*20 de Mayo de 2024*

Redes de Teleinformática I - Práctico VIII

*Nombres y foto de los integrantes:*

* Lucero Ruiz, Maximo
* Borgogno, Francisco
* Quesada, Santiago
* Cañas, Felipe



**Tema:**

* Capa 2 de modelo OSI, Capa de Enlace, Funciones

**Objetivo de la clase:**

Comprender las funciones necesarias que debe cumplir el segundo nivel del modelo OSI, o nivel de enlace para que una vez que hemos vinculado dos o mas equipos mediante la utilización de un medio físico, podamos dar forma a la información tomando un molde que se llama trama o frame y con el establecer la comunicación lo más eficientemente posible y sin errores.

1. **Transmisión asíncrona y síncrona**

Una vez que hemos definido el medio de transmisión surge la pregunta, ¿Cómo hacen los dispositivos que se quieren comunicar para procesar esos bit o señales que le llegan desde un medio y linkearse o enlazarse entre ellos?.

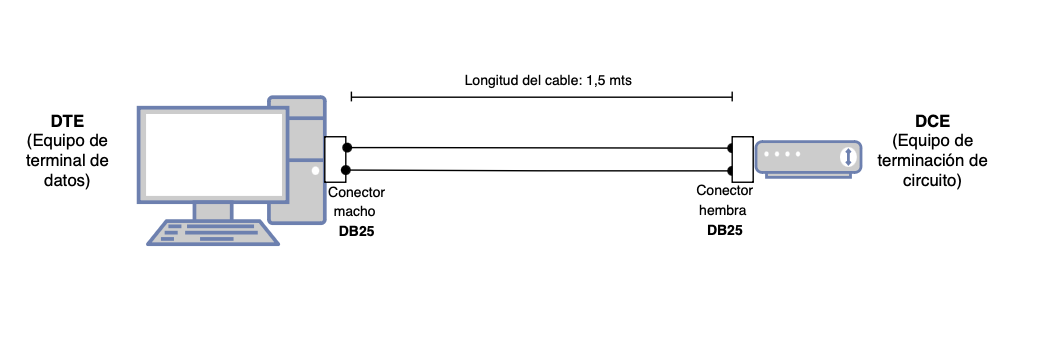
**Preguntas**

<https://www.serial-port-monitor.org/es/articles/serial-communication/rs232-interface/#:~:text=En%20el%20protocolo%20RS232%2C%20se,ambos%20lados%20de%20la%20comunicaci%C3%B3n>.

