

Universidad de Los Andes
Facultad de Ingeniería
Escuela de Sistemas

Redes de Computadoras Capa Física y de Enlace

Mérida - Venezuela
Prof. Gilberto Díaz

Conectando Físicamente los nodos

Conectar dos o más nodos de red a través de un enlace es sólo el primer paso. Existen otros factores que deben ser considerados para que un par de nodos puedan conectarse de forma satisfactoria

- Codificación (encoding): formato de los bits en el cable
- Framing: delinear una secuencia de bits para convertirlos en un mensaje
- Detección de errores (Error detection): algunos frames pueden corromperse y debe manejarse estas situaciones
- Transmisión confiable a pesar de los errores
- Control de Acceso al Medio: gestión de un enlace compartido

Bloques básicos de construcción de redes

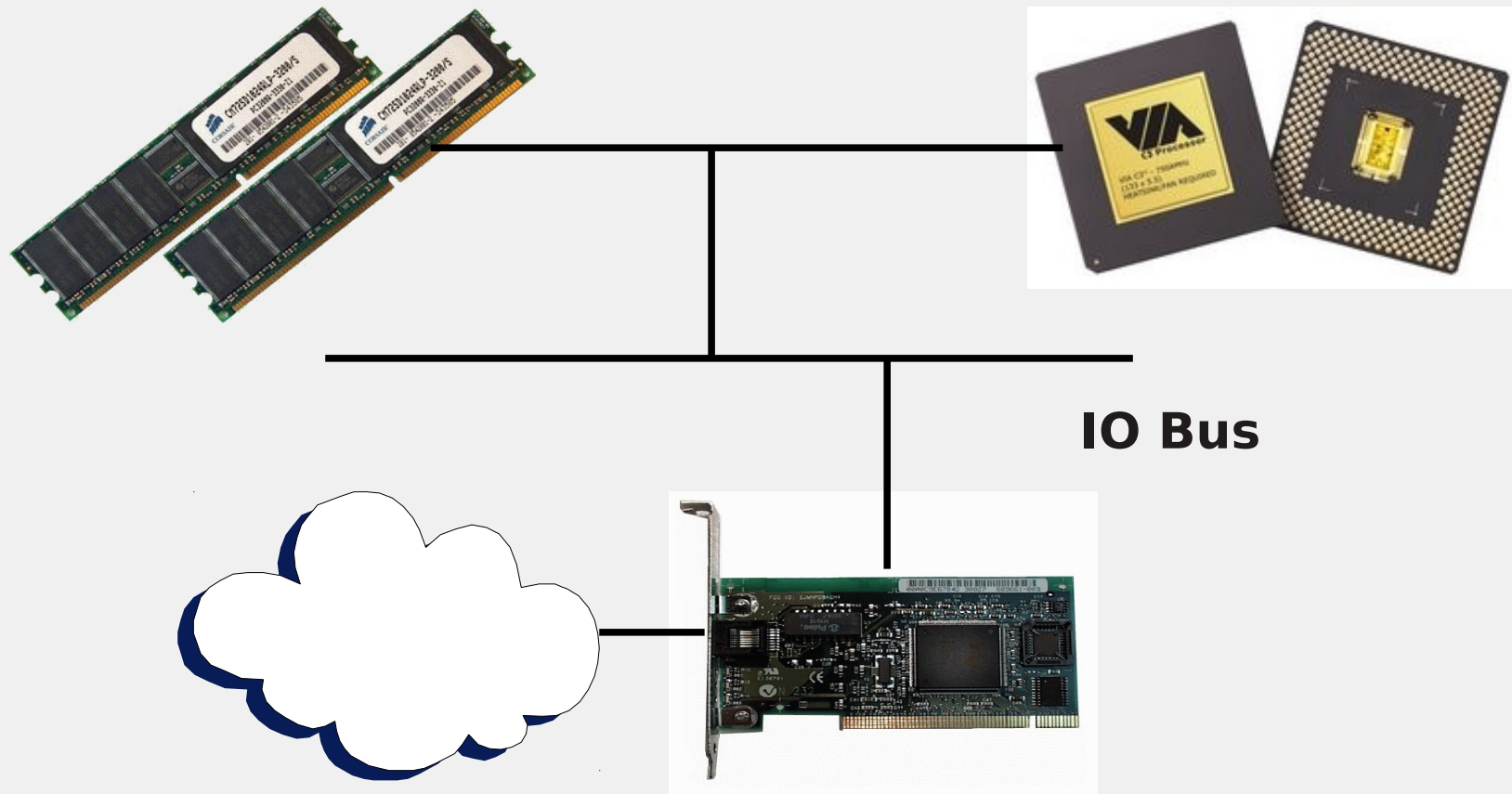
Hemos estado utilizando dos tipos de nodos para describir las redes sencillas que hemos visto

- Computadores
- Enlaces

Describamos un poco más cada uno de estos dos elementos

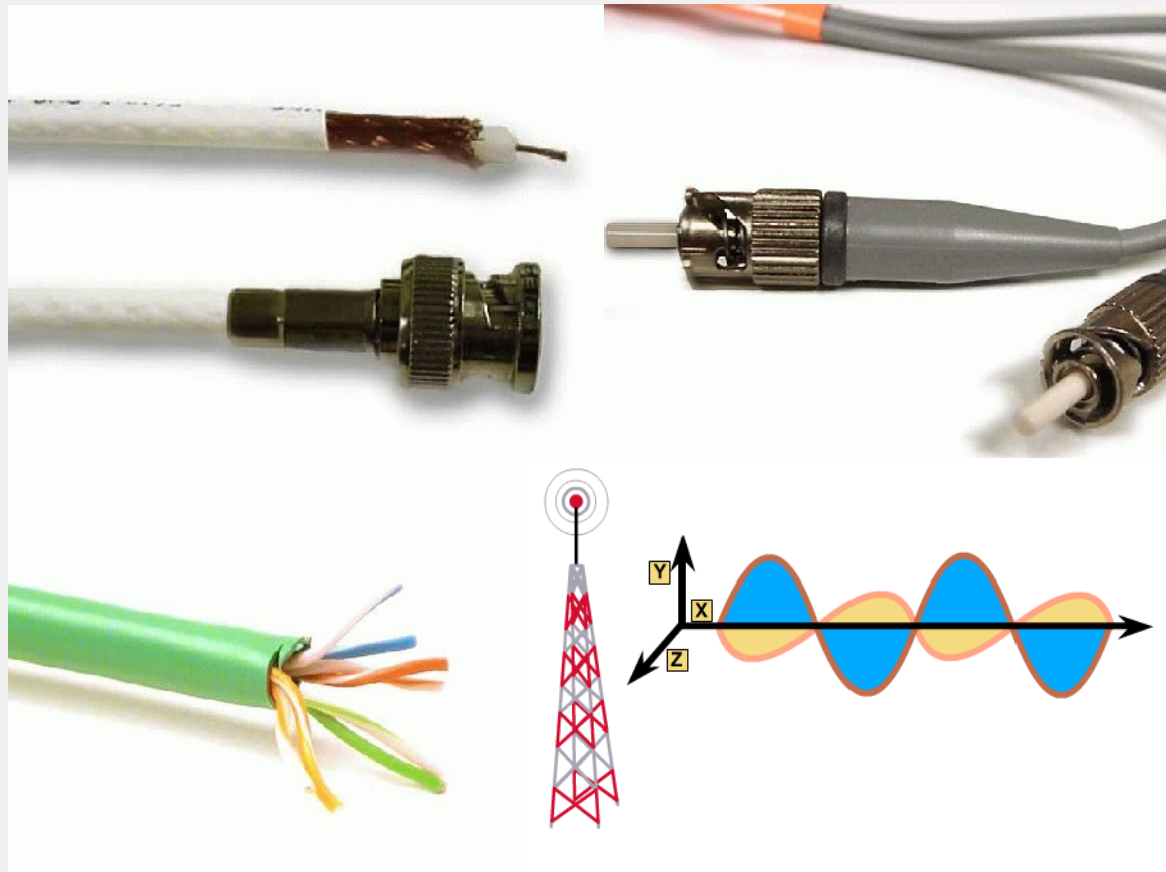
Computador

Los computadores que consideraremos son de índole general pudiendo fungir tanto de cliente como de servidor.



