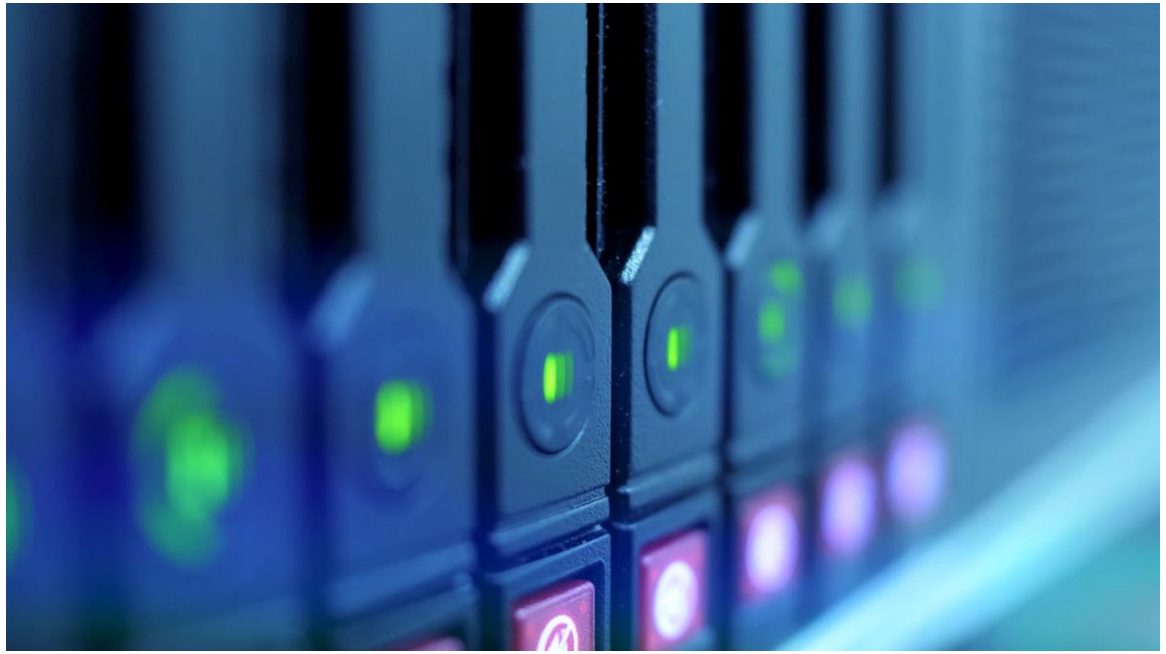
Definición y Características de los Sistemas de Transmisión



Unidad III

Uaaccbasdseici7

eiraagodic2024

Control de enlace de datos

Control de flujo

El control de flujo es un aspecto fundamental que dicta la eficacia de cualquier proceso, sistema u operación.

Es la mano invisible que guía el buen funcionamiento de los sistemas

- Garantiza que todas las piezas funcionen juntas en equilibrio
- Resulta crucial para que los sistemas informáticos sean más organizados y manejables

En el contexto de la TI

- Conjunto de procedimientos utilizados para gestionar la tasa de transferencia a la que se transmiten los datos entre dos nodos.
  - Garantiza que un emisor, si opera a un ritmo más rápido, no inunde de datos a un receptor más lento.
- El mecanismo empleado permite al nodo receptor controlar la velocidad de transmisión.

Control de flujo

- Finalidad
  - Fundamental para mantener el equilibrio en la velocidad de transmisión de datos entre un emisor y un receptor
  - Al activar el control de flujo, se puede mejorar significativamente:
    - El rendimiento de la red
    - Reducir las retransmisiones
    - Aumentar la eficacia al evitar la pérdida de datos o la congestión.
  - Ejemplos
    - Servidor rápido cliente lento
    - Tráfico elevado
  - Estos métodos de control de flujo se emplean también para corregir errores

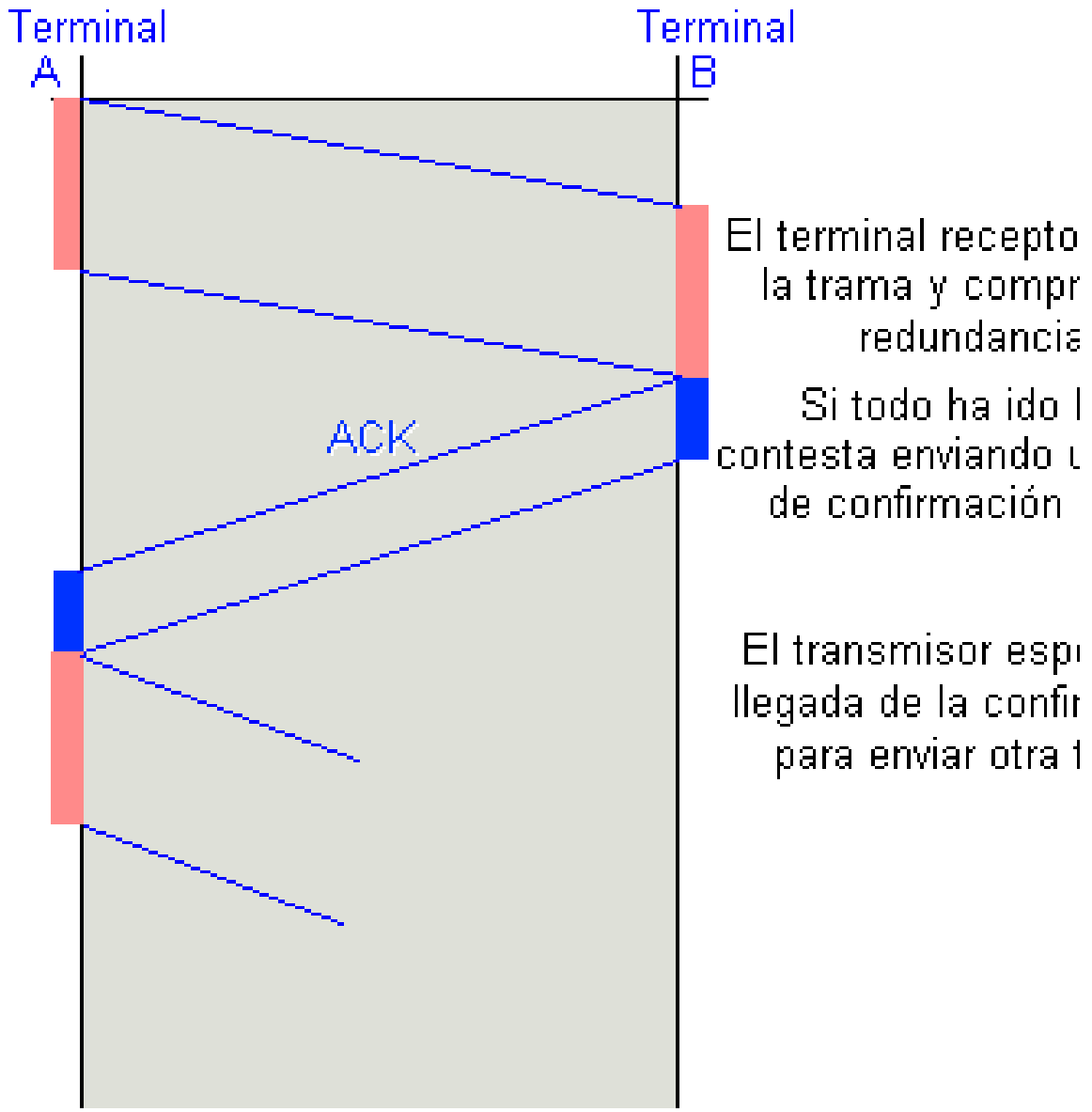
Control de flujo

- ARQ, Requerimiento automático de repetición
  - Se trata de sistemas de corrección hacia atrás
    - En estos sistemas la estación receptora que ha detectado la recepción de caracteres o bloques con errores procede a pedir a la estación emisora que repita lo recibido con error.
    - Debe observarse que esto requiere dar al sistema de comunicación algún medio para facilitar el diálogo entre la estación emisora y la estación receptora
    - El extremo receptor abandona el papel pasivo en la comunicación para participar en forma activa en el proceso.
- Existen dos estrategias principales en el diseño de los sistemas de corrección hacia atrás:
  - Pare y espere (stop and wait ARQ).
  - Ventana deslizante (Continuos ARQ).
  - Ambos trabajan en conjunto con métodos de detección de errores

Control de flujo

- Parada y espera
  - Las tramas se van intercambiando una a una.
  - Cuando el receptor recibe una trama procede a validarla
    - Si resulta que no contiene errores envía una señal de confirmación hacia el emisor
      - Esta señal se denomina ACK (acrónimo del término ingles acknowledge: confirmación).
    - Si hay errores se envía hacia el emisor una señal de recepción errónea
      - Denominada NACK (por negative acknowledge).
  - Mientras espera la recepción de ACK ó de NACK el emisor mantiene el mensaje enviado en un buffer
  - Cuando recibe NACK vuelve a enviar el contenido del buffe
  - Si recibe un ACK copia en el buffer la trama ó bloque siguiente y procede a enviarla

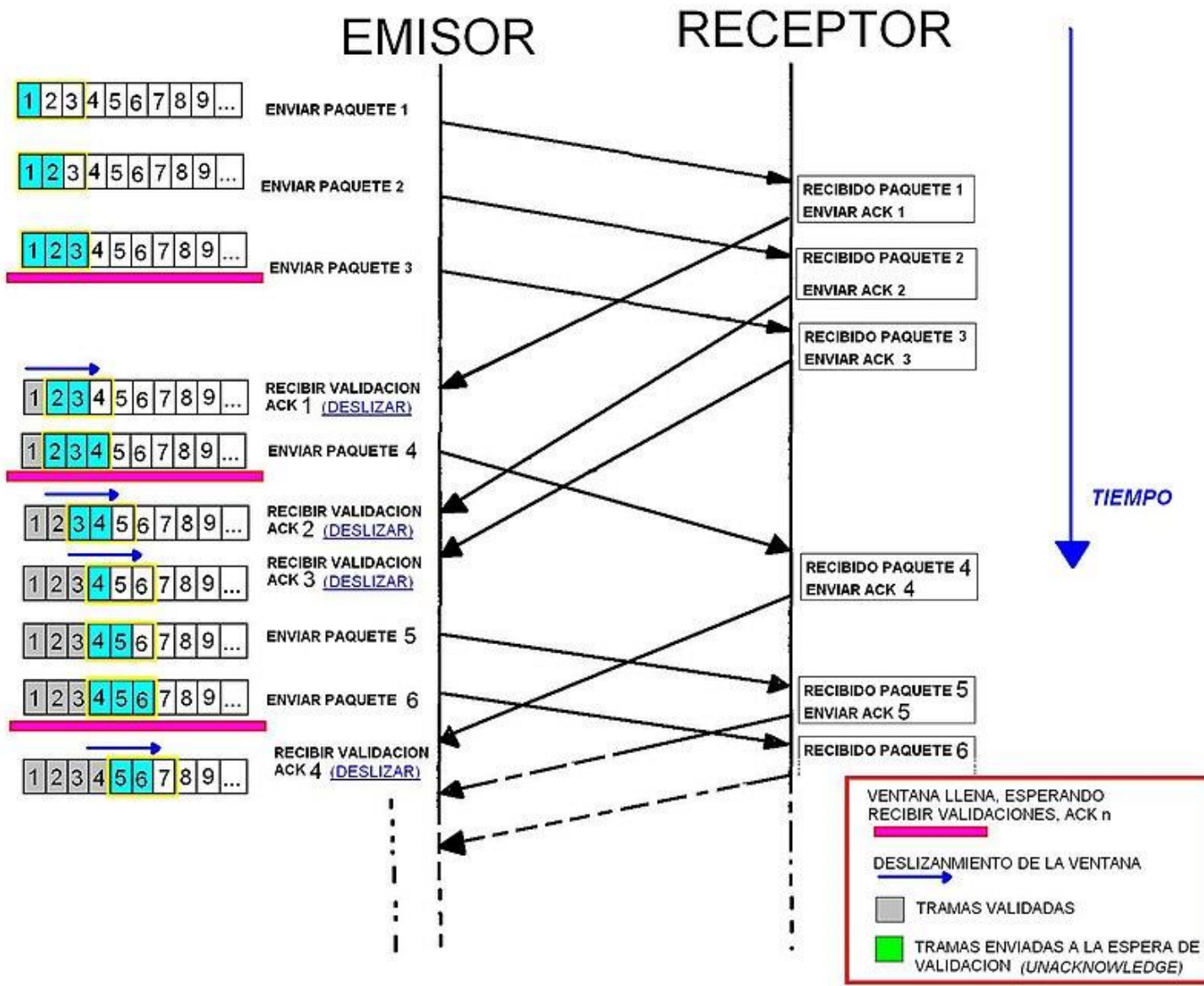
Control de flujo



Control de flujo

- El método de parada y espera tiene inconveniente de
  - Reducir el tiempo de utilización efectiva de los canales de comunicación
    - Cada mensaje debe ser confirmado individualmente
    - Todo se paraliza hasta que ello ocurre.
- Para corregir esto los métodos de ventana deslizante utilizan el mecanismo de:
  - Enviar continuamente la información sin esperar confirmación
    - Previamente se conviene en un número “m” que dará el número de mensajes al cabo de los cuales se va a enviar respuesta ACK ó NACK.
    - Cada bloque o trama contiene un número (o varios) de secuencia que la identifica.
    - Existen diversos métodos para enviar al ACK o el NACK, de los más conocidos:
      - Adelante y atrás N
        - En caso de error en el mensaje x, se pide que se retransmita la secuencia a partir de x retrocediendo n = m - x.
      - Rechazo selectivo
        - Se reenvía solamente la trama defectuosa

Control de flujo

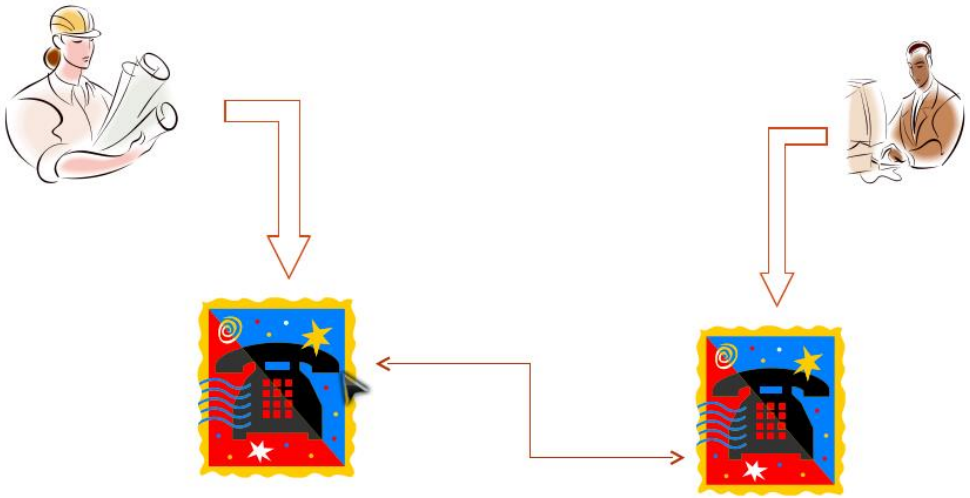


Control de flujo

- En muchos casos, y mientras espera la llegada de NACK ó ACK, el emisor arranca un temporizador.
  - El temporizador se detendrá al llegar cualquiera de las señales de confirmación
  - Si el temporizador llega a su término sin recibirlas pueden ocurrir dos cosas:
    - La primera consiste en abortar el proceso de comunicación dado que no hay respuesta del receptor
    - La segunda en enviar nuevamente la trama sin confirmar y arrancar nuevamente el temporizador.
      - Si esta situación se repite varias veces el sistema aborta la comunicación.
- En el receptor también se arranca un temporizador ya sea al recibir la trama o al enviar una señal ACK
  - Al vencer este tiempo el receptor procede a reenviar una señal de ACK u otra predeterminada
  - Si se repite esto un cierto número de veces se procede a abortar la comunicación pues no se recibe respuesta del emisor.
- Generalmente tanto el receptor como el emisor dispondrán de un contador
  - Para determinar el número de veces que se ha intentado retransmitir una trama sin éxito.
  - De alcanzar el contador el valor prefijado se procede a abortar la comunicación.

Control de errores

Introducción



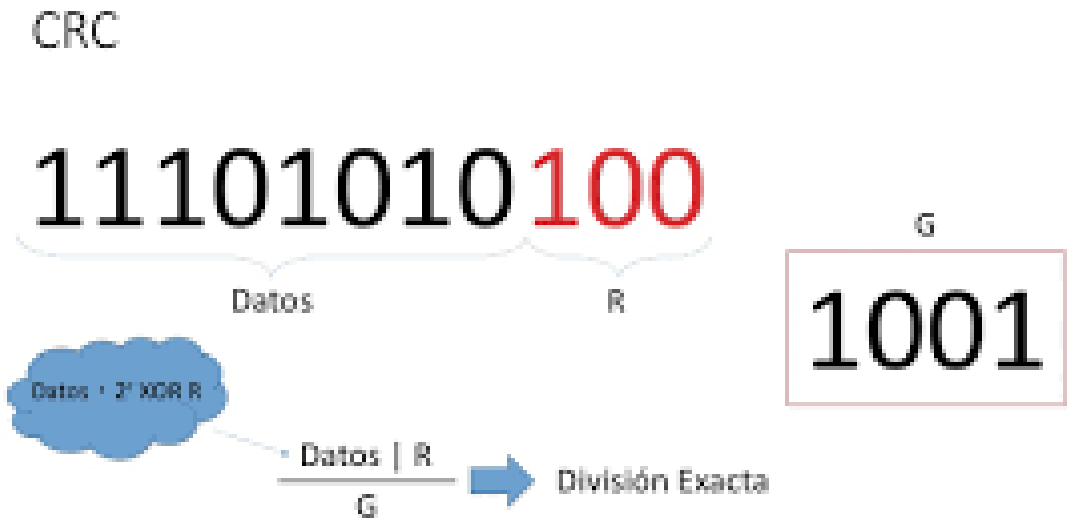
- Las causas por las que la señal electromagnética se deteriora al viajar por el canal de comunicación son:
  - Distorsión
  - Atenuación
  - Limitación del ancho de banda
  - Ruido
  - Interferencia
  - Diafonía.
- Esta degradación de la señal puede hacer que se reciban en el receptor un carácter distinto al que fue emitido por el extremo transmisor, entonces se ha producido un error.

