

Tema 4: PROTOCOLOS DEL NIVEL DE ENLACE

1. *Introducción a los protocolos del nivel de enlace*
2. *Configuraciones y tipos de estaciones en un enlace de datos*
3. *Tipos de protocolos*
4. *Protocolo BSC*
5. *Protocolos XMODEM e YMODEM*
6. *Protocolo KERMIT*
7. *Protocolo HDLC*
8. *Eficiencia de los protocolos*

4.1.- Introducción a los protocolos del nivel de enlace

El nivel de Enlace o nivel 2 del modelo OSI, es el encargado de establecer una línea de comunicación, libre de errores, que pueda ser utilizada por el nivel de Red. La capa de Enlace ya no opera con bits, como hace la capa inmediatamente inferior, la capa Física,

Mientras el nivel Físico proporciona únicamente un servicio de transmisión de bits aislados, que no tienen significado por sí mismos, a través de un canal, el nivel de Enlace tiene el objetivo de hacer la comunicación fiable y libre de errores proporcionando los medios para activar, mantener y desconectar el Enlace. Por ello, la capa de Enlace trabaja con bloques fraccionados del mensaje, llamados **tramas**, constituidos por parte de la información del usuario y por información adicional que se le añade para el encaminamiento de las tramas, la recuperación de errores, etc.

En definitiva, el principal servicio proporcionado por este nivel es el de **Detección y Control de Errores** en la transmisión, tanto desde el punto de vista físico como desde el lógico.

Para ello la capa de enlace deberá ser capaz de realizar las siguientes funciones:

- **Entramado de la información.** La capa de enlace encapsula la información procedente de la capa superior en tramas de datos con un formato específico.

La función de entramado deberá habilitar un mecanismo que permita al extremo receptor determinar donde comienza y donde termina cada trama, incluyendo patrones especiales de principio y fin de trama.

- **Detección de Errores.** La capa de enlace incluye "*Mecanismos de Detección de Errores*" que permiten determinar al extremo receptor si una trama es correcta o si, por el contrario, ha sufrido alguna alteración durante la transmisión debido a ruidos u otros fenómenos adversos. La detección de errores suele llevarse a cabo mediante la inclusión de códigos de paridad o códigos de redundancia cíclica (**CRC**) en cada trama transmitida.
- **Recuperación ante fallos.** La capa de enlace se encarga de la retransmisión de todas las tramas perdidas o erróneas cuando se produce un fallo en la transmisión.

El mecanismo más habitual consiste en que el extremo receptor envía una confirmación (**ACK, acknowledge**) por cada trama correcta recibida. Si el emisor no recibe la trama

de confirmación, significa que la trama de datos correspondiente se ha perdido o ha sufrido un error en la transmisión, por lo que deberá retransmitirla.

- **Control de flujo.** Para evitar la pérdida de datos es necesario evitar que un emisor rápido pueda llegar a saturar a un receptor lento. Para ello los "*Mecanismos de Control de Flujo*" permiten regular la cantidad de información que se puede enviar al extremo receptor en función de las limitaciones de éste.
- **Gestión del enlace.** Cuya función es garantizar la transmisión administrando la utilización del medio físico.

La utilización de protocolos garantiza la compatibilidad de los distintos equipos que se quieren comunicar. Los protocolos realizan una serie de funciones, destacando, entre ellas, el establecimiento y la finalización de la comunicación, el envío de los mensajes y la detección y corrección de errores, obteniendo así un máximo rendimiento de los medios de transmisión utilizados

4.1.1. - Servicios proporcionados a la capa de red. -

El nivel de enlace de datos puede ser diseñado para ofrecer diferentes clases de servicios. La clase de servicio puede variar de un sistema a otro. Las tres posibilidades más razonables son:

- ❖ **Servicio sin conexión y sin verificación:** Consiste en hacer que la máquina fuente mande tramas a la máquina destino sin que esta última tenga que reconocerlas. No se establece ninguna conexión (o acuerdo previo) antes de la transmisión de los datos. Si una trama se pierde o queda dañada por ruido en el canal de transmisión no será misión del nivel de enlace el corregir la deficiencia.
- ❖ **Servicio sin conexión pero con verificación:** Significa que por cada trama que manda espera que le llegue un reconocimiento. De esta manera, el emisor sabe si la trama ha llegado satisfactoriamente. Si no llega el reconocimiento correspondiente pasado un tiempo determinado desde la emisión de la trama, el emisor asume que su trama no llegó o llegó dañada y la retransmite.
- ❖ **Servicio con conexión:** Es el servicio más sofisticado que el nivel de enlace de datos puede proporcionar al nivel de red. Con este servicio las máquinas fuente y destino establecen una conexión antes de transmitir los datos. Cada trama que se envía, sobre la conexión establecida, se numera y el nivel de enlace garantiza que cada trama se recibe una sola vez y que se reciben en el orden correcto. Esto no se puede garantizar con el servicio sin conexión, pues la pérdida de reconocimiento provoca que una trama pueda ser enviada varias veces y, por lo tanto, recibida otras tantas.

La configuración del enlace de datos vendrá establecida principalmente por tres características de la transmisión: el tipo de canal, el modo de transmisión y la disciplina de línea.

4.1.2. - Enmarcado o entramado. -

A fin de proporcionar servicios a la capa de red, la capa de Enlace debe usar los servicios proporcionados a ella por la capa física.

La capa física acepta un flujo de bits en bruto e intenta entregarlo en su destino, sin garantizar que este flujo de bits esté libre de errores. El número de bits recibidos puede ser menor igual o mayor que el número de bits transmitidos y, además, pueden tener diferentes valores, por lo que es responsabilidad de la capa de enlace de datos detectar y, de ser necesario, corregir los errores.

Habitualmente, la capa de enlace de datos divide el flujo de bits en marcos discretos y calcula la *suma de comprobación* (Código de Redundancia Cíclica CRC) para cada marco. Cuando un marco llega al destino, se recalcula la suma de comprobación. Si la nueva suma de comprobación calculada es distinta de la contenida en el marco, la capa de enlace sabe que se ha producido un error y toma medidas para tratarlo (por ejemplo descartando el marco que contiene el error y enviando un mensaje de notificación de error).

La unidad de intercambio de información en los protocolos de enlace de datos es, pues, la **trama** o **marco** que se define como un bloque de datos que, además, contiene información de control, empleada por el protocolo para identificar la trama.

Cuando el método de transmisión utilizado está orientado a bloques (como muchas transmisiones síncronas) la sincronización de la trama está resuelta por el propio método de transmisión. Por el contrario, al utilizar métodos de transmisión orientado a caracteres (como las transmisiones asíncronas), la sincronización de la trama debe resolverse por algún procedimiento adicional.

En otras palabras, lo que se pretende conseguir es determinar donde empieza y donde acaba una trama en el flujo de caracteres transmitido. Pueden considerarse múltiples soluciones:

❖ **Detección del final de la trama mediante un temporizador.** -

El Host destino, transcurrido un cierto tiempo sin recibir ningún carácter, supone que ha finalizado la recepción de la trama. Las redes, sin embargo, muy rara vez llegan a garantizar esta temporización, por lo que es posible que estos intervalos de tiempo lleguen a desaparecer, o bien, que se inserten intervalos durante la transmisión (por ejemplo, por control de flujo). El método es demasiado arriesgado y no suele utilizarse.

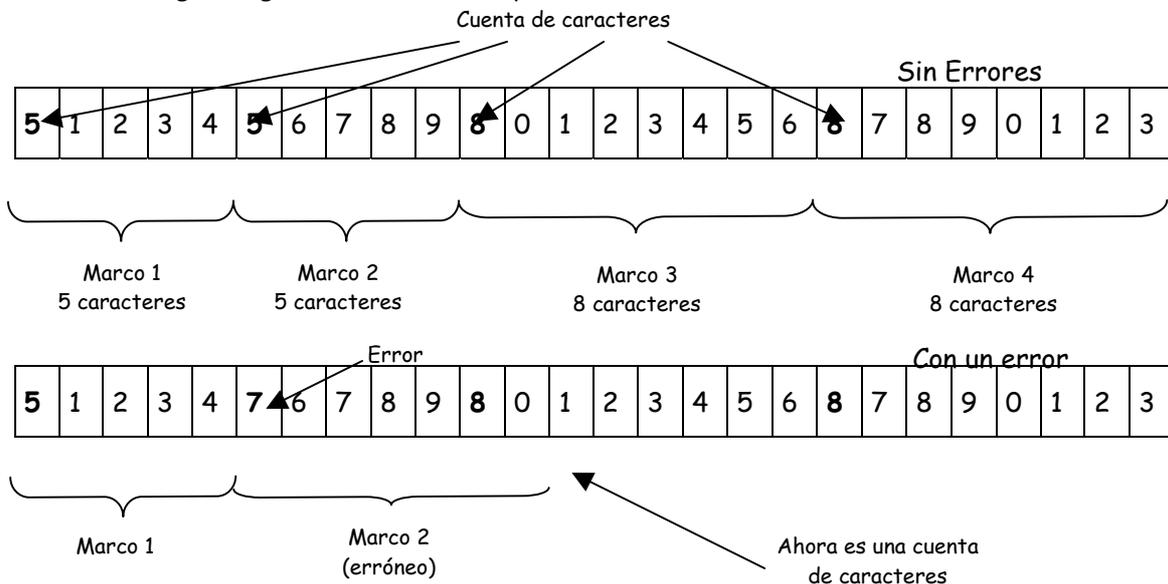
❖ **Cómputo de caracteres.** -

Con este método, se agrega un campo de cabecera para especificar el número de caracteres de la trama.

Cuando la capa de enlace del ordenador destinatario ve dicho campo, sabe el número de caracteres que le siguen y, por consiguiente, donde termina la trama. El inconveniente de este algoritmo radica en la posibilidad de que la cuenta se distorsione por error de transmisión. Aunque el código de redundancia cíclica sea incorrecto, de forma que el destinatario sepa que la trama es también incorrecta, no tiene forma de saber donde comienza la siguiente trama.

Por otro lado, la solicitud al ordenador emisor de que retransmita la trama tampoco ayuda de manera significativa, puesto que el destinatario tampoco sabe cuantos caracteres debe omitir para llegar al inicio de la retransmisión. El método de cuenta de caracteres no es bueno y se utiliza muy rara vez.

La figura siguiente ilustra este proceso



- ❖ **Utilización de caracteres especiales o secuencias de bits que indiquen el comienzo y el fin de las tramas.-**

El método, aunque bueno, plantea problemas de transparencia de datos. El añadir información de control para delimitar las tramas se emplea con frecuencia, pero presenta el problema de que las mismas combinaciones utilizadas para control podrían aparecer en la combinación de datos a transmitir, generándose un conflicto de interpretación en el receptor al detectar un falso final de trama, confundiendo, por tanto datos con secuencias de control.

Existen distintas soluciones para este problema de la **transparencia de datos** que varían según que la técnica de transmisión utilizada sea orientada a carácter u orientada a bloques.

En la transmisión orientada a caracteres puede emplearse la técnica de **relleno de carácter** (*Character Stuffing*), que se basa en insertar un carácter especial en las secuencias de control para conseguir que sean éstas irrepetibles.

