeo longitudinal LRC

Históricamente entre el 75 y el 98% de los errores presentes son detectados por LRC, los que pasan desapercibidos se deben a limitaciones propias del método, así por ejemplo un error en b_2 en dos diferentes caracteres simultáneamente produce un LRC válido.

Chequeo de paridad bidimensional (VRC/LRC).

La combinación de los dos métodos precedentes proporciona mayor protección y no supone gran consumo de recursos y, aunque tiene la misma sencillez conceptual de los métodos de paridad lineal, es más complicado y por ello menos popular.

El uso simultáneo de VRC y LRC hace que pasen indetectados errores en un número par de bits que ocupan iguales posiciones en un número par de caracteres, circunstancia muy poco probable. Aunque no es el objeto de esta Sección debe hacerse notar que en caso que se trate de un solo error el uso simultáneo de VRC y LRC permite determinar con precisión **cual es el bit erróneo** y por lo tanto **corregirlo**. Otras combinaciones de errores pueden ser detectadas y algunas además corregidas. Las Figuras 4.4 y 4.5 ilustran algunas circunstancias del chequeo bidimensional.

Paridad de carácter computada en el receptor	Bit #									
	8	7	6	5	4	3	2	1		
0	0	1	0	0	0	0	0	1	A	Carácter #1
0	0	1	0	0	1	1	0	1	M	Carácter #2
1	1	1	0	0	0	1	0	1	E	Carácter #3
1	1	1	0	1	0	0	1	0	R	Carácter #4
1	1	1	0	0	1	0	0	1	I	Carácter #5
1	1	1	0	0	0	0	1	1	C	Carácter #6
1*	0	1	0	1	0	0	0	1	A	Carácter #7
0	0	1	0	1	0	0	0	0		Paridad de columna recibida
1	0	1	0	0*	0	0	0	0		Paridad de columna computada en el extremo receptor

Diferencias de paridad

Figura 4.4 Mensaje con un error en la segunda letra A

Parid. de car. Computada en el receptor	Bit #									
	8	7	6	5	4	3	2	1		
0	0	1	0	0	0	0	0	1	A	Carácter # 1
1*	0	1	0	0	1	1	1	1	M	Carácter # 2
1	1	1	0	0	0	1	0	1	E	Carácter # 3

COMUNICACIÓN DIGITAL									
1	1	1	0	1	0	0	1	0	R Carácter # 4
1	1	1	0	0	1	0	0	1	I Carácter # 5
1	1	1	0	0	0	0	1	1	C Carácter # 6
0	0	1	1	1	0	0	0	1	A Carácter # 7
0	0	1	0	1	0	0	0	0	Paridad de columna recibida
0	0	1	1	0	0	0	1	0	Paridad de columna computada en el extremo receptor
			*	*			*		

Figura 4.5 Mensaje con tres errores, uno es detectado e identificado, los otros dos solo detectados.

CHECKSUMS

Es otro método simple orientado al **mensaje**, en él los valores (por ejemplo, decimales) que corresponden a cada carácter en el código ASCII son sumados y la suma es enviada al final del mensaje. En el extremo receptor se repite el procedimiento de sumar los valores de los caracteres y se compara el resultado obtenido con el recibido al final del mensaje, ver **Figura 4.6**.

<u># Carácter</u>	<u>Carácter</u>	<u>Valor decimal</u>
1	A	65
2	M	77
3	E	69
4	R	82
5	I	73
6	C	67
7	A	65
CHECKSUM		498

Figura 3.6. Método de checksums

CÓDIGO DE REDUNDANCIA CÍCLICA.

Los métodos basados en el uso de paridad son sencillos de comprender y de implementar, suministran cierto grado de protección contra los errores, pero son limitados y su efectividad es cuestionable en determinadas aplicaciones. Por ello se utilizan solamente cuando resulta muy complicado ó muy costoso implementar otros métodos. Además, el de paridad vertical requiere que cada carácter lleve su protección contra errores, lo que lo hace adecuado en entornos asíncronos, en entornos síncronos el uso de tantos bits de detección de errores consume un porcentaje importante de la capacidad del canal y resulta oneroso. Por ello es necesario, en entornos síncronos, emplear métodos que tengan en cuenta dos factores importantes:

1. **Detección más segura de los errores.** Dado que los datos se envían en bloques un solo error corrompe toda la información contenida en él, que es considerable, además muchas veces los errores se presentan en "ráfagas", por ello se requieren esquemas más poderosos
2. **Eficiencia.** No se deben consumir demasiados recursos dejando libre la mayor parte del canal para datos.

Un grupo de métodos que cumplen con dichos requisitos son los llamados **códigos de redundancia cíclica**, que se basan en propiedades matemáticas de los códigos empleados para la transmisión de datos, para dar una idea del método veremos un ejemplo sencillo.

Deseamos transmitir al extremo receptor, mediante un canal de comunicación muy vulnerable a errores, un número. Dadas las circunstancias es muy posible que, si enviamos, digamos el número 23, llegue al extremo receptor un número distinto, una solución es elegir un número

COMUNICACIÓN DIGITAL

clave, por ejemplo, el 5. Ahora dividimos el número a transmitir entre la clave y calculamos el resto: $23/5 = 4$ resto 3 y enviamos conjuntamente con el 23 el resto, o sea, transmitimos 233. En el extremo receptor se efectúa el proceso inverso, supongamos que hemos recibido 253 al dividir $25/5$ el resto es 0 y 0 es distinto de 3 lo que indica error.

Lo mismo se hace en los **métodos de redundancia cíclica**, que están basados en las propiedades de la operación módulo (se define módulo de dos números a mod b al resto de dividir a por b).

Para ello se considera la cadena de bits a transmitir como el conjunto de coeficientes de un polinomio, por ejemplo, si enviamos 1100100110, el polinomio equivalente $P(x)$ es:

$$1x^9 + 1x^8 + 0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 1x + 0$$

Sea

$$1100100110 = P(x) = x^9 + x^8 + x^5 + x^2 + x.$$

Debemos ahora especificar **la clave** para efectuar la división. La selección de esta clave es esencial para la capacidad de respuesta del código frente a los diversos tipos de errores. El CCITT especifica algunas claves, que como se van a emplear para dividir un polinomio serán también polinomios, denominados **polinomio generador**. En el CRC denominado **CRC-16** correspondiente a la norma CCITT V.41, se utiliza el siguiente **polinomio generador**:

$$G(x) = x^{16} + x^{12} + x^5 + x^0$$

donde $x^0 = 1$.

El procedimiento es el siguiente: se toma el polinomio de datos $P(x)$ y se multiplica por x^k , donde k es el exponente más alto de $G(x)$, este polinomio así construido se divide por $G(x)$ y se obtiene un **polinomio resto** $R(x)$, llamado BCS (Block Character Sequence), luego se procede a enviar el polinomio $T(x)$ construido así:

$$T(x) = x^k P(x) + R(x)$$

En el extremo receptor se procederá a extraer lo que suponemos es $x^k P(x)$, lo dividimos por $G(x)$ y calculamos un polinomio resto que si coincide con el $R(x)$ recibido indicará que no hay errores.

La longitud de $P(x)$ es variable, algunos autores hablan de 4 bytes para CRC-16 y otros dicen que se aplica a la trama ó bloque ó paquete.

La operación de división empleada en CRC se denomina de **módulo 2**. Esta división es diferente de la que estamos acostumbrados y funciona así:

- Las restas durante la división (o sea la obtención del resto parcial ó final) no son aritméticas sino **módulo 2**, lo que significa una operación XOR entre los dígitos binarios que se están restando (1 y 0 da 1, 0 y 1 da 1, 0 y 0 da 0, 1 y 1 da 0).
- Sí el primer bit del resto parcial es 1 y queda uno ó más bits del dividendo se baja el primer bit de la izquierda no usado y se hace el XOR. En caso contrario se bajan bits de la izquierda del dividendo hasta que este resto con los bits bajados esté encabezado por un 1 y tenga la misma longitud del divisor. De no lograrse el resto con todos los bits bajados será el resto final de la división

EJEMPLO 4.1:

Determine el BSC(Block Character Sequence), para los siguientes polinomios generadores de datos y CRC.

$$\text{datos } P(x) = x^7 + x^5 + x^4 + x^2 + x^1 + x^0 \quad \text{ó}$$

$$10110111$$

$$\text{CRC } G(x) = x^5 + x^4 + x^1 + x^0 \quad \text{ó} \quad 110011$$

Solución

Primero $P(x)$ es multiplicado por el número de bits en el código

$$\text{CRC, 5. } x^5(x^7 + x^5 + x^4 + x^2 + x^1 + x^0) = x^{12} + x^{10} +$$

$$x^9 + x^7 + x^6 + x^5$$

$$= 1011011100000$$

Al dividir este polinomio por $G(x)$ obtenemos $R(x)$ ó BCS que resulta:

$$\begin{array}{r}
 110011 \overline{) 11010111} \\
 \underline{00000} \\
 110011 \\
 \underline{111101} \\
 110011 \\
 \underline{111010} \\
 110011 \\
 \underline{100100} \\
 110011 \\
 \underline{101110} \\
 110011 \\
 \underline{111010} \\
 110011 \\
 \underline{1001} = \text{Resto}
 \end{array}$$

$$\text{Resto } 01001 \text{ significa } R(x) = 0x^4 + 1x^3 + 0x^2 + 0x$$

$$+ 1 \text{ Se transmite entonces } T(x) = x^k P(x) + R(x) \text{ o sea}$$

$$1011011101001$$

Suponiendo que se reciba $T_R(x)$ 1011011101001 (lo mismo que se transmitió), al dividir $T_R(x)$ por $G(x)$ el resto será cero indicando que se recibió correctamente ó bien se separa lo que se supone es $x^k P(x)$ se divide por $G(x)$ y el resto se compara con $R(x)$ (se recomienda efectuar la división).

