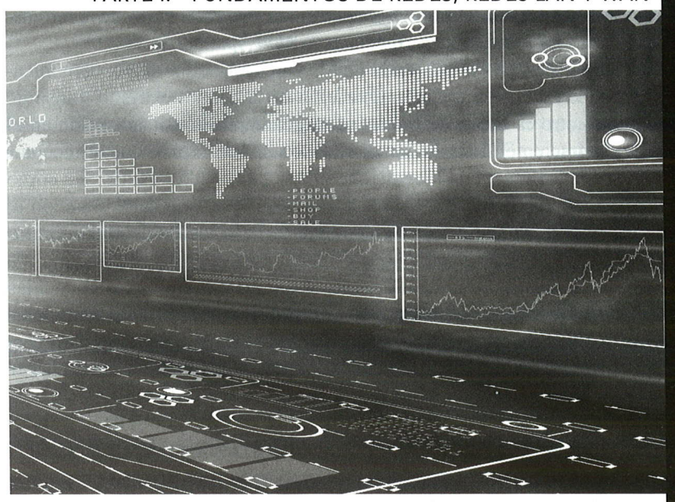
del sistema debería proporcionar un valor de u alrededor de $u = 0{,}8$, a pesar de que ello implique la infrautilización de la línea de salida.

Evidentemente, en un sistema real no es posible que el retardo sea infinito, puesto que tampoco será infinito el tamaño de la cola de servicio. En este caso, habría que considerar un sistema de colas con tamaño de memoria finito, M/D/1/N, y aparecerá una probabilidad de pérdida de ranuras relacionada con la posibilidad de que se llene la memoria. Esta probabilidad también puede ser evaluada con teoría de colas, aunque no lo abordaremos en este libro.



CAPÍTULO 4

CONTROL DEL ENLACE DE DATOS

4.1. Introducción
4.2. Delimitación de tramas
4.3. Códigos para la gestión de errores
4.4. Control de flujo
4.5. Técnicas ARQ
4.6. Ejemplo de protocolo de control de enlace: LLC

4.1. Introducción

A lo largo de este capítulo se aborda el estudio de las funciones de la capa de enlace ya apuntadas en el Apartado 1.3 del tema introductorio. El objetivo de esta capa es proporcionar las funcionalidades necesarias para posibilitar un servicio de transferencia de datos fiable a través del enlace físico. Esto es, para garantizar que todos y cada uno de los datos emitidos se reciben correctamente en el destino cuando estos pueden interactuar directamente entre sí. Para ello se requiere el análisis de los problemas que pueden aparecer y la propuesta de mecanismos o funcionalidades que los resuelvan.

Una de ellas, tal vez la más básica y evidente, es la del control de errores. Dado que todo canal de comunicación es real, es posible la ocurrencia de errores de transmisión que provoquen que los datos recibidos en el receptor difieran de los emitidos por el origen. Es por ello necesario arbitrar algún esquema o método en base al cual el receptor de una transmisión sea capaz de, en primer lugar, determinar la potencial ocurrencia de errores y, a continuación, actuar en consecuencia a fin de solucionarlos. Las técnicas de control de errores pasan por la inclusión de redundancia en la transmisión, de forma que se dota al receptor de capacidad para detectar la aparición de errores en los datos y, en algunos casos, incluso para su corrección por el mismo receptor. En el caso de que únicamente se pueda realizar la detección de los errores, la recuperación de los datos se realiza a partir de la solicitud al emisor de su retransmisión. Esta forma de operación, denominada solicitud de retransmisión automática (ARQ, «Automatic Repeat reQuest») suele ser la habitual en las redes de computadores y se suele combinar, como veremos a lo largo de este capítulo, con otras funcionalidades. Por ello, a la hora de

estudiar el problema diferenciaremos entre los mecanismos de inclusión de redundancia, tanto para detección como para detección y corrección (Apartado 4.3) y el control de errores a partir de la retransmisión de los bloques detectados como dañados, esto es, mediante ARQ (Apartado 4.5).

La segunda de las funciones requeridas está relacionada con la sincronización de las comunicaciones entre el emisor y el receptor. Pensemos en una comunicación en la que un extremo transmite datos a mayor velocidad de lo que el otro es capaz de procesarlos, generándose así saturación en el receptor. Esta situación puede producirse en el caso de que el emisor sea más rápido que el receptor o, sencillamente, que este último se encuentre ocupado realizando otras tareas adicionales. Sea como fuere, es claro que dicha situación resulta del todo indeseable y, por tanto, hay que darle solución o, en caso contrario, se produciría pérdida de información. Evidentemente, la solución debe proporcionar la máxima eficiencia posible en las comunicaciones, lo que invalida soluciones en principio triviales como el uso de velocidades de transmisión suficientemente bajas. El estudio de las conocidas como técnicas de control de flujo, que permiten solucionar este problema, será así otra de las funciones presentadas a lo largo de este capítulo (Apartado 4.4). Estos esquemas están diseñados para permitir una transmisión emisor-receptor ordenada, en la que ambas partes «dialoguen» a fin de hacer posible la recepción de todos los datos enviados.

Para facilitar y hacer más eficiente el uso de las técnicas de control de errores y de flujo, así como todas las funcionalidades que deben implementarse en las capas superiores, resulta más adecuado trabajar sobre bloques de datos que hacerlo a nivel de bits individuales. Desde este punto de vista, los datos procedentes de las capas superiores se agrupan en unidades de información para su transmisión sobre el canal de comunicación o enlace de datos. Relacionada con esta agrupación de bits, una tercera función realizada en la capa de enlace se refiere a la delimitación de los bloques de datos a fin de posibilitar una adecuada sincronización emisor-receptor desde el punto de vista de la correcta recepción temporal de los mismos. Este aspecto se discutirá en el Apartado 4.2.

Finalmente, el capítulo concluye con la presentación, en el Apartado 4.6, de uno de los protocolos de enlace más utilizados en redes de computadores: LLC. Para este se referirán las tres funciones comentadas anteriormente, evidenciándose cómo se materializan los procedimientos y servicios ofrecidos a este nivel.

Antes de dar paso a la exposición de los distintos contenidos planteados, es importante aclarar un par de cuestiones al respecto. Aunque la delimitación es la última de las funciones de la capa de enlace en realizarse antes de pasar los datos a la capa física para su transmisión, dada su simplicidad, y con objeto de facilitar el desarrollo y comprensión del resto del tema, será la primera de las tres que analizaremos. Respecto de las otras dos funciones, control de flujo y de errores, hemos de aclarar que su consideración práctica no es obligatoria por cuanto que dependerá del tipo de servicio ofrecido: orientado a conexión o no, y con confirmación o sin ella. Esto es, la capa de control de enlace debe proporcionar las funcionalidades que permitan una transmisión fiable, pero no es obligatorio su uso en todas las comunicaciones. En cualquier caso, sí es importante mencionar que ambas funciones suelen ser implementadas conjuntamente, o sea, el control de errores implica, generalmente, un control de flujo sobre los datos transmitidos.

4.2. Delimitación de tramas

La transmisión de un mensaje entre un emisor y un receptor pasa por su fragmentación en bloques de bits que reciben el nombre de *tramas* (*«frame»* en inglés) en la capa de enlace. Dos son las cuestiones a analizar respecto de esta fragmentación: el tamaño más adecuado para las tramas y la forma de determinar dónde comienzan y terminan. A continuación se discuten ambas cuestiones.

