El objetivo de este capítulo es ofrecer una visión general y servir de base al material que se detalla en el resto de la Parte IV. También servirá para mostrar cómo encajan los conceptos de las Partes I, II y III en las redes de computadores de área más amplia y en las comunicaciones entre computadores.

Comenzamos con una exposición del concepto de protocolo de comunicaciones, mostrando que son elementos fundamentales para todas las comunicaciones de datos. A continuación veremos una forma de describir e implementar sistemáticamente la función de comunicaciones viendo las tareas de comunicaciones en términos de un conjunto de capas superpuestas, incluyendo protocolos en cada una de ellas. Éste es el punto de vista del ahora famoso modelo de interconexión de sistemas abiertos (OSI, "Open Systems Interconection").

Aunque el modelo OSI es casi universalmente aceptado como el marco de trabajo en este área, existe otro modelo, conocido como arquitectura del protocolo TCP/IP, que ha conseguido el predominio comercial a expensas de OSI. La mayor parte de los protocolos específicos descritos en la Parte IV forman parte del conjunto de protocolos de TCP/IP. En este capítulo ofrecemos una visión general de ellos.

15.1 PROTOCOLOS

Comenzamos con una visión general sobre las características de los protocolos. Antes de continuar, el lector debería revisar los conceptos de OSI y TCP/IP presentados en el Capítulo 1. Tras esta visión general estudiaremos con mayor profundidad OSI y TCP/IP.

Características

Los conceptos de procesamiento distribuido y redes de computadores implican la necesidad de comunicación entre entidades en diferentes sistemas, y donde los términos "entidad" y "sistema" se usan en un sentido muy general. Ejemplos de entidades son programas de aplicaciones de usuario, paquetes de transferencia de ficheros, sistemas de gestión de base de datos, facilidades de correo electrónico y terminales. Ejemplos de sistemas son computadores, terminales y sensores remotos. Observemos que, en algunos casos, la entidad y el sistema en que ésta reside coexisten (por ejemplo, terminales). En general, una entidad es cualquier cosa con capacidad de enviar o recibir información, y un sistema es un objeto físicamente diferenciado que contiene una o más entidades.

Para que dos entidades se comuniquen con éxito deben "hablar el mismo lenguaje". Qué comunican, cómo se comunican y cuándo lo hacen deben constituir un conjunto de reglas aceptadas mutuamente entre las entidades involucradas. El conjunto de convenciones se denomina protocolo, que puede definirse como un conjunto de normas que gestionan el intercambio de datos entre dos entidades. Los elementos clave de un protocolo son:

- *Sintaxis*: comprende cuestiones tales como formato de datos, codificación y niveles de señal.
- *Semántica*: comprende información de control para la coordinación y la gestión de errores.
- *Temporización*: comprende la coordinación en la velocidad y el orden secuencial.

HDLC es un ejemplo de un protocolo. Los datos a intercambiar deben ser enviados en tramas con un formato específico (sintaxis). El campo de control proporciona una gran variedad de funciones de regulación, tales como especificación de modo y establecimiento de conexión (semántica). También se incluyen métodos para control de flujo (temporización). La mayor parte de la Parte IV de este texto se dedicará a discutir otros ejemplos de protocolos.

Algunas características importantes de un protocolo son:

- Directo/indirecto.
- Monolítico/estructurado.
- Simétrico/asimétrico.
- Normalizado/no normalizado.

La comunicación entre dos entidades puede ser directa o indirecta. En la Figura 15.1 se muestran diferentes situaciones posibles. Si dos sistemas comparten un enlace punto a punto, las entidades en estos sistemas pueden comunicarse directamente; es decir, se pasan directamente datos e informa-