Enlaces

Los enlaces son implementados utilizando diferentes medios físicos

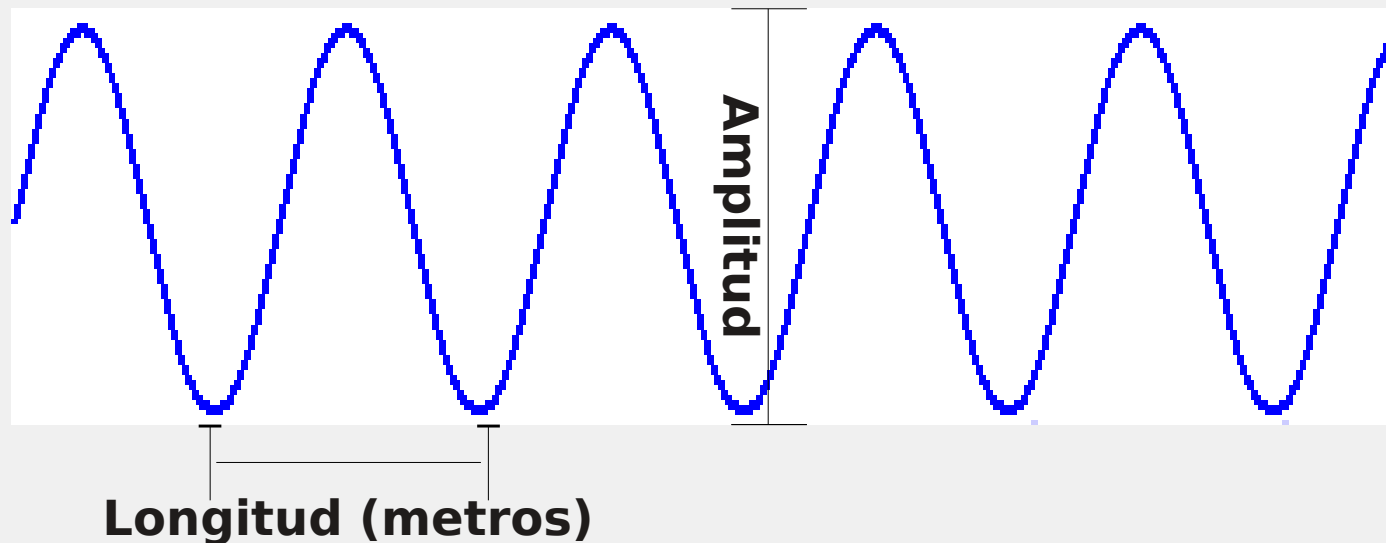


Enlaces

- Cualquiera sea el medio de transmisión, este se utiliza para transmitir señales.
- Estas señales son ondas electromagnéticas que viajan a la velocidad de la luz.
- Sin embargo, la velocidad de la luz es dependiente del medio: cobre, vidrio, etc.

Enlaces

- Una propiedad importante de una onda electromagnética es la frecuencia (medida en Hertz) a la cual la esta oscila.



Enlaces

- Estas ondas viajan a la velocidad de la luz, esa velocidad dividida por la frecuencia es igual a la longitud. Ejemplo: la voz viaja por el cobre de 300 Hz a 3300Hz, en el caso de 300Hz tenemos

$$\text{Long} = \text{VelocidaLuz} / \text{Frecuencia}$$

$$\text{Long} = 2/3 \times 3 \times 10^8 / 300$$

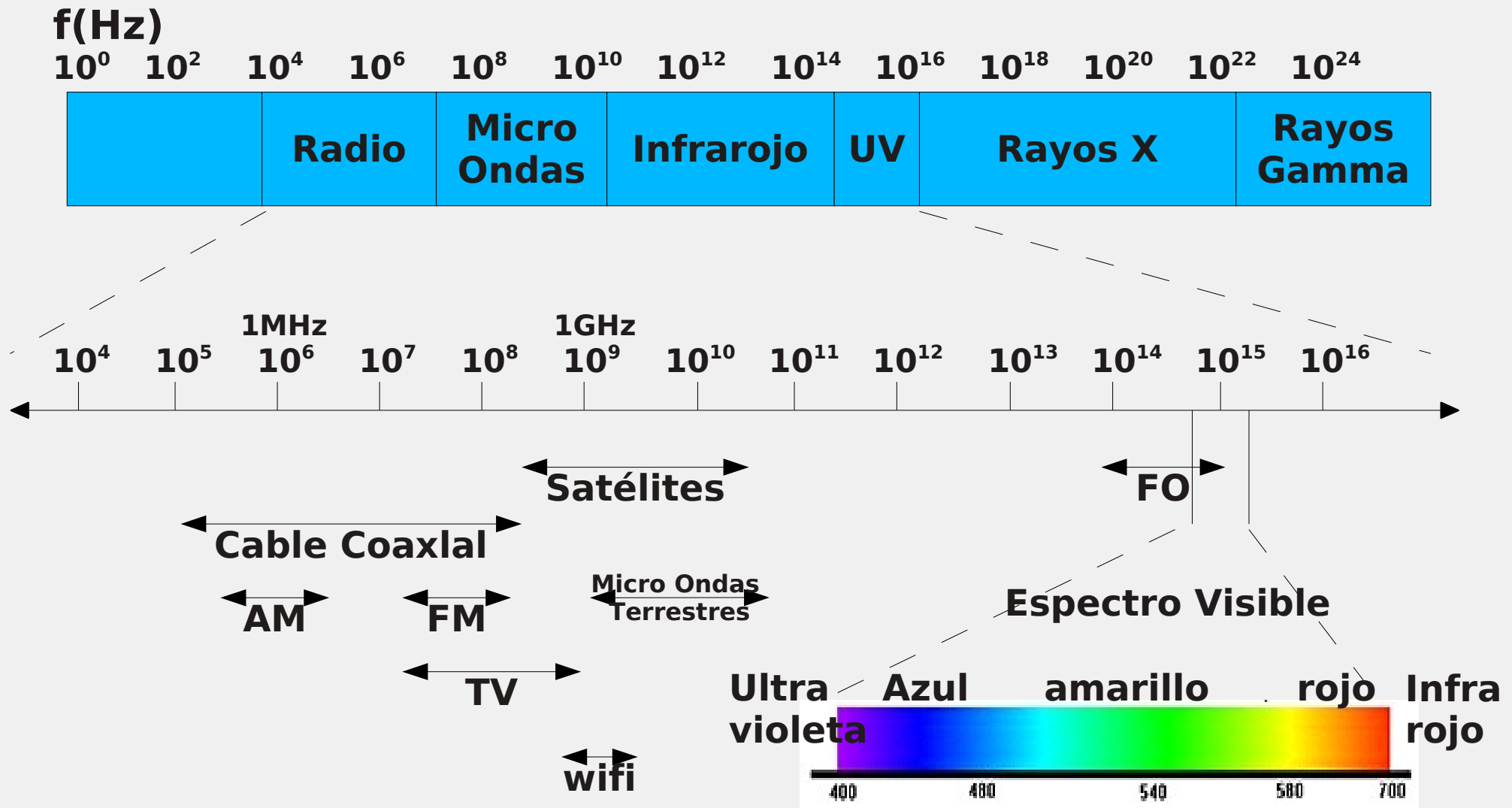
$$\text{Long} = 667 \text{ metros}$$



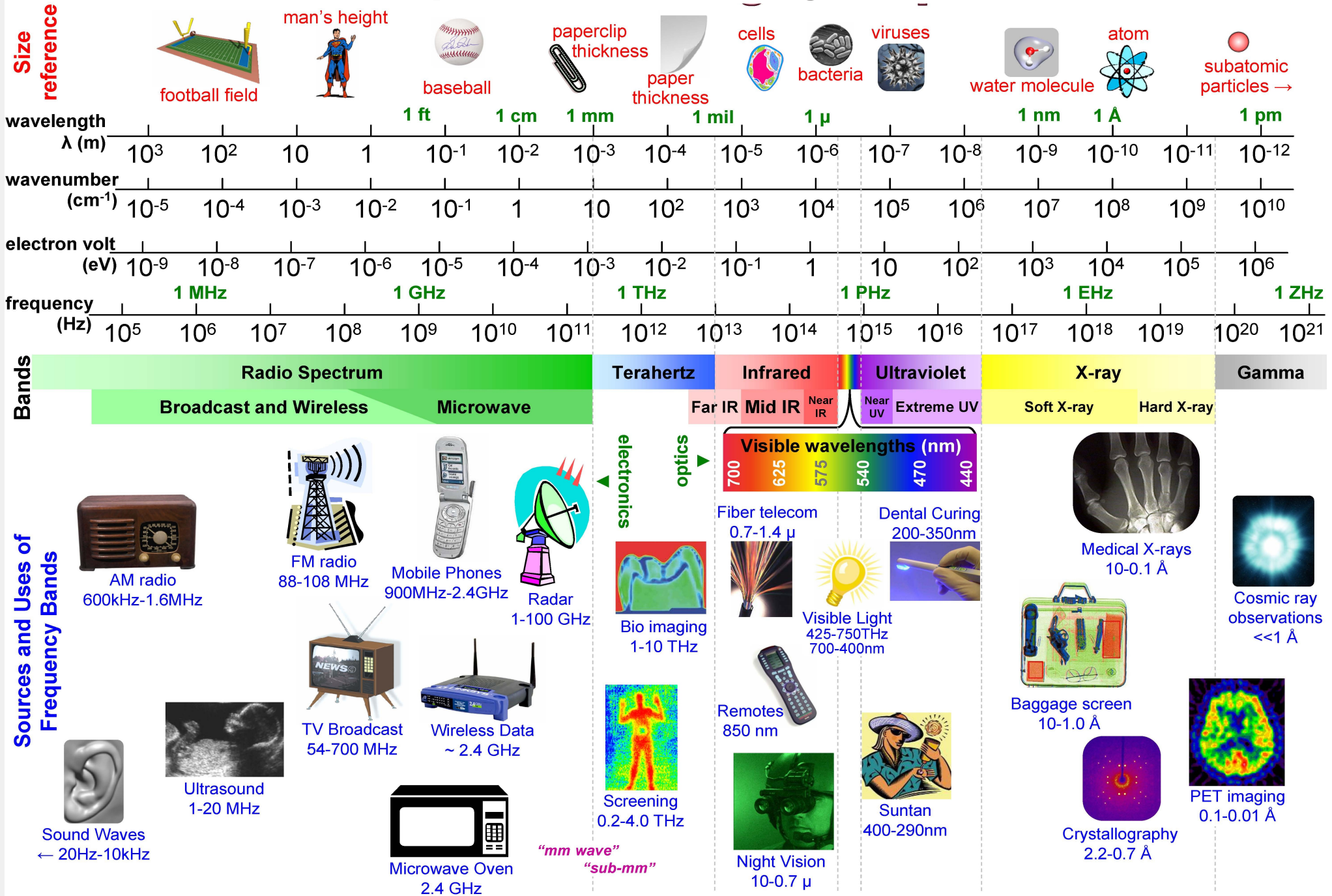
Longitud

Enlaces

Podemos encontrar diferentes tipos de ondas en el espectro electromagnético



Espectro Electromagnético



$$\lambda = 3 \times 10^8 / \text{freq} = 1 / (\text{wn} * 100) = 1.24 \times 10^{-6} / \text{eV}$$