La elección de un tamaño adecuado para las tramas debe tener en cuenta varios aspectos contrapuestos, no debiendo ser demasiado largas ni demasiado cortas. Si se eligen demasiado largas provocaría que, en caso de ocurrencia de errores de transmisión (situación bastante normal dada la naturaleza no ideal de los canales de comunicación), se desaprovecharía el ancho de banda del canal durante la transmisión de la misma. Además, cuanto mayor sea la trama mayor será la probabilidad de error, lo que acentúa el efecto anterior. Asimismo, el uso de tramas largas tiene efectos no deseados en relación al uso compartido del canal (véase el Capítulo 3), ya que aumenta el retardo en el acceso por parte de las otras estaciones, y también plantea algunas dificultades en relación a los tamaños de las memorias de almacenamiento, que deben ser mayores. Por el contrario, la transmisión de bloques excesivamente cortos haría que la comunicación fuese ineficiente, puesto que, dado que es necesario incluir bits adicionales en las cabeceras por parte de cada capa, la relación bits suplementarios-bits de datos aumentaría. Así pues, para una transmisión eficiente hemos de llegar a un compromiso en el tamaño de los bloques de datos a transmitir. Hay que señalar que esta no es una decisión inmediata, como así se evidencia en la existencia de distintos tamaños de trama en diferentes entornos de transmisión reales.

Sea cual sea el tamaño decidido para las tramas surge el problema de la sincronización emisor-receptor para una adecuada recepción temporal de estas. Como se vio en el Apartado 2.4, correspondiente a interfaces, la mayoría de las transmisiones realizadas sobre canales de comunicación son síncronas. Esto significa que la sincronización emisor-receptor se realiza usando caracteres de delimitación de trama (Figura 4.1(a)), que permiten establecer el inicio y fin de cada bloque de datos. Dos son las técnicas básicas que podemos encontrar en este sentido:

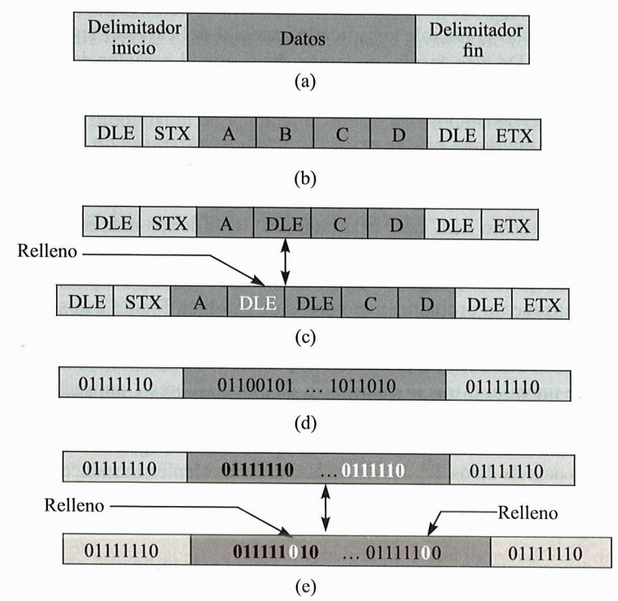


Figura 4.1. Delimitación de tramas en transmisión síncrona: conceptual (a), delimitación con caracteres de principio y fin con inserción de caracteres (b) y (c), delimitación con bits de principio y fin con inserción de bits (d) y (e).

E.T.S.I.I.T. BIBLIOTECA GRANADA

a) *Delimitación mediante caracteres de principio y fin con inserción de caracteres.* En esta técnica se emplea uno o más caracteres especiales para indicar el comienzo de la trama de datos y uno o más caracteres (los mismos o distintos de los de comienzo) para señalizar su final. En el caso de utilizar código ASCII («American Standard Code for Information Interchange»), los caracteres especiales para esta finalidad son (Figura 4.1(b)) DLE («Data Link Escape») y STX («Start of TeXt») para indicar el comienzo y DLE seguido de ETX («End of TeXt») para el final.

En este esquema la aparición de la secuencia delimitadora en los datos daría lugar a una delimitación inadecuada de la trama y, en consecuencia, la recepción resultaría incorrecta. Este hecho se soluciona de forma fácil sin más que incluir de forma sistemática algún carácter especial que evite la aparición de la secuencia delimitadora, proceso conocido como *inserción de caracteres*. Por ejemplo, se podría incluir un carácter especial a continuación, o delante, de la secuencia delimitadora. El receptor, dado que la sustitución sería sistemática, únicamente tendría que reconocer dicha inserción y eliminarla. Este proceso puede resultar algo complejo, al depender el significado de los caracteres especiales de los caracteres adyacentes.

Existe una posibilidad más simple, tanto para el receptor como para el emisor, que evita el problema. Así, bastaría con que el emisor insertase un carácter DLE detrás de todo octeto de datos DLE aparecido en el flujo de datos a transmitir (Figura 4.1(c)). De esta forma, se evitaría la aparición de la secuencia DLE-STX o DLE-ETX en los datos, ya que el carácter DLE estaría duplicado. El proceso de delimitación a seguir en el receptor es bien sencillo: la secuencia DLE-STX se interpreta como inicio de trama, DLE-ETX como fin de trama y, en caso de recibirse la secuencia DLE-DLE, el receptor desecha automáticamente uno de los caracteres DLE por considerarlo una inserción del emisor y no parte de un delimitador de trama (ignorando el posible significado especial del carácter siguiente en caso de que fuese STX o ETX). De esta forma, podemos decir que el carácter de escape utilizado (DLE), al que se atribuye un significado especial (inicio de delimitador de trama), pierde ese significado especial si se duplica.

b) *Delimitación mediante bits de principio y fin con inserción de bits.* En este caso, en lugar de usar una combinación de caracteres, se considera una secuencia fija de bits como delimitador de trama, habitualmente tanto para el inicio como el fin de la misma.

En el protocolo LLC (ver Apartado 4.6), uno de los más extendidos, se utiliza el carácter 01111110 (correspondiente al carácter ASCII ~) como octeto delimitador (Figura 4.1(d)).

Al igual que en el esquema de delimitación anterior, la aparición del delimitador en los datos constituye un problema que hemos de resolver. Frente a la solución adoptada anteriormente, que en este caso pasaría, por ejemplo, por insertar una carácter prefijado cualquiera distinto del delimitador, la solución adoptada en LLC consiste en la inclusión, por parte del emisor, de un valor 0 tras la ocurrencia de cinco valores 1 consecutivos (Figura 4.1(e)). Este procedimiento es denominado *inserción de bits*.

Si bien la inserción de bits resuelve adecuadamente el problema de aparición del delimitador en la secuencia de datos, hemos de señalar que esta técnica introduce cierta dificultad en la transmisión dado que esta no corresponderá necesariamente a un número de bits múltiplo de ocho; es decir, el proceso de emisión-recepción no se lleva a cabo en base a octetos, sino a bits individuales.

Aunque los procedimientos de delimitación de trama más utilizados son los anteriormente comentados, otros posibles esquemas son los dos siguientes:

— *Cuenta de caracteres.* En este caso, como se muestra en la Figura 4.2, se precede la secuencia de bytes de datos a transmitir con uno o más octetos indicativos de su número, de forma que el receptor conoce perfectamente dónde finaliza la trama enviada.

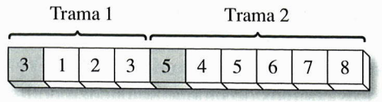


Figura 4.2. Delimitación de tramas mediante cuenta de caracteres.

El principal problema de esta técnica radica en el hecho de que un posible error de sincronización emisor-receptor provocaría la incorrecta delimitación, y consecuente recepción errónea, de la información a partir de ese punto, no siendo posible recuperarse del error.