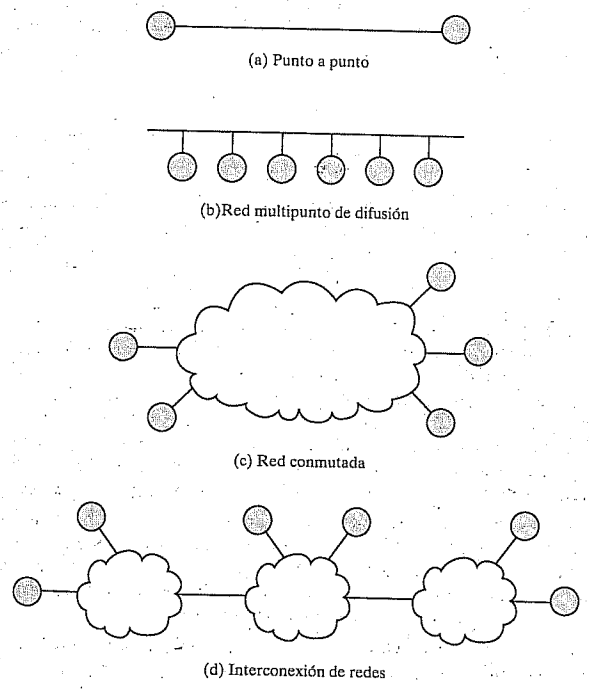


FIGURA 15.1 Medios de conexión de sistemas de comunicación.

ción de control entre ellas sin intervención de un agente activo. Lo mismo se puede decir de una configuración multipunto, aunque en este caso las entidades deben estar relacionadas con la tarea de control de acceso, provocando que el protocolo sea más complejo. Si los sistemas se conectan a través de una red de comunicaciones conmutada, no es posible el uso de un protocolo directo. Las dos entidades deben depender del funcionamiento de otras entidades para intercambiar datos. Un caso más extremo es una situación en la que dos entidades no comparten ni siquiera la misma red conmutada, sino que se encuentran conectadas indirectamente a través de dos o más redes. Un conjunto de redes interconectadas de esta forma se denomina internet o interconexión de redes.

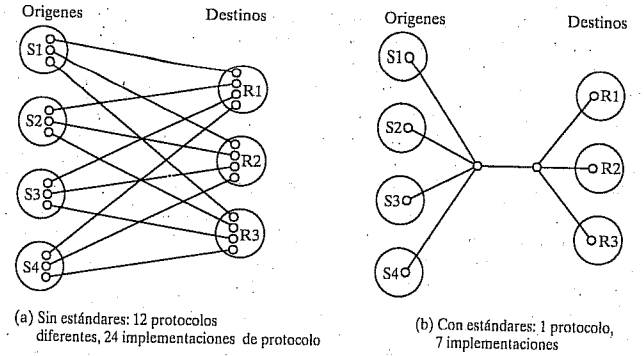
Otra característica de un protocolo es si éste es monolítico o estructurado. Tal como se establece en la Parte III, es evidente que la tarea de comunicación entre entidades en sistemas diferentes es demasiado compleja para gestionarla como una sola unidad. Por ejemplo, consideremos un paquete de correo electrónico operando en dos computadores conectados mediante un enlace HDLC síncrono. Para que sea verdaderamente monolítico, el paquete necesitaría incluir toda la lógica HDLC. Si la conexión se estableciese a través de una red conmutada, el paquete seguiría necesitando la lógica HDLC (o algo equivalente) para conectarse con la red. También necesitaría una lógica para dividir el mensaje en trozos del tamaño de paquetes, una lógica para solicitar un circuito virtual, y así sucesivamente. El correo sólo podría enviarse cuando el sistema de destino y la entidad se encuentren activos y preparados para recibir. Es necesario el uso de una lógica para esta coordinación. Y, como veremos, la lista continúa. Un cambio en cualquier aspecto implica la modificación del paquete, con el riesgo de introducir problemas difíciles de determinar.

Una alternativa consiste en hacer uso de técnicas de diseño y de implementación estructurados. En lugar de un único protocolo, existe un conjunto de ellos que presentan una estructura jerárquica o en capas. Se implementan funciones primitivas en entidades de nivel inferior que ofrecen servicios a entidades de nivel superior. Por ejemplo, podría existir un módulo HDLC (entidad) usado por una facilidad de correo electrónico cuando sea necesario. Obsérvese que ésta es otra forma de conexión indirecta: entidades de nivel superior se apoyan en entidades de nivel inferior para intercambiar datos.