Enlaces

- Entendemos entonces que un enlace es un medio físico capaz de transitar toda clase de ondas electromagnéticas
- Tales enlaces proporcionan la base para transmitir cualquier tipo de información incluyendo la que nosotros estamos interesados, datos binarios (ceros y unos).
- El proceso de convertir esos datos binarios a ondas se le llama codificación

Enlaces

El proceso de convertir esos datos binarios a ondas es bastante complejo y para hacerlo más manejable dividimos el problema en dos capas

- **Modulación:** tiene que ver con la variación de la frecuencia, la amplitud y fase de la señal.
- **Codificación:** convertir los datos binarios a esas señales

Enlaces

Como el problema de modulación no es relevante para nosotros en el diseño de redes simplemente consideraremos que es posible transmitir un par de señales que se puede distinguir una de la otra:

- Una señal baja
- Una señal alta

Así que sólo consideraremos el problema de codificar los datos binarios en este par de señales

Enlaces

Un problema adicional con el que se tiene que lidiar es la degradación de las señales con la distancia.

El teorema de Shannon trata este problema y le asigna un límite superior a la capacidad de un enlace en términos de bits por segundo, como una función de la relación señal – ruido del enlace, y se mide en decibeles (dB)

Enlaces

El teorema de Shannon es descrito por la siguiente fórmula

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

Donde

- C es la capacidad alcanzable del enlace medida en Hertz
- B es el ancho de banda de la línea
- S es la potencia promedio de la señal
- N es la potencia de la señal de ruido

Cables

Actualmente tenemos una serie de medios de transmisión que se utilizan en el diseño de redes los cuales utilizan distintos anchos de banda

• Par trenzado categoría 5	10-1000Mbps	100 m
• Coaxial fino	10-100Mbps	200 m
• Coaxial grueso	10-100Mbps	500 m
• Fibra óptica multimodo	100Mbps	2 km
• Fibra óptica monomodo	100-2400Mbps	40 km

Conectando Físicamente los nodos

Factores a considerados para interconectar un par de nodos.

- **Codificación**
- Framing
- Detección de errores
- Transmisión confiable
- Control de Acceso al Medio

Codificación

Como hemos dicho hicimos a un lado el problema de modulación asumiendo que tenemos dos señales: una baja y otra alta

En la práctica esto puede ser traducido a dos voltajes diferentes en un cable de cobre o dos niveles diferentes de potencia de luz en un enlace óptico

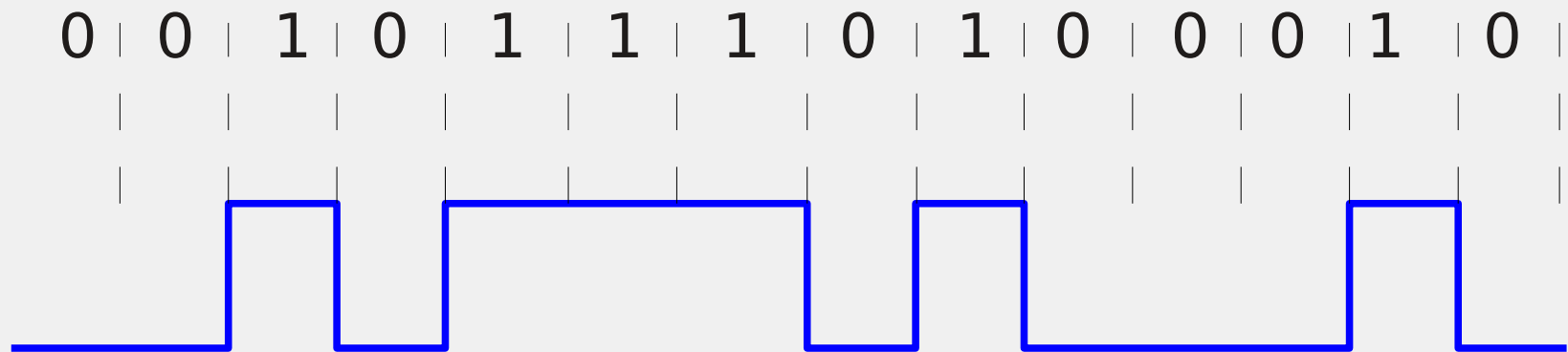
Codificación

La tarea de convertir los datos binarios a las señales es realizada por un aparato llamado adaptador de red, una pieza de hardware que conecta el nodo a un enlace.

En el problema de codificación lo más obvio es mapear un 1 a la señal alta y un 0 a la señal baja.

Codificación

Non Return to Zero (NRZ): Es un esquema de codificación que hace este tipo de mapeo.



El problema con este esquema es que una secuencia de 1s o 0s mantienen una señal alta o baja por mucho tiempo.

Codificación

Non-Return to Zero (NRZ): hay dos problemas con estas secuencias largas:

- El receptor utiliza un promedio para distinguir entre una señal baja y una señal alta. El problema es que muchos 0s o 1s afectan ese promedio y le es más difícil realizar la distinción.
- El otro problema es que tanto el emisor como el receptor deben tener sincronizado el reloj pues en el instante que el emisor codifica un bit el receptor tiene que decodificar un bit.

Codificación

Podríamos solucionar este último problema enviando la señal del reloj por un cable alterno, pero esto duplicaría los costos.

Lo que se hace es que el receptor obtiene el reloj de la misma señal de los datos. Cuando la señal cambia de 1 a 0 o de 0 a 1 el receptor sabe que ocurre un ciclo de reloj.

Sin embargo, en situaciones donde hay periodos de tiempo largos sin ocurrir estas transiciones el reloj puede desajustarse

Codificación

Non-Return to Zero Inverted (NRZI): Este es un enfoque diferente que trata de solucionar los problemas especificados anteriormente.

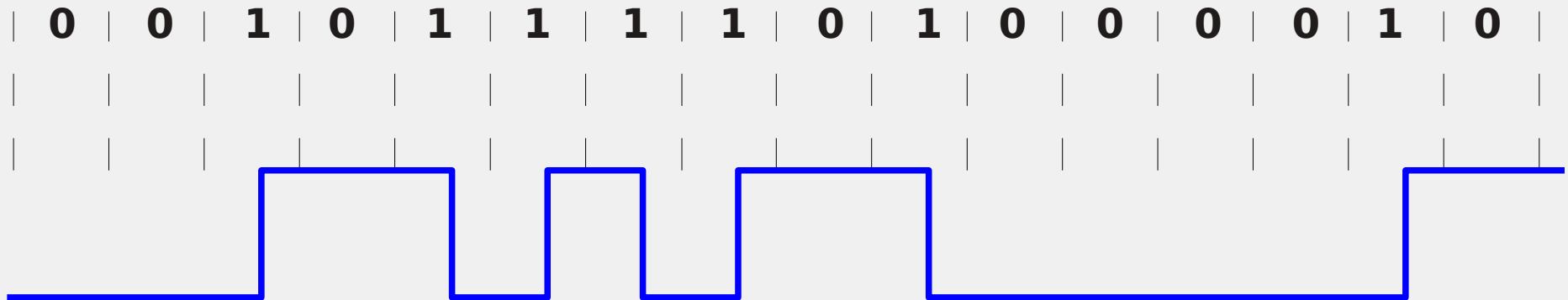
Aquí se codifican los 1s como un cambio de señal y los 0s se interpretan cuando no hay cambio de señal.

Esto resuelve el problema de 1s consecutivos pero no el de los 0s

Codificación

Non-Return to Zero Inverted (NRZI)

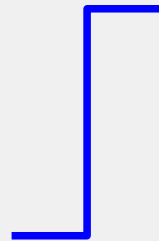
Ejemplo:



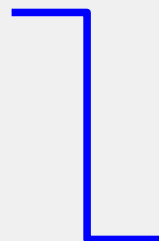
Codificación

Manchester: Este es un esquema de codificación que hace un trabajo más explícito agregando el reloj a la señal.

Los 1s los representa con un cambio en la señal de baja a alta



Y los 0s los representa con un cambio de alta a baja.