— *Violación de código.* El empleo de ciertos códigos de línea es adecuado para delimitar mediante la técnica de violación de código ya comentada en el Apartado 2.3.2. Así, por ejemplo, en caso de utilizar codificación Mánchester podríamos considerar como delimitador un símbolo no previsto inicialmente en el código, como por ejemplo el formado por estados de señalización de tipo bajo-bajo o alto-alto, frente a los de tipo alto-bajo y bajo-alto aceptados en el esquema de codificación mencionado.

En la práctica, resulta habitual encontrar protocolos en los que se combinan algunas de las técnicas descritas. Así, es usual que se combine el empleo de delimitación mediante bits junto con la cuenta de caracteres.

Como se ha comentado en el apartado de introducción, aunque la delimitación es la primera de las funciones que hemos presentado, es la última en realizarse en la capa de enlace de la entidad emisora. Es decir, dada la secuencia de datos a transmitir, esta se procesa de forma conveniente, incluyéndose los bits de «cabecera» necesarios, para posibilitar el control de errores y de flujo; solo después se lleva a cabo la delimitación del conjunto de bits que forman la trama, cabecera incluida. También hemos de mencionar que, en la subdivisión de la capa de enlace en subcapa de control de acceso y subcapa de enlace, es la subcapa de control de acceso al medio la encargada de realizar la delimitación de trama.

4.3. Códigos para la gestión de errores

Uno de los procesos más importantes a considerar en una transmisión de datos es el control de errores. Dadas las limitaciones propias de los medios de comunicación (ver Apartado 2.2.2), es clara la necesidad de arbitrar técnicas que determinen la ocurrencia o no de errores debidos al canal de transmisión. Es lo que se conoce como *codificación del canal*. Para ello se precisa la inclusión de redundancia en las transmisiones para que el receptor pueda determinar si los datos son correctos o no. Para este fin se utilizan *códigos* que realizan una transformación de los datos a enviar, lo que permite detectar, y en algunos casos incluso corregir, los errores producidos.

Aunque existen otras alternativas, alguna de las cuales consideraremos más adelante, la mayoría de los códigos de canal utilizados en las redes de computadores pertenecen a los denominados *códigos de bloque*. En estos, los datos son transmitidos como bloques de bits de longitud fija, n, no siendo aceptadas todas las 2^n combinaciones posibles de bits de dicha longitud. Así, dada una secuencia de k bits de datos, el código de bloque asociará a dicha secuencia otra de longitud n. Al conjunto de combinaciones de bits válidas en un código de bloque se le denomina *vocabulario* del código y a cada una de las combinaciones válidas *palabra código* («codeword» en inglés).

Habitualmente, los códigos de bloque generan palabras código de n bits de datos, de los cuales $r = n - k$, bits de comprobación («check bits») o redundancia, se obtienen a partir de los k bits

de datos de entrada mediante operaciones aritméticas. La palabra código correspondiente se formará mediante la adición, normalmente al final del bloque, de los r bits de redundancia a los k bits de datos (Figura 4.3). El código resultante se dice que es un código (n, k) y se denomina a la relación k/n *velocidad* o *razón del código*. Los códigos de bloque se pueden definir a partir de matrices generadoras, aunque son posibles otras representaciones como las que usaremos en el resto del apartado.

La capacidad para detectar y, en su caso, corregir errores del código viene determinada por las propiedades que presenta el vocabulario. Dado que no todas las combinaciones de n bits corresponden a una palabra código, el receptor podrá determinar que ha recibido una secuencia incorrecta si esta no es una palabra código. A modo de ejemplo, consideremos el conocido código de comprobación de paridad, que no es sino un código de bloque en el que se añade un bit de comprobación de tal forma que el número de bits 1 en la palabra código sea par o impar, según se haya establecido. Supongamos el caso de paridad par para tres bits de datos (Figura 4.4(a)), que será un código (4,3). En este caso, si el emisor envía la secuencia 0110 y el receptor recibe 0010, este último podrá determinar que la secuencia recibida es errónea, puesto que no corresponde a ninguna palabra código en el vocabulario. Sin embargo, si recibe 0000 considerará que es correcta, puesto que es una palabra código del vocabulario. Vemos, por tanto, que con este código se puede detectar la aparición de un error de bit, pero no de dos (o más) errores, por cuanto que la aparición de dos errores puede transformar una palabra código en otra diferente también válida. Generalizando esta argumentación, podemos afirmar que las capacidades de detección de un código dependen del número de errores necesarios para que una palabra código se transforme en otra palabra código. Argumentando desde el punto de vista de diseño del código, el número de bits de comprobación a incluir dependerá de las capacidades de detección deseadas.

Es posible formalizar esta cuestión mediante la definición de algunos parámetros. Así, se define la *distancia de Hamming* de dos secuencias de bits como el número de posiciones en que difieren los valores en dichas secuencias. La distancia de Hamming puede obtenerse a partir de la cuenta del

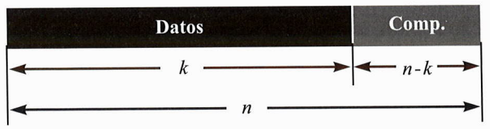
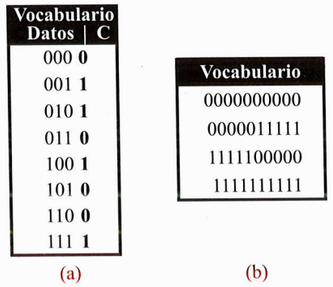


Figura 4.3. Estructura habitual de las palabras código en un código (n, k).



Vocabulario Datos	C
000	**0**
001	**1**
010	**1**
011	**0**
100	**1**
101	**0**
110	**0**
111	**1**

(a)

Vocabulario
0000000000
0000011111
1111100000
1111111111

(b)

Figura 4.4. Ejemplos de vocabularios de códigos de bloque. Paridad par para tres bits (a), código con capacidad para corregir dos errores (b).

número de bits a valor 1 resultantes de la operación XOR de ambas secuencias. Continuando con el ejemplo anterior, la distancia de Hamming entre las secuencias 0110 y 0000, ambas palabras código, es $d_H = 2$, ya que $0110 \oplus 0000 = 0110$. A partir de esta, se define la distancia de Hamming de un código como la mínima distancia de Hamming para cualquier pareja de palabras código del código. En el ejemplo del párrafo anterior $d_H = 2$.

A partir de esta definición es fácil deducir que para que un código tenga capacidad de detección de e errores, su distancia de Hamming debe ser $d_H \geq e + 1$; con ello se conseguiría que la modificación de e bits no convierta una palabra válida en otra palabra válida. En cambio, se puede demostrar que la distancia de Hamming de un código con capacidad de corrección de e errores debe ser $d_H \geq 2 \cdot e + 1$. Esto también resulta fácilmente comprensible si consideramos que la aparición de e errores no solo no debe convertir una palabra válida en otra palabra válida, sino que la errónea debe estar más cerca de la original correcta que de cualquier otra.

A modo de ejemplo, el código asociado al vocabulario en la Figura 4.4(b), que presenta $d_H = 5$, puede detectar hasta 4 errores y corregir hasta 2 de ellos. Así, si se recibe la secuencia 0000000111, el receptor podrá, bajo la hipótesis de que el número máximo de errores es 2, no solo detectar la aparición de los errores, sino también corregirlos, ya que, en este caso, la secuencia recibida debe proceder, necesariamente, de la secuencia 0000011111. Hemos de notar que únicamente podremos codificar 2 bits de datos, por cuanto que el número de palabras código es 4, lo que implica que se usan 8 bits de redundancia y el código resultante será (10,2). En resumen, como era presumible, la redundancia a incluir en una transmisión es inferior si se desea dotar al receptor únicamente de capacidad de detección de errores, que si se pretende que este sea capaz, además, de corregirlos. Este factor es determinante a la hora de diseñar un esquema de control de errores, ya que suele resultar más eficiente detectar y solicitar la retransmisión de los datos erróneos (usando técnicas ARQ), en escenarios con baja probabilidad de error, que incorporar la información necesaria para la corrección de los errores. Volveremos a esta cuestión en el Apartado 4.5.