Es posible demostrar que la condición para que un código polinomial no detecte un error es que el polinomio que lo representa sea múltiplo del polinomio generador. Por ello la selección del polinomio generador es muy importante ya que si se elige adecuadamente es muy difícil que un error pase indetectado. En el caso del polinomio generador de 17 bits descrito como CCITT V.41 puede demostrarse que detecta:

COMUNICACIÓN DIGITAL

- Todos los bits erróneos que se produzcan en número impar
- Todos los errores que afecten en bloque 16 ó menos bits (single-error bursts, seb)
- 99.9969% de todos los posibles bloques de error(seb) de 17 bits.
- 99.9984% de todos los posibles bloques de error(seb) de longitud superior a 17 bits.

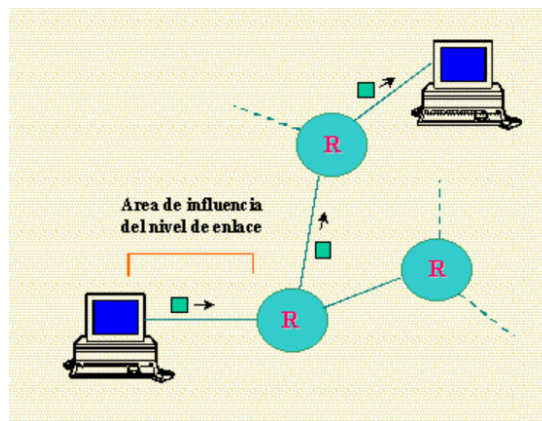
A modo de comentario, la línea telefónica de la Subsección 3.2.3 con CRC mejoraría su BER a 10^{-14} .

Sin embargo, detectar los errores no es suficiente, **hay que corregirlos**.

CONTROL DE FLUJO y CONTROL DE ERRORES

Cuando una trama llega a una máquina conectada a algún tipo de red, antes de pasar la información a niveles superiores, la capa de enlace realiza una serie de operaciones sobre la trama que ocupan un espacio en la memoria e implican un tiempo, función de la máquina, de manera que el proceso de recepción no es instantáneo.

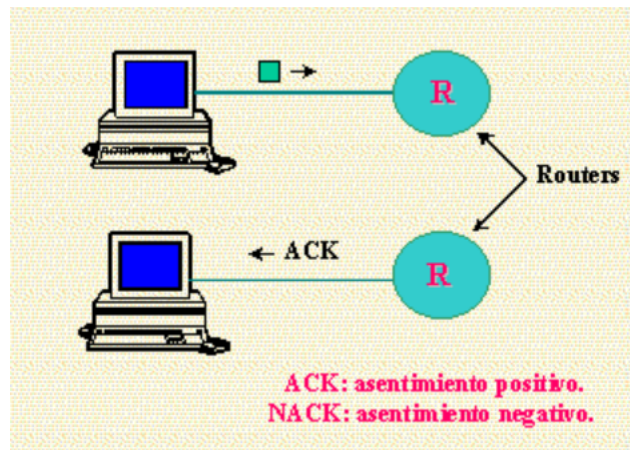
Esta limitación en el espacio de memoria hace que se presente un serio problema cuando un transmisor sistemáticamente quiere transmitir tramas a mayor velocidad que aquella con que puede recibirlas el receptor. Esta situación puede ocurrir fácilmente cuando el transmisor opera en una computadora rápida (o con baja carga) y el receptor en una máquina lenta (o con sobrecarga). El transmisor puede enviar tramas rápidamente hasta que satura al receptor, que comenzará a desechar aquellas a las que no pueda atender.



Para evitar esta situación se hace necesario llevar un control del flujo en el enlace, manejando la velocidad a la que el emisor envía las tramas para que no sature al receptor. Este control de la velocidad generalmente requiere algún mecanismo de realimentación, para que el transmisor pueda saber si el receptor puede mantener el ritmo o no.

La mayoría de las técnicas de control de flujo tienen un principio de funcionamiento igual: el protocolo contiene reglas bien definidas sobre el momento en que el transmisor puede enviar alguna trama, y generalmente estas reglas prohíben el envío de información hasta que el receptor no lo haya autorizado.

Un protocolo de nivel de enlace que quiere enviar tramas eficientemente debe de alguna manera ser capaz de recuperar las tramas perdidas o descartadas. Esto se consigue normalmente usando una combinación de dos mecanismos fundamentales: **acuses de recibo** (*acknowledgments*) y **temporizadores** (*timeouts*). Un acuse de recibo, comunmente referido como **ACK**, es una pequeña trama de control con que el receptor informa al emisor de que ha recibido la transmisión. Si el emisor no recibe un ACK en un tiempo razonable la retransmite; este tiempo está medido por un temporizador.

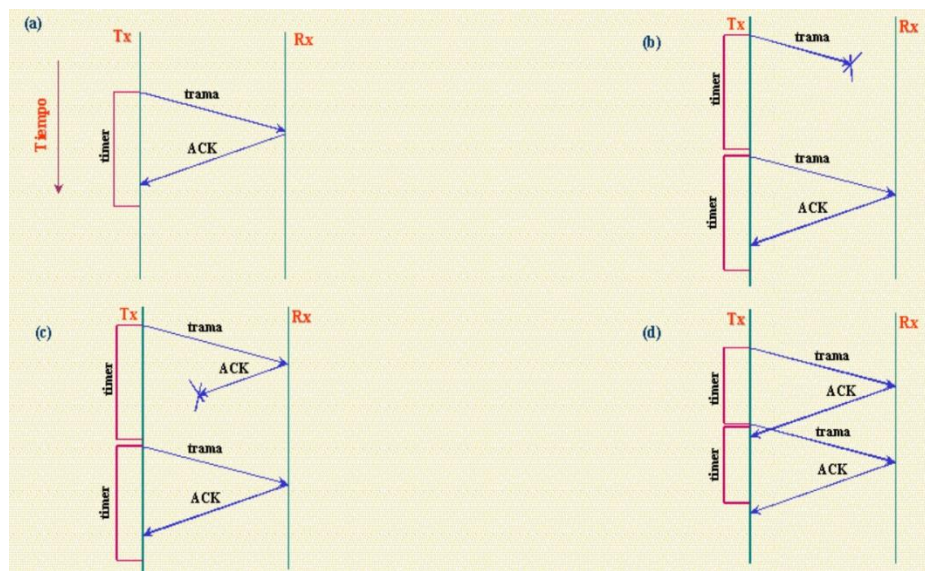


La estrategia general de usar ACKs y "timeouts" para implementar un envío eficiente se suele denominar **automatic repeat request**, normalmente abreviado **ARQ**.

Parada-y-Espera. Es la más simple de las técnicas. Los pasos que llevarían a cabo las dos máquinas en diálogo serían:

1. El transmisor envía una trama al receptor.
2. El receptor la recoge, y devuelve otra trama de aceptación (ACK).
3. Cuando el transmisor recibe esta trama sabe que puede realizar un nuevo envío....
4. Si pasado un cierto tiempo predeterminado no ha llegado acuse de recibo, el emisor retransmite la trama.

