

Capítulo **5**. DISEÑO DEL NIVEL DE ENLACE Y CONTROL DE ERRORES

En este capítulo se presentan las funciones del nivel de enlace de la arquitectura de red OSI. Dentro de estas funciones cabe mencionar algunas como las de crear y delimitar tramas, identificar estaciones, control de flujo, etc. Una de las principales funciones es la de ofrecer al nivel de red un enlace libre de errores. En este capítulo se describirán también distintos métodos para que un receptor sea capaz de determinar si una trama es correcta o si algún bit se ha alterado durante la transmisión. También se mostrarán algoritmos para la corrección de dichos errores.

5.1 Servicios del nivel de enlace

El objetivo fundamental del nivel de enlace del modelo OSI es el de ofrecer un enlace lógico sobre el medio físico libre de errores al nivel de red. La capa de enlace está en un nivel superior a la capa física. Por lo tanto, la capa de enlace de datos es responsable de la transferencia fiable de información a través de un circuito de transmisión de datos. A este nivel la transmisión de datos se realiza mediante tramas que son las unidades de información con sentido lógico para el intercambio de datos en la capa de enlace (ver Figura 5-1).

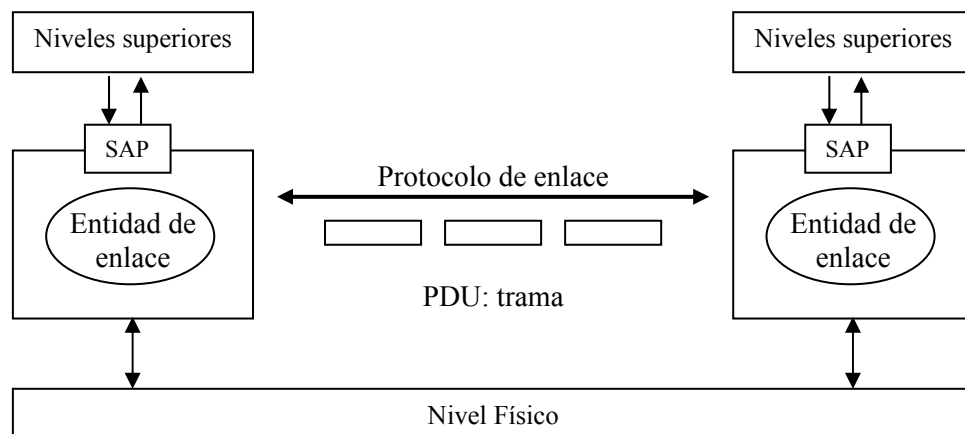


Figura 5-1. Nivel de enlace del modelo OSI

Según el modelo OSI, la capa o nivel de enlace se encarga de los servicios de transferencia de datos seguros a través del enlace físico. Por lo tanto, envía bloques de datos denominados tramas, llevando a cabo la sincronización, el control de errores y de flujo necesario. La capa de enlace pretende conseguir una comunicación eficiente, es decir, aprovechar al máximo el medio de transmisión. Además, esa comunicación debe realizarse de manera fiable, es decir, debe garantizarse que los datos lleguen correctamente al destino. En esta comunicación pueden surgir algunos problemas como son posibles errores en el medio de transmisión o, por ejemplo, que el emisor realice el envío de datos a una velocidad mayor a la que es capaz de procesar el receptor. El nivel de enlace debe proporcionar protocolos que consigan realizar una comunicación eficiente y fiable y que sean capaces de resolver los problemas como los anteriores.

Como se ha indicado anteriormente, el nivel de enlace de datos detecta y corrige los errores que ocurren en el nivel físico, proporcionando un medio libre de errores de transmisión al nivel de red. Para ello, trocea los datos del nivel de red en tramas de datos, los transmite secuencialmente, y procesa las tramas reconocidas, ofreciendo una comunicación fiable y eficiente entre dos máquinas adyacentes. Como se describirá a lo largo de este capítulo, el nivel de enlace implementará algoritmos de detección y/o corrección de errores que permiten detectar si se ha producido un error en la transmisión de algunos de los bits que constituyen la trama y, en ese caso, corregir los errores si es posible.

Para evitar el segundo problema indicado anteriormente, es decir, que el emisor trabaje a una velocidad mayor a la que es capaz de procesar al receptor, se implementan técnicas denominadas de control de flujo, con las cuales se sincroniza el envío y confirmación de datos entre emisor y receptor. Los circuitos de comunicación tienen una velocidad de transmisión de datos finita y hay un retardo de propagación diferente de cero, es decir, existe un determinado tiempo que transcurre desde que un bit se envía hasta que se recibe. Estas limitaciones, asociadas con la velocidad finita de las máquinas para

procesar los datos, tienen implicaciones muy importantes en la eficiencia de la transferencia de datos. El nivel de enlace se diseña para paliar estos problemas y que la comunicación sea fiable y eficiente a través de una serie de algoritmos que los resuelven.

Como resumen, cabe mencionar que el servicio principal de la capa de enlace es la transferencia de datos de la capa de red de la máquina origen a la capa de red de la máquina destino. El nivel de enlace proporciona al nivel de red diferentes servicios de acuerdo a la calidad que se quiere conseguir en la transmisión:

- *Servicios sin conexión y sin reconocimiento.* En este tipo de servicio no se establece conexión previa y las tramas se envían a la máquina destino sin acuse de recibo, es decir, no se confirma la recepción correcta de los datos. Es válido para aplicaciones en tiempo real (Se emplea en medios físicos donde los retardos perjudican mas que los errores). Además, este tipo de servicios es adecuado en líneas de comunicación con tasa de errores bajas.
- *Servicios sin conexión y con reconocimiento.* En este caso tampoco se establece conexión previa, pero ahora cada trama enviada por el emisor es notificada y confirmada por el destino empleando tramas de asentimiento ACKs (por lo tanto, se llevará a cabo un reenvío de tramas ante errores). Es apropiada para líneas con errores y evita la sobrecarga en niveles superiores. Este tipo de servicio es ofrecido por el nivel de enlace en redes inalámbricas ya que se busca comunicación fiable.
- *Servicios con conexión y con reconocimiento.* Las tramas no se envían independientemente, sino que son numeradas y antes del envío de las mismas es necesaria una fase de establecimiento de la conexión en la que se negocian los parámetros de comunicación entre emisor y receptor. Una vez establecida la conexión se envía la información y finalmente, tras ser recibida y confirmada, se cierra la conexión. Se usan tramas numeradas y cada trama se recibe una única una vez. Estas serán confirmadas por el receptor y si hay errores se producirá el reenvío de los datos erróneos. Este tipo de servicios es utilizado en comunicaciones donde prima la fiabilidad.