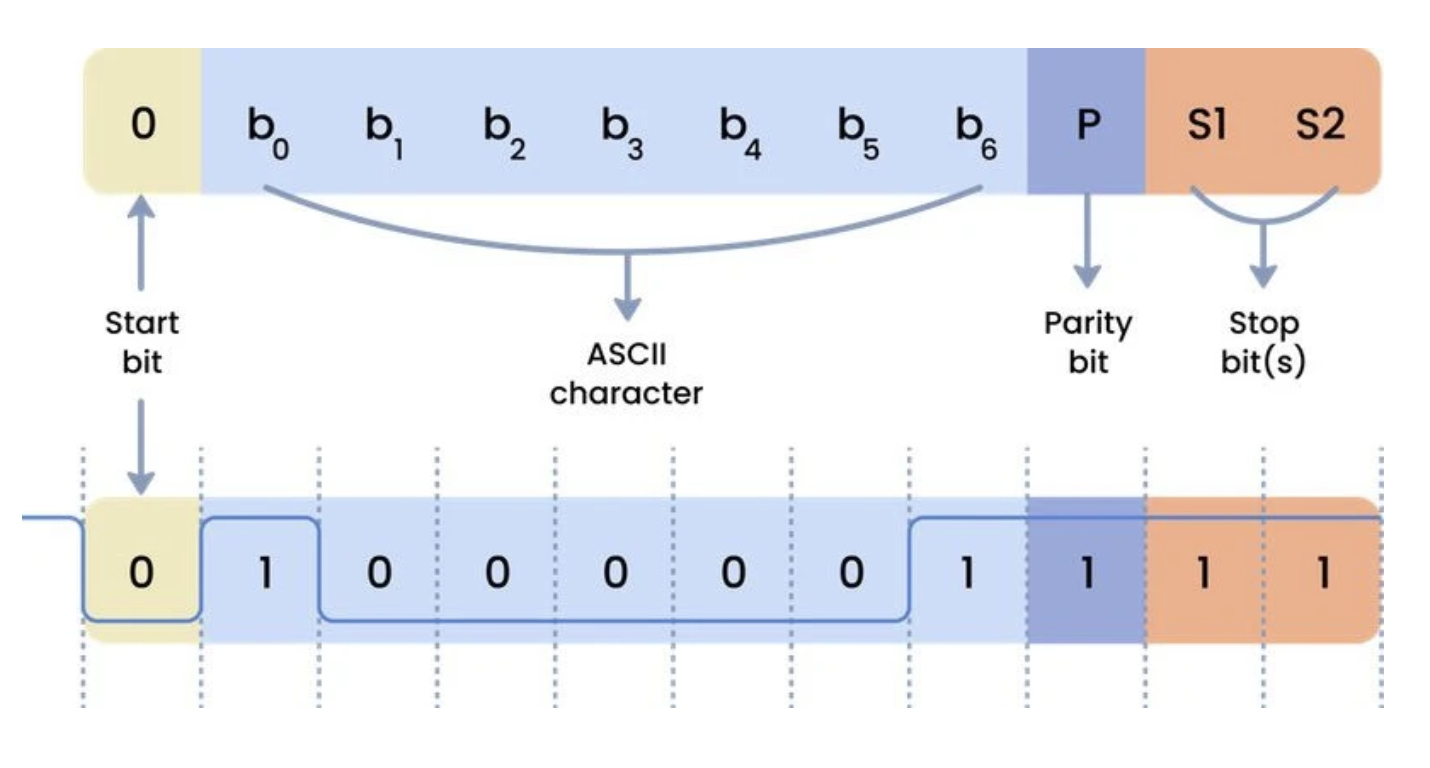
1. ¿A qué llamamos transmisión asíncrona?

La transmisión asíncrona es un método de transmisión de datos en el cual los datos se envían uno a la vez sin una sincronización precisa entre el transmisor y el receptor. Cada carácter enviado está delimitado por bits de inicio y de parada, que indican el comienzo y el final del carácter, respectivamente. Este tipo de transmisión es útil para comunicaciones donde la cantidad de datos no es constante o donde la sincronización constante no es práctica.

1. Realice un gráfico donde se pueda visualizar una conexión asíncrona entre una PC y un modem utilizando la RS- 232. (Defina el conector, la longitud del cable y la interfaz a la que se conecta en la PC y el MODEM)



1. ¿Cómo es el formato de una trama asíncrona definida en la RS 232?



1. ¿Qué funciones cumplen los diferentes campos del carácter o trama?

* **Bit de inicio:** Marca el comienzo de la transmisión de un carácter.
* Bits de datos: Contienen el valor binario del carácter transmitido.
* **Bit de paridad:** Facilita la detección de errores al añadir un bit que hace que el número total de bits de 1 sea par o impar.
* **Bits de parada:** Indican el final del carácter y proporcionan un intervalo de tiempo para la sincronización del receptor.

1. ¿Qué velocidades de transmisión se pueden alcanzar con las comunicaciones asíncronas y porque no se usan para transmitir a velocidades mayores?

Las velocidades de transmisión para la comunicación asíncrona varían entre 110 bps y 230400 bps (siendo 1200, 4800, 9600 y 115200 bps las más utilizadas). La limitación en velocidades más altas se debe a la necesidad de una sincronización precisa; a mayores velocidades, la acumulación de errores de sincronización puede provocar pérdida de datos o errores en la transmisión.