Cuando se usa un diseño de protocolo estructurado, denominamos arquitectura de comunicaciones a los soportes físico y lógico empleados en la implementación de las funciones de comunicación. El resto del presente capítulo, después de esta sección, se dedica a este concepto.

Un protocolo puede ser simétrico o asimétrico. La mayor parte de los protocolos que estudiaremos son simétricos; es decir, la comunicación se realiza entre entidades paritarias. La asimetría puede deberse a la lógica de un intercambio (por ejemplo, un proceso "cliente" y otro "servidor"), o por el deseo de hacer una de las entidades o sistemas tan sencillo como sea posible. Un ejemplo de esto último es el modo de respuesta normal de HDLC. Generalmente, esto implica que un computador sondea y selecciona varios terminales. La lógica en el terminal final es sencilla.

Por último, un protocolo puede ser normalizado o no. Un protocolo no normalizado es un protocolo diseñado para una situación de comunicaciones específica o, a lo sumo, para un modelo particular de un computador. Así, si K tipos diferentes de fuentes de información deben comunicarse con L tipos de receptores de información, se requieren $K \times L$ protocolos no normalizados distintos y se necesita un total de $2 \times K \times L$ implementaciones (Figura 15.2a). Si todos los sistemas comparten un protocolo común, se necesitarían $K + L$ implementaciones (Figura 15.2b). El uso creciente de procesamiento distribuido y la tendencia decreciente de la dependencia de los consumidores a un solo proveedor hacen que todos estos últimos implementen protocolos que siguen una normalización.



(a) Sin estándares: 12 protocolos diferentes, 24 implementaciones de protocolo

(b) Con estándares: 1 protocolo, 7 implementaciones

FIGURA 15.2 Uso de protocolos normalizados.

Funciones

Antes de entrar a discutir la arquitectura de comunicaciones y los distintos niveles de protocolos, consideremos un conjunto bastante más reducido de funciones que forman la base de todos los protocolos. No todos los protocolos realizan todas las funciones, pues implicaría una importante duplicación de esfuerzo. Existen, no obstante, varios ejemplos del mismo tipo de funciones presentes en protocolos de niveles diferentes.

Esta discusión será, por necesidad, bastante abstracta, y proporciona una panorámica integrada de las características y funciones de los protocolos de comunicaciones. El concepto de protocolo es fundamental para el resto de la Parte IV de este texto, y, como hemos hecho hasta el momento, se verán ejemplos específicos de todas las funciones.

Las funciones de los protocolos se pueden agrupar en las siguientes categorías:

- Segmentación y ensamblado
- Encapsulado
- Control de conexión
- Envío ordenado
- Control de flujo
- Control de errores
- Direccionamiento
- Multiplexación
- Servicios de transmisión

Segmentación y ensamblado[1]

Un protocolo está relacionado con el intercambio de secuencias de datos entre dos entidades. Usualmente, la transferencia se puede caracterizar mediante una secuencia de bloques de datos

[1] Aunque el significado es el mismo, en la mayor parte de las especificaciones de protocolos en TCP/IP se usa el término fragmentación en lugar del de segmentación.

del mismo tamaño. En el nivel de aplicación, denominaremos mensaje a una unidad lógica de transferencia de datos. Ahora bien, si la entidad de aplicación envía datos en mensajes o en una secuencia continua, los protocolos de nivel inferior pueden necesitar dividir los datos en bloques de menor tamaño y todos del mismo. Este proceso se llama segmentación. Por conveniencia, denominaremos unidad de datos de protocolo (PDU, "Protocol Data Unit") a un bloque de datos intercambiado entre dos entidades a través de un protocolo.

Dependiendo del contexto, existen varias razones para hacer uso de la segmentación. Entre las razones usuales se encuentran:

- La red de comunicaciones sólo puede aceptar bloques de datos de tamaño máximo. Por ejemplo, una red ATM está limitada a bloques de 53 octetos y Ethernet impone un tamaño máximo de 1.526 octetos.
- El control de errores puede resultar más eficiente con tamaños menores de PDU. Por ejemplo, el uso de bloques más pequeños requiere la retransmisión de pocos bits con la técnica de repetición selectiva.
- Se puede proporcionar un acceso más equitativo a las facilidades de transmisión compartidas con un retardo más reducido. Por ejemplo, sin un tamaño máximo de bloque una estación podría monopolizar un medio multipunto.
- Un tamaño menor de PDU podría significar que las entidades receptoras necesiten reservar memorias temporales de menor capacidad.
- Una entidad puede necesitar que la transferencia de datos entre de vez en cuando en algún tipo de "cierre" para operaciones de comprobación y de reinicio/recuperación.