Definición

- Es imposible evitar que ocurran errores
  - Un buen diseño los minimizará.
  - En primer lugar, determinar la presencia de los errores
    - Aquí es donde aparecen las técnicas de detección de errores
  - Luego tratar de corregirlos, lo que da lugar a la corrección de errores
- La denominación genérica de estas técnicas es **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, o un grupo de datos, caso sincrónico, presentan algún o 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)

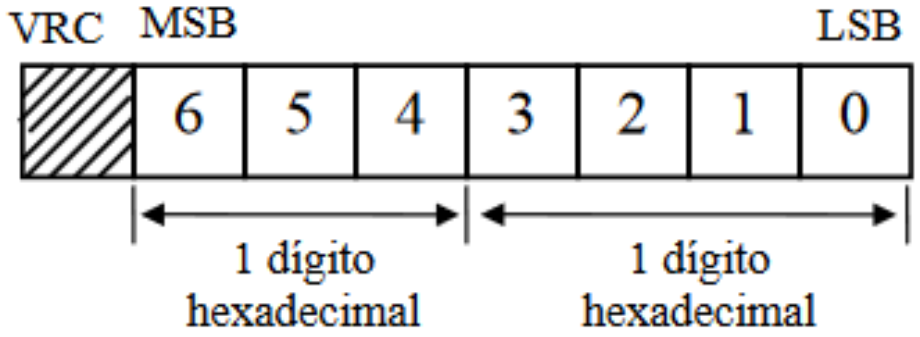


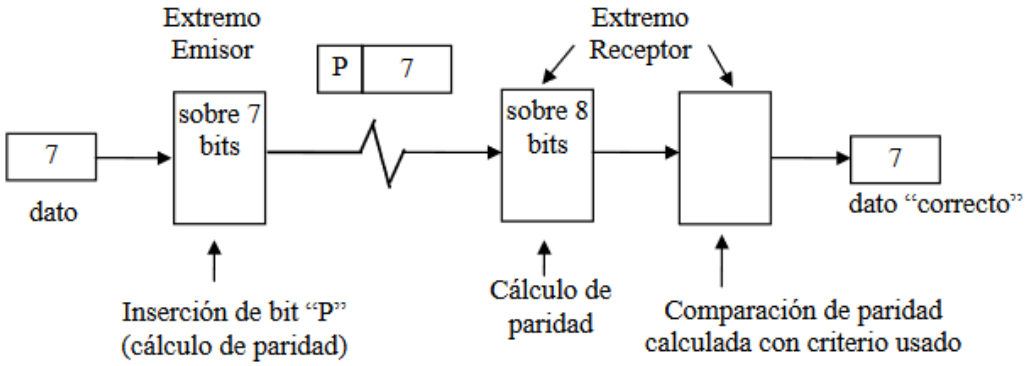
Detección de errores

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

- Checksum
  - Método simple orientado al mensaje
    - 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

Detección de errores

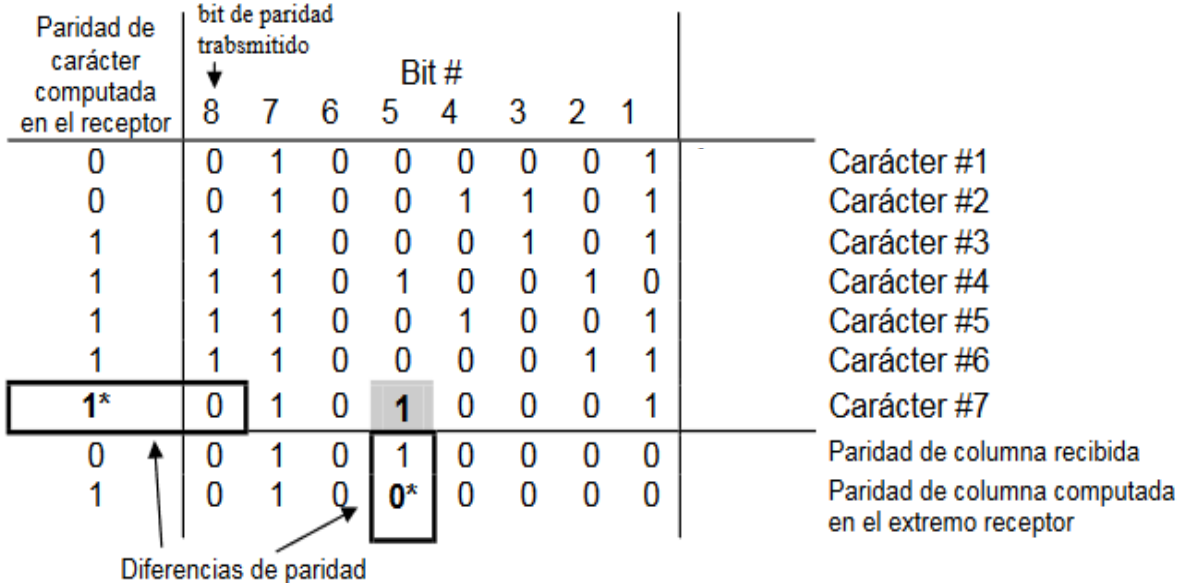




- Chequeo de paridad vertical o paridad de carácter (VRC).
  - Este método, como todos los que siguen, hace uso del agregado de bits de control.
  - Se trata de la técnica más simple usada en los sistemas de comunicación digitales (Redes Digitales, Comunicaciones de Datos)
    - Paridad par
    - Paridad impar

Detección de errores

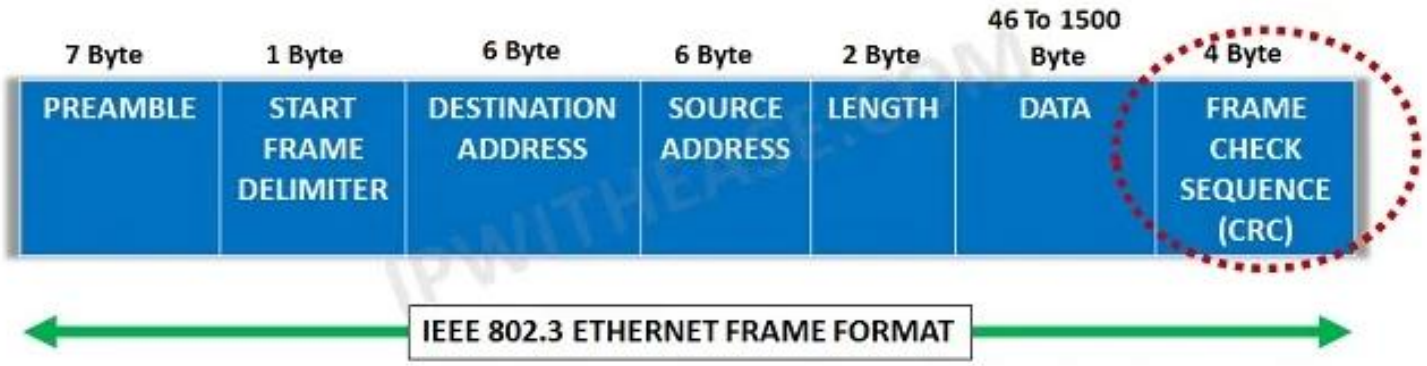
- Chequeo de paridad bidimensional
  - Es la combinación de verificación de paridad vertical y paridad horizontal
  - Proporciona mayor protección
  - No supone gran consumo de recursos
  - Aunque tiene la misma sencillez conceptual de los métodos de paridad lineal, es más complicado y por ello menos popular.



Detección de errores

- Métodos de paridad
  - Los métodos basados en el uso de paridad son sencillos de comprender y de implementar
  - Suministran cierto grado de protección contra los errores
    - Son limitados
    - Su efectividad es cuestionable en determinadas aplicaciones.
    - Por ello se utilizan solamente cuando resulta muy complicado ó muy costoso implementar otros métodos.
  - Paridad vertical requiere que cada carácter lleve su protección contra errores
    - 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:
    - Detección más segura de errores
    - Eficiencia

Detección de errores



- Métodos de Código de Redundancia Ciclíca (CRC)
  - Cumplen con requisitos de
    - Detección más segura de errores
    - Eficiencia
  - Se basan en propiedades matemáticas de los códigos empleados para la transmisión de datos
  - Principio de funcionamiento
    - 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 clave, por ejemplo el 5.
    - Ahora dividimos el número a transmitir entre la clave y calculamos el resto: 23/5 = 4 resto 3
    - 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.

Detección de errores

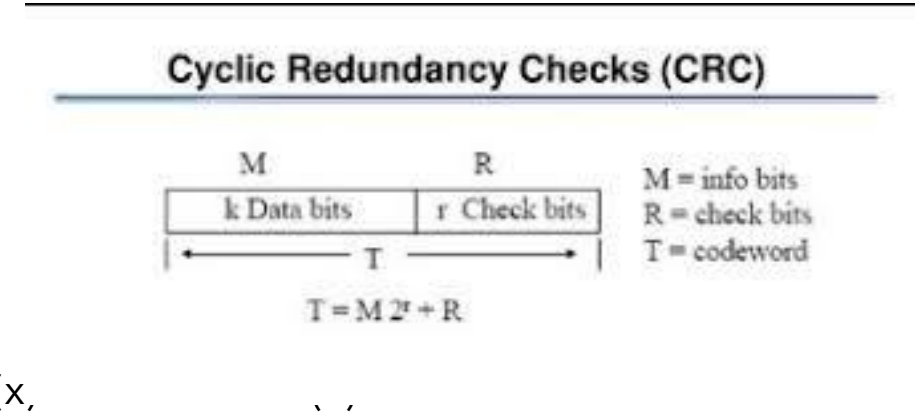
- CRC
  - Emplean el principio de las propiedades de la operación módulo (se define módulo de dos números a mod b al resto de dividir a por b).
  - Para ello se considera la cadena de bits a transmitir como el conjunto de coeficientes de un polinomio
    - Por ejemplo, al enviar 1100100110, el polinomio equivalente P(x) es: $P(x) = x^9 + x^8 + x^5 + x^2 + x.$

$$1100100110 \Rightarrow 1x^9 + 1x^8 + 0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 1x + 0$$

  - Se debe 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:

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

Detección de errores



- CRC
  - El procedimiento es el siguiente:
    - Se toma el polinomio de datos P(x)
    - Se multiplica por x^k, (el número de ceros que se agrega a P(x,
      - donde k es el exponente más alto de G(x)
    - Este polinomio así construido se divide por G(x), división en módulo 2
    - Se obtiene un polinomio resto R(x),llamado BCS (Block Character Sequence)
    - 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 se supone es $x^k P(x)$
    - Se divide nuevamente por G(x)
    - Se calcula un polinomio resto que si coincide con el R(x) recibido indicará que no hay errores.

Detección de errores

- CRC (División módulo 2)
  - La operación de división empleada en CRC se denomina de módulo 2.
    - Es diferente de la división binaria:
      - Las restas durante la división(o sea la obtención del resto parcial o 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 dá 0, 1 y 1 dá 0).
    - Sí el primer bit del resto parcial es 1 y queda uno o 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

Detección de errores

- CRC Ejemplo
  - Determinar el BSC(Block Character Sequence), para los siguientes polinomios generadores de datos y CRC

$$\text{datos } P(x) = x^7 + x^5 + x^4 + x^2 + x^1 + x^0 \quad \text{ó} \quad 10110111$$

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

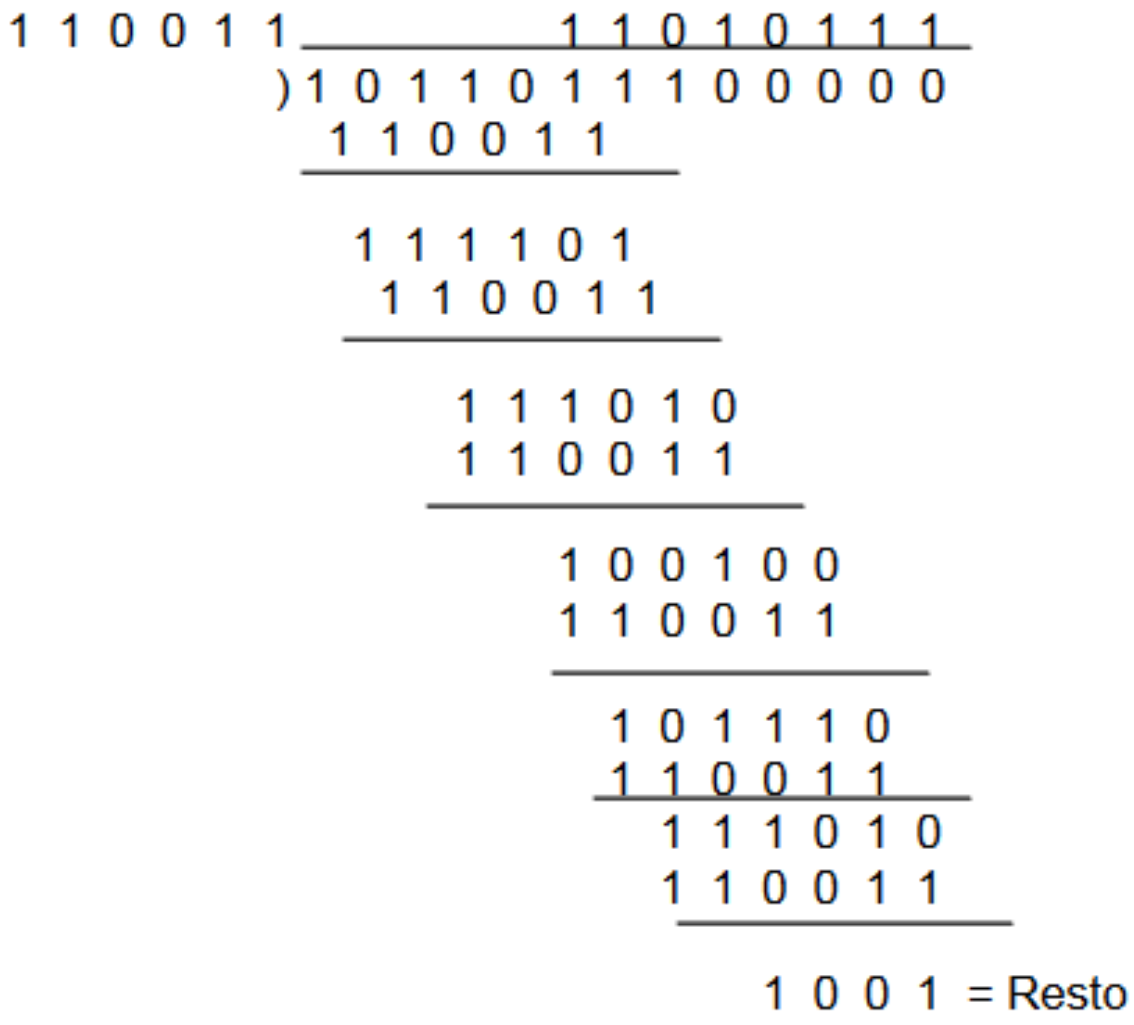
  - Solución
    - Primero P(x) es multiplicado por el número de bits en el código CRC, 5

$$x^5(x^7 + x^5 + x^4 + x^2 + x^1 + x^0) = x^{12} + x^{10} + x^9 + x^7 + x^6 + x^5$$

$$= 1011011100000$$

    - Al dividir este polinomio por G(x) obtenemos R(x) o BCS que resulta

Detección de errores



- CRC (ejemplo)
  - Resto 01001 significa R(x) = $0x^4 + 1x^3 + 0x^2 + 0x + 1$
  - Se transmite entonces

    $T(x) = x^k P(x) + R(x)$ o sea 1011011101001
  - Suponiendo que se reciba $T_R(x)$ 1011011101001 (lo mismo que se transmitió)
    - Al dividir TR(x) por G(x) el resto será cero
      - Indicando que se recibió correctamente
    - O 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)

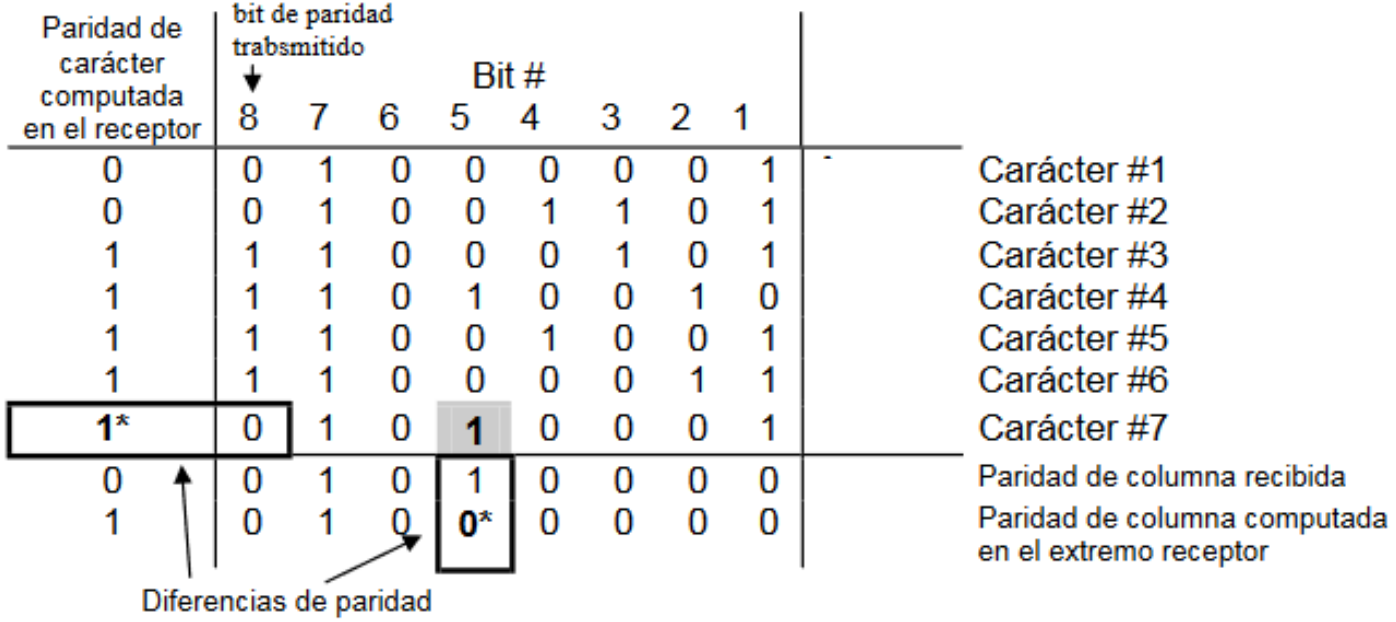
Corrección de errores

- Detectar los errores no es suficiente, hay que corregirlos.
- Dos formas principales de corrección de errores son:
  - Requerimiento automático de repetición: (ARQ) (Automatic Request for Repeat).
    - Pare y espere (stop and wait ARQ).
    - Envío continuo (Continuos ARQ).
      - Adelante y Atrás N
      - Rechazo selectivo
    - Estos métodos se analizaron en control de flujo
  - Corrección de errores hacia adelante: FEC (Forward Error Correction).
    - Bloque
      - Paridad Bidimensional
      - Hamming
      - Otros

Corrección de errores

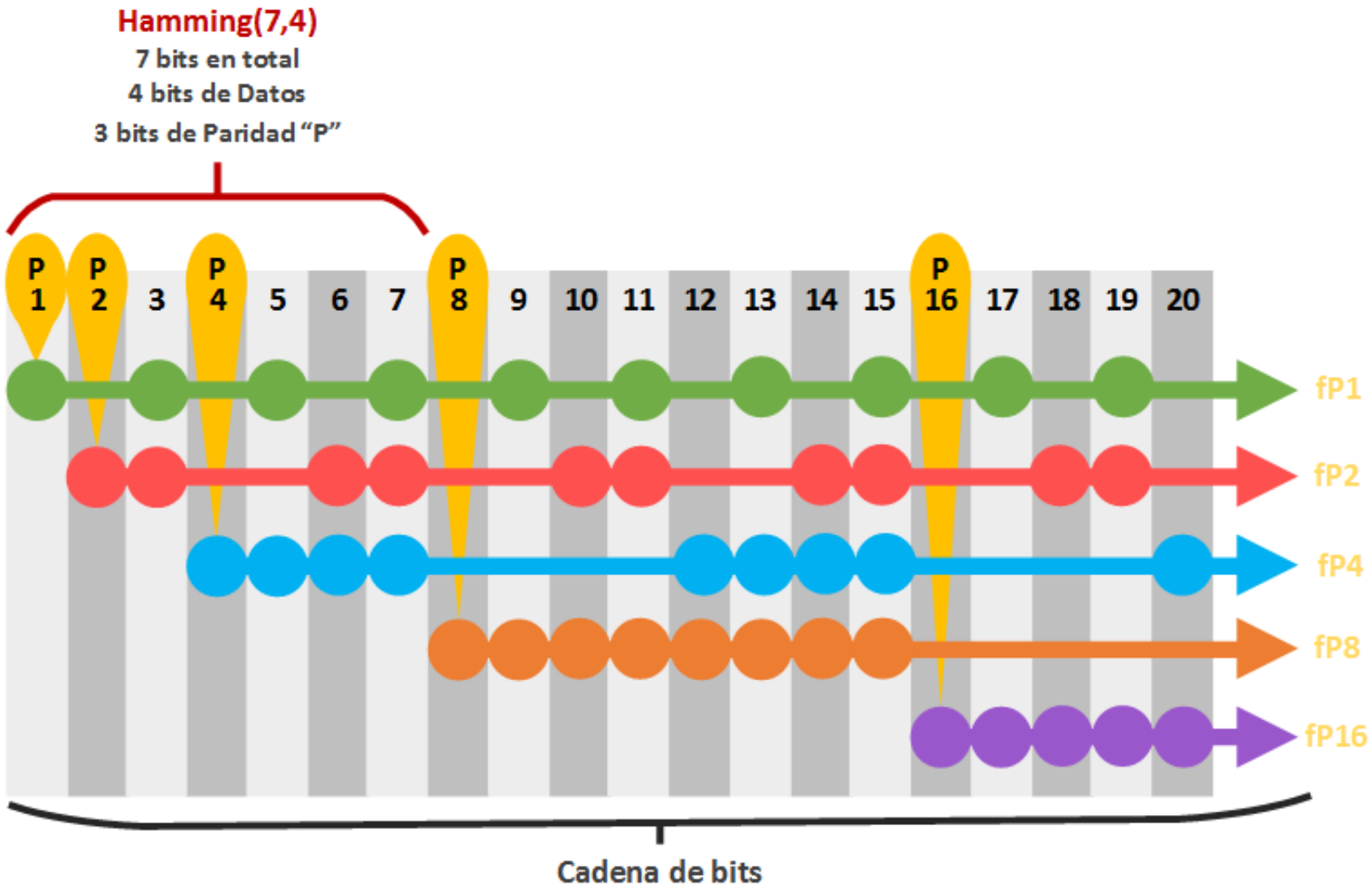
- Se basan en la idea de reconstruir la información deteriorada por los errores
- La reconstrucción tiene lugar en el equipo receptor
  - Deben emplearse en los códigos un gran número de bits lo que disminuye la efectividad del código.
- En la detección por paridad bidimensional se menciona que este método puede corregir el error si solo se presenta uno
- Otro método empleando paridad es el Código Hamming que igualmente solo puede corregir cuando se presenta un solo error

