

Introducción. En este capítulo se estudiarán los principios de diseño de la capa 2. Esto tiene que ver con algoritmos para lograr una comunicación confiable y eficiente entre 2 máquinas adyacentes. La propiedad esencial es que los bits se entregan con exactitud en el mismo orden en que fueron enviados, en analogía con un alambre.

Los circuitos de comunicación cometen errores ocasionales. Además tienen una tasa de datos finita y hay un retardo de propagación diferente de cero entre el momento que se envía un bit y el momento en que se recibe. Estas limitaciones tienen implicaciones importantes en la eficiencia de la transmisión de datos. Los protocolos usados para comunicaciones deben considerar todos estos factores.

3.1 Cuestiones de diseño de la capa de enlace de datos.

La capa de enlace tiene que desempeñar varias funciones específicas, entre ellas:

- Proporcionar una interfaz de servicio bien definida con la capa de red. (ENTRAMAR).
- Manejar los errores de transmisión.
- Regular el flujo de datos para que receptores lentos no sean saturados por emisores rápidos.

Para ello, la capa de enlace *toma los paquetes de la capa de red* y los encapsula en **tramas** para transmitirlos. Cada trama contiene un **encabezado**, un **campo de carga útil** (payload) para almacenar el paquete y un **terminador** o final. El manejo de las tramas es la tarea primordial de la capa de enlace de datos.

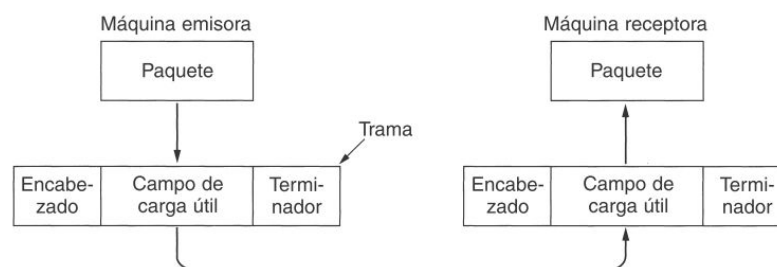


Figura 3-1. Relación entre los paquetes y las tramas.

3.1.1 Servicios proporcionados a la capa de red.

La función de la capa de enlace es suministrar servicios a la capa de red. El servicio principal es transferir datos de la capa de red desde la máquina origen a la de destino.

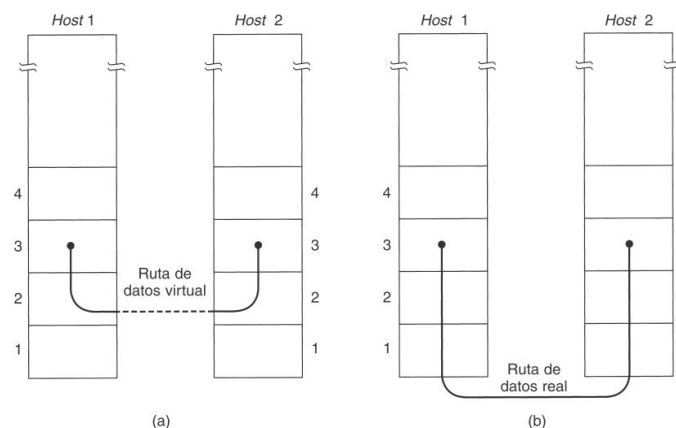


Figura 3-2. (a) Comunicación virtual. (b) Comunicación real.

La capa de enlace de datos puede diseñarse para ofrecer varios servicios. Tres posibilidades razonables que normalmente se proporcionan son:

- 1-Servicio no orientado a la conexión sin confirmación de recepción.
- 2-Servicio no orientado a la conexión con confirmación de recepción.
- 3-Servicio orientado a la conexión con confirmación de recepción.

El Servicio no orientado a la conexión sin confirmación de recepción (1) consiste en hacer que la máquina de origen **envíe tramas independientes** a la máquina destino sin pedir que ésta confirme la recepción. No se establece conexión de antemano ni se libera después. Si se pierde una trama por ruido, en la capa de enlace no se realiza ningún intento por detectar la pérdida ni por recuperarse de ella. Esta clase de servicio es apropiado cuando la tasa de errores es muy baja, por lo que la recuperación se deja a las capas superiores. También es apropiado para el tráfico en tiempo real. (LANs).

En el servicio no orientado a la conexión con confirmación de recepción (2), tampoco se utilizan conexiones lógicas, pero se **confirma de manera individual la recepción de cada trama enviada**, así el emisor sabe si la trama ha llegado bien o no. Si no ha llegado en un tiempo especificado, **puede enviarse nuevamente**. Es útil en canales inestables, como los de sistemas inalámbricos.

La capa de red siempre puede enviar un paquete y esperar que se confirme su recepción. Si la confirmación no llega antes de que expire el temporizador, el emisor puede volver a enviar el mensaje. El problema con esta estrategia es que las tramas tienen una longitud máxima impuesta por el HW mientras que los paquetes de la capa de red no la tienen.

Con el servicio orientado a la conexión (3), la máquina de origen y de destino establecen una conexión antes de transferir datos. Cada **trama** enviada a través de la conexión está **numerada**, y la capa de enlace **garantiza que cada trama enviada llegue a su destino**. Es más, garantiza que cada trama será recibida exactamente **una vez** y que todas las tramas se recibirán en el orden adecuado. En contraste, con el servicio no orientado a la conexión es posible que una confirmación de recepción perdida cause que una trama se envíe varias veces y, por lo tanto, que se reciba varias veces. Por suerte, el servicio orientado a la conexión proporciona a los procesos de la capa de red el equivalente de un flujo de bits confiables.

En el servicio orientado a la conexión las transferencias tienen 3 fases distintas. La primera, la conexión se establece haciendo que ambos lados inicialicen las variables y los contadores necesarios para seguir la pista de las tramas que han sido recibidas y las que no. En la segunda fase se transmiten una o más tramas. En la tercera fase, la conexión se cierra y libera las variables, los búferes y otros recursos utilizados para mantener la conexión.

3.1.2 Entramado. A fin de proporcionar servicios a la capa de red, la de enlace debe utilizar los servicios que la capa física proporciona. Lo que hace la capa física es aceptar un flujo de bits puros e intentar entregarlo al destino. No se garantiza que este flujo de bits esté libre de errores. Es responsabilidad de la capa de enlace detectar y corregir los errores.