5.2 Funciones del nivel de enlace

En el apartado anterior se han ido enunciando las principales características del nivel de enlace. Entre todas ellas la más importante es garantizar que los datos llegan sin errores y ordenados al receptor sea cual sea el medio físico empleado. Las funciones del nivel de enlace que permiten llevar a cabo los servicios indicados en el apartado anterior al nivel de red son las siguientes:

- *Iniciación y terminación de la comunicación.* El inicio, mantenimiento y la conclusión del envío de datos requiere una cantidad razonable de coordinación y cooperación entre las estaciones emisora y receptora. Se necesitan pues una serie de procedimientos para gestionar este intercambio. Mientras que la capa física proporciona solamente un servicio bruto de flujo de datos, la capa de enlace de datos intenta hacer el enlace físico seguro y proporciona medios para activar, mantener y desactivar el enlace.
- *Delimitación de tramas.* Los datos se envían en bloques que se denominan tramas. El comienzo y el final de cada trama debe identificarse. Puesto que la capa física se encarga de la transmisión de cadenas de bits no estructuradas, una de las funciones del nivel de enlace será agrupar estos bits en tramas en las que se incluirán, además de los bits de información, campos de delimitación de la trama para saber cuál es el inicio y el final de la misma.
- *Direccionamiento de tramas.* En diversos tipos de redes como pueden ser las multipunto y las de difusión es necesario algún tipo de identificación de las estaciones que conforman la red. El

direccionamiento de tramas permite identificar la estación origen y destino de la comunicación.

- *Control de errores.* Todos los errores en los bits generados por el sistema de transmisión se deben corregir o, al menos, detectar. Entre las principales funciones que realizan el nivel de enlace están la detección y corrección de dichos errores. Para ello se implementan mecanismos para detectar cuándo ha habido errores en una trama y posteriormente corregir los mismos si es posible o interesa hacerlo (En ocasiones es más costosa la corrección de errores que realizar el reenvío de la trama errónea)
- *Control de flujo.* La estación emisora no debe enviar datos a una velocidad mayor que la estación receptora pueda absorber. Para ello, el nivel de enlace se encarga también del control de flujo, es decir, de la regulación del ritmo de envío de tramas desde el transmisor con el fin de no congestionar a los receptores lentos.
- *Gestión y coordinación de la comunicación.* Se encarga de establecer quien tiene la responsabilidad en la gestión del enlace, es decir, si existe una estación que controla el enlace o si esa gestión es distribuida. Dependiendo del tipo de gestión del enlace, las entidades de nivel de enlace se encargarán de determinar como se coordinan las distintas estaciones de la red para realizar el envío de los datos a los respectivos destinos.

En los siguientes apartados se realizará una descripción detallada de cada una de las funciones del nivel de enlace enumeradas en este apartado.

5.3 Iniciación y terminación de la comunicación

La función de *iniciación* comprende los procesos necesarios para activar el enlace e implica el intercambio de tramas de control con el fin de establecer la disponibilidad de las estaciones para transmitir y recibir información.

Las funciones de terminación se emplean para liberar los recursos ocupados después de la recepción/envío de la última trama. Para lograr el intercambio de información entre las estaciones a través del enlace se pueden considerar las fases siguientes:

- *Conexión del circuito.* Consiste en la obtención de un circuito físico que conecta las estaciones. En el caso de líneas conmutadas comprenden todos los procesos necesarios para el establecimiento del circuito (marcación, conmutación, etc.).
- *Establecimiento del enlace lógico.* Comprende los procesos necesarios para poder iniciar la transferencia de datos a través de un circuito físico ya establecido. La estación que quiera iniciar la transferencia deberá comprobar si la otra está preparada para recibir, si se disponen de los recursos necesarios, etc. Durante esta fase la información intercambiada por el enlace es información de control generada en ambas estaciones.
- *Transferencia de datos.* Comprende todos los procesos necesarios para lograr la transferencia de mensajes de una estación a otra.
- *Terminación.* Durante esta fase tiene lugar la liberación de los recursos asociados a la transferencia de información (buffers en las estaciones, etc.).
- *Desconexión del circuito.* Comprende los procesos de liberación del circuito. Las fases de conexión y desconexión del circuito son competencia del nivel físico, en el resto de fases interviene el nivel de enlace.

5.4 Delimitación de tramas

Como se ha indicado anteriormente la unidad de información con la que se trabaja a nivel de enlace es la trama. Una trama no es más que una agrupación de bits en una unidad de información mayor. Una función del nivel de enlace consiste en identificar el inicio y fin de la trama de datos. En general, las tramas presentan la siguiente estructura:

CABECERA	PAQUETE DEL NIVEL DE RED	CONTROL DE ERRORES (FCS)	COLA
----------	-----------------------------	-----------------------------	------

Figura 5-2. Estructura de una trama

Los distintos campos representados en la Figura 5-2 son los siguientes:

- *Cabecera.* La cabecera identifica el comienzo de la transmisión, por lo tanto, delimita el inicio de la trama. En el campo de cabecera también se especifican las direcciones de las estaciones origen y destino. Asimismo, puede especificarse otra información de control como un código identificador de la trama, el tipo de trama (si es de datos, confirmación, u otro tipo de trama de control) o longitud de la trama. Cuando se requiere sincronización en la cabecera puede especificarse una secuencia especial de bits que es empleada por el receptor para sincronizarse con el emisor.
- *Paquete de nivel de red.* En este campo se encapsula el paquete generado a nivel de red. En muchas ocasiones este campo presenta un tamaño máximo por lo que si el paquete a nivel de red sobrepasa este tamaño máximo será necesario realizar la fragmentación del paquete original.
- *Control de errores (FCS, Secuencia de verificación de trama).* Este campo es utilizado para verificar si la transmisión se ha llevado a cabo correctamente, es decir, los bits que recibe el receptor no se han alterado respecto a los enviados por el emisor.
- *Cola.* El emisor tiene la responsabilidad de agrupar la información en tramas que serán retransmitidas posteriormente a través del nivel físico como una secuencia de bits. El receptor recibe los bits individuales procedentes del nivel físico y ha de agruparlos en tramas, indicando donde empieza y acaba cada uno. El inicio de la trama se especifica en la cabecera y el final de la misma en la cola.