Por ejemplo, si se considera la secuencia de caracteres:

dle stx dato1 dato2 dato3 dle dato4 dato5 dle etx

donde:

dle: Escape de enlace (*Data Link Escape*)

stx: Inicio del texto (*Start of Text*)

etx: Fin del texto (*End of Text*)

El **dle** subrayado indica que ha sido insertado en el procesado del marco aplicando la técnica de relleno de carácter, debido a que **dato4 = dle**. Sólo en ese caso, y para asegurar la transparencia de los datos se inserta (o duplica) cada **dle** que aparece en los datos.

Con este método, las secuencias **dle stx** y **dle etx** indican respectivamente el principio y el final de cada bloque de información y, para evitar que se repitan dentro del bloque de datos, se duplican los **dle**, con lo cual, si se desea transmitir precisamente la secuencia **dle etx** dentro de los datos, ésta se transmitirá como **dle dle etx**, con lo que el receptor no confunde la información recibida con el fin de la trama, ya que la recepción de dos **dle** seguidos le indica que no debe interpretarlo como una secuencia de control y, además, que debe eliminar uno de esos dos caracteres.

Cuando se utilizan transmisiones orientadas a bit (como por ejemplo, transmisiones síncronas), la transparencia de datos se logra por medio de la técnica de **relleno de bit** (*Bit Stuffing*) que, en esencia, trabaja bajo el mismo principio que la técnica de relleno de carácter, pero insertando un solo bit en vez de un carácter, por lo que, *a priori* es más eficiente.

El relleno de bit consiste en insertar un cero cada vez que se encuentran dentro del bloque de datos *cinco unos* seguidos. Esto se hace para evitar que se puedan generar secuencias en la transmisión que coincidan accidentalmente con el flag empleado como señalizador de comienzo y final de bloque en la transmisión síncrona, que suele ser la secuencia 0 1 1 1 1 1 0 .

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

(c) 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Bits de relleno

(a) datos originales (b) los datos, según aparece en la línea (c) los datos, como se guardan en la memoria del receptor tras eliminar el relleno

❖ Empleo de violaciones de código para señalar las condiciones de principio y final de trama. -

Estas violaciones consisten en condiciones anormales de la transmisión. Sólo se aplica en redes en las que la codificación en el medio físico contiene una cierta redundancia, por ejemplo, algunas LAN codifican un bit de datos utilizando dos bit físicos (**método de codificación Manchester**). Normalmente un bit 1 es un *par alto - bajo* y un bit 0 es un *par bajo - alto*. Las combinaciones alto - bajo y bajo - alto no se usan para datos. El esquema implica que cada bit de datos tiene una *transición a medio camino*, lo que hace fácil al receptor localizar los límites de los bits.

La utilización de combinaciones sin transición (y por tanto inválidas bajo el sistema Manchester) permite encapsular claramente las tramas y no requieren un proceso de inserción de métodos anteriores.

4.1.3. - Control de Errores. -

Habiendo solventado el problema de marcar el principio y el final de la trama el siguiente problema a tratar es como garantizar que todas las tramas lleguen a su destino en condiciones. Si el emisor se limita a mandar la trama sin preocuparse de si llega correctamente a su destino, esta estrategia puede ser adecuada para un servicio sin conexión y sin verificación, pero es totalmente inadecuada para un servicio con conexión.

El método usual de asegurar una entrega es que el emisor tenga alguna forma de realimentación que le informe de lo que está ocurriendo en el otro extremo de la línea.

Normalmente en los protocolos se plantea que el receptor mande al emisor una **trama especial de control** conteniendo un *reconocimiento positivo o negativo sobre las tramas que va recibiendo*. Si el emisor recibe un reconocimiento positivo de una trama sabe que esa trama ha llegado a destino correctamente. Por el contrario, un reconocimiento negativo indica que algo ha funcionado mal y la trama debe ser retransmitida.

Una complicación adicional es que existe la posibilidad de que *una trama se pierda completamente por problemas en el hardware* (por ejemplo, una ráfaga de ruido). En este caso el receptor no podrá reaccionar ya que no tendrá motivo para hacerlo. Debe quedar claro que en un protocolo en el que el emisor transmite la trama y entonces espera por el reconocimiento positivo o negativo podría bloquearse si la trama se pierde completamente.

Para evitar tal situación se introducen **temporizadores en el nivel de enlace**. Cuando el emisor transmite una trama arranca un temporizador. El temporizador se fija a un tiempo lo suficientemente grande para que la trama llegue a su destino, correctamente y que un reconocimiento venga del receptor. Normalmente la trama llegará correctamente al receptor y el reconocimiento volverá antes de que el temporizador finalice la cuenta, en cuyo caso se desconectará el temporizador. Sin embargo, si se pierde alguna de las tramas, la de datos o la de reconocimiento, el temporizador finalizará la cuenta, lo que avisará al emisor de que debe existir algún problema. La opción más lógica en este caso es retransmitir la trama.

Ahora bien, si las tramas pueden ser retransmitidas varias veces existe el peligro de que el receptor acepte la misma trama más de una vez. Para prevenir que esto ocurra es generalmente necesario asignar **números de secuencia** a las tramas, para que el receptor pueda distinguir las tramas retransmitidas (duplicados) de las originales.

El manejo de los temporizadores y de la numeración de las tramas, así como garantizar que cada trama entregará su contenido al nivel de red de la máquina destino una y solo una vez (sin repeticiones) es una de las tareas importantes a desarrollar por el nivel de enlace de datos.

4.1.4. - Control de flujo. -

Otra cuestión que deben tener en cuenta en este nivel es qué hacer cuando el emisor quiere transmitir tramas con una frecuencia (velocidad) superior a la que puede aceptar el receptor. Esta situación puede darse fácilmente cuando, por ejemplo, el emisor es un proceso ejecutándose en una máquina potente o poco cargada, y el receptor es un proceso en una máquina lenta.

La solución más usual es introducir un **mecanismo de control de flujo** que impida transmitir al emisor a una velocidad superior a la que puede aceptar el receptor. Este mecanismo de freno requiere generalmente algún tipo de realimentación que permita al procedimiento emisor conocer si el receptor está ya listo para aceptar otra trama.

Existen muchas posibilidades para realizar este control de flujo, pero la mayoría utilizan el mismo principio básico. El protocolo contiene reglas bien definidas acerca de cuando el emisor puede transmitir la siguiente trama. Estas reglas generalmente prohíben que se envíen tramas hasta que el receptor dé su **permiso** de forma implícita o explícita. Por ejemplo, cuando se establece una conexión el receptor puede decir:

Me puedes enviar n tramas ahora, pero después de haberlo hecho no envíes ninguna más hasta que yo te diga que continúes.

Se estudiarán diversos mecanismos de control de flujo en este tema.

4.1.5. - Gestión de enlace. -

Otra de las funciones de la capa de enlace corresponde al manejo de la gestión de enlace. Con un servicio sin conexión esta gestión es mínima, pero para el caso de un servicio orientado a conexión es más compleja. Las conexiones se deben establecer y después liberar, las secuencias de números deben iniciarse y reiniciarse en caso de que sucedan errores, y así sucesivamente.

Además, la configuración del enlace se debe administrar; en el caso más sencillo, sólo un hilo físico se extiende entre dos máquinas. Sin embargo, es muy común encontrar que varias máquinas compartan el mismo canal. Una de ellas, tradicionalmente, es la **primera** (es decir, un ordenador), en tanto que las demás son **secundarias** (es decir, terminales pasivas).

La gestión de tráfico se logra cuando la primaria transmite una trama corta, denominada **sondeo**, a la primera secundaria, preguntando si tiene alguna información que transmitir. Si es así, el terminal envía dicha información; si no, la primera sondea a la siguiente secundaria. En otros sistemas, a los terminales se les permite que envíen información al ordenador, aun en ausencia del sondeo. Por último, en otros sistemas, como por ejemplo las redes tipo **LAN**, no hay primarios ni secundarios. Todas las estaciones son iguales y tienen los mismos derechos para acceder al canal. En cualquier caso, el tema completo de los primarios y secundarios versus correspondientes, es una solución que se lleva a cabo en la capa de enlace.

4.2. - Configuraciones y tipos de estaciones en un enlace de datos

Tomando como base de partida la forma en la que se establece la comunicación entre dos nodos de la red, es decir, dos ETD, los protocolos del nivel de Enlace proponen dos modelos básicos:

- **Modelo de Estación Primaria - Secundaria o Modelo Maestra - Esclava.** En esta configuración la estación secundaria o esclava transmite datos a la red cuando es interrogada u obtiene permiso de otra estación, especialmente privilegiada en el control del enlace (estación **maestra** o **primaria**).

En terminología de protocolos de comunicaciones se dice que los protocolos que utilizan este modelo son **no balanceados**

- **Modelo igual a igual (peer to peer).** En este modelo, cada estación tiene responsabilidad tanto en el envío de datos como en el establecimiento y la liberación de los enlaces.

Los protocolos que siguen este modelo se dice que son **balanceados**.

Consecuentemente, los tipos de terminales que se consideran, a efectos del enlace son:

- ❖ **Estación Maestra / Principal / Primaria** es la encargada de enviar órdenes y recibir respuestas. Tiene la responsabilidad de ciertas funciones de control, gestión del enlace así como de toma de decisiones ante situaciones anómalas.
- ❖ **Estación Esclava / Secundaria** es la que recibe órdenes y envía respuestas. Suele tener un papel más pasivo, sin funciones de gestión de enlace.
- ❖ **Estación Combinada** es la que puede generar tanto órdenes como respuestas. Este tipo de estaciones siempre se utiliza en configuraciones punto a punto.

En cuanto al *número de interlocutores posibles* en una comunicación, se podrán establecer comunicaciones de dos formas distintas:

- ❖ **Enlace Punto a Punto.** Es aquella configuración básica en la que solo participan dos nodos.
- ❖ **Enlace Multipunto.** Aquella configuración en la que conectan varios terminales a un sistema central.

La capa de Enlace debe tener en cuenta si las transmisiones de bits se hacen de modo síncrono o asíncrono, puesto que esto determina la forma en que los bits serán señalizados en el canal de transmisión.

En la transmisión asíncrona cada carácter, precedido por un bit de *arranque* y seguido por uno de *parada*, forma un bloque informativo que permite organizar el control de la transmisión.

En la transmisión síncrona, sin embargo, no se produce delimitación entre los diferentes caracteres, por lo que hay que establecer sistemas que, en el nivel de Enlace, delimiten perfectamente los diferentes bloques de datos (**marcos** o **tramas**) y que, en el nivel físico, sincronicen los relojes del emisor y del receptor.

4.3. - Tipos de protocolos

4.3.1. - Tipos de protocolos para la función de entramado

El entramado o *framing* es la función de cualquier protocolo de enlace que determina la composición de cada trama: los campos que la compone, su tamaño, la función de cada carácter según su valor, la función de cada bit según su posición, etc.

Los protocolos pueden ser:

Protocolos orientados a carácter

Este tipo de protocolos se caracterizan por utilizar transmisiones de información codificadas, normalmente en código ASCII, donde se reservan una serie de códigos especiales - *códigos de control* - para controlar la comunicación.