El método común es que la capa de enlace de datos divida el flujo de bits en tramas separadas y que calcule la suma de verificación de cada trama. Cuando una trama llega a destino, se recalcula la suma de verificación. Si la suma de verificación calculada es distinta de la contenida en la trama, la capa de enlace de datos sabe que ha ocurrido un error y toma medidas para manejarlo.

Una manera de lograr la división de tramas es introducir intervalos de tiempo entre las tramas, de la misma manera que los espacios entre palabras. Sin embargo, las redes pocas veces ofrecen garantías sobre la temporización, por lo que es posible que estos intervalos sean eliminados o que puedan introducirse otros intervalos durante la transmisión.

Puesto que es demasiado **riesgoso depender de la temporización para marcar el inicio y el final de cada trama, se han diseñado otros métodos**. Veremos 4 métodos:

- 1-Conteo de caracteres.
- 2-Banderas, con relleno de caracteres.
- 3-Banderas de inicio y fin, con relleno de bits.
- 4-Violaciones de codificación de la capa física.

El primer método de entramado se vale de un campo en el encabezado para especificar el número de caracteres en la trama. Cuando la capa de enlace de datos del destino ve la cuenta de caracteres, sabe cuántos caracteres siguen y, por lo tanto, dónde está el fin de la trama.

El problema con este algoritmo es que la cuenta puede alterarse por un error de transmisión (sincronización). Si la cuenta cambia de 5 a 7, el destino perderá sincronía y será incapaz de localizar el inicio de la siguiente trama. Incluso si el destino sabe que la trama está mal porque la suma de verificación es incorrecta, no tiene forma de saber dónde comienza la siguiente trama. Regresar una trama a la fuente solicitando una retransmisión tampoco ayuda, ya que el destino no sabe cuántos caracteres tiene que saltar para llegar al inicio de la retransmisión.

El segundo método de entramado evita el problema de tener que sincronizar nuevamente después de un error, haciendo que cada trama inicie y termine con bytes especiales. En los años recientes la mayoría de los protocolos han utilizado el byte llamado **bandera** (o indicador), como delimitador de inicio y final, en la Fig. se denomina FLAG.

FLAG	Enca- bezado	Campo de carga útil	Termi- nador	FLAG
------	-----------------	---------------------	-----------------	------

De esta manera si el receptor pierde la sincronía, simplemente puede buscar la bandera para encontrar el final e inicio de la trama actual. Dos banderas consecutivas señalan el final de una trama y el inicio de la siguiente.

Cuando se utiliza este método para transmitir datos binarios, como programas objeto o números de punto flotante, surge un problema serio. Se puede dar el caso con mucha facilidad de que el patrón de bits de la bandera aparezca en los datos (payload), lo que interferiría en el entramado. Una forma de resolver este problema es hacer que la capa de enlace de datos del emisor inserte un byte de escape especial (ESC) justo antes de cada bandera “accidental” en los datos. La capa de enlace del lado receptor quita el byte de escape antes de entregar los datos a la capa de red. Esta técnica se llama **relleno de caracteres**. Por lo tanto, una bandera de entramado se puede distinguir de una en los datos por la ausencia o presencia de un byte de escape que la antecede. Si un byte de escape aparece en medio de los datos también se rellena con byte de escape. Por lo tanto, cualquier byte de escape individual es parte de una secuencia de escape, mientras que uno doble indica que un escape sencillo apareció de manera natural en los datos. Una *desventaja* importante del uso de esta técnica de entramado es que está fuertemente atada

a los caracteres de 8 bits. No todos los códigos utilizan caracteres de 8 bits, por lo que tuvo que desarrollarse una técnica nueva que permitiera caracteres de tamaño arbitrario. La técnica nueva permite que las tramas de datos contengan un número arbitrario de bits y admite códigos de caracteres con un número arbitrario de bits por carácter. Dicha técnica funciona de la siguiente manera: cada trama comienza y termina con un patrón especial de bits 01111110 (bandera). Cada vez que la capa de enlace del emisor encuentra 5 unos consecutivos en los datos, automáticamente inserta un 0 en el flujo de bits saliente. Este **relleno de bits** es análogo al relleno de caracteres, en el cual un byte de escape se inserta en el flujo de caracteres saliente antes de un byte igual a la bandera de entramado de los datos. Cuando el receptor ve 5 bits unos de entrada consecutivos, seguidos de un bit 0, automáticamente borra el bit 0 de relleno. Así como el relleno de caracteres es completamente transparente para la capa de red en ambas pcs, también lo es el relleno de bits. Con el relleno de bits, el límite entre las 2 tramas puede ser reconocido sin ambigüedades mediante el patrón de banderas. Si el receptor pierde la pista de donde está, debe explorar la entrada en busca de las **banderas**, pues **sólo ocurren en los límites de las tramas y nunca en los datos**.

El último método de entramado sólo se aplica a las redes en las que la codificación en el medio físico contiene cierta redundancia. Por ejemplo, algunas LANs codifican un bit de datos usando 2 bits físicos. Normalmente, un bit 1 es un alto-bajo y un bit 0 es un bajo-alto. Cada bit de datos tiene una transición a medio camino, lo que hace fácil para el receptor localizar los límites de los bits. Las combinaciones alto-alto y bajo-bajo se usan para delimitar tramas.

3.1.3 Control de errores. Cómo asegurar que todas las tramas se entreguen en el orden apropiado a la capa de red del destino? La manera normal de asegurar la entrega confiable de datos es proporcionar retroalimentación al emisor sobre lo que está ocurriendo en el otro lado de la línea. El protocolo exige que el receptor regrese tramas de control especiales que contengan confirmaciones de recepción positivas o negativas de las tramas que llegan. Una confirmación de recepción negativa significa que la trama debe transmitirse otra vez.

Una complicación adicional es que problemas de HW causen la desaparición de una trama completa (por ejemplo ruido). En este caso el emisor quedaría esperando eternamente la confirmación de recepción. Esto se maneja introduciendo temporizadores en la capa de enlace. Cuando un emisor envía una trama, por lo general también inicia un temporizador. Éste se ajusta de modo que expire cuando haya transcurrido un intervalo suficiente para que la trama llegue a su destino, se procese ahí y la confirmación de recepción regrese al emisor. Por lo general, la confirmación llegará antes de que el temporizador expire.

