

# ARQUITECTURA DE REDES SISTEMAS Y SERVICIOS.

2º Ing. Telecomunicación. Curso 2005/06

## EL NIVEL DE ENLACE

<b>1. INTRODUCCIÓN .....</b>	<b>3</b>
<b>2. SERVICIOS SUMINISTRADOS A LA CAPA DE RED .....</b>	<b>4</b>
2.1 SERVICIO SIN CONEXIÓN Y SIN ASENTIMIENTO .....	4
2.2 SERVICIO SIN CONEXIÓN Y CON ASENTIMIENTO .....	5
2.3 SERVICIO ORIENTADO A CONEXIÓN .....	5
<b>3. FORMACIÓN DE TRAMAS .....</b>	<b>7</b>
3.1 CUENTA CARACTERES .....	7
3.2 CARACTERES INICIO Y FINAL .....	7
3.3 BANDERAS INICIO Y FINAL .....	8
3.4 VIOLACIÓN DE CÓDIGO .....	8
<b>4. CONTROL DE ERRORES Y DE FLUJO.....</b>	<b>9</b>
4.1 DETECCIÓN DE ERRORES .....	9
4.2 CORRECCIÓN DE ERRORES .....	9
4.3 CONTROL DE FLUJO.....	12
<b>5. PROTOCOLOS DE ENLACE .....</b>	<b>13</b>
5.1 CONSIDERACIONES GENERALES.....	13
5.2 DATOS DE CONTROL EN LA TRAMA .....	13
5.3 PROTOCOLOS ELEMENTALES .....	14
5.3.1 UNILATERAL NO RESTRINGIDO .....	14
5.3.2 UNILATERAL DE PARADA Y ESPERA.....	14
5.3.3 UNILATERAL PARA CANAL RUIDOSO .....	15
5.4 VENTANA DESLIZANTE .....	17
5.4.1 VENTANA DESLIZANTE DE UN BIT.....	19
5.4.2 REPETICIÓN NO SELECTIVA .....	20
5.4.3 REPETICIÓN SELECTIVA .....	21

<b>6.</b>	<b><u>MEDIDA DE LA EFICIENCIA DEL ENLACE</u></b> .....	<b>24</b>
6.1	INTRODUCCIÓN: RENDIMIENTO Y CADENCIA EFICAZ. ....	24
6.2	PROTOCOLOS DE PARADA Y ESPERA SIN ERRORES .....	25
6.3	PROTOCOLOS DE PARADA Y ESPERA CON ERRORES.....	26
6.4	PROTOCOLOS DE VENTANA DESLIZANTE SIN ERRORES .....	27
6.5	PROTOCOLOS DE VENTANA DESLIZANTE CON ERRORES .....	28
<b>7.</b>	<b><u>EJEMPLOS: HDLC</u></b> .....	<b>29</b>
7.1	INTRODUCCIÓN:.....	29
7.2	GENERALIDADES .....	29
7.3	CONFIGURACIONES BÁSICAS Y MODOS DE OPERACIÓN.....	30
7.4	TRAMAS HDLC .....	32
7.4.1	TRAMAS SUPERVISORAS .....	33
7.4.2	TRAMAS SIN NUMERAR .....	34
7.5	EJEMPLOS .....	36
<b>8.</b>	<b><u>CONCLUSIONES</u></b> .....	<b>40</b>

## 1. INTRODUCCIÓN

En este tema se presentará el diseño de la capa 2, la capa de enlace. Se verán algoritmos para llevar a cabo una *comunicación fiable y eficiente entre dos máquinas adyacentes*, es decir físicamente conectadas mediante un canal de comunicación que actúa como un cable. Se puede considerar un canal “semejante a un cable” cuando los bits son entregados exactamente en el mismo orden que se transmitieron. Aunque puede parecer un problema trivial, tan fácil como que un equipo transmita y otro reciba datos, hay realizar una serie de tareas para coordinar este “sencillo” diálogo entre dos máquinas adyacentes. Estas tareas, de las que se ocupa el nivel de enlace, podríamos dividir las en tres tipos:

- **Control de errores:** Durante la transmisión pueden ocurrir errores con cierta probabilidad. Los fenómenos de atenuación distorsión y ruido afectarán a la señal transmitida y podrán provocar equívocos en la interpretación de la información recibida, de manera que lo que se transmitió como un uno puede entenderse como un cero en el otro extremo (o viceversa). Una de las tareas del nivel de enlace será asegurarse de corregir estos posibles errores en la transmisión.
- **Control de flujo:** Servirá para coordinar la cantidad de datos que puede enviar el transmisor sin saturar al receptor. Hay que tener en cuenta que las velocidades de proceso pueden no ser las mismas en ambos extremos de la comunicación y que los buffers de entrada del receptor podrían llegar a saturar.
- **Disciplina de línea:** Con estas técnicas se coordina el enlace decidiendo en cada momento qué dispositivo puede transmitir. Así, por ejemplo, en una línea semiduplex las máquinas implicadas en una comunicación tendrán que ponerse de acuerdo en los turnos de transmisión.

Como se verá posteriormente todas estas funciones están muy relacionadas unas con otras y es necesario considerar una serie de factores que afectan a la conexión entre dos equipos como por ejemplo; la probabilidad de errores en la transmisión, que la velocidad de transmisión y de proceso son finitas, que existe un retardo de propagación distinto de cero... todos estos factores tendrán implicaciones en la eficiencia de la transferencia de datos. Para ofrecer las prestaciones especificadas la capa de enlace deberá proporcionar una interfaz de servicio bien definida a la capa de red y utilizará los servicios proporcionados por el nivel físico.

## 2. SERVICIOS SUMINISTRADOS A LA CAPA DE RED

La función de la capa de enlace será la de proporcionar servicios a la capa de red (no olvidemos que estamos dentro del modelo OSI). El principal servicio ofrecido será transferir datos de la capa de red de la máquina origen a la capa de red de la máquina destino, separada de la anterior únicamente por un enlace. En la capa de red del origen habrá una entidad (proceso), que entregará los bits a la capa de enlace para su transmisión hacia el destino. Una entidad de nivel de enlace recogerá esta información y la procesará convenientemente para transmitirla a su entidad gemela (o paritaria) en el lado receptor. En este procesamiento, entre otras cosas, se añade información de control, sólo útil para las entidades de nivel de enlace, a la información original. Esta información de control es sólo para coordinar el diálogo (protocolo) entre entidades de nivel de enlace y por tanto no pasará, en ningún caso, al nivel de red. Aunque la trayectoria real de los datos será bajar en la torre de protocolos hasta el nivel físico, en este tema estudiaremos únicamente los protocolos entre dos procesos (entidades) en la capa de enlace considerando las capas adyacentes como proveedoras (nivel físico) o consumidoras (nivel de red) de servicios. La información intercambiada entre entidades paritarias del nivel de enlace se agrupa en PDUs que en este caso se suelen conocer como tramas.

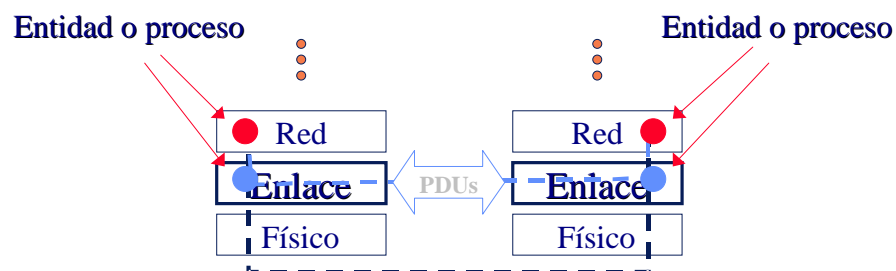


Fig. 1: El nivel de enlace en la torre OSI

*Cuestión 1: La finalidad de esta introducción es recordar el concepto de modelo de capas y situar al alumno en el nivel de enlace. ¿Qué tareas tenía el nivel físico? ¿Y el nivel de red?*

La capa de enlace se puede diseñar para que ofrezca distintos servicios, por lo tanto los servicios ofrecidos pueden variar de sistema a sistema. Hay tres posibilidades:

- Servicio sin conexión y sin asentimiento
- Servicio sin conexión y con asentimiento
- Servicio orientado a conexión

### 2.1 Servicio sin conexión y sin asentimiento

El origen transmitirá tramas independientes al destino, sin que se proporcionen asentimientos (reconocimientos) por parte de éste. No se establecerá ninguna conexión previa. Si la trama se perdiera o se dañara durante la transmisión no se realiza ningún intento por recuperarla por parte de la capa de enlace, que ni siquiera se daría cuenta del problema. Este tipo de servicio es muy conveniente cuando la tasa de error resulta muy

baja y la recuperación se delega a capas más altas. Resulta apropiado para los casos de tráfico en tiempo real, como el caso de voz, en la que la tardanza en la llegada de datos es peor que tener datos erróneos. Es muy utilizado en LAN.

*Cuestión 2: En RDSI se especifica que para el servicio de telefonía (conmutación de circuitos sobre canal B) no es necesario utilizar ningún protocolo de enlace. Analiza qué ventajas puede aportar utilizar un protocolo de enlace en aplicaciones telefónicas. ¿Tiene algún sentido?*

## 2.2 Servicio sin conexión y con asentimiento

En este servicio tampoco se establece una conexión entre ambos extremos, esto implica que se transmiten tramas totalmente independientes unas de otras, no se consideran pertenecientes a la misma conexión y por tanto no están relacionadas y por supuesto no van numeradas. Sin embargo sí existe un asentimiento individual a cada trama transmitida, es decir el receptor informa de que recibió correctamente o incorrectamente una trama, de manera que el emisor puede darse cuenta de que falló la transmisión en algún momento. Cuando el asentimiento no llegue dentro de un intervalo de tiempo especificado o cuando se reciba un asentimiento negativo se consideraría que la trama no llegó y se retransmitirá. De este modo la pérdida de un asentimiento puede ocasionar que una trama se transmita varias veces y por tanto se reciba (en la capa de red) varias veces, tampoco garantiza que el orden en el que se ofrecen las tramas a la capa de red del receptor sea el mismo que el orden en el que se transmitieron las tramas desde el emisor. Lo que sí asegura es que todo lo que se transmite se recibe.

*Cuestión 3: ¿En qué ocasiones podría servir esto?*

## 2.3 Servicio orientado a conexión

Es el servicio más complejo que la capa de enlace puede ofrecer a la de red. Ambos lados de la comunicación establecen una conexión antes de transmitir algún dato y todas las tramas que pertenezcan a esta conexión están relacionadas. Cada una de las tramas transmitidas a través de la conexión se numera y la capa de enlace garantiza que cada una de las tramas se reciba, exactamente una vez y que todas las tramas se reciban en el orden correcto. Por tanto, proporciona a los procesos de la capa de red el equivalente a un flujo de bits fiable. Es el servicio que aporta mayores prestaciones a capas superiores, y el que se verá cuando se expliquen, un poco más adelante, los protocolos de nivel de enlace.

Si se utiliza este servicio la transferencia tiene tres fases distintas. En la primera fase la conexión se establece cuando los dos lados han iniciado las variables y los contadores necesarios para mantener el seguimiento de las tramas recibidas, orden... La segunda fase será la de transmisión, en la que se mandan los datos de interés. Por último la conexión se libera, dejando libres las variables, memorias temporales, así como cualquier recurso que se necesite utilizar para mantener la conexión.

*Cuestión 4: Si el servicio ofrecido es "ligero" es decir, tiene pocas prestaciones, como ocurre con el servicio sin conexión ni asentimiento ¿qué cree que ocurrirá con las tareas de las capas superiores?.*

Para la comunicación entre la capa 3 y la 2 utiliza las primitivas de servicio de OSI, que estudiaremos brevemente a continuación. Las primitivas son:

- *Solicitud*: La capa de red utiliza estas primitivas para pedir a la capa de enlace que haga algo (p.e. establecimiento o liberación de la conexión, transmisión de una trama...).
- *Indicación*: Se utilizan para indicarle a la capa de red que se ha producido un evento (p.e. que otra máquina desea establecer o liberar una conexión, que ha llegado una trama...).
- *Respuesta*: La capa de red, en el extremo de recepción, utiliza las primitivas de respuesta para contestar a una indicación anterior.
- *Confirmación*: Proporcionan una manera de saber, en el extremo solicitante, si la solicitud fue realizada con éxito y si no, la razón por la cual no se llevó a cabo.



Fig.2: Primitivas en la interfaz Enlace/Red

### 3. FORMACIÓN DE TRAMAS

Como se ha visto en el apartado anterior la capa de enlace utilizará el servicio que le proporciona la capa física para proporcionar un servicio a la capa de red. La capa física aceptará los datos enviados por el nivel de enlace e intentará entregarlos al extremo destinatario, aunque no garantiza que este flujo de información esté libre de errores, como se ve en los temas dedicados al nivel físico. Dependerá de la capa de enlace detectar y, si fuera necesario, corregir estos errores.

Para lograr su objetivo las entidades de capa de enlace tienen que intercambiarse, junto a la información del nivel de red a transmitir, información de control del nivel de enlace que les servirá para coordinar su diálogo y ofrecer los servicios requeridos por el nivel superior. En definitiva hay que establecer un protocolo o normas de comunicación entre las entidades de enlace que asegure que se ofrecen al nivel de red los servicios solicitados. Para conseguir esto las entidades de nivel de enlace dividen el flujo de datos a transmitir en trozos discretos a los que añaden la información de control necesaria formando así las PDU de nivel de enlace o *tramas*.