Algunos de los caracteres de control mas utilizados en los protocolos orientados a carácter son:

- ACK0** Confirmación de tramas pares (*Acknowledge*)
- ACK1** Confirmación de tramas impares (*Acknowledge*)
- CAN** Cancelación de la orden anterior (*Cancel*)
- DLE** Garantía de transparencia de código (*Data Link Scape*)
- ENQ** Carácter interrogador en el modelo de estación primaria - secundaria (*Enquire*)
- EOT** Final de transmisión (*End Of Transmission*)
- ETB** Final de transmisión de bloque (*End Of Transmisión Block*)
- ETX** Final de texto (*End Of Text*)
- NAK** Confirmación negativa (*Negative Acknowledge*)
- SOH** Comienzo de cabecera (*Start Of Heading*)
- STX** Comienzo de texto (*Start Of Text*)
- SYN** Indica la inactividad del canal, carácter de sincronía (*Synchrhonous Idle*)

La característica sobre la que fundamentan este tipo de protocolos es la transparencia de código. Si el protocolo interpreta ciertos caracteres como instrucciones de control, ninguno de estos caracteres podrá aparecer como información. Así, si se emplea el código ASCII para la transmisión, no se podrán utilizar 256 caracteres diferentes para codificar la información, ya que algunos de ellos están reservados para control.

Sin embargo, la función regulada por un protocolo debe ser lo mas independiente posible de la información que transmite. Cuando esto ocurre, se dice que el protocolo es **transparente** al código utilizado. Si no se garantiza la transparencia existe la posibilidad de que los datos de usuario sean interpretados como datos de control de trama.

La transparencia se garantiza haciendo uso de reglas de interpretación especiales cuando se reciben asociaciones de caracteres de control. Por ejemplo, el carácter <ETX> se utiliza para indicar al receptor que llega el final de texto transmitido. Si en el contenido del texto aparece dicho carácter, para evitar que el receptor lo identifique como final del texto cuando realmente no ha llegado al final, se le codifica anteponiéndole el carácter

<DLE> que indica al receptor que el siguiente carácter debe interpretarse como información y no como control:



Las tramas de los protocolos orientados a carácter no tienen un formato único, son dependientes del contenido informativo de la trama. Esto produce rendimientos más bajos que otros protocolos, sin embargo, su forma de codificación es muy simple.

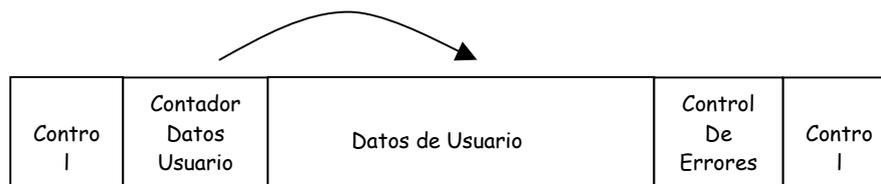
Entre los protocolos orientados a carácter mas extendidos se encuentran el **BSC** (*Binary Synchronic Control*) o bisync de IBM; el **XON/XOFF** o **ASCII** por su simplicidad; los protocolos de la familia **ARP** entre los que destacan el **XMODEM**, **YMODEM**, **ZMODEM** y Kermit en todas sus variantes.

Protocolos de bloque

Son protocolos basados en el cómputo de caracteres. Fueron creados como un intento de erradicar los problemas de dependencia de código, con el fin de facilitar la transparencia.

La utilización de este protocolo implica que el receptor no investiga el contenido de cada trama porque sabe que el final de cada trama no puede confundirse con los datos del usuario, simplemente codifica un campo que indica el número de caracteres de usuario contenidos en la trama.

El carácter siguiente al apuntado por este campo contador será un carácter de control del final de la trama. En medio quedan los datos del usuario. Todos los datos tienen, por tanto una longitud fija.



Este tipo de protocolos es muy sensible a errores, especialmente en redes digitales, puesto que éstas pueden insertar datos sobre control de tiempos durante la transmisión en las tramas de los usuarios.

Protocolos orientados a bit

Esta familia de protocolos es la que, a la postre, a prevalecido en la industria, siendo la base de la mayor parte de los protocolos de enlace que se utilizan en la actualidad, y, en particular, en Redes de Área Local.

Los protocolos orientados a bit son absolutamente independientes del código utilizado. Codifican el control de las comunicaciones por la posición y el valor de los

distintos bits que componen los campos de control de la trama. Su eficiencia es mayor que la de los protocolos orientados a carácter y, obviamente, son absolutamente transparentes.

Aunque pueden utilizar distintos tipos de tramas, siempre se reducen a un número muy limitado y nunca dependen de la información de usuario contenida en la trama. Están especialmente diseñados para funcionar en entornos **dúplex**, siendo, además, menos sensibles al ruido.

Las tramas están delimitadas por campos señalizadores (**banderas o flags**), que utilizan combinaciones de bits imposibles en cualquier otro campo de la trama utilizando, por ejemplo, la técnica de **relleno de bits o stuffing**. Esta técnica consiste en impedir que se transmitan más de cinco unos (1) seguidos. El sistema intercalará un cero (0) inmediatamente después de la secuencia de cinco unos. El receptor conoce, por tanto que la secuencia "11110" debe ser interpretada por "1111" sin considerar (descarga) el cero final.

Las banderas señalizadoras se definen por la secuencia "0111110" que es una configuración imposible en el resto de los campos de la trama, sean campos de control o de datos de usuario.

También son posibles secuencias prohibidas para señalar la **inactividad** del canal, como posteriormente se verá en el protocolo HDLC.

La mayor parte de los protocolos de la capa de enlace para redes de área local y los utilizados en las redes públicas de datos están orientados a bit.

4.3.2. - Métodos para el control de errores

Aunque las redes de comunicaciones deben garantizar la fiabilidad de los datos, debe aceptarse la posibilidad de que en una transmisión se produzcan errores, siendo inaceptable desde cualquier punto de vista el hecho de que se produzca una transmisión errónea de forma que ni el emisor ni el receptor detecten el problema.

En general, son fuentes causantes de errores:

- **Ruido térmico.**- La temperatura de un cuerpo (conductor en el caso de una red) determina el nivel de agitación de los átomos del material aumentando el estado vibracional de sus electrones, produciendo, consecuentemente colisiones entre éstos y los átomos. Estos movimientos electrónicos y estos choques generan corrientes eléctricas no controladas y aleatorias (*ruido térmico*).
- **Ruido producido por componentes electromecánicos.**- Algunos componentes electrónicos producen conmutación mecánica entre sus circuitos (por ejemplo, el caso de un timbre). Estas conmutaciones son instantáneas y generan *picos de corriente autoinducidas en los circuitos*.
- **Faltas de linealidad en los medios de transmisión.**- En la práctica, no todos los circuitos eléctricos son lineales, esto es, tienen la misma respuesta en cualquier frecuencia o velocidad de transmisión. Realmente, suele atenuarse la amplitud de una señal al transmitirse en altas frecuencias, lo que genera alteraciones no deseadas en la señal que, aunque son fácilmente controlables, deben evitarse y corregirse.

- **Cruces entre líneas.**- Cualquier corriente eléctrica genera corrientes inducidas en los conductores próximos. Cuando dos líneas de comunicaciones circulan paralelas, la información de una línea se puede inducir en otra línea produciendo un *cruce de líneas*. Para evitar estas situaciones se utilizan líneas de pares trenzados que lleven cruzadas la transmisión y la recepción, de modo que se anule el efecto de inducción de uno en el otro.
- **Falta de sincronismo.**- Si el emisor y el receptor no están convenientemente sincronizados, pueden producirse situaciones de error en la interpretación de los datos aunque la transferencia haya sido correcta.
- **Eco.**- Cuando no hay una adaptación perfecta en el *interface* entre dos líneas de transmisión, pueden producirse reflexiones de señal no deseadas que producen una interferencia con la señal original.
- **Atenuación.**- Es la pérdida de señal debida a la resistencia eléctrica del medio de transmisión o de las máquinas de la red.

De entre los métodos de corrección de errores, ya estudiados, el más utilizado es el **Código de Redundancia Cíclica (CRC)**, método comúnmente aceptado en las redes de área local.

4.3.3. - Tipos de protocolos para la función de control de flujo

El control de flujo es una función propia de los protocolos de la capa de enlace, encargada de adecuar las velocidades de transferencia de datos entre emisores rápidos y receptores lentos, y viceversa.

No todos los emisores y receptores escriben y leen tramas del canal con la misma eficiencia. En ocasiones es preciso paralizar la transmisión de tramas hasta que el receptor más lento procese las tramas anteriormente recibidas.

Para realizar esta función, los nodos de la red deben disponer de mecanismos (protocolos) por los que se establecen diálogos con los que se eviten pérdidas de datos, no ya por error, sino por inadecuación entre las características del emisor y del receptor.

Estos protocolos son los siguientes:

Envío y Espera

Un modo muy simple de controlar el flujo consiste en, cuando el receptor reciba una trama, tenga capacidad de paralizar, por unos instantes, la transmisión, como consecuencia de que el protocolo acordado por ambos interlocutores tenga previsto que el emisor no envíe una nueva trama mientras no reciba la confirmación de la trama anterior procedente del receptor. Los protocolos que utilizan la técnica de envío y espera cumplen con la función de control de flujo prevista en la capa de enlace.

Superposición

Si el enlace es semidúplex los protocolos de envío y espera ofrecen una operativa de comunicación sencilla. Los problemas aparecen en entornos dúplex y, especialmente

cuando además se posee un único canal que debe transportar tanto las tramas de datos como las de confirmaciones

En comunicaciones bidireccionales se puede utilizar la técnica de **superposición** o *piggybacking* que consiste, básicamente, en utilizar las propias tramas de datos para enviar confirmaciones.

Dos nodos A y B están intercambiando tramas de datos por un único canal dúplex. El nodo A enviará una trama de datos al nodo B. Cuando la información llega al nivel de enlace del nodo B, éste pasa el paquete que contiene la trama a su capa de red y genera una confirmación positiva o negativa para el emisor, dependiendo del estado en que llegaron los datos. Estas confirmaciones pueden enviarse por separado, como en el método de envío y espera, o bien se puede hacer que se envíen como un suplemento a otra trama de datos en que el nodo B envíe información al nodo A. De esta manera, cada trama de datos lleva información de usuario y la confirmación de la trama anterior.

El sistema ofrece un problema en el caso de que el nodo B no tuviera ya mas datos que transmitir al nodo A. El nodo A quedaría a la espera de una confirmación que no le llegará. El problema se soluciona generando una trama especial de confirmación sin contenido informativo. Para ello se requiere el disparo de un temporizador que generará esta trama si transcurre un tiempo previamente determinado sin que el nodo B tenga necesidad de transmitir al nodo A.

Protocolos de ventana deslizante

El conjunto de protocolos de **ventana deslizante** o *sliding window* tienen las siguientes características:

- Cada una de las tramas de salida está numerada secuencialmente, de modo que queden unívocamente identificadas. Supuestos adecuadamente elegidos los parámetros de la transmisión, cuanto mayor sea el número de bits de la secuencia tanto mayor será el rendimiento del protocolo en ausencia de errores.
- En cualquier instante de la transmisión, el emisor posee una lista con los números de trama que ha enviado al receptor. Todas las tramas constituyen lo que se denomina **ventana emisora**.
- Análogamente, el receptor posee una lista con las tramas que está dispuesto a aceptar procedentes del emisor, es decir, una **ventana receptora**. El tamaño de la ventana emisora del receptor no tiene por qué coincidir con el tamaño de la ventana receptora del receptor.
- El receptor aceptará la trama procedente del emisor si cae dentro de su ventana receptora; sin embargo, no importa el orden en que las tramas son enviadas, lo que proporciona mas libertad a la capa de enlace para realizar la gestión de las tramas.
- Una vez iniciada la transmisión, la ventana emisora contiene los números de trama que se han enviado y de las que todavía no se ha recibido confirmación. Cuando la capa de red suministra un paquete a la capa de enlace, ésta lo encapsula en una o varias tramas a las que les asigna los números de secuencia siguientes y son puestos en la ventana

emisora, se envían al receptor y se espera en este estado hasta que lleguen las confirmaciones.