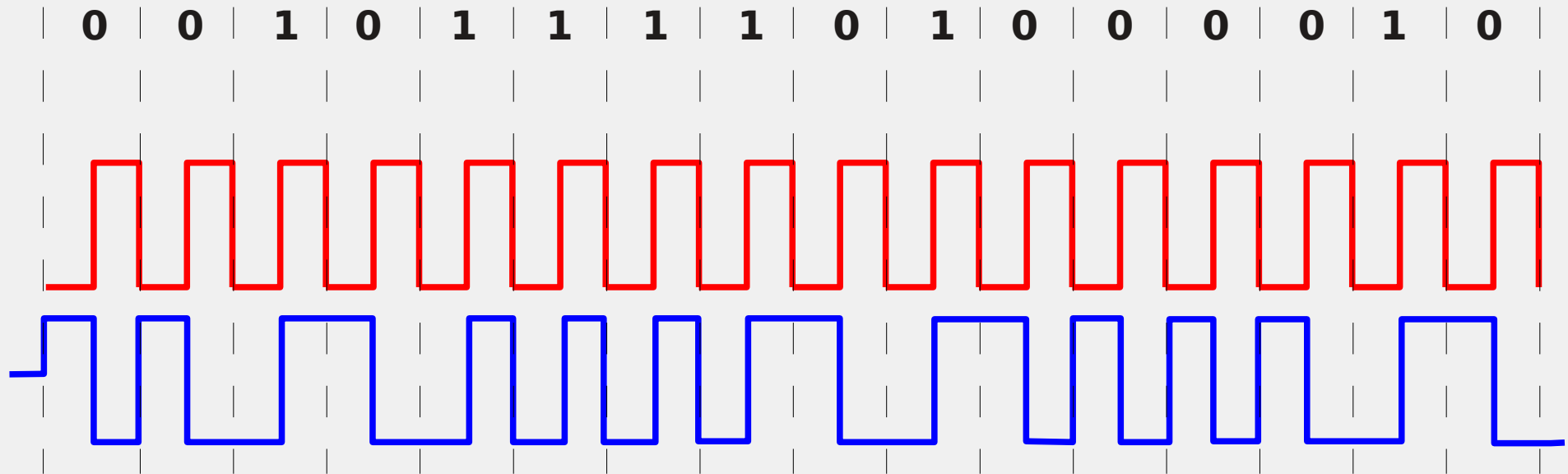


Codificación

Manchester:

Este garantiza que los relojes tanto del receptor como del emisor esten sincronizados

Esta es la codificación utilizada en Ethernet



Codificación

Manchester:

El problema de este esquema es que duplica la tasa a la cual la señal de transmisión es hecha sobre el enlace

Esto se traduce en que el receptor tiene la mitad del tiempo para detectar cada pulso de la señal

Codificación

4B5B:

Este esquema de codificación resuelve la ineficiencia del código manchester sin sufrir del inconveniente de las duraciones largas de señales bajas o altas.

La idea es insertar bits adicionales para romper las secuencias de 1s y 0s largas.

Específicamente, cada 4 bits de datos se codifican en 5 bits.

Codificación

4B5B:

Los códigos de 5 bits se seleccionan de tal manera que no exista más de un cero al comienzo ni más de dos ceros seguidos.

0000	11110	1110	11100
0001	01001	1111	11101
0010	10100		
0011	10101		
0100	01010		
0101	01011		
0110	01110		
0111	01111		
1000	10010		
1001	10011		
1010	10110		
1011	10111		
1100	11010		
1101	11011		

Codificación

4B5B:

Sólo se utilizan 16 de las 32 posibles combinaciones

11111 se utiliza para indicar que la línea está desocupada

00000 indica que la línea está muerta

00100 significa halt

Siete de las restantes combinaciones no sirven pues violan la regla de los ceros

Codificación

4B5B:

Las 6 combinaciones restantes se utilizan para representar símbolos de control.

Se utiliza NRZI para transmitir el código de 5 bits

Codificación

4B5B:

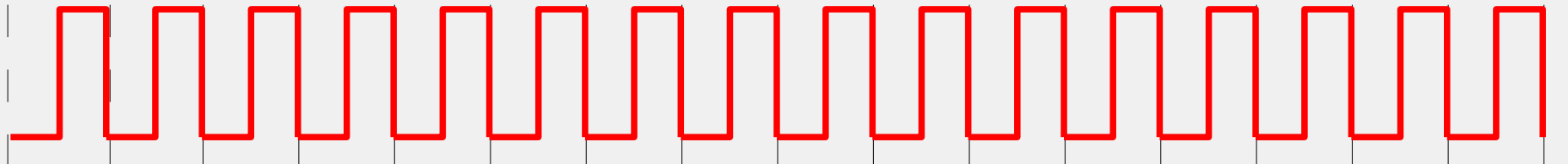
El problema de este esquema es que se necesita 25% más de ancho de banda para transmitir la misma información.

Esta codificación es la que se utiliza en el estándar Ethernet 100BASE-TX

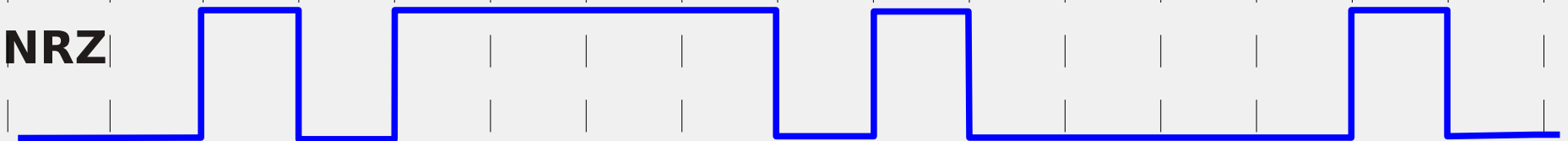
Codificación

0 0 1 0 1 1 1 1 0 1 0 0 0 0 1 0

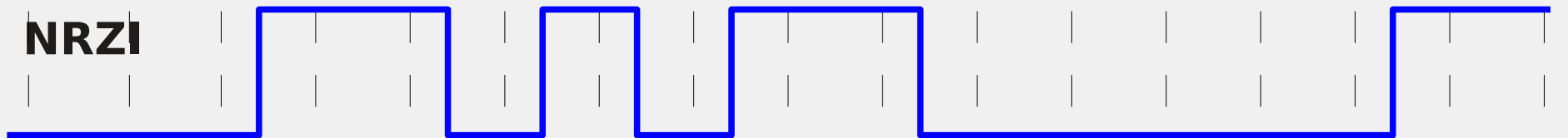
reloj



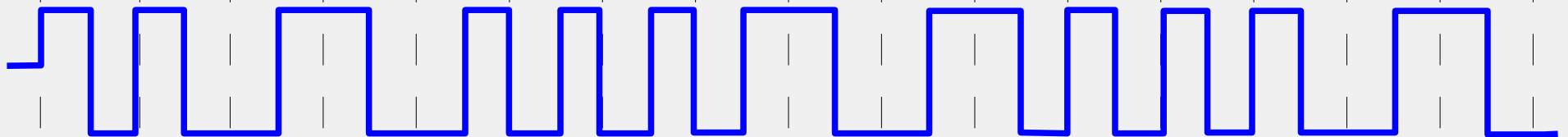
NRZ



NRZI



Manchester



Codificación

Bit rate y Baud rate

El baudio (en inglés, baud) se utilizó originariamente para medir la velocidad de las transmisiones telegráficas, tomando su nombre del ingeniero francés Jean Maurice Baudot, que fue el primero en realizar este tipo de mediciones.

Codificación

Bit rate y Baud rate

El baudio es la unidad informática que se utiliza para cuantificar el número de cambios de estado, o eventos de señalización, que se producen cada segundo durante la transferencia de datos.

La velocidad de transferencia de datos puede medirse en baudios o en bit/segundo. Lo habitual, hoy por hoy, es medirla en bits por segundo.

Codificación

Bit rate y Baud rate

Es importante resaltar que no se debe confundir el **baud rate** o velocidad en baudios con el **bit rate** o velocidad en bits por segundo, ya que cada evento de señalización transmitido puede transportar uno o más bits. Sólo cuando cada evento de señalización transporta un solo bit coinciden la velocidad de transmisión de datos en baudios y en bits por segundo.

Conectando Físicamente los nodos

Factores a considerados para interconectar un par de nodos.

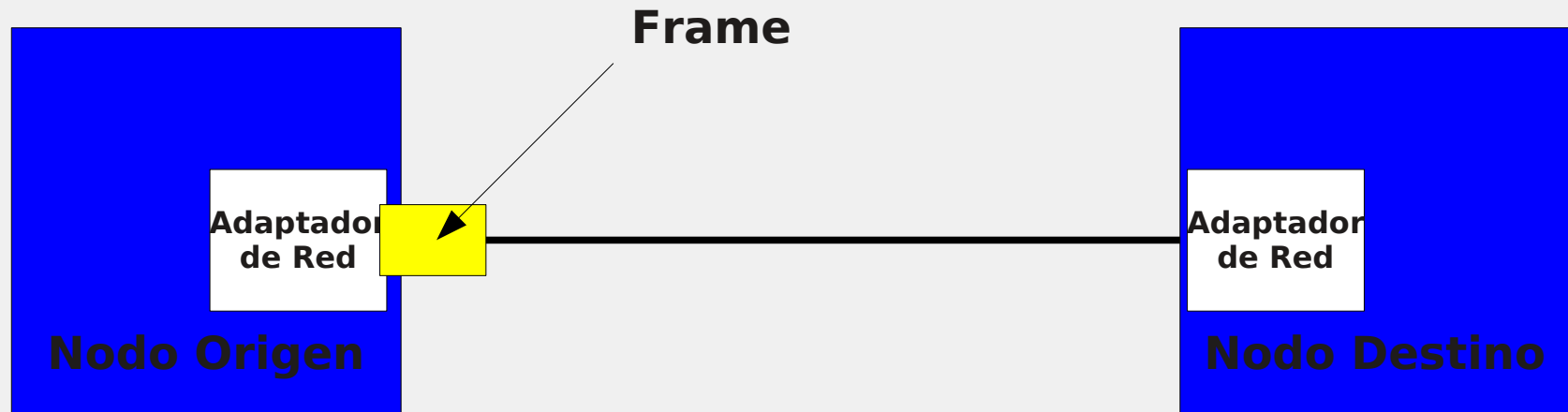
- Codificación
- **Framing**
- Detección de errores
- Transmisión confiable
- Control de Acceso al Medio

Framing

Es el proceso de insertar elementos de control antes y después de transmitir datos constituyendo un límite para ellos al servir de bordes.