Corrección de errores



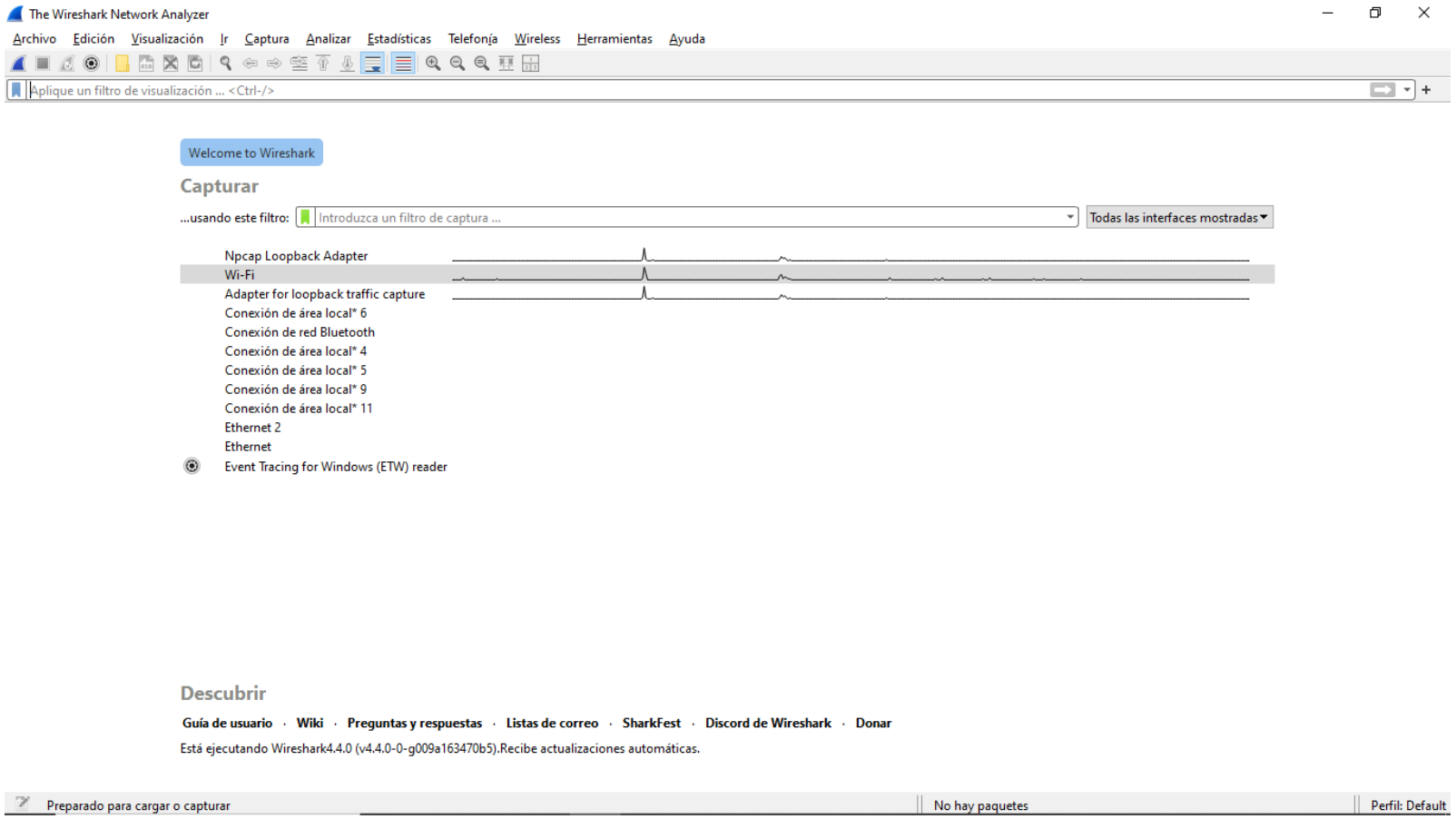
- Paridad bidimensional
  - Requiere de una matriz de paridad

Corrección de errores



- Código de Hamming

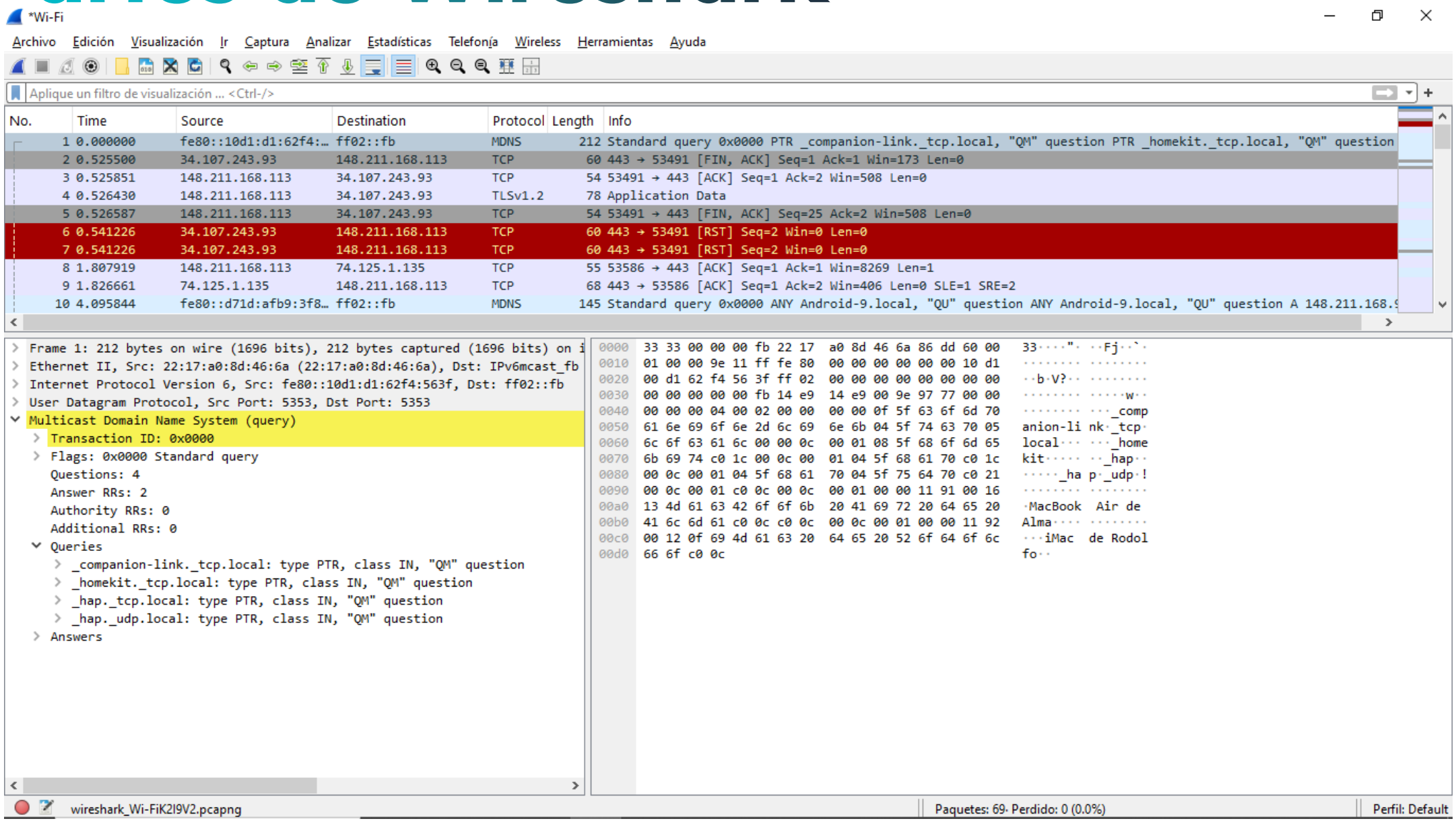
Whireshark



Inicio de captura

- Elegir la interfaz de captura
- Elegir filtro de captura si así se desea

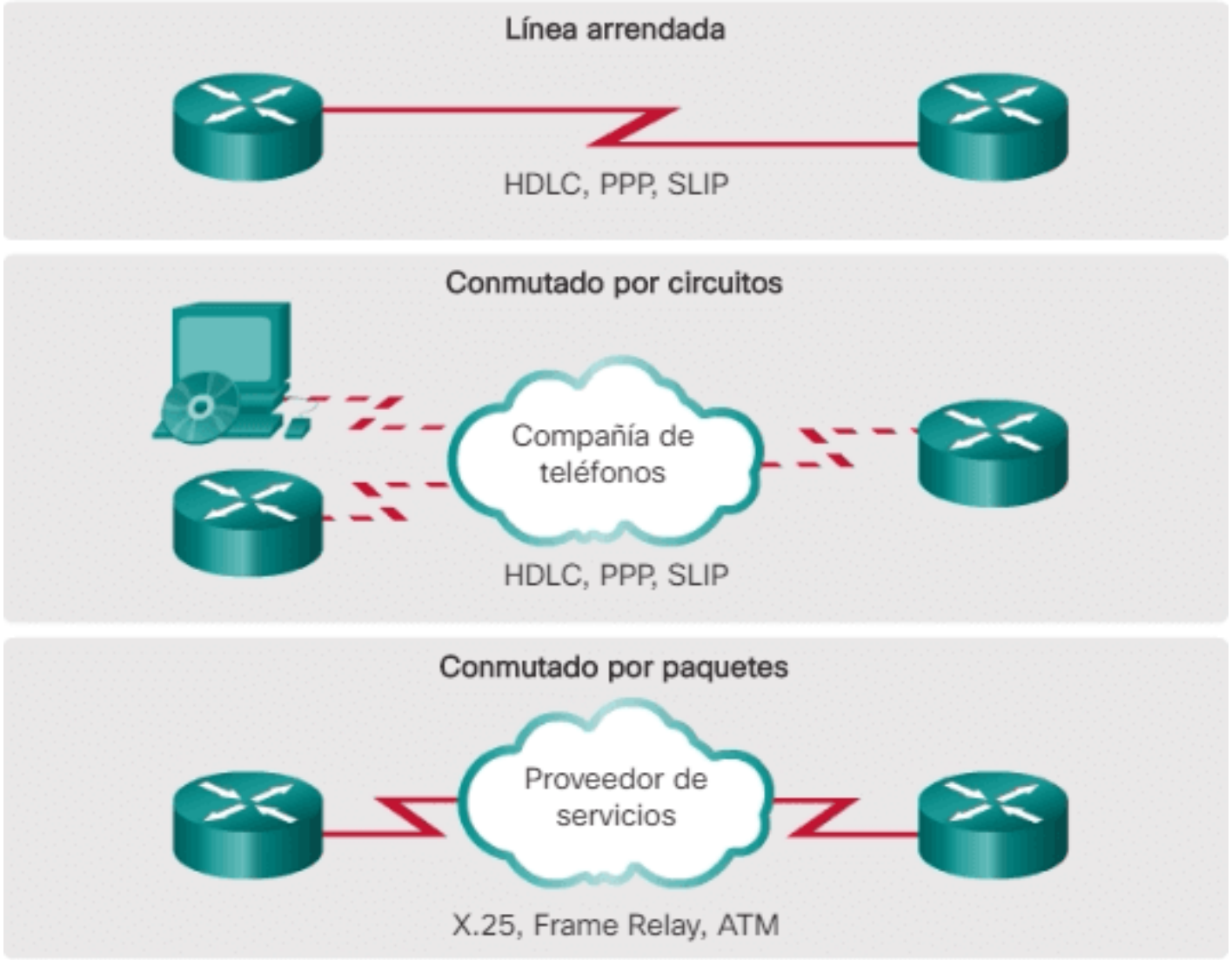
Partes de Wireshark



Filtro de display

- Filtro DNS
- Filtro HTTP
- Filtro de sincronización
  - tcp.flags.syn
  - tcp.flags.syn==1 && tcp.flags.ack==1
- Filtro por puerto
  - tcp.port==80
  - Tcp.srcport==80
- Estadísticas
  - Jerarquía de protocolo
  - Lista de propiedades del archivo
  - Conversaciones

HDLC y otros protocolos

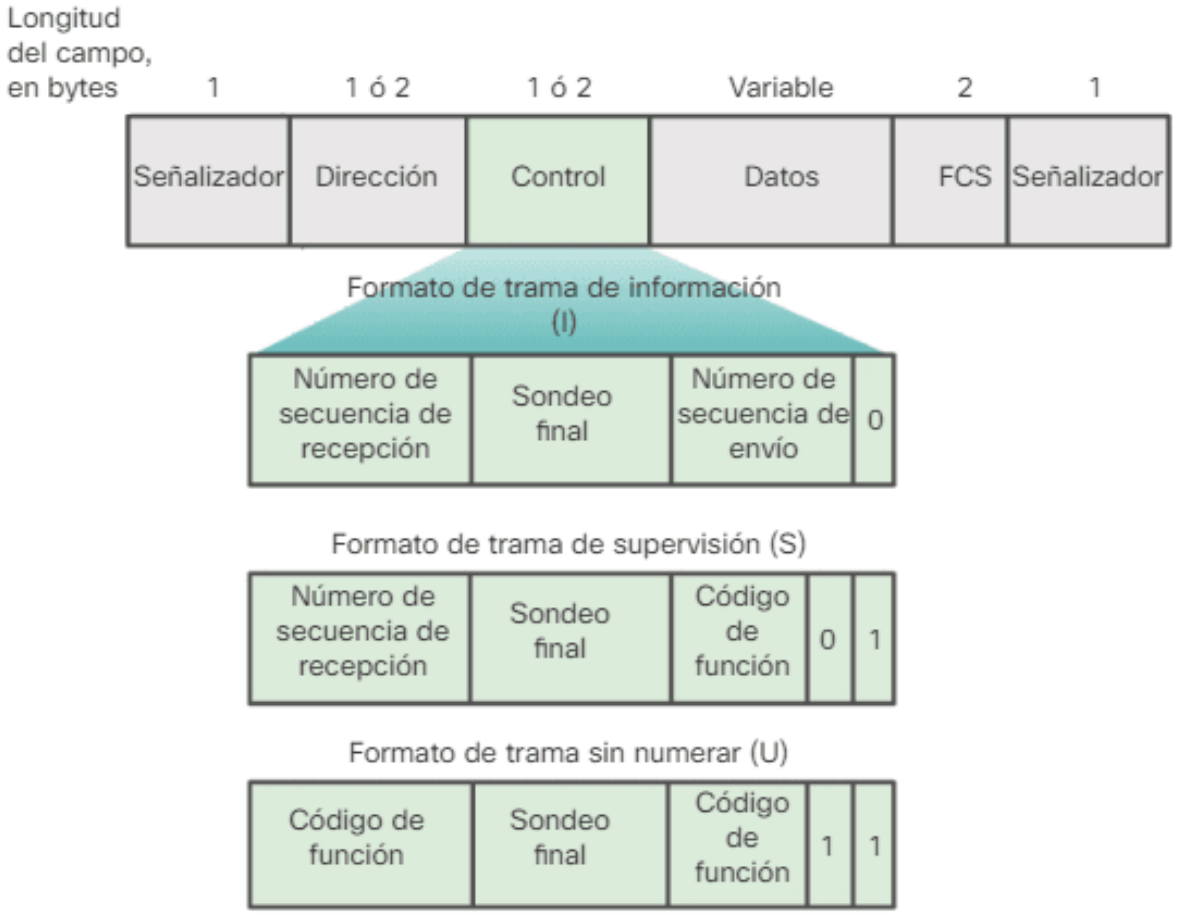


Protocolos de encapsulación WAN

- En cada conexión WAN, se encapsulan los datos en las tramas antes de cruzar el enlace WAN.
- Para asegurar que se utilice el protocolo correcto, se debe configurar el tipo de encapsulación de capa 2 correspondiente.
- La opción de protocolo depende de la tecnología WAN y el equipo de comunicación.

Tipos de Protocolo WAN

- **HDLC**:
  - Es el tipo de encapsulación predeterminado en las conexiones
    - Punto a punto
    - Enlaces dedicados
    - Conexiones conmutadas por circuitos.
  - HDLC es la base para PPP síncrono que usan muchos servidores para conectarse a una WAN, generalmente Internet.



Otros tipos de protocolos WAN

- **PPP**: proporciona conexiones de router a router y de host a red a través de circuitos síncronos y asíncronos.

  PPP funciona con varios protocolos de capa de red, como IPv4 e IPv6.

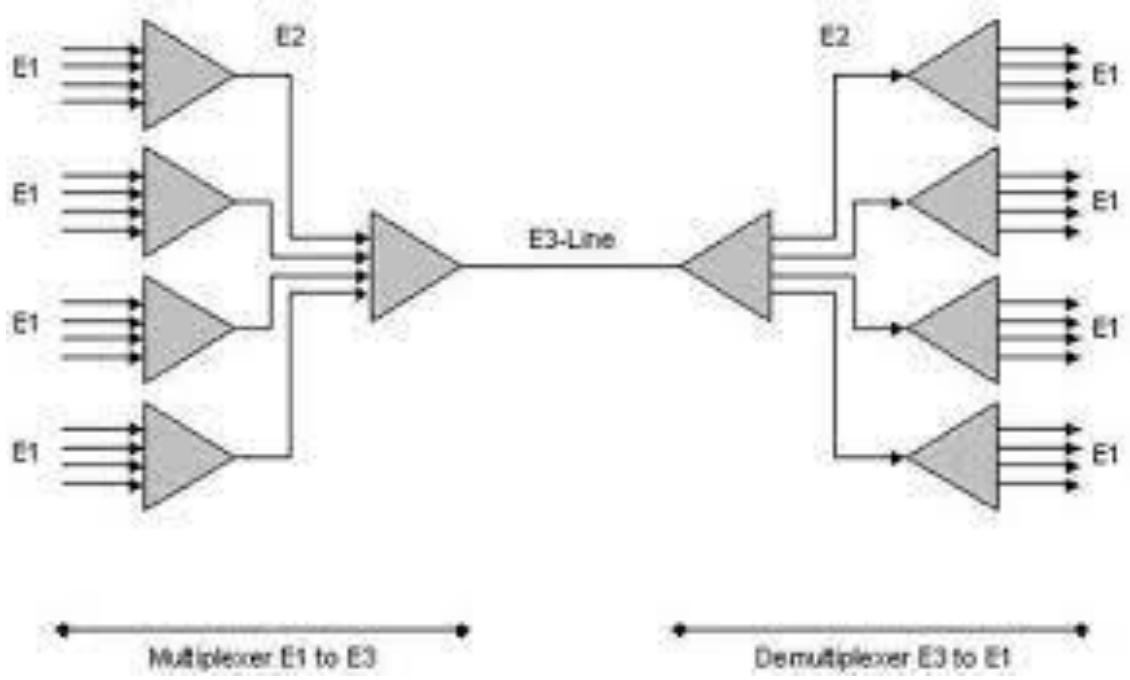
  Utiliza el protocolo de encapsulación HDLC, pero también tiene mecanismos de seguridad incorporados como PAP y CHAP.
- **Protocolo de Internet de línea serial (SLIP)**: es un protocolo estándar para conexiones seriales punto a punto mediante TCP/IP. PPP reemplazó ampliamente al protocolo SLIP.
- **Procedimiento de acceso al enlace balanceado (LAPB) X.25**: es un estándar del UIT-T que define cómo se mantienen las conexiones entre un DTE y un DCE para el acceso remoto a terminales y las comunicaciones por computadora en las redes de datos públicas.
- **Frame Relay**: es un protocolo de capa de enlace de datos conmutado y un estándar del sector que maneja varios circuitos virtuales. Frame Relay elimina algunos de los procesos prolongados (como la corrección de errores y el control del flujo) empleados en X.25.
- **ATM**: es el estándar internacional de retransmisión de celdas en el que los dispositivos envían varios tipos de servicios (como voz, video o datos) en celdas de longitud fija (53 bytes). Las celdas de longitud fija permiten que el procesamiento se lleve a cabo en el hardware, lo que disminuye las demoras en el tránsito.