Sin embargo, si la trama o la confirmación se pierden, el temporizador expirará, alertando al emisor sobre un problema potencial. La solución obvia es simplemente transmitir de nuevo la trama. Sin embargo, existe el peligro de que el receptor acepte la misma trama 2 o más veces y que la pase a la capa de red más de una vez. Para evitar esto, es necesario asignar números de secuencia a las tramas que salen, a fin de que el receptor pueda distinguir las retransmisiones de los originales. El asunto de la administración de temporizadores y números de secuencia para asegurar que cada trama llegue a la capa de red en el destino una sola vez, es una parte importante de las tareas de la capa de enlace de datos.

3.1.4 Control de flujo. Otro tema importante de diseño que se presenta en la capa de enlace es qué hacer con un emisor que quiere transmitir tramas a mayor velocidad de lo que puede aceptarlas el receptor. Por lo general se utilizan 2 métodos. En el primero, el **control de flujo basado en retroalimentación**, el receptor regresa información al emisor autorizándolo para enviar más datos o indicándole su estado. En el segundo, el **control de flujo basado en tasa**, el protocolo tiene un mecanismo integrado que limita la tasa a la que el emisor puede transmitir los datos, sin recurrir a retroalimentación por parte del receptor (no se usa en capa de enlace).

Se conocen varios esquemas de control de flujo basados en retroalimentación, pero la mayoría se fundamenta en el mismo principio. El protocolo contiene reglas bien definidas respecto al momento en que un emisor puede enviar la siguiente trama.

3.3 Protocolos elementales de enlace de datos. Para comenzar, estamos suponiendo que en las capas física, de enlace de datos y de red hay procesos independientes que se comunican pasando mensajes de un lado a otro.

Otro supuesto clave es que la máquina A desea mandar un flujo considerable de datos a la máquina B usando un servicio confiable orientado a la conexión. Después consideraremos el caso de que B también quiere mandar datos a A de manera simultánea. Otro supuesto es que las máquinas no fallan en la transmisión.

Cuando la capa de enlace acepta un paquete, lo encapsula en una trama agregándole un encabezado y un terminador de enlace de datos. Por lo tanto, una trama consiste en un paquete incorporado, cierta información de control (en el encabezado) y una suma de verificación (en el terminador). A continuación la trama se transmite a la capa de enlace de la otra máquina. El HW emisor calcula y agrega la suma de verificación por lo que el SW de la capa de enlace no necesita preocuparse por ella. Cuando llega la trama al receptor, el HW calcula la suma de verificación. Si ésta es incorrecta (es decir, si hubo un error de transmisión) o no, se le informa a la capa de enlace de datos. Tan pronto como la capa de enlace receptora adquiere una trama sin daños, revisa la información de control del encabezado y, si todo está bien, pasa la parte que corresponde al paquete a la capa de red. En ninguna circunstancia se entrega un encabezado de trama a la capa de red, para mantener completamente separados el protocolo de red y de enlace de datos.

Un *packet* es la unidad de intercambio de información entre la capa de red y la de enlace de datos en la misma máquina, o entre entidades iguales. Un *frame (trama)* está compuesto de 4 campos: *kind*, *seq*, *ack* e *info*, que componen el **encabezado de la trama**.

El campo *kind* indica si hay datos en la trama, porque algunos protocolos distinguen entre las tramas que contienen exclusivamente información de control y los que también contienen datos. Los campos *seq* y *ack* se emplean para números de secuencia y confirmaciones de recepción. El campo *info* de una trama de datos contiene un solo paquete.

Es importante entender la relación entre un paquete y una trama. La capa de red construye un paquete tomando un mensaje de la capa de transporte y agrega un encabezado de la capa de red. Este paquete se pasa a la capa de enlace para incluirlo en el campo *info* de una trama saliente. Cuando ésta llega a su destino, la capa de enlace extrae el paquete y a continuación lo pasa a la capa de red. De esta manera, esta capa puede actuar como si las máquinas pudieran intercambiar paquetes directamente.

En la mayoría de los protocolos suponemos un canal inestable que pierde tramas completas ocasionalmente. Para poder recuperarse, la capa de enlace emisora debe arrancar un temporizador cada vez que envía una trama. Si no obtiene respuesta tras transcurrir cierto tiempo, el temporizador expira y la capa de enlace recibe una señal de interrupción.

En otros protocolos (utilizados cuando se supone que la capa de red no siempre tiene paquetes para enviar) la capa de enlace puede habilitar y deshabilitar la capa de red, cuando la capa de enlace habilita a la de red, ésta tiene permitido interrumpir cuando tenga un paquete para enviar. Con ésto, la capa de enlace puede evitar que la capa de red la sature con paquetes para los que no tiene espacio de búfer. Los números de secuencia de las tramas siempre están en el intervalo 0 a MAX_SEQ que difiere en cada protocolo.

3.3.1 Un protocolo simplex sin restricciones. Consideraremos un protocolo que es lo más sencillo posible, los datos se transmiten sólo en una dirección; las capas de red siempre están listas; el tiempo de procesamiento puede ignorarse; hay un espacio infinito de búfer y el canal nunca pierde tramas. (Utopía).

El protocolo consiste en 2 procedimientos diferentes, uno emisor y uno receptor. No se usan números de secuencia ni confirmaciones de recepción. El emisor está en un ciclo while infinito que sólo envía datos a la línea. El cuerpo del ciclo consiste en 3 acciones: obtener un paquete de la capa de red, construir una trama de salida y enviar la trama a su destino. Aquí no hay restricciones de control de errores ni de flujo.

El receptor espera a que algo ocurra, siendo la única posibilidad una trama sin daños. La parte de datos se pasa a la capa de red y la capa de enlace se suspende hasta la llegada de la siguiente trama.

3.3.2 Protocolo simplex de parada y espera. Ahora omitimos el supuesto más irreal en el protocolo 1: La capacidad infinita de procesamiento de la capa de red (es igual a: espacio infinito en el búfer de la capa de enlace). El canal está libre de errores y es simplex.

El problema es cómo evitar que el emisor sature al receptor enviando datos a mayor velocidad de la que el otro puede procesarlos. Si el receptor necesita Δt , el emisor debe transmitir a una tasa media menor que una trama por Δt .

En ciertas circunstancias, el emisor podría introducir simplemente un retardo en el protocolo 1 y así reducir su velocidad lo suficiente para evitar que se sature el receptor. Sin embargo, es más común que la capa de enlace tenga varias líneas a la cuales atender, y el intervalo de tiempo entre la llegada de una trama y su procesamiento puede variar en forma considerable. El problema con este método es que es demasiado conservador. Conduce a un aprovechamiento del ancho de banda muy por debajo del óptimo.