Por lo tanto, una de las principales funciones del nivel de enlace es delimitar el inicio y final de las tramas y para ello puede emplearse algunas de las siguientes aproximaciones que serán descritas en los siguientes apartados:

- Delimitación temporal.
- Delimitación por numeración de caracteres.
- Delimitación por caracteres especiales
- Delimitación por bits especiales.

5.4.1 Delimitación temporal

Una posibilidad de delimitar una trama es lo que se denomina dejar “silencios” entre tramas, es decir, dejar de transmitir entre trama y trama. De esta forma, cuando se detecta un silencio será por que ha finalizado la transmisión de la trama.

Otra manera de realizar una delimitación temporal es suponer que todas las tramas presentan el mismo tamaño. En este caso no es necesario emplear cabeceras adicionales con el objetivo de la delimitación. El tamaño se asigna en función de una estimación de tiempo que tarda la trama en llegar al receptor. Este método es sensible a los retardos lo que provoca que se introduzcan errores.

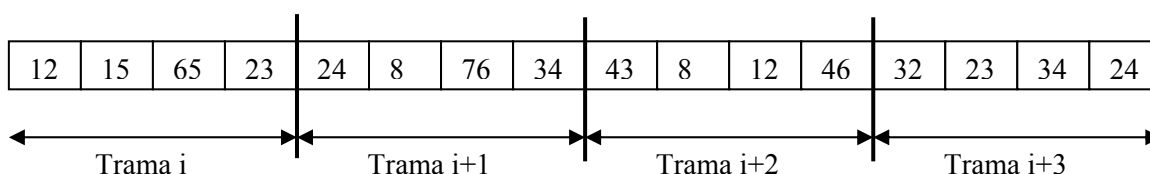


Figura 5-3. Delimitación temporal de tramas

5.4.2 Delimitación por número de caracteres

En este caso sí que es necesario emplear una cabecera para delimitar la trama. En concreto, en la cabecera se indica el número de bytes que posee la trama. Este método es sensible al ruido, ya que si hay variaciones en los datos de la cabecera debido al ruido se pierde la sincronización.

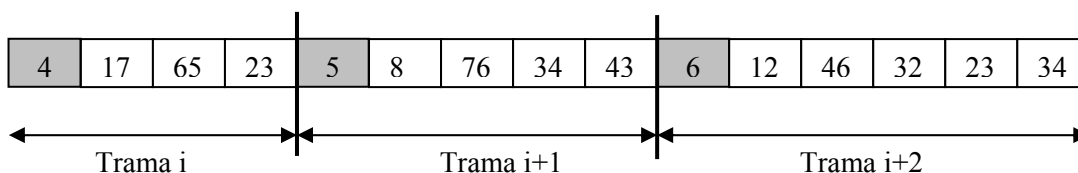


Figura 5-4. Delimitación por número de caracteres

5.4.3 Delimitación por caracteres especiales

Para realizar esta delimitación la trama a nivel de enlace debe presentar una cabecera y una cola. Tanto la cabecera como la cola almacena caracteres especiales que no se emplean en los datos transmitidos. Estos caracteres especiales suelen ser caracteres ASCII con código < 32 ya que se suponen que están reservados y no se emplean en la transmisión de datos alfabéticos (p. ej. caracteres ASCII SYN, DLE, STX, ACK, NACK, etc). Por lo tanto, este método se emplea cuando los datos transmitidos son caracteres de texto. Además, si algún carácter especial se usa como dato, se usa un carácter de relleno que no es más que el carácter duplicado.

En la Figura 5-5 se muestra un ejemplo de trama delimitada por caracteres especiales. Se emplea la secuencia de caracteres DLE, STX para delimitar la trama. Los datos transmitidos son los caracteres A, B, C, D, F, G, DLE, C, F, H. Como dentro de los datos aparece el carácter especial DLE que en principio está reservado a la delimitación, éste carácter aparece duplicado en la trama.

DLE	STX	A	B	C	D	F	G	DLE	DLE	C	F	H	DLE	STX
-----	-----	---	---	---	---	---	---	-----	-----	---	---	---	-----	-----

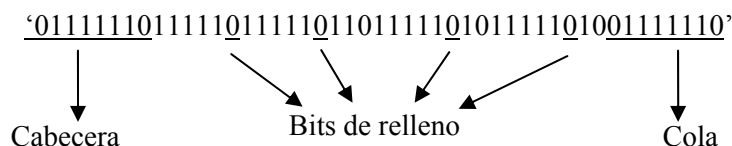
Figura 5-5. Delimitación por caracteres especiales.

5.4.4 Delimitación por bits especiales

Al igual que en el caso anterior, ahora también se van a emplear cabeceras y colas para la delimitación. La cabecera y la cola almacenan una secuencia de bits única que no se repetirá en los datos de la trama para indicar cual es el inicio y final de la trama. Esta secuencia de bits o delimitadores es una combinación especial de bits llamada flag que indica el comienzo y final de la trama. El procedimiento es similar a la delimitación de tramas por caracteres especiales pero en este caso a nivel de bit.

Para evitar que la secuencia de bits especial pueda aparecer como datos se usan 'bits de relleno'. Esto se observa en el siguiente ejemplo:

Supóngase que se desea transmitir los siguientes bits de datos: '11111111111011111101111110'. Además como delimitador de inicio y fin se va a considerar la secuencia de bits 01111110. Por lo tanto, la trama final que se enviaría, es decir, los datos junto con los delimitadores será:



Se observa que se introduce un '0' en los datos cada vez que aparece la secuencia 011111. Cuando esta trama llegue al receptor comprobará cual es el valor del siguiente bit cada vez que recibe la secuencia 011111. Si ese bit es '1' interpretará que es el fin o inicio de trama. Mientras que si es '0' elimina el bit ya que se tratará de un bit de relleno.