Multicanalización

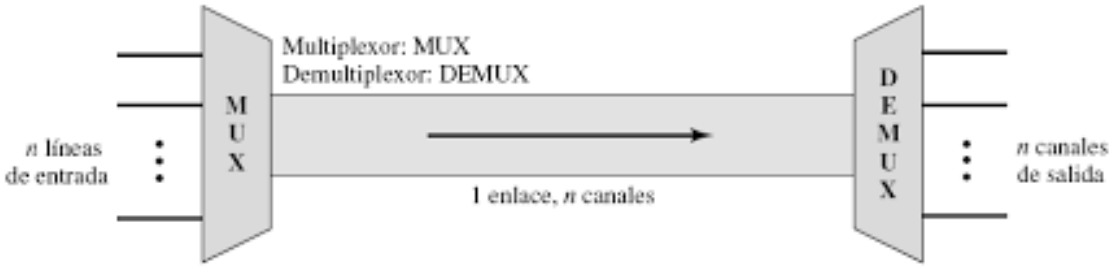
Multiplexión

Definición

- La Multiplexación es la combinación de dos o más canales de información en un solo medio de transmisión usando un dispositivo llamado multiplexor.
  - Procedimiento por el cual diferentes canales pueden compartir un mismo medio de transmisión de información.



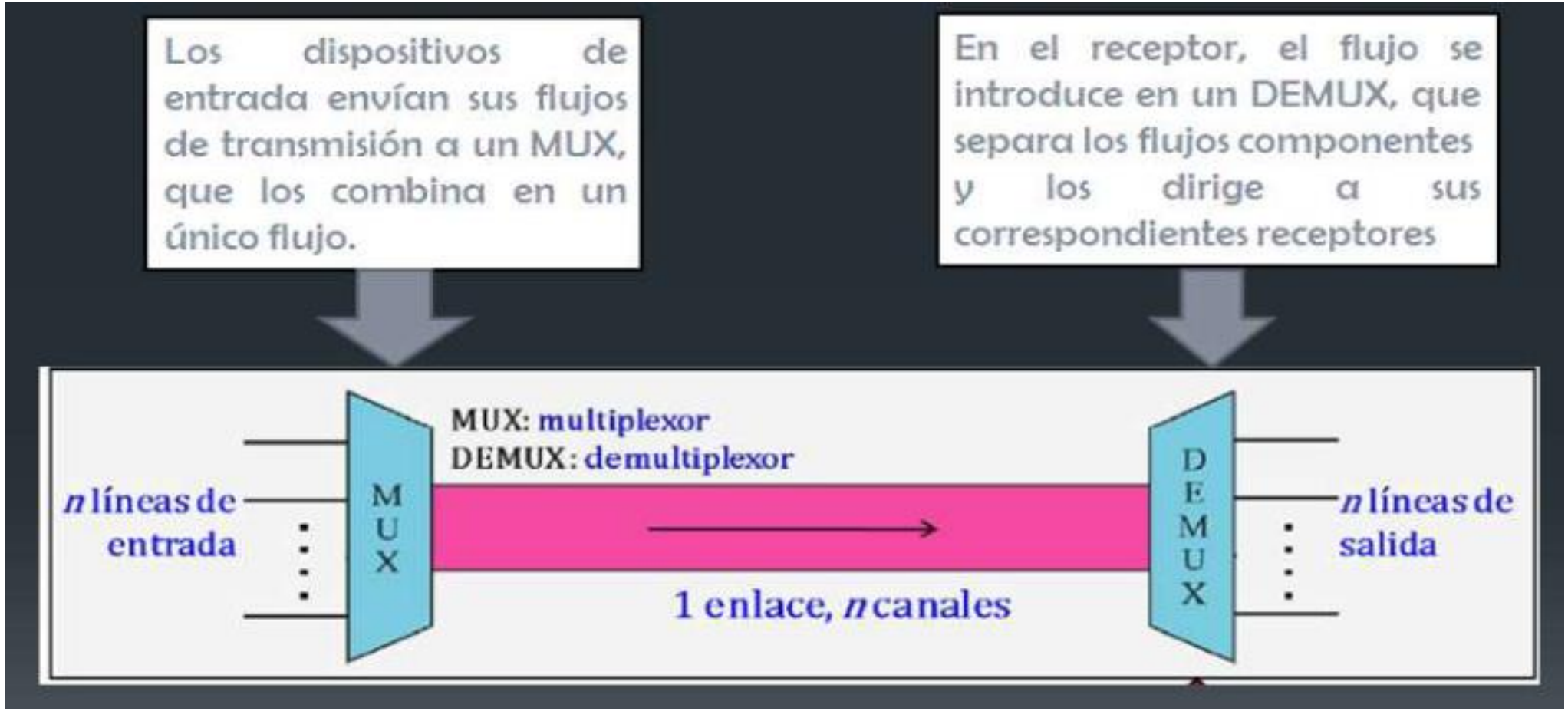
Objetivo de la multicanalización



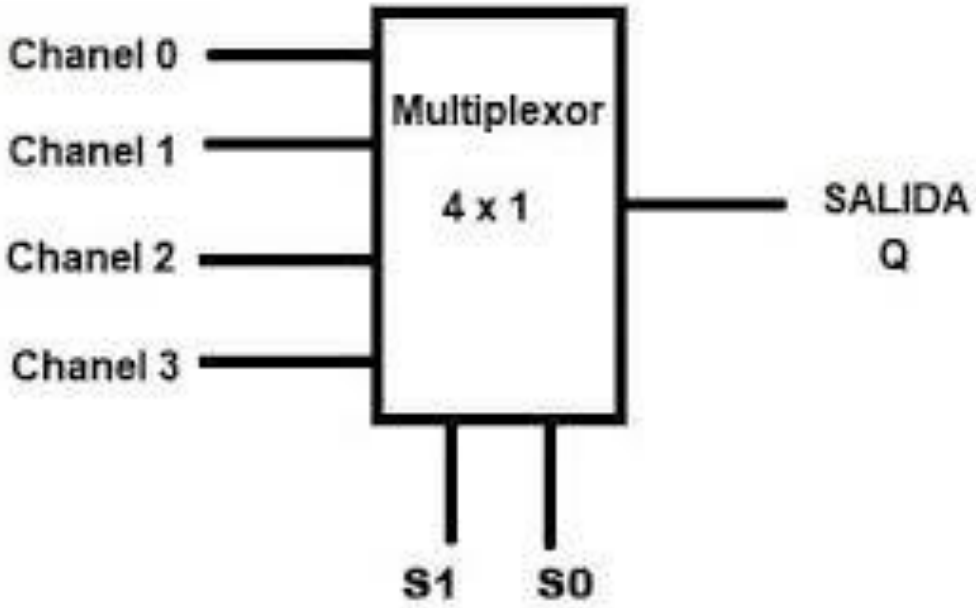
- Compartir la capacidad de transmisión de datos sobre un mismo enlace para aumentar la eficiencia (sobre todo en líneas de grandes distancias).
- Minimizar la cantidad de líneas físicas requeridas y maximizar el uso del ancho de banda de los medios

Multicanalización ¿qué es?

- Optimización de la utilización del medio de transmisión
  - Mediante un conjunto de técnicas que permite la transmisión simultanea de múltiples señales a través de un único enlace



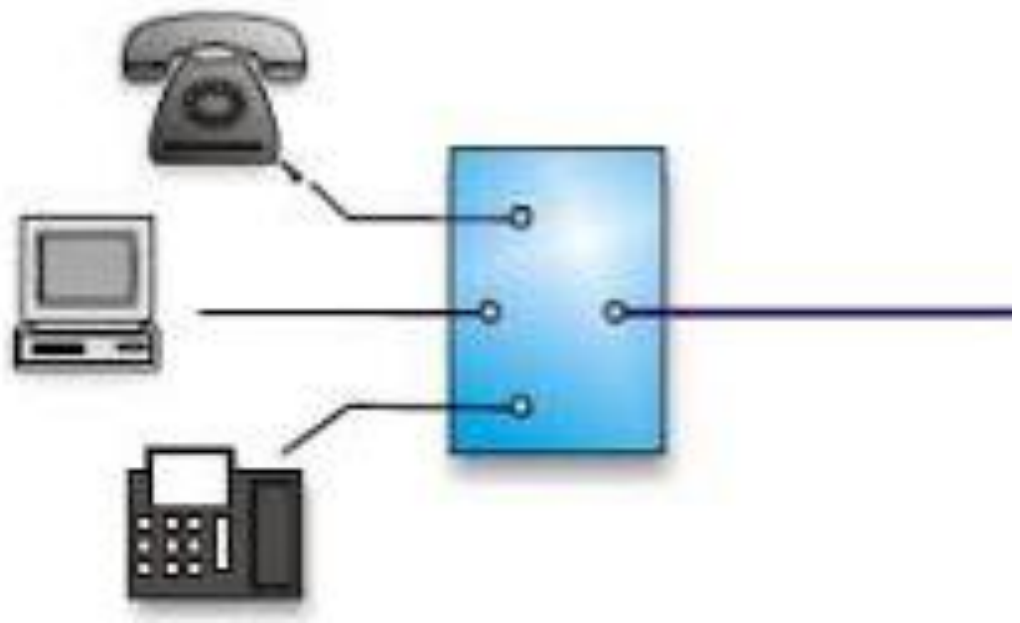
Definiendo un multiplexor



- Es un dispositivo que puede recibir varias entradas y transmitirlas por un medio de transmisión compartido
  - Divide el medio de transmisión en múltiples canales, para que varios nodos puedan comunicarse al mismo tiempo.

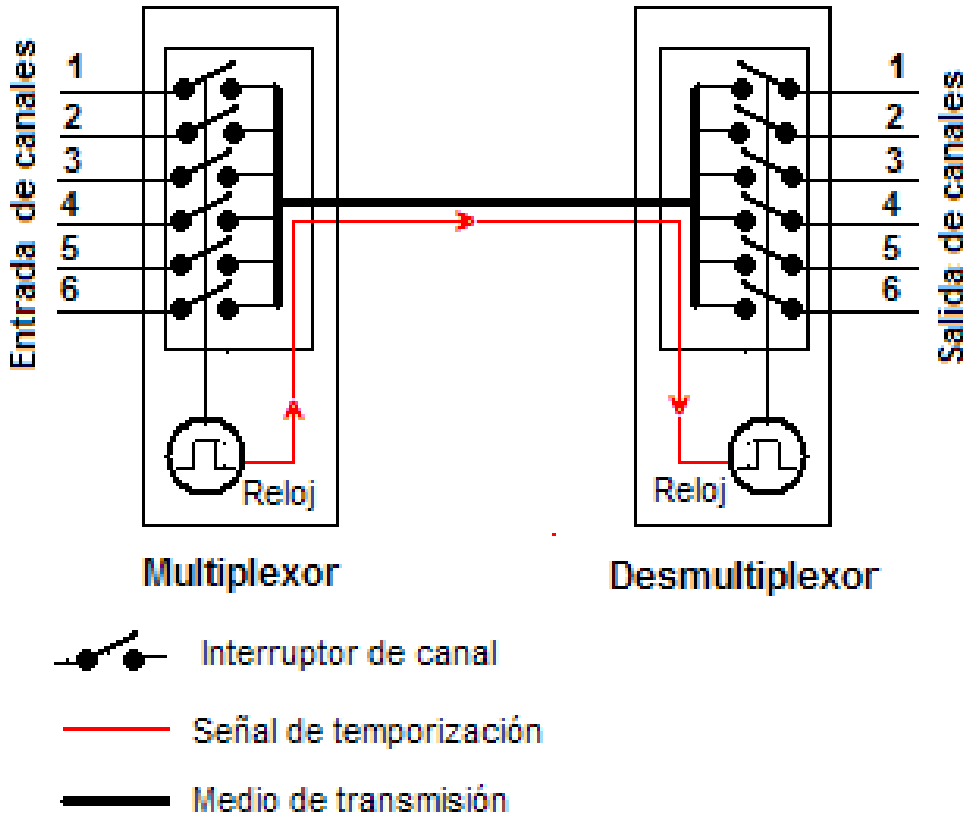
Función de un multiplexor

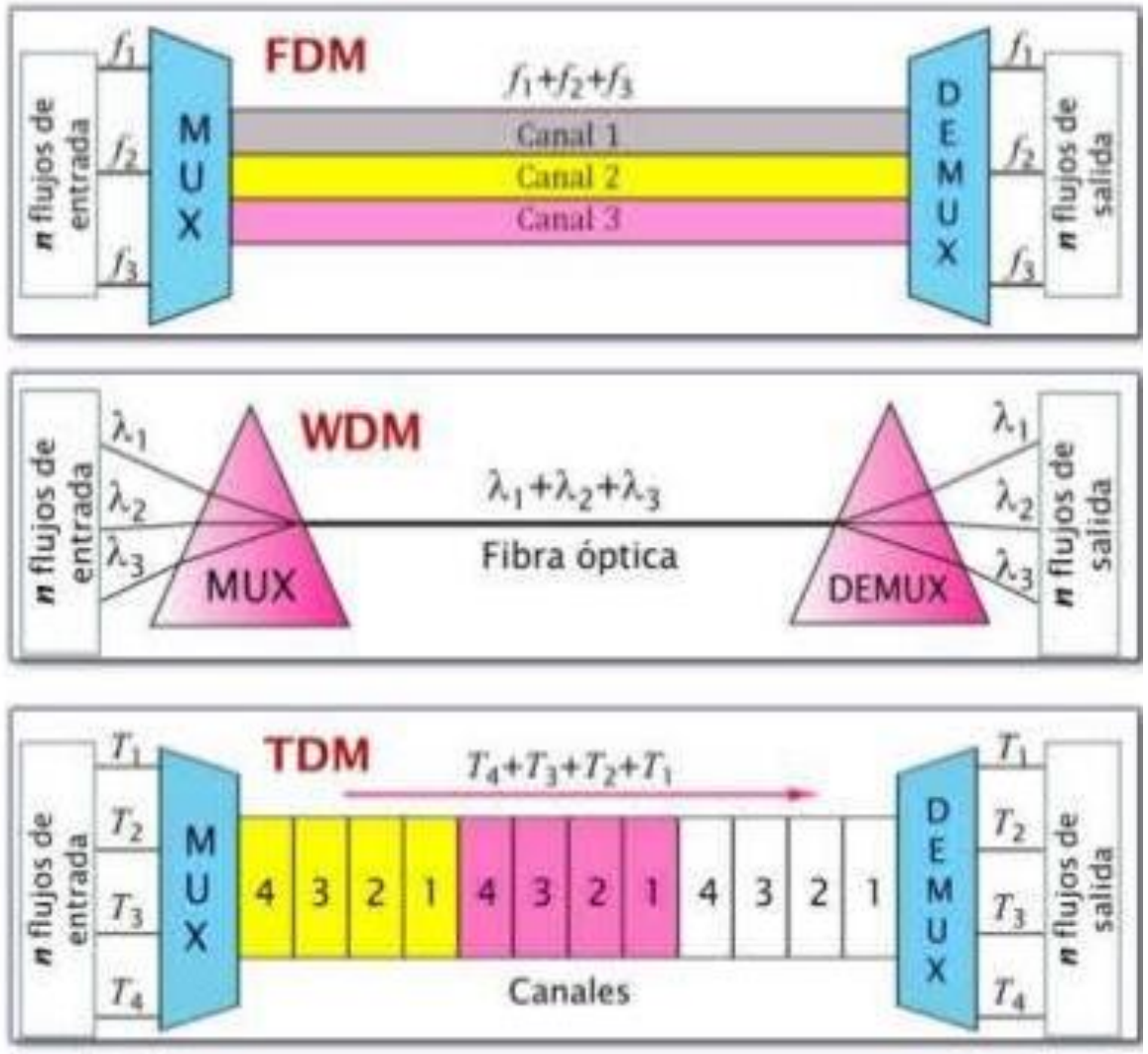
- La función de un multiplexor da lugar a diversas aplicaciones:
  - Selector de entradas.
  - Serializador: Convierte datos de paralelo a serial.
  - Transmisión multiplexada: En una misma línea de conexión, transmite diferentes datos de distinta procedencia.
  - Realización de funciones lógicas: Utilizando inversores y conectando a 0 ó 1 las entradas se puede diseñar funciones de un modo más compacto, que utilizando puertas lógicas.



Implementación de la multicanalización

- Las entradas de 6 canales llegan a los interruptores de canal controlados por una señal de reloj
  - Cada canal es conectado al medio de Tx durante un tiempo determinado por la duración de los impulsos del reloj.
  - El Desmultiplexor realiza la función inversa: conecta al medio de Tx, secuencialmente con la salida de cada uno de los 5 canales mediante interruptores controlados por el reloj del D.
  - El reloj del extremo receptor funciona de forma sincronizada con el del Multiplexor mediante señales de temporización que son trasmitidas a través del propio medio de Tx



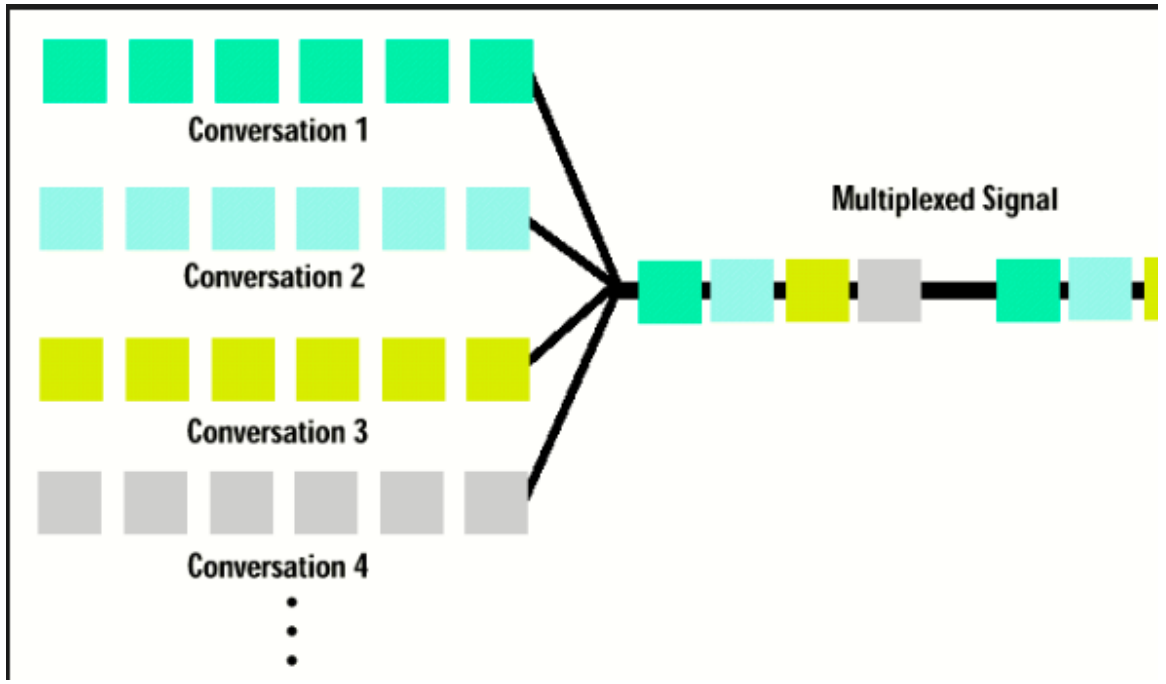


Métodos de multicanalización

- La Multiplexación o multicanalización es la transmisión de información, de más de una fuente a más de un destino, por el mismo medio de transmisión.
- Los principales métodos de realizar este proceso son:
  - La multiplexación de división de frecuencia (FDM: Frequency Division Multiplexing),
  - La multiplexación por división de código (CDM: Coded Division Multiplexing),
  - La multiplexación por división de longitud de onda (WDM: Wavelength Division Multiplexing)
  - La multiplexación por división de tiempo (TDM: Time Division Multiplexing)

Multicanalización por División de Tiempo (TDM)

- La multiplexación por división de tiempo es una técnica para compartir un canal de transmisión entre varios usuarios.
- Consiste en asignar a cada usuario, durante unas determinadas "ranuras de tiempo", la totalidad del ancho de banda disponible.