1. ¿Cuáles son las ventajas que tiene una comunicación asíncrona con respecto a una comunicación síncrona?

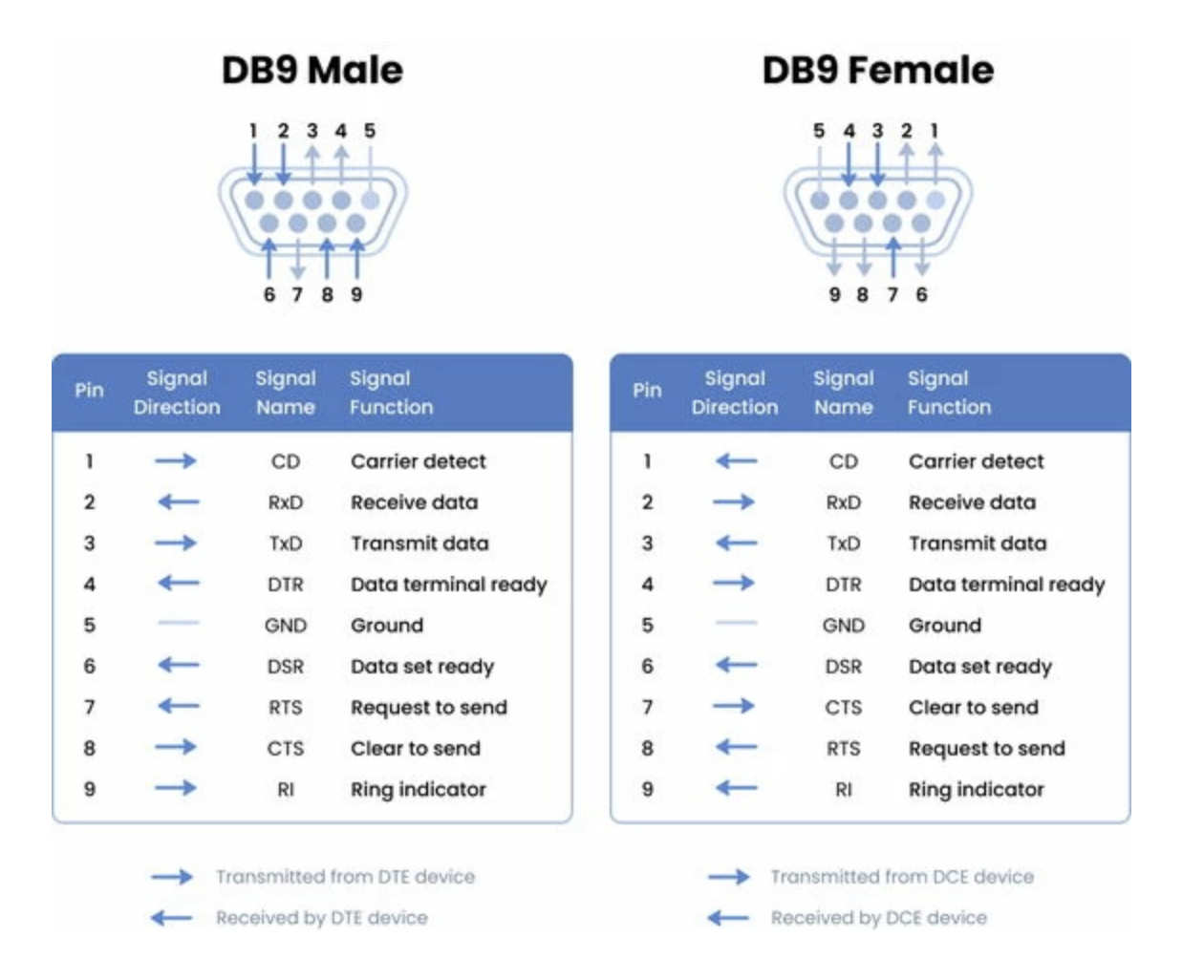
* **Simplicidad:** No requiere la sincronización constante de los relojes de transmisión y recepción.
* **Flexibilidad:** Ideal para transmisiones intermitentes y donde la cantidad de datos es variable.
* **Costo:** Menor costo y complejidad del hardware necesario.