Una última consideración a mencionar es que un código con capacidad de corrección de e errores presenta capacidad para detectar $2 \cdot e$ errores. Es decir, si la capacidad de corrección de errores es de 1 bit, la de detección será de 2 bits.

4.3.1. Códigos de corrección

Como ejemplo de códigos de corrección, es decir, aquellos en los que se dota al receptor de la capacidad de detectar y corregir in situ los potenciales errores aparecidos en la transmisión, podemos mencionar el *código de Hamming*. Este se caracteriza por la capacidad de corrección de errores simples, es decir, aquellos que afectan a un solo bit. Para ello, dada una secuencia de bits de datos a transmitir, se siguen los siguientes pasos (Figura 4.5(a)):

- Los bits de datos, m_i, se colocan en posiciones no potencia de 2 (3, 5, 6, 7, 9, ...).
- Las posiciones potencia de 2 (1, 2, 4, 8,...) se reservan para los bits de redundancia, obteniéndose de la siguiente forma:
  - Los valores no potencia de 2 se descomponen como sumas de potencias de este número. Así, por ejemplo, $3 = 2 + 1$, $5 = 4 + 1$, $6 = 4 + 2$, $7 = 4 + 2 + 1$, $9 = 8 + 1$, $10 = 8 + 2$, $11 = 8 + 2 + 1$, $12 = 8 + 4$, ...
  - A partir de la descomposición anterior, cada una de las posiciones potencia de 2, o sea, los bits de redundancia, se rellenará con un valor 1 o 0 según una comprobación de paridad par o impar realizada sobre los bits de datos en cuya descomposición aparece dicho bit de redundancia. Es decir, el bit de redundancia 1 controlará la paridad sobre los bits de datos 3, 5, 7, 9, 11, etc., el bit 2 lo hará sobre las posiciones 3, 6, 7, 10, 11, etc., y así sucesivamente.

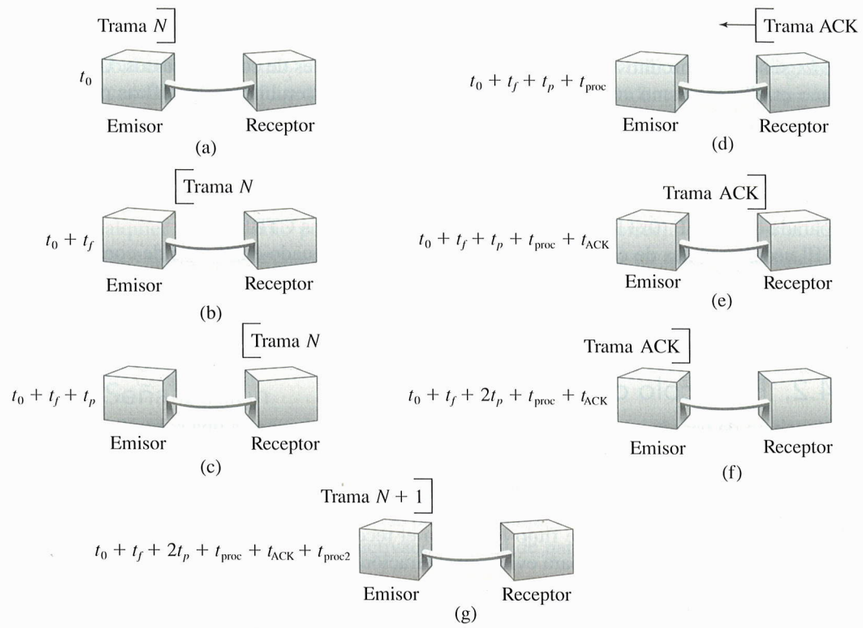


Figura 4.11. Temporización en la transmisión mediante parada y espera.

6. El emisor procesa la trama de confirmación en un tiempo t_{proc2}, distinto en general del tiempo de procesamiento de la trama de datos habida cuenta que las tramas a procesar son distintas, así como también los equipos terminales de datos origen y destino.
7. Tras dicho procesamiento habrá transcurrido un tiempo total desde a) igual a $t_f + 2t_p + t_{proc} + t_{ACK} + t_{proc2}$. Es justo en este instante de tiempo cuando el emisor se encuentra en disposición de proceder a la transmisión de la siguiente trama de datos, $N + 1$, hacia el receptor.

La implementación del protocolo de parada y espera es muy simple, pero presenta un importante inconveniente como es fácil de comprender a partir de la explicación desarrollada con anterioridad: se desaprovecha el canal debido a la espera entre transmisiones. Para evaluar dicho desaprovechamiento se puede estimar la eficiencia del protocolo, según (4.3). El tiempo útil, $t_{útil}$, será el empleado para la transmisión de las tramas, frente al tiempo total empleado en el procedimiento. Dado que la transmisión completa no es más que la repetición de un mismo proceso para cada trama (suponiendo que todas tienen la misma longitud y que tanto la velocidad de transmisión como el resto de términos temporales antes señalados al respecto de propagación y procesamiento son constantes), la eficiencia puede evaluarse sin más que considerar el envío de una trama y el tiempo total, t_{total}, que transcurre desde que se inicia su envío hasta que el sistema está en disposición de poder comenzar el envío de la trama siguiente (Figura 4.12). Utilizando (4.3), podemos evaluar U como:

$$U = \frac{t_f}{t_{total}} \tag{4.4}$$

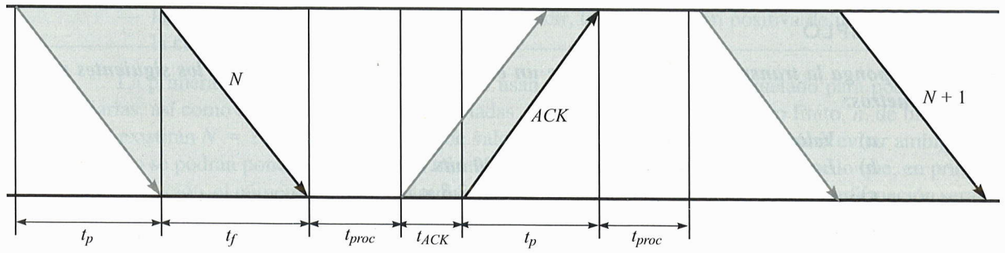


Figura 4.12. Evaluación del rendimiento en parada y espera. Se representa el inicio de cada trama en un tono más claro.

De acuerdo con lo detallado previamente con respecto al proceso de transmisión, es claro que $t_{total} = t_f + 2t_p + t_{proc} + t_{ACK} + t_{proc2}$, por lo que

$$U = \frac{t_f}{t_f + 2t_p + t_{proc} + t_{ACK} + t_{proc2}} = \frac{1}{1 + 2\frac{t_p}{t_f} + \frac{t_{proc}}{t_f} + \frac{t_{ACK}}{t_f} + \frac{t_{proc2}}{t_f}} \tag{4.5}$$

Considerando que $t_f \gg t_{ACK}$, t_{proc} y t_{proc2}, asunción fácilmente aceptada, podemos despreciar varios términos del denominador en (4.5), quedando la expresión para la eficiencia del protocolo de parada y espera en ausencia de errores como sigue:

$$U_{PE} = \frac{1}{1 + 2a} \tag{4.6}$$

definiéndose el parámetro adimensional a, como ya hicimos en (3.22), como el cociente entre el tiempo de propagación en el medio, t_p, y el tiempo de generación de trama, t_f; o sea,

$$a = \frac{t_p}{t_f} = \frac{d \cdot v_t}{L \cdot v_p} \tag{4.7}$$

En la Figura 4.13 se muestra gráficamente el comportamiento de U frente a a dado en (4.6). La justificación teórica de dicho comportamiento puede explicarse de forma breve como sigue:

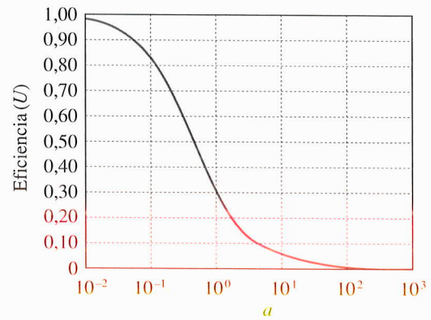


Figura 4.13. Eficiencia del protocolo de parada y espera en función del parámetro a en ausencia de errores.