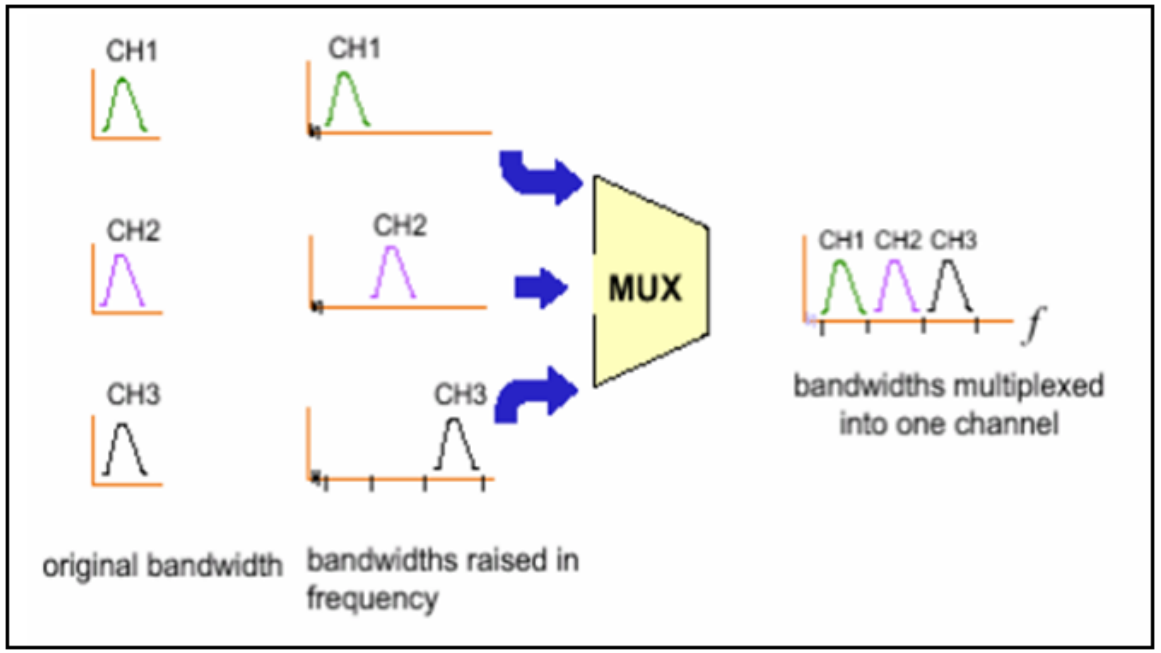


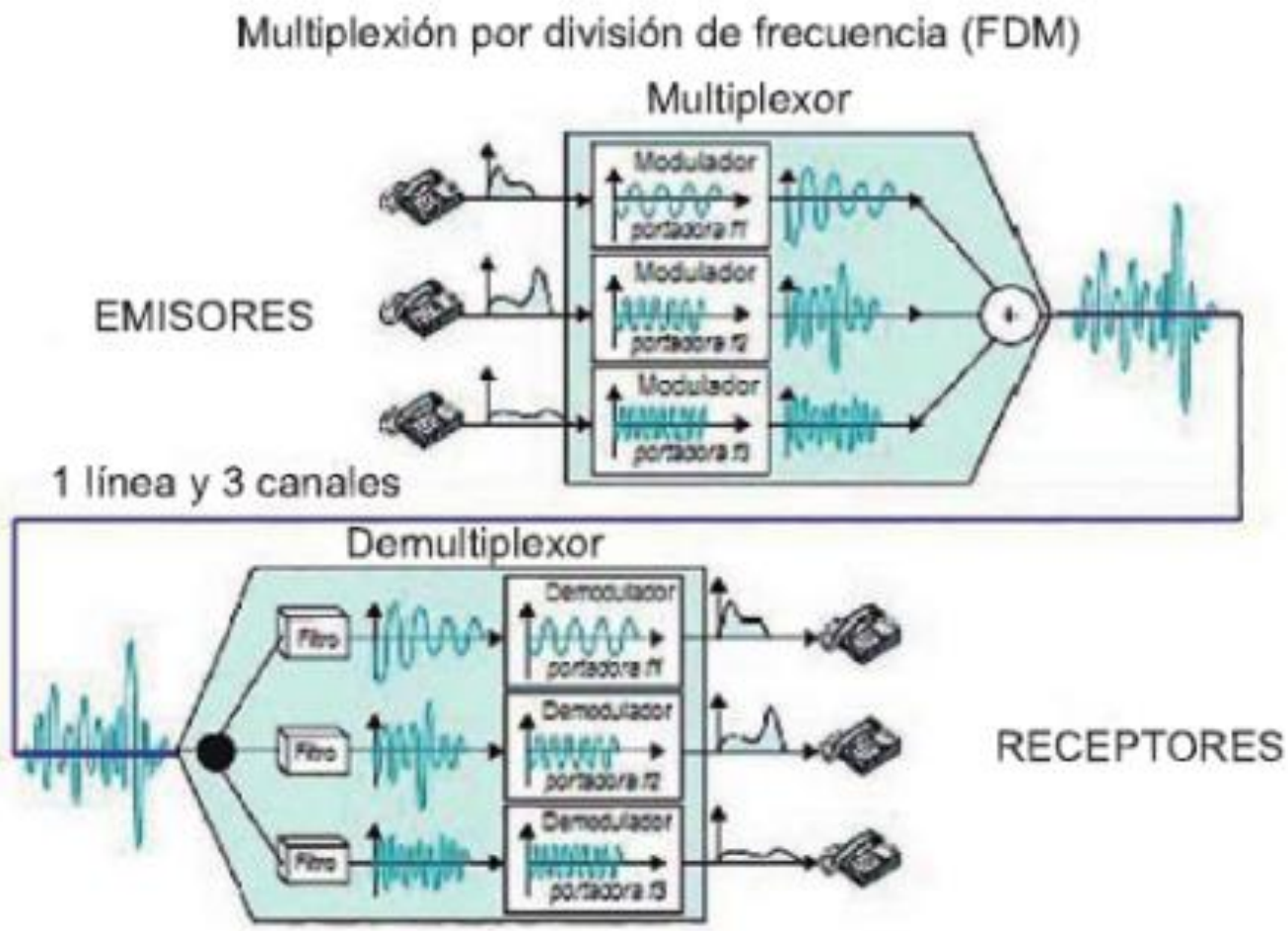
Multicanalización por división de tiempo (TDM)

- Características
  - Consiste en ocupar un canal de transmisión a partir de distintas fuentes, mejor aprovechamiento del medio de transmisión.
  - El ancho de banda total del medio de transmisión es asignado a cada canal durante una fracción del tiempo total (intervalo de tiempo).
- Ventajas de TDM
  - El uso de la capacidad es alto.
  - Cada uno para ampliar el número de usuarios en un sistema en un coste bajo.
- Desventajas de TDM
  - La sensibilidad frente a otro problema de usuario es alta.
  - El coste inicial es alto.
  - La complejidad técnica.

Multicanalización por división de frecuencia (FDM)

- Esta técnica que consiste en:
  - Dividir mediante filtros el espectro de frecuencias del canal de transmisión
  - Desplazar la señal a transmitir dentro del margen del espectro correspondiente mediante modulaciones
  - Cada usuario tiene posesión exclusiva de su banda de frecuencias.





Multicanalización por división de frecuencia

- Características del FDM
  - El ancho de banda del medio debe ser mayor que le ancho de banda de la señal transmitida.
  - Capacidad de transmisión de varias señales a la vez.
  - La señal lógica trasmitida a través del medio es analógica.
  - La señal recibida puede ser analógica o digital.
  - Para la comunicación análoga el ruido tiene menos efecto.

Multicanalización por división de frecuencia

VENTAJAS DE FDM

- El sistema de FDM apoya el flujo de dúplex total de información que es requerido por la mayor parte de la aplicación.
- El problema del ruido para la comunicación análoga tiene menos el efecto.
- Aquí el usuario puede ser añadido al sistema por simplemente añadiendo otro par de modulador de transmisor y modulador receptor.

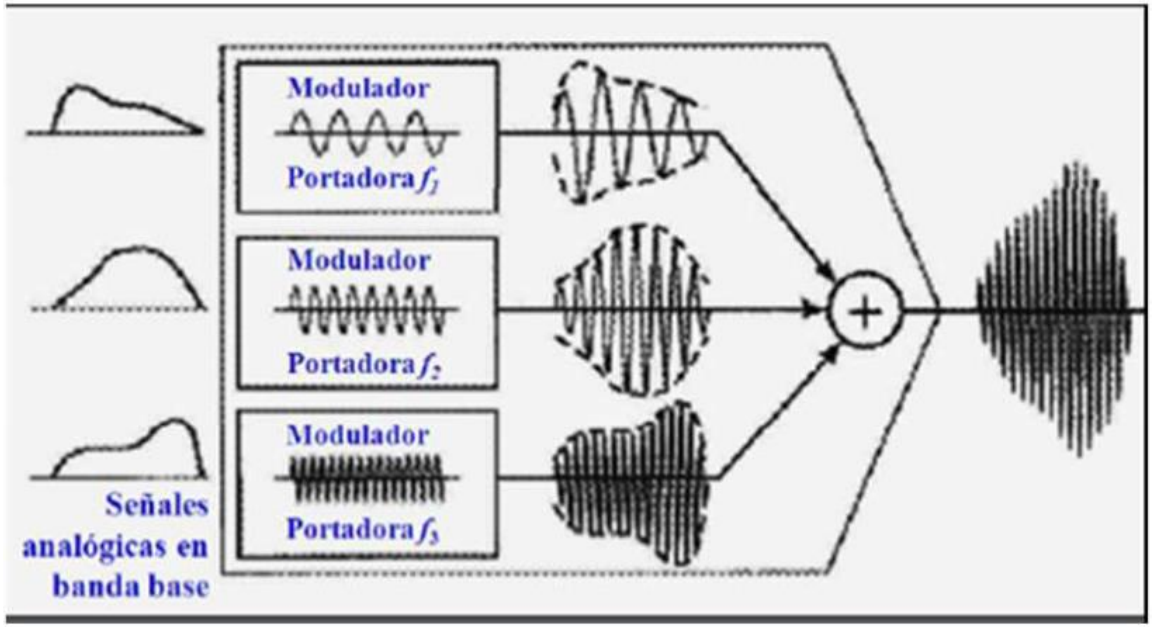
DESVENTAJAS DE FDM

- En el sistema FDM, el coste inicial es alto. Este puede incluir el cable entre los dos finales y los conectores asociados para el cable.
- En el sistema FDM, un problema para un usuario puede afectar a veces a otros.
- En el sistema FDM, cada usuario requiere una frecuencia de portador precisa.

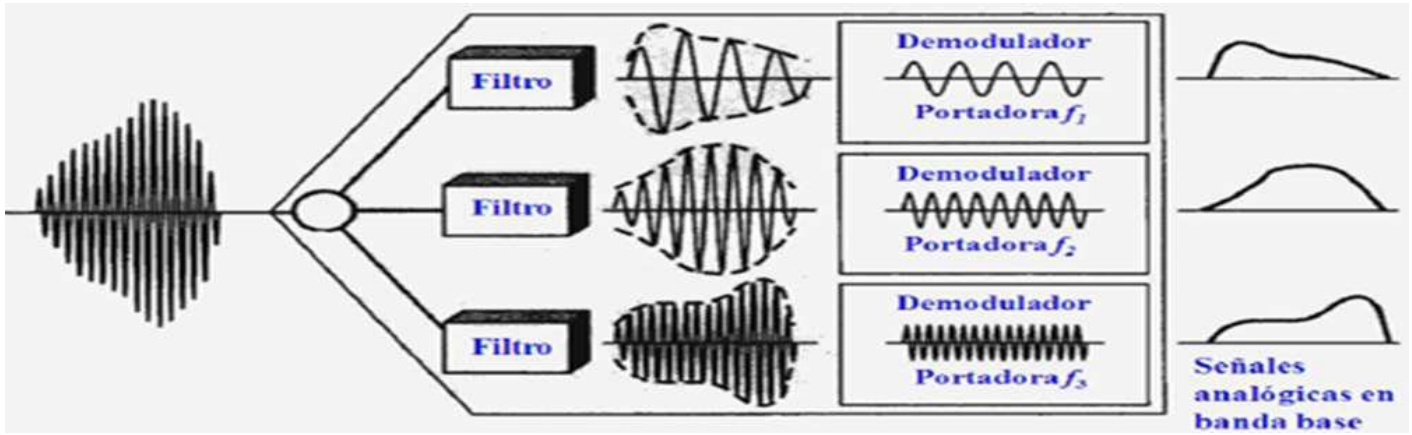
Multicanalización por división de frecuencia

Proceso de Multicanalización

- Cada fuente genera una señal con un rango de frecuencia similar.
  - Dentro del MUX, estas señales similares se modulan sobre distintas frecuencias portadoras (f1, f2, f3, etc.)
- Las señales moduladas se combinan en una única señal compuesta que se envía sobre un enlace.



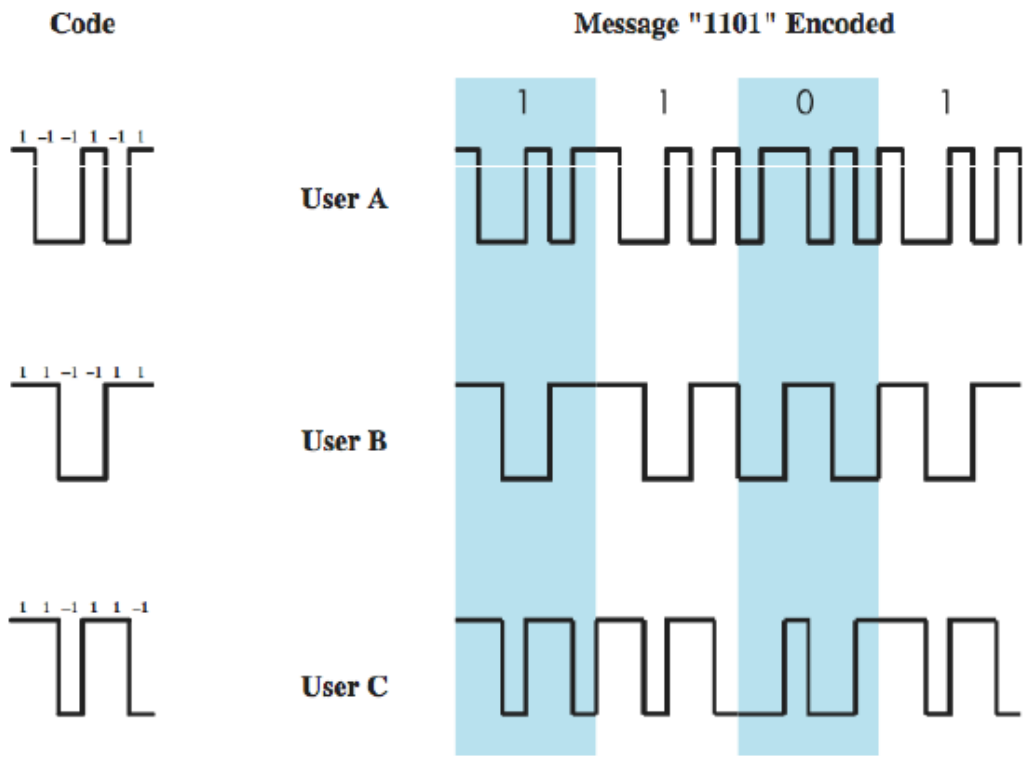
Multicanalización por división de frecuencia



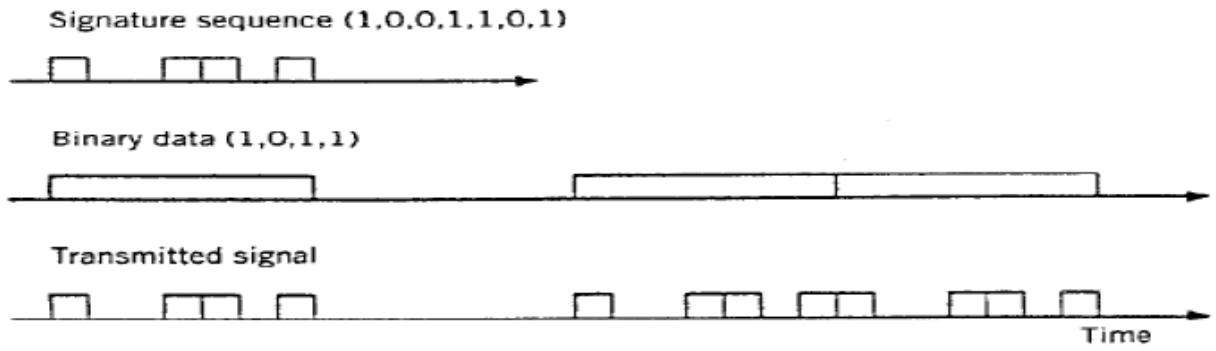
- El DEMUX usa filtros para descomponer la señal multiplexada en las señales que la constituyen.
- Las señales individuales se pasan después a un demodulador que las separa de sus portadoras y las pasa a la línea de salida

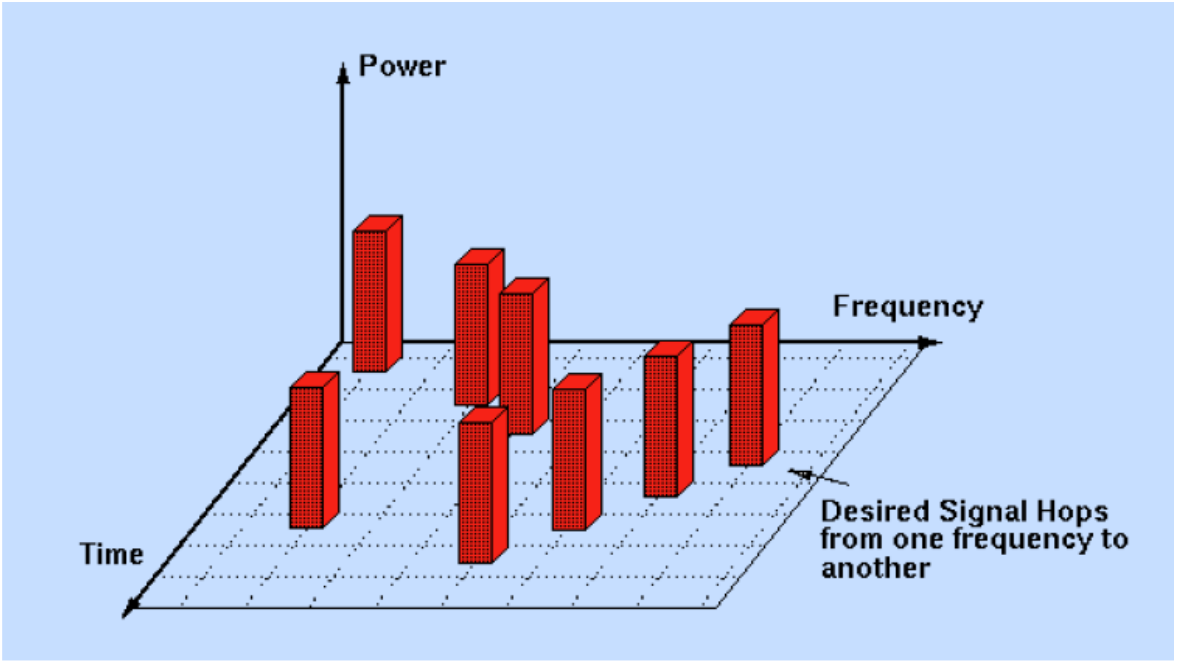
Multicanalización por división de código

- La **multiplexación por división de código**, **acceso múltiple por división de código** o **CDMA**
- Es un término genérico para varios métodos de multiplexación o control de acceso al medio basado en la tecnología de espectro expandido.
- CDMA emplea una tecnología de espectro expandido y un esquema especial de codificación, por el que a cada transmisor se le asigna un código único, escogido de forma que sea ortogonal respecto al del resto
- En CDMA, la señal se emite con un ancho de banda mucho mayor que el precisado por los datos a transmitir
    - La división por código es una técnica de acceso múltiple de espectro expandido.