Una solución más general es hacer que el receptor proporcione retroalimentación al emisor. Tras haber pasado un paquete a su capa de red, el receptor regresa al emisor una pequeña trama ficticia que autoriza al emisor para transmitir la siguiente trama. Tras haber enviado una trama, el protocolo exige que el emisor espere hasta que llegue la pequeña trama (confirmación de recepción). Utilizar la retroalimentación del receptor para indicar al emisor cuándo puede enviar más datos. es un ejemplo de control de flujo.

Los protocolos en los que el emisor envía una trama y luego espera una confirmación de recepción antes de continuar se denominan de **parada y espera**. Se necesita tener capacidad de transferencia de información bidireccional. Este protocolo implica una

alternancia estricta de flujo: primero el emisor envía una trama y después el receptor envía una trama, aquí sería suficiente un canal semidúplex.

3.3.3 Protocolo simplex para un canal con ruido. Consideraremos la situación normal de un canal que comete errores. Las tramas pueden llegar dañadas o perderse por completo. Suponemos que si una trama se daña en el tránsito, el HW del receptor detectará esto cuando calcule la suma de verificación. Si la trama está dañada y la suma de verificación sigue siendo correcta (poco probable), este protocolo (y todos los demás) puede fallar, osea, entrega un paquete incorrecto a la capa de red.

A primera vista puede parecer que funcionaría una variación del protocolo 2: agregar un temporizador. Este esquema tiene un defecto mortal, recordar que la capa de enlace debe proporcionar una comunicación transparente y libre de errores entre los procesos de las capas de red. La capa de red del receptor no tiene manera de saber si el paquete se ha perdido o duplicado, por lo que la capa de enlace debe garantizar que ninguna combinación de errores de transmisión, por improbables que sea, pueda causar la entrega de un paquete duplicado a la capa de red.

Se necesita alguna manera de que el receptor sea capaz de distinguir entre una trama que está viendo por primera vez y una retransmisión. Una forma de lograrlo es hacer que el emisor ponga un número de secuencia en el encabezado de cada trama que envía. El receptor puede examinar el número de secuencia de cada trama que llega para ver si es una trama nueva o un duplicado que debe descartarse.

Dado que es deseable que el encabezado de las tramas sea pequeño, ¿Cuál es la cantidad mínima de bits necesarios para el número de secuencia? La única ambigüedad es entre una trama y su antecesor o sucesor inmediatos. Por lo tanto, basta con 1 bit (0 ó 1). En cada instante, el receptor espera un número de secuencia particular. Cualquier trama que contenga un número de secuencia equivocado se rechaza como duplicado. Cuando llega una trama que contiene el número correcto, se acepta y se pasa a la capa de red, y el número de secuencia se incrementa módulo 2.

Los protocolos en los que el emisor espera una confirmación de recepción positiva antes de avanzar al siguiente elemento de datos se llama **PAR (Confirmación de Recepción Positiva con Retransmisión)** o **ARQ (Solicitud Automática de Recepción)**. ///Control de errores de datos garantizando su integridad///.

El protocolo 3 difiere de sus antecesores en que tanto el emisor como el receptor tienen una variable cuyo valor se recuerda mientras la capa de enlace de datos está en estado de espera.

Tras transmitir una trama, el emisor arranca el temporizador. (Si el intervalo establecido es muy pequeño, el emisor transmitirá tramas innecesarias que dañan el rendimiento). Mientras el emisor espera, hay 3 posibilidades: llega una trama de confirmación de recepción sin daño, llega una trama de confirmación de recepción dañada o expira el temporizador. Si recibe una confirmación de recepción válida, el emisor obtiene el siguiente paquete de la capa de red y lo coloca en el búfer, sobreescribiendo el paquete previo. También avanza el número de secuencia. Si llega una trama dañada o no llega ninguna, ni el búfer ni el número de secuencia cambia, con el fin de que se pueda enviar un duplicado. Cuando llega una trama válida al receptor, su número de secuencia se verifica para saber si es un duplicado. Si no lo es, se pasa a la capa de red y se genera una confirmación de recepción. Los duplicados y las tramas dañadas no pasan a la capa de red.

Examinamos una serie de protocolos. El protocolo 1 se designa para un entorno libre de errores, en el que el receptor puede manejar cualquier flujo que se le haya enviado. El protocolo 2 también da por hecho un entorno libre de errores pero introduce control de flujo. El protocolo 3 maneja los errores con números de secuencia y con el algoritmo de parada y espera. Ahora, el protocolo 4 permite la comunicación bidireccional e introduce el concepto de superposición. El protocolo 5 utiliza un protocolo de ventana corrediza con retroceso n. Por último, el protocolo 6 utiliza repetición selectiva y confirmaciones de recepción negativas.

3.4 Protocolos de ventana corrediza. En la mayoría de las situaciones prácticas hay necesidad de transmitir datos en ambas direcciones. Una manera de lograr una transmisión dúplex total es tener 2 canales separados y utilizar cada uno para tráfico simplex (ida-datos; vuelta-Confirmación). Si se hace esto, el ancho de banda usado para confirmaciones de recepción se desperdicia casi por completo. Una mejor idea es utilizar el mismo circuito para datos en ambas direcciones. En este modelo, las tramas de datos A a B se mezclan en las mismas tramas de confirmación de recepción de A a B. Analizando el campo de tipo *kind* en el encabezado de una trama de entrada, el receptor puede saber si la trama es de datos o de confirmación de recepción. Se puede lograr otra mejora. Cuando llega una trama de datos, en lugar de enviar inmediatamente una trama de control independiente, el receptor se aguanta y espera hasta que la capa de red le pasa el siguiente paquete. La confirmación de recepción se anexa a la trama de datos de salida (usando el campo *ack* del encabezado de la trama). La confirmación viaja en la siguiente trama de salida. La técnica de retardar temporalmente las confirmaciones de recepción para que puedan viajar en la siguiente trama de datos de salida se conoce como **superposición** (piggybacking).

La ventaja principal de es un aprovechamiento del ancho de banda disponible del canal. El campo *ack* del encabezado de la trama ocupa sólo unos cuantos bits, mientras que una trama aparte requiere un encabezado, la confirmación y la suma de verificación. En el siguiente protocolo, el campo de superposición ocupa sólo 1 bit en el encabezado de la trama y pocas veces ocupa más de algunos bits.