- Cada trama es mantenida en un buffer de memoria en espera de la confirmación. Si ésta es positiva, se libera el buffer, pero si es negativa o no se recibe ninguna confirmación, se efectúa una retransmisión. La ventana emisora contiene el máximo número de tramas que pueden ser enviadas al receptor sin necesidad de confirmación. Una vez se ha superado ese número, el proceso de envío se detiene en espera de alguna confirmación que libere alguna trama enviada y pueda ser repuesta por un nuevo envío.
- De la misma forma, el receptor mantiene su ventana receptora con los números de secuencia que es capaz de recibir. Si recibe una trama con un número de secuencia que no está en la ventana receptora la descarta sin más. Como el emisor nunca recibirá una confirmación de esta trama descartada, se tendrá que ocupar más tarde de retransmitirla, una vez hayan vencido sus temporizadores. Sólo las tramas recibidas dentro de su ventana receptora son pasadas a la capa de red, generando una confirmación que liberará una o más tramas de la ventana emisora en el emisor, lo que dará a éste permiso para nuevos envíos.
- La ventana del receptor siempre tiene un tamaño constante a diferencia de la del emisor, que puede ir creciendo paulatinamente hasta llegar a un máximo, fijado por el protocolo.

El protocolo de envío y espera es un protocolo de ventana deslizante en el que tanto el tamaño de la ventana emisora como el tamaño de la ventana receptora es uno. Así cuando el emisor envía la única trama que puede enviar debe pararse hasta que el receptor le envíe la confirmación de que le llegó la trama. El emisor libera entonces la trama enviada y da paso a la siguiente.

Por su parte, el receptor puede recibir sólo una trama, concretamente la siguiente a la anterior en su número de secuencia. Si no recibe ésta, entiende que ha habido alguna trama que se ha perdido y no confirma la trama afirmativamente, con lo que el emisor la retransmitirá. Si la trama llegó correctamente, genera una confirmación afirmativa para liberar de su espera al emisor y activar la transmisión de la siguiente trama.

Utilizando los protocolos de ventana deslizante, las confirmaciones se pueden enviar una a una por cada trama recibida, o por conjuntos de tramas.

Protocolos de petición automática de respuesta

Los protocolos de **petición automática de respuesta** (*Automatic Request for Repeat*) o protocolos **ARQ** esencialmente se caracterizan porque solicitan retransmisiones de una o varias tramas de un modo automático cuando se producen errores o pérdidas de información sin intervención de agentes externos al nivel de enlace. Así, los métodos de *envío y espera* o de *envío continuo* no son más que dos modalidades distintas de técnicas ARQ.

Ejemplos de protocolos ARQ son XMODEM, YMODEM, ZMODEM, Kermit, MODEM7, TELINK, etc.

Protocolos hardware y software

Al multiplexar por la misma línea los datos del usuario y la información de control, se hace necesario el establecimiento de normas muy estrictas para el establecimiento de la comunicación, normas que establece el software de la capa de red.

Sin embargo, es posible controlar la comunicación a través de hardware. Para ello es preciso que la comunicación opere en los canales en los que pueda distinguir qué es información de usuario y qué es información de control.

El ejemplo mas sencillo de protocolo *hardware* es el protocolo **DTR/DSR** basado en las características de procedimiento de la norma *RS-232* o especificaciones similares. Según este protocolo, las líneas de datos TD y RD sólo envían información de usuario cuando el estado de los circuitos de control (DTR, DSR, CTS, RQS, CD, etc.) lo permiten, siempre de acuerdo con las especificaciones del protocolo.

En este caso, la multiplexación necesaria para distinguir las señales de datos de las de control se consigue utilizando distintas líneas para datos y para el control.

También se simplifican al máximo las comunicaciones utilizando protocolos poco exigentes y fáciles de implementar. El protocolo mas utilizado es **XON/XOFF**, que toma se nombre de dos caracteres de control del código ASCII y utiliza exclusivamente dos líneas de comunicación: la línea **TD** y la línea **RD**.

Es un protocolo no transparente que se utiliza frecuentemente para transmisiones de textos ASCII. Los caracteres son enviados por la línea serie secuencialmente y de modo asíncrono, con o sin paridad.

Las tramas en este protocolo están constituidas por cada carácter transmitido. El control de flujo se realiza enviando un carácter especial de control: así, el envío de un XOFF del receptor a emisor informa al emisor de que el receptor no está preparado para la recepción de ningún carácter mas. Cuando el receptor envía un XON al emisor, libera el bloqueo y continúa la transmisión.

Tanto el protocolo DTR/DSR como el protocolo XON/XOFF son ampliamente utilizados en las comunicaciones serie en ordenadores personales, por ejemplo, en la conexión de módems a los puestos serie del PC, etc.

4.3.4. - Tipos de protocolos para la función de acceso al medio

El control de acceso al medio es una de las grandes funciones de la capa de enlace. De hecho, la mayor parte de las arquitecturas de red con referencia en el modelo OSI descomponen el nivel 2 en una subcapa inferior de acceso al medio (MAC) y otra subcapa superior que se encarga del gobierno de la comunicación.

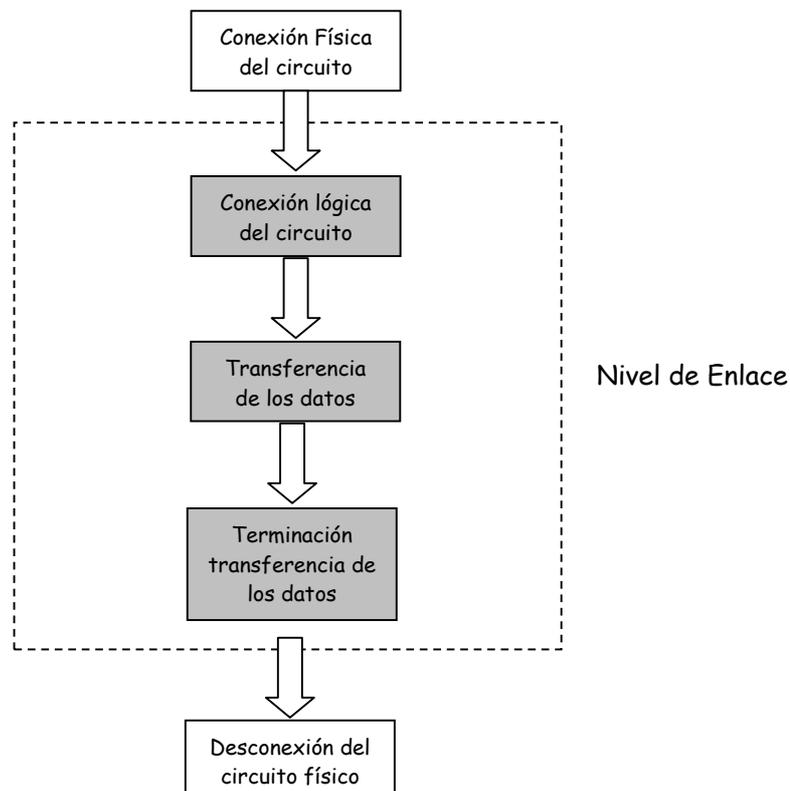
Los principales métodos de acceso al medio que se utilizan en redes de área local son el CSMA, el paso de testigo y la multiplexación en tiempo o en frecuencia, como ya se ha visto anteriormente.

4.3.5. - Fases de un protocolo del nivel de enlace de datos

Como resumen, para la realización de una transmisión entre dos estaciones a través del enlace de datos, los protocolos pasan por las siguientes fases:

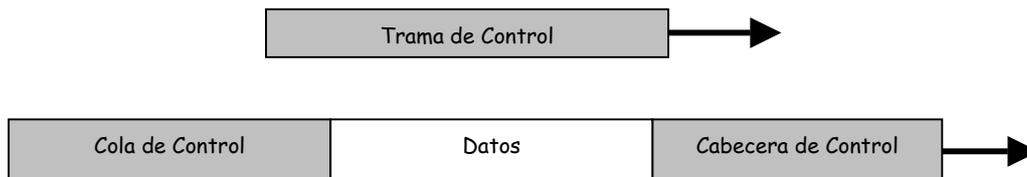
- ❖ **Conexión física del circuito.**- Constituye esta fase todo el conjunto de operaciones necesarias para el establecimiento del circuito físico que une las dos estaciones que desean comunicarse
- ❖ **Conexión lógica del circuito.**- Esta fase se compone del conjunto de procesos necesarios para la preparación de la transmisión de la información a través del circuito físico ya establecido, comprobándose que ambas estaciones se encuentran en disposición de empezar.
- ❖ **Transferencia de datos.**- Esta fase está compuesta por los procesos necesarios para que el conjunto de tramas de información llegue desde la estación emisora a la receptora sin ningún tipo de problemas y sin la aparición de errores.
- ❖ **Terminación de la transferencia de datos.**- Esta fase comprende los procedimientos de liberación de los elementos que han intervenido en la transmisión de los datos.
- ❖ **Desconexión del circuito físico.**- Constituye esta fase el conjunto de operaciones necesarias para la desconexión del circuito físico entre las dos estaciones.

En la figura adjunta se muestran las fases en una comunicación al nivel de enlace de datos. En ella puede observarse que tanto la conexión física como la desconexión física no pertenecen al nivel de enlace de datos.



También como resumen, cabe recordar que la unidad de datos de este nivel, se llama trama o marco, existiendo dos tipos diferentes:

- **Trama de datos.**- Es la unidad que se transmite. Contiene un elemento de información acompañado por una *cabecera* y una *cola de control*
- **Trama de control.**- Son las unidades de información que se envían las estaciones para el control y coordinación del establecimiento y terminación del enlace de datos



4.4.- Protocolo BSC o BISYNC

BSC son las siglas de *Binary Synchronous Control* del protocolo ideado por IBM como primer protocolo de propósito general, utilizado para conectar equipos en la configuración punto a punto o multipunto en canales conmutados o sin conmutación.

Es un protocolo para transmisiones semidúplex, sensible al código y orientado a carácter. Se utiliza frecuentemente para el sondeo de terminales remotos. El protocolo original de IBM soporta los códigos ASCII, EBCDIC y un código especial de IBM.

El formato de la trama BSC es variable puesto que es un protocolo no transparente. Cuando los datos de usuario contienen caracteres que coinciden con los caracteres de control del propio protocolo, es necesario colocar caracteres de control especiales para evitar que sean mal interpretados (relleno de caracteres). El enlace BSC opera de dos modos distintos:

- **Modo control.**- Es utilizado por la estación primaria para efectuar operaciones de control de enlace, tales como interrogar de modo rotatorio a las estaciones secundarias sobre sus necesidades de transmisión. A este sistema de preguntas se le llama *polling*.
- **Modo texto o mensaje.**- Es utilizado por cualquier estación para efectuar la transmisión de sus bloques de datos.