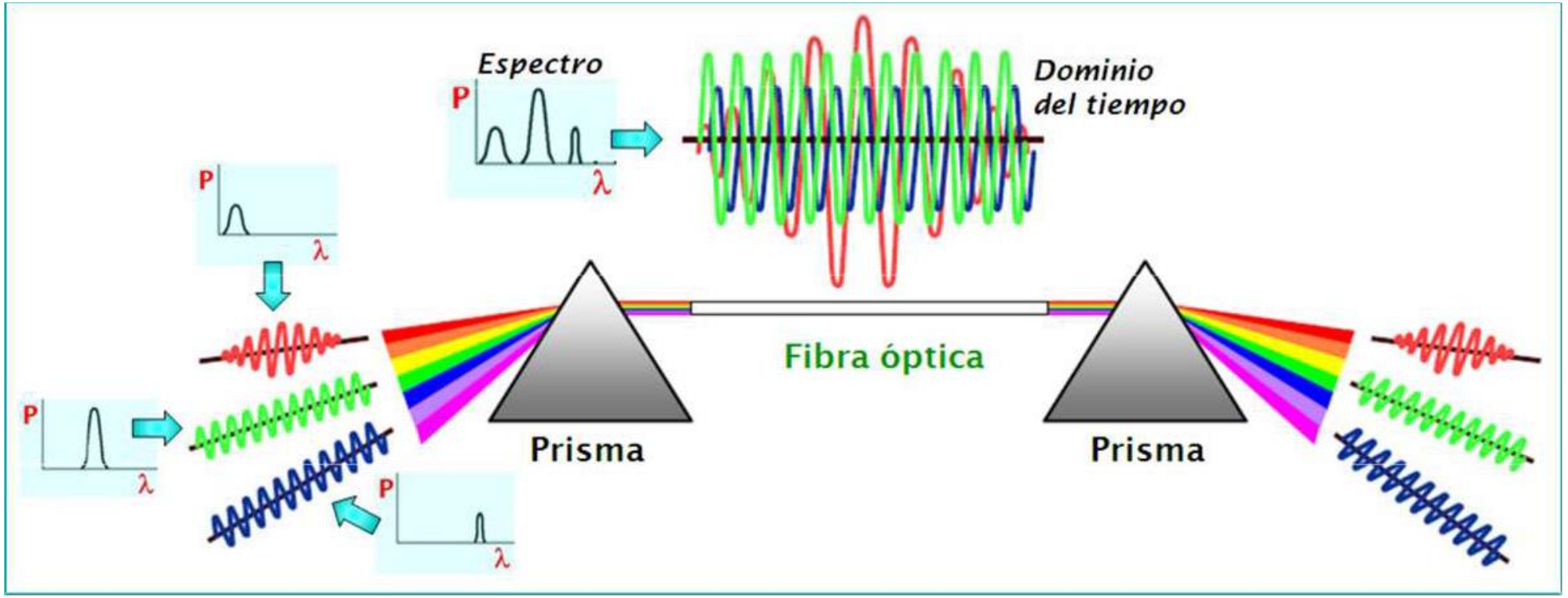


Multicanalización por división de código





- Tipos
  - Ensanchamiento espectral dado por la clave
    - Las claves las conocen el Rx y el Tx
  - Espectral
    - Esparcimiento espectral se hace con saltos en frecuencia (frecuency hopping).
    - Los saltos de frecuencia están dados por el código.

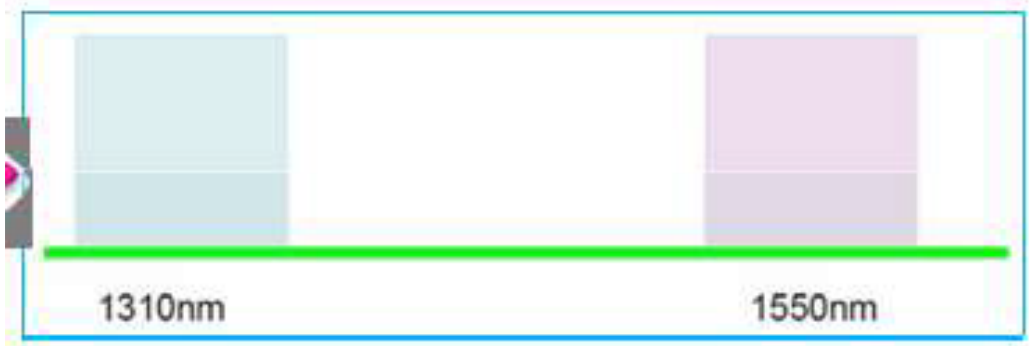


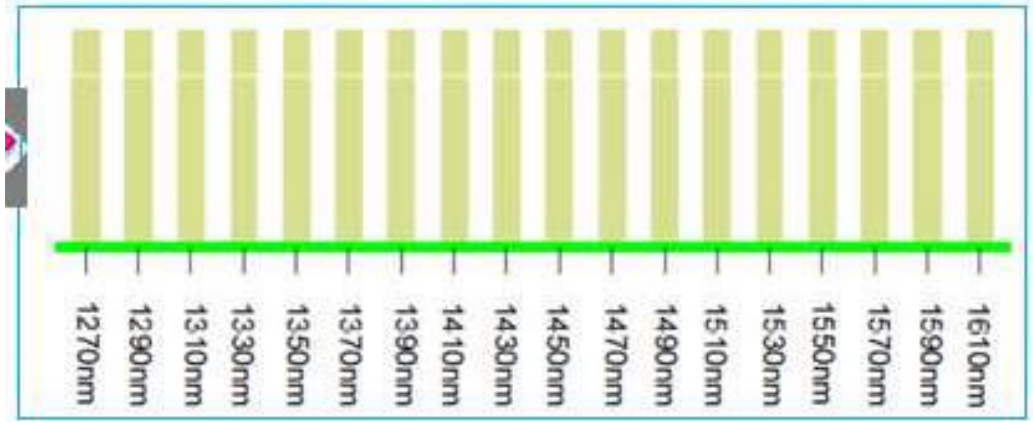
Multicanalización por división de onda

- Es una tecnología que multiplexa un número de señales portadoras ópticas en una sola fibra óptica mediante el uso de diferentes longitudes de onda de la luz láser.
- Esta técnica permite comunicaciones bidireccionales sobre una hebra de fibra, así como la multiplicación de la capacidad.
- Se diseñó para utilizar la capacidad de alta tasa de datos de la fibra.
- Conceptualmente es la misma que FDM, excepto que involucra señales luminosas de frecuencias muy altas.

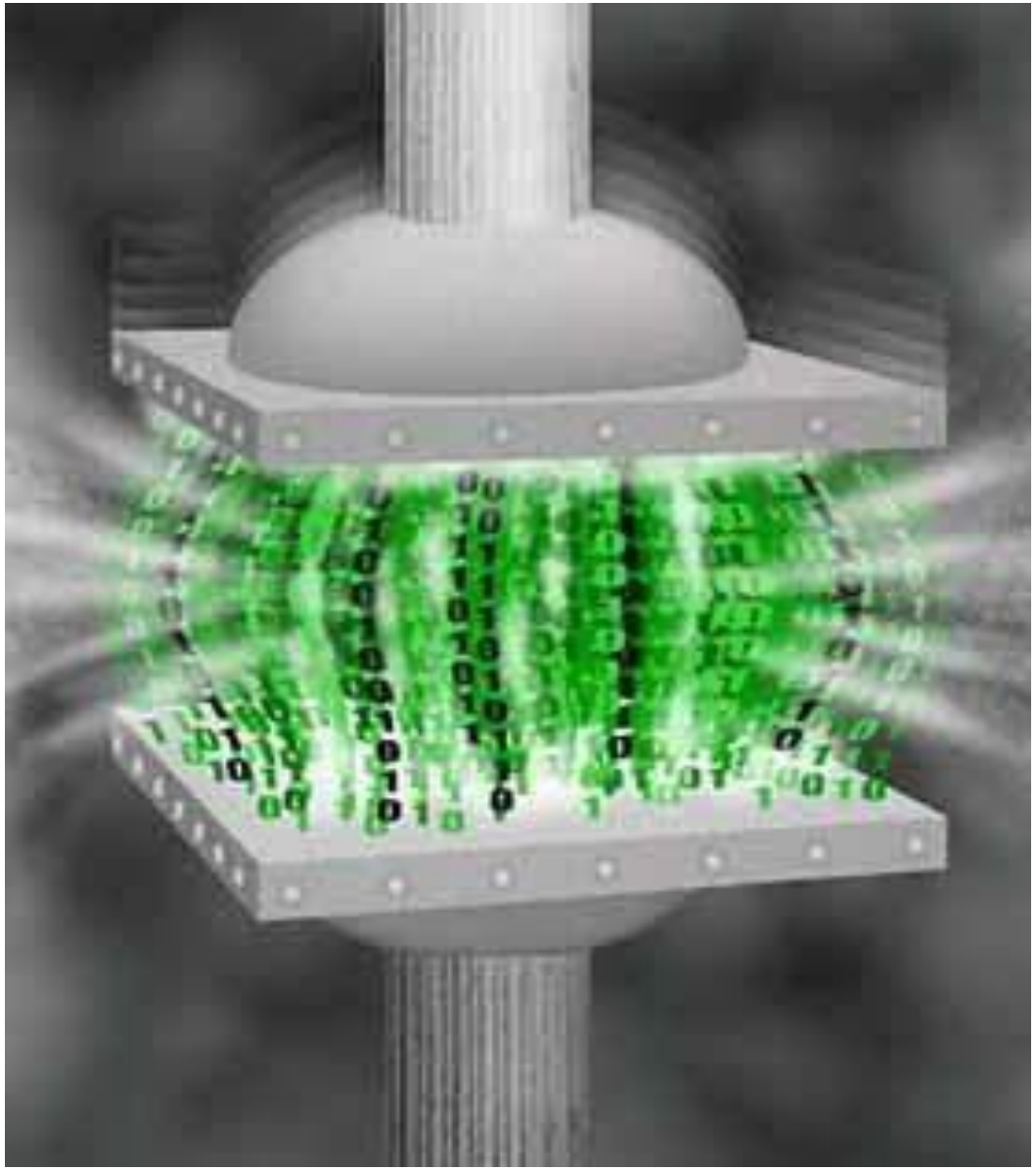
Multicanalización por división de onda

- Multiplexación por división de longitud de onda se aplica comúnmente a una portadora óptica
- Multiplexación por división de frecuencia típicamente se aplica a una portadora de radio
- Dado que la longitud de onda y la frecuencia están unidas entre sí a través de una simple relación directamente inversa, los dos términos en realidad describen el mismo concepto.
- Tipos de sistemas WDM**TOS DE TELECOMUNICACIONES**
  - Los primeros sistemas WDM usaron 2 longitudes de onda centradas en las ventanas de 1310 nm y 1550 nm.
  - Después fue CWDM (*Coarse WDM*) . La ITU (G.694.2) define una banda óptica de 18 l´s, entre 1270 y 1610 nm, espaciadas entre ellas 20 nm.





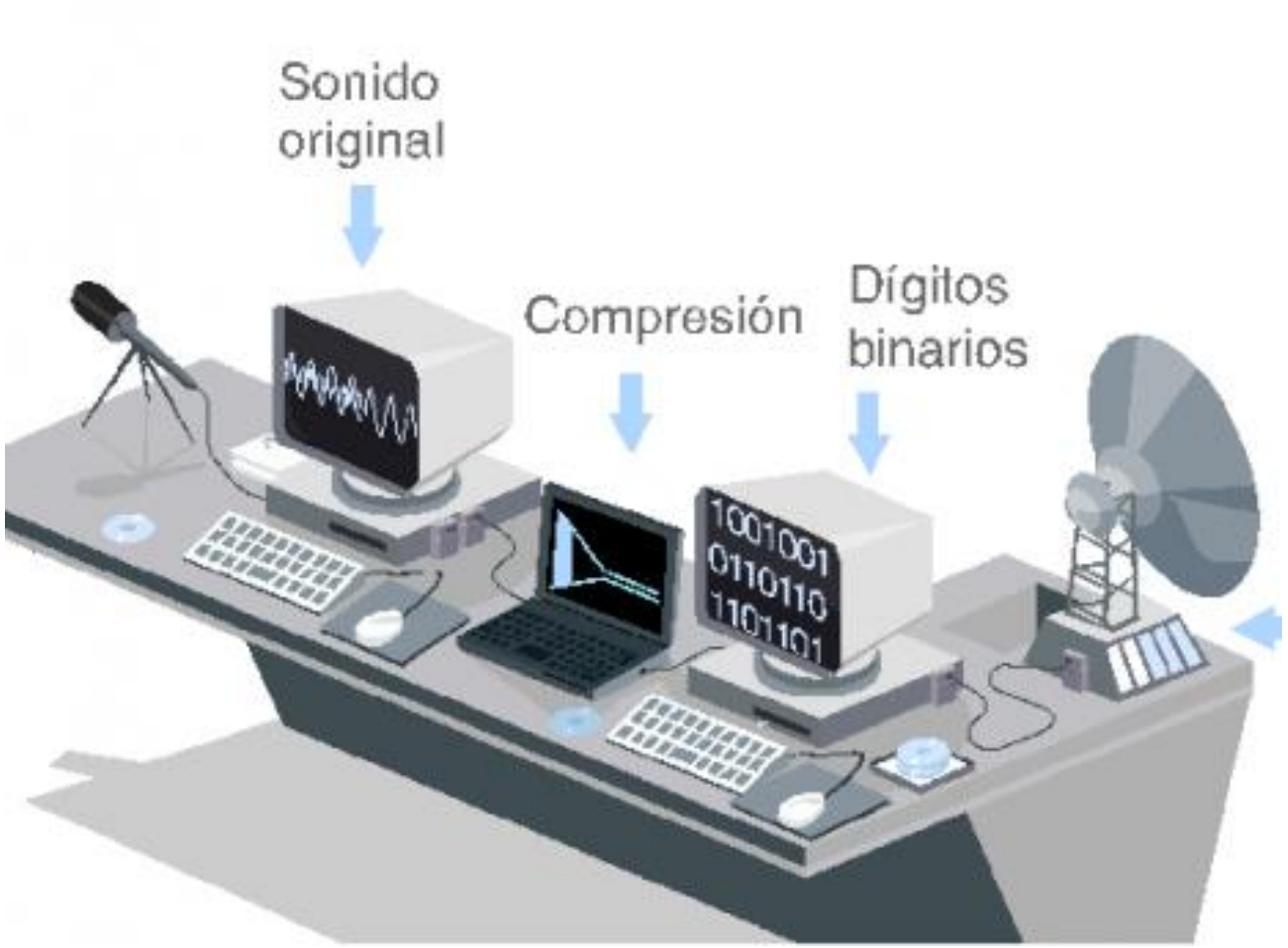
Compresión de datos



¿Qué es la compresión de datos?

- Se refiere al proceso de reducir la cantidad de datos necesarios para almacenar o transmitir información.
  - Consiste en codificar la información utilizando menos bits que la representación original.
  - Este proceso es similar a reducir un documento físico a un tamaño más pequeño y manejable sin perder su legibilidad.

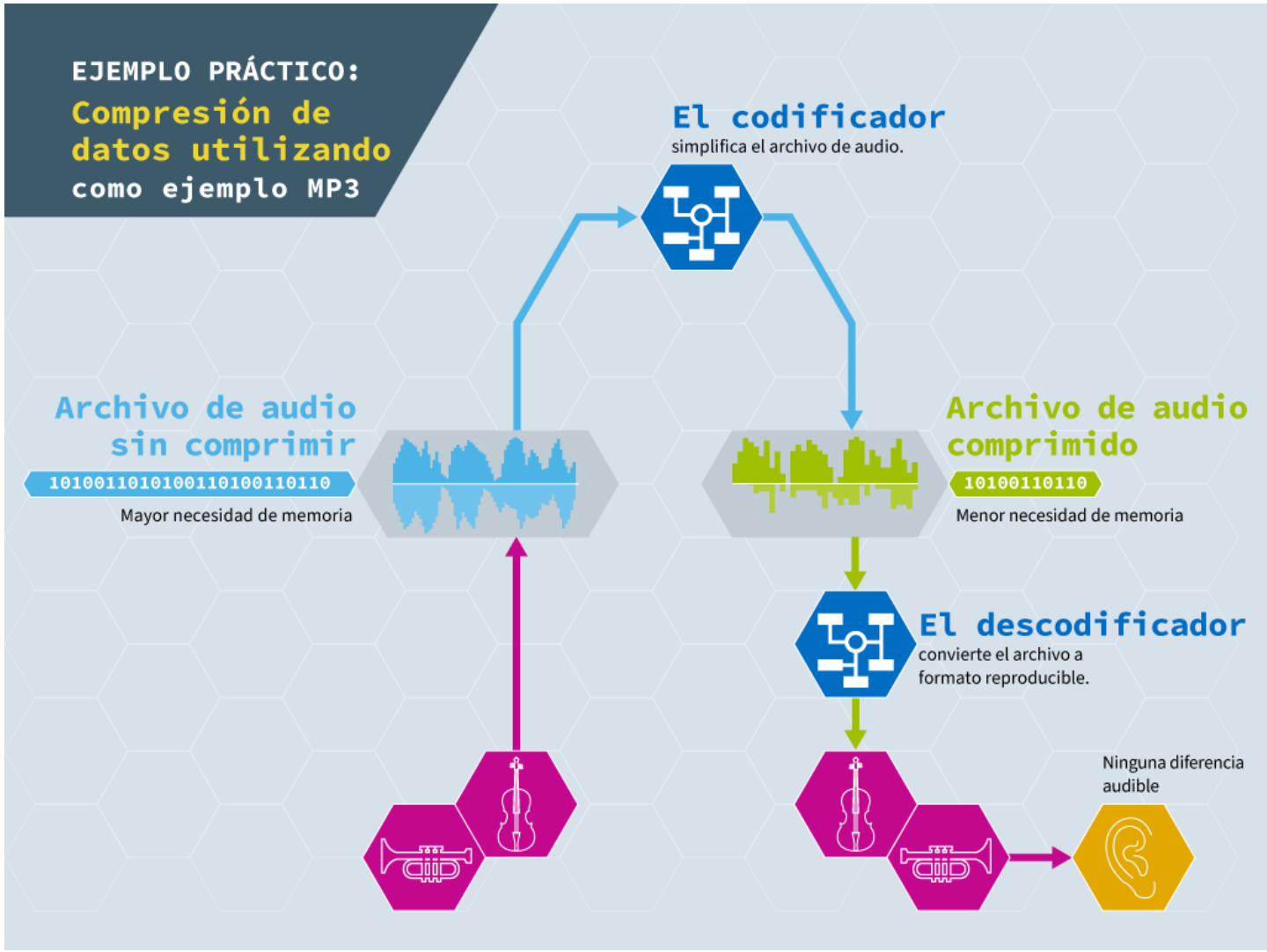
Importancia



- Los datos se han vuelto tan valiosos como cualquier otra mercancía
- La compresión de datos es clave en la era digital actual
  - Permite almacenar y transmitir datos de forma eficaz
  - Reduce
    - El espacio necesario
    - El tiempo de transferencia de datos.
  - Desempeña un papel fundamental en la gestión y utilización eficaz de este recurso.