Frame

Es el término utilizado para hacer referencia a un paquete de datos codificado para ser transmitido por un enlace (Capa de enlace del modelo OSI)



Framing

Existen varias estrategias para tratar este problema:

- Enfoque de Bytes
- Enfoque de Bits
- Enfoque Basado en reloj

Framing

Enfoque de Bytes

Binary Synchronous Communication (BISYNC)

Point to Point Protocol (PPP)

Digital Data Communication Message Protocol (DDCMP)

Este es uno de los más antiguos enfoques y tiene sus orígenes en la interconexión de terminales a mainframes.

Framing

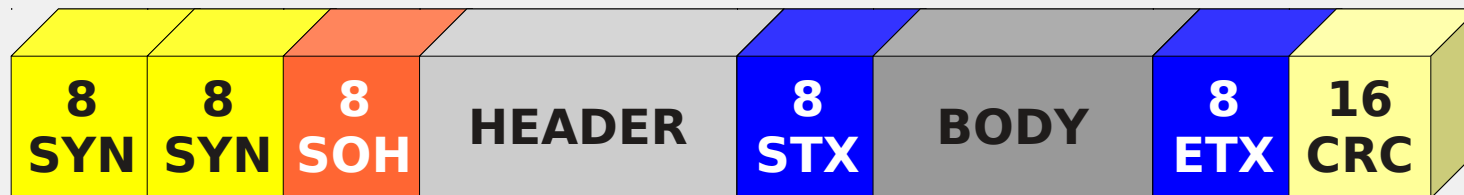
El enfoque de Bytes utiliza dos mecanismos

- Centinela
- Conteo de bytes

Framing

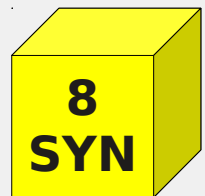
Enfoque de Bytes

- Centinela: Formato del frame de BISYNC



Framing

- Centinela: Formato del frame de BISYNC



Synchronization character



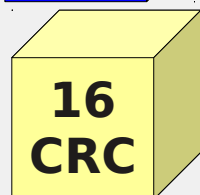
Start Of Header



Start of Text: se escapa con data link escape DLE



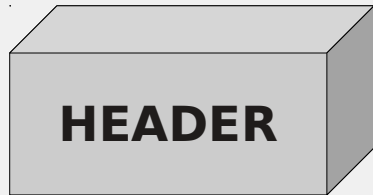
End of Text



Cyclic Redundancy Check

Framing

- Centinela: Formato del frame de BISYNC



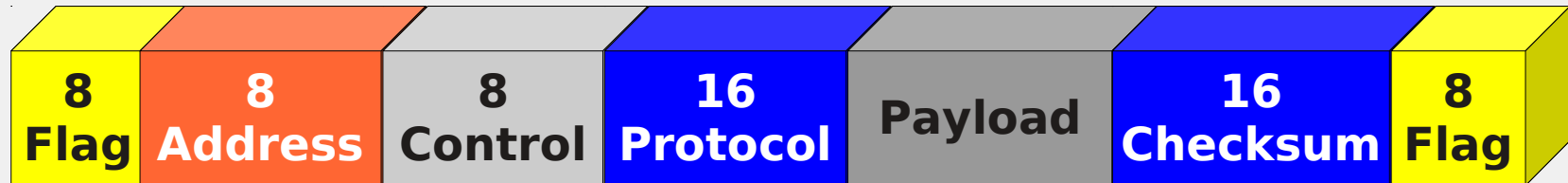
La cabecera tiene otros campos que se utilizan, entre otras cosas, en los algoritmos de transmisión confiable.

Este enfoque es conocido también como *character stuffing* porque caracteres extras son insertados en la porción de los datos del frame

Framing

Enfoque de Bytes

- Centinela: Formato del frame de PPP. Este protocolo es utilizado generalmente en enlaces discados



Framing

Enfoque de Bytes

- Centinela: Formato del frame de PPP



Start of text



El campo de dirección y control siempre contienen valores predefinidos.

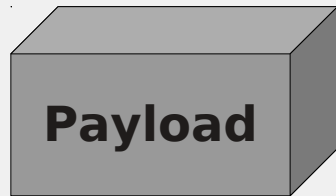


Es utilizado para el proceso de multiplexación. Este define el protocolo de alto nivel tal como IP o IPX

Framing

Enfoque de Bytes

- Centinela: Formato del frame de PPP



El tamaño de este campo puede ser negociado pero por omisión es 1500



Control de errores

Framing

Enfoque de Bytes

- **Conteo de bytes:** La estrategia utilizada aquí es colocar el número de bytes del frame como un campo más en la cabecera.

Uno de los peligros en este enfoque es la posible corrupción del campo de bytes en el proceso de transmisión.

Framing

Enfoque de Bytes

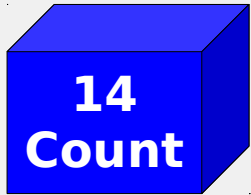
- Conteo de bytes: Formato del protocolo Digital Data Communication Message Protocol (DDCMP)



Framing

Enfoque de Bytes

- Conteo de bytes: DDCMP



Indica cuantos bytes hay en el campo body

Framing

Enfoque de Bits

High Level Data Link Control (HDLC)

Synchronous Data Link Control (SDLC)

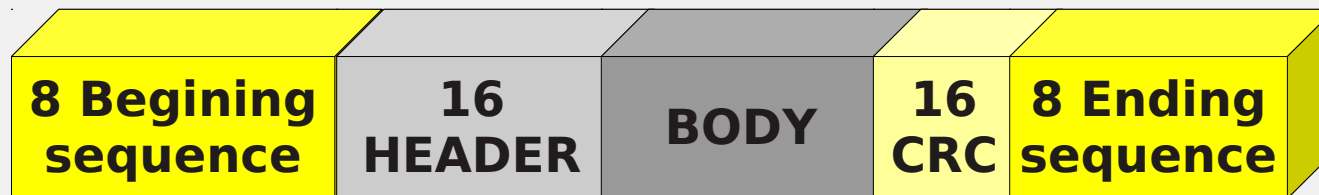
Este enfoque considera los frames como un conjunto de bits en lugar de bytes

Framing

Enfoque de Bits

Formato del frame de HDLC

SDLC fue creado por IBM y luego fue incorporado al modelo OSI como HDLC

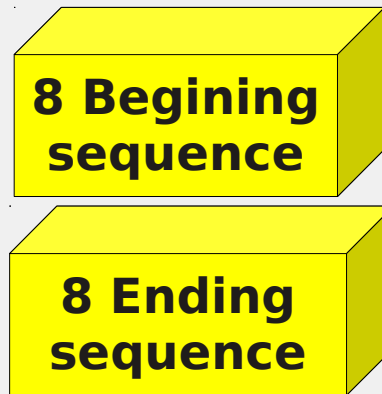


Este protocolo denota el inicio y el fin del frame con una secuencia de bits (01111110)

Framing

Enfoque de Bits

Formato del frame de HDLC



Esta secuencia es transmitida cada vez que el canal esta desocupado

Framing

Enfoque Basado en Reloj

Synchronous Optical Network (SONET)