5.5 Direccionamiento de tramas

El direccionamiento de las tramas a nivel de enlace es empleado para identificar el origen y el destino de la trama de datos y permite identificar las estaciones o extremos de la comunicación que intercambian la información.

Los métodos de direccionamiento pueden clasificarse en primer lugar en función de la forma de especificar el origen y el destino en:

- *Implícito.* No es necesario especificar las direcciones de las estaciones origen y destino. Este tipo de direccionamiento se emplea en conexiones punto a punto en las que los datos enviados

por el origen llegan al destino sin necesidad de especificar el destino ya que existe una única posible estación destino.

- *Explicito.* En este caso sí que hay que indicar cual es la dirección origen y la destino en la trama. Cada equipo conectado a la red presenta una dirección única. Este tipo de direccionamiento se emplea en redes de difusión y en redes multipunto ya que existen varias estaciones destino en la misma red.

Asimismo, en función de cómo están conectadas las estaciones al canal los métodos de direccionamiento pueden clasificarse en:

- *Preselección.* Existe un dispositivo central que se encarga de seleccionar secuencialmente cada una de las estaciones de destino accesibles desde una estación.
- *Master único.* En este tipo de direccionamiento existe una estación central denominada master que se encarga de gestionar las comunicaciones. Sólo el equipo master tiene iniciativa de enviar datos. El resto de estaciones se le denominan esclavos y se encargan únicamente de responder a las peticiones del master.
- *Master múltiple.* Este caso es similar al anterior pero pueden existir varios equipos master en la red.

Además del direccionamiento individual de cada máquina suelen emplearse dos tipos de direcciones especiales:

- *Direcciones de broadcast.* Una trama enviada a la dirección de broadcast será destinada a todos los equipos de la red.
- *Direcciones de multicast.* Una trama enviada a la dirección de multicast será destinada a un subconjunto de los equipos que componen la red.

5.6 Detección y corrección de errores en tramas

Cualquier medio de transmisión se ve sometido a ruidos e interferencias externas que pueden causar que, al reducirse la relación señal-ruido, se produzcan errores en la transmisión. En general, se dice que se ha producido un error cuando el valor de un bit enviado por el receptor es alterado durante la transmisión. Básicamente, es posible distinguir dos tipos de errores:

- *Errores aislados.* Cuando, por ejemplo, el emisor envía una trama y un bit de la trama que se correspondía a un '1' binario es recibido por el receptor como un '0' (error aislado en un bit sin llegar a afectar a los bits vecinos).
- *Errores a ráfagas.* Se dice que se ha producido un error en una ráfaga de longitud B cuando se recibe una secuencia de B bits en la que el primero, el último y cualquier número de bits intermedios son erróneos.

Los protocolos de nivel de enlace deben ser capaces de asegurar la transmisión sin errores producidos por ruido o atenuaciones del medio físico. Los errores aislados son más fáciles de detectar y corregir, sin embargo, las ráfagas son más difíciles de tratar y también más frecuentes. Así, en cuanto a tratamiento de errores realizado por los protocolos de nivel de enlace se distingue dos posibilidades:

- *Detección de errores.* Consiste en únicamente detectar si los bits recibidos son los que realmente quiso enviar el emisor. Se implementa mediante técnicas de reenvío de tramas, de forma que si el receptor determina que se ha producido algún error, se encarga de notificar esta situación al emisor y este último deberá reenviar la trama de datos que había llegado de

forma incorrecta al receptor. Este tipo de aproximaciones se emplean cuando el tiempo de reenvío es inferior al tiempo de chequeo y corrección.

- *Corrección de errores.* Consiste en, no solo detectar si se ha producido algún error o no, sino también determinar cuales son los bits de la trama que han llegado alterados al receptor. Una vez detectados, el propio receptor modifica los datos erróneos de forma que no es necesario que el emisor realice una retransmisión de la trama original. Se emplea detección y corrección de errores simultáneamente en los medios físicos donde el reenvío produce retardos elevados.

Por lo tanto, el proceso de control de errores se lleva a cabo de forma combinada tanto en el emisor como en el receptor. Así, el emisor añade en la trama información para el control de errores. Normalmente, este campo se denomina FCS. Esta información llegará al receptor el cual se encarga de interpretar la información recibida con el fin de detectar errores y corregirlos si fuera posible.

Se van a describir dos métodos para la detección de errores: *comprobación de paridad* y *comprobación de redundancia cíclica*. Por último, para la corrección de errores se describirá la utilización de *códigos Hamming*.

5.6.1 Comprobación de paridad

Este método constituye la forma más sencilla de detectar errores y básicamente consiste en añadir un único bit al final de la trama de datos. El valor de este bit se determina de tal forma que el bloque de bits resultante tenga un número par de unos (paridad par) o un número impar de unos (paridad impar).

Supóngase que el emisor debe realizar la transmisión de la siguiente trama de datos: 1011. Si se emplea paridad par, el bloque de datos que se enviaría finalmente será: 10111. Se observa que en este caso se añade un bit '1' para obtener un número par de unos en el bloque final de bits obtenido. Sin embargo, si se empleara paridad impar el bloque final sería: 10110. Considérese ahora que se envía el bloque de datos anterior empleando paridad par. En este caso le llegará al receptor el bloque de bits 10111 (si no se ha producido ningún error durante la transmisión). El receptor volverá a calcular la paridad y si el bit de paridad calculado teóricamente coincide con el bit de paridad introducido por el emisor significará que no se ha producido ningún error.

Este tipo de detección de errores presenta la ventaja de que no introduce excesiva información de control adicional para la detección y es muy sencillo de implementar. Sin embargo, este método no detectará el error si dos o cualquier número par de bits se alteran debido a algún error en la transmisión.