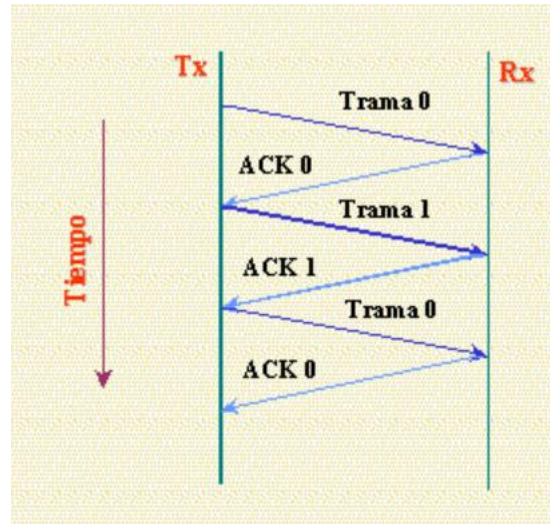
En la figura se representan cuatro diferentes escenarios que se pueden producir con este algoritmo básico. Esta figura es una línea de tiempos, y es una manera muy común de representar el comportamiento de un protocolo. El emisor está representado a la derecha, el receptor a la izquierda, y el tiempo transcurre de arriba a abajo. La figura (a) muestra la situación en que el ACK se recibe antes de que el timer expire. La (b) y (c) muestran el caso en que se pierden la trama original y el ACK respectivamente. En la (d) el timer expira demasiado pronto.



Sin embargo, la técnica de parada-y-espera presenta un importante inconveniente. Supongamos que el transmisor envía una trama y el receptor da el acuse de recibo, pero de

COMUNICACIÓN DIGITAL

alguna manera el ACK se pierde o se retrasa en llegar. Esta situación se ha ilustrado en las figuras (c) y (d). En ambos casos el emisor piensa que el tiempo ha expirado y retransmite la trama, pero el receptor ya había recogido una y cree que ésta que le llega ahora es otra diferente. Para solucionar este problema, la cabecera de una trama del protocolo de parada-y-espera incluye un bit a modo de número de secuencia, que puede tomar los valores 0 y 1; los números de secuencia empleados para tramas consecutivas son alternos, como se ilustra en la figura:



De esta manera, cuando el emisor retransmite la trama 0, el receptor puede determinar que está viendo una segunda copia de la trama 0 y que no se trata de la primera copia de la trama 1, ignorándola (aunque si devuelve un ACK).

La principal limitación del algoritmo de parada-y-espera es que el enlace está ocupado por una única trama cada vez, es decir, se está desaprovechando la capacidad del enlace. Consideremos, por ejemplo, un enlace de 1.5Mbps, y un tiempo de propagación de 45ms (*round-trip time*, RTT). Este enlace tiene un producto de retraso x ancho de banda de 67.5Kb, o aproximadamente 8KB. Como el emisor sólo puede enviar una trama cada 45ms, y asumiendo que el tamaño de la trama es de 1KB, esto implica una tasa de transmisión máxima de $1024 \times 8 / 0.045 = 128\text{Kbps}$, que es aproximadamente un octavo de la capacidad total del canal. Para hacer un uso más eficiente del canal sería pues deseable el transmitir más tramas antes de recibir acuse de las anteriores.

El significado del producto ancho de banda x retraso es que representa la cantidad de datos que pueden estar en tránsito por el enlace simultáneamente. Lo ideal sería mandar tramas sin esperar al primer ACK. Este principio de trabajo se denomina *keeping de pipe full*. El algoritmo de la siguiente sección es precisamente así.

PRESTACIONES

Restringiéndonos al caso en que sólo se puede enviar una trama cada vez, encontramos dos posibles situaciones, definidas por el tiempo de transmisión y el tiempo de propagación:

1.- **Tiempo de Transmisión, T_{tx}** : Tiempo que tarda una máquina en pasar una trama al medio desde que sale el primer bit hasta el último. Se define como el cociente entre la longitud de la trama (L) y el régimen binario en el canal (R).

COMUNICACIÓN DIGITAL

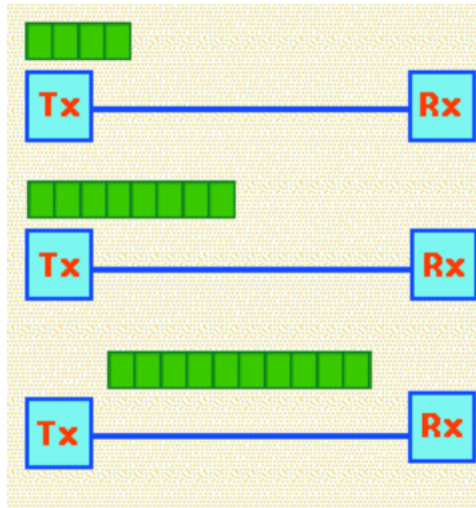
$$T_{tx} = L / R$$

2.- **Tiempo de Propagación, T_{prop}** : Tiempo que tarda una unidad de información en pasar de un extremo del canal al otro. Se define como el cociente entre la *distancia* (d) o longitud del enlace, y la *velocidad del medio de transmisión* (v).

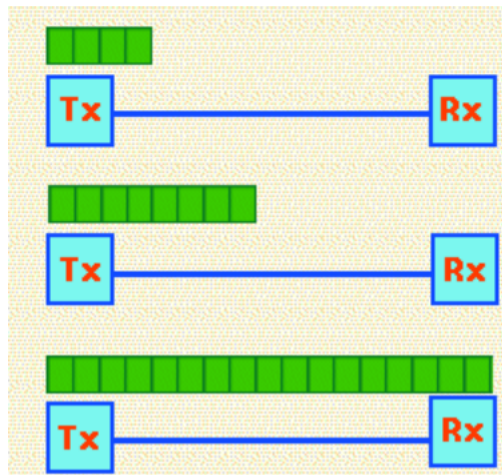
$$T_{prop} = d / v$$

Las dos situaciones posibles son las siguientes:

1ª: antes de empezar a ser recibida, la trama ya se ha terminado de transmitir: $T_{tx} < T_{prop}$



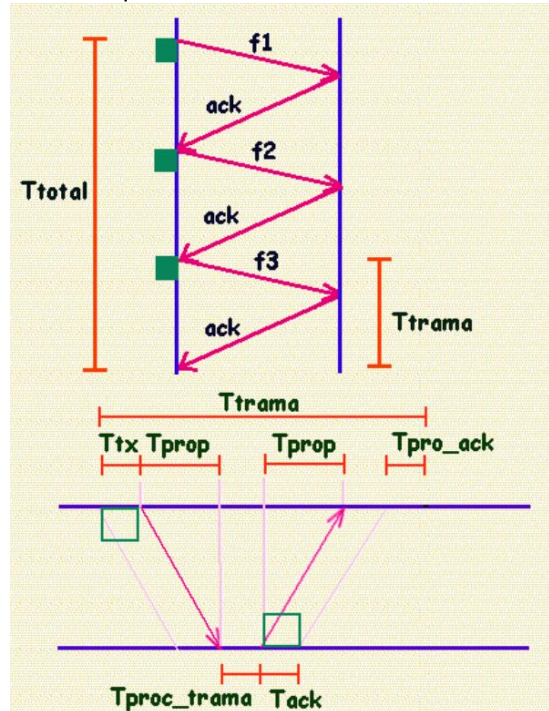
2ª: la trama está siendo recibida, y aún no se ha terminado de transmitir: $T_{tx} > T_{prop}$



Claramente se observa que el canal se aprovecha más eficientemente en la segunda ocasión. Si se define la **eficiencia (U)** como el porcentaje de tiempo que el canal está ocupado durante una transmisión, la eficiencia es mucho mayor cuando $T_{tx} > T_{prop}$:

$U = \frac{\text{Tiempo de Trama}}{T_{total}}$	Eficiencia respecto a la trama.
$U = \frac{\text{Tiempo de Datos}}{T_{total}}$	Eficiencia respecto a los datos de la trama.