EJEMPLO

Suponga la transmisión de datos entre un emisor y un receptor en base a los siguientes parámetros:

a) ***Velocidad de transmisión: 1 Mbps.***
b) ***Longitud de las tramas de datos: 1.000 bits.***
c) ***Velocidad de propagación en el medio: $2{,}8 \times 10^8$ m/s.***

Estime la eficiencia de la transmisión si se implementa un protocolo de parada y espera y se consideran los siguientes entornos de trabajo:

— ***Entorno LAN con una distancia máxima entre estaciones igual a 1 km.***
— ***Comunicación por satélite en órbita geoestacionaria (36.000 km).***

La repuesta a las dos situaciones planteadas es inmediata:

$$U = \frac{1}{1 + 2 \cdot \dfrac{d/2{,}8 \cdot 10^8}{1.000/1.000.000}}$$

tomando el parámetro d los siguientes valores en función del entorno de trabajo: $d_{LAN} = 1.000$ metros y $d_{sat} = 72.000.000$ metros.

En consecuencia, $U_{LAN} = 0{,}9929$, mientras que $U_{sat} = 0{,}0019$.

a) Si a aumenta, ya sea porque lo hace d o v_t o porque disminuye v_p o L, es claro que el tiempo de espera será mayor en relación al de transmisión. En consecuencia, la eficiencia disminuirá.
b) En cambio, si el valor de a disminuye, ya sea porque lo hace d o v_t o porque aumenta v_p o L, el tiempo de espera será menor en relación al de transmisión. Por tanto, en este caso la eficiencia aumentará.

De acuerdo con estos resultados, es de esperar que la eficiencia del protocolo de control de flujo de parada y espera sea mayor en entornos LAN que en comunicaciones satélite, estas últimas caracterizadas por distancias emisor-receptor muy elevadas y velocidades de transmisión menores.

4.4.3. Esquema de ventana deslizante

El principal inconveniente que presenta el protocolo de parada y espera es la, generalmente, baja eficiencia conseguida debido a la elevada proporción entre el tiempo consumido en la espera frente al empleado en la transmisión efectiva de los datos. Esto es debido fundamentalmente al hecho de que únicamente puede haber una trama en circulación. Para solventar este problema puede permitirse la existencia de varias tramas en tránsito, aunque esto implicará una mayor complejidad en la gestión del protocolo. Así, entre otras funcionalidades, habrá que contabilizar las tramas enviadas pendientes de confirmación, las tramas recibidas que no han sido confirmadas, gestionar los temporizadores de cada trama, etc.

Las técnicas de *ventana de deslizante* incorporan todas las funciones necesarias para el control de flujo en el caso en que:

— se permite el envío de varias tramas de datos consecutivas sin necesidad de haber recibido la confirmación de cada una de ellas, y

— las confirmaciones son acumulativas, es decir, la confirmación positiva de una trama implica la confirmación de todas las anteriores.

La primera cuestión que surge cuando se usan varias tramas es su etiquetado para poder diferenciarlas, así como a las confirmaciones asociadas. Para ello se usará un número finito, n, de bits, por lo que existirán $N = 2^n$ etiquetas posibles (de valores 0 a $N - 1$). Evidentemente, para evitar ambigüedades, no se podrán poner en circulación tramas diferentes con la misma etiqueta, por lo que, en primera aproximación, el número máximo de tramas que pueden encontrarse pendientes de confirmación será N. Las tramas que se vayan enviando se etiquetarán secuencial y cíclicamente de 0 a $N - 1$ (Figura 4.14).

Por otra parte, y para posibilitar algunas mejoras que describiremos con posterioridad, las confirmaciones positivas incorporan la etiqueta de la siguiente trama esperada, esto es, la trama siguiente a la última confirmada. Así la recepción de un ACK3 implica la confirmación de todas las tramas anteriores a la 3, indicando así que esta es la próxima trama que se espera.

Para gestionar las listas de tramas y confirmaciones, el protocolo de ventana deslizante utiliza el concepto de ventanas de envío y de recepción, que consisten en (Figura 4.15):

— *Ventana de envío (o de emisión).* Se utiliza para la gestión de las tramas en el emisor. Dado un flujo de datos procedente de la capa inmediatamente superior, y una vez delimitados estos en tramas, la transmisión se realiza en base a la consideración de una *ventana de envío* (*buffer*) de longitud $N_{envío}$, correspondiente al número máximo de tramas que se pueden enviar de forma consecutiva sin que se hayan recibido confirmaciones. Evidentemente, $N_{envío} \leq 2^n$. Así, por ejemplo, en un protocolo de ventana deslizante de 3 bits, el número máximo permitido de tramas a transmitir de forma consecutiva sin esperar confirmación será $N_{envío} = 8$ (Figura 4.15).

La ventana de emisión se desplaza a lo largo de la secuencia de tramas de la siguiente forma (Figura 4.15(a)):

a) Las tramas en la ventana son las que el emisor tiene autorizadas a transmitir de forma consecutiva en un momento dado, correspondiendo las situadas a su izquierda a las tramas ya enviadas y confirmadas por el receptor y las situadas a la derecha de la ventana a las aún por considerar para su transmisión.
b) Se considera un puntero de envío dentro de la ventana que indica la trama siguiente a transmitir y que avanzará a medida que se van enviando tramas. Este puntero separa las tramas ya enviadas y no confirmadas de las aún por transmitir.
c) La ventana presenta dos límites, el inferior, el cual se desplaza hacia delante en base a la recepción de confirmaciones correspondientes al envío de las tramas consideradas en la ventana, y el superior, que estará $N_{envío}$ posiciones por encima del límite inferior. De esta forma, la ventana de envío va desplazándose a medida que se reciben confirmaciones.

A modo de resumen, podemos afirmar que la ventana de envío contiene todas las tramas que pueden ser enviadas o reenviadas por el emisor en un momento dado sin que se tengan que recibir confirmaciones.

0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	...

Figura 4.14. Numeración de tramas en ventana deslizante para $n = 3$ ($N = 2^3 = 8$).

Suponiendo que $t_{NACK} = t_{ACK}$, a partir de (4.6), la expresión de la eficiencia del protocolo de parada y espera para el caso real de que aparezcan errores en las tramas de datos es:

$$U = \frac{1}{N_t(1 + 2a)} \tag{4.9}$$

donde N_t es el número medio de transmisiones necesario para conseguir transmitir con éxito una trama de datos. Supuesto que la probabilidad de error en el medio para una trama es P, N_t puede obtenerse a partir de:

$$N_t = E[\text{trans.}] = \sum_{i=1}^{\infty} i \cdot \Pr[i\,\text{trans.}] = \sum_{i=1}^{\infty} i \cdot P^{i-1}(1 - P) = \frac{1}{1 - P} \tag{4.10}$$

donde cada uno de los términos de la serie corresponde a la ocurrencia de i transmisiones por trama, de las cuales $i - 1$ serán incorrectas y, claro está, solo una, la última, correcta. Sustituyendo (4.10) en (4.9), la expresión final para la eficiencia del protocolo de control de flujo de parada y espera considerando errores en la transmisión resulta:

$$U_{PE} = \frac{1 - P}{1 + 2a} \tag{4.11}$$

De esta expresión se concluye que si $P \rightarrow 0$, la eficiencia corresponde con la dada en (4.6). En cambio, si $P \rightarrow 1$, $U \rightarrow 0$, lo que se justifica por el hecho de que todas las transmisiones serían erróneas. Esto viene a confirmar el hecho de que para valores elevados de P resulta más productivo hacer uso de técnicas de corrección de errores.