Una de las primeras tareas a la hora de establecer las normas para el diálogo entre entidades paritarias de nivel de enlace será definir cómo se forman estas tramas y cómo identificar el principio y final de las mismas. Una manera de distinguir las tramas podría ser incluir intervalos de tiempo entre éstas, como se hace al escribir entre palabra y palabra, pero este mecanismo exigiría una sincronización muy exacta y compleja entre emisor y receptor, muy difícil de lograr. Por tanto se diseñan métodos alternativos como son:

- Cuenta caracteres
- Caracteres de inicio y final, con inserción de carácter
- Banderas de inicio y final, con inserción de bit
- Violaciones de código en la capa física

#### 3.1 Cuenta caracteres

Se utiliza un campo en la cabecera para especificar el número de caracteres de la trama. En el extremo receptor se sabrá así dónde termina la trama. El problema es que si existe un error de transmisión que distorsiona el campo de cuenta se desincronizan emisor y receptor y no será posible localizar el inicio de la siguiente trama. En la actualidad no se suele utilizar este método de forma aislada, aunque sí que se añade a los que vamos a ver a continuación para dar mayor seguridad.

#### 3.2 Caracteres inicio y final

Cada trama comenzará con una secuencia de caracteres ASCII predeterminada (Un carácter denominado generalmente de escape y otro que indique el principio de trama [DLE][STX]) y terminará con una secuencia distinta (Carácter de escape y otro

que indique el final de trama [DLE][ETX]). De manera que el carácter de escape indicará que lo que sigue es información de principio o fin de trama. Sin embargo podría suceder, con facilidad, que las secuencias de caracteres elegidas como [DLE][STX] y [DLE][ETX] ocurrieran dentro de los datos de nivel de red (los datos útiles que se están transmitiendo), con lo que el proceso de entramado tendría problemas (se identifica un principio o final de trama donde no lo hay). Una forma de resolver este problema consiste en hacer que la capa de enlace del extremo emisor inserte un carácter ASCII de escape [DLE], justo antes de cada ocurrencia del este carácter de escape [DLE] en los datos útiles. Esto obliga a que el receptor tenga que eliminar este carácter [DLE] duplicado antes de entregar los datos a la capa de red. Este proceso deberá ser totalmente transparente a la capa 3. A esta técnica se le conoce como inserción de carácter. El problema principal de este método de entramado es su gran dependencia de la utilización de caracteres de 8 bits y que está orientado por tanto a trabajar con tramas de un número entero de octetos, lo que puede no resultar eficiente en algunos casos.

*Cuestión 5: Analice cómo puede afectar tener que duplicar caracteres al pensar en el rendimiento del enlace, es decir, en la cantidad de datos útiles que se transmiten hacia el nivel de red respecto al número total de bits transmitidos sobre el enlace.*

### 3.3 Banderas inicio y final

Con este mecanismo se podrá utilizar un número arbitrario de bits por carácter, no está orientado a caracteres de 8 bits. Cada trama comenzará y terminará con un patrón de bits especial (p.e. 01111110). El receptor podrá así identificar el principio y fin de trama. Sin embargo podría ocurrir que la secuencia elegida como bandera se encontrara dentro de los datos útiles, en ese caso hay que utilizar algún mecanismo que asegure la transparencia. Así cada vez que la entidad de nivel de enlace emisora encuentre cinco unos consecutivos en los datos útiles a transmitir deberá insertar un bit 0, que será eliminado por la entidad receptora, todo de forma transparente a la capa de red. Esta técnica se conoce como de inserción de bit y asegura el reconocimiento de la frontera entre dos tramas sin ninguna ambigüedad.

*Cuestión 6: Cree que el rendimiento en este caso será mejor o peor que con la técnica anterior.*

### 3.4 Violación de código

Sólo se utiliza en redes en las que la codificación que se hace en el medio físico lo permite. Por ejemplo: Si se utiliza el código Manchester en un intervalo de bit se utilizan las secuencias +V -V ó -V +V, pero no +V +V ni -V -V, estas se utilizarán para identificar el principio y final de trama.

*Nota: Al estudiar los códigos en banda base en temas posteriores se verá el código Manchester más detenidamente.*

## 4. CONTROL DE ERRORES Y DE FLUJO

Una vez decidida la forma de identificar el principio y fin de las tramas es necesario asegurar que *toda* la información de nivel de red que se transmite dentro de esas tramas se entrega a la entidad de red destino en *orden* y *sin errores*. Esto obliga a las entidades de nivel de enlace a utilizar mecanismos para, en primer lugar, detectar posibles errores en los datos recibidos y, en segundo lugar, recuperarse de los errores ocurridos en caso de ser necesario.

### 4.1 Detección de errores

En cuanto a la detección de errores en el destino los mecanismos utilizados se basan en añadir a los datos a transmitir información de control (información redundante) que permita que el extremo receptor deduzca que alguno/s de los bits recibidos son erróneos. De este modo la trama tendrá  $m$  bits de datos y  $r$  bits de comprobación de errores (redundantes). Un ejemplo sencillo de código detector de errores sería un código en el que solamente se utiliza un bit de redundante denominado bit de paridad que se transmite junto al resto de información, en este caso  $r=1$ . Este bit se selecciona de forma que el número de bits con valor 1 en la trama sea par. Así si ocurre un error en un bit y este cambia de 0 a 1 el número de unos será impar y por tanto el receptor se dará cuenta de que algo ha ido mal.

*Ejemplo: Se quiere transmitir el bloque:*

*00010100010 ( $m=11$ ) el número de unos es 3 (impar) así que será necesario que el bit  $r$  sea un 1 para que el número de unos sea par (4). La secuencia a transmitir es entonces:*

*000101000101 ( $m+r=12$ ), (4 unos, par). Se recibe la secuencia:*

*010101000101 (Error en el segundo bit), (5 unos, impar), el receptor sabe que debía ser par así que sabe que ha habido algún error.*

*Cuestión 7: ¿Qué ocurriría si hubiera errores en más de un bit?*

*Nota: En los temas de teoría de información se ven códigos de redundancia más eficaces.*

### 4.2 Corrección de errores

En lo referente a la corrección de los posibles errores que ocurran en la transmisión es importante considerar la naturaleza y patrón de estos errores. Generalmente son provocados por fenómenos físicos, ya estudiados en temas anteriores, como ruido, interferencias, atenuación, distorsión, diafonía... De ahí que una característica importante es que los errores tienden a presentarse a ráfagas, más que aisladamente. Es decir, cuando una línea de transmisión falla lo hace durante un intervalo de tiempo continuado (por ejemplo porque ocurre una interferencia por la

puesta en marcha de un motor cercano) por lo que todos los bits que se transmitan en ese intervalo pueden tener errores. Esto tiene ciertas ventajas y desventajas frente a la existencia de errores aislados que afecten a un solo bit. Por un lado como los datos se envían en bloques de bits si los errores fueran aislados con la misma probabilidad de error casi todos los bloques serían erróneos, mientras que si se son a ráfagas hay muchos bloques que no son erróneos. Por ejemplo si se consideran bloques de 1000 bits y que la probabilidad de error es de 0.001 (un bit mal de cada 1000) con una distribución homogénea prácticamente todos los bloques tendrían error, pero si los errores ocurren en ráfagas de 1000 bits erróneos sólo un bloque de cada 1000, aproximadamente, será erróneo. Si hubiera que retransmitir cada vez que un bloque fuera erróneo en el primer caso habría que retransmitir prácticamente todos mientras que en el segundo tan solo uno de cada 1000, aproximadamente. Sin embargo los errores a ráfagas son más difíciles de detectar y corregir que los errores aislados y más difíciles de modelar analíticamente.

Se han desarrollado dos estrategias fundamentales para enfrentarse a la tarea de recuperación frente a fallos:

**FEC (Forward Error Correction):** La idea es incluir una cantidad suficiente de información redundante, junto con cada bloque de datos enviado, para permitir en el extremo receptor deducir qué información se transmitió originalmente. Se emplean por tanto códigos correctores de errores como por ejemplo códigos cíclicos, que se verán en temas posteriores. En estos códigos el número de bits redundantes ( $r$ ) tiene que ser suficientemente grande para permitir esta corrección de los errores y eso puede dar lugar a una disminución de la eficiencia, sobre todo cuando la probabilidad de error crece y el número de bits redundantes tiene que ampliarse. Para evitar esta disminución de eficiencia es necesario limitar el número de bits redundantes y por tanto la capacidad correctora de los códigos está limitada, con lo que en muchas ocasiones se combinan estas técnicas con las de ARQ, que veremos a continuación.

**ARQ (Automatic Repeat reQuest):** En este caso la información redundante que se incluye sólo pretende permitir que el extremo receptor pueda descubrir que ocurrió un error y si es así se informará al extremo transmisor para solicitar un reenvío de la información. Se utilizan códigos que sólo necesitan detectar los errores y por tanto son necesarios menos bits de redundancia. De manera que la entrega fiable se proporciona gracias a la existencia de una realimentación hacia el transmisor respecto a lo que está sucediendo en el otro extremo del enlace. Es decir que el lado receptor indicará de algún modo al emisor qué está ocurriendo en esa comunicación. Los protocolos de comunicación suelen basarse entonces en el envío, por parte del receptor, de tramas especiales de control (asentimientos positivos o negativos) que contienen información acerca de cómo se han recibido las tramas (sin error o con error). De este modo recibir un asentimiento positivo sobre una trama indica al transmisor que la trama ha llegado bien, mientras que recibir un asentimiento negativo significa que la trama se deberá transmitir de nuevo.

*Cuestión 8: ¿Cree que podría funcionar esto sólo con asentimientos positivos?  
¿por qué?*

Será necesario considerar también el caso de que una trama se perdiera por completo. En este caso el receptor no reaccionará y no mandará ningún asentimiento, ni positivo ni negativo, por otro lado también se podría perder el asentimiento. En estos casos el efecto sería que el emisor se quedaría parado esperando a recibir un asentimiento que nunca llegará. Esta situación se controla introduciendo el uso de temporizadores en la capa de enlace. Cuando el emisor envía una trama inicia un temporizador de manera que transcurrido un intervalo de tiempo determinado sin recibir noticias del receptor se retransmitirá la trama, sin necesidad de recibir ningún asentimiento negativo. Evidentemente si lo que se había perdido era el asentimiento se corre el peligro de que el receptor acepte dos veces (o más) la misma trama, para evitar esto se asignan números de secuencia a las tramas, de esta forma se identifica cada trama y el receptor puede distinguir tramas originales de tramas retransmitidas (que tendrán el mismo número de secuencia).

### **Retardo de ida y vuelta**

La elección de la duración de los temporizadores exige considerar un importante parámetro el denominado retardo de ida y vuelta asociado al enlace, también conocido como **round-trip delay**.

Se define el **RTD** como *el tiempo desde que se empieza a transmitir el primer bit de una trama hasta el instante en que se recibe (en el transmisor) el último bit del reconocimiento asociado a esa trama.*

Este tiempo es función por un lado del **tiempo de transmisión de las tramas** en el enlace, que dependerá de su régimen binario o capacidad y de la longitud de las tramas a transmitir. Por otro lado del **retardo de propagación en el enlace**, es decir del tiempo transcurrido desde que la señal se introduce en un extremo del “cable” hasta que aparece en el otro extremo. Este retardo será por tanto función de la velocidad de propagación en el enlace y de la longitud del mismo. Resumiendo los tiempos que afectan al RTD son:

$$T_{tx} = \text{Longitud de la trama (b)}/R_b \text{ del enlace (b/s)}$$

$$T_p = \text{Separación física (m)}/\text{Velocidad de propagación (m/s)}$$

La figura 3 representa este parámetro. A lo largo del tema se utilizará este tipo de diagramas para mostrar el intercambio de PDUs o tramas a lo largo del tiempo entre dos entidades el nivel de enlace (Tx y Rx).

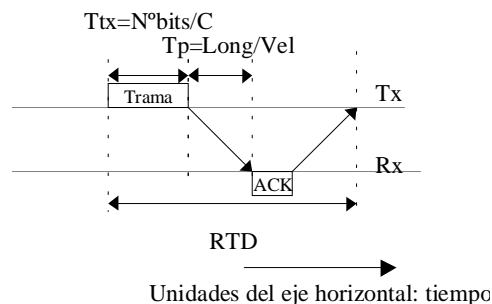


Fig. 3: Round Trip Delay

Este parámetro influye fuertemente en la duración de los temporizadores que se utilizan en los algoritmos de control del enlace. Por un lado la duración del temporizador tiene que ser superior al RTD para evitar retransmisiones innecesarias, por otro lado se debe procurar que el temporizador sea lo más pequeño posible para aumentar la eficiencia del enlace.

*Cuestión 8: ¿Qué ocurre si el temporizador es menor que el RTD?*

*Cuestión 9: ¿Qué problema hay en que el temporizador sea demasiado grande?*

Considerando todo esto una de las tareas más importantes de las entidades de enlace será el control de temporizadores y números de secuencia de las tramas, con la finalidad de asegurar que éstas sean entregadas una vez, y sólo una vez, y en el orden correcto al nivel de red, de este modo se ofrecerá un servicio orientado a conexión a las entidades de nivel de red.