Para calcular la eficiencia suponemos que un terminal quiere transmitir un mensaje que divide en n tramas; esto supone que el tiempo de transmisión del mensaje sea: $T_{txm} = n \times T_{trama}$, donde T_{trama} el tiempo de transmisión y propagación de una sola de las tramas (y su ACK correspondiente), y que $T_{total} = n \times T_{trama}$. Todos estos tiempos están representados en el siguiente diagrama en el caso en que $n=3$:



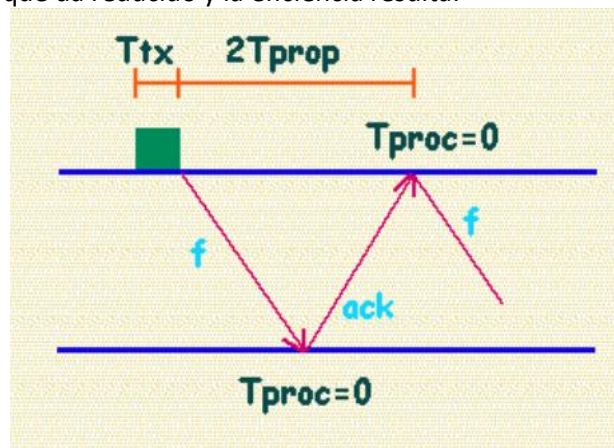
$$U = nT_{tx} / nT_{trama} = T_{tx} / T_{trama}.$$

$$T_{trama} = T_{tx} + T_{prop} + T_{pro_trama} + T_{ack} + T_{prop} + T_{proc_ack}$$

Suponiendo que los tiempos de procesamiento (T_{proc}) y el tiempo de transmisión de la trama ack (T_{ack}) son despreciables frente al resto, encontramos que:

$$T_{trama} = T_{tx} + 2T_{prop}$$

El esquema de antes queda reducido y la eficiencia resulta:

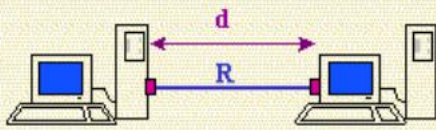


$$U = T_{tx} / (T_{tx} + 2T_{prop}) = 1 / (1 + 2a) \text{ siendo } a = T_{prop} / T_{tx}$$

Como consecuencia de la fórmula anterior se extrae que:

Si $T_{tx} < T_{prop}$:	$a > 1$
	U pequeña (menor del 33%)
	parada-y-espera resulta muy ineficiente
* Si $T_{tx} > T_{prop}$	$a < 1$
	U más elevada (mayor del 33%)
	parada-y-espera resulta menos ineficiente

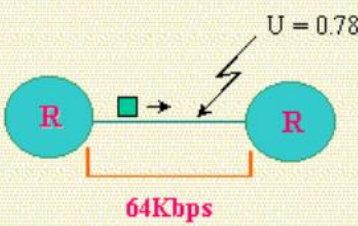
Otra forma de expresar el parámetro **a**, recordando las definiciones de tiempo de propagación y tiempo de transmisión es:



$$a = \frac{T_{prop}}{T_{tx}} = \frac{\frac{d}{v}}{\frac{L}{R}} = \frac{Rd}{vL}$$

R, **v** y **L** son valores prefijados por el diseño; sin embargo la eficiencia es inversamente proporcional a la distancia.

Por último, la eficiencia permite calcular la tasa binaria real con que se están transmitiendo y recibiendo datos por un enlace. Un canal puede ofrecer una capacidad determinada, pero quizá el protocolo no permita alcanzar esa cifra. Se define la capacidad eficaz como la tasa binaria con que los bits se desplazan por la línea:



$$C_{ef} = U \times R = 0.78 \times 64Kbps = 35.8Kbps$$

VENTANA DESLIZANTE.

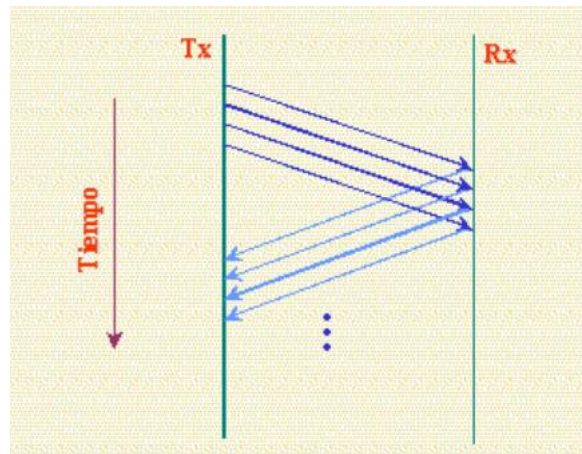
Retomando el ejemplo del enlace que tenía un producto de ancho de banda x retraso de 8KB y las tramas de 1KB, se comprueba que la mejor utilización que se puede hacer del canal requiere que el emisor transmita la novena trama nada más recibir el acuse de recibo de la primera.

En este algoritmo el término **ventana de transmisión** se refiere a un buffer en el cual se almacenan copias de las tramas enviadas, en espera de recibir el ACK correspondiente; si no llegan en el tiempo previsto, se realiza una nueva copia y se retransmite la trama. El **número de secuencia de transmisión, $N(S)$** , es la posición que ocupa la trama enviada en el buffer. El número de secuencia viaja en la cabecera de la trama, dentro del campo de control.

Por **ventana de recepción** se entiende el buffer donde se almacenan las tramas que llegan a una máquina por alguno de sus enlaces. En este buffer esperan a ser procesadas, y a que se devuelva el acuse de recibo correspondiente a cada una de ellas, para que la máquina origen sepan que la transmisión ha llegado sin problemas a su destino.

El **número de secuencia de recepción, $N(R)$** , es la posición que ocupa la trama recibida en el buffer de recepción.

El tamaño de la ventana puede estar preestablecido, o puede negociarse durante el establecimiento de la conexión. En la figura se ilustra el mecanismo del algoritmo para una ventana de tamaño 4:



En el campo de control de la trama habrá 0 bits que expresarán el número de secuencia.

El tamaño máximo de la ventana debe cumplir que:

$$W = 2^n - 1$$

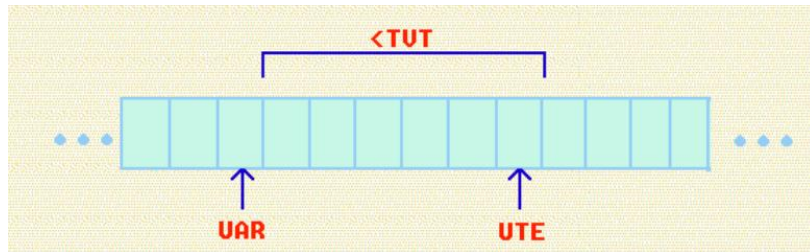
Por ejemplo, para $W=4$, $n>2$

El algoritmo de ventana deslizante es como sigue: primero el emisor asigna un número de secuencia a cada trama. El emisor controla tres variables:

1. El **tamaño de la ventana de transmisión (TVT)**: que será finito. Representa el número máximo de tramas que el emisor puede enviar sin recibir ACK de la primera de ellas.
2. El **número de secuencia del último ACK recibido (UAR)**.
3. El **número de secuencia de la última trama enviada (UTE)**.

El transmisor debe respetar la siguiente inecuación: $UTE - UAR < TVT$, o como mucho igual. Esta situación se ilustra en la figura:

COMUNICACIÓN DIGITAL

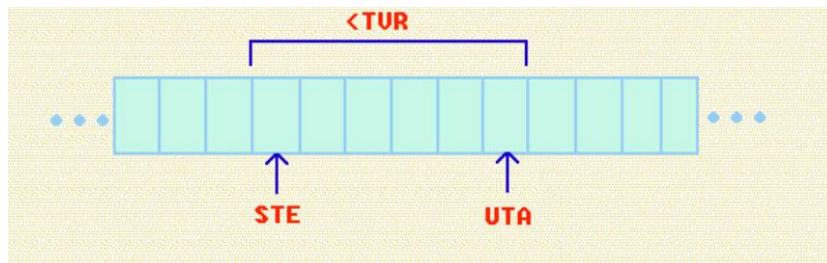


Cuando llega un ACK , el emisor desplaza UAR a la derecha, permitiendo como consecuencia que se transmita otra trama. Además, el emisor asocia a cada trama que envía un timer, retransmitiendo la trama si el timer expira antes de que llegue su ACK correspondiente. Por tanto, estamos asumiendo que el emisor dispone de un buffer donde almacena TVT tramas, que pueden tener que ser retransmitidas.

Por su parte el receptor mantiene otras tres variables:

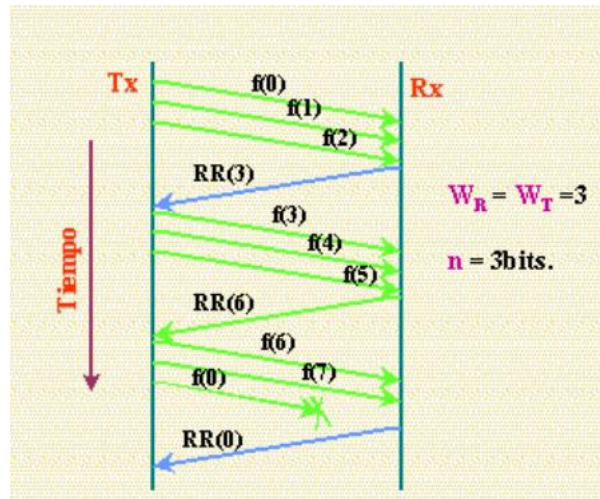
1. El **tamaño de la ventana de recepción (TVR)**: que indica el máximo número de tramas que el receptor puede aceptar.
2. El **número de secuencia de la última trama aceptada (UTA)**: última trama procesada.
3. El **número de secuencia de la siguiente trama esperable (STE)**.

El receptor respeta la inecuación: $UTA - STE < TVR$, o igual como mucho.