4.5.2. ARQ de vuelta atrás *N*

El protocolo ARQ de vuelta atrás *N* retroceso *N* es un esquema de ventana deslizante en el que los errores se recuperan mediante el reenvío de la trama de datos errónea y de todas las tramas de datos que se hubiesen podido enviar tras la errónea. Por tanto, no tiene sentido almacenar tramas posteriores a una

EJEMPLO

¿Cómo se modificaría la expresión de la eficiencia del protocolo de parada y espera (4.9) si en lugar de utilizarse confirmaciones negativas para indicar la ocurrencia de errores se recurriese exclusivamente a la retransmisión de las tramas en base a la expiración del temporizador de reenvío asociado?

Para dar respuesta a esta cuestión hemos de tener en consideración dos tipos de envío en la transmisión:

— Correcto: uno solo, el último. En este caso, el tiempo involucrado es el ya conocido por los estudios anteriores: $t_f + 2 \cdot t_p$.
— Incorrectos: $N_t - 1$. El tiempo involucrado en este caso para cada envío será $t_f + t_{out}$.

Por tanto, la expresión final para la eficiencia en el supuesto planteado será:

$$U = \frac{t_f}{(N_t - 1) \cdot (t_f + t_{out}) + t_f + 2 \cdot t_p} = \frac{1}{(N_t - 1) \cdot (1 + t_{out}/t_f) + 1 + 2 \cdot a}$$

errónea o fuera de secuencia, ya que serán retransmitidas tras la errónea. Es decir, es obligatoria la recepción ordenada de las tramas. En consecuencia, el tamaño de la ventana de recepción es uno, ya que el receptor solo acepta en cada instante de tiempo la trama siguiente a la última recibida correctamente.

Al igual que en el caso del ARQ de parada y espera (ver Apartado 4.5.1), es necesario el uso de temporizadores para evitar los problemas derivados de la pérdida de tramas.

Antes de analizar el comportamiento ante cada potencial situación de error, hemos de puntualizar algunas aclaraciones respecto de lo estudiado en el Apartado 4.4.3, donde se obvió la posible ocurrencia de errores en las transmisiones, lo cual, desgraciadamente, no se corresponde con la realidad. La aparición de errores debidos al canal tiene así implicaciones importantes en lo establecido de forma genérica hasta el momento para el esquema de ventana deslizante. La primera de ellas está relacionada con el tamaño de la ventana considerada. Si se utilizan n bits para numerar secuencias, hemos argumentado que el tamaño máximo de las ventanas debería ser $N = 2^n$, ya que en caso contrario se podrían poner en juego dos tramas con la misma numeración. Sin embargo, en el caso real de posible aparición de errores, esta limitación ya no es válida. En este caso existen situaciones en las que se pueden producir ambigüedades. Veámoslo con un ejemplo. Supongamos $n = 3$ y que podemos utilizar $N = 8$. En esta situación, el emisor podría enviar todas las tramas en su ventana de envío (Figura 4.20), desde la 0 a la 7. Podría darse el caso, mostrado en la Figura 4.20(a), de que la confirmación acumulada de todas las tramas, ACK0, se pierda. Antes del envío de ACK0, las ventanas de envío y recepción estarían como se muestra en la Figura 4.20(b), de forma que A no puede enviar más tramas, ni B aceptará nuevas tramas. Al enviar el ACK0, B desplaza el puntero y la ventana, mientras que A, al no recibir la confirmación, acabaría activando la retransmisión de la trama 0 (el puntero de envío vuelve atrás). El resultado (Figura 4.20(c)) es que la retransmisión de la trama 0 se interpreta en B como una nueva trama 0. La consideración de tramas de confirmación negativas no soluciona el problema, como podemos deducir fácilmente.

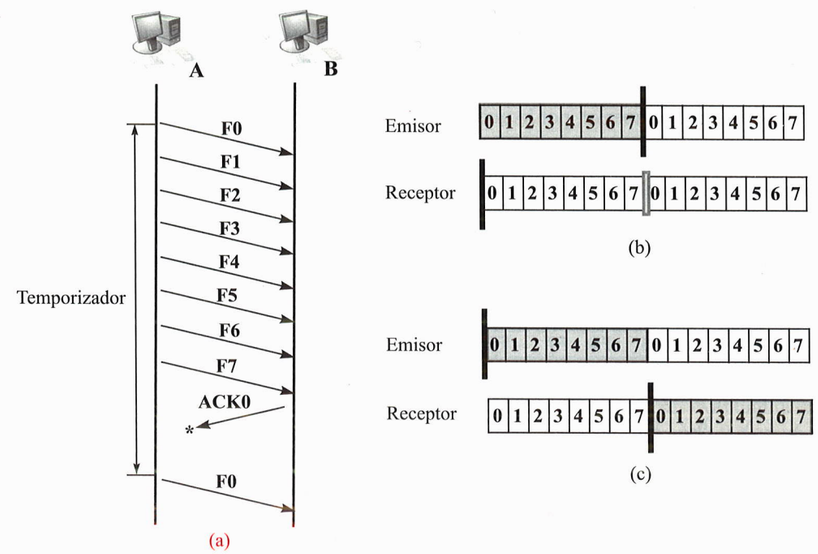


Figura 4.20. Ambigüedad generada por la aparición de errores si $N = 8$ con $n = 3$. Esquema del intercambio de tramas (a), ventanas antes del envío de ACK0 (b) y después de la expiración del temporizador (c).

- *No numeradas*. Como se ha establecido al principio, LLC ofrece un servicio orientado a conexión. Esto implica que, previamente a la transferencia de los datos, se precisa el establecimiento de una conexión entre el emisor y el receptor, así como la desconexión posterior. Es evidente entonces que se requiere la existencia de tramas distintas de las de información y supervisión con objeto de llevar a cabo la gestión de la conexión. Son las tramas conocidas como *no numeradas*. Como se muestra en la Figura 4.23(b) (campo *M*), para la definición de este tipo de tramas se utilizan 5 bits. De las 32 posibles tramas no numeradas hemos de decir que no todas se utilizan ni siquiera en HDLC. En LLC solo se incluyen algunas de las definidas en HDLC y se incorporan dos tramas no numeradas más, pudiéndose destacar de entre las existentes las siguientes:
  - SABM/SABME («Set ABM/E»): trama de establecimiento de conexión en modo de transferencia balanceado asíncrono (el único permitido en LLC), normal o extendido.
  - DISC («DISConnect»): trama de solicitud de desconexión.
  - UA («Unnumbered Acknowledgment»): Confirmación para tramas no numeradas.
  - FRMR («FRaMe Reject»): indicación de la recepción de una trama inaceptable.
  - UI («Unnumbered Information»): trama para el intercambio de información de control.

— *Campo de datos*, que contiene la información procedente de las capas superiores. Es de longitud variable con un tamaño máximo de 1.492 bytes.