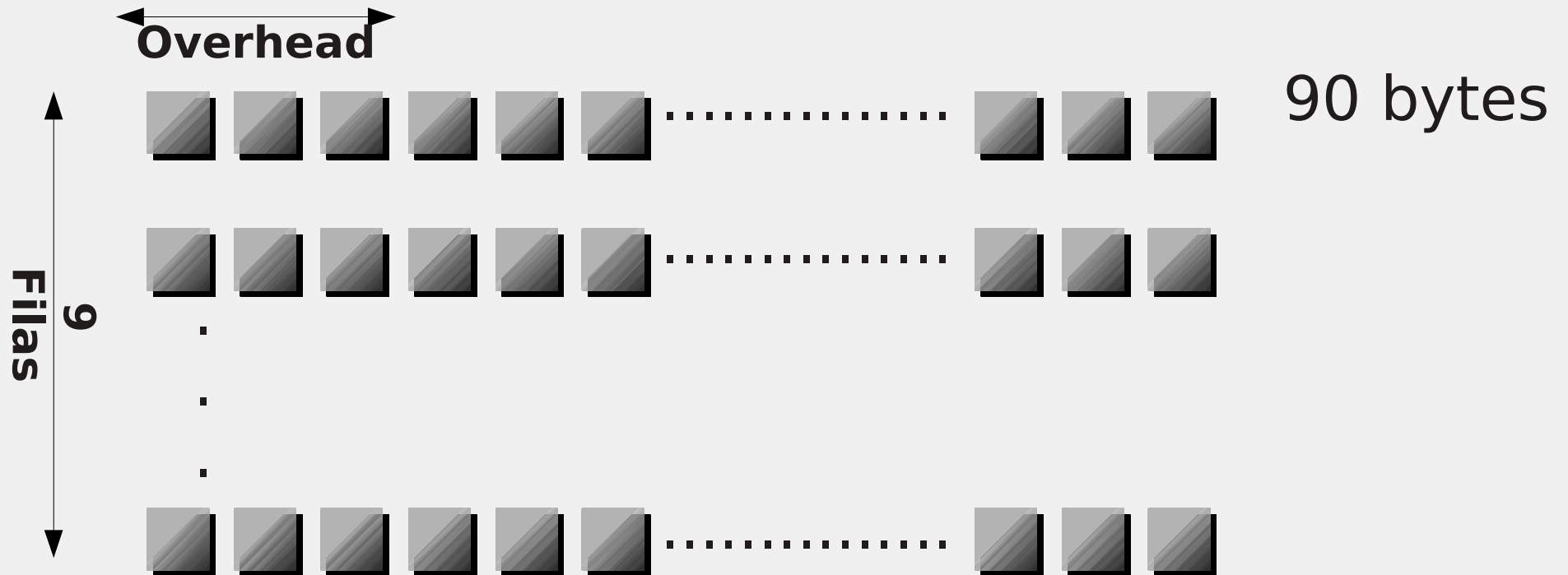
SONET es el protocolo utilizado en fibra óptica y fue propuesto por los laboratorios Bell y desarrollado por el American National Standard Institute (ANSI)

SONET cubre tanto el problema de la codificación como el de framing.

Este utiliza varios enlaces a distintas velocidades. A continuación describimos el más lento de ellos (STS-1)

Framing

Enfoque Basado en Reloj



Los primeros 3 bytes de cada fila son overhead y el resto se utiliza para datos.

Framing

Enfoque Basado en Reloj

SONET utiliza los primeros 2 bytes para indicar el principio del frame. Estos contienen un patrón de bits particular.

El receptor busca este patrón cada 810 bytes para evitar que los bits puedan confundirse con el relleno del frame

Framing

Enfoque Basado en Reloj

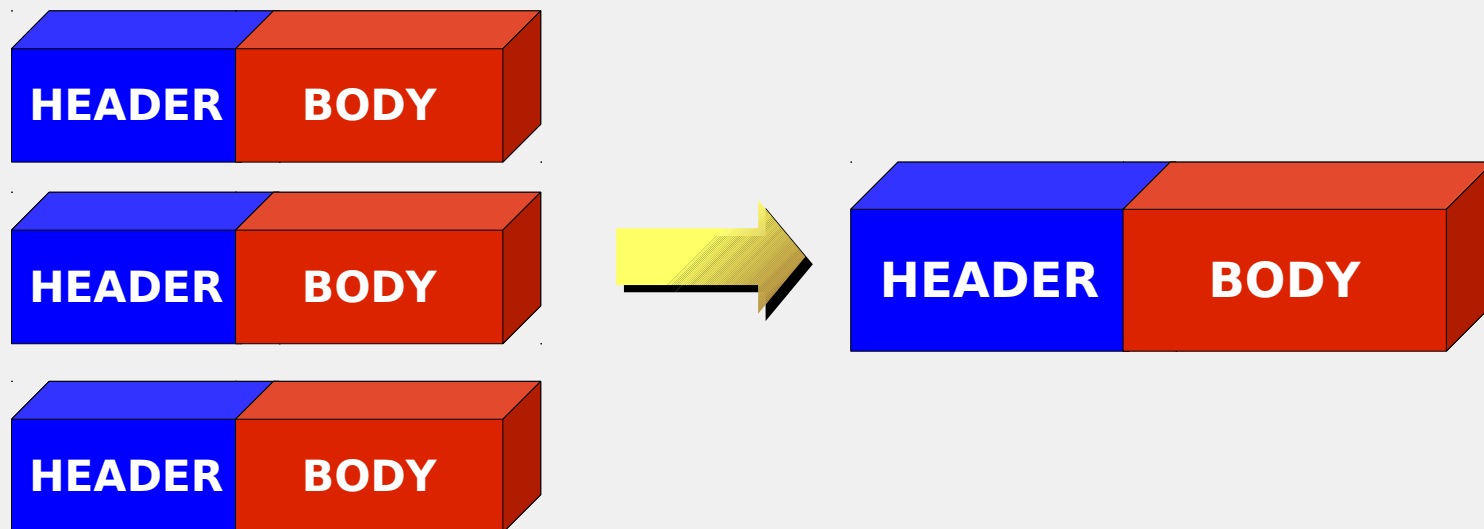
Los bytes de overhead se transmiten utilizando NRZ. Para evitar el inconveniente de cadenas largas de 0s y 1s, se transmite el XOR de los bytes del relleno (payload)

Framing

Enfoque Basado en Reloj

SONET también se ocupa de la multiplexación del canal.

Por ejemplo en un STS-3 caben exactamente 3 frames de STS-1 de 51.84 Mbps.



Conectando Físicamente los nodos

Factores a considerados para interconectar un par de nodos.

- Codificación
- Framing
- **Detección de errores**
- Transmisión confiable
- Control de Acceso al Medio

Deteccción de Errores

Los errores ocurren con poca frecuencia pero es necesario contar con mecanismos para detectarlos y corregirlos.

Estos se originan por interferencia eléctrica o ruido térmico.

Detección de Errores

Básicamente existen dos enfoques para el tratamiento de errores

- Notificar al emisor que envíe una copia del mensaje
- Reconstruir el mensaje

Sin embargo, antes de corregir algún error, es necesario detectar si ha ocurrido

Detección de Errores

Introducir información redundante en un frame es uno de los mecanismos utilizados para determinar si ha ocurrido un error.

Un enfoque es enviar dos copias de un mensaje y al final las dos siguen siendo idénticas, es probable que no haya ocurrido ningún error. Si éstas difieren, entonces puede haber un error.

Detección de Errores

Sin embargo, es deficiente y los errores pueden pasar desapercibidos si una misma posición en los bits son afectados.

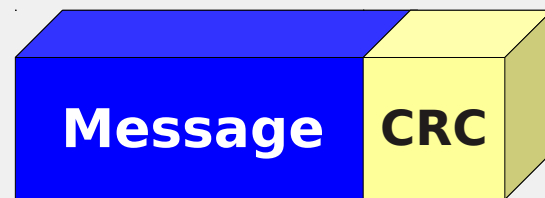
Otra técnica es enviar sólo unos cuantos bits redundantes.

Cyclic Redundancy Check (CRC)

Estos bits se derivan del mensaje original utilizando algún algoritmo y no agregan información adicional.

Detección de Errores

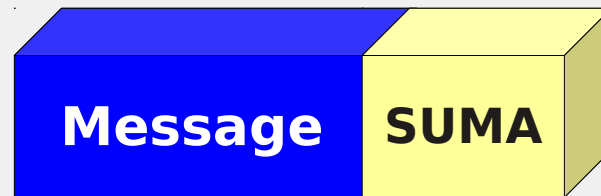
El emisor aplica un algoritmo al mensaje para generar los bits redundantes, y transmite tanto el mensaje como los bits adicionales



El receptor aplica el mismo algoritmo sobre el mensaje y lo compara con el valor que los bits adicionales contiene. Si no coinciden, se concluye que hubo errores en el mensaje o en los bits.

Detección de Errores

Si el algoritmo se basa en una suma el resultado se denomina *checksum*.

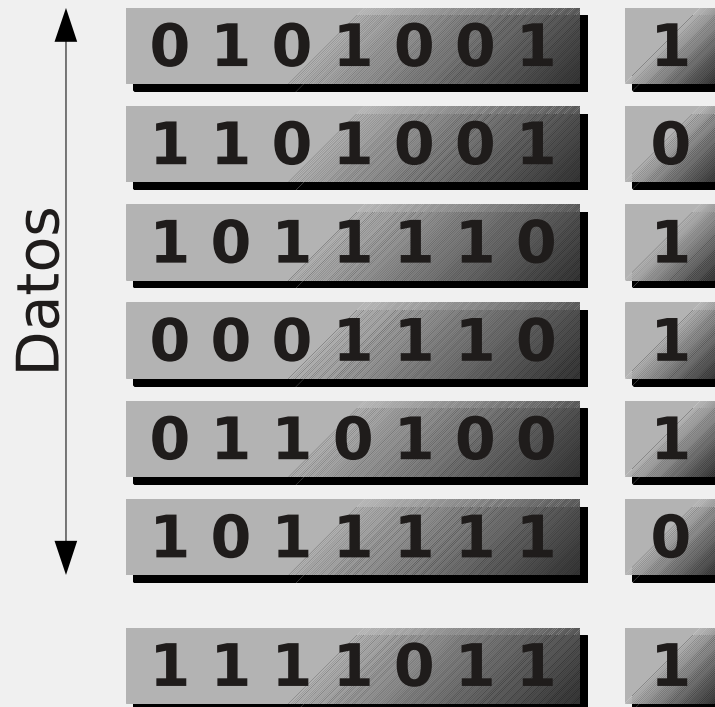


Es mejor utilizar *código de detección de errores* para hablar en forma general.