### **4.3 Control de flujo**

Otra de las tareas del nivel de enlace era el control de flujo. Se deberá asegurar que el extremo emisor envía datos a una tasa suficientemente lenta como para que se puedan procesar estos datos en el extremo receptor sin que éste saturé. Para ello también debe existir realimentación de lo que ocurre en el extremo receptor. Existen varios esquemas de control de flujo pero todos se basan en el mismo principio básico, definir reglas respecto a cuándo el emisor debe transmitir la siguiente trama. De manera que el emisor sólo transmite la siguiente trama cuando tiene constancia de que el receptor ha dado permiso para ello. Por ejemplo el emisor puede tener permiso para transmitir “n” tramas (n créditos de transmisión), cada vez que transmita una trama perderá un crédito y cuando haya transmitido las n tendrá que parar. El receptor podrá enviar créditos de transmisión al emisor cuando quiera, este envío lo hará en función de su capacidad para procesar las tramas que está recibiendo, esto le dará autorización al emisor para seguir transmitiendo, le aumentará el crédito de transmisión. Como se verá más adelante los mecanismos de control de flujo y de errores están íntimamente ligados en los protocolos de nivel de enlace.

## 5. PROTOCOLOS DE ENLACE

### 5.1 Consideraciones generales

A partir de este momento se considerará que el servicio que necesita la capa de red es un servicio orientado a conexión y fiable de manera que las entidades de nivel de enlace tienen que realizar todos los procedimientos necesarios para conseguir ofrecer este servicio. Se considerará también que la capa de red siempre tiene datos para transmitir y por tanto la capa de enlace no tiene que esperar a que le lleguen datos de nivel de red. Esto quiere decir que si en algún momento los protocolos exigen una parada de los procedimientos ésta será impuesta por los propios protocolos de enlace y no porque no haya datos que transmitir por parte del nivel de red. Por supuesto no será necesario analizar las cabeceras de niveles superiores, todo lo que venga del nivel de red se consideran datos a transmitir, esta consideración es propia de los modelos de capas.

*Cuestión 10: ¿Qué ventajas tiene el hecho de que las entidades de un nivel no necesiten analizar las cabeceras de orden superior?*

Al aceptar un paquete de la capa de red la capa de enlace lo encapsula en una trama, agregándole una cabecera y una cola con información de control del enlace. De este modo una trama estará formada **por DATOS a TRANSMITIR + DATOS de CONTROL**. En estos bits de control se incluyen también los de redundancia para el control de errores. El emisor calcula el código de redundancia y lo incluye en la trama mientras que en el extremo receptor se comprobará este código asegurándose así de que efectivamente los datos llegaron sin error. Cuando se comprueba que la trama llegó correctamente, se entregarán los datos útiles de la trama a la capa de red, por supuesto en ningún caso se entregará la cabecera de trama a la capa de red y se asegura así la autonomía de los protocolos de distintas capas.

### 5.2 Datos de control en la trama

**Número de Secuencia:** será un entero que se utiliza para numerar las tramas con objeto de identificarlas separadamente pero proporcionando una relación entre ellas (servicio orientado a conexión). Se define la constante Máximo Número de Secuencia (MNS), que podrá ser distinto para cada protocolo. El número de secuencia irá en el rango de 0 hasta MNS, incluido éste.

**Código de Redundancia:** para que el receptor verifique la integridad de los datos recibidos. Si es suficientemente grande puede incluso servir para corregir los errores de transmisión. A partir de ahora consideraremos sólo la posibilidad de detectar errores y no de corregirlos gracias a la información redundante.

**Tipo de trama:** será un campo (o varios) que permiten identificar el tipo de trama que se está intercambiando. Cuando se estudie HDLC, que es un caso particular de protocolo de nivel de enlace, se verán ejemplos de los tipos de tramas que pueden

existir. Como adelanto decir que hay tramas que no llevan información de nivel de red y que sólo sirven para llevar el control de los procedimientos del nivel de enlace.

**Asentimiento:** como se verá más adelante uno de los campos de control de la trama será el campo de asentimiento, que servirá para indicar al otro lado las tramas que se recibieron correctamente.

Supondremos que el canal es inseguro en la mayoría de los protocolos, por lo que se podrán perder tramas enteras. Para recuperarse de estos errores la capa de enlace emisora utiliza temporizadores siempre que envía una trama y retransmite si estos vencen. Vamos a ver distintos protocolos de complejidad creciente. Si en todo caso el servicio buscado es el de un servicio orientado a conexión y fiable entonces en los protocolos ligeros, de pocas prestaciones, es necesario imponer condiciones para que el servicio ofrecido realmente esté cumpliendo lo que se pide. En los protocolos más complejos, sin embargo, no se impondrán restricciones externas.

### 5.3 Protocolos elementales

#### 5.3.1 Unilateral no restringido

Es un protocolo muy sencillo. Los datos de nivel de red se transmiten en una sola dirección (unilateral) de manera que uno de los extremos del enlace actúa siempre como transmisor y el otro siempre como receptor. Para considerar que el servicio ofrecido por el nivel de enlace utilizando este protocolo es orientado a conexión es necesario que no haya errores, no se pierden datos en el canal y que la velocidad de proceso de las tramas recibidas sea infinita (utopía). El proceso emisor opera en la capa de enlace de la máquina fuente y el receptor en la capa de enlace de la máquina destino.

Este protocolo consiste en no hacer prácticamente nada, como indica la figura 4. No se usan números de secuencia ni asentimientos, sólo se utiliza el campo de información de las tramas. Simplemente el equipo transmisor mandará los datos que le llegan de la capa de red y el receptor los cogerá y los mandará a la capa de red receptora. Si las condiciones no fueran las que se han especificado el servicio proporcionado por este protocolo sería no orientado a conexión y sin asentimiento (no fiable). Como sólo hay tráfico en un sentido un canal simplex serviría para llevar a cabo este proceso.

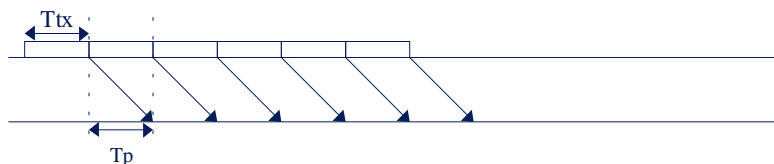


Fig. 4: Protocolo unilateral no restringido

#### 5.3.2 Unilateral de parada y espera

Sigue siendo un lado el transmisor y otro el receptor (unilateral). En este caso el único “esfuerzo” de la capa de enlace consiste en que se proporciona una realimentación al emisor por parte del receptor. El transmisor sólo está autorizado a enviar una trama,

cuando el receptor pase la información a la capa de red devolverá un asentimiento hacia el otro lado que efectivamente autoriza al emisor a enviar la siguiente trama (de ahí el nombre de parada y espera, se envía, se para y se espera al asentimiento). Este procedimiento evita que el emisor inunde la parte receptora transmitiendo datos a una velocidad superior de la que la parte receptora puede procesar y por tanto proporciona un sencillo mecanismo de control de flujo. Hay que considerar que se necesita un tiempo para procesar la trama, como se aprecia en la figura 5, y que si el transmisor envía tramas a una tasa demasiado elevada podría hacer saturar el buffer de recepción, que no tendría capacidad para almacenar tantas tramas, con este mecanismo esto nunca ocurriría ya que las tramas se mandan de una en una.

Aunque la información de nivel de red sólo viaja en un sentido es necesario transferir datos bilateralmente, ya que los asentimientos se transmiten en sentido contrario. Se llevará a cabo una estricta alternancia de flujo; emisor envía trama, receptor envía ACK, emisor envía otra trama... Un canal físico semi-duplex serviría para este caso. En el procedimiento mostrado no hay ningún mecanismo de control de errores, de manera que para ofrecer el servicio orientado a conexión es imprescindible que el canal sea sin errores.

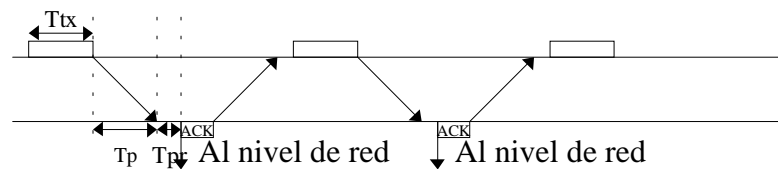


Fig. 5: Protocolo unilateral de parada y espera

### 5.3.3 Unilateral para canal ruidoso

Este protocolo de nivel de enlace es más complejo y ofrecerá un servicio orientado a conexión aún cuando exista la posibilidad de que las tramas tengan errores o se pierdan. Se añaden mecanismos para detectar errores, como se explicó anteriormente se añade redundancia de control que permite al receptor comprobar si lo que se recibe es realmente lo que se transmitió. De este modo el receptor sólo envía el asentimiento cuando la trama llega correctamente, si tiene errores se desecha y no se envía asentimiento. Si el emisor está demasiado tiempo esperando el asentimiento le vence un temporizador y reenvía la trama, este procedimiento se repetiría hasta que por fin se reciba el asentimiento positivo.

El inconveniente de este tratamiento sería que la trama que se perdiera fuera la de asentimiento y por tanto se recibieran tramas duplicadas. De manera que hay que añadir algún mecanismo que permita que el receptor distinga una trama que ve por primera vez de aquella que es retransmitida. La manera de hacer esto es que el emisor coloque un número de secuencia en la cabecera de cada trama transmitida, es decir identificar la trama con un número. De este modo el receptor comprueba el número de secuencia de cada trama recibida y puede deducir si se trata de una nueva trama o de un duplicado que debe desechar y no enviar al nivel de red.

*Cuestión 11: ¿Cuándo debe enviar asentimiento el receptor la primera vez que se reciba la trama bien o todas las veces que se reciba bien?*

Para optimizar la eficiencia del protocolo será deseable tener la cabecera lo menor posible ¿Cuál será el mínimo número de bits necesarios para el número de secuencia? La única ambigüedad que existe en este protocolo se encuentra entre una trama y su sucesora directa ( $m$  y  $m+1$ ). Si se pierde o se daña la trama  $m$  el receptor no la reconocerá, así que el emisor seguirá tratando de enviarla. Una vez que se reciba correctamente, el receptor devuelve el ACK. Según se reciba o no este ACK en el lado transmisor, la siguiente trama que se recibirá el otro lado será la  $m+1$  (si se recibe correctamente el asentimiento) o de nuevo la  $m$  (si se pierde el ack). En el primer caso todo ha ido bien, se asiente a la  $m+1$  y se espera la siguiente. En el segundo caso es necesario que el receptor la identifique como un duplicado y no la reenvíe al nivel de red, así que hay que evitar confundir la  $m$  con la  $m+1$ . Concluyendo, debido a que el protocolo es de parada y espera la única posibilidad en recepción es recibir realmente la trama esperada o recibir de nuevo la anterior (si hubo problemas con el asentimiento) de manera que sólo es necesario distinguir dos tramas. Así que un número de secuencia de un bit, 0 ó 1, es suficiente para evitar confusiones. En cada instante de tiempo el receptor espera el siguiente número de secuencia. Cualquier trama que llegue con un número de secuencia incorrecto se rechaza como un duplicado. Cuando llega la trama con el número de secuencia correcto se acepta y se manda a la capa de red, cambiando el número de secuencia esperado. La figura 6 muestra el funcionamiento de este protocolo.

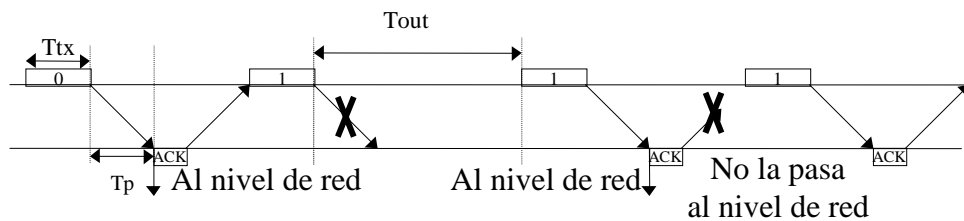


Fig. 6: Protocolo de parada y espera con control de errores.

A estos protocolos, en los que el emisor espera un asentimiento positivo antes de avanzar a la siguiente trama, se les suele denominar como PAR (Asentimiento con retransmisión positivo). Los datos de información viajan en una sola dirección, aunque la transmisión es bidireccional (semi-duplex por la alternancia de eventos).

El protocolo maneja bien la posibilidad de que se pierdan tramas pero es necesario controlar muy bien la duración de los temporizadores que deben ser suficientemente largos para impedir vencimientos prematuros. La figura 7 muestra un caso con el temporizador mal ajustado y que da lugar a un funcionamiento defectuoso.