5.6.2 Comprobación de redundancia cíclica (CRC)

Los códigos de redundancia cíclica son una secuencia de bits que se colocan en el campo control de errores de la trama (FCS) y constituyen una de las formas más habituales y potente de detección de errores. Se basa en la división binaria e inclusión de un resto en el campo control de errores de la trama de datos. En la Figura 5-6 se muestra la aritmética básica para la aplicación del método de generación y comprobación de errores mediante CRC. La trama enviada realmente es la diferencia entre los datos, D, y el resto, R, obtenido al dividir D entre un divisor conocido G. Ese divisor es conocido tanto por el emisor como el receptor. Cuando esta trama llegue al receptor dividirá la trama entre ese divisor conocido y si no hay resto en la división supondrá que no ha habido errores.

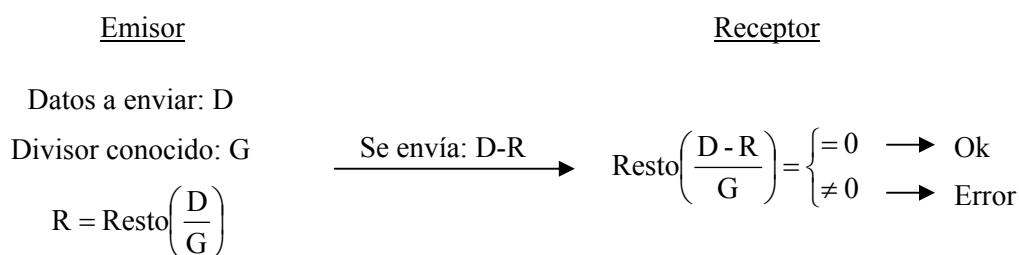


Figura 5-6. Aritmética para el cálculo y comprobación de CRC.

Sin embargo, a diferencia del esquema representado en la Figura 5-6 los datos realmente transmitidos no son números sino un conjunto de bits que pueden representarse en forma de polinomios de manera que los bits constituyan los coeficientes del polinomio. Por ejemplo, la secuencia de bits 10010110 puede representarse con el polinomio $1x^7+0x^6+0x^5+1x^4+0x^3+1x^2+1x^1+0x^0 = x^7+x^4+x^2+x$.

Considerando los datos representados en forma de polinomio y que el divisor, G, en este caso se denomina polinomio generador, G(x) de grado r, el esquema de la Figura 5-6 sería el que se muestra en la Figura 5-7.

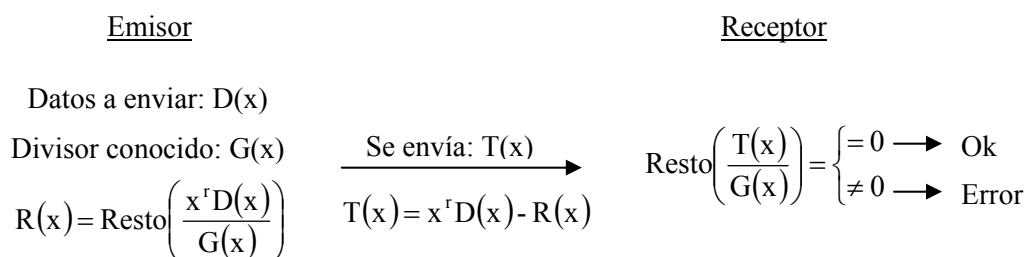


Figura 5-7. Proceso de cálculo y comprobación de CRC considerando los datos representados en forma de polinomio.

Se observa como los datos a enviar D(x) son multiplicados por x^r . Esto equivale a poner r ceros tras el valor binario de D(x). Por ejemplo, considerando los datos a enviar 10010110, entonces $D(x) = x^7+x^4+x^2+x$. Si $r=2$, $x^r D(x) = x^9+x^6+x^4+x^3$. Así, a la hora de calcular T(x) no se mezcla D(x) y R(x), y los datos y el CRC irán por separado (véase Figura 5-8).

$$\begin{array}{rcl}
 & & \overbrace{\hspace{2cm}}^r \\
 x^r D(x): & 10010110 & 00 \dots 0 \\
 R(x): & & 10 \dots 1 \\
 \hline
 T(x): & 10010110 & 10 \dots 1 \\
 & \text{Datos} & \text{CRC} = R(x)
 \end{array}
 \quad - \text{ Resta módulo 2}$$

Figura 5-8. Combinación de datos, D(x) y CRC, R(x).

El proceso seguido para la generación del CRC y la posterior comprobación de errores en el receptor se puede sintetizar en los siguientes pasos:

- En el emisor:
 - Para crear el resto CRC es necesario disponer de un divisor CRC, o polinomio generador $G(x)$.
 - A los datos, $D(x)$, que se quieren transmitir se le añaden r ceros por la derecha, siendo $r+1$ la longitud del divisor CRC.
 - Se divide $x^r D(x)$ entre el divisor, $G(x)$ para determinar el resto, $R(x)$, donde r es el grado del polinomio generador.
 - El resto de la división $R(x)$ es el CRC.
 - Se calcula $T(x)$. Para ello se calcula la siguiente diferencia: $x^r D(x) - R(x)$.
- En el receptor:
 - Se divide $T(x)$ entre el polinomio generador, $G(x)$, y se calcula el nuevo resto, $R'(x)$.
 - Si resto, $R'(x)$, es cero la trama no presenta errores. Si el resto, $R'(x)$, no es cero será porque la trama presenta errores.

Se observa que es necesario que tanto el emisor como el receptor conozcan el valor del polinomio generador, $G(x)$. Existen algunos valores estándar para $G(x)$ como son los siguientes:

- CRC-12. Se utiliza para la transmisión de secuencias de caracteres de 6 bits y generará un CRC de 12. El valor de $G(x)$ en este caso es $x^{12} + x^{11} + x^3 + x^2 + x + 1$.
- CRC-16 y CRC-CCITT. Ambos son habituales para los caracteres de 8 bits son empleados en Estados Unidos y en Europa respectivamente y sus valores son $x^{16} + x^{15} + x^2 + 1$ y $x^{16} + x^{12} + x^5 + 1$.
- CRC-32. Es empleado en algunas normas de transmisión síncrona sobre enlaces punto a punto y su valor es $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$.