1. ¿Qué parámetros mecánicos, eléctricos y funcionales definen las Recomendaciones RS 232 en las comunicaciones asíncronas?

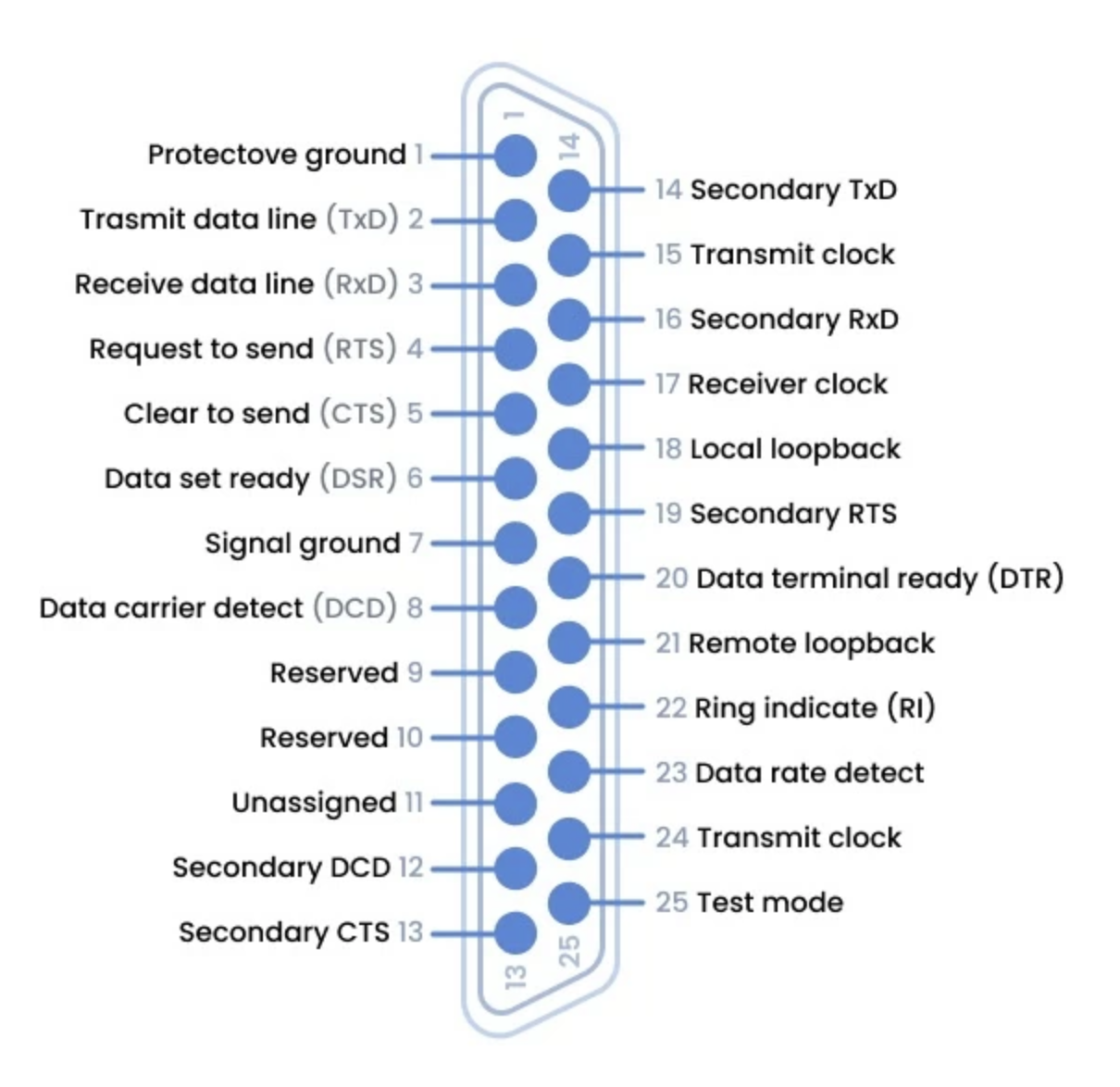
* **Mecánicos:** Tipo de conector (DB-9 o DB-25), longitud máxima del cable (hasta 15 metros).
* **Eléctricos:** Niveles de voltaje (±3V a ±15V), resistencia de terminación.
* **Funcionales:** Funciones de control de flujo, sincronización de bits de inicio y parada.