Detección de Errores

Paridad de dos dimensiones

A un código de 7 bits se le agrega un octavo para hacer par el número de 1s.



Se hace lo mismo en relación a las posiciones de bits de los diferentes bytes

Detección de Errores

Cyclic Redundancy Check

Este enfoque se basa en un área de las matemáticas (campos finitos) que reduce el número de bits necesarios (32) para la detección de errores en un mensaje de miles de bytes

Detección de Errores

Cyclic Redundancy Check

Este enfoque se fundamenta en lo siguiente:

- Un mensaje de $n+1$ bits podemos representarlo como un polinomio de grado n utilizando los bits como los coeficientes del polinomio.

$$M(x) = 1X^7 + 0X^6 + 0X^5 + 1X^4 + 1X^3 + 0X^2 + 1X^1 + 0X^0$$

- El bit más significativo representa el coeficiente del término de más alto grado

Detección de Errores

Cyclic Redundancy Check

La base matemática sin mucho detalle es como sigue:

- Cualquier polinomio $B(x)$ puede ser dividido por un polinomio divisor $C(x)$ si $B(x)$ es de mayor grado que $C(x)$
- Cualquier polinomio $B(x)$ puede ser dividido una vez por un polinomio divisor $C(x)$ si $B(x)$ es del mismo grado que $C(x)$

Detección de Errores

Cyclic Redundancy Check

La base matemática sin mucho detalle es como sigue:

- El resto de la división de $B(x)$ entre $C(x)$ se puede obtener substrayendo $C(x)$ de $B(x)$
- La substracción de $C(x)$ de $B(x)$ se puede realizar ejecutando una operación OR exclusiva (XOR) de los coeficientes correspondientes

Detección de Errores

Cyclic Redundancy Check

- La idea es construir un polinomio $P(x)$ a partir de $M(x)$ (k bits más largo) y que sea divisible exactamente por otro polinomio $C(x)$ de k bits
- Esto lo hacemos de la siguiente manera
 - Multiplicamos $M(x)$ por $X^k = T(x)$
 - Dividimos $T(x)$ por $C(x)$ y encontramos el resto
 - Restamos el resto a $T(x)$

Detección de Errores

Cyclic Redundancy Check

- Si $P(x)$ es transmitido sin errores entonces $P(x)$ será dividido de forma exacta por $C(x)$.
- Si el resto no es cero, entonces hay una alta probabilidad de que hubo errores en la transmisión.

Detección de Errores

Cyclic Redundancy Check

Ejemplo: considere el mensaje $X^7 + X^4 + X^3 + X^1$
es decir, 10011010

Multiplicando por X^3 resulta

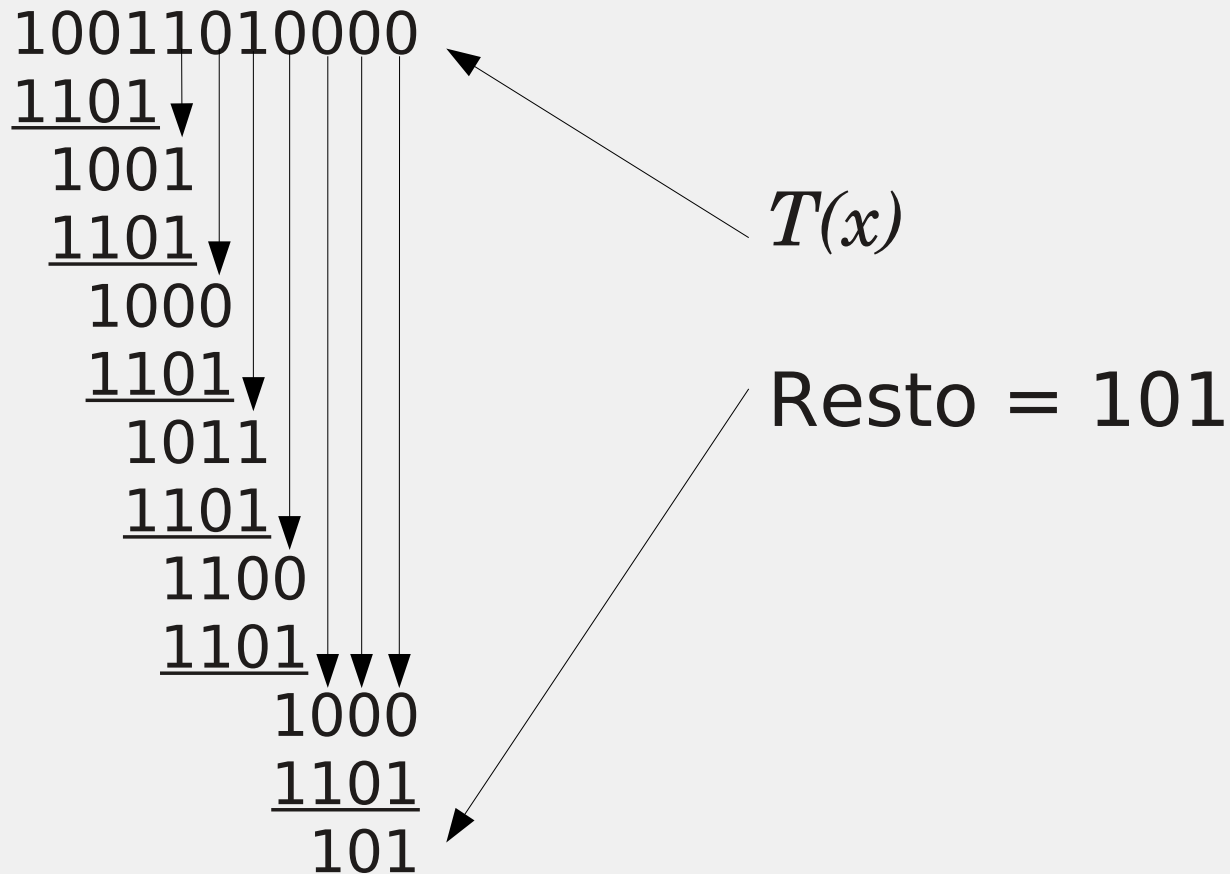
10011010000

Observe que multiplicar X^k es lo mismo que
agregar k ceros a la derecha

Detección de Errores

Cyclic Redundancy Check

Dividimos por $C(x) = 1101$ para conseguir el resto



Detección de Errores

Cyclic Redundancy Check

Substraemos el resto de $T(x)$ Eso es una operación XOR para finalmente obtener $P(x)$

$$10011010000 - 101 = 10011010101$$

$$P(x) = 10011010101$$

Y este resultado es lo que se envía

Detección de Errores

Cyclic Redundancy Check

Ahora, ¿De donde sacamos $C(x)$?

La idea es obtener un polinomio de manera tal que si $P(x)$ tiene errores entonces la probabilidad que $C(x)$ lo divida exactamente sea muy baja

Este problema tiene sus fundamentos matemáticos también pero no entraremos en detalles

DetECCIÓN DE ERRORES

Cyclic Redundancy Check

CRC-8 $X^8 + X^2 + X^1 + 1$

CRC-10 $X^{10} + X^9 + X^5 + X^4 + X^1 + 1$

CRC-12 $X^{12} + X^{11} + X^3 + X^2 + 1$

CRC-16 $X^{16} + X^{15} + X^2 + 1$

CRC-CCITT $X^{16} + X^{15} + X^5 + 1$

CRC-32 $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} +$

$X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

Detección de Errores

Cyclic Redundancy Check

El algoritmo para hallar $P(x)$ es complicado pero por las propiedades particulares que tiene puede ser implantado utilizando compuertas XOR y registros de desplazamiento (*shift*)

Conectando Físicamente los nodos

Factores a considerados para interconectar un par de nodos.

- Codificación
- Framing
- Detección de errores
- **Transmisión confiable**
- Control de Acceso al Medio

Conectando Físicamente los nodos

Transmisión Confiable

- Hemos visto como podemos detectar errores
- Sin embargo, los mecanismos para realizar tal tarea introduce una sobrecarga sustancial
- Por lo tanto los frames corruptos deben ser descartados
- Debemos contar entonces con mecanismos que nos ayude a recuperar la comunicación luego de eliminar los frames con errores

Conectando Físicamente los nodos

Transmisión Confiable

Contamos con dos mecanismos que utilizados en conjunto pueden recuperar la transmisión de los errores

- Reconocimientos (*Acknowledgments o Acks*)
- Tiempos límites (*Timeouts*)

Conectando Físicamente los nodos

Transmisión Confiable

Esta estrategia de utilizar *acks* y *timeouts* para implantar una transmisión confiable se conoce también como solicitud de repetición automática (*automatic repeat request ARQ*)

- Parar y Esperar (Stop-and-Wait)
- Ventana deslizante (Sliding Window)
- Canales lógicos concurrentes (Concurrent Logical Channels)