Sin embargo, la superposición introduce una complicación nueva. ¿Cuánto tiempo debe esperar la capa de enlace un paquete al cual superponer la confirmación de recepción? Si la capa de enlace espera más tiempo del que tarda en terminar el temporizador del emisor, la trama será retransmitida. La capa de enlace espera un número fijo de milisegundos. Si llega rápidamente un nuevo paquete, la confirmación de recepción se superpone; de otra manera, la capa de enlace manda una trama de confirmación de recepción independiente.

Los siguientes 3 protocolos son bidireccionales y son de la clase llamada **ventana corrediza**. Los 3 difieren entre ellos en eficiencia, complejidad y requerimientos de búfer. Cada trama tiene un número de secuencia. La esencia de todos los protocolos de ventana corrediza es que, en cualquier instante, el emisor mantiene un grupo de números de

secuencia que corresponde a las tramas que tiene permitido enviar. Se dice que estas tramas caen dentro de la **ventana emisora**. De manera semejante, el receptor mantiene una **ventana receptora** correspondiente al grupo de tramas que tiene permitido aceptar. En algunos protocolos las ventanas son de tamaño fijo, pero en otros pueden variar.

Estos protocolos dan a la capa de enlace mayor libertad en cuanto al orden en que puede enviar y recibir tramas, hemos conservado decididamente el requisito de que el protocolo debe entregar los paquetes a la capa de red destino en el mismo orden en que fueron pasados a la capa de enlace emisora. Seguimos suponiendo que **el protocolo debe entregar todas las tramas a la capa de red receptora en el orden en que fueron enviadas a la capa de enlace emisora**.

Los protocolos de ventana corrediza pueden clasificarse por el tamaño de la ventana del emisor y por el tamaño de la ventana del receptor. Cuando ambos son iguales a 1, el protocolo es de parada y espera. Cuando el tamaño de la ventana del emisor es mayor que 1, el receptor puede programarse para descartar todas las tramas diferentes a la siguiente de la secuencia o almacenar en el búfer tramas fuera de orden hasta que se necesiten.

Los números de secuencia en la ventana del emisor representan tramas enviadas, o que puedan ser enviadas, pero cuya recepción aún no se ha confirmado. Cuando llega un paquete nuevo de la capa de red, se le da el siguiente número secuencial mayor, y el extremo superior de la ventana avanza en uno. De esta manera, la ventana mantiene continuamente una lista de tramas sin confirmación de recepción.

Dado que las tramas que están en la ventana del emisor pueden perderse o dañarse en el tránsito, el emisor debe mantener todas estas tramas en su memoria para su posible retransmisión. Por lo tanto, si el tamaño máximo de la ventana es n , el emisor necesita n búferes para contener las tramas sin confirmación de recepción. Si la ventana llega a crecer a su tamaño máximo, la capa de enlace de datos emisora deberá hacer que la capa de red se detenga hasta que se libere otro búfer.

La ventana de la capa de enlace receptora corresponde a las tramas que puede aceptar. Toda trama que caiga fuera de la ventana se descartará. Cuando se recibe la trama cuyo número de secuencia es igual al extremo inferior de la ventana, se pasa a la capa de red, se genera una confirmación de recepción y se avanza la ventana en 1. La ventana del receptor conserva siempre el mismo tamaño inicial. Con un tamaño de ventana de 1 significa que la capa de enlace de datos sólo acepta tramas en orden, pero con ventanas más grandes esto no es así. La capa de red siempre recibe los datos en el orden correcto, sin importar el tamaño de la ventana de la capa de enlace.

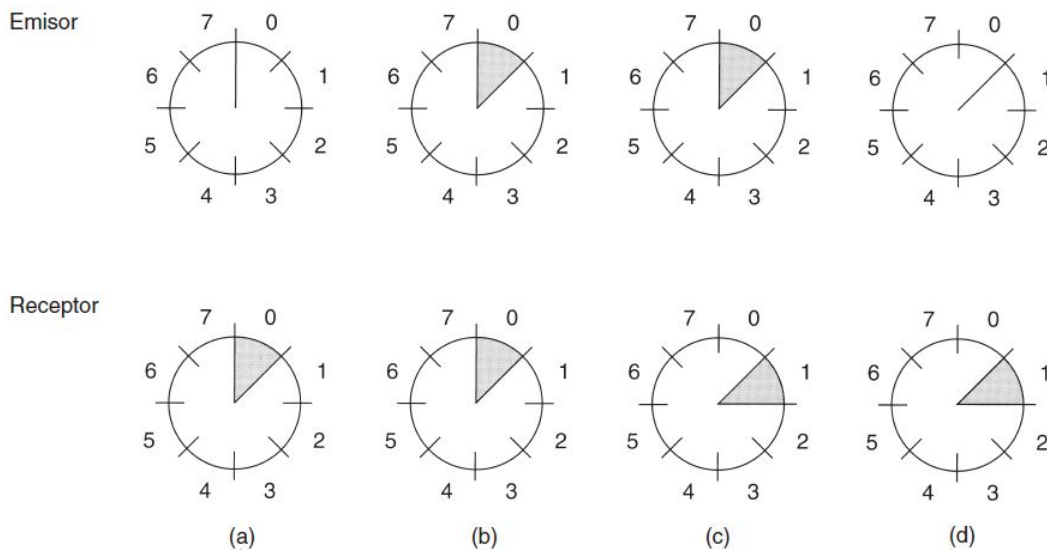


Figura 3-13. Ventana corrediza de tamaño 1, con un número de secuencia de 3 bits. (a) Al inicio. (b) Tras la transmisión de la primera trama. (c) Tras la recepción de la primera trama. (d) Tras recibir la primera confirmación de recepción.

3.4.1 Un protocolo de ventana corrediza de un bit. Este protocolo utiliza parada y espera, ya que el emisor envía una trama y espera su confirmación de recepción antes de transmitir la siguiente.

La máquina que empieza a transmitir obtiene el primer paquete de su capa de red, construye una trama y la envía. Al llegar la trama, la capa de enlace receptora la revisa para saber si es un duplicado, (igual que el protocolo 3). Si la trama es la esperada, se pasa a la capa de red y la ventana del receptor se recorre hacia arriba.

El campo de confirmación de recepción contiene el número de la última trama recibida sin error. Si este número concuerda con el de secuencia de la trama que está tratando de enviar el emisor, éste sabe que ha terminado con la trama almacenada en el búfer y que puede obtener el siguiente paquete de la capa de red. Por cada trama que se recibe, se regresa una.