1. ¿Cómo es el pinout de una comunicación RS 232 tanto para un conector DB-9 y uno DB-25?

**Conector DB-9**



**Conector DB-25**



1. ¿A que llamamos rendimiento de una trama?

El rendimiento de una trama se refiere a la eficiencia con la cual los datos útiles son transmitidos en comparación con la sobrecarga (bits de inicio, parada y paridad).

1. ¿Cómo se calcula el rendimiento de una trama asíncrona?

El rendimiento se calcula como la proporción de bits de datos útiles en una trama con respecto al total de bits transmitidos:

**Funciones de la capa de enlace**

**Tamenbaum:**

<https://elibro.net/es/ereader/bibliotecas-ucc/37871?page=193>

**Stalling (recomendado)**

<https://elibro.net/es/ereader/bibliotecas-ucc/45316?page=205>

La Capa II, o capa de enlace, cumple 5 funciones esenciales, pueden ser más pero nos vamos a focalizar en estas 5:

1. **Sincronización, comienzo y fin de trama**
2. **Control de Flujo de datos**
3. **Detección de Errores**
4. **Control de Errores**
5. **Direccionamiento**

Para comprender las funciones que realiza la capa 2, o capa de enlace, es necesario que previamente nos preguntemos algunas cosas…

* **1 - Sincronización, comienzo y fin de trama**

Habiendo confiado en el medio de transmisión, sabiendo sus capacidades y respetando alguna Norma que me defina las características de la señal que envío y las interfaces que uso para conectar ese medio a los equipos que quieren comunicarse, ahora debo considerar QUE información voy a enviar y en que orden para que mi receptor pueda reconocerla.

Entonces

1. ¿Como puedo hacer para que mi receptor sepa **Cuando** comenzar a leer las señales que llegan por el medio que lo vincula al transmisor?

El receptor puede saber cuándo comenzar a leer las señales mediante:

* **Delimitadores de trama:** Utilizando caracteres o secuencias especiales al inicio y fin de cada trama.
* **Bits de inicio y parada:** En transmisión asíncrona, cada carácter de datos está precedido por un bit de inicio y seguido por uno o más bits de parada, indicando al receptor cuándo comenzar y terminar de leer cada carácter.

1. ¿Cómo puede mi receptor sincronizarse con el reloj ( clock) del transmisor?¿Que técnicas o métodos se utilizan?