Funcionamiento

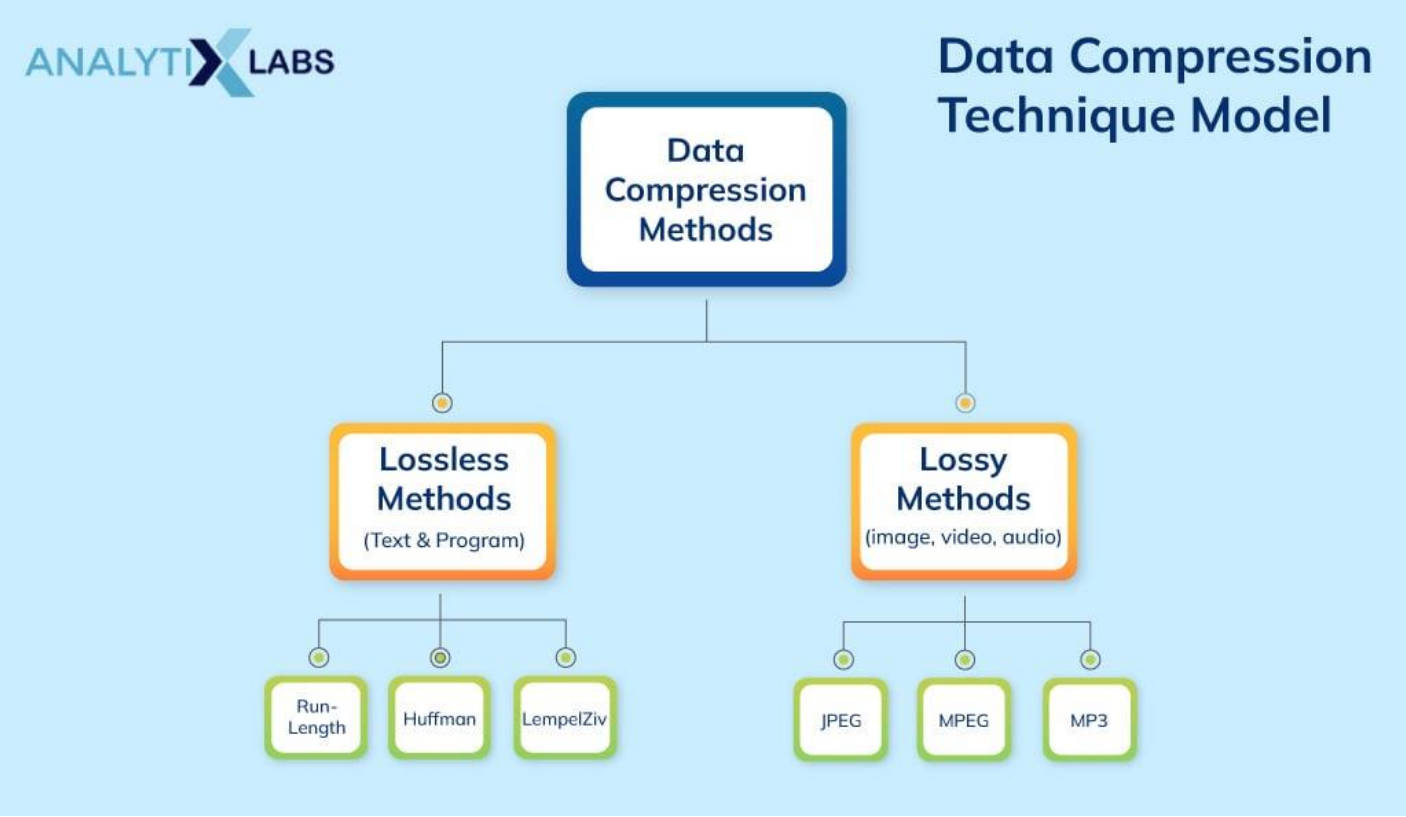
- La compresión de datos funciona identificando y eliminando la redundancia estadística.
  - Siempre hay grado de redundancia, independientemente del tipo de datos que se procesen:
    - Texto
    - Imágenes
    - Vídeo
    - Sonido
- Eliminar redundancia da como resultado datos comprimidos que ocupan menos espacio.



Utilización

- La compresión de datos tiene aplicación en multitud de ámbitos:
  - **Redes informáticas**: reduce la cantidad de datos transmitidos a través de las redes, aumentando así la velocidad de transmisión e incluso el ancho de banda.
  - **Almacenamiento**: al comprimir los archivos, se pueden almacenar más datos en el mismo espacio, lo cual es especialmente útil en dispositivos con capacidad de almacenamiento limitada.
  - **Servicios de streaming**: las plataformas de streaming utilizan la compresión de datos para ofrecer contenidos de forma más rápida y fluida, mejorando la experiencia del usuario.
  - **Copia de seguridad y archivo**: los datos comprimidos ocupan menos espacio, por lo que son ideales para copias de seguridad y archivos.

Compresión con o sin pérdida de datos

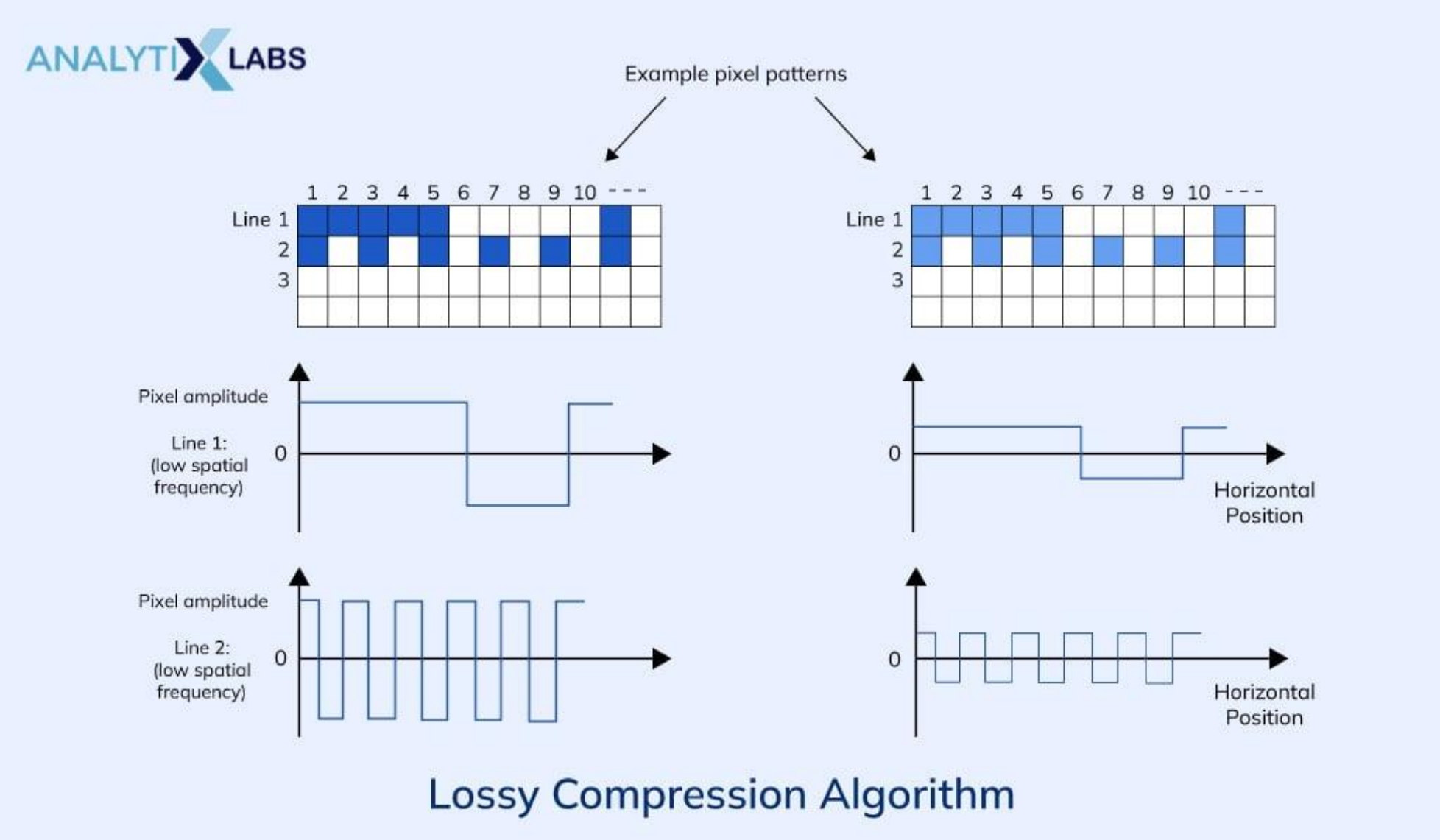


- Existen dos métodos principales de compresión de datos:
  - Con pérdida.
    - Reduce el tamaño del archivo eliminando la información innecesaria o menos importante.
    - Se suele utilizar en ámbitos en los que es aceptable una pérdida de calidad, como los archivos de audio y vídeo.
  - Sin pérdida
    - Método que reduce el tamaño del archivo sin pérdida de calidad.
    - Este tipo de compresión es ideal para aplicaciones en las que los datos originales deben conservarse perfectamente, como los archivos de texto.

- Diferencia entre datos e información.
  - Los datos son una colección de hechos o valores en bruto, a menudo desorganizados, y pueden significar números, texto, símbolos, etc.
  - Por otro lado, la información aporta contexto al organizar cuidadosamente los hechos.
    - Para poner esto en contexto, una imagen en blanco y negro de 4×6 pulgadas en 100 dpi (puntos por pulgada) tendrá 240.000 píxeles.
    - Cada uno de estos píxeles contiene datos en forma de un número entre 0 y 255, que representa la densidad de píxeles (0 es negro y 255 es blanco).
    - Esta imagen en su conjunto puede tener cierta información, como si fuera una foto del decimosexto presidente de los EE. UU., Abraham Lincoln.
    - Si mostramos una imagen en 50 dpi, es decir, en 60.000 píxeles, los datos necesarios para guardar la imagen se reducirán, y quizás también la calidad, pero la información permanecerá intacta.
    - Solo después de una pérdida considerable de datos, podemos perder la información.

Algoritmos con pérdida

Algoritmo con pérdida

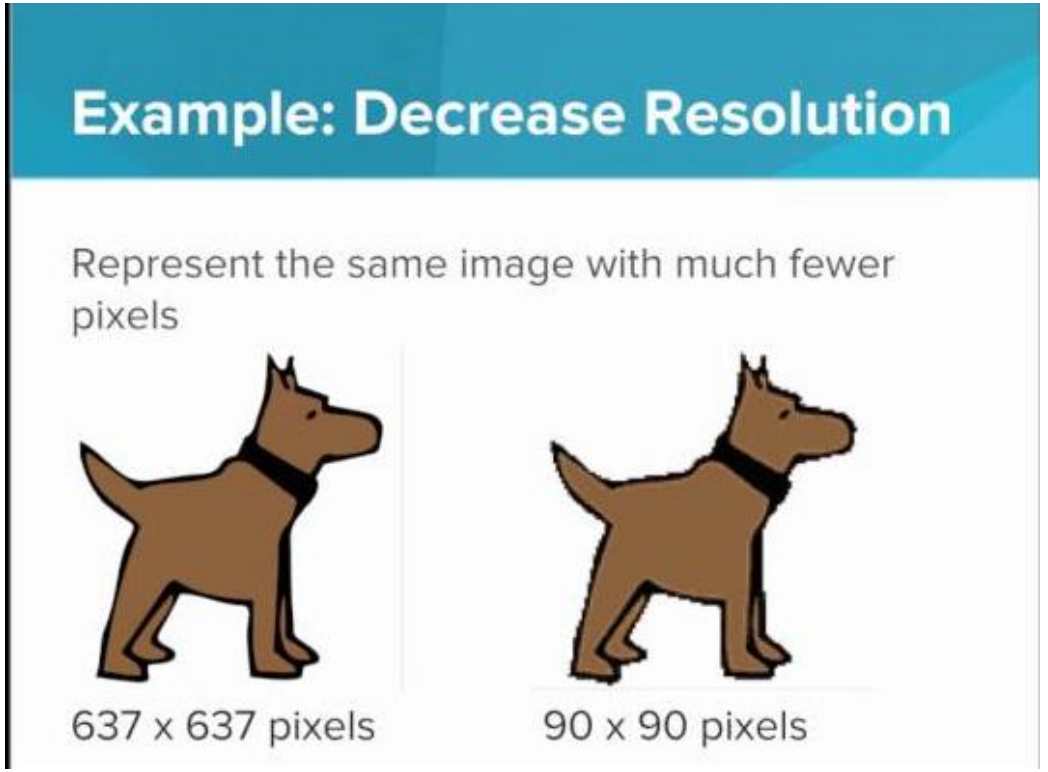


Algoritmo con pérdida

- Ventajas de la compresión con pérdida
  - Es relativamente rápida
  - Puede reducir drásticamente el tamaño del archivo
  - El usuario puede seleccionar el nivel de compresión.
  - Es beneficiosa para comprimir datos como:
    - Imágenes
    - Videos
    - Audio
    - Esto se debe a la limitación de nuestros ojos y oídos, ya que no pueden percibir una diferencia en la calidad de una imagen y un audio antes de cierto punto.

Algoritmos con pérdida

- Desventajas de la compresión con pérdida
  - No devolverá los mismos datos (en términos de calidad, tamaño, etc.).
  - Aun así, contendrá información similar (esto, de hecho, es útil en algunos casos, como la transmisión o la descarga de contenido de Internet).
  - La descarga y carga constante de un archivo puede comprimirlo y, en consecuencia, distorsionarlo más allá del punto de reconocimiento
    - Pérdida permanente de información.
  - Si el usuario utiliza un nivel de compresión severo, es posible que el archivo de salida no se parezca en nada al archivo de entrada original.



Algoritmos con pérdida

- Modelos de técnicas de compresión con pérdida:
  - Los modelos más comunes basados en la técnica con pérdida son:
    - Codificación por transformada
    - Transformada discreta del coseno
    - Transformada discreta de wavelet
    - Compresión fractal

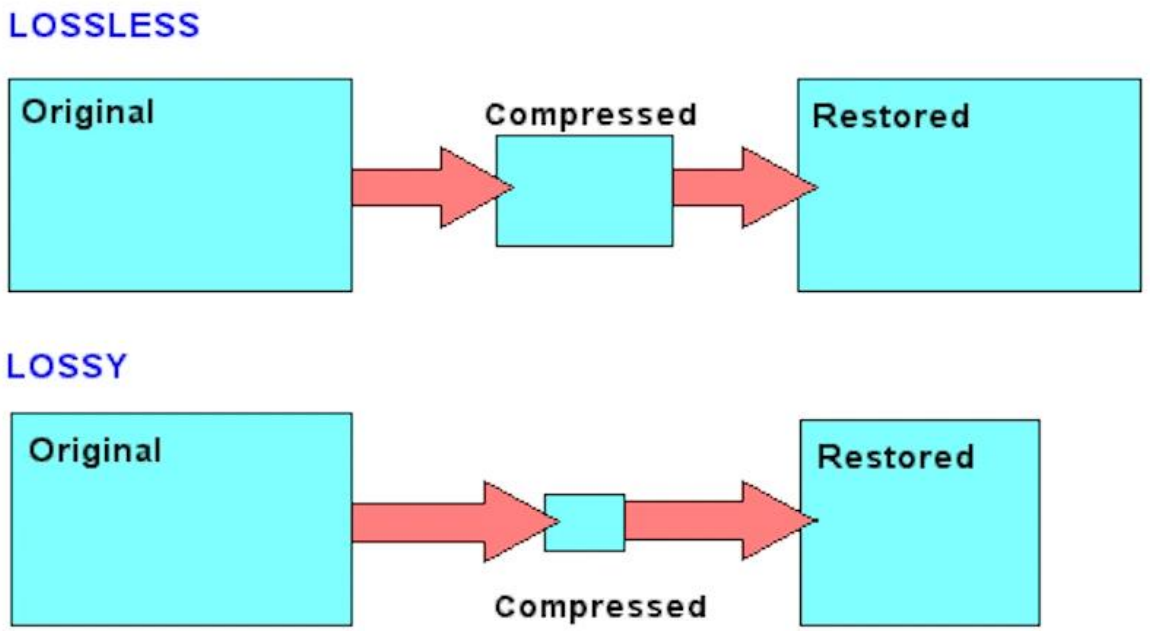
Algoritmo con pérdida

- La codificación por transformación es un tipo de compresión de datos para datos "naturales" como señales de audio o imágenes fotográficas.
- Normalmente, la transformación conlleva pérdida de información, resultando una copia de menor calidad que la entrada original.
- En la codificación por transformación, el conocimiento de la aplicación se utiliza para elegir la información a descartar para, de esa forma, disminuir su ancho de banda.
- La información restante se puede comprimir mediante varios métodos.
- Cuando se descodifica la salida, el resultado puede no ser idéntico a la entrada original, pero se espera que sea lo suficientemente parecido para los propósitos de la aplicación.

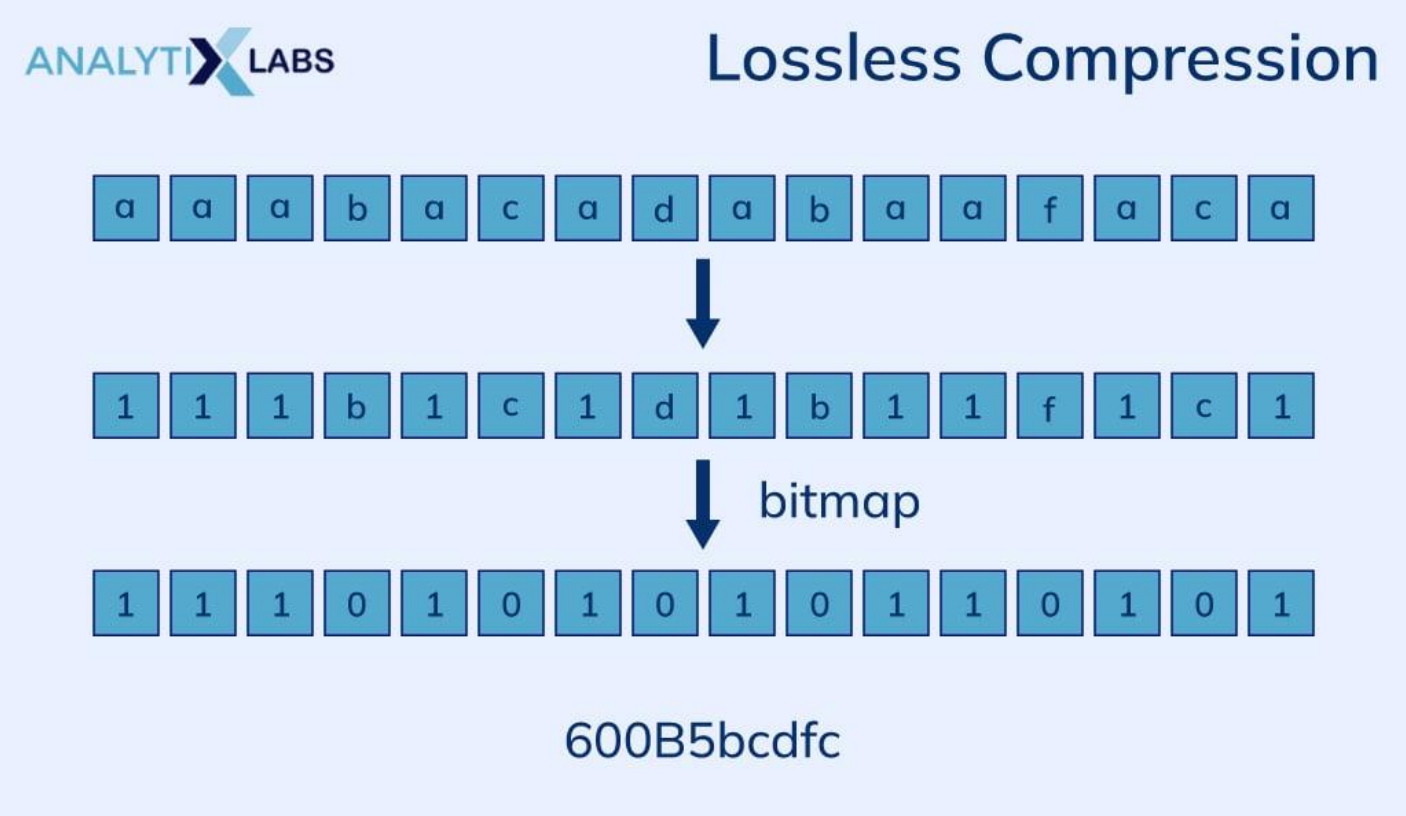
Algoritmos con pérdida

- Compresión de vídeo con pérdida
  - Flash (también admite sprites JPEG)
  - H.261
  - H.263
  - H.264/MPEG-4 AVC
  - MNG (admite sprites JPEG)
  - Motion JPEG
  - MPEG-1
  - MPEG-2
  - MPEG-4
  - OGG
  - Theora
  - códec para video Sorenson
  - VC-1
  - MP4
- Compresión de imagen con pérdida
  - Compresión fractal
  - JPEG
  - Compresión Wavelet
- Compresión de audio con pérdida
  - AAC
  - ADPCM
  - ATRAC
  - Dolby AC-3
  - DTS
  - MP2
  - MP3
  - Musepack
  - OGG
  - Vorbis
  - WMA
  - Opus (CELT)

Algoritmo sin pérdida



- La compresión sin pérdida no elimina ningún dato
  - Lo transforma para reducir su tamaño.
  - Entendiendo el concepto
    - Hay un fragmento de texto en el que la palabra "porque" se repite con bastante frecuencia.
      - El término está compuesto por siete letras
      - Utilizar una versión abreviada del mismo como "bcz", se puede transformar el texto.
      - Esta información de reemplazar "porque" por "bcz" se puede almacenar en un diccionario para su uso posterior (durante la descompresión).