En general, empleando el método de comprobación de código de redundancia es posible detectar los siguientes errores:

- Se detectan todos los errores de un bit si el polinomio generador $G(x)$ tiene más de un término distinto de cero.
- Se detectan todos los errores dobles, siempre que $G(x)$ tenga al menos un factor con tres términos.
- Se detecta cualquier número impar de errores, siempre que $G(x)$ contenga el factor $x+1$.
- Se detecta cualquier ráfaga de errores con longitud menor o igual a la longitud del CRC.

Ejercicio 5.1

Calcular la trama de datos a enviar $T(x)$ si el emisor quiere enviar la cadena de datos '10100111' usando como polinomio generador $G(x) = x^2 + x + 1$. A continuación, comprobar si se han producido errores en los datos recibidos por el receptor.

Paso 1. Calcular $x^r D(x)$ para ello habrá que añadir tantos ceros a la derecha de los datos a enviar como grado r tiene el polinomio generador $G(x)$ ($r=2$).

$$G(x) = x^2 + x + 1 \rightarrow '111'. (r=2)$$

$$D(x) = x^7 + x^5 + x^2 + x^1 + 1 \rightarrow '10100111'$$

$$x^r D(x) = x^9 + x^7 + x^4 + x^3 + x^2 \rightarrow '1010011100'$$

Paso 2. En este paso hay que calcular el resto $R(x)$. Para ello, se divide $x^r D(x)$ entre $G(x)$ usando división módulo 2. En este caso el resto obtenido es '11'.

[illegible]

Paso 3. Calcular T(x). Para ello, a la cadena resultante del paso 1 se le subtrae el resto R(x) obtenido de calcular la división del paso 2. Como se mostró en la Figura 5-8 esto equivale a combinar los datos y el resto, $R(x) = CRC$. Por lo tanto, en este caso $T(x) = x^9 + x^7 + x^4 + x^3 + x^2 + x + 1 \rightarrow '101001110011'$.

Paso 4. En este caso se va a describir la comprobación que realiza el receptor con el objetivo de determinar si la trama presenta o no errores a partir de los datos y CRC recibido. Para realizar dicha comprobación se divide $T(x)$ entre $G(x)$ y se observa el valor del resto. En este caso el resto es nulo, por lo que no se habrá producido ningún error al enviar los datos.

[illegible]

5.6.3 Corrección de errores empleando códigos de bloque

La mayoría de los protocolos de nivel de enlace implementan alguna técnica de detección de errores como la mostrada en el apartado anterior basada en códigos de redundancia cíclica. Sin embargo, existen aplicaciones en las cuales puede ser más conveniente realizar una corrección de los datos recibidos que solicitar al emisor que vuelva a enviar las tramas erróneas. Supóngase, por ejemplo, la transmisión de información entre satélites. En este caso, el tiempo que tarda una trama en propagarse desde el emisor hasta el receptor puede ser muy elevado. Además, debido a las características del medio de transmisión, la tasa de errores también puede ser bastante elevada, lo que provocará un gran número de retransmisiones. Una alternativa sería que el receptor haga uso del conjunto de bits recibidos, no solo para detectar la existencia o no de errores, sino también para corregirlos. En este apartado se va a describir una aproximación basada en códigos de bloque que permitirá al receptor detectar la existencia de errores y poder reconstruir la trama que originalmente había enviado el emisor.

En las técnicas de corrección de errores basadas en códigos de bloque no se emplea un campo FCS (como se hacía en la detección basada en CRC) sino que la información adicional necesaria para la detección y corrección se intercala entre los bits de datos que el emisor desea hacer llegar al receptor. Cuando la trama de datos llega al receptor, éste se encarga de separar los bits de datos de la información adicional intercalada para enviar sólo al nivel de red la parte de datos. Para realizar la comprobación y corrección de errores empleando códigos de bloque es necesario definir los siguientes conceptos previos:

- *Palabras de datos.* A partir de un conjunto de k bits es posible obtener un total de 2^k palabras de datos. Estos van a ser los bloques de datos que en principio desea enviar el emisor.
- *Palabras-código.* Como se ha indicado anteriormente, para la corrección de errores se intercalará un conjunto de bits (un total de r bits redundantes) a las palabras de datos. Por lo tanto, se enviarán finalmente $n=k+r$ bits y a partir de estos bits es posible generar un total de 2^n palabras código. Las palabras-códigos serán los datos que finalmente se enviarán al medio.
- *Distancia Hamming entre dos palabras código.* La distancia Hamming entre dos palabras-código se define como el número de bits en que difieren las dos palabras. Por ejemplo, la distancia de Hamming $d(\mathbf{v}_1, \mathbf{v}_2)$ entre las palabras $\mathbf{v}_1=011101$, $\mathbf{v}_2=010011$ será de 3.
- *Distancia Hamming de un código.* Dado un conjunto de palabras código correctas, se define la distancia Hamming de un código como la menor de todas las distancias Hamming entre las palabras-código.

Cuando se emplea corrección por códigos de bloque, en lugar de transmitir cada bloque de k bits (palabras de datos), se asigna cada secuencia de entrada a una única palabra-código de n bits. En la Figura 5-9 se muestran los pasos seguidos en la corrección, que se pueden resumir en los siguientes:

- En el emisor, a través de un proceso de codificación, se asigna una palabra-código a la palabra de datos que se desea enviar. Por lo tanto, cada palabra de datos de k bits tendrá asociada una palabra-código de n bits.
- Cuando el receptor recibe una palabra-código sin errores, es decir, idéntica a la palabra-código original, el decodificador generará el bloque de datos correspondiente, una vez eliminados los bits añadidos.
- Para ciertos patrones de error, es posible que el decodificador detecte y corrija esos errores. Así, aunque el bloque de datos recibido difiera de la palabra-código transmitida por el emisor, el decodificador será capaz de extraer la palabra de datos original.
- Para otros patrones de error el decodificador no será capaz de realizar la corrección. En este caso, el receptor notifica esta situación al nivel de red.