Para concluir el estudio del protocolo LLC, en la Figura 4.24(a) se muestra un ejemplo de transmisión de datos en modo orientado a conexión con confirmación. En ella se muestra el establecimiento y cierre de una conexión a través del empleo de tramas no numeradas y la transferencia de datos mediante el uso de las tramas de información y de supervisión. También se incluyen ejemplos de intercambio de datos en diversas situaciones (subfiguras (b) a (e)). Nótese que las tramas están etiquetadas según su tipo y los datos relevantes ($N(S)$, $N(R)$ y P/F) que contiene el campo de control.

RESUMEN

A lo largo de este capítulo se han presentado las funciones propias de la capa de enlace de una red de computadores. Se ha comenzado estudiando la delimitación de tramas como proceso necesario para la sincronización en la transmisión emisor-receptor. Como técnicas desarrolladas para abordar esta cuestión se han presentado los esquemas de delimitación síncrona, consistentes en la consideración de caracteres de inicio y de fin de trama, comentándose el uso de inserción de caracteres y de bits con objeto de evitar problemas cuando las secuencias de caracteres usadas como delimitadores aparecen en el flujo de datos a transmitir. Aunque menos utilizadas que las anteriores, también se han presentado las técnicas de delimitación basadas en la cuenta de caracteres y en violación de código.

Dada la necesidad de arbitrar métodos que permitan resolver adecuadamente la existencia de errores en la transmisión, se han introducido técnicas tanto de corrección como de detección de estos. Presentados los conceptos de redundancia y distancia de Hamming, seguidamente se han descrito los códigos de Hamming como técnica de corrección de errores simples y las codificaciones basadas en suma de palabras y CRC como alternativas ampliamente usadas en detección y base para la puesta en práctica de las técnicas ARQ. Adicionalmente a los anteriores esquemas de codificación, también se han presentado los códigos convolucionales, ampliamente usados, entre otras, en esquemas de modulación y en telefonía celular.

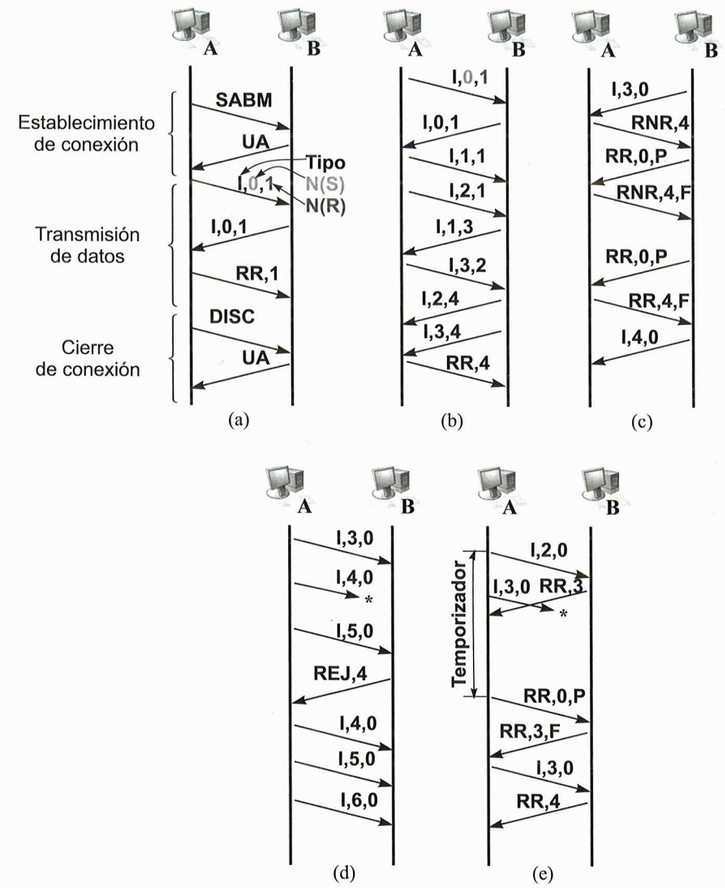


Figura 4.24. Ejemplos de transmisión en LLC en modo 2. Fases de la comunicación (a), intercambio de tramas (b), receptor ocupado (c), recuperación de rechazo (d) y recuperación de expiración con sondeo (e).

Un aspecto de suma importancia abordado en este capítulo es el referente al control de flujo. Planteando como objetivo la necesidad de una transferencia ordenada emisor-receptor, se ha partido del protocolo de parada y espera y se ha concluido con las técnicas de recuperación de errores adelante-atrás *N* y repetición selectiva, correspondientes ambas al esquema de ventana deslizante. Para todas las técnicas mencionadas se ha estimado su eficiencia, concluyéndose las mejores prestaciones que presentan las últimas, de implementación más compleja, frente a la de parada y espera, de concepción más simple. Además, se comenta que como ambas técnicas (control de errores y flujo) por lo general implican procedimientos de realimentación, en la práctica se suelen implementar conjuntamente.

El tema ha concluido con la descripción del protocolo de enlace LLC, de amplio uso en las redes LAN IEEE802.

EJERCICIOS

1. Explique los factores que justifican la necesidad de fragmentar los mensajes a transmitir sobre una red. ¿Cómo afectan estos de cara a la especificación del tamaño de trama óptimo a considerar?

2. Suponga que se desea llevar a cabo la transmisión de las siguientes tres tramas de información: 0101010011111101 1011111010001010 0101011001111110. ¿Cuál sería la secuencia de bits finalmente transmitida si se considera como método de delimitación de tramas el de «delimitación de principio y fin con relleno de bits» utilizado en el protocolo LLC?

3. Supuesto que se desea posibilitar el control de E posibles errores en una transmisión:

   a) ¿Cuál debe ser la distancia de Hamming de un código que permita la corrección de estos E errores? Ponga un ejemplo práctico de palabras código que permitan la corrección de $E = 3$ errores.
   b) Repita el apartado anterior para el caso de un código con solo capacidad de detección.

4. Un bloque de información consistente en 4 bits a valor 1 se va a transmitir mediante un código de Hamming.

   a) Determine la palabra código transmitida si se usa paridad par.
   b) Se recibe el mensaje $m = 1111011$. ¿Es correcto? En caso negativo, indique cuál sería este.

5. Obtener la secuencia de bits a enviar para la cadena de datos $M = 1110101011$ si se usa un CRC basado en el polinomio generador $G(x) = x^4 + x^3 + x + 1$.

6. Se lleva a cabo la transferencia de un fichero de longitud M bits entre dos estaciones finales, distanciadas entre sí D metros en una red con topología en bus. Si la velocidad de transmisión es B bps, la transferencia se realiza en tramas de P bits, H de los cuales son suplementarios, y cada trama se confirma siguiendo el esquema *parada y espera* mediante una especial de confirmación de longitud C bits:

   a) Estime el tiempo consumido en la transferencia del mensaje completo.
   b) ¿Cuál será la eficiencia del proceso si no se producen errores en el canal? ¿Y si la probabilidad de error fuese tal que hubiera que transmitir cada trama un promedio de N veces?

7. Los esquemas de ventana deslizante integran el control de flujo y de errores en un mismo procedimiento. Razone esta afirmación e idee un esquema propio que realice ambas funciones de forma independiente.

8. Un enlace transoceánico utiliza un procedimiento ARQ de *parada y espera* con un valor del temporizador igual a 20 ms, utilizándose tramas de tamaño 20.000 bits y una velocidad de transferencia de 10 Mbps. La longitud del cable submarino es de 5.600 km. Las confirmaciones tienen un tamaño de 100 bits y los tiempos de procesamiento se consideran despreciables.

   a) Realice un esquema de la secuencia de intercambios de tramas entre las dos estaciones cuando la estación A envía 3 tramas a la estación B y no aparecen errores durante la transmisión.
   b) ¿Cuál es la eficiencia del sistema resultante?
   c) Proponga otro mecanismo de control de enlace que consiga una eficiencia del 100% en ausencia de errores, indicando los parámetros de diseño del mismo.