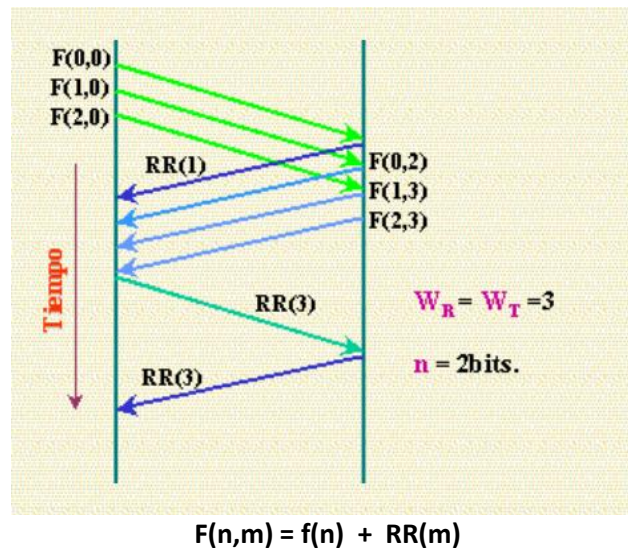
Cuando se recibe una trama con número de secuencia N se comprueba si está dentro o fuera de la ventana, es decir, si $N < STE$ o $N > UTA$ la trama está fuera de la ventana y es descartada. Si $N > STE$ o $N < UTA$ la trama se recibe y se guarda en la posición del buffer correspondiente. Ahora el receptor tiene que decidir si envía o no un ACK (**RR o RNR**), y que clase de ACK:

RR (<i>n</i>) (receptor preparado)	Indica a la fuente que ha recibido bien hasta la (<i>n-1</i>), y que espera la <i>n</i> .
RNR (<i>n</i>) (receptor no preparado)	Indica a la fuente que se han recibido bien hasta la trama (<i>n-1</i>) incluida, pero que no siga transmitiendo por el momento.



En el último intercambio de tramas de la figura se ilustra como implementar un NACK: basta con dar acuse de la última trama que se recibió bien y solicitar el envío de la que no llegó.

NOTACIÓN: una sola trama puede poseer no sólo su propio número de secuencia, sino también información sobre el buffer de recepción de la máquina de la que procede, es decir, lleva un acuse de recibo. Eso se expresa de la siguiente manera:



$$F(n,m) = f(n) + RR(m)$$

PRESTACIONES.

Al igual que en caso del protocolo de parada y espera pueden ocurrir dos cosas: que el uso del canal sea eficiente o ineficiente.

ENVIO CONTINUO.

El tamaño de la ventana es suficientemente grande, de manera que antes de llenarse e interrumpir la transmisión ya han comenzado a llegar ACKs. La condición de envío continuo es: $N > 2a + 1$

y: $U=1$

La demostración es muy sencilla: El tamaño de la secuencia es un N tal que: $N \cdot T_{tx} \text{ trama} \geq T_1$, siendo T_1 el tiempo que tarda una trama y su ACK correspondiente en ser transmitidos.

$$T_{tx} + T_{prop} + T_{ack} + T_{prop} + T_{proc} = T_1$$

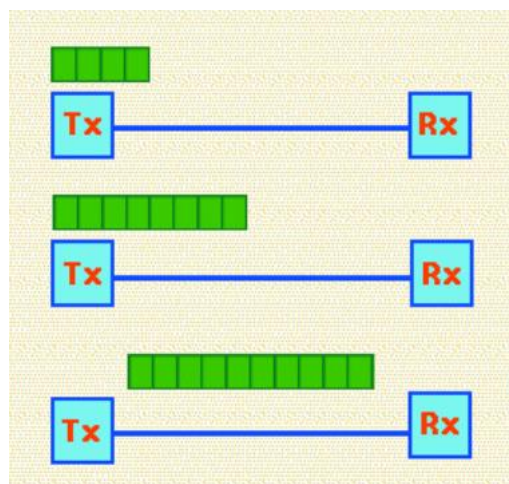
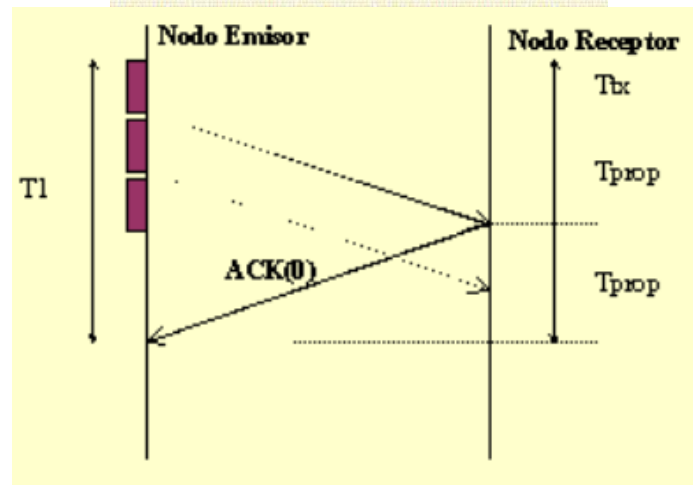
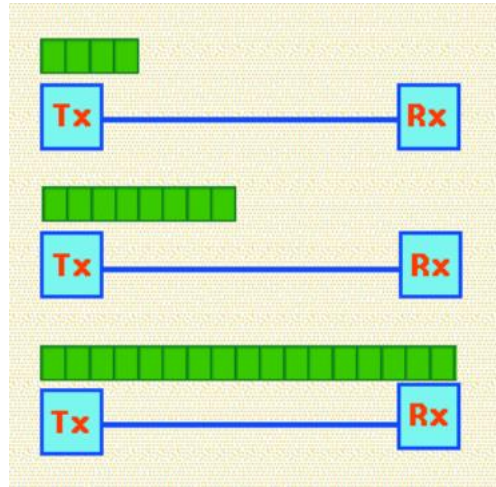
$$\text{Si } T_{ack} = T_{proc} = 0, \text{ entonces: } T_1 = T_{tx} + 2 \cdot T_{prop}$$

COMUNICACIÓN DIGITAL

$$N = T_1 / T_{tx} = (1/T_{tx}) \cdot (T_{tx} + 2 \cdot T_{prop})$$

$$\text{Finalmente: } N = 1 + 2 \cdot (T_{prop}/T_{tx}) = 1 + 2^a$$

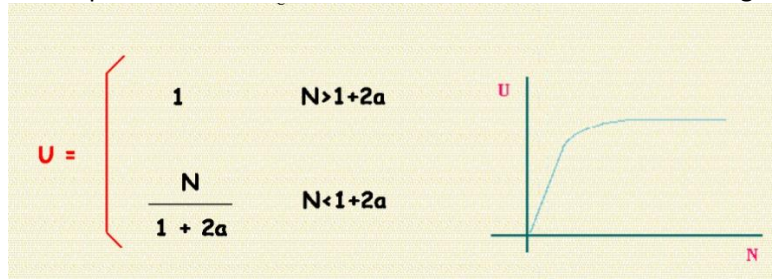
Pero también puede ocurrir que la ventana de transmisión esté llena y no lleguen los ACKs, por lo que la transmisión se interrumpe. La eficiencia es menor porque aunque las tramas se desplazan por el canal a igual velocidad que antes, este está vacío más tiempo:



$$\text{Se cumple que: } N < 1 + 2a$$

$$\text{y: } U = N / (1 + 2a)$$

En definitiva, para un protocolo de ventana deslizante la eficiencia tiene la siguiente forma:



DETECCIÓN, CORRECCIÓN DE ERRORES Y COMPRESIÓN EN MODEMS ANALÓGICOS.

El advenimiento de módems de más alta velocidad está vinculado a señalización multinivel, esto es consecuencia del siguiente análisis.

CAPACIDAD DE UNA LÍNEA TELEFÓNICA.

Una línea telefónica (circuito de voz) se aproxima a un canal Gaussiano de banda limitada (GBLC) con ruido Gaussiano aditivo, para ese caso es aplicable el Teorema de Shannon que establece que si la velocidad a que introducimos los datos en el canal es menor que su capacidad C , existe un código para el cual la tasa de error tiende a cero si la longitud del mensaje es infinita. Esa capacidad C se calcula con la expresión:

$$C = B \log_2 (1 + S/N)$$

donde B es el ancho de banda limitado del canal en Hz, S/N la relación señal a ruido y C la capacidad del canal en bits por segundo(bps).

En el caso de la línea telefónica con un ancho de banda de 3100 Hz y una S/N de 1023 (aproximadamente 30 dB) resulta la capacidad del canal 31.000 bps.