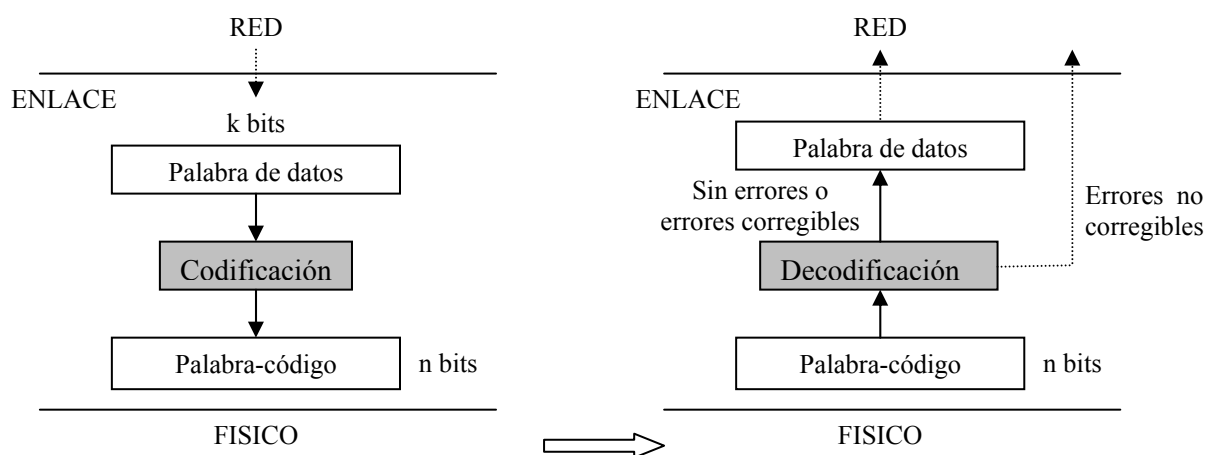


Figura 5-9. Corrección de errores empleando palabras código.

Por ejemplo, para $k=2$ y $n=5$ se pueden realizar las siguientes asignaciones:

Bloque de datos	Palabra-código
00	00000
01	00111
10	11001
11	11110

Si el receptor recibe el patrón de bits 00100, ésta no será una palabra-código válida, por lo que el receptor detecta un error. A partir de la palabra-código errónea recibida no es posible asegurar qué bloque de datos fue originalmente enviado ya que el ruido puede haber alterado 1, 2 o más bits. Sin embargo, para convertir la palabra-código 00100 en 00000 (palabra-código válida) solo se necesita alterar un bit. Además, para convertirla en cualquier otra palabra se necesitará alterar más bits. Por lo tanto, puede suponerse que el bloque de datos originalmente enviado es 00. De esta manera, si se recibe una palabra código inválida, entonces se selecciona la palabra código válida a menor distancia Hamming. Por lo tanto, tanto el emisor como el receptor deben conocer todas las palabras código y su correspondencia con palabras de datos. Estas palabras-código dependerán del tipo de paridad que se escoja.

Supóngase ahora que se recibe la palabra 01010. Esta palabra-código tiene como menor distancia a cualquier palabra-código 2. Además, se encuentra a la misma distancia de 00000 y 11110 (distancia = 2 como se ha indicado anteriormente). En este caso el receptor no tendría forma de elegir entre las dos alternativas y no se podría corregir el error. Así, este código puede corregir errores simples pero no errores dobles. Dicho de otra forma, la distancia del código es 3. Por tanto, un error en un bit dará lugar a una palabra-código inválida que está a distancia 1 de la palabra-código válida, pero al menos a distancia 2 de cualquiera de las otras palabras código.

A partir de las definiciones anteriores se puede afirmar que, dado un código binario de distancia Hamming d , es posible detectar errores de $d-1$ bits y corregir errores de $(d-1)/2$ bits.

Un caso específico de detección y corrección de errores es el Algoritmo de Hamming. El algoritmo de Hamming permite realizar la corrección de errores en un bit y para ello deberá cumplirse

que $k+r+1 \leq 2^r$. Esta relación permite obtener cuantos bits redundantes, r , es necesario introducir para una palabra de datos con un total de k bits. A continuación se muestran los pasos que se han de seguir para la generación de palabras código que permitan realizar la posterior corrección de errores en 1 bit:

1. Determinar la cantidad de bits redundantes, r , que son necesarios. Para ello se hace uso de la relación $k+r+1 \leq 2^r$.
2. Numerar los bits de la palabra-código de izquierda a derecha, en orden ascendente, desde $i=1 \dots n$.
3. Todos los bits cuya posición es potencia de dos se utilizan como bits redundantes (posiciones 1, 2, 4, 8, 16, 32, 64, etc.).
4. Los bits del resto de posiciones son utilizados como bits de datos (posiciones 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, etc.).
5. El valor de los bits redundantes se va a determinar como bits de paridad. Cada bit de paridad se obtiene calculando la paridad de alguno de los bits de datos. La posición del bit de paridad determina la secuencia de los bits que alternativamente comprueba y salta, tal y como se explica a continuación:
 - Posición 1: salta 1, comprueba 1, salta 1, comprueba 1, etc.
 - Posición 2: comprueba 1, salta 2, comprueba 2, salta 2, comprueba 2, etc.
 - Posición 4: comprueba 3, salta 4, comprueba 4, salta 4, comprueba 4, etc.
 - Posición 8: comprueba 7, salta 8, comprueba 8, salta 8, comprueba 8, etc.
 - Posición 16: comprueba 15, salta 16, comprueba 16, salta 16, comprueba 16, etc.
 - Y así sucesivamente.

En otras palabras, el bit de paridad de la posición 2^k comprueba los bits en las posiciones que tengan al bit k en su representación binaria.

Ejercicio 5.2

Dada la palabra de datos de 7 bits "0110110" determinar la palabra-código asociada teniendo en cuenta que se emplea paridad par.

Paso 1. Determinar el conjunto de bits necesarios. Para ello, hay que aplicar la relación $k+r+1 \leq 2^r$. En este caso $k=7$, por lo que $r+8 \leq 2^r$. Si $r=1 \rightarrow 9 > 2^1$, Si $r=2 \rightarrow 10 > 2^2$, Si $r=3 \rightarrow 11 > 2^3$, Si $r=4 \rightarrow 12 \leq 2^4$ (para $r=4$ se cumple la relación $k+r+1 \leq 2^r$). Por lo tanto, es necesario introducir 4 bits redundantes.