La segmentación presenta varias desventajas que proporcionan argumentos para la creación de bloques tan grandes como sea posible:

- Como veremos, cada PDU contiene una cantidad mínima fija de información de control. Así, cuanto menor sea el bloque, mayor será el porcentaje global de bits suplementarios.
- La recepción de una PDU puede generar una interrupción que debe ser atendida, con lo que el uso de bloques pequeños da lugar a un número mayor de interrupciones.
- Se consume más tiempo en procesar PDU más pequeñas y numerosas.

Todos estos factores deben ser tenidos en consideración por el diseñador de protocolos para determinar el tamaño máximo y mínimo de las PDU.

Lo contrario a segmentar es ensamblar. Eventualmente, los datos segmentados deben agruparse en mensajes apropiados para el nivel de aplicación. La tarea se complica si se reciben las PDU fuera de secuencia.

El proceso de segmentación se ilustró en la Figura 1.7.

Encapsulado

Cada PDU consta no sólo de datos, sino también de información de control. En cambio, algunas PDU contienen sólo información de control, sin datos. La información de control se clasifica en tres categorías:

- *Dirección*: se puede indicar la dirección del emisor y/o la del receptor.
- *Código de detección de errores*: a veces se incluye algún tipo de secuencia de comprobación de trama para detección de errores.

- *Control de protocolo*: se incluye información adicional para implementar las funciones de protocolo enumeradas en el resto de la presente sección.

La incorporación de información de control a los datos se denomina encapsulado. Una entidad captura o genera datos y los encapsula en una PDU que contiene datos además de información de control (ver Figuras 1.7 y 1.8). Un ejemplo es la trama HDLC (Figura 6.10).

Control de conexión

Una entidad puede transmitir datos a otra entidad de forma que cada PDU se trate independientemente de las PDU anteriores. Esto se conoce como transferencia de datos no orientada a conexión; un ejemplo es el uso de datagramas. Aunque este modo es útil, una técnica igualmente importante es la transferencia de datos orientada a conexión, de la que el circuito virtual es un ejemplo.

Si las estaciones prevén un intercambio largo de datos y/o algunos detalles de su protocolo cambian dinámicamente, es preferible (incluso necesaria) la transferencia de datos orientada a conexión. Se establece una asociación lógica, o conexión, entre entidades. Tiene lugar en tres fases (Figura 15.3):

- Establecimiento de conexión
- Transferencia de datos
- Liberación de conexión

Haciendo uso de protocolos más sofisticados, pueden existir también fases de interrupción de conexión y de recuperación para gestionar la aparición de errores y otros tipos de interrupciones.

Durante la fase de establecimiento de conexión, dos entidades se ponen de acuerdo para intercambiar datos. Generalmente, una estación enviará una petición de conexión (de forma no orientada a conexión) a la otra. Una autoridad central puede o no estar involucrada. En los protocolos más sencillos, la entidad receptora acepta o rechaza la petición y, en el primer caso, se pasa a las

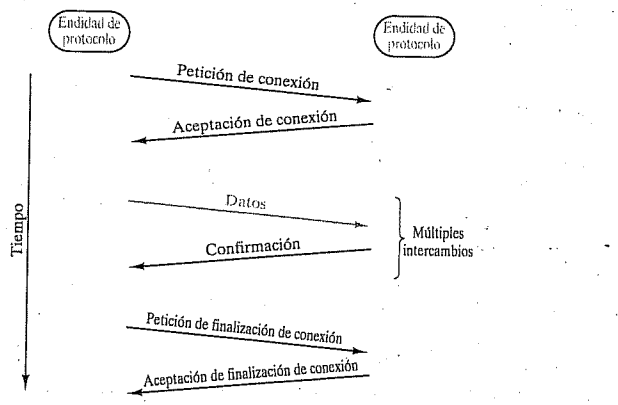


FIGURA 15.3 Fases de una transferencia de datos orientada a conexión.