3.4.2 Protocolo que usa retroceso N. El tiempo de viaje de ida y vuelta prolongado puede tener implicaciones importantes para la eficiencia del aprovechamiento del ancho de banda. Si relajamos la restricción de que el emisor requiere una confirmación de recepción antes de enviar otra trama, se puede mejorar la eficiencia. Básicamente la solución está en permitir que el emisor envíe hasta W tramas antes de bloquearse, en lugar de sólo 1. Con una selección adecuada de W , el emisor podrá transmitir tramas continuamente durante un tiempo igual al tiempo de tránsito de ida y vuelta sin llenar la ventana.

La necesidad de una ventana grande en el lado emisor se presenta cuando el producto del ancho de banda por el retardo del viaje ida y vuelta es grande. Si el ancho de banda es alto, incluso para un retardo moderado, el emisor agotará su ventana rápidamente a menos que tenga una ventana grande. Si el retardo es grande (satélite), el emisor agotará su ventana incluso con un ancho de banda moderado. El producto de estos 2 factores indica cual es la

capacidad del canal, y el emisor necesita la capacidad de llenarlo sin detenerse para poder funcionar con una eficiencia máxima.

Esta técnica se conoce como **canalización**. Si la capacidad del canal es de b bits/seg, el tamaño de la trama de L bits y el tiempo de propagación de ida y vuelta de R segundos, el tiempo requerido para transmitir una sola trama es de L/b segundos.

El envío de tramas en canalización por un canal de comunicación inestable presenta problemas serios. Primero, ¿qué ocurre si una trama a la mitad de una serie larga se daña o pierde? Llegarán grandes cantidades de tramas sucesivas al receptor antes de que el emisor se entere de que algo anda mal. Cuando llega una trama dañada al receptor, debe descartarse, pero, ¿qué debe hacerse con las tramas correctas que le siguen? (Hay que entregarlos en secuencia).

Hay 2 métodos básicos para manejar los errores durante la canalización.

Uno es el **retroceso N**, donde el receptor simplemente descarte todas las tramas subsecuentes, sin enviar confirmaciones de recepción para las tramas descartadas. Esta estrategia corresponde a una ventana de recepción de tamaño 1. En otras palabras, la capa de enlace se niega a aceptar cualquier trama excepto la siguiente que debe entregar a la capa de red. Si la ventana del emisor se llena antes de terminar el temporizador, el canal comenzará a vaciarse. En algún momento el emisor terminará de esperar y retransmitirá en orden todas las tramas cuya recepción aún no se haya confirmado, comenzando por la dañada o perdida. Esta técnica puede desperdiciar bastante ancho de banda si la tasa de errores es alta.

La otra estrategia general para el manejo de errores cuando las tramas se colocan en canalizaciones se conoce como **repetición selectiva**. Cuando se descarta una trama dañada recibida, pero las tramas en buen estado recibidas después de ésta se almacenan en el búfer. Cuando el emisor termina, sólo la última trama sin confirmación se retransmite. Si la trama llega correctamente, el receptor puede entregar a la capa de red, en secuencia, todas las tramas almacenadas. La repetición selectiva con frecuencia se combina con el hecho de que el receptor envíe una confirmación de recepción negativa (NAK) cuando detecta un error. Las NAK estimulan la retransmisión antes de que el temporizador correspondiente expire, por ello mejora el rendimiento. La NAK acelera la retransmisión de una trama específica.

La repetición selectiva corresponde a una ventana del receptor mayor que 1. Cualquier trama dentro de la ventana puede ser aceptada y mantenida en el búfer hasta que todas las que le preceden hayan sido pasadas a la capa de red. Puede requerir más memoria si las ventanas son grandes.

Estas 2 estrategias son intercambios entre el ancho de banda y el espacio de búfer en la capa de enlace. Dependiendo de qué recurso sea más valioso, se puede utilizar uno o el otro.

Aunque el protocolo 5 no pone en el búfer las tramas que llegan tras un error, no escapa del problema de los búferes por completo. Dado que un emisor puede tener que retransmitir en un momento futuro todas las tramas no confirmadas, debe retener todas las tramas retransmitidas hasta saber con certeza que han sido aceptadas por el receptor. Al llegar una confirmación de recepción para la trama n , las tramas $n-1$ y $n-2$, y demás, se confirman de manera automática. Esta propiedad es importante cuando algunas tramas previas portadoras de confirmaciones de recepción se perdieron o dañaron. Cuando llega una confirmación de recepción, la capa de enlace revisa si se pueden liberar los búferes, si esto

es posible, ya puede permitirse que una capa de red previamente bloqueada produzca más eventos.

Para este protocolo damos por hecho que siempre hay tráfico de regreso en el que se pueden superponer confirmaciones de recepción. El protocolo 4 no necesita este supuesto debido a que envía una trama cada vez que recibe una trama. En el siguiente protocolo resolveremos el problema del tráfico de una vía de una forma elegante.

Debido a que este protocolo tiene múltiples tramas pendientes, necesita múltiples temporizadores, uno por cada trama pendiente. Cada trama termina de temporizar independientemente de las demás.

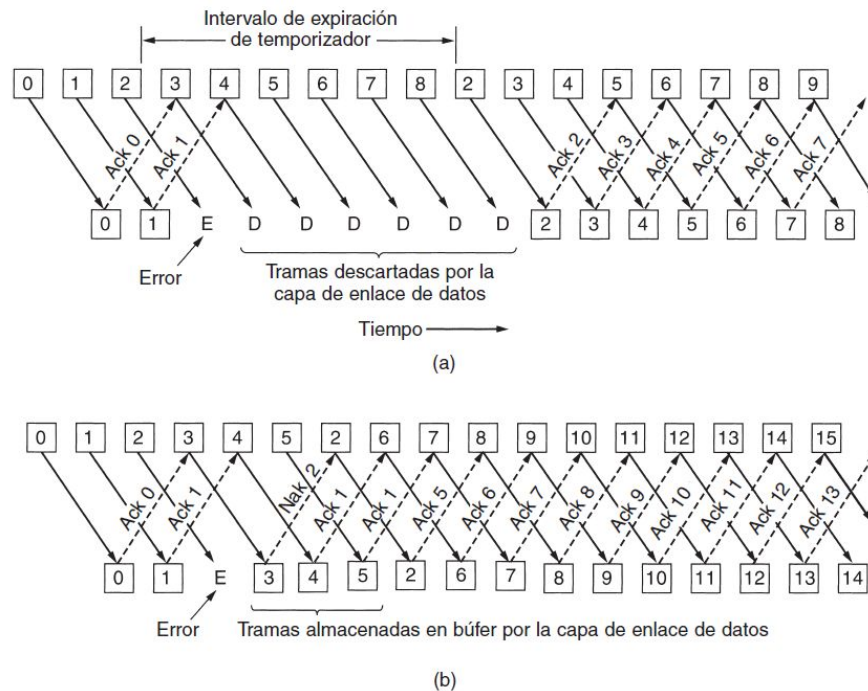


Figura 3-16. Canalización y recuperación de un error. (a) Efecto de un error cuando el tamaño de la ventana del receptor es de 1. (b) Efecto de un error cuando el tamaño de la ventana del receptor es grande.