Los valores de B y S/N son razonables y sabemos que rara vez podemos alcanzar más de 3000 baudios de velocidad de símbolos en una línea telefónica. Ello se debe a la interferencia Inter simbólica producida por la respuesta no uniforme del canal al pulso.

Por lo que si empleamos un sistema de dos niveles (binario) estaríamos hablando de 3000 bps, el valor límite de Shannon puede aproximarse más mediante sistemas multinivel, sin embargo, al emplear estos sistemas debe incrementarse la potencia pues de otro modo la relación S/N se deteriora rápidamente.

DETECCIÓN Y CORRECCIÓN DE ERRORES.

Al implementarse los primeros modems a 1200 y 2400 bps se observó que el ruido en las líneas discadas dificultaba grandemente la comunicación al introducir errores severos en los textos (*garbage* ó basura) haciendo inútiles los archivos binarios. Surgió entonces la necesidad de implementar **modems con detección y corrección de errores**.

La empresa Microcom desarrollo un protocolo llamado MNP (Microcom Networking Protocol) que se ha convertido en el estándar en modems

MNP se puede implementar en software ó en hardware, sin embargo, opera más eficientemente cuando se coloca en el “firmware” del módem (que es software grabado en chips ROM que se encuentran en la tarjeta del módem). Los modems en ambos extremos deben usar el mismo MNP (veremos que hay varios) y eso se logra porque al iniciarse la comunicación el módem

COMUNICACIÓN DIGITAL

que la solicita “informa” que está usando MNP y el otro extremo automáticamente responde que sí, si tiene esa opción, lo que activa el software del módem solicitante. Cuando se implementa así el MNP **cambia** una serie de datos **asíncronicos** por una **equivalente** de datos **síncronicos** y forma **paquetes de datos continuos con CRC** para detección de errores.

Ya se dijo que hay varios MNP:

- MNP2: en este MNP inicial los paquetes de caracteres asíncronicos se transmiten con un encabezamiento, que reduce la velocidad (throughput) a 84%, por ello prácticamente no se ha usado.
- MNP3: en él los bits de start y stop del byte son eliminados por lo que se pasa de 10 a 8 bits por carácter, con lo que se gana un 20%, sin embargo al convertirlo en una serie sincrónica de datos (paquetes) se agregan una serie de bits que suman un 12%, por lo que se está ganando un 8%, lo que lleva la velocidad de 2400 bps a 2600 bps cuando no hay ruido que produzca errores. Los protocolos basados en software (no en firmware) no quitan los bits de start y stop, por lo que su eficiencia es menor. El método de detección de errores de MNP es CRC de 16 bits sobre bloques de igual longitud en MNP3 (en otros la longitud es variable), una vez detectado un error se emplea retransmisión para corregirlo, o sea es ARQ. Sin embargo se transmiten 8 bloques y al final de ellos es que se espera la información de si hubo errores y de cual fue el primer bloque con errores, ordenándose la retransmisión a partir del primer bloque erróneo, es entonces un método n-atrás (go back-n).
- MNP4: es casi idéntico al 3 sólo que tiene un mejor rendimiento (throughput) debido a que ajusta la longitud de los bloques permitiendo bloques más largos durante lapsos de pocos errores, lo que aumenta la eficiencia a 120%.
- MNP5: incluye las características descritas de las clases 3 y 4, pero usa además compresión de datos para mejorar la eficiencia. La mejora depende del tipo de datos, si se trata de textos de lectura la eficiencia puede llegar a duplicarse (mejora típica 85%), en datos y archivos ya comprimidos la mejora es ínfima. El método de compresión empleado hace dos cosas: primero, busca caracteres repetidos, cuando tres o más caracteres aparecen en sucesión los comprime; segundo, usa tabulación dinámica de caracteres repetidos, lo que le da habilidad de aprendizaje.
- MNP7: comprime 3 a 1, y emplea técnicas de codificación de longitud en el algoritmo de compresión, de modo que registra la recurrencia de caracteres y ajusta el código en consecuencia.

LAP-M (Link Access Procedure for Modems): es un estándar de detección y corrección de errores del CCITT basado también en CRC.

Como existen millones de modems con MNP, CCITT adoptó en 1988 nuevos estándares:

- V.42: incluye dos protocolos de detección y corrección de errores: LAP-M y MNP 2, 3 y 4. Además permite comprimir los datos usando MNP 5 ó 7.
- V.42bis: es un estándar de compresión de datos destinado a reemplazar MNP5 en modems que usan detección y corrección de errores de V.42. Se logran relaciones de compresión de hasta 4 a 1 mediante un algoritmo llamado de Lempel-Ziv (en él los caracteres usados frecuentemente son codificados y se construye un diccionario, y se transmiten solo los símbolos del diccionario en lugar del carácter completo, los diccionarios son contruidos en ambos extremos usando encabezamientos mínimos), con esto se logra que un módem de 9600 bps pase a 56700 bps.
- V.Fast (llamado también V.32turbo): es un estándar que aprobado por el CCITT soporta velocidades hasta 28800 bps para datos sin comprimir, y usando V42bis podrá llegar a 86400 bps en líneas discadas comunes (de voz).

COMPRESIÓN.

La compresión es el “Holy Grail” de las telecomunicaciones ya que en la medida que se desarrollan mejores esquemas de compresión es posible utilizar más eficientemente el canal de comunicaciones. Esquemas como la codificación de Huffman tiene más de 20

años. La compresión se utiliza para voz y para video, esquemas como JPEG, QCIF, MPEG 1,2,3 y 4(200:1) y Wavelets (480:1) son muy interesantes.

El esquema de compresión **MP3**, utilizado para música y video en Internet, es en realidad un **MPEG-1 capa 3** con compresiones que van desde 96:1 en sonido telefónico a 12:1 en calidad CD.

Es necesario consultar publicaciones recientes ya que es un campo en el que se está trabajando a diario.

SEGURIDAD EN REDES.

Introducción.

Un sistema de comunicación de datos que no presente una protección adecuada está bajo la amenaza de un uso indebido o ilegal de la información presente en él. Por esta razón todos los sistemas de comunicación de datos deberían poseer cierto grado de seguridad o privacidad de acorde a la importancia de la información manejada. Para esto existe una gran variedad de técnicas tales como: protección física, sistemas programados, encriptamiento, etc.

Hoy en día un especialista en comunicaciones de datos no puede estar armado solamente de sus conocimientos de modulación, debe tener como tarea fundamental mantener un traslado de datos seguro y confiable de un sitio a otro, por lo que es muy importante el conocimiento y el manejo de las técnicas de encriptamiento para la seguridad y confiabilidad del traslado de datos.

Aplicaciones de los Sistemas de Comunicación de Datos (SCD).

Hasta hace 30 años el uso y los usuarios de SCD eran muy limitados. Hoy en día esta técnica se utiliza en casi todas partes: negocios, institutos financieros y agencias de gobierno dependen de los SCD. Por ejemplo:

- **Negocios.** Usan los SCD para todas sus tareas diarias: Registros financieros; recepción, expedición y mantenimiento de cuentas; transferencia de fondos electrónicamente; inventarios y control de stock; control de procesos; órdenes de compra, comercio electrónico, reservaciones aéreas, hoteleras, etc.
- **Institutos Financieros.** Varios de los primeros sistemas de computadoras comerciales fueron usados en bancos y finanzas para operaciones financieras, traslado de fondos, terminales financieros, cajeros automáticos, redes de datos.
- **Agencias de Gobierno.** Aparte del uso militar están los sistemas de base de datos usados por la policía, el impuesto sobre la renta, etc.
- **Empresas.** Intranet (redes internas de las empresas con correo electrónico, conferencias (audio ó tele), acceso a bases de datos, bibliotecas, etc.).
- **Público.** Internet.

ACCESO NO AUTORIZADO DE DATOS.

Cuando más y más datos se mueven entre computadores, mantener la confidencialidad de los datos comienza a ser un problema. Para nuestros propósitos, *seguridad de datos* significa “**proteger los datos de alteración o acceso por partes no autorizadas**”.

La información es un artículo único. Puede ser copiada sin alterar la información original, esta característica hace del robo de información algo difícil de detectar. O alterada, sin técnicas especiales es difícil de detectar si la información de un SCD a sido modificada sin el conocimiento del usuario del SCD.

TÉCNICAS PARA SEGURIDAD DE DATOS.

INTRODUCCIÓN.