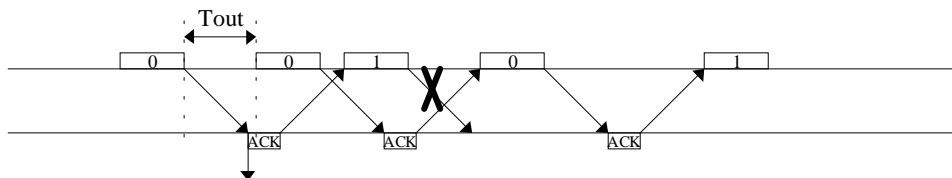


Fig. 7: Funcionamiento defectuoso del protocolo

*Cuestión 12: Analiza detenidamente el error en el protocolo de la figura 7.*

## 5.4 VENTANA DESLIZANTE

En los protocolos anteriores la información del nivel de red se transmite en una sola dirección, pero casi siempre existe la necesidad de transmitir en ambas direcciones y además simultáneamente, es decir que la comunicación entre entidades de red, por lo general, será full-duplex. Se podría considerar utilizar dos circuitos físicos separados, uno para cada sentido de la comunicación entre las entidades de red, cada circuito constaría de un canal para datos y otro para asentimientos. Pero en ese caso la capacidad del canal de asentimientos se desperdiciaría mucho, ya que las tramas de asentimiento pueden ser de pequeña longitud al no contener datos de red. Una mejor idea será utilizar el mismo circuito para datos en ambas direcciones, de forma que las tramas con información de las entidades de red del sistema A hacia las entidades de red del B se mezclan con las tramas de asentimiento que viajan en ese mismo sentido, es decir las que asienten a la información que se transmitió desde B a A. De este modo transmitiremos tramas de datos y de asentimiento sobre el mismo canal y se utiliza parte del campo de control para distinguirlas.

Pero existe una forma aún más eficiente de aprovechar la capacidad del canal, el método de *Superposición o piggybacking*. Al utilizar esta técnica en lugar de enviar inmediatamente los asentimientos en un sentido se espera a tener información que transmitir en ese sentido y se envía todo junto en la misma trama, como se muestra en la figura 8. De esta forma el asentimiento desde A hacia B se adjunta a la trama de datos en este mismo sentido, para ello en la cabecera de la trama se añade un campo de asentimiento. Así se aprovecha mejor el ancho de banda disponible del canal ya que añadir un campo de asentimiento en la cabecera de una trama supone sólo enviar unos cuantos bits más, en tanto que mandar una trama exclusivamente para esto necesita una cabecera completa, código redundancia... Este método presenta la complicación de tener que calcular correctamente cuánto tiempo debe esperar la capa de enlace a que llegue un paquete desde el nivel de red sobre el que se superpondrá el asentimiento. Si en el plazo fijado no llega nada para transmitir desde la capa de red es necesario enviar el asentimiento sólo, ya que si no el otro lado se quedaría eternamente esperando.

	Datos	NS	ACK	
--	-------	----	-----	--

Fig. 8: Superposición de datos y asentimientos

En los protocolos de ventana deslizante cada trama de salida contiene un número de secuencia (NS en la figura 8), cuyo valor se encuentra en un rango de 0 a un valor máximo ( $2^n - 1$ , siendo  $n$  el número de bits asignados al campo de secuencia). En cada protocolo se utilizará un número de bits distinto, según las necesidades concretas. Los números de secuencia se van asignando de forma secuencial, valga la redundancia, a las tramas que se van formando. Cuando se alcanza el máximo número de secuencia se vuelve a empezar por el cero, es decir que la numeración es cíclica.

Son dos los conceptos clave en este tipo de protocolos:

- **La ventana de transmisión:** En estos protocolos el emisor puede enviar varias tramas consecutivas, sin necesidad de esperar el asentimiento a una para transmitir la siguiente.

*La ventana de transmisión en un momento determinado es el conjunto de números de secuencia correspondientes a tramas que ya se han enviado y de las que aún no ha recibido asentimiento.*

Cuando se manda una trama su número de secuencia se introduce en la ventana de transmisión, cuando se recibe el asentimiento correspondiente éste se saca de la ventana de transmisión. El tamaño de esta ventana está limitado, de manera que hay un número máximo de tramas que pueden estar esperando a ser asentidas. Este mecanismo es por tanto una transmisión basada en créditos, enviar una trama implica consumir un crédito, recibir un asentimiento es ganar un crédito. Mientras que no se reciba el asentimiento a una trama ésta se mantiene en la memoria del emisor, de modo que si fuera necesaria la retransmisión, porque venciera el temporizador sin recibir el asentimiento correspondiente, no hay que pedir los datos de nuevo al nivel de red haciendo el procedimiento totalmente transparente a éste.

*Cuestión 13: ¿Qué implicación tendrá el tamaño máximo de la ventana de transmisión en la memoria necesaria en el transmisor?*

- **La ventana de recepción:** En el receptor se mantiene una ventana de recepción que corresponde a la lista de los números de secuencia de las tramas que éste está autorizado a aceptar.

*La ventana de recepción es el conjunto de números de secuencia que el receptor está autorizado a aceptar en un momento determinado*

Cualquier trama que llegue y cuyo número de secuencia no esté en esta ventana se descarta, si está dentro de la ventana sí se acepta. Sin embargo es importante recordar que los datos deben ser pasados al nivel de red en orden. De manera que sólo si se recibe la trama con el número de secuencia del borde inferior de la ventana, la primera que se está esperando en orden, se pasa a la capa de red y se envía asentimiento. Cada vez que esto ocurra se modifica la ventana de recepción ya que los números de secuencia que ahora se van a aceptar son otros. Si la trama recibida pertenece a la ventana de recepción pero no es la primera esperada entonces el receptor la almacena mientras llegan las que tendrían que haber llegado antes y no se envía asentimiento ni se modifica la ventana de recepción. Cuando lleguen las esperadas en orden se enviarán al nivel de red, se asentirá a la última trama correcta y ordenada que se recibió y se modificará la ventana de recepción. El asentimiento recibido indicará al emisor que todas las anteriores también se recibieron bien (los asentimientos son por tanto acumulativos, un asentimiento a una trama asiente a las anteriores también).

Las ventanas de emisión y recepción no tienen que tener el mismo tamaño.

*Cuestión 14: ¿Cómo puede ocurrir que se reciban tramas desordenadas en el receptor?*

*Cuestión 15: ¿Qué significa avanzar o modificar la ventana de recepción?*

*Cuestión 16: Analice si hay algún beneficio en que la ventana de recepción sea mayor que la de transmisión.*

*Cuestión 17: En una comunicación full-duplex (ambos lados envían y reciben) ¿cuántas ventanas de recepción y transmisión hay que controlar?*

### 5.4.1 Ventana deslizante de un bit

Se va a utilizar un tamaño máximo de ventana de transmisión de 1 y el tamaño de la ventana de recepción también será 1. Como sólo puede haber una trama pendiente de asentimiento el protocolo es de parada y espera, ya que el emisor envía una trama y espera su asentimiento antes de poder enviar la siguiente. En este caso es suficiente tener dos números de secuencia, sólo se puede confundir una trama con la antecesora o la sucesora. La entidad de enlace emisora recoge los datos a transmitir del nivel de red y forma la primera trama (número de secuencia 0) que se envía al otro extremo. La entidad receptora comprueba la trama recibida y ve si el número de secuencia coincide con el esperado, si es así se desliza una posición la ventana de recepción (que ahora pasaría a ser 1) y se asiente. La figura 9 muestra este procedimiento. En círculo del extremo superior representa la ventana de transmisión de ese lado y el del extremo inferior la de recepción del otro. El primer número que aparece en las tramas es el número de secuencia de la trama, mientras que el segundo es el asentimiento. Existen dos posibilidades para el campo de asentimiento de la trama, o bien se envía el número de la última trama que se recibió correctamente o bien lo que se envía es el número de la primera trama que se está esperando, cualquiera de las dos posibilidades funciona exactamente igual. En el ejemplo se está enviando el número de la trama esperada.

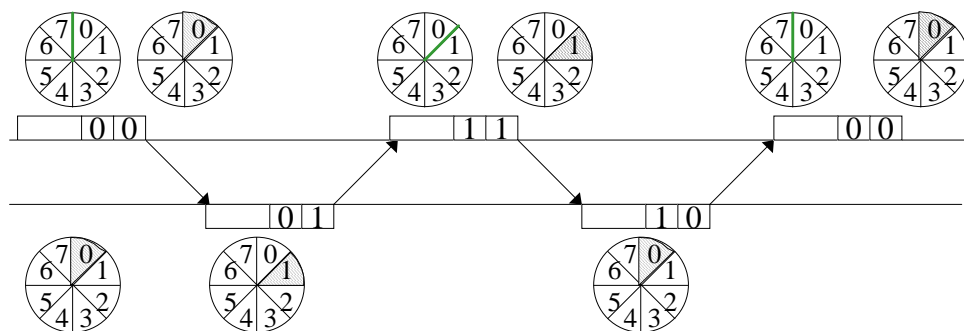


Fig. 9: Protocolo de ventana deslizante de un bit

Si el asentimiento recibido es a la trama que el emisor acaba de enviar entonces se da cuenta que ya se ha recibido correctamente en el otro extremo y vuelve a buscar más información de la capa de red para formar la siguiente trama. Si el asentimiento no es a la trama anteriormente transmitida la retransmite. Puede ocurrir un caso en el que el protocolo descrito no sea eficiente, y es cuando ambos extremos empiecen la transmisión de forma simultánea. La figura 10 muestra este caso.

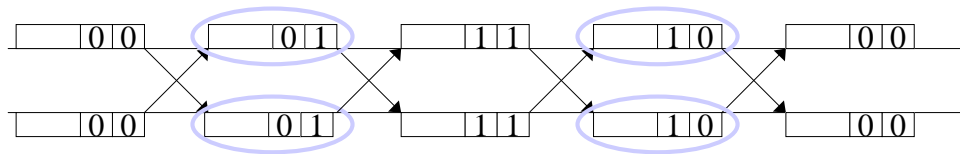


Fig. 10: Protocolo de ventana deslizante de un bit cuando ambos extremos comienzan a transmitir simultáneamente.

*Cuestión 18: Analiza detenidamente la figura 10 ¿Qué problema hay?*

#### 5.4.2 Repetición no selectiva

En este caso el tamaño máximo de la ventana de transmisión es  $N$  y la ventana de recepción tendrá tamaño 1. Se permitirá entonces *enviar  $N$  tramas y tenerlas pendientes de asentimiento*. Si se llena la ventana (se transmiten  $N$  tramas y no se recibe asentimiento a ninguna de ellas) el emisor queda bloqueado esperando asentimientos. Con la adecuada selección de los parámetros el emisor será capaz de transmitir tramas de forma continua y sin tener que parar porque la ventana sature, ya que aunque se gaste un crédito cada vez que se envía una trama se ganará otro cada vez que se reciba un asentimiento. Así el emisor comienza transmitiendo la trama 0, la 1... si antes de llegar a la  $N$  recibe el asentimiento de la 0 puede seguir transmitiendo sin necesidad de esperar, si recibe el asentimiento a la 1 también... de manera que, si todo va bien, irán llegando asentimientos de forma periódica y el transmisor siempre tendrá crédito para transmitir. Esta técnica de envío continuo se denomina también pipelining.

*Cuestión 19: Dibuja dos esquemas del funcionamiento de este protocolo si no hay errores, uno con envío no continuo y otro con envío continuo.*

*Cuestión 20: ¿Qué parámetros influirán para que haya o no envío continuo?*

Ahora bien esto plantea un problema. Si hay  $N$  tramas pendientes de asentimiento (en la ventana de transmisión) y una de ellas llega dañada ¿Qué se hace con el resto de las tramas que llegan correctamente y detrás de la dañada? En este caso la ventana de recepción es 1 y por tanto el receptor *sólo acepta una trama, la que le toca según el orden*. El planteamiento de este protocolo es el llamado de *repetición no selectiva*, lo que quiere decir que el receptor desechará todas las tramas que no sean la esperada en orden, aunque lleguen sin error, y por supuesto dejará de enviar asentimientos porque no han sido aceptadas en recepción. De esta forma la entidad de enlace receptora rechaza cualquier trama excepto la siguiente que debe pasar a la capa de red. Como a partir de la trama que llega con error se dejan de transmitir asentimientos los temporizadores de las tramas de la ventana de transmisión irán venciendo de forma ordenada y se irán retransmitiendo todas las tramas desde la que llegó con error. Este procedimiento, como se muestra en la figura 11, obliga a la retransmisión de todas las tramas que se transmitieron después de una con error, por eso este protocolo es conocido también como *vuelta atrás  $n$  (Go-back- $n$ )*, esto puede suponer un desperdicio considerable de la capacidad del enlace ya que se retransmiten tramas que en realidad llegaron sin error.

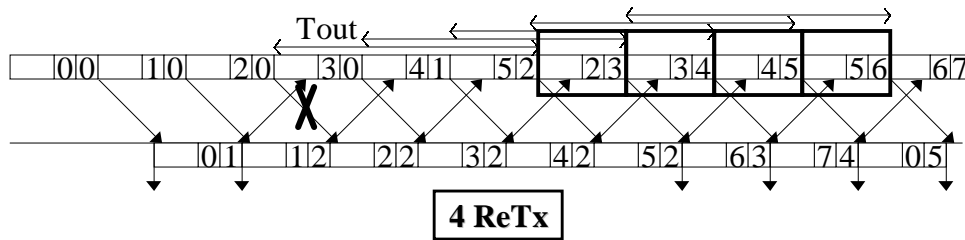


Fig. 11: Protocolo de ventana deslizante con repetición no selectiva y envío continuo.