3.4.3 Protocolo que utiliza repetición selectiva. El protocolo 5 funciona bien si los errores son poco frecuentes, pero si la línea es mala, se desperdicia mucho ancho de banda en las tramas retransmitidas. Una estrategia alterna para el manejo de errores es permitir que el receptor acepte y coloque en búferes las tramas que siguen a una trama dañada o perdida. Tal protocolo no rechaza tramas simplemente porque se dañó o se perdió la trama anterior. En este protocolo, tanto el emisor como el receptor mantienen una ventana de números de secuencia aceptables. El tamaño de la ventana del emisor empieza en 0 y crece hasta el MAX predefinido. La ventana del receptor siempre es de tamaño fijo e igual a MAX. El receptor tiene un búfer reservado para cada número de secuencia en su ventana fija. Cada búfer tiene un bit asociado (*arrived*) que indica si el búfer está lleno o vacío. Cuando llega una trama, su número de secuencia es revisado para ver si cae dentro de la ventana. De ser así, y si no ha sido recibida aún, se acepta y se almacena. Esta acción se lleva a cabo

sin importar si la trama contiene el siguiente paquete esperado por la capa de red. Por supuesto, debe mantenerse dentro de la capa de enlace sin entregarse a la capa de red hasta que todas las tramas de número menor hayan sido entregadas en el orden correcto. La recepción no secuencial introduce ciertos problemas que no se presentan en los protocolos en los que las tramas sólo se aceptan en orden. El problema esencial es que una vez que el receptor ha avanzado su ventana, el nuevo intervalo de números de secuencia válidos se traslapa con el anterior. En consecuencia, el siguiente grupo de tramas podría ser de tramas duplicadas (si se perdieron todas las tramas de confirmación de recepción) o de nuevas (si se recibieron todas las tramas de confirmación de recepción). El receptor no tiene manera de distinguir estos 2 casos.

La salida de esto es asegurarse que, una vez que el emisor haya avanzado su ventana, no haya traslape con la ventana original. Para asegurarse de que no haya traslape, el tamaño máximo de la ventana debe ser cuando menos de la mitad del intervalo de los números de secuencia. Por ejemplo, si se utilizan 4 bits para los números de secuencia, estos tendrán un intervalo de 0 a 15. Sólo 8 tramas sin confirmación de recepción deben estar pendientes en cualquier instante. De esa manera, si el receptor apenas ha aceptado las tramas 0 a 7 y ha avanzado su ventana para permitir la aceptación de las tramas 8 a 15, puede distinguir sin ambigüedades si las tramas subsiguientes son retransmisiones (0 a 7) o nuevas (8 a 15). En general, el tamaño de la ventana del protocolo **repetición selectiva**, será de $(MAX+1)/2$. Por lo tanto para números de secuencia de 3 bits (8 nros), el tamaño es 4.

El número de búferes que debe tener el receptor es igual al tamaño de la ventana, no al intervalo de números de secuencia. El número de temporizadores es igual al número de búferes. Hay un temporizador asociado a cada búfer. Cuando termina el temporizador, el contenido del búfer se retransmite.

En el protocolo 5 (retroceso n) se supone de manera implícita que el canal está fuertemente cargado. Cuando llega una trama, no se envía de inmediato la confirmación de recepción. En cambio, esta última se superpone en la siguiente trama de datos de salida. Si el tráfico de regreso es ligero, la confirmación de recepción se detendrá durante un período largo. Si hay mucho tráfico en una dirección y no hay tráfico en la otra, sólo se envían MAX paquetes y luego se bloquea el protocolo, que es por lo cual dimos por hecho que siempre había tráfico de regreso.

En el protocolo 6 (repetición selectiva) se corrige este problema. Tras llegar una trama de datos en secuencia, se arranca un temporizador auxiliar. Si no se ha presentado tráfico de regreso antes de que termine el temporizador, se envía una trama de confirmación de recepción independiente. con este arreglo ahora es posible el flujo de tráfico unidireccional, pues la falta de tramas de datos de regreso a las que puedan superponerse las confirmaciones de recepción ya no es un obstáculo.

Es indispensable que el tiempo asociado al temporizador auxiliar sea notablemente más corto que el del temporizador usado para la terminación de tramas de datos. Esta condición es necesaria para asegurarse de que la confirmación de recepción de una trama correctamente recibida llegue antes de que el emisor termine su temporizador y retransmita la trama.

El protocolo 6 utiliza una estrategia más eficiente que el 5 para manejar los errores. Cuando el receptor tiene razones para suponer que ha ocurrido un error, envía al emisor una trama de confirmación de recepción negativa (NAK) que es una solicitud de retransmisión de una trama. Hay 2 casos en los que el receptor debe sospechar: cuando llega una trama dañada,

o cuando llega una trama diferente a la esperada. Para evitar hacer múltiples solicitudes de retransmisión, el receptor debe saber si ya se ha enviado una NAK para una trama dada. Si la NAK se altera o se pierde no hay un daño real, pues en algún momento el temporizador del emisor terminará y se retransmitirá la trama perdida.

3.6 Ejemplos de protocolos de enlace de datos. El primero HDLC, orientado a bits, el segundo PPP utilizado para conectar a Internet computadoras domésticas.

3.6.1 HDLC- Control de Enlace de Datos de Alto Nivel. Es un protocolo orientado a bits y usa el relleno de bits para lograr la transparencia de los datos.

Todos los protocolos orientados a bits utilizan la estructura de trama mostrada en la Fig. 3-24.