La seguridad de los datos se complica debido a que cuando se hace una copia de la información en una computadora la información no es alterada ni destruida. Por lo tanto, descubrir esta clase de hurto de información es muy difícil. Sin embargo, varias cosas pueden hacerse para mejorar la seguridad de los datos y de este modo decrementar la posibilidad de acceso no autorizado a la información. Existen ciertas recomendaciones para mejorar la seguridad de los datos, tales como:

- **Revisión del Personal:** Para evitar el acceso de personas no autorizadas a la información, se debe decidir primero quien está autorizado al acceso de la información y a qué clase de información.
- **Seguridad Física:** Se refiere a los métodos de “boxing out” para evitar el fisgoneo. La medida más directa debe ser el de asegurarse de mantener los computadores y los terminales en áreas de acceso controlado. Esto reduce grandemente la oportunidad de que alguien haga una visita no autorizada a los datos y haga una copia por sí mismo.
- **Validación de Acceso:** Dado que hay un número de personas con acceso al sistema se debe mantener una clase de control, ya que no todos los usuarios tienen la misma necesidad de información, por lo que deben ser implementadas bases de datos segmentadas y passwords (contraseñas).
- **Passwords:** Permiten que los terminales y computadores den acceso a cierta clase de información permitida previo conocimiento del respectivo password. Deben ser cambiados frecuentemente para mayor efectividad.
- **Sistemas de Call-Back:** La mayoría de los sistemas de computadoras que pueden ser accedidos por línea telefónica son vulnerables a Break-ins por parte de un llamador anónimo. Por esto se requiere que el sistema detecte llamadas no autorizadas, regrese las llamadas para transferir datos, utilice passwords y exista chequeo de identidad del que realizó la llamada.
- **Codificación:** Si no se puede mantener los datos fuera del alcance de manos no autorizadas, lo mejor que sigue es realizar una transmisión disfrazada. Una forma de realizar esto es codificar los datos de manera que parezcan sin sentido a los ojos del ladrón de datos. Por largo tiempo simplemente se codificaban los datos con códigos de computadoras que servían para todo propósito, ya que no había muchas personas que aprendieran a *romper* el código. Al surgir la literatura sobre computadoras estas técnicas quedaron obsoletas.

ENCRIPTACIÓN.

Encriptación es el proceso de codificación de la información, de tal manera de hacer difícil para el ladrón de información que lo robado tenga sentido para él. La codificación no debe ser demasiado compleja pues al usuario le sería engorroso entenderla, ni muy simple como para que un lector dedicado pueda aprender a descifrarla. En términos de encriptación, el mensaje original es referido como *plain text*, debido a que está en lenguaje sencillo o llano. La salida del proceso de encriptación se llama *cipher text*, debido a que es una versión codificada del texto sencillo de entrada (un cipher es un sistema de escritura secreto que usa un esquema prearreglado o clave).

El uso de equipos de codificación automática data desde 1930. Hasta hace poco tales equipos sólo estaban disponibles para los gobierno y usos militares, esto ya no es así.

MÉTODOS DE ENCRIPCIÓN.

a. Sustitución. Un cifrado por sustitución reemplaza los caracteres originales uno a uno por otro set de caracteres. Los dos sets carácter a carácter como en el siguiente ejemplo:

Alfabeto original:	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Alfabeto de sustitución:	BCDEFGHIJKLMNOPQRSTUVWXYZA
Mensaje original:	MEET ME AT EIGHT ON FRIDAY
Mensaje codificado:	NFFU NF BU FJHIU PO GSJEBZ

En este ejemplo se ilustra un cifrado de sustitución monoalfabético donde el alfabeto de sustitución es la clave, la cual produce el mensaje en clave. Así un cifrado por sustitución monoalfabético reemplaza cada letra del mensaje original con una letra del código clave. En nuestro ejemplo el código es un caso especial llamado *Caesar cipher*, llamado así por Julius Caesar. Un código Caesar usa un alfabeto de sustitución único, consistiendo en el alfabeto básico (A a la Z), que es desplazado en posición. El corrimiento es prearreglado. Se puede notar en el ejemplo que el corrimiento es de una posición. Para otros cifrados por sustitución de alfabeto el orden de los caracteres del texto original no es retenido.

Una mirada al código de salida nos puede revelar que este método no cambia la longitud del mensaje, ni la longitud de cada palabra, por lo que conociendo el idioma en que está el mensaje es relativamente fácil saber que dice el mensaje final basándonos en la repetibilidad de las letras en cada idioma como veremos a continuación:

Mensaje codificado:	NFFU NF BU FJHIU PO GSJEBZ
Número de letras:	21
Número de veces usadas cada letras:	
	F4 U3 B2 N2 J2 G1 H1 I1 O1 P1 S1 Z1

En inglés la letra E es la letra más usada comúnmente. En orden estadístico de repetición tenemos que las letras siguen el siguiente orden: ETAON RISHD, así como algunas palabras también son de uso común como “THE”, con todo esto podemos descifrar fácilmente el texto. Por la facilidad de descifrar el texto el código de sustitución no es usado actualmente.

b. Cifrado por transposición. Otro método simple de cifrado es el de transposición. Un poco mejor que el de sustitución de caracteres, usa los caracteres actuales escritos de otra forma. En el siguiente ejemplo podemos apreciar un método simple de transposición, donde el mensaje se escribe de derecha a izquierda

Mensaje original:	MEET ME AT THE MOVIES
Mensaje codificado:	SEIVOM EHT TA EM TEEM

COMUNICACIÓN DIGITAL

Este ejemplo ilustra la naturaleza del cifrado, pero el ejemplo es sencillo a propósito. Es posible reescribir el mensaje de otras formas, por lo tanto la transposición no es tan obvia como se muestra:

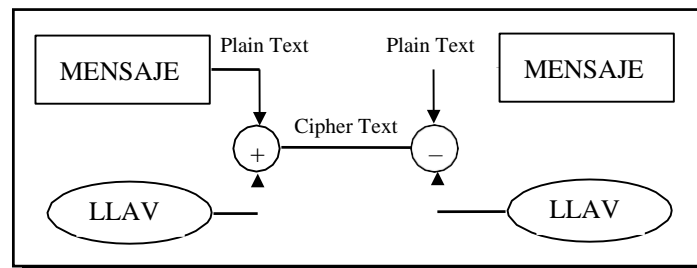
Mensaje original:	BEGIN THE ATTACK AT DAWN
Reescribiendo:	B G N H A T C A D W
	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
	E I T E T A K T A N
Mensaje a transmitir:	BGNHA TCADW EITET AKTAN

En este ejemplo se puede ver que el mensaje es transpuesto en un patrón repetitivo, arriba y abajo, izquierda a derecha. El mensaje fue separado en posiciones pares e impares, la fila superior se escribe primero que la inferior y el mensaje final fue separado en bloques de cinco letras para la transmisión. Mientras este mensaje parece mucho más seguro que el de sustitución, realmente no ofrece mucho. La longitud de las palabras y los puntos de ruptura desaparecen, pero la frecuencia de las letras tiende a ser retenida. Además, no es necesario un código de cifrado por sustitución, el texto original está aquí, solo que rearrreglado.

c. Sistemas de llave privada. Los más modernos métodos de cifrado de texto dan forma al texto original a partir de cálculos matemáticos. La operación más común consiste en añadir al texto original otra cadena de caracteres, la cual sirve como llave. O sea:

Plain Text + Key (Data Stream) = Cipher Text

La salida de la operación es el texto cifrado:



Veamos ahora un ejemplo de cómo un sistema basado en técnicas computacionales codifica. Aquí cada carácter puede ser representado por su valor en ASCII, tanto la clave, el texto original y el texto cifrado. Vamos a considerar una clave *random* corta con el fin de explicar el ejemplo. Supongamos que el texto original es “Meet me on friday” y la clave aleatoria es “ALMQVIYXRGNB95FPO” como se muestra a continuación:

Mensaje original:	Meet me on friday
Clave:	ALMQVIYXRGNB95FPO

Ahora construyamos el texto cifrado por codificación del código ASCII del mensaje original con la clave:

Código ASCII	
Mensaje:	77 101 101 116 32 77 101 32 111 110 ...
Clave:	65 76 77 81 86 49 89 88 82 71...
Suma:	142 177 178 197 118 126 190 120 193 181...
-128*:	14 49 50 69 118 126 62 120 65 53...
Cifrado:	SO 1 2 E v ~ > x A 5...

*Se sustrae solamente si Suma > 127

Al analizar este ejemplo se puede observar que el valor ASCII de cada carácter del mensaje original (incluidos los espacios) se le añade el valor ASCII de cada carácter de la clave. Nótese que el resultado de la suma se reduce si esta excede a 127. Por lo tanto son retranslados al código ASCII y no necesariamente todos pueden ser impresos: al primer carácter del cifrado del ejemplo corresponde el código SO (Shift Out) del ASCII. Éste es un carácter para control de periféricos y no tiene representación impresa, pero puede ser transmitido fácilmente como dato.