*Nota: Fíjate como un asentimiento indica al otro lado que esa trama se envió bien al nivel de red*

### 5.4.3 Repetición selectiva

En este caso la ventana de transmisión sigue siendo  $N$  y la ventana receptora tiene un tamaño mayor que 1,  $M$ . El receptor aceptará todas las tramas que pertenecen a la ventana de recepción. Si además la trama que llega es la primera esperada (la que toca según el orden) se enviará al nivel de red, se avanzará la ventana de recepción y se enviará el asentimiento correspondiente. Si no es la esperada en orden no se puede enviar todavía la información al nivel de red, hay que esperar a las anteriores, así que se almacena hasta recibir las anteriores. De este modo si hay un error en una trama el receptor irá almacenando todas las que lleguen después y que pertenezcan a la ventana de recepción.

*Cuestión 21: ¿Por qué no se pasan al nivel de red todas las tramas que lleguen correctas aunque no sea la primera esperada?*

*Cuestión 22: ¿Habrán que asentir inmediatamente las tramas que han llegado bien pero fuera de orden? ¿Por qué?*

Cuando el emisor vea que una trama en concreto se perdió (porque vence el temporizador para esperar el asentimiento correspondiente) se retransmitirá y a partir de ahí, como antes, los temporizadores de las otras tramas también empezarán a vencer ya que, aunque se recibieron bien, no se había recibido asentimiento del otro lado.

Hasta aquí todo parece igual que el protocolo anteriormente mostrado, la diferencia estriba en que cuando se reciba la trama del borde inferior de la ventana de recepción (la esperada en orden) el extremo receptor mandará, en orden, al nivel de red todas las tramas que tenía almacenadas y que ahora sí que puede pasar ordenadas. Entonces enviará al lado emisor un asentimiento indicando la siguiente trama que se espera, que es la siguiente a la última que haya enviado hacia el nivel de red, y actualizará la ventana de recepción. El asentimiento enviado servirá para confirmar al emisor que todas las tramas anteriores efectivamente han llegado bien y ahora seguirá a partir de la última que se pasó correctamente al nivel superior, la figura 12 muestra como en este caso el número de tramas retransmitidas puede reducirse considerablemente.

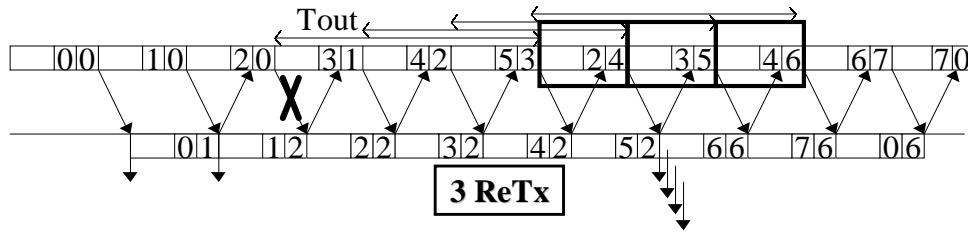


Fig. 12: Protocolo de ventana deslizante con repetición selectiva y envío continuo.

*Cuestión 23: Compara la figura 11 con la 12. ¿Qué diferencias observas?*

*Cuestión 24: Dibuja un diagrama de este protocolo en el que ocurra lo siguiente: Se envían 10 tramas y llegan mal la primera y la quinta...todas las demás llegan bien al otro extremo.*

Por supuesto se necesitará suficiente espacio en la memoria del receptor para almacenar las tramas fuera de secuencia. La mayor ventaja de este protocolo es la posibilidad de añadir **tramas de rechazo (o de asentimiento negativo)**, es decir que el receptor puede enviar una trama especial al emisor indicándole que se ha recibido mal una trama en concreto, de manera que cuando el emisor recibe este asentimiento negativo retransmite exclusivamente la trama que llegó mal. Esta trama, por tanto, es una solicitud de retransmisión de la trama especificada en el asentimiento negativo. En este caso el aprovechamiento del ancho de banda disponible puede ser máximo ya que sólo es necesario retransmitir las tramas que han llegado mal al otro lado.

Es importante considerar el número máximo de secuencia que hay que utilizar. Lo más sencillo es pensar que el número máximo de secuencia coincidiera con el tamaño de la ventana ( $N$ ), sólo se pueden enviar  $N$  tramas seguidas, aunque los números de secuencia posibles son  $N+1$  ( $0, 1, \dots, N$ ).

*Cuestión 25: ¿Por qué  $N+1$  números de secuencia y no  $N$ ?*

Esto va bien cuando la ventana de recepción es 1 pero si es mayor que uno puede ocurrir un problema. Imagine un caso con ventana de transmisión y recepción 3, los números de secuencia irían de 0 a 3 (4). Se transmite la primera ventana bien (las tramas de 0 a 2). Entonces el receptor envía los ACK y avanza su ventana de recepción, ahora esperaría las siguientes, de la 4 a la 6 que tendrían los números de secuencia 3,0,1. Pero ocurre un desastre en la línea y se pierden todos los asentimientos. ¿Qué ocurriría en ese caso? La figura 13 muestra este ejemplo.



## 6. MEDIDA DE LA EFICIENCIA DEL ENLACE

### 6.1 Introducción: Rendimiento y cadencia eficaz.

La eficiencia del enlace se puede medir con dos parámetros íntimamente relacionados, el rendimiento y la cadencia eficaz.

Empecemos definiendo el concepto de *rendimiento* del enlace.

*El rendimiento especifica la relación entre el número de bits transmitidos entre las entidades de red durante un periodo y el número de bits que se pueden transmitir, durante ese periodo, entre las entidades de enlace.*

Es decir la utilización del enlace para transmitir datos útiles y la utilización real que se hace del mismo. En términos temporales indica relación entre el tiempo que se utiliza el enlace exclusivamente para hacer la transmisión de los datos de red y el tiempo que se necesita para que esa transmisión pueda ser efectuada. Si se piensa en una sola trama sería el tiempo que se usa para transmitir los bits del campo de información entre el tiempo que se utiliza el enlace, de forma exclusiva, para que la trama completa llegue correctamente al destino. El tiempo que se utiliza el enlace exclusivamente para transmitir determinados bits útiles se denomina normalmente *Tiempo de Ocupación*.

$$\text{Rendimiento} = \text{Ttx datos útiles} / \text{Tocupación}$$

De manera que si entre 2 entidades de nivel de red se transmiten 1000 bits y la capacidad del enlace que se está utilizando es de 1000 bps el tiempo que efectivamente se utiliza el enlace para hacer la transmisión de datos útiles es de sólo 1 segundo. Sin embargo para que realmente esa información llegue al nivel de red de forma correcta es necesario transmitir también información de control del nivel de enlace, puede que los protocolos obliguen a hacer paradas, quizás haya que retransmitir tramas...lo que generalmente supondrá un tiempo de ocupación mayor. El caso óptimo sería utilizar un protocolo de enlace que no necesitara bits de control, paradas, retransmisiones... en ese caso el tiempo de ocupación coincidiría con el tiempo de transmisión de datos útiles y el rendimiento será 1. Como ya se ha visto a lo largo del tema este tipo de protocolos es impracticable y por tanto el rendimiento del enlace siempre estará por debajo de 1, el reto será conseguir protocolos con el rendimiento lo más alto posible optimizando así la utilización del canal.

Otro parámetro para medir la utilización efectiva que se está haciendo del enlace es la *Cadencia eficaz*. La cadencia eficaz podría definirse como el régimen binario efectivo del enlace, es decir, como los bits por segundo que se están intercambiando las entidades de nivel de red.

$$\text{Cef} = \text{Bits útiles transmitidos} / \text{Tocupación}$$

Como se puede observar este parámetro está muy relacionado con el rendimiento ya que ambos dan una medida de la eficiencia del enlace, aunque utilizan distintas unidades.

*Cuestión 26: ¿Conociendo uno de los dos parámetros cómo obtendría el otro?*

Muchos son los aspectos que influyen en el rendimiento de los protocolos de enlace: El tamaño de las tramas, de las cabeceras, los tamaños de ventana, los temporizadores, el RTD, la probabilidad de error...lo más importante es entender el concepto de rendimiento y calcularlo en función del protocolo concreto que se está estudiando. A continuación se presentan algunos ejemplos para protocolos simples, pero es necesario no olvidar que hay que estudiar cada caso en particular.

## 6.2 Protocolos de Parada y Espera sin errores

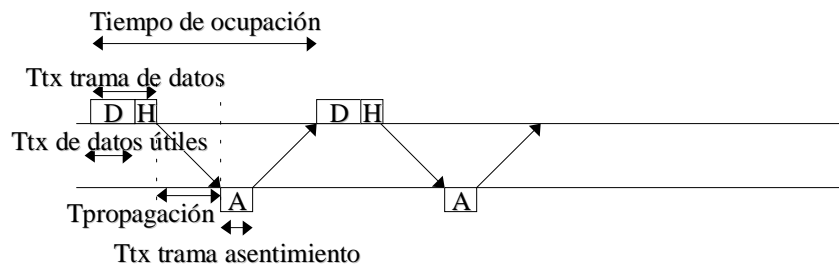


Fig. 14: Parámetros que afectan al rendimiento en un protocolo de parada y espera sin errores.

En este caso se va a considerar un protocolo unilateral de parada y espera con tramas de tamaño fijo. Se va a calcular la capacidad que efectivamente se está utilizando para enviar los datos del nivel de red. Los parámetros considerados, muchos de ellos reflejados en la figura 14 son:

- C= Capacidad del enlace (bits/s)
- D= Número de bits útiles en la trama
- H= Número de bits de cabecera en la trama
- A= Número de bits en la trama de asentimiento

De modo que las entidades de nivel de red se intercambian los datos en bloques de tamaño D (datos útiles de la trama de enlace). La pregunta sería ¿cuánto tiempo es necesario ocupar el enlace para que esos D bits lleguen bien al destino? Es otras palabras ¿Cuál es el tiempo de ocupación del enlace para transmitir esos D bits? Observando la figura 14 se puede ver que en este caso el tiempo de ocupación iría desde que se empieza a transmitir el primer bit de la trama hasta que se recibe el asentimiento correspondiente (el RTD), ya que durante el tiempo en que el emisor está parado el enlace no se puede utilizar para otra cosa que no sea esperar el asentimiento y por tanto se está usando el enlace para soportar los mecanismos necesarios para que la trama llegue correctamente al destino. En este caso por tanto la cadencia eficaz sería  $D/Toc$  y el rendimiento  $D/(C \cdot Toc)$ .

### 6.3 Protocolos de Parada y Espera con errores

Seguimos considerando protocolos de parada y espera unilaterales, pero en este caso añadimos la posibilidad de que ocurran errores. Es necesario considerar que para que una trama llegue correctamente al destino puede ser necesario realizar retransmisiones, ya que algunas tramas pueden llegar dañadas. La figura 15 representa el caso en que ocurre un solo error pero podrían ocurrir más y en ese caso sería necesario realizar más de una retransmisión.

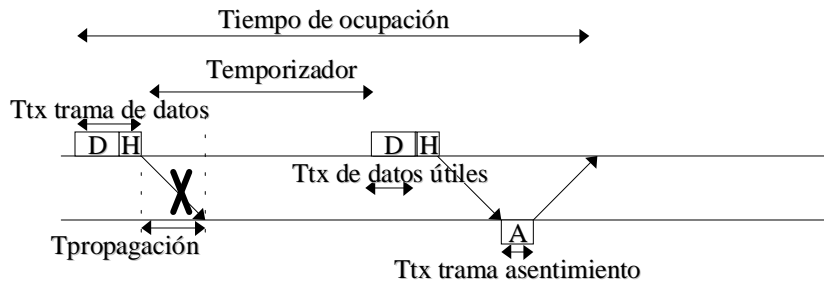


Fig. 15: Protocolo de parada y espera con una trama errónea.

En este caso el tiempo de ocupación no es constante para cada trama ya que hay tramas que se transmiten bien la primera vez, otras que necesitan una retransmisión, dos... es decir que el tiempo de ocupación es una variable aleatoria y no tiene por tanto un valor constante. En este caso es necesario trabajar con el valor medio de esta variable, hay que calcular el valor medio del tiempo de ocupación. La media de una variable aleatoria se calcula realizando la suma de los productos de cada posible valor de la variable por la probabilidad de que ocurra dicho valor.

$$T_{oc} = \sum_{i=0}^{\infty} T_{oci} * P_i$$

El tiempo de ocupación en cada caso dependerá del número de transmisiones que sea necesario hacer para que por fin la trama llegue bien al destino, eso dependerá de la probabilidad de error de la trama. La figura 16 representa el número de transmisiones frente al tiempo de ocupación y la probabilidad de cada caso. Pet es la probabilidad de que una trama llegue con error.

n Transmisiones	Tiempo ocupación con n transmisiones	Probabilidad n transmisiones
1	RTD	1- Pet
2	Ttx+Tout+RTD	Pet*(1-Pet)
3	2* (Ttx+Tout)+RTD	Pet <sup>2</sup> *(1-Pet)
4	3* (Ttx+Tout)+RTD	Pet <sup>3</sup> *(1-Pet)
n	n-1 * (Ttx+Tout)+RTD	Pet <sup>n-1</sup> *(1-Pet)

Fig. 16: Número de transmisiones, tiempo de ocupación y Probabilidad

Habría ahora que calcular la media del tiempo de ocupación que sería:

$$\overline{Toc} = \sum_{i=0}^{\infty} ((i-1) * (T_{tx} + T_{out}) + RTD) * Pet^{i-1} * (1-Pet)$$

*Cuestión 27: ¿Cómo calcularía este sumatorio?*

En este caso el cálculo podría simplificarse observando que el tiempo de ocupación medio será el número medio de veces que una trama se transmite mal por  $T_{tx} + T_{out}$  más lo que se ocupa con la transmisión de la correcta el RTD. Para calcular el número medio de transmisiones por trama sólo hay que hacer la media de la variable aleatoria número de transmisiones (n):

$$\overline{N} = \sum_{n=0}^{\infty} n * P_n = \sum_{n=0}^{\infty} n * Pet^{n-1} * (1-Pet) = (1-Pet) \sum_{n=0}^{\infty} n * Pet^{n-1} = 1/(1-Pet)$$

Si este es el número medio de veces que hay que transmitir una trama para que llegue bien el  $Toc$  quedaría:

$$\overline{Toc} = (\overline{N} - 1) * (T_{tx} + T_{out}) + RTD$$

Para calcular la cadencia eficaz o el rendimiento basta con recordar el número de bits útiles de una trama (D en este caso).