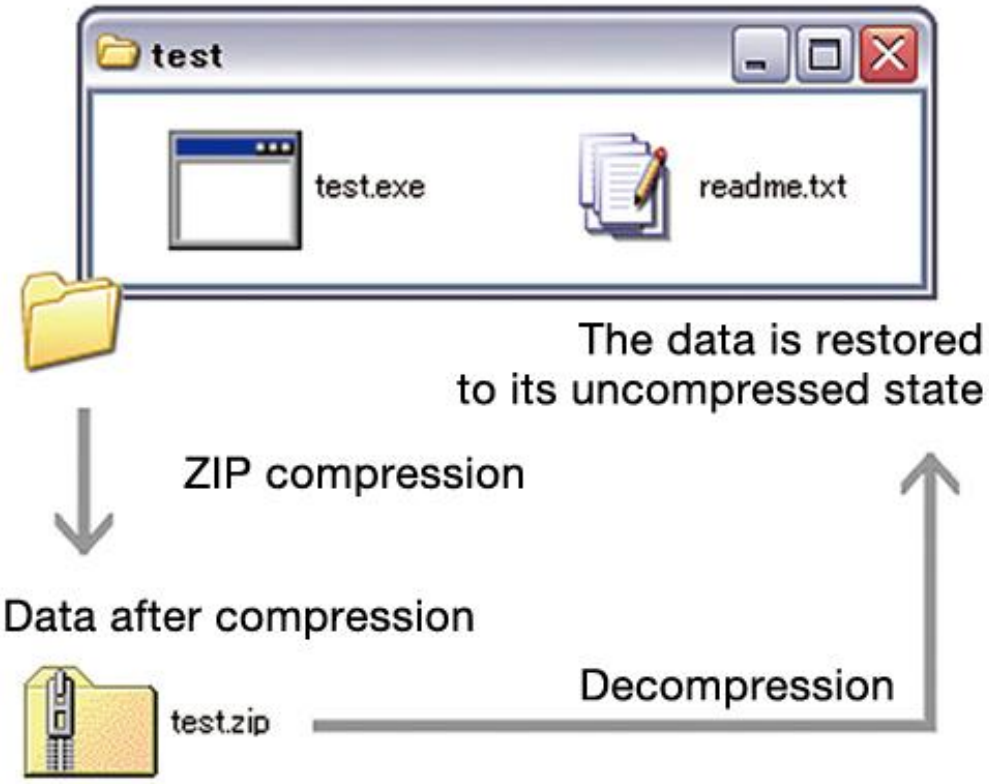


Algoritmo sin pérdida

- Metodología:
  - La compresión con pérdida elimina fragmentos de datos redundantes o imperceptibles para reducir el tamaño
  - La compresión sin pérdida los transforma codificándolos mediante alguna fórmula o lógica. Así es como funciona la compresión sin pérdida.

Algoritmo sin pérdida

- Ventajas:
  - Existen tipos de datos en los que la compresión con pérdida no es viable.
    - Una hoja de cálculo
    - Un software
    - Un programa o cualquier dato compuesto por texto o números
  - La compresión con pérdida no puede funcionar
    - Todos los números pueden ser esenciales y no pueden considerarse redundantes
    - Cualquier reducción provocará inmediatamente la pérdida de información
  - La compresión sin pérdida se vuelve crucial,
    - Tras la descompresión, el archivo se puede restaurar a su estado original sin perder ningún dato.

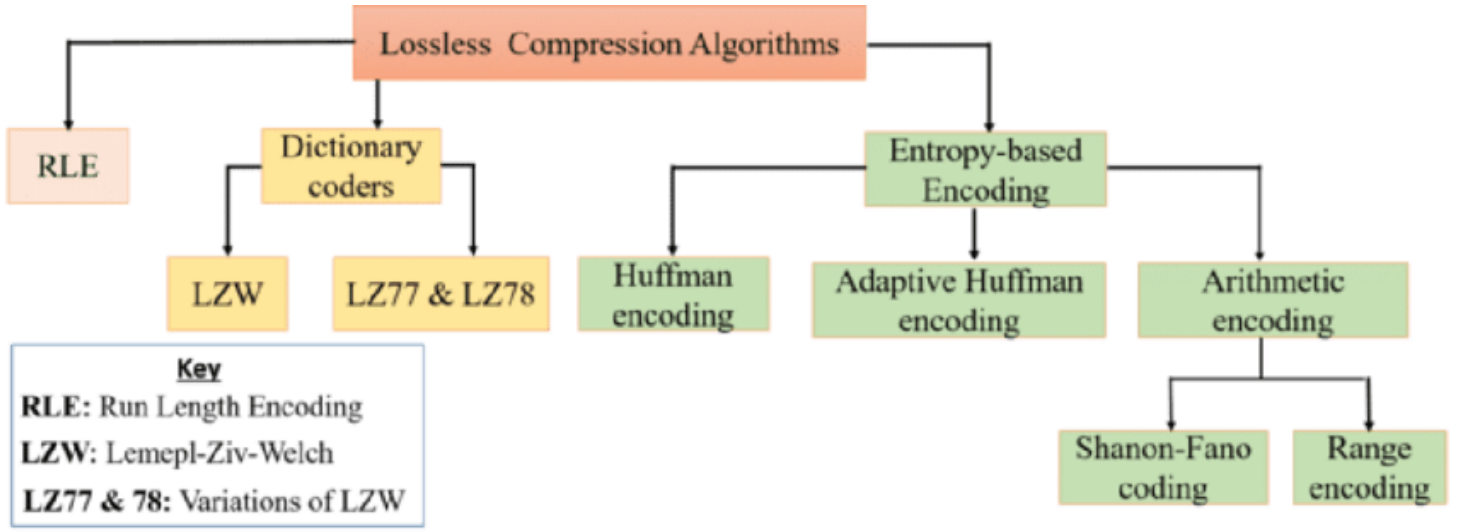


Algoritmos sin pérdida

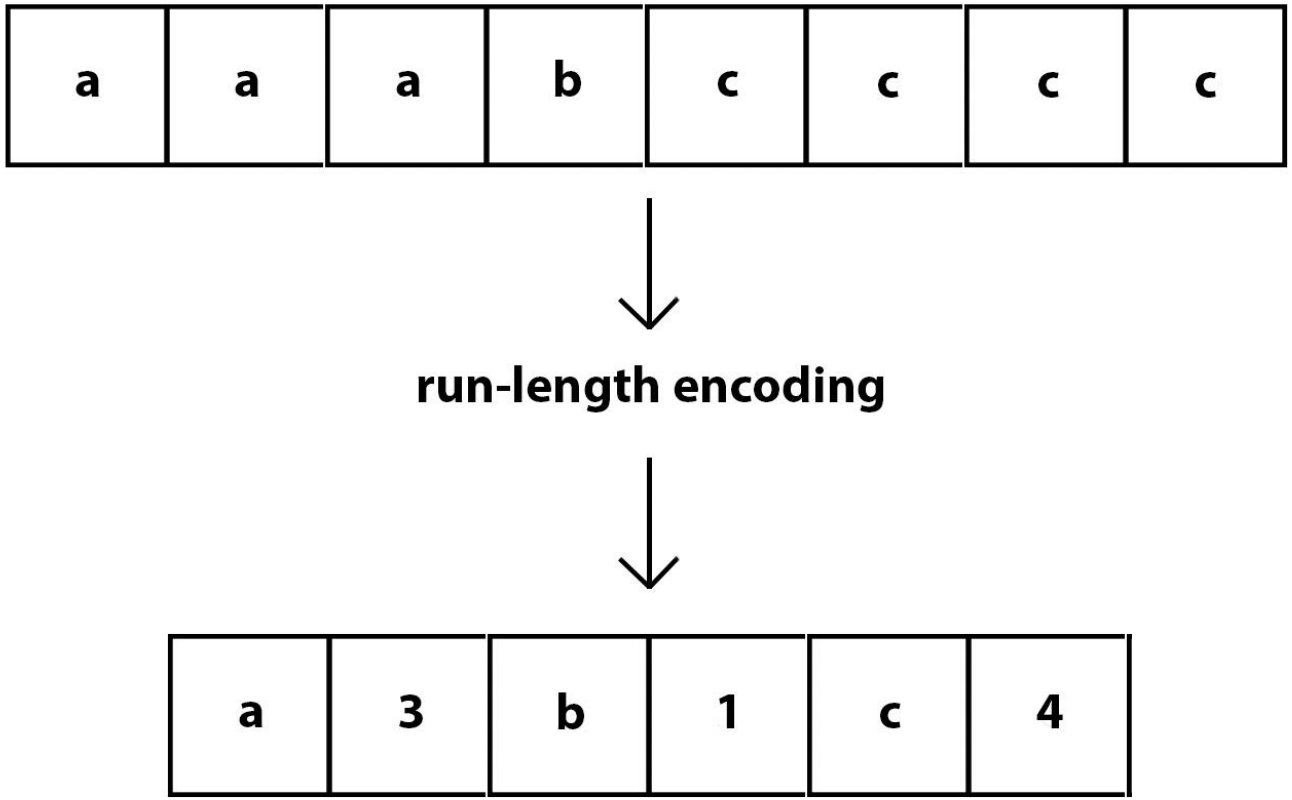
- Desventaja:
  - La compresión de datos tiene un límite.
  - Si los datos ya están comprimidos, volver a comprimirlos reducirá poco o nada su tamaño.
  - Es menos eficaz con archivos de mayor tamaño.

Algoritmos sin pérdida

- Modelos de técnicas de compresión sin pérdida
  - Los modelos más comunes basados en la técnica sin pérdida son:
    - RLE (codificación de longitud de ejecución)
    - Codificador de diccionario (LZ77, LZ78, LZR, LZW, LZSS, LZMA, LZMA2)
    - Predicción por coincidencia parcial (PPM)
    - Deflate Mezcla de contenido
    - Codificación Huffman
    - Codificación Huffman adaptativa
    - Codificación Shannon Fano
    - Codificación aritmética
    - Codificación Lempel Ziv Welch
    - Zstandard
    - Bzip2 (Burrows y Wheeler)



Algoritmo sin pérdida

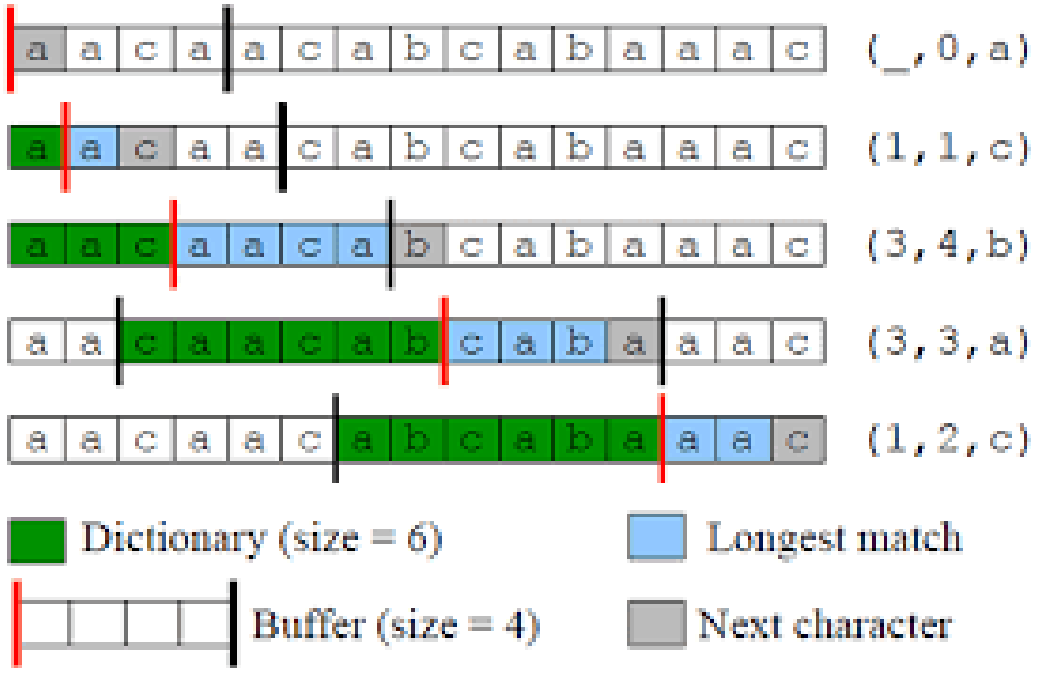


- RLE (codificación de longitud de ejecución)
  - El algoritmo Run-Length Encoding, o RLE, comprime datos que contienen una gran cantidad de repeticiones consecutivas de un mismo valor o símbolo.
  - En lugar de almacenar cada valor o símbolo por separado, la codificación RLE almacena el valor o símbolo repetido junto con la longitud de la repetición.

python

```python
def rle_encode(data: str) -> str:
    # Función auxiliar recursiva que toma la cadena de entrada y un índice
    def rle_helper(data: str, i: int) -> str:
        # Verificar si ya se ha llegado al final de la cadena de entrada
        if i >= len(data):
            return ""
        # Contar la cantidad de repeticiones consecutivas del carácter en la posición actual i
        def count_sequence(data, i):
            if i + 1 >= len(data) or data[i] != data[i + 1]:
                return 1
            count = count_sequence(data, i + 1)
            return count + 1
        count = count_sequence(data, i)
        # Si la cantidad de repeticiones es 1, simplemente agregar el carácter a la cadena comprimida
        if count == 1:
            return data[i] + rle_helper(data, i + 1)
        # Si la cantidad de repeticiones es mayor que 1, agregar la cantidad y el carácter a la cadena comprimida
        else:
```

Algoritmos sin pérdida



- LZ77
  - Es un compresor basado en algoritmo sin pérdida
  - Es un tipo de codificador diccionario en el cual existen los literales, banderas y palabras claves
  - Se recorre la cadena
    - Si se encuentra con un literal lo deja totalmente igual,
    - Si encuentra una bandera especifica si lo que sigue es
      - Un literal
      - Un comprimido (que es una especie de palabra clave)
        - Se lleva a una posición en un diccionario que arroja que bytes continúan.

Algoritmo sin pérdida

LZ77, ejemplo con abracadabrarray

1	2	3	4	5	6	7	8	1	2	3	4	5	6	Output
								a	b	r	a	c	a	<0,0,a>
							a	b	r	a	c	a	d	<0,0,b>
						a	b	r	a	c	a	d	a	<0,0,r>
					a	b	r	a	c	a	d	a	b	<3,1,c>
			a	b	r	a	c	a	d	a	b	r	a	<2,1,d>
	a	b	r	a	c	a	d	a	b	r	a	r	r	<7,4,r>
c	a	d	a	b	r	a	r	r	a	y				<3,2,y>

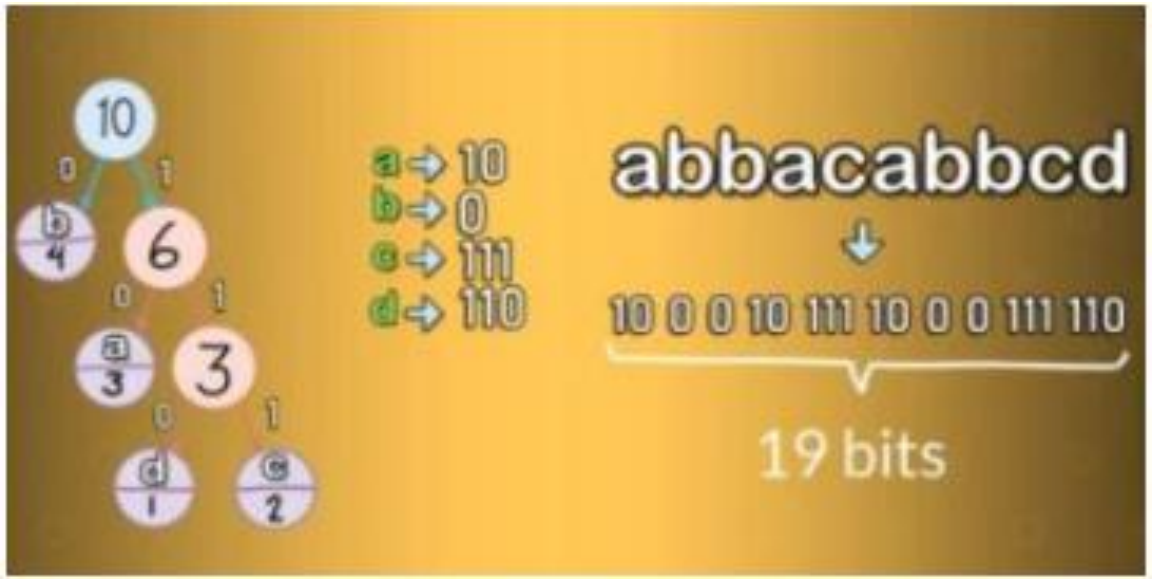
Search Buffer

Look ahead buffer Buffer

Algoritmo sin Pérdida

```
def compress(self, text: str) -> list[Token]:
    """
    Args:
      text: string to be compressed

    Returns:
      output: the compressed text as a list of Tokens
    """
    output = []
    search_buffer = ""
    # while there are still characters in text to compress
    while text:
      # find the next encoding phrase
      # - triplet with offset, length, indicator (the next encoding character)
      token = self._find_encoding_token(text, search_buffer)

      # update the search buffer:
      # - add new characters from text into it
      # - check if size exceed the max search buffer size, if so, drop the
      #   oldest elements
      search_buffer += text[: token.length + 1]
      if len(search_buffer) > self.search_buffer_size:
        search_buffer = search_buffer[-self.search_buffer_size :]
      # update the text
      text = text[token.length + 1 :]

      # append the token to output
      output.append(token)

    return output
```

Algoritmo sin pérdida



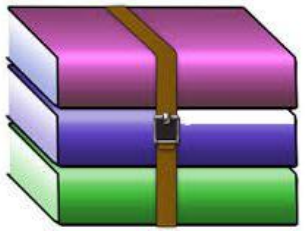
Ejemplo:
ilaaaseeaalllssallsas

- Codificación de Huffman
  - Es un procedimiento que permite asignar a los diferentes símbolos a comprimir, un código binario.
  - Este algoritmo crea un árbol de nodos
    - Primero ro se debe establecer un orden prioritario
      - El más importante es el símbolo que aparece con menor frecuencia en la cadena
    - Luego, se eliminan los dos símbolos más prioritarios
    - Construyendo así un nuevo "padre" que es el resultado de la suma de las frecuencias eliminadas
      - Se ubica nuevamente en la cola de prioridad
    - Iterativamente se realiza este proceso hasta que solo quede un elemento
      - Así queda construido el árbol.
  - Para asignar el código binario se contarán los pasos efectuados para llegar a cada símbolo del árbol
    - Movimiento a la izquierda=0, a la derecha=1
    - Obteniendo el valor de cada uno
    - Por último, reemplazándolos en la cadena.

```
def Huffman_Encoding(data):
    symbol_with_probs = Calculate_Probability(data)
    symbols = symbol_with_probs.keys()
    probabilities = symbol_with_probs.values()
    print("symbols: ", symbols)
    print("probabilities: ", probabilities)

    nodes = []

    # converting symbols and probabilities into huffman tree nodes
    for symbol in symbols:
        nodes.append(Node(symbol_with_probs.get(symbol), symbol))

    while len(nodes) > 1:
        # sort all the nodes in ascending order based on their probability
        nodes = sorted(nodes, key=lambda x: x.prob)
        # for node in nodes:
        #       print(node.symbol, node.prob)

        # pick 2 smallest nodes
        right = nodes[0]
        left = nodes[1]

        left.code = 0
        right.code = 1

        # combine the 2 smallest nodes to create new node
        newNode = Node(left.prob+right.prob, left.symbol+right.symbol, left, right)

        nodes.remove(left)
        nodes.remove(right)
        nodes.append(newNode)

    huffman_encoding = Calculate_Codes(nodes[0])
    print(huffman_encoding)
    Total_Gain(data, huffman_encoding)
    encoded_output = Output_Encoded(data,huffman_encoding)
    print("Encoded output:", encoded_output)
    return encoded_output, nodes[0]
```

ARCHIVOS ZIP O RAR

archivo tipo zip



archivo tipo rar

compresión sin perdida

Algoritmos sin pérdida

- Wavelets
- Zip
- rar
- CAB
- LHA
- DGCA
- GCA

Modelos basados en redes neuronales

- Algunos modelos basados en redes neuronales también se utilizan para la compresión, como:
  - Compresión basada en perceptrón multicapa (MLP) (utilizada para la compresión de imágenes)
  - Compresión basada en red neuronal convolucional (CNN) como Deep Coder (utilizada para la compresión de video)
  - Compresión basada en red generativa (GAN) (utilizada para la compresión en tiempo real)