Cuando una estación secundaria que desea transmitir recibe la invitación para efectuar la transmisión, envía un STX o un SOH como cabecera de la trama. Este envío pone al enlace en el modo texto. Posteriormente se intercambian datos hasta que se recibe un EOT que produce el cambio al modo control. Cualquier intercambio de datos se produce exclusivamente entre dos estaciones, permaneciendo inactivas las estaciones restantes.

S	S	S	Cabecera	S	Datos	ETB	Control de
Y	Y	O		T		o	Errores
N	N	H		X		ETX	

Para controlar la posible pérdida de tramas, BSC utiliza los caracteres de confirmación de tramas ACK0 y ACK1, uno para tramas pares y el otro para tramas impares.

En la figura anterior se han representados algunos campos típicos de tramas BSC: El carácter SOH abre la trama y deja el enlace a modo texto; el campo de cabecera va encerrado entre un carácter SOH y un STX. Este campo no está definido por el protocolo dependiendo de la red en la que se esté utilizando. Posteriormente viene un campo de datos de usuario, que acaba con un ETB si es el final del bloque pero no el último bloque de la transmisión y ETX si ya se envió el último bloque. La trama finaliza con un campo de control de errores del tipo CRC.

4.5. -Protocolos XMODEM e YMODEM

En la necesidad de transferir ficheros entre ordenadores centrales y ordenadores personales a través de líneas telefónicas está el origen de este protocolo que en su día alcanzó una gran difusión y ha hecho que se haya implantado como un estándar de facto.

XMODEM es un protocolo orientado a carácter, de tipo ARQ, dependiente del código en que se expresen los datos, para entornos semidúplex y con un sistema de envío y espera para la corrección de errores.

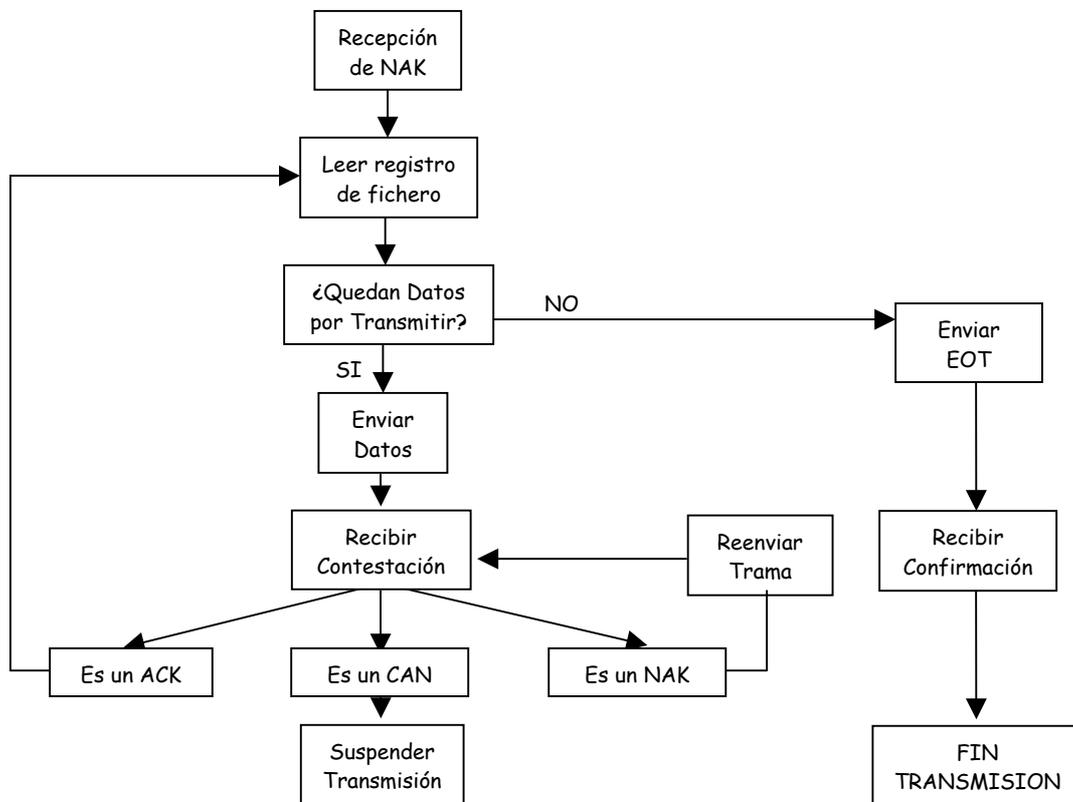
Las tramas del protocolo XMODEM tiene una longitud de datos fija de 128 bytes; los otros campos que posee la trama son de 1 byte cada uno.

Cabecera (SOH)	Número de Secuencia	C-1 Secuencia	Datos de Usuario (128 bytes)	Control de Errores
----------------	---------------------	---------------	------------------------------	--------------------

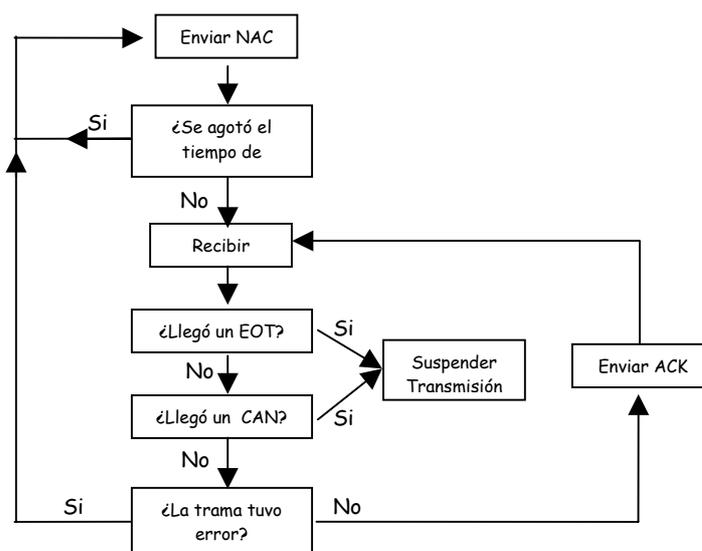
Los campos que componen una trama XMODEM son los siguientes:

- **Cabecera.**- Consiste en el envío de un carácter SOH (*Start Of Header*) del código ASCII. Actúa de bandera señalizadora de comienzo de la trama.
- **Número de Secuencia de Trama.**- Este campo es un contador que numera las tramas, de modo que la primera trama transmitida lleva el valor 1 y que todas las tramas se identifiquen unívocamente hasta que se de la vuelta al contador. Si una trama se pierde, se detectará en el receptor puesto que llegará una trama con un número de secuencia superior a la que se esperaba.
- **Complemento a 1 de la Secuencia de Trama.**- Este campo se calcula como el complemento a 1 del campo de secuencia de trama. Cuando el receptor recibe una trama, calcula la operación lógica XOR (comparación bit a bit de dos secuencias de bits. Da como resultado, para cada bit, un 0 si los bits son iguales y un 1 si son diferentes) de los campos secuencia de trama y complemento a 1 de secuencia de trama. Si el resultado es 0, implica que la secuencia es correcta. Si la recepción no es correcta, XMODEM enviará al emisor un NAK para informarle que debe retransmitir la trama.
- **Datos de usuario.**- Este campo codifica los datos con una longitud fija de 128 bytes.
- **Control de errores.**- Este campo es un código de paridad de bloque o *checksum* calculado sobre los datos de usuario.

Las transmisiones comienzan con el envío de un NAK, a partir de aquí se envían las sucesivas tramas de datos y sus confirmaciones. Las retransmisiones por error se efectúan del siguiente modo:



Cuando una trama llega a su destino correctamente, el receptor envía al emisor una trama con un ACK; si no es así, lo que envía es un NAK solicitando la retransmisión de la trama, también es posible enviar el código ASCII "CAN", que significa cancelar la operación solicitada anteriormente.



El esquema de la recepción es complementario del de emisión, sin embargo, intervienen *temporizadores* que determinan nuevas posibilidades de errores de comunicación o bien la inactividad del canal.

El receptor debe recalcular el *checksum* de los datos de usuario de la trama inmediatamente recibida y lo compara con el *checksum* recibido. Si éstos son distintos o bien la secuencia de trama no es la esperada, se procede a reenviar la trama.

Existen versiones de XMODEM que utilizan como checksum el código de redundancia cíclica (XMODEM-CRC)

Por su parte, el protocolo **YMODEM** es un protocolo muy semejante a XMODEM al que se le añaden algunas características que lo mejoran: Se puede utilizar para enviar múltiples ficheros codificando la separación entre ficheros con un sistema de cabeceras, empleando códigos ASCII. Normalmente el campo de datos de usuario es de 1 Kbyte, aunque es posible codificar campos de 128 bytes (como XMODEM) mediante caracteres. Si en vez de enviar un SOH se enviara un STX se entendería que el campo de datos es de 1 Kbyte, lo que lo hace compatible con XMODEM.

YMODEM siempre utiliza como código de errores el código de Redundancia Cíclica de 16 bits.

4.6. -Protocolo KERMIT

En la actualidad es uno de los protocolos mas extendidos en la transferencia de ficheros por línea serie.

Como está orientado a la transferencia de ficheros, Kermit debe encapsular cada fichero con unas *cabeceras de principio y de final de fichero*. En la cabecera de principio de fichero se codifica el nombre del fichero que se empezará a transmitir:

Cabecera (SOH)	Longitud resto de la trama	Número de secuencia	Tipo de Trama	Datos de usuario	Control de Errores
----------------	----------------------------	---------------------	---------------	------------------	--------------------

Además permite transferencias binarias, por lo que debe tener en cuenta el contenido de los datos de los usuarios para garantizar la transparencia del protocolo.

La operativa del protocolo Kermit es semejante a la de XMODEM.

La trama del protocolo Kermit consta de los siguientes campos:

- **Cabecera.**- Contiene el carácter SOH del código ASCII como en muchos otros protocolos. Indica el comienzo de la trama.
- **Longitud.**- Codifica el número de bytes que restan en la trama, de modo que el receptor pueda conocer con antelación al final de la trama lo que debe esperar en su lectura.
- **Número de secuencia.**- Codifica de 0 a 63 la sucesión de tramas que el emisor envía al receptor, de modo semejante al protocolo XMODEM.

- **Tipo de trama.**- Especifica la función de la trama, esto es: la trama puede hacer la función de ACK, NAK, EOT, EOF, Datos, etc.
- **Datos de usuario.**- Contiene los datos de usuario si el tipo de trama los necesita. En algunos casos puede estar vacío.
- **Copntrol de errores.**- Puede estar construido tanto por un cheksum como por un código de redundancia cíclica.

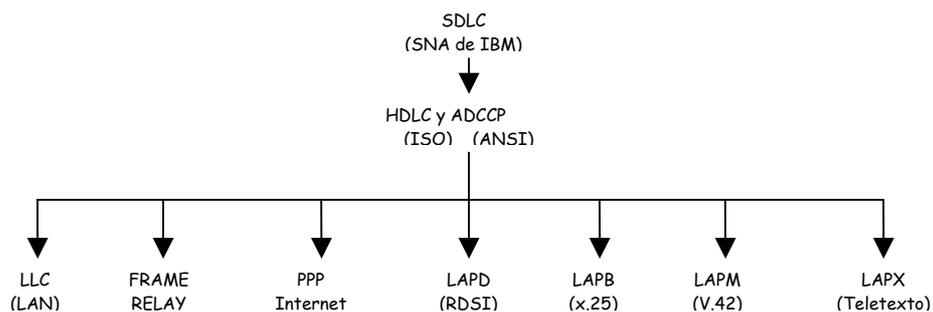
Las últimas versiones de Kermit han extendido la longitud de las tramas, codificando nuevos campos, hasta llegar a 9 kbytes, con lo que se consigue una eficiencia mayor en el rendimiento del protocolo.