9. Dos usuarios de una red han decidido implementar un protocolo simple de *parada y espera* para la transferencia de ficheros. Con este protocolo se aseguran la llegada al destino de bloques individuales de 1.000 bits (datos más control). Suponiendo que ambos usuarios se comunican usando enlaces a 128 kbps y que el retardo a través de la línea es únicamente debido a los medios físicos, se pide:

   a) Obtenga la máxima distancia de separación entre los dos usuarios para conseguir una eficiencia en el protocolo, sin errores, de al menos un 90%.
   b) Calcule el valor del temporizador de retransmisión de tramas más adecuado en las condiciones del apartado anterior.
   c) Suponiendo que se producen errores en la transmisión, determine qué campos de las tramas y/o variables de estado (y/o mecanismos adicionales) serían necesarios para evitar los duplicados de tramas de información ante la potencial pérdida de confirmaciones.

10. Considere un canal con una distancia extremo a extremo de 100 km, sobre el que se transmiten datos a una velocidad de 10 Mbps. La velocidad de propagación en el medio es 200 m/μs. Suponiendo el canal libre de errores, calcule:

    a) El tamaño mínimo de trama para que el protocolo de *parada y espera* proporcione una eficiencia superior al 60%.
    b) La máxima eficiencia que se podría conseguir usando el tamaño de trama obtenido en a) y el esquema ARQ *adelante atrás N* con numeración de trama de 4 bits.

11. Para un esquema de control de flujo y errores de *ventana deslizante* con *adelante atrás N* se establece que el tamaño máximo de la ventana de emisión es $2^n - 1$, siendo n el número de bits considerado para la numeración de las tramas. ¿Se vería afectado este límite superior en el caso hipotético de que se garantizase que las tramas no se pierden durante la transmisión? Justifique la respuesta.

12. Suponga que se modifica ARQ de *repetición selectiva* de forma que los mensajes ACK contengan una lista con las siguientes m tramas que espera el receptor. ¿Cuál es el efecto sobre el rendimiento del protocolo? ¿Cómo debe modificarse para adaptarse al cambio apuntado?

13. ¿Cómo justifica que, a pesar de su mayor eficiencia teórica, el procedimiento ARQ de *repetición selectiva* se utilice menos que el ARQ con *adelante atrás N*? ¿Por qué en el caso de comunicaciones satelitales se recurre usualmente al primero de los esquemas mencionados?

14. Se dispone de dos entidades de enlace A (emisora) y B (receptora) que distan D km entre sí y se comunican mediante un protocolo de *ventana deslizante* de tamaño de ventana N, en el que se envían tramas de L bits a una tasa de R bps. Dicho protocolo está levemente modificado de modo que A debe esperar un tiempo de E segundos entre el envío de una trama y el de la siguiente. Obtenga la eficiencia de este protocolo.

15. Dos nodos vecinos (A y B) usan un protocolo con *ventana deslizante* con 3 bits para los números de secuencia. Se utiliza como procedimiento ARQ un *adelante atrás N* con un tamaño de ventana de envío igual a 4. Supóngase que A transmite y B recibe. Muestre las distintas posiciones de las ventanas para la siguiente sucesión de eventos:

    a) Antes de que A envíe ninguna trama.

longitud fija igual a B bits. Para el control de flujo se utiliza un protocolo de ventana deslizante *adelante atrás N*. Obtenga la expresión que da el tamaño del campo de numeración de secuencia mínimo necesario en función de R, T, B y L (considerando utilización máxima). Suponga que las tramas de confirmación tienen un tamaño despreciable, que no existen errores y que el procesamiento en los nodos es instantáneo.

23. Se pretende diseñar un protocolo de nivel de enlace que implemente un servicio fiable de intercambio de tramas, con las siguientes especificaciones:

— Capacidad para operar en una red de difusión con hasta 60 equipos simultáneos.
— Longitud máxima de la red de difusión: 10 km.
— Cada nodo puede enviar datos a cualquier otro de forma asíncrona, es decir, espontáneamente y sin necesidad de esperar una pregunta de otro nodo.
— Velocidad de transmisión: 5 Mbps.
— Tamaño máximo de trama: 300 bytes.
— Esquema de control de flujo: *repetición selectiva*.
— En ausencia de errores se debe alcanzar una eficiencia (U) del 100% en la transmisión.
— El control de errores se realiza usando un código CRC obtenido con el siguiente polinomio generador: $x^6 + x^3 + x + 1$.

Responda a las siguientes cuestiones:

a) Proponga un formato de trama que incluya todas las funcionalidades precisas para un nivel de enlace genérico con las características anteriormente precisadas. Para ello, especifique los campos que deben componer la trama, su uso y significado, y el tamaño en bits de cada uno de ellos.
b) Indique si deben contemplarse varios tipos de tramas en este protocolo. En caso afirmativo, comente cuáles, su uso y cómo se distinguirían unas de otras.

BIBLIOGRAFÍA

Duck, M.: *Data Communications and Computer Networks*, Prentice-Hall, 2003, 2.ª edición.
Haykin, S.: *Communication Systems*. John Wiley & Sons, 2009, 5.ª edición.
León-García, A.; Widjaja, I.: *Redes de Comunicación. Conceptos Fundamentales y Arquitecturas Básicas*. McGraw-Hill, 2001.
Proakis, J., Salehi, M.: *Digital Communications*. McGraw-Hill, 2008, 5.ª edición.
Stallings, W.: *Comunicaciones y Redes de Computadores*. Prentice-Hall, 2004. 7.ª edición.
Sklar, B.: *Digital Communications: Fundamentals and Applications*, Pearson, 2013.
Shay, W.: *Understanding Data Communications and Networks*, Thomson, 2004, 3.ª edición.
Tanenbaum, A. D.: *Computer Networks*. Prentice-Hall, 2011. 5.ª edición.

CAPÍTULO 5

REDES DE ÁREA LOCAL Y DE ACCESO

5.1. Introducción
5.2. Redes de área local
5.3. Redes PAN
5.4. Redes de acceso

5.1. Introducción

Una vez estudiadas las capas física y de enlace estamos en disposición de analizar aquellas redes en las que no son imprescindibles las funcionalidades de la capa de red y superiores para la comunicación entre los equipos que la componen. Este es el caso de las redes LAN, donde no existen nodos intermedios para establecer rutas origen-destino. Por tanto, están compuestas únicamente por nodos terminales (Figura 1.4 y Capítulo 6).

Por otra parte, este tipo de redes constituye el elemento básico sobre el que se construyen redes complejas en base a su interconexión con otras redes, aspecto que será abordado en capítulos posteriores.

En este contexto resultan de interés las denominadas redes SOHO («Small Office-Home Office»), ampliamente utilizadas en el ámbito residencial y de pequeñas empresas y compuestas por una red LAN, o un número reducido de ellas, a la que se proporciona acceso a Internet a través de una conexión de acceso. Dada su relevancia, y antes de proceder al estudio del nivel de red según el análisis en capas propuesto al comienzo de este texto, a lo largo de este capítulo estudiaremos los estándares LAN y las tecnologías más difundidas para la conexión de estas a Internet a fin de proporcionar la interconectividad global de sistemas y usuarios.

Abordaremos este estudio considerando primero las redes de área local, su arquitectura y los aspectos generales relativos a la capa física y la subcapa de control del enlace. A partir de estos se presentarán las dos tecnologías de redes LAN más utilizadas en la actualidad: Ethernet, para redes cableadas,