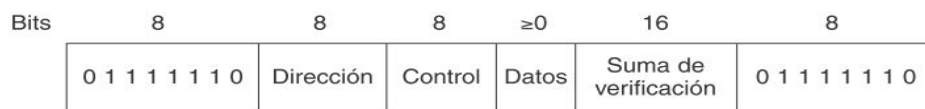


Figura 3-24. Formato de trama para protocolos orientados a bits.

El campo *Dirección* es de importancia primordial en las líneas con múltiples terminales, pues sirven para identificar una de las terminales. El campo *Control* se utiliza para números de secuencia, confirmaciones de recepción y otros depósitos. El campo *Datos* puede contener cualquier información y puede tener longitud arbitraria, aunque la eficiencia de la suma de verificación disminuye conforme el tamaño de la trama aumenta, debido a la probabilidad de múltiples errores. El campo *Suma de verificación* es un código de redundancia cíclica. La trama está delimitada por otra secuencia bandera. En líneas punto a punto inactivas se transmiten secuencias de bandera continuamente. La trama mínima contiene 3 campos y un total de 32 bits, excluyendo las banderas.

Hay 3 tipos de tramas: **de información, de supervisión y no numeradas**. El protocolo emplea una ventana corrediza, con un número de secuencia de 3 bits. En cualquier momento pueden estar pendientes hasta 7 tramas sin confirmación de recepción. El campo *secuencia* de la Fig. 3-25a es el número de secuencia de la trama. El campo *Siguiente* es una confirmación de recepción superpuesta. Sin embargo, en lugar de superponer el número de la última trama recibida correctamente, usan el número de la primera trama no recibida (trama esperada). El bit *P/F* significa sondeo/final. Se utiliza cuando una pc está sondeando un grupo de terminales. Cuando se usa como P, la pc está invitando a la terminal a enviar datos.



Figura 3-25. Campo de Control de (a) una trama de información, (b) una trama de supervisión y (c) una trama no numerada.

Los tipos de **tramas de supervisión** se distinguen por el campo de *Tipo*. El tipo 0 es una trama de confirmación de recepción que sirve para indicar la siguiente trama esperada.

El tipo 1 es una trama de confirmación negativa, sirve para indicar que se ha detectado un error de transmisión. El campo *siguiente* indica la primera trama en la secuencia que no se ha recibido en forma correcta. Se pide al emisor retransmitir todas las tramas pendientes comenzando por *siguiente*.

El tipo 2 es RECEIVE NOT READY (receptor no listo); reconoce todas las tramas hasta, pero sin incluir siguiente, pero le dice que detenga el envío.

El tipo 3 es SELECT REJECT, solicita la retransmisión de sólo la trama especificada.

La 3era clase de trama es la trama **no numerada** que a veces se usa para propósitos de control, aunque también puede servir para llevar datos cuando se solicita un servicio no confiable sin conexión. Hay 5 bits disponibles para indicar el tipo de trama enviada, pero no se utilizan las 32 posibilidades.

Las tramas de control pueden perderse o dañarse, por lo que también se debe confirmar su recepción. Se proporciona una trama de control especial para este propósito, llamada UA (confirmación de recepción no numerada).

Las tramas de control restantes tienen que ver con la inicialización, sondeo e informe de estado. Estos datos no se pasan a la capa de red, pues son para uso de la capa de enlace.

3.6.2 Protocolo PPP- Protocolo Punto a Punto. Internet necesita de un protocolo punto a punto para diversos propósitos, entre ellos para el tráfico enrutador a enrutador y tráfico usuario a ISP (proveedor de internet).

PPP proporciona 3 características:

1-Un método de entramado que delinea sin ambigüedades el final de una trama y el inicio de la siguiente. El formato de trama también maneja detección de errores.

2-Un protocolo de control de enlace para activar líneas, probarlas, negociar opciones y desactivarlas cuando ya no son necesarias. Este protocolo se llama **LCP (Protocolo de control de Enlace)**. Admite circuitos síncrono y asíncronos y codificaciones orientadas a bits y a caracteres.

3-Un mecanismo para negociar opciones de capa de red con independencia del protocolo de red usado. El método consiste en tener un **NCP (Protocolo de Control de Red)** para cada protocolo de capa de red soportado.

El formato de la trama PPP se escogió de modo que fuera muy parecido al de HDLC. La diferencia principal entre PPP y HDLC es que el primero está orientado a caracteres, no a bits. En particular, PPP usa relleno de bytes en las líneas de acceso telefónico con módem,

por lo que todas las tramas tienen un número entero de bytes. No sólo pueden mandarse tramas PPP a través de líneas telefónicas, sino que también pueden enviarse sobre líneas HDLC.

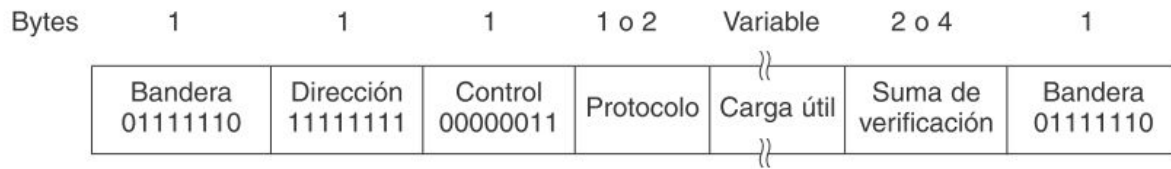


Figura 3-27. Formato de trama completa PPP para el modo de operación no numerado.

Todas las tramas PPP comienzan con la bandera estándar de HDLC, que se rellena con bytes si ocurre dentro del campo de carga útil. Luego está el campo de *Dirección*, que siempre se establece al valor binario 11111111 para indicar que todas las estaciones deben aceptar la trama.

El campo de *Control*, cuyo valor predeterminado es 00000011. Este valor indica una trama no numerada. En otras palabras, PPP no proporciona de manera predeterminada transmisión confiable usando números de secuencia y confirmaciones de recepción.

El cuarto campo es el de *Protocolo*. Su tarea es indicar la clase de paquete que está en el campo de *Carga útil*. El campo de *Carga útil* es de longitud variable, hasta algún máximo negociado. Finalmente el campo de *Suma de verificación*.

En resumen, PPP es un mecanismo de entramado multiprotocolo adecuado para utilizarse a través de módems, líneas seriales de bits HDLC y otras capas físicas. Soporta detección de errores, negociación de opciones, compresión de encabezados y, opcionalmente, transmisión confiable con formato de tramas similar a HDLC.