4.7.-Protocolo HDLC

El protocolo **HDLC** o **Protocolo de Alto nivel del Enlace de Datos** (*High Level Data Link Control*) es un protocolo propuesto por OSI, basado en el protocolo SDLC (*Synchronous Data Link Control Protocol* o **Protocolo de Control Síncrono de Enlace de Datos**), y tomado de la capa de enlace de la arquitectura SNA de IBM.

Después de desarrollar SDLC, IBM lo propuso a la ANSI como estándar en Estados Unidos y a la ISO para su aceptación internacional. ANSI tomó las especificaciones del SDLC y construyó las suyas propias con el nombre de ADCCP (*Advanced Data Communication Control Procedure* o **Procedimiento para el Control Avanzado de Comunicación de Datos**). Por su parte, la ISO propuso el protocolo HDLC.

Posteriormentete otras organizaciones de estándares han dado su propia versión, acomodado a las aplicaciones concretas para las que se desea utilizar. Esta es la razón por la que HDLC es un protocolo, derivado del SDLC, del que dimanan otros muchos protocolos:



Todos los protocolos se basan en el mismo principio. Todos están orientados a bits y utilizan la técnica de relleno de bits para lograr la transparencia de los datos

4.7.1. - Modos de operación

El protocolo HDLC propone tres modos de funcionamiento o de operaciones posibles para la interconexión entre el emisor y el receptor:

- **Modo NRM.-** (*Normal Response Mode*) . El modo de respuesta normal requiere que la estación secundaria reciba permiso explícito de la primaria para que pueda comenzar la transferencia de datos. La transmisión puede constar de una o mas tramas. La comunicación siempre se efectúa entre una estación secundaria y una estación primaria, nunca entre dos estaciones secundarias. Una vez se ha transmitido la última trama, la estación secundaria debe esperar a la concesión de otro permiso para volver a transmitir. El modo NRM es ampliamente utilizado en las conexiones multipunto.
- **Modo ARM.-** (*Asynchronous Response Mode*). El modo de respuesta asíncrona permite a la estación secundaria comenzar una transmisión de datos sin recibir explícitamente el permiso de una estación primaria. Sin embargo, la estación primaria tiene toda la responsabilidad sobre el control, la corrección de errores y la conexión y desconexión del enlace. El modo ARM es muy poco utilizado en la actualidad.
- **Modo ABM.-** (*Asynchronous Balanced Mode*). El modo asíncrono balanceado utiliza para sus transmisiones dos estaciones equivalentes, no hay ninguna privilegiada, es propio de las comunicaciones de igual a igual y, por lo tanto, es el modo mas utilizado en Redes de Área Local.

Hay que aclarar que, cuando se utiliza el término *asíncrono* para catalogar un modo de transmisión no se refiere al modo de transmisión físico estudiado anteriormente en comunicaciones síncronas y asíncronas, mas bien , los modos asíncronos para las tramas se refieren al hecho de que la estación transmisora *no necesita permiso previo de otra estación para iniciar la comunicación*.

4.7.2. - Formato de las tramas

HDLC utiliza una trama semejante a la de cualquier protocolo orientado a bit. Todas las tramas deben comenzar y acabar por una bandera (*flag*) señalizadora de 8 bits, en concreto, se trata de la secuencia "0 1 1 1 1 1 0". Entre estas dos banderas se encierra la trama:

Bandera 0 1 1 1 1 1 0	Dirección	Control	Información	Suma de Comprobación	Bandera 0 1 1 1 1 1 0
Bits 8	8	8	0	16	8

Los campos de la trama son los siguientes:

- **Dirección.-** Este campo codifica con 1 byte o múltiplos de 1 byte la dirección del terminal, identificando si se trata de la estación primaria o la secundaria y, por tanto, si es transmisión o recepción. En líneas punto a punto a veces se usa para distinguir los comandos de las respuestas.
- **Control.-** El campo de control contiene los datos necesarios para el control del enlace, por ejemplo, las peticiones y las respuestas, los números de secuencia de tramas para las ventanas del emisor y del receptor, etc.

- **Información.**- El campo de información contiene los datos del usuario. Puede ser de longitud arbitraria, aunque la eficiencia de la suma de comprobación decae al aumentar el tamaño del marco, debido a la mayor probabilidad de errores en la ráfaga. No todas las tramas tienen este campo, dependerá de las funciones que la trama tenga asociadas, lo que se indica en el campo de control de la trama.
- **Suma de Comprobación (FCS Frame Check Sequence).**- Este campo es el de control de errores. Utiliza el método de redundancia cíclica para la detección de los posibles errores de la transmisión, usando CRC-CCITT como generador de polinomios.

El marco está delimitado por banderas o secuencias de indicación (0 1 1 1 1 1 0). En líneas punto a punto inactivas se transmiten secuencias de indicación continuamente.

La trama mínima contiene tres campos y está formada por 32 bits excluyendo los indicadores de ambos lados.

La transparencia de código se establece no permitiendo en toda la trama secuencias de más de cinco 1 seguidos aplicando la técnica de relleno de bits. La bandera es detectada por los receptores precisamente por violar esta regla, puesto que consta de seis 1 consecutivos.

4.7.3. -El campo de control en HDLC

Hay tres tipos de tramas definidas en el protocolo HDLC: de **información**, de **supervisión** y **no enumerados**. El contenido del campo de control para estos tres tipos de tramas se indica en la figura

Bits	1	3	1	3
a)	0	Secuencia	S/F	Siguiente
b)	1	0	Tipo	S/F Siguiente
c)	1	1	Tipo	S/F Modificador

a.- Tramas informativas

La trama informativa o de transferencia de información se utiliza para producir el transporte de datos de usuario entre dos dispositivos terminales de la red. Las tramas pueden ser confirmadas en destino, además, pueden tener otras funciones añadidas, como es la de interrogar (*sondeo* o *polling*) a otras estaciones sobre sus necesidades de comunicación.

Las estaciones de la red reconocen una trama informativa fijándose en la estructura del campo de control, que comienza con un "0". Posteriormente se codifica el **número de secuencia** de la ventana del emisor.

El protocolo emplea una ventana deslizante, con un número de secuencia de tres bits. En cualquier momento pueden estar pendientes hasta siete marcos no reconocidos. El campo **secuencia** de las tramas informativas es el número de secuencia de la trama. Se utiliza para controlar que la secuencia de las tramas emitidas es correcta en el receptor.

El bit **S/F** significa *Sondeo/Final*. Se usa cuando un terminal (o un concentrador) está sondeando un grupo de terminales. Cuando se utiliza como **S**, la máquina está pidiendo al terminal enviar datos. Todos los marcos enviados por el terminal, excepto el último, tiene el bit S/F puesto a S. El último se pone a F.

En algunos protocolos el bit S/F sirve para obligar a otra máquina a enviar de inmediato un *marco de supervisión*, en lugar de esperar al tráfico de retorno para incorporar la información de la ventana.

El campo **siguiente** (o **Secuencia del receptor**) es un acuse de recibo incorporado. Sin embargo, todos los protocolos usan la convención de que, en lugar de incorporar el último marco recibido correctamente, utilizan el número del primer marco no recibido (esto es, el siguiente marco esperado). La decisión de utilizar la última trama recibida o el siguiente marco esperado es arbitraria; pero debe usarse con consistencia.

Es un método es muy cómodo para detectar pérdidas de tramas.

b.- Tramas de supervisión

Las tramas supervisoras efectúan funciones de control tales como bloqueos y desbloqueos de las transmisiones, la confirmación de las tramas recibidas, las peticiones de retransmisión, etc.

Los diferentes marcos de supervisión se distinguen por el campo **tipo**, que se codifica con dos bits. Pueden haber, por tanto, cuatro tipos distintos de tramas supervisoras:

- **Tipo 0 o Receive Ready (Receptor Listo)**.- Es un marco de acuse de recibo que sirve para indicar que se está a la espera de la siguiente trama esperada. Este marco se envía cuando, en el método de superposición de confirmación de tramas, el receptor no necesita enviar tramas de datos al emisor con las que incluir la confirmación, necesitándose una trama específica para ello: La trama de supervisión de tipo 0.
- **Tipo 1 o Reject (Rechazo)**.- Es una trama de acuse negativo y sirve para indicar que se ha detectado un error de transmisión, bien porque se ha recibido una trama con error, o bien porque no se ha recibido una trama esperada y ha vencido el temporizador.. En esta trama. El campo **siguiente** indica el primer marco en la secuencia que no se ha recibido correctamente (es decir, el marco a retransmitir). Se pide entonces al emisor retransmitir todos los marcos pendientes comenzando por **siguiente**.

- **Tipo 2 o Receive Not Ready (Receptor no Listo).**- Esta trama supervisora indica que el receptor no está listo para recibir. Sirve para indicar al emisor que suspenda la transmisión momentáneamente mientras se resuelven los problemas que originaron la parada del receptor. Reconoce todos los marcos hasta, pero sin incluir, *siguiente* al igual que Receive Ready, pero le dice al emisor que detenga el envío. El propósito de Receive Not Ready es señalar ciertos problemas temporales del receptor, como falta de *buffers*, pero no servir como alternativa del control de flujo de ventana deslizante. Cuando el problema se resuelve, el receptor envía el Receive Ready, Rejetc o ciertos marcos de control.
- **Tipo 3 o Selective Reject (Rechazo Selectivo).**- Con este tipo de trama supervisora se solicita al emisor que reenvíe una trama concreta, es decir, una trama que utiliza la técnica del rechazo selectivo

c. -Tramas no enumeradas

Las tramas no enumeradas también tiene funciones de control aunque también puede servir para llevar datos cuando se solicita un servicio no confiable sin conexión. . Los diferentes protocolos orientados a bits tienen diferencias considerables en este apartado, en contraste con los otros dos tipos, donde son casi idénticos. Hay cinco bits para indicar el tipo de trama enviado , pero no se usan las 32 (2^5) posibilidades.

De los cinco bits utilizados, dos se usan para el código de trama y tres mas para el modificador. Normalmente el protocolo no suele agotar todas.