El receptor puede sincronizarse con el reloj del transmisor mediante:

* **Reloj compartido (Transmisión síncrona):**
  + Señalización en banda: Embebiendo el reloj en la señal de datos. Ejemplo: Manchester encoding, donde las transiciones en la señal codifican tanto el dato como el reloj.
  + Canal de reloj separado: Usar un canal adicional dedicado a la señal de reloj, manteniendo sincronizados al transmisor y al receptor.
* **Bits de inicio y parada (Transmisión asíncrona):**
  + Bits de inicio y parada: Cada carácter comienza con un bit de inicio y termina con uno o más bits de parada, permitiendo que el receptor se resincronice al inicio de cada carácter.
* **Codificación de línea:**
  + Manchester Encoding: Utiliza transiciones dentro de cada bit para codificar el dato y el reloj, asegurando que haya suficiente transición para mantener la sincronización del reloj.
* **2- Control de Flujo de datos**

Muy bien, hemos definido una técnica para que nuestros TX y RX estén sincronizados, ahora bien,

1. ¿Qué sucede si comienzo a transmitir tramas a una velocidad mayor a las que mi receptor puede procesar?

Si comienzas a transmitir tramas a una velocidad mayor de la que el receptor puede procesar, ocurren varios problemas:

* **Pérdida de datos:** El receptor no puede procesar las tramas entrantes lo suficientemente rápido, lo que provoca que las tramas se pierdan.
* **Desbordamiento de buffer:** Los buffers del receptor se llenan, y si no hay mecanismos adecuados de control de flujo, las tramas adicionales se descartarán.
* **Errores en la transmisión:** El receptor puede empezar a procesar datos de manera incorrecta, llevando a errores en la transmisión.

1. ¿Qué técnicas se usan para el control de flujo? ¿Como funcionan?

* **Stop and Wait**
  + **Funcionamiento:**
    - El transmisor envía una trama y luego espera a recibir un acuse de recibo (ACK) del receptor antes de enviar la siguiente trama.
    - Si no recibe un ACK dentro de un tiempo específico (debido a que la trama o el ACK se perdió), reenvía la trama.
  + **Ventajas:**
    - Simple de implementar.
    - Asegura que el receptor no sea sobrecargado ya que solo procesa una trama a la vez.
  + **Desventajas:**
    - Ineficiencia en el uso del canal, especialmente en canales de alta velocidad o alta latencia, ya que el transmisor está inactivo mientras espera el ACK.
* **Sliding Window (Ventana Deslizante)**
  + **Funcionamiento:**
    - El transmisor puede enviar varias tramas antes de necesitar un ACK para la primera de ellas, dependiendo del tamaño de la ventana deslizante.
    - El receptor puede enviar un ACK para varias tramas a la vez, lo que permite mayor eficiencia.
    - El número de tramas que se pueden enviar sin recibir un ACK está limitado por el tamaño de la ventana.
    - La ventana se “desliza” conforme se reciben ACKs, permitiendo el envío de nuevas tramas.
  + **Ventajas:**
    - Mayor eficiencia en el uso del canal, ya que permite la transmisión continua de tramas.
    - Mejor uso de la capacidad del canal, especialmente en canales de alta latencia y alta velocidad.
    - Reduce el tiempo de espera del transmisor y aumenta la tasa de transferencia de datos.
  + **Desventajas:**
    - Más complejo de implementar que el método Stop and Wait.
    - Requiere más memoria y procesamiento para manejar la ventana deslizante y los buffers de datos.

1. ¿Que ventajas tiene el método de control de flujo de ventana deslizante sobre el de Stop and Wait?

* **Eficiencia:** Sliding Window utiliza el canal de manera más eficiente, permitiendo que múltiples tramas sean enviadas sin esperar individualmente por un ACK. Esto es especialmente útil en redes de alta velocidad y alta latencia.
* **Mayor Throughput:** Permite un mayor rendimiento porque reduce los tiempos de inactividad del transmisor. El transmisor puede seguir enviando tramas dentro del límite de la ventana antes de necesitar un ACK.
* **Mejor manejo de latencias:** Sliding Window es más adecuado para enlaces de larga distancia donde la latencia puede ser significativa. El método permite que la transmisión continúe durante el tiempo que toma recibir un ACK.
* **Flexibilidad:** La capacidad de ajustar el tamaño de la ventana permite optimizar el control de flujo para diferentes condiciones de red.
* **3 - Detección de Errores**

Ahora bien, sabemos, como ya lo hemos dicho al estudiar medios de enlace, que **los ruidos** que se introducen en el camino, pueden afectar a nuestra información produciendo errores.