Pasos 2, 3 y 4. En la Figura 5-10 se muestra una tabla en la que se representa la palabra de datos original y se ha identificado la posición de los bits de datos, d_i , y la de los bits redundantes, r_i .

Paso 5. Se determinan los valores de los bits redundantes como bits de paridad. En la tabla de la Figura 5-10 se ha introducido una fila para cada bit redundante en la que se observa, no solo el valor de la paridad, sino también qué bits se utiliza para su cálculo. En este caso todos los bits redundantes son nulos.

Por lo tanto, la nueva palabra código será 00001100110. En la tabla de la Figura 5-11 se describe el proceso de detección de errores cuando la trama llega al bit sin alteración en ninguno de sus bits y en la Figura 5-12 la palabra código que llega al receptor tiene el último bit alterado.

	r_1	r_2	d_1	r_3	d_2	d_3	d_4	r_4	d_5	d_6	d_7
Palabra de datos:			0		1	1	0		1	1	0
r_1	0		0		1		0		1		0
r_2		0	0			1	0			1	0
r_3				0	1	1	0				
r_4								0	1	1	0
Palabra-código:	0	0	0	0	1	1	0	0	1	1	0

Figura 5-10. Ejemplo de cálculo de palabra-código.

	r_1	r_2	d_1	r_3	d_2	d_3	d_4	r_4	d_5	d_6	d_7	Prueba de paridad	Bit de paridad
Palabra de recibida:	0	0	0	0	1	1	0	0	1	1	0	0	
r_1	0		0		1		0		1		0	Ok	0
r_2		0	0			1	0			1	0	Ok	0
r_3				0	1	1	0					Ok	0
r_4								0	1	1	0	Ok	0

Figura 5-11. Comprobación de errores. La palabra código llega sin errores.

	r_1	r_2	d_1	r_3	d_2	d_3	d_4	r_4	d_5	d_6	d_7	Prueba de paridad	Bit de paridad
Palabra de recibida:	0	0	0	0	1	1	0	0	1	1	1	0	
r_1	0		0		1		0		1		1	Error	1
r_2		0	0			1	0			1	1	Error	1
r_3				0	1	1	0					Ok	0
r_4								0	1	1	1	Error	1

Figura 5-12. Comprobación de errores. La palabra código llega con errores.

Hasta ahora se ha descrito como generar las palabras código. Sin embargo, también es necesario definir la manera en que el receptor, una vez detectado el error, es capaz de determinar cuál es el bit que ha llegado alterado. En el caso de la Figura 5-12 ese bit es el undécimo (el número once empezando a contar los bits de izquierda a derecha). Para ello, se hace uso de un contador c que se inicializa a 0. Si se analiza la tabla de la Figura 5-12 la paridad que se debe obtener tras la llegada del mensaje sin errores debe ser siempre 0 (por cada fila), pero en el momento en que ocurre un error esta paridad cambia a 1. Se observa que en la fila en que el cambio no afectó la paridad es cero y llega sin errores. Por lo tanto, si un determinado bit redundante tiene la paridad errónea (se comprueba que

debería ser 1), se incrementa la variable c en el valor de la posición del bit redundante (nótese que los bits se había numerado de izquierda a derecha comenzando por 1). Si, una vez verificados todos los bits redundantes $c=0$, entonces no habrá error. Si $c \neq 0$, el valor de c indicará cual es la posición del bit redundante.

Para el caso mostrado en la Figura 5-12 los bits redundantes con paridad errónea eran r_1 , r_2 y r_4 que se corresponden con las posiciones 1, 2 y 8. Por lo tanto $c=1+2+8=11$. Por lo tanto, como se había indicado en un principio, el bit erróneo es el undécimo.

5.7 Medición de la tasa de errores

En el capítulo 4 se describieron los medios de transmisión que se emplean habitualmente en redes de computadores. Independientemente de las características de los medios de transmisión, éstos sufren una determinada atenuación dependiendo de la frecuencia de transmisión y de la longitud del medio. Además, en cualquier medio existirá un determinado ruido que puede provocar errores que harán que uno o varios bits se vean alterados durante el proceso de transmisión.

Esa tasa de error, que a menudo también se denomina como BER (Bit Error Rate) define la probabilidad de que un bit recibido sea erróneo y depende de diversos factores. Dentro de estos factores cabe mencionar algunos como el tipo del medio físico empleado, el entorno del medio, la velocidad de transmisión, la calidad del servicio del medio físico o el horario en que se realiza la comunicación. El BER puede calcularse como la relación entre el número de errores y el número de bits totales transmitidos; y presenta valores típicos comprendidos entre 10^{-3} y 10^{-6} .

Asociado a los errores que puede dar lugar la transmisión de tramas a nivel de enlace se pueden definir las siguientes probabilidades:

- P_b : Probabilidad de que un bit recibido sea erróneo (BER).
- P_1 : Probabilidad de que una trama llegue sin errores.
- P_2 : Probabilidad de que, utilizando un algoritmo para la detección de errores, una trama llegue con uno o más errores no detectados.
- P_3 : Probabilidad de que, utilizando un algoritmo para la detección de errores, una trama llegue con uno o más errores detectados y sin errores indetectados.

Supóngase que P_3 es cero, es decir, no se aplica ningún método para la detección de errores. Además, todos los bits transmitidos (un total de N bits) tengan una probabilidad de error igual a P_b . En este caso las probabilidades P_1 y P_2 serán respectivamente:

$$P_1 = (1-P_b)^N \quad (5.1)$$

$$P_2 = 1-P_1 \quad (5.2)$$

Por lo tanto, se puede concluir que:

- La probabilidad de que una trama llegue sin ningún bit erróneo disminuye al aumentar la probabilidad de que un bit sea erróneo.
- La probabilidad de que una trama llegue sin errores disminuye al aumentar la longitud de la misma.

Ejercicio 5.3

Calcular la probabilidad de que una trama de 4 bits llegue al receptor sin errores si se sabe que la probabilidad de que se produzca un error en un bit es de 0,4.

$$P_b=0,4$$

$$P_1 = (1-0,4)^N = (1-0,4)^4 = 0,129$$