*Cuestión 28: ¿Cómo quedarían la Cef y el rendimiento?*

#### 6.4 Protocolos de ventana deslizante sin errores

Es necesario distinguir en este caso si hay o no envío continuo, la figura 17 muestra el caso con envío continuo. Al contrario que en los protocolos anteriores, en los que había parada, aquí mientras se espera el asentimiento se está usando el enlace para transmitir más tramas, por lo tanto no se puede considerar este tiempo como parte del tiempo de ocupación ya que el canal no se usa en exclusiva para que la trama que se está considerando llegue al otro lado. Por tanto el tiempo que se ocupa el canal de forma exclusiva para que una trama llegue bien al receptor es en este caso el tiempo de transmisión de la trama ya que el resto del tiempo el canal se usa para transmitir otras tramas y nunca hay retransmisiones porque no hay errores.

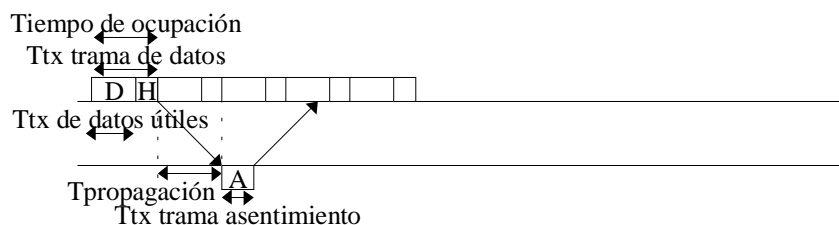


Fig. 17: Protocolo de ventana deslizante sin errores y con envío continuo.

En caso de que no existiera envío continuo y el transmisor quedara detenido en algún momento este tiempo que el emisor está detenido sí que hay que considerarlo como parte del tiempo de ocupación, ya que durante ese tiempo el enlace no puede ser utilizado para enviar nada. La figura 18 muestra este caso. Para hacer un cálculo más sencillo se podría utilizar toda la ventana para calcular el tiempo de ocupación, en ese caso el número de datos útiles transmitidos no sería los datos transmitidos en una trama si no los transmitidos en toda la ventana.

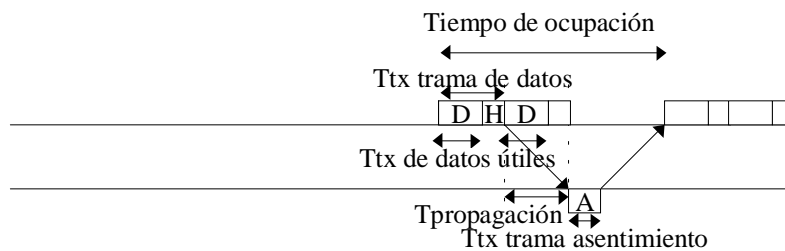


Fig. 18: Protocolo de ventana deslizante sin errores y sin envío continuo.

*Cuestión 29: ¿Cómo quedarían la Cef y el rendimiento?*

### 6.5 Protocolos de ventana deslizante con errores

Son más los factores que hay que considerar ahora y por tanto es necesario estudiar cada protocolo detenidamente. Habrá que estudiar si el protocolo es con rechazo selectivo o simple. La figura 19 muestra el caso de rechazo simple. En esta ocasión como cada vez que una trama llega errónea hay que retransmitir todas las tramas que se transmitieron después de ésta el tiempo de ocupación para la transmisión de una trama será el número medio de transmisiones incorrectas por el tiempo que se usa en transmitir la trama incorrecta y todas las que después habrá que retransmitir también (ya que la transmisión de todas las demás es tiempo perdido en este caso) más el tiempo de transmisión de la trama correcta.

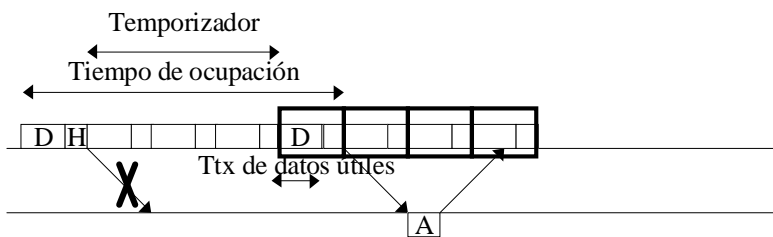


Fig. 19: Protocolo de ventana deslizante con rechazo simple y envío continuo cuando hay errores

Si se logra envío continuo, rechazo selectivo y se usan tramas de asentimiento negativo el rendimiento puede aumentar mucho.

*Cuestión 30: ¿Cómo quedarían la Cef y el rendimiento en estos dos casos?*

## 7. EJEMPLOS: HDLC

### 7.1 Introducción:

Se han estudiado hasta aquí los mecanismos básicos de los protocolos de nivel de enlace, sin especificar ninguna implementación concreta de éstos. Se podrían dividir las implementaciones de protocolos de enlace existentes en dos grandes grupos, *protocolos asíncronos y protocolos síncronos*. En el primer caso la transmisión entre entidades de enlace es asíncrona, es decir que una unidad de datos se transmite entre ambos extremos sin coordinación del tiempo ya que no existe un reloj común entre transmisor y receptor. Esto obliga al uso de bits adicionales para controlar el principio y fin de la transmisión de información y para facilitar la extracción de datos. De manera que estos protocolos tienen problemas de velocidad y han quedado prácticamente relegados al uso en los módems sustituyéndose por protocolos síncronos en otros entornos que necesitan más eficiencia como LAN, MAN o WAN. Dentro de los protocolos asíncronos pueden destacarse XMODEM, YMODEM, ZMODEM, BLAST o Kermit.

En cuanto a los protocolos síncronos se dividen en dos grandes grupos, los *protocolos orientados a carácter y los orientados a bit*. Los primeros interpretan una trama como una sucesión de caracteres y toda la información de control está en un formato de algún sistema de codificación de carácter (p.e. ASCII). De los protocolos de este tipo el más conocido es el BSC (Binary synchronous communication) desarrollado por IBM. En cuanto a los segundos, los orientados a bit, interpretan la trama como una sucesión de bits cuyo significado se extrae según su posición en la trama, que por supuesto tiene un formato bien definido. De este modo la información de control puede estar dada por uno varios bits, según las necesidades. Los más conocidos de este tipo son SDLC, HDLC y la familia LAP. Estos protocolos son más eficientes que los protocolos orientados a carácter y son los más utilizados en la actualidad, por ello estudiaremos más en profundidad un protocolo de esta familia: HDLC

### 7.2 Generalidades

HDLC (High data link control) se basa en el usado en la capa de enlace en SNA (SDLC). ANSI modificó este protocolo y generó el ADCCP que más tarde la ISO convierte en el HDLC. Posteriormente la UIT-T lo modifica y especifica la familia LAP (B,D,M...). Muchas de las ideas que se ven en este apartado son comunes a todos estos protocolos, aunque las explicaciones están centradas en HDLC.

Todos estos protocolos están basados en los mismos principios. Son orientados a bit, como se ha especificado, y utilizan el mecanismo de inserción de bit, estudiado al principio de este tema, para lograr la transparencia de datos. La estructura de la trama es también común:

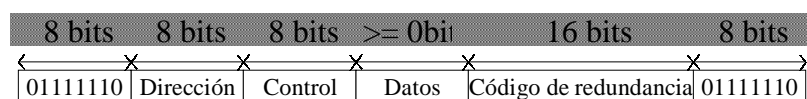


Fig. 20: Formato general de trama

- Las banderas indican el principio y fin de trama, si las tramas se transmiten consecutivamente la bandera de inicio de una podría sustituir a la de fin de la anterior, ahorrándose así la transmisión de 8 bits. Las banderas se transmiten continuamente si la línea está inactiva.
- El campo dirección se utiliza fundamental en líneas multipunto, es decir, cuando el enlace está compartido por varios terminales, como muestra la figura 21. En este caso la dirección se emplea para identificar el terminal. Si fuera necesario utilizar más de un octeto podría utilizarse. En caso de necesitar más de un octeto el primer bit del último octeto de dirección irá a 1 y en todos los demás octetos de dirección el primer bit irá a cero. En las líneas punto, a punto, dos terminales directamente unidos como indica la figura 22, se usa para distinguir los comandos (órdenes) de las respuestas, la dirección que se envía siempre es la del que trabaja como secundario (el destino en las órdenes o el origen en las respuestas). En el siguiente apartado se estudian las configuraciones básicas y se revisará esta idea. El uso de este campo es como sigue:
  - 00000000 Dirección Nula
  - 0xxxxxxx Puede haber más octetos
  - 1xxxxxxx Octeto final (dirección extendida)
  - 11111111 Difusión
- El campo de control se usa para los números de secuencia, asentimientos e información relativa a otras tareas que necesiten realizar las entidades de enlace.
- En el campo de datos se envía la información del nivel de red (y de enlace en algunas tramas de control que necesiten información). Puede ser arbitrariamente largo, aunque la eficiencia del código de redundancia decrecerá a medida que se aumente la longitud de la trama, ya que la probabilidad de errores aumentará. En el campo de redundancia se utiliza una variante del código de redundancia cíclico CRC dado por la UIT-T y este código controla los posibles errores tanto en los datos como en las cabeceras.

### 7.3 Configuraciones básicas y modos de operación

Las distintas variantes del protocolo HDLC pueden funcionar con distintas configuraciones básicas:

- **Configuración no equilibrada o no balanceada** (Mostrada en la figura 21): Un equipo trabaja como principal, dando órdenes y los demás como secundarios, respondiendo a estos comandos. El enlace tiene que ser compartido entre todos los secundarios y para lograr esto se utilizan ciclos de sondeo y selección gobernados por el principal. En el ciclo de sondeo el principal recoge datos de los secundarios, uno a uno, en el de selección el principal envía datos a los secundarios. La dirección que se envía en las tramas siempre es la del terminal secundario al que se envía la orden o que responde.

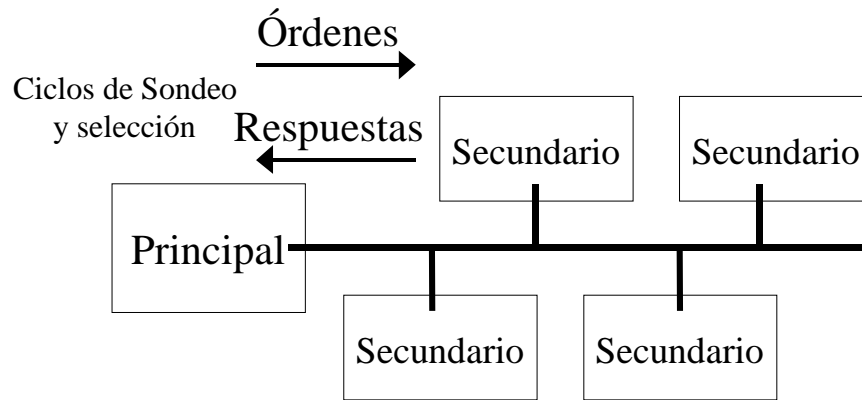


Fig. 21: Configuración NO equilibrada

- **Configuración equilibrada o balanceada** (Mostrada en la figura 22): en este caso se tiene una configuración punto a punto en la que ambos lados pueden intercambiar el papel de principal y secundario, de manera que se envían órdenes y respuestas según las necesidades. Siempre se enviará la dirección del secundario (el que recibe la orden o envía la respuesta).

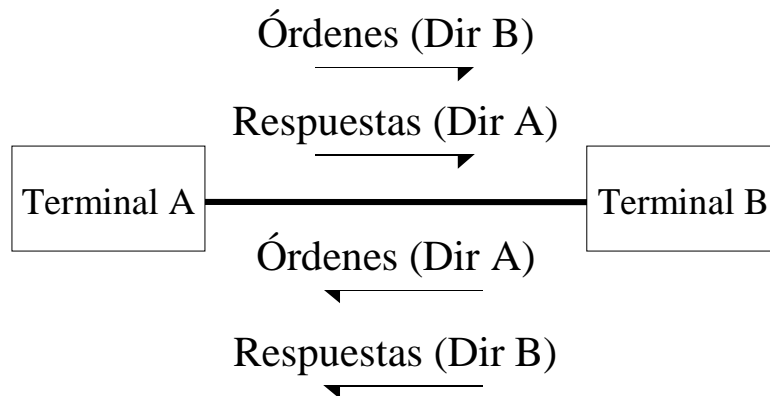


Fig. 22: Configuración Equilibrada

El término asíncrono se utiliza para indicar que las estaciones no necesitan recibir una señal preliminar procedente de otra estación antes de comenzar a transmitir tráfico. Es decir no se necesita ninguna autorización antes de transmitir cualquier trama, no existe sondeo.