*¿Qué otra función asignaría a este nivel de enlace, ante esta situación?*

Otra de las funciones es la de **detectar** cuando una trama llega con errores al receptor.

Para ello se ha asignado otro campo bien definido dentro de la trama que esta formado por el resultado obtenido al aplicar un algoritmo entre la trama que se va a enviar y un código binario patrón que poseen tanto el transmisor como el receptor. El resultado de este algoritmo entre el **Patrón** y el mensaje se escribe en un campo asignado a tal fin en la trama denominado Frame Check Secuence o **FCS**.

Los dos métodos para la detección de errores son:

1. **Control de Paridad**

* **Funcionamiento:**
  + En el método de paridad, se añade un bit de paridad a los datos que se transmiten. Este bit puede ser de paridad par o impar.
  + Paridad par: El bit de paridad se ajusta de manera que el número total de bits 1 en la trama (incluyendo el bit de paridad) sea par.
  + Paridad impar: El bit de paridad se ajusta de manera que el número total de bits 1 en la trama sea impar.
  + Al recibir la trama, el receptor cuenta los bits 1 y verifica si coinciden con la paridad esperada.
* **Ventajas:**
  + Simple y fácil de implementar.
  + Requiere poco hardware adicional.
* **Desventajas:**
  + No detecta todos los errores. Por ejemplo, si hay un número par de bits erróneos, el error no será detectado.
  + Menor precisión en la detección de errores en comparación con CRC.

1. **CRC Cyclic Redundancy Check**

* **Funcionamiento:**
  + CRC utiliza un algoritmo más complejo basado en la teoría de polinomios.
  + Tanto el transmisor como el receptor comparten un polinomio generador predefinido.
  + El transmisor realiza una división polinómica de los datos por el polinomio generador, y el residuo de esta división se añade al final de la trama como el FCS.
  + Al recibir la trama, el receptor realiza la misma división polinómica y compara el residuo calculado con el FCS recibido.
  + Si los residuos coinciden, se considera que la trama está libre de errores; de lo contrario, se detecta un error.
* **Ventajas:**
  + Alta probabilidad de detección de errores, incluyendo errores de ráfaga.
  + Detecta errores múltiples y la mayoría de los errores de mayor complejidad.
* **Desventajas:**
  + Más complejo de implementar y requiere más procesamiento que el control de paridad.
  + Requiere más bits adicionales (el tamaño del FCS es mayor que el bit de paridad).

Si nos referimos a los dos métodos mas usados, ¿cuál considera que posee una probabilidad mas alta de detectar error en la trama?

Explique brevemente como funciona cada uno

Entre los dos métodos, CRC tiene una probabilidad mucho más alta de detectar errores en la trama en comparación con el Control de Paridad. Esto se debe a su capacidad de detectar múltiples bits erróneos y errores de ráfaga, a la complejidad de su algoritmo y al tamaño del campo de verificación.

*Ok, hasta aquí con esta técnica logramos detectar un error, pero, ¿Que hacemos ante esta situación? Veamos otra de las funciones de la capa II:*

* **4 - Control de errores**

*Bien aquí aparece otra de las funciones que debe cumplir nuestro segundo nivel y es* ***que hacer cuando el receptor detecta un error, e***sta función es llamada **CONTROL DE LOS ERRORES.**

¿Qué técnicas se emplean para realizar la función de control de errores? ¿Explique como funcionan las mismas?