Las tramas no enumeradas pueden clasificarse del siguiente modo:

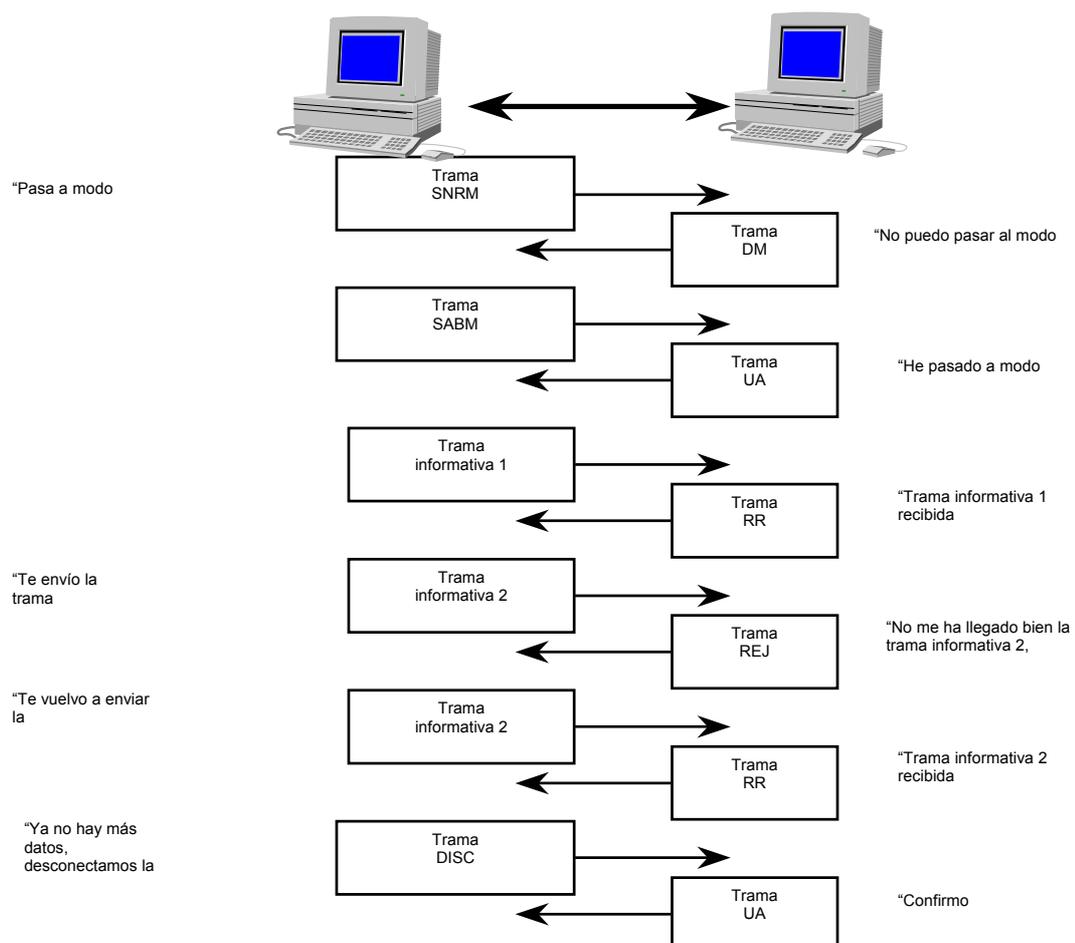
- **Tramas que seleccionan el modo de operación.**- Sirven para informar al receptor del modo de operación en que debe situarse para efectuar la conmutación. La trama **SNRM** (*Set Normal Response Mode*) se utiliza para dejar al receptor en modo **NRM**. La trama **SARM** (*Set Asynchronous Response Mode*) pone al redeptor en modo **ARM**; por último, la trama **SABM** (*Set Asynchronous Balanced Mode*) lo pone en modo **ABM**.
- **Tramas que ordenan la desconexión.**- Para este fin se utiliza la trama **DIS** (*Disconnect*). Informa al receptor de que se producirá una desconexión temporal del equipo.
- **Confirmación de trama no numerada.**- La trama UA (*Unnumbered Acknowledge*) se utiliza para que el receptor avise al emisor de que recibió una trama no numerada que ha sido aceptada, esto es, representa la confirmación de una trama no numerada.
- **Confirmación negativa de la selección de modo de operación.**- Cuando una trama de selección de modo de operación (tramas SNRM, SARM o SABM) llega al receptor, éste comprueba si es posible o no ponerse en el modo indicado en la trama. Si puede ponerse, se envía una trama UA, pero si no lo puede hacer envía una trama **DM** (*Disconnected Mode*).
- **Confirmación de rechazo de trama.**- La trama **FRMR** (*Frame Reject*) indica que ha llegado una trama correcta desde el punto de vista del chequeo de errores, pero sintácticamente incorrecta.

4.7.4.-Fases del protocolo HDLC y ejemplo de funcionamiento

La comunicación de datos en el nivel de enlace entre dos estaciones, utilizando el protocolo HDLC debe seguir las siguientes fases:

- **Conexión del circuito.**- Esta fase presupone la conexión física a través de un medio de transmisión, propia del nivel físico. Consiste en el establecimiento de una conexión lógica entre el emisor y el receptor en el que se ponen de acuerdo en los parámetros de la comunicación, ponen a cero las variables de estado de comunicación , ajustan sus ventanas de comunicación, etc.
- **Transporte de datos.**- En esta fase se produce el intercambio de datos entre el emisor y el receptor, incluyendo tramas informativas, de supervisión y no numeradas que regulan la comunicación.
- **Desconexión del circuito.**- Una vez concluida la transmisión de datos, debe procederse a la liberación de los recursos utilizados para la comunicación en el emisor y en el receptor. Para ello se utiliza la trama de desconexión DIS.

Gráficamente, la comunicación entre dos estaciones a través del protocolo HDLC se simula en la figura adjunta:



4.8.- Eficiencia de un protocolo

La eficiencia en un protocolo es el porcentaje de tiempo en el que el transmisor está enviando tramas al receptor. Se define como el cociente:

$$U_p = \frac{T_t}{T_c}$$

donde T_t es el tiempo utilizado para transmitir íntegramente una trama (Tiempo de Transmisión de la trama) y T_c es el Tiempo de Ciclo asignado a la transmisión de la trama.

El **tiempo de transmisión de la trama** viene dado por:

$$T_t = L \cdot T_b = \frac{L}{V_t}$$

donde L es el número de bits que conforman la trama; T_b es el tiempo de transmisión de un bit y V_t es la velocidad de transmisión de la trama

Se define como **tiempo de propagación** al tiempo que tarda 1 bit en viajar desde el emisor al receptor:

$$T_p = \frac{\text{Distancia (metros)}}{V_p \text{ (metros / segundo)}} = \frac{D}{c \cdot C}$$

donde D es la distancia (en metros que separa el emisor del receptor) y C es la velocidad de la luz.

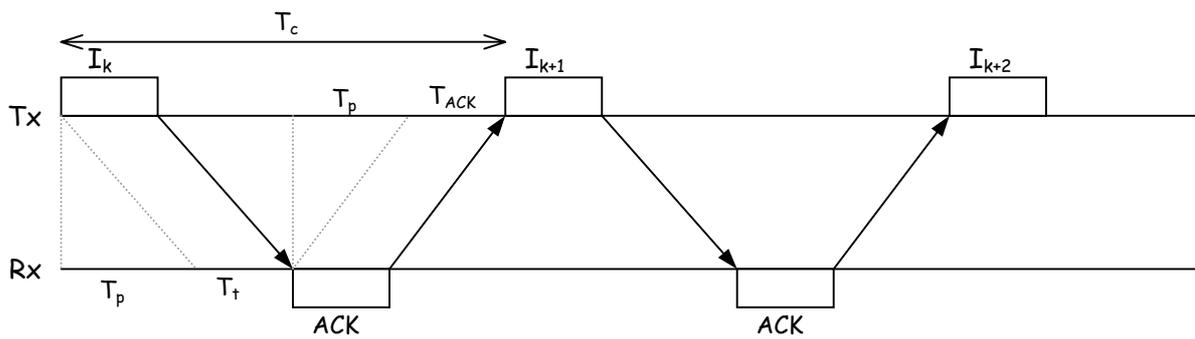
Se define el **Tiempo de proceso** de una trama T_{proc} como el tiempo que una trama tarda en ser procesada. En general suele ser despreciable, ya que:

$$T_{proc} \ll T_t + T_p + T_{ACK}$$

Siendo T_{ack} el tiempo invertido en la transmisión de una trama de confirmación (trama ACK)

El **tiempo de ciclo**, al que se ha hecho referencia anteriormente, se define como el tiempo transcurrido entre el envío del primer bit de una trama y el envío del primer bit de la trama siguiente. Viene dado por:

$$T_c = 2 \cdot T_p + T_t + T_{ACK} + 2 \cdot T_{proc}$$



Por lo que, en definitiva, la eficiencia de un protocolo (sin errores) viene dada por:

$$U_p = \frac{T_t}{T_c} = \frac{T_t}{2 \cdot T_p + T_t + T_{ACK}} \approx (L_{ack} \ll L_t) \approx \frac{T_t}{2 \cdot T_p + T_t} \approx \frac{1}{1 + 2 \cdot \frac{T_p}{T_t}}$$

La eficiencia del protocolo (sin considerar errores) será máxima al hacer tender a 1 el cociente anterior. Esto se conseguirá cuando:

a.- $\frac{T_p}{T_t} \rightarrow 0 \Rightarrow T_p$ tenderá hacia 0, es decir, el tiempo de propagación será muy pequeño o, lo que es lo mismo, la distancia que separa emisor y receptor es muy pequeña.

b.- $T_t \rightarrow \infty \Rightarrow$ Si el tiempo de transmisión crece mucho se debe a que L (tamaño de la trama crece a su vez mucho. La eficiencia aumenta, pero la probabilidad de errores en la trama también.

Además debe considerarse un temporizador o periodo de tiempo suficiente para permitir la llegada de la confirmación (ACK) enviada por el receptor. Un valor aproximado del temporizador puede ser:

$$T_{temp} \geq 2 \cdot T_p + T_{ACK}$$

Que garantiza que la confirmación ACK llegue a tiempo.

Por otra parte, si el circuito es semidúplex, al tiempo transcurrido en la transmisión (tiempo de ciclo) debe añadirse el tiempo gastado en la inversión del circuito.

Si se considera la posibilidad de que existan errores el tiempo de ciclo será ahora el tiempo que transcurre desde que se envía una trama hasta que se puede enviar la siguiente trama que no sea la misma (T'_c). La eficiencia del protocolo vendrá entonces dada por:

$$U_{p_{media}} = \frac{T_t}{T_c'} = \frac{T_t}{N_c \cdot T_c}$$

donde N_c es el número medio de transmisiones de una trama.

Si se define por P_f la probabilidad de que una trama sea errónea, consecuentemente $(1-P_f)$ será la probabilidad de que la trama sea correcta, por lo que:

$$T_c = 1 \cdot T_c \cdot (1-P_f) + 2 \cdot T_c \cdot (1-P_f) \cdot P_f + 3 \cdot T_c \cdot (1-P_f) \cdot P_f \cdot P_f + \dots = (1-P_f) \cdot T_c$$

Métodos como el de parada y espera son muy ineficientes para largas distancias al afectar el tiempo de propagación al denominador:

$$U_p = \frac{T_t}{2 \cdot T_p + T_t + T_{ACK}}$$

Por su parte, el temporizador solo ha de preocuparse de dar tiempo a llegar al ACK, y éste no afecta a la eficiencia.