Sobre estas configuraciones básicas se pueden utilizar distintos modos de operación:

**Modo Normal (NRM o UN, Normal Response Mode o Unbalanced Normal):** Utilizado en enlaces multipunto (1 principal y N secundarias), muy normal en entornos centralizados. Evidentemente la configuración de este modo será la multipunto.

**Modo Asíncrono no balanceado (ARM o UA, Asynchronous Response Mode o Unbalanced Asynchronous):** Usado en enlaces punto a punto, se basará en utilizar la configuración punto a punto pero trabajando siempre uno como principal y el otro como secundario. En este caso los secundarios pueden transmitir sin permiso del principal ya que no habría problema de colisión al tener líneas punto a punto.

**Modo Asíncrono Balanceado (ABM o BA, Asynchornous Balanced Mode o Balanced Asinchronous):** Se utiliza también en enlaces punto a punto. En este caso las dos estaciones alternan los papeles de principal y secundario según sea conveniente (balanceado).

#### 7.4 Tramas HDLC

Hay tres tipos de tramas:

- Información: Transportan información del nivel de red en el campo información.
- Supervisora: Se utilizan para asentir tramas (positiva y negativamente) y controlar el flujo en el enlace. También informan del estado a la entidad de enlace del otro lado.
- Sin numerar: En general tienen propósitos de control del enlace de datos que veremos más adelante.

En la figura 23 se presenta el campo de control en cada una de estas tramas.

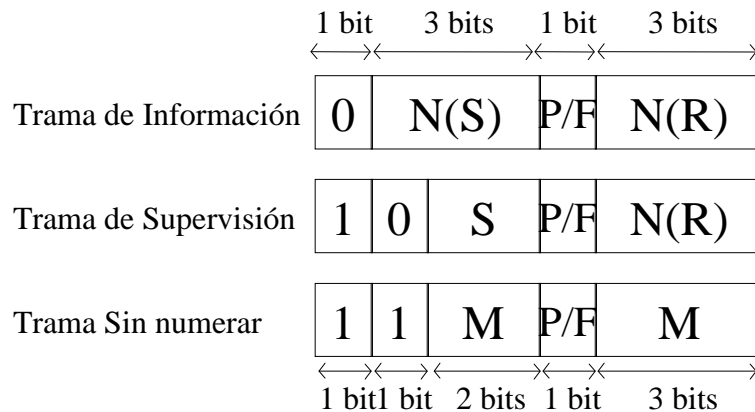


Fig. 23: Campo de control de las tramas HDLC

Como se observa en la figura el campo de “número de secuencia” de las tramas de información, N(S), tiene tres bits y se utiliza para indicar el orden de la trama dentro de una secuencia de tramas (facilita la entrega en orden al nivel de red y la identificación de tramas perdidas o erróneas).

*Cuestión 31: ¿Cuál sería el tamaño máximo de la ventana de transmisión?*

El campo “número de asentimiento” N(R), que aparece en las tramas de información y las de supervisión, se utiliza para enviar asentimientos superpuestos. En este campo se pone **el número de secuencia de la siguiente trama que se espera**. Por otro lado el bit P/F, pregunta/final o comando/respuesta, se usa para tareas de disciplina de línea consiguiendo así una gestión adecuada del enlace. Solamente tiene sentido cuando está activo (bit=1) y significa P si lo activa una estación principal (es decir si va en una orden) o F si lo activa una secundaria (es decir si va en una respuesta). El funcionamiento es sencillo y depende de la configuración que se esté usando:

**En configuración multipunto:** Durante el ciclo de sondeo el concentrador invita a un terminal a enviar datos (orden) enviándole una trama con el bit P activo (bit a 1). A partir de ese momento el terminal empieza a enviar datos y en la última trama

(trama de respuesta) activa el bit F (lo pone a uno) de manera que así el principal sabe que ya no le va a enviar más datos y puede pasar a sondear otro terminal. Puede considerarse como la existencia de un testigo que da permiso para transmitir sobre el enlace, con el bit P el principal envía el testigo a un secundario concreto, con el bit F el secundario devuelve el testigo al principal. Como puede observarse el bit P/F es el mismo, simplemente cuando lo activa el equipo principal es P (es decir en una orden significa P) mientras que cuando lo activa el secundario es F (es decir en una respuesta significa F).

**En configuración punto a punto:** En este caso se suele utilizar el bit P/F para preguntar sobre el estado de la entidad de enlace gemela. De manera que si un lado activa el bit P cuando está enviando una orden, o comando, obliga al otro lado a enviar una trama de respuesta con el bit F activo. De nuevo se puede considerar que el que actúa de principal utiliza el bit como P mientras que el que actúa de secundario lo utiliza como F.

*Cuestión 32: Piense en un caso en el que a uno de los dos lados le interese activar el bit P*

#### 7.4.1 Tramas supervisoras

Existen 4 tipos de tramas supervisoras que se distinguen por el campo S (Tipo), según la figura 23:

- **00: RR Receptor listo**, indica que la siguiente trama esperada es la que tiene el número de secuencia indicado en el campo N(R). Se usa cuando no hay tráfico en sentido contrario para la superposición del asentimiento a la información.

*Cuestión 33: ¿Por qué es necesario esta trama y no simplemente se espera a que haya algo que transmitir en el otro sentido?*

- **01: REJ Asentimiento negativo o rechazo**. Indica que la trama con el número de secuencia indicado en N(R) no se recibió bien y solicita al otro lado que se retransmita *esa y todas las tramas que se hayan enviado después*.

*Cuestión 34: Este tipo de retransmisión es característica de protocolos con ¿Qué tamaño de ventana de recepción? ¿Qué tipo de rechazo es?*

- **10: RNR Receptor no listo**. Es una trama de asentimiento positivo, indica que la siguiente trama que se espera es la que va en el campo N(R), y por tanto todas las anteriores se recibieron bien. Pero en este caso se solicita al otro lado que deje de transmitir. Sirve por tanto como un mecanismo de control de flujo, que asegura que si hay algún problema en el receptor el emisor no lo inundará con tramas que no se podrán gestionar. Cuando de nuevo el receptor está listo se enviará alguna trama supervisora para que continúe la comunicación (RR,REJ...)

- **11: RS Rechazo selectivo.** Con esta trama se solicita la retransmisión, exclusivamente, de la trama que se indica en el campo N(R). Es una trama exclusiva de HDLC (No existe en SDLC ni LAPB)

*Cuestión 35: ¿Qué tamaño de ventana de recepción se necesita? Relaciónelo con los números de secuencia.*

**7.4.2 Tramas sin numerar**

Las tramas sin numerar se utilizan normalmente para propósitos de control, es decir que no transmiten información del nivel de red (salvo la trama UI que veremos más adelante). No se utilizan todas siempre si no que depende de la variante de protocolo que se esté utilizando, como se vio anteriormente existen distintos modos de operación en HDLC y como se verá inmediatamente distintos procedimientos, según el que se esté usando se usarán unas tramas u otras. En la figura 24 se representa una tabla con todas las tramas HDLC, y se pueden observar los distintos tipos de tramas sin numerar que hay.

TIPO	COMANDO		CAMPO DE CONTROL						RESPUESTA ¿BÁSICO?		MODO DE OPERACIÓN
	I	0	SEC ENVIÓ	P/F	SEC RECEP	I	SÍ				
<b>INFORMACIÓN</b>	RR	1	0	0	0	P/F	SEC RECEP	RR	SÍ		
	REJ	1	0	0	1	P/F	SEC RECEP	REJ			
	RNR	1	0	1	0	P/F	SEC RECEP	RNR	SÍ		
	SREJ	1	0	1	1	P/F	SEC RECEP	SREJ			
<b>SUPERVISORA</b>	UI	1	1	0	0	P/F	0 0 0	UI			
	SNRM	1	1	0	0	P	0 0 1			UN	
	DISC	1	1	0	0	P/F	0 1 0	RD	SÍ		
	UP	1	1	0	0	P	1 0 0				
		1	1	0	0	F	1 1 0	UA	SÍ		
	SIM	1	1	1	0	P/F	0 0 0	RIM			
		1	1	1	1	F	0 0 1	FRMR	SÍ		
	SARM	1	1	1	1	P/F	0 0 0	DM	SÍ(DM)	UA	
	RSET	1	1	1	1	P	0 0 1				
	SARME	1	1	1	1	P	0 1 0			UA (Nume. Extendida)	
<b>SIN NUMERAR</b>	SNRME	1	1	1	1	P	0 1 1			UN (Nume. Extendida)	
	SABM	1	1	1	1	P	1 0 0			AB	
	XID	1	1	1	1	P/F	1 0 1	XID			
	SABME	1	1	1	1	P	1 1 0			AB (Nume. Extendida)	

Fig. 24: Tramas HDLC

*Cuestión 36: Compara la figura 22 y la 21, observa que los bits S y M del campo de control determinan el tipo de trama.*

*Cuestión 37: Observa el campo bit P/F de las tramas. ¿Por qué en todas las tramas que sólo pueden ser comandos este bit sólo puede ser P? ¿Por qué en todas las tramas que sólo pueden ser respuestas este bit sólo puede ser F?*

A continuación se verán algunas de las tramas sin numerar más importantes.

**FRMR:** Llegó trama con semántica errónea, no entendible por el receptor, pero con CRC correcto. Por ejemplo se envía si se recibe un asentimiento de trama fuera de ventana, una trama supervisora con menos de 32bits, una trama de control inexistente en el protocolo que se está usando...

**XID:** Trama para el intercambio de identificadores.

**UA:** Asentimiento sin numerar, para asentir las tramas de control. Las tramas de control también pueden perderse o dañarse y es necesario que el otro extremo confirme que se recibieron bien. Evidentemente en este caso no se necesita numeración porque las tramas de control no van numeradas y sólo una trama de control puede estar pendiente de ser asentida en un momento dado

**UI:** Información sin numerar, se utiliza cuando una entidad de capa 3 (red) solicita la transferencia de información sin acuse de recibo. Se pueden perder sin notificación ya que no llevan número de secuencia y no se asienten. El servicio de transmisión de datos ofrecido por la capa de enlace si se usa esta trama es sin conexión ni asentimiento.

Hay una serie de tramas que sirven para el establecimiento de la conexión, con ellas los extremos reinician números de secuencia, contadores, temporizadores y todos los parámetros que necesiten usar durante la conexión. Según el modo de operación de HDLC con el que se vaya a funcionar se utilizará una trama de conexión u otra.

**SARM:** Establecimiento del modo asíncrono no balanceado.

**SARME:** Asíncrono no balanceado con numeración extendida (7 bits para número de secuencia, ventana de transmisión mayor)

**SNRM:** Establecimiento del modo normal (no balanceado)

**SNRME:** Modo normal con numeración extendida (7 bits para secuencia, ventana de transmisión mayor)

**SABM:** Establecimiento del modo asíncrono balanceado

**SABME:** Modo asíncrono balanceado con numeración extendida (7 bits para secuencia, ventana de transmisión mayor)

Otro conjunto de tramas sirve para desconectar

**DISC:** Uno de los extremos anuncia que se va a desconectar

**RD:** Un terminal secundario pide al principal que envíe un DISC

**DM:** Respuesta negativa a una orden de activación de conexión o positiva a una orden de desconexión.

Como se dijo anteriormente existen distintas clases de procedimientos en HDLC, la figura 25 representa estos procedimientos con las tramas y restricciones concretas de cada uno de ellos. Puede observarse que hay una serie de tramas básicas que se utilizan siempre (en la figura 24 se señala cuáles son). Un protocolo se identifica por el modo de operación (UN, UA, BA) y una serie de números que indican los procedimientos que se utilizan, mostrados en la figura 25, que indican restricciones y tramas concretas.

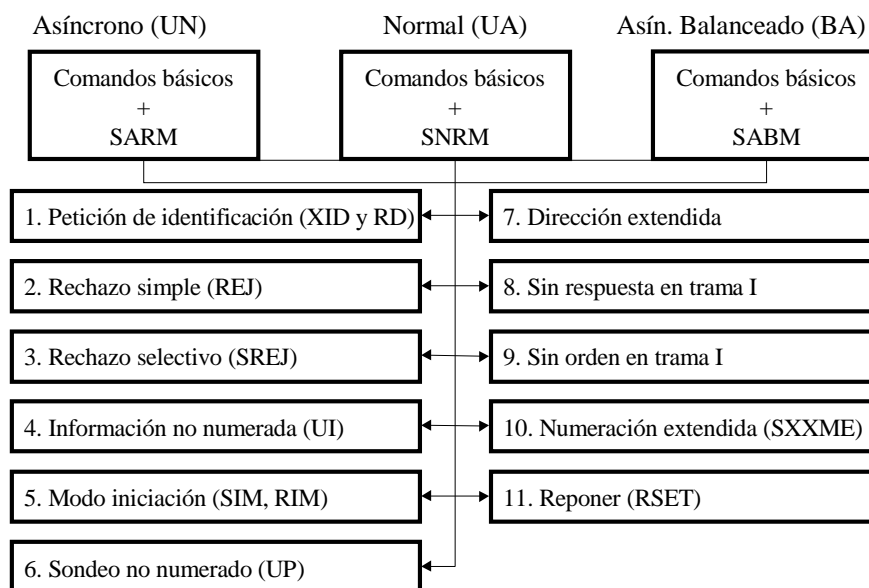


Fig 25: Tramas y restricciones según el procedimiento

Así si se utiliza BA 2,8 se estará utilizando modo de operación balanceado, con rechazo simple y que no se pueden enviar respuestas (es decir activar el bit F) en las tramas I.