Conectando Físicamente los nodos

Transmisión Confiable

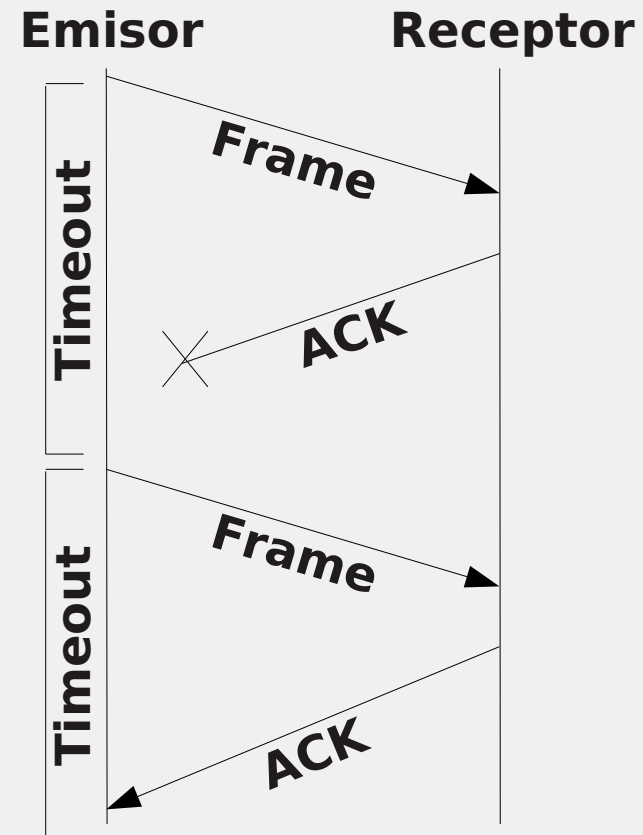
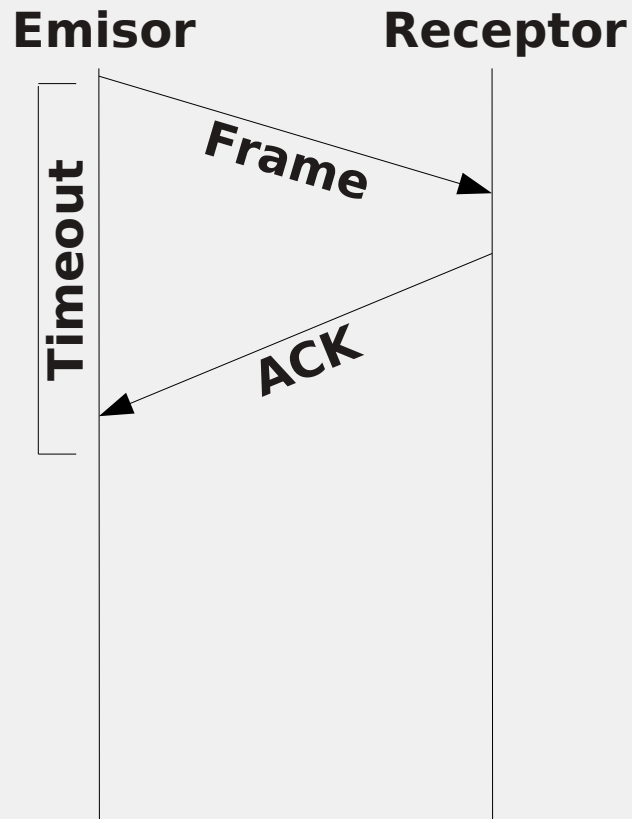
- Para y Esperar

Esta es la estrategia más sencilla: El emisor, después de enviar un frame, espera por un ack antes de enviar el siguiente frame

Conectando Físicamente los nodos

Transmisión Confiable

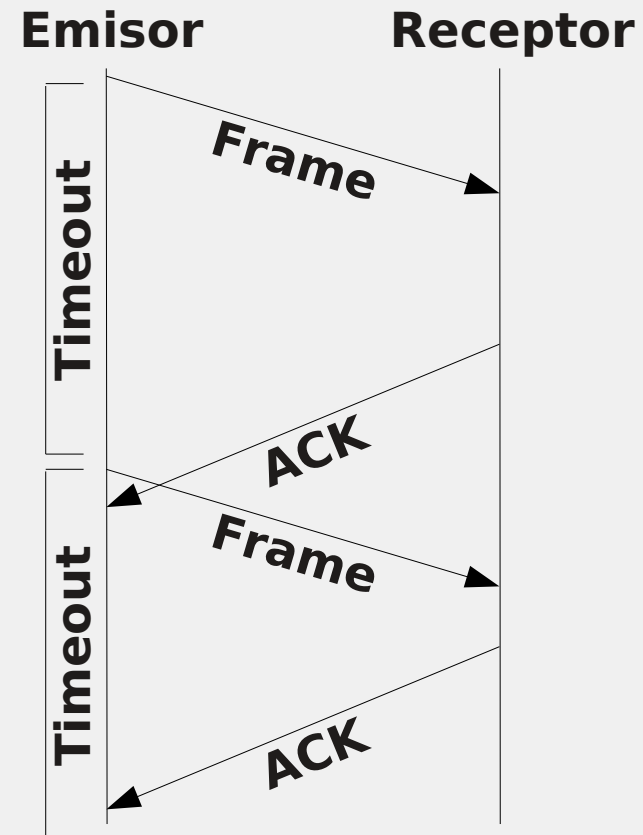
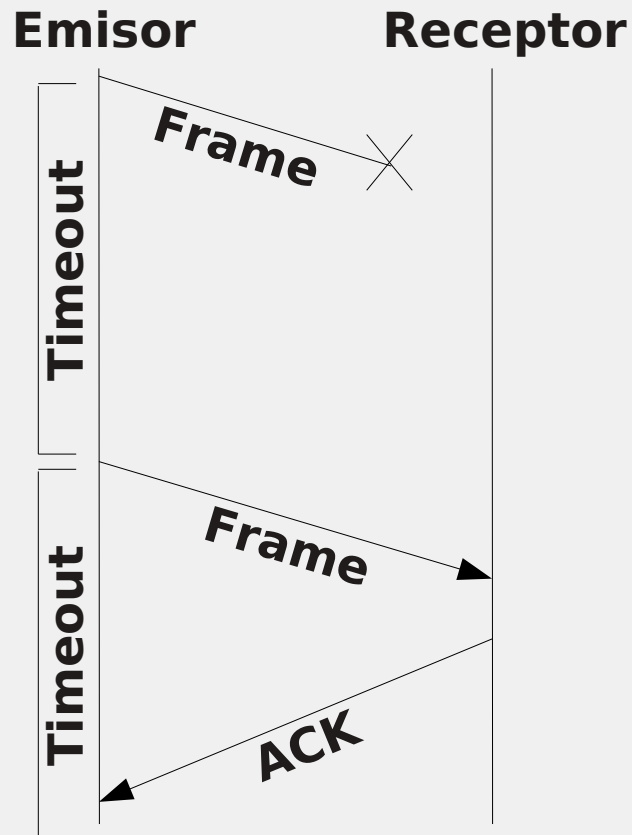
- Para y Esperar



Conectando Físicamente los nodos

Transmisión Confiable

- Para y Esperar



Conectando Físicamente los nodos

Transmisión Confiable

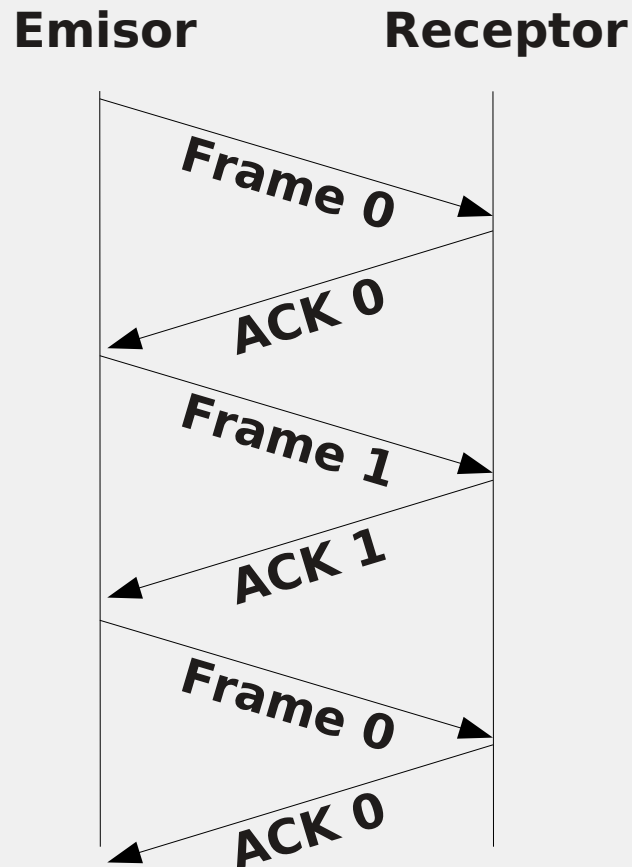
- Para y Esperar

Para evitar la duplicación de frames que puede resultar del último caso, se agrega un bit para identificar al frame

Conectando Físicamente los nodos

Transmisión Confiable

- Para y Esperar



Conectando Físicamente los nodos

Transmisión Confiable

- Para y Esperar

El problema principal de este esquema es que puede no aprovechar completamente el canal de comunicaciones

Conectando Físicamente los nodos

Transmisión Confiable

- Ventana Deslizante

Suponga que tenemos un canal con un producto de demora por ancho de banda de 16 KB. Si tenemos frames de 1KB entonces el emisor puede enviar 16 frames antes de recibir el ack del primero de ellos.

Conectando Físicamente los nodos

Transmisión Confiable

- Ventana Deslizante