Retransmisión continua

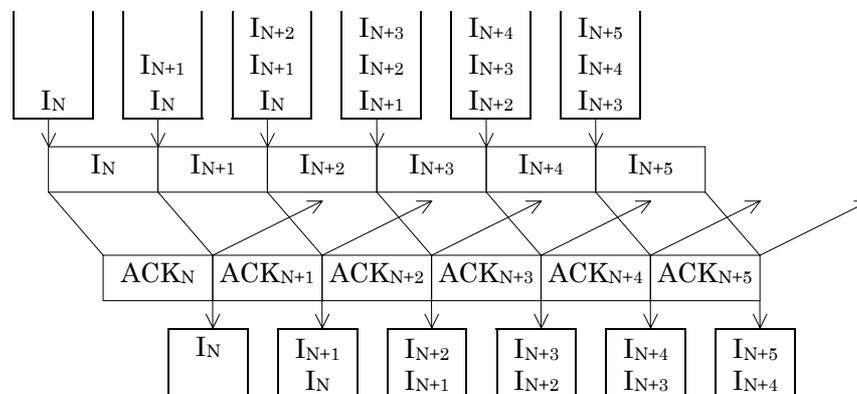
Como se ve, el problema del método de parada y espera es el tiempo de bloqueo o tiempo de espera a que llegue la confirmación:

$$T_{bloqueo} = T_c - T_t = 2 \cdot T_p + T_{ack}$$

El emisor, por tanto, sólo transmite en un tiempo igual a $T_c - T_{bloqueo}$. La eficiencia considerando el bloqueo será entonces:

$$U_{bloqueo} = 1 - U_p = \frac{T_c - T_t}{T_c}$$

En este tiempo de bloqueo el transmisor solo espera a transmitir y no hace nada más, por lo tanto se pierde tiempo. Para aprovechar el tiempo de bloqueo se utiliza la retransmisión continua.



Con esta retransmisión se tiene que buscar una estrategia de retransmisión por si alguna trama es errónea. En este caso la eficiencia máxima si no hay errores es:

$$U_p=1 \quad U_{\text{bloqueado}}=0$$

ya que siempre se está transmitiendo, y si no se transmite es que no hay nada que transmitir, y no porque se esté bloqueado.

En este sistema se necesitan memorias para guardar la trama enviada hasta que llegue la trama ACK correspondiente a la trama, para poder retransmitir la trama en caso de error. Esta memoria es el buffer de transmisión.

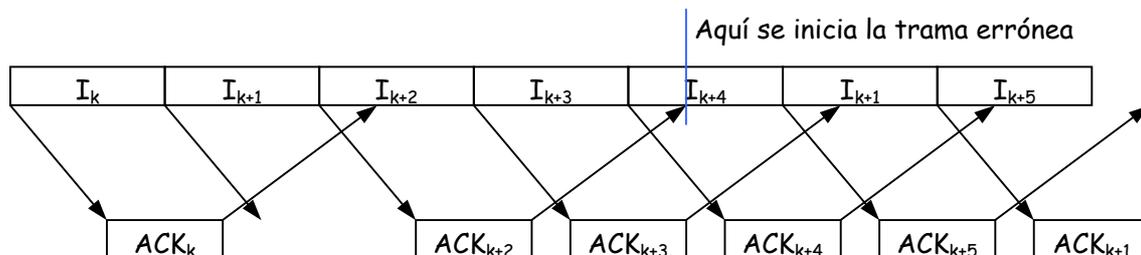
Es el emisor quien tiene el buffer de transmisión, porque tiene que ordenar las tramas, por si el receptor es mas lento, y para poder realizar copias (para saber si una trama es copia de otra o no).

Existen dos estrategias de retransmisión: la repetición selectiva y el Go_back_N.

Repetición selectiva

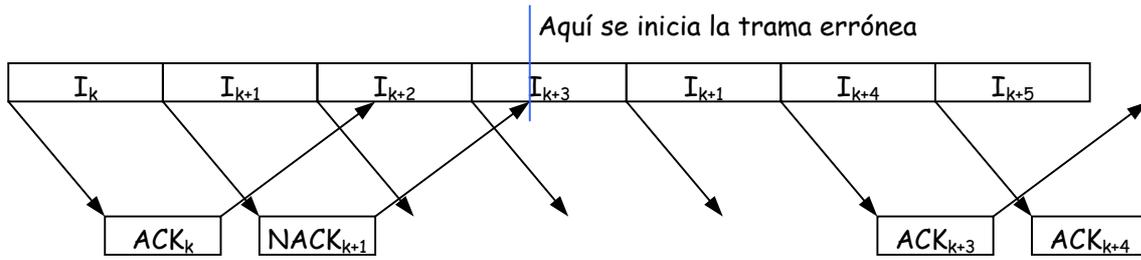
En este caso solo se retransmiten las tramas erróneas. Hay dos formas de trabajo:

Implícito: el receptor solo transmite los ACK. Cuando una transmisión es incorrecta, al transmisor no le llega el ACK pero se sigue transmitiendo tramas, hasta que se da cuenta que le falta una trama que es cuando la saca del buffer y la vuelve a enviar, una vez enviada se siguen enviando tramas normalmente. También utiliza el temporizador, por lo tanto se utiliza al igual que en el parada y espera, se utiliza por si hay tramas



consecutivas erróneas (y así se impide que se llenen los buffers), o por si es la ultima trama.

Explícito: el receptor transmite NACK, además de los ACK. transmite los NACK si la trama ha sido errónea. Los ACK son acumuladores, es decir, si transmite el ACK_{k+5}, esto nos indica que ha reconocido todas las tramas hasta la I_{k+5}, un solo reconocimiento de una trama reconoce esa trama y todas las anteriores. Cuando se envía el NACK no se pueden enviar ACK hasta que no nos llegue el ACK correspondiente a la trama que había



provocado el error.

En todos estos sistemas existe el temporizador por si la transmisión es errónea en los dos sentidos o de receptor a transmisor.

La eficiencia es la siguiente:

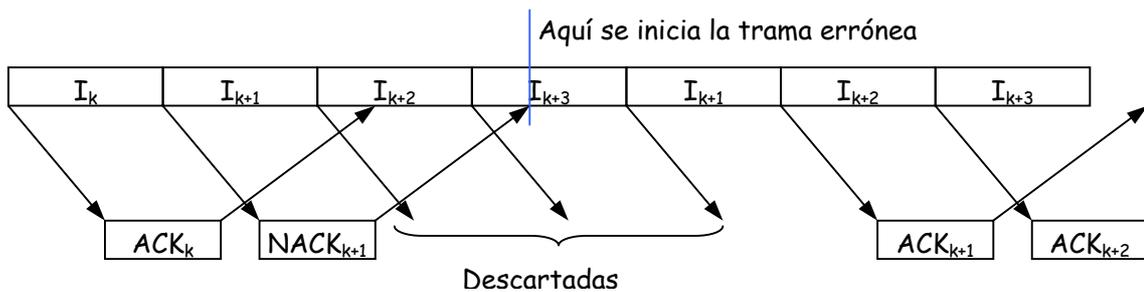
Sin errores: $U_p=1$

Con errores: $N_t \rightarrow$ número de transmisiones para una trama. $N_t=1/1-P_f$.

$$U_p = \frac{T_t}{N_t \cdot T_t} = \frac{1}{N_t}$$

Go_Back_N

Este sistema es explícito y cuando una trama es errónea, descarta las tramas que vengan hasta que la trama sea correcta. Este sistema recibe las tramas ordenadas.



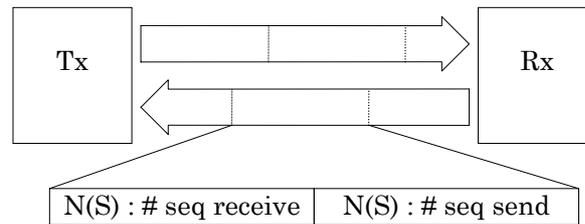
La eficiencia de este protocolo es:

- Sin errores: $U_p=1$
- Con errores:

$$U_p = \frac{T_t}{(N_t - 1) \cdot T_c + T_t} \stackrel{T_{ACK} \rightarrow 0}{=} \frac{T_t}{(N_t - 1) \cdot (2 \cdot T_p + T_t) + T_t} = \frac{T_t}{(N_t - 1) \cdot 2 \cdot T_p + (N_t - 1) \cdot T_t + T_t} = \frac{T_t}{(N_t - 1) \cdot 2 \cdot T_p + N_t \cdot T_t}$$

Si $N_t \rightarrow 1$ entonces $U_p=1/N_t$.

Piggy Backing



$$U_p = \frac{T_t}{T_c} = \frac{T_t}{2 \cdot T_p + T_t + T_{ack}} \stackrel{T_t \gg T_{ack}}{=} \frac{T_t}{2 \cdot T_p + T_t} \text{ caso óptimo}$$

Control de flujo

El receptor controla la velocidad a la que el transmisor emite tramas, porque el receptor puede ser mas lento que el transmisor. Para hacer el control de flujo el receptor bloquea al transmisor.

Un método es la **ventana deslizante** (sliding window).

Ventana deslizante

K_t = ventana de transmisión, que normalmente será una constante. La ventana de transmisión es el número de tramas que se pueden transmitir sin bloquearse.

L_{inf} = límite inferior de la ventana.

L_{sup} = límite superior de la ventana.

Si se quiere transmitir una trama se mira si tenemos espacio en la ventana para transmitir, esto se hace según el algoritmo siguiente:

si $L_{sup} - L_{inf} < K_t \rightarrow$ podemos transmitir;

$L_{sup} ++$;

sino ($L_{sup} - L_{inf} = K_t$) \rightarrow bloqueo del transmisor;

Si $L_{sup} - L_{inf} = K_t$ se bloquea el transmisor hasta que se reciba un ACK, con el que se aumenta el L_{inf} :

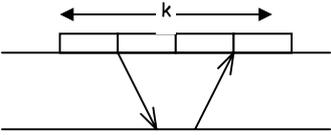
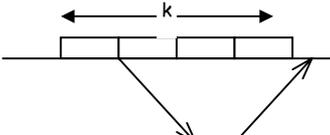
Solo se pueden transmitir K_t tramas, de esta forma nos aseguramos que el buffer del transmisor haya, como mucho, K_t tramas..

K_r = ventana de recepción, se define como el tamaño del buffer de recepción. Esta ventana se utiliza para controlar las tramas repetidas que puedan llegar al receptor.

A continuación podemos ver valores de K según el protocolo:

	K_t	K_r	
Parada y Espera	1	1	
Repetición selectiva	k	k	
Go_back_N	k	1	\rightarrow porque descarta las tramas y solo le importa la última que haya llegado

Eficiencia de una transmisión con ARQ continuo y control de flujo con ventana deslizante y sin errores:

	
Si ACK llega antes del límite de la ventana no nos bloquearemos nunca	Si ACK llega después del límite de la ventana nos bloquearemos.

Si $k \cdot T_t < T_c \rightarrow$ hay bloqueo; $U_p = k \cdot T_t / t_c$

si $k \cdot T_t \geq T_c \rightarrow$ No hay bloqueo; $U_p = 1$.

Si hay errores se ha de dividir la eficiencia del protocolo por N_t , en cualquiera de los casos.

Números de secuencia

Los números de secuencia sirven para la identificación de las tramas.

Si tenemos n bits entonces podemos secuencias 2^n tramas. Por lo tanto tenemos 2^n tramas que podemos generar.

- Parada y Espera: solo se necesita un bit (2 tramas) para las secuencias ya que solo hace falta numerar 2 tramas, se ha de poder distinguir las tramas.
- Repetición selectiva: $N \geq 2 \cdot Kt$
- Go_Back_N: $N \geq Kt + 1$ ya que si uno es incorrecto descarta las demás.

La ventana optima (K_{opt}) es aquella ventana que produce una eficiencia muy cercana a uno:

$$U_p = 1 \text{ si } Kt \cdot T_t \geq T_c \quad K_{opt} \geq \frac{T_c}{T_t}$$