Si se usa UN 3,4 se estará utilizando modo de operación normal, con rechazo selectivo y tramas de información no numerada.

## 7.5 Ejemplos

Los dos protocolos anteriormente señalados son los más utilizados y el resto de figuras son ejemplos que representan casos particulares de los mismos. En todos ellos las tramas se representan de la siguiente forma. En primer lugar se indica la dirección que va en la trama, a continuación el tipo de trama y por último se indica si está activo el bit P o el F. En las tramas de información se indican, además, el número de secuencia de la trama N(S) y el asentimiento N(R). En tramas supervisoras se indica el campo N(R)

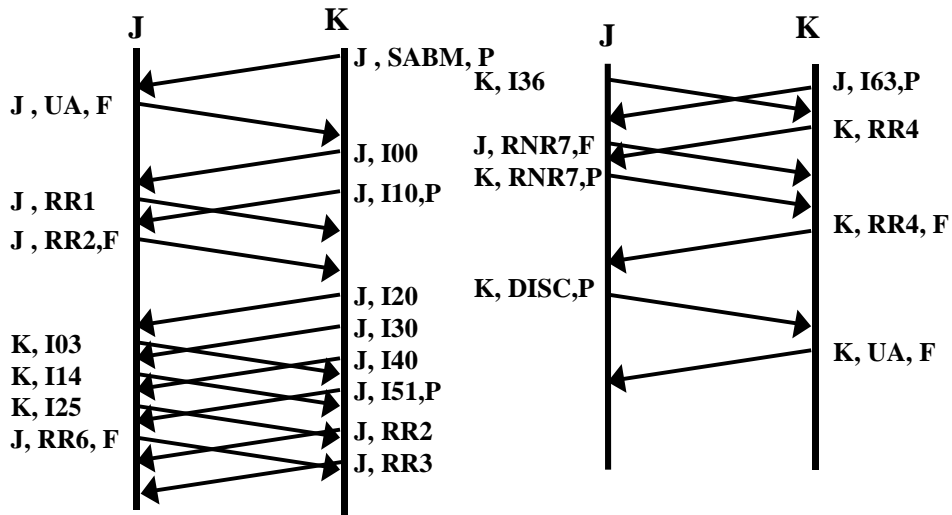


Fig. 26: Ejemplo 1: Protocolo BA 2,8 sin errores

*Cuestión 38: Fíjate en el uso de los bits P/F, ¿Cómo se responde a los bits P activos? ¿Cuándo se activa el bit P qué dirección se usa? ¿Y cuándo se activa el bit F?*

*Cuestión 39: Identifica las restricciones o tramas de los procedimientos 2,8*

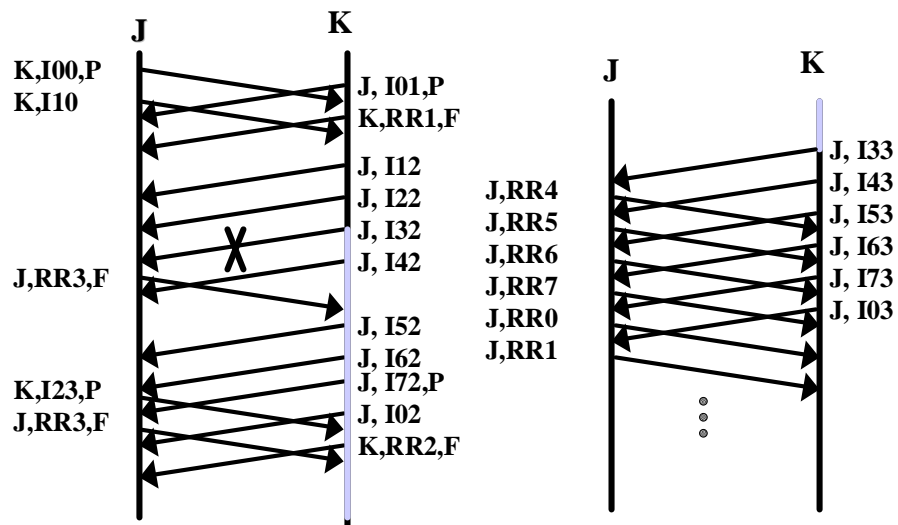


Fig. 27: Ejemplo 2: Protocolo BA 2,8 con errores

*Cuestión 40: Fíjese en la reacción a fallos: ¿Cuándo falla una trama cuáles se retransmiten? ¿Qué ventana de recepción hay? ¿Y de transmisión?*

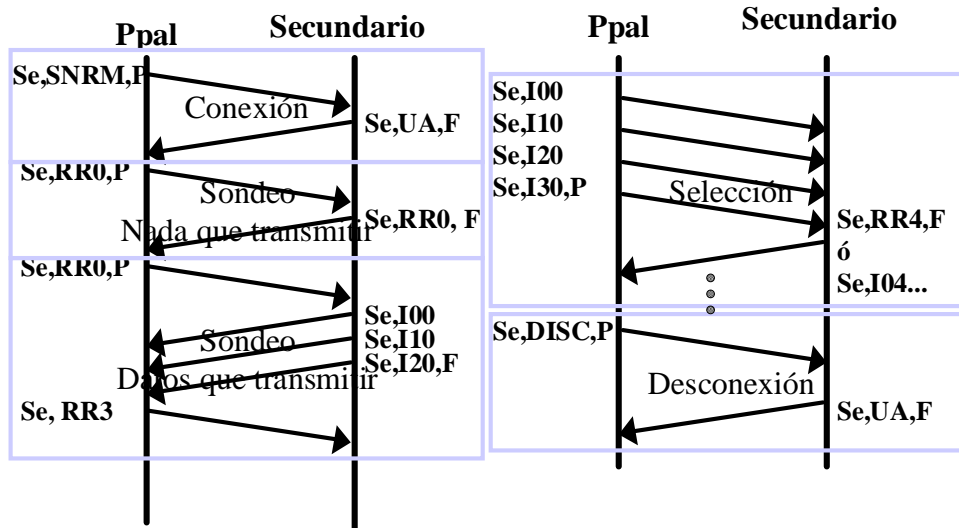


Fig. 28: Ejemplo 3: Protocolo UN 3,4 sin errores

Cuestión 41: Identifique las restricciones o tramas de los procedimientos 3,4

Cuestión 42: Identifique los ciclos de sondeo y selección. ¿Cuándo empieza el ciclo de sondeo? ¿Cuándo termina? ¿Quién envía la información en el ciclo de sondeo? ¿Y en el de selección?

Cuestión 43: ¿Qué pasaría si fuera el secundario el que quisiera desconectar? Repase las tramas sin numerar que se usan para desconectar.

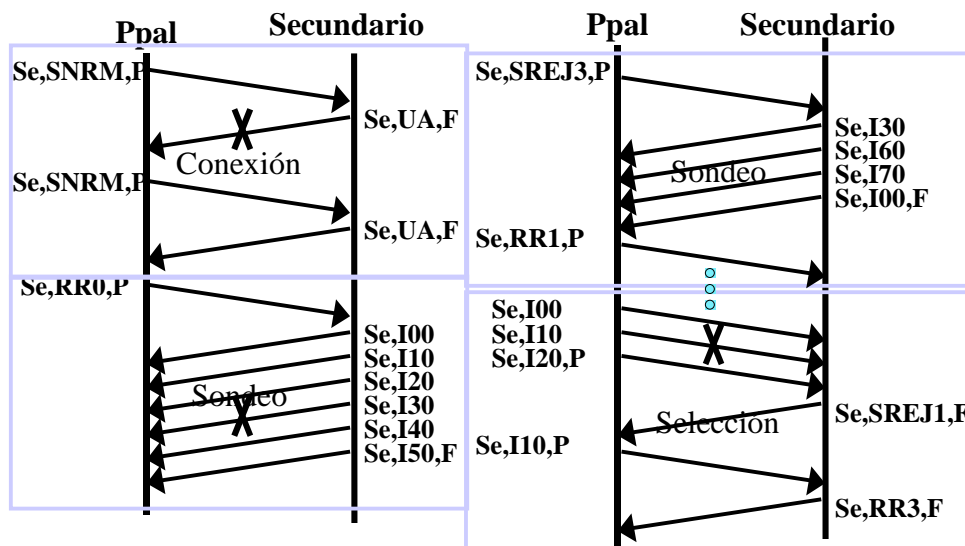


Fig. 29: Ejemplo 4: Protocolo UN 3,4 con errores

Cuestión 44: ¿Qué ocurre aquí cuando una trama llega mal? ¿Cuáles hay que retransmitir?

Los protocolos de nivel de enlace más utilizados en la actualidad están basados en HDLC. La tabla que se muestra a continuación refleja algunos protocolos de nivel de enlace y dónde se utilizan principalmente.

<i>Protocolo (Siglas)</i>	<i>Protocolo (Nombre)</i>	<i>Equivalente HDLC</i>	<i>Norma/Usa</i>
HDLC	High Level Data Link Control	HDLC	ISO 13239
ADCCP	Advanced Data Communications Control Procedure		ANSI x3.66
LLC	Logical Link Control	BA 2,4	IEEE 802.2
LAP-B	Link Access Procedure Balanced	BA 2,8	X.25 (RPDCP)
LAP-D	Link Acces Procedure D-channel	BA 1,2,4,7,8,10	Q.921 (RDSI)
LAP-F	Link Access Procedure for Frame Mode Bearer Services	BA 1,2,4,7,10	Q.922 (Frame Relay)
LAP-M	Link Access Procedure- Modem		Modems RTC (V.32,V.34...)
PPP	Point to Point Protocol		Internet

Fig. 30 Protocolos nivel de Enlace y relación con HDLC

## 8. CONCLUSIONES

Al final de este tema el alumno debe tener claras las funciones del nivel de enlace y su papel dentro de los modelos de capas. Debe dominar los conceptos básicos de los protocolos de nivel 2; control de errores y de flujo, ventanas de transmisión y recepción, números de secuencia y asentimientos, cadencia eficaz y rendimiento...

Como caso particular se presenta el protocolo HDLC que servirá muy bien como base para el estudio de otros protocolos de nivel 2 ya que es el origen de los más utilizados actualmente.

*Cuestión 45: Según lo visto de estos protocolos ¿Se asegura la transmisión fiable entre dos aplicaciones remotas comunicándose a través de una WAN? ¿Por qué?*

### Cuestiones

1. Se utiliza un protocolo de nivel de enlace con tramas de 1016 bits y cabeceras de 2 octetos. Cuando la trama llega correctamente se envía como asentimiento una trama sin datos. El enlace que conecta dos equipos tiene una capacidad de 2'048 Mb/s, una longitud de 5 metros y la velocidad de propagación en el medio es  $2 \cdot 10^8$  m/s. Suponiendo que no hay errores calcule el RTD del enlace.
2. En un protocolo de nivel de enlace se tienen los siguientes valores: Tiempo de transmisión de trama (Ttx), Tiempo de propagación (Tp), Tiempo de asentimiento de trama (Desde que se envía el último bit de la trama hasta que se recibe el asentimiento correspondiente, Tas) y Temporizador de retransmisión de trama igual al tiempo de asentimiento de trama (Tas)
  - a) ¿Cuál será el tamaño mínimo de la ventana de transmisión para que haya envío continuo? Justifique la respuesta.
  - b) Suponiendo que se utiliza justamente la ventana de transmisión del apartado anterior calcule el tamaño de la ventana de recepción necesario para mejorar el rendimiento respecto al que se tiene con una ventana de recepción de uno. Justifique la respuesta
3. El siguiente cuadro muestra el formato del campo de control en las tramas del protocolo LAPD, utilizado en RDSI.

**Cuadro 4/Q.921 – Formatos de campo de control**

Bits del campo de control (módulo 128)	8	7	6	5	4	3	2	1	
Formato I	N(S)							0	Octeto 4
	N(R)							P	5
Formato S	X	X	X	X	S	S	0	1	Octeto 4
	N(R)							P/F	5
Formato U	M	M	M	P/F	M	M	1	1	Octeto 4
N(S) Número secuencial en emisión del transmisor	M			Bit de la función de modificación					
N(R) Número secuencial en recepción del transmisor	P/F			Bit de petición cuando se transmite como instrucción; bit final cuando se transmite como respuesta					
S Bit de la función de supervisión	X			Reservado y puesto a 0					

Sabiendo que está basado en **HDLC** y que es un protocolo **balanceado** indique el número máximo de tramas pendientes de asentimiento que podría llegar a soportar el protocolo y qué trama de inicio de conexión se va a utilizar en LAPD.