Como técnicas ARQ tenemos tres:

1. **ARQ Stop and Wait:**

* **Funcionamiento:**
  + El transmisor envía una trama y espera un acuse de recibo (ACK) del receptor antes de enviar la siguiente trama.
  + Si el receptor detecta un error en la trama recibida, envía un mensaje de no acuse de recibo (NAK).
  + Si el transmisor recibe un NAK o no recibe un ACK dentro de un tiempo determinado, retransmite la trama.
* **Ventajas:**
  + Simple de implementar y entender.
  + Asegura que cada trama es correctamente recibida antes de enviar la siguiente.
* **Desventajas:**
  + Ineficiente en términos de utilización del canal, ya que el transmisor pasa mucho tiempo esperando ACKs.
  + Menor throughput en enlaces de alta latencia o alta velocidad.

1. **ARQ- Go Back to N**

* **Funcionamiento:**
  + El transmisor puede enviar varias tramas antes de recibir ACKs, hasta un número máximo de tramas (N), determinado por el tamaño de la ventana deslizante.
  + El receptor envía ACKs para cada trama recibida sin errores.
  + Si el receptor detecta una trama con error, descarta esa trama y todas las tramas siguientes hasta que reciba la trama correcta.
  + El transmisor, al no recibir ACKs para las tramas con errores, retransmite desde la trama con error (y todas las subsiguientes).
* **Ventajas:**
  + Mejor utilización del canal en comparación con Stop and Wait, ya que permite múltiples tramas en tránsito.
  + Mayor throughput en enlaces de alta velocidad y alta latencia.
* **Desventajas:**
  + Ineficiencia en términos de retransmisión, ya que puede retransmitir tramas correctamente recibidas (tramas posteriores a la que tenía error).

1. **ARQ Selective Reject.**

* **Funcionamiento:**
  + Similar a Go-Back-N, pero más eficiente en la retransmisión.
  + El transmisor puede enviar múltiples tramas antes de recibir ACKs, hasta un número máximo de tramas (N).
  + El receptor envía ACKs individuales para cada trama correctamente recibida.
  + Si el receptor detecta una trama con error, envía un NAK para esa trama específica.
  + El transmisor retransmite únicamente las tramas para las cuales ha recibido NAKs.
* **Ventajas:**
  + Mayor eficiencia en la utilización del canal, ya que sólo se retransmiten las tramas con errores.
  + Mayor throughput y mejor manejo de errores en comparación con Go-Back-N.
* **Desventajas:**
  + Más complejo de implementar debido a la necesidad de mantener un buffer y gestionar las tramas fuera de secuencia.
  + Requiere más memoria y procesamiento en el receptor.
* **5 - Direccionamiento**

Y nos queda un condimento muy importante, pensemos que sucede cuando son muchos los que se quieren comunicar empleando un mismo medio de enlace.

*¿Qué podemos hacer para reconocernos y saber quien envía y quien debe recibir la información que está en el medio compartido?*

Cuando varios dispositivos desean comunicarse a través de un mismo medio de enlace, se utilizan diversas técnicas y protocolos para asegurar una comunicación eficiente y ordenada. Cada dispositivo en la red tiene una dirección MAC única que se usa para identificar de manera exclusiva cada dispositivo en las tramas enviadas. Para gestionar el acceso al medio y evitar colisiones, se emplean protocolos como CSMA/CD, que permite que los dispositivos escuchen el medio antes de transmitir y manejen colisiones si ocurren, y CSMA/CA, que minimiza las colisiones mediante el uso de tiempos de espera aleatorios, especialmente en redes inalámbricas. Métodos como Token Passing, donde un token circula permitiendo la transmisión solo al dispositivo que lo posee, y TDMA, que asigna ranuras de tiempo específicas a cada dispositivo, evitan colisiones de manera más estructurada. Además, el protocolo ARP se usa para mapear direcciones IP a direcciones MAC, facilitando la comunicación en redes donde las direcciones IP no son estáticamente asignadas. Estas técnicas aseguran que los dispositivos se reconozcan mutuamente y gestionen de forma eficiente quién puede transmitir y recibir datos en un medio compartido.