La salida de este proceso parece ser más *random* que la salida de los otros métodos de cifrado estudiados hasta ahora. En parte esto es debido a que los espacios entre las palabras también son encriptados (lo que no podía hacerse con los otros métodos) y también debido a que el mensaje se encuentra en mayúsculas y minúsculas. El texto cifrado puede ser diferente para el mismo mensaje si éste es escrito totalmente en mayúsculas y con la misma clave. La verdadera razón de que la salida parezca más aleatoria es la naturaleza de la clave. Si la clave es verdaderamente aleatoria, entonces la secuencia de caracteres de la clave nunca se repetiría. Si los caracteres no se repiten, entonces nunca se podría descifrar dos segmentos del mensaje original con la misma secuencia de la clave. Si se genera una clave absolutamente aleatoria, el mensaje no puede ser descifrado sin la clave.

Sin embargo, no es posible producir una verdadera secuencia aleatoria de números. En su lugar se obtienen diferentes puntos de partida o semillas (a través de algoritmos) en el programa que genera la secuencia de números. Para un observador, estos números parecen ser aleatorios, sin patrón alguno. Los matemáticos se refieren a ellos como números pseudo-aleatorios, porque el patrón no es realmente aleatorio por lo menos para quien conozca el algoritmo.

Tarde o temprano, una clave producida por un generador de números pseudo-aleatorios puede repetirse. Si el “rompe-códigos” puede comparar dos secciones de texto cifrado que fue producido con la misma secuencia de la clave para dos mensajes diferentes, puede comenzar a romper la clave. Así, si el rompe-códigos tiene suficiente paciencia, puede eventualmente duplicar la clave.

Ésta no es la única mala noticia. Si el patrón fuese totalmente aleatorio, ¿cómo podría ser recreada la clave por el receptor para que el mensaje fuera descifrado?. Se podría pensar en enviar la clave (el algoritmo) como dato, pero esto sería frustrar el propósito de la encriptación, ya que alguien casualmente podría capturar tanto la clave como el texto cifrado. Las secuencia pseudo-aleatorias, sin embargo, pueden ser recreadas por el receptor para producir una copia de la llave usada para encriptar el mensaje. Esta es una de las razones por las cuales se usa.

Por último, el método de sumar la clave no es el único método para encriptar. Se pueden usar diversos algoritmos, tales como multiplicar, lógica complementaria, etc. y la encriptación puede ser realizada por software o hardware automáticamente.

d. Sistemas de llave pública. La distribución y uso de las llaves de cifrado en una red de datos puede ser un trabajo difícil. El primer problema es como distribuir las llaves. Obviamente, no se pueden distribuir las llaves por el enlace de comunicaciones. En lugar de eso, las llaves deben ser enviadas por rutas más seguras. Otro problema es que si la llave no es obtenida uniformemente debido a problemas de sincronismo, la red no puede trabajar apropiadamente hasta que la llave correcta esté en su sitio. Una forma fácil de resolver el problema del manejo de llaves es el uso de sistemas de *llave pública*.

En un sistema de llave pública (desarrollado por Whitfield Diffie y Martin Hellman en 1976, usa algoritmos sobre campos finitos para asegurar y descifrar los datos) la llave se divide en dos partes. Una parte de la llave es información pública. La segunda parte es información privada conocida solamente por las personas que quieren utilizar la llave para comunicarse. La teoría de los sistemas de llave pública es compleja y todavía no es usada al máximo. Sin

embargo, estos sistemas ofrecen la promesa de comunicaciones seguras con un reducido problema de manejo de llaves.

Para usar un sistema de llave pública cada usuario construye una función de una vía que se usa para encriptar los mensajes. Ésta es una porción pública de la llave. El usuario también construye la función inversa de esta función, la cual es necesaria para descifrar los mensajes encriptados con la porción pública de la llave. Esta función inversa es mantenida en secreto y es la porción privada de la llave. Aún si se conoce la porción de la llave pública, un ladrón de información no puede calcular la porción privada.

Los sistemas de llave pública están basados en el concepto conocido como **one-way trap door functions**. Las funciones son fáciles de desarrollar, aunque la teoría fundamental no es sencilla. Las funciones inversas son únicas y no pueden ser derivadas de funciones trampas conocidas. Para derivar la función inversa, el rompe-códigos debe conocer como la función fue construida y no sólo conocer el resultado que la construcción produce.

Para entrar a un sistema de llave pública, cada usuario primero debe publicar una llave pública de encriptación en un directorio. La llave consiste en dos números. Entonces, cuando dos usuarios A y B quieren comunicarse, cada uno de ellos va al directorio. Usando la llave pública que A ha publicado en el directorio, B cifra un mensaje y lo envía a A. A, conociendo la función inversa de esta única función trampa, puede descifrar este mensaje fácilmente y reconstruir el texto original. De manera similar, si A desea enviar una respuesta a B, el mensaje es cifrada con la llave pública de B, la cual fue colocada por B en el directorio. B puede descifrar este mensaje usando la inversa de su función *trap-door*.

Esta llave de dos partes es menos segura que los que usan una sola llave completa mantenida en secreto dado que midiendo el tiempo de procesamiento de ciertas operaciones de cifrado pueden "romperse" la llave pública.¹ Esto es verdad, sin embargo, recuerden que la información no necesita mantenerse segura por siempre. De esta manera, si un sistema de llave pública puede mantener por períodos de tiempo razonable una adecuada seguridad, podemos sentirnos satisfechos. Debido a que ellos pueden llevar a cabo esta meta, son usados más y más sistemas de llave pública.

ESTÁNDARES DE ENCRIPCIÓN DE DATOS.

Para probar si los requerimientos de seguridad de datos se cumplen y también para proveer cierto grado de uniformidad, el *National Bureau of Standards* fue encargado en 1970 para desarrollar un *Data Encryption Standard* (DES) para ser usado en aplicaciones no relacionadas con defensa dentro del gobierno de los U.S. DES fue diseñado para satisfacer los siguientes criterios:

- Alto nivel de protección contra acceso y modificaciones de datos no autorizadas.
- Simple de entender, pero difícil de violar.
- Protección basada en llave y no dependiente de cualquier programa o proceso de encriptación.
- Económico y eficiente.
- Adaptable a diversas aplicaciones.
- Disponible para todos los usuarios y proveedores a un costo razonable.

El DES resultó aprobado por el Gobierno de los Estados Unidos en 1977.

El DES encripta bloques de 64 bits usando una llave de 56 bits dando lugar a billones (trillones en inglés) de posibles permutaciones, la llave es un algoritmo usado para codificar los datos.

COMUNICACIÓN DIGITAL

Independientemente de que se trate de “llave privada” ó de “llave pública” con estaciones de trabajo poderosas puede “romperse” el código, por ello se han considerado varias alternativas:

- usar llaves de 128 bits
- Utilizar “triple DES”, cada bloque es codificado tres veces con tres llaves diferentes.
- usar algoritmos diferentes del DES

El tema además de sus aspectos de seguridad de datos tiene connotaciones públicas. Los gobiernos, sus cuerpos de seguridad, organismos de inteligencia, etc. se niegan a permitir esquemas que ellos no conozcan ó que no puedan de algún modo controlar para poder “examinar” el contenido de los datos. Por ello el Gobierno norteamericano solo autorizaba la exportación de sistemas con llaves de 40 bits, luego autorizó a tres empresas a exportar las de DES de 56 bits.

Casos como el del profesor de matemática David J. Bernstein, de Illinois, procesado por intentar exportar un programa de cifrado de su creación han causado polémicas, intervenciones judiciales en resguardo de los derechos constitucionales y preocupación en la industria estadounidense.

En Noviembre del año 2001 la Secretaría de Comercio de los Estados Unidos aprobó un nuevo estándar de cifrado el “Advanced Encryption Standard (AES) (FIPS PUB 197)”, este AES fue aprobado por la agencia “National Institute of Standards and Technology (NIST)” y publicado como FIPS(Federal Information Processing Standards) PUB 197 y consiste en un algoritmo de cifrado simétrico de bloque y es capaz de utilizar llaves de 128, 192 y 256 bits para cifrar y descifrar los datos en bloques de 128 bits, para ello utiliza el algoritmo de **Rijndael**.

AUTENTICACIÓN.

Encriptar no es suficiente es necesario “autenticar”, ó sea, asegurar que las personas ó entes participando en una “conversación” (ó intercambio de datos) son quienes dicen ser. Los mecanismos empleados van desde tarjetas de seguridad hasta autoridades de certificación.

Una técnica muy popular es la “**firma digital**”, en ella se calcula con una función criptográfica de un solo sentido un valor llamado “hash”, se envía el texto original y el valor hash encriptado con una llave privada (que se denomina “**firma**”), en el extremo receptor se vuelve a calcular el valor hash, se descifra la firma con clave pública y se compara con el hash calculado, si verifica el mensaje es auténtico.