siguientes fases. En protocolos más complejos, esta fase incluye una negociación relativa a la sintaxis, semántica y temporización del protocolo. Es evidente que ambas entidades deben usar el mismo protocolo, pero éste debe permitir determinadas características opcionales que deben ser aceptadas a través del proceso de negociación. Por ejemplo, el protocolo puede especificar un tamaño máximo de PDU de 8.000 octetos; una estación lo puede reducir a 1.000 octetos.

Una vez establecida la conexión, se pasa a la fase de transferencia de datos. Durante esta fase se intercambian tanto datos como información de control (por ejemplo, control de flujo, control de errores). La figura muestra una situación en la que todos los datos fluyen en una dirección, con envío de confirmaciones en la otra dirección. Resulta más usual que tanto los datos como las confirmaciones fluyan en ambas direcciones. Por último, un extremo o el otro desean liberar la conexión, lo que consiguen enviando una petición de liberación. Alternativamente, una autoridad central podría ser la encargada de liberarla.

La característica más importante de la transferencia de datos orientada a conexión es que sigue un orden secuencial. Cada extremo numera en orden secuencial las PDU y las envía hacia el otro extremo. Dado que cada extremo recuerda que hay establecida una conexión lógica, puede mantener los números de salida, generados por él, y los entrantes, generados por el otro extremo. Realmente podemos definir una transferencia de datos orientada a conexión esencialmente como una en la que ambos extremos numeran las PDU y gestionan los números de entrada y de salida. El orden secuencial permite la realización de tres funciones principales: entrega ordenada, control de flujo y control de errores.

Entrega ordenada

Si dos entidades comunicadas se encuentran en estaciones diferentes en una red, existe el peligro de que las PDU no se reciban en el mismo orden en que fueron enviadas debido a que siguen diferentes caminos a través de la red. En protocolos orientados a conexión se necesita generalmente que se mantenga el orden de las PDU. Por ejemplo, si se transfiere un fichero entre dos sistemas, nos gustaría estar seguros de que los registros del fichero recibido se encuentren en el mismo orden que los del fichero transmitido, y no mezclados. Si cada PDU tiene un único número, y los números se asignan de forma secuencial, la reordenación de las PDU recibidas en base a los números de secuencia resulta una tarea lógica sencilla para la entidad receptora. El único problema de este esquema es que los números de secuencia se repitan debido al uso de un campo finito de números de secuencia (módulo algún número máximo). Evidentemente, el número de secuencia máximo debe ser mayor que el número máximo de PDU que pueden estar pendientes en cualquier instante de tiempo. De hecho, el número máximo puede necesitar ser el doble del número máximo de PDU pendientes (por ejemplo, ARQ de repetición selectiva; ver Capítulo 6).

Control de flujo

El control de flujo se introdujo en el Capítulo 6 y se vió de nuevo en el Capítulo 9. En esencia, el control de flujo es una función realizada por la entidad receptora para limitar la cantidad o tasa de datos que envía la entidad emisora.

La forma más sencilla de control de flujo es un procedimiento de parada y espera ("stop-and-wait"), en el que cada PDU debe ser confirmada antes de que se envíe la siguiente. El uso de protocolos más eficientes implica la utilización de alguna forma de crédito ofrecido por el emisor, que es la cantidad de datos que se pueden enviar sin necesidad de confirmación. La técnica de ventana deslizante es un ejemplo de este mecanismo.

El control de flujo es un buen ejemplo de una función que debe implementarse en varios protocolos. Consideremos de nuevo la Figura 1.4. La red A necesitará realizar control de flujo sobre el módulo de servicios de red de la estación 1 a través del protocolo de acceso a la red con el fin de forzar el control de tráfico en la red. Al mismo tiempo, el módulo de servicios de red de la estación 2 tiene limitada su memoria temporal, requiriendo, por tanto, llevar a cabo el control de flujo de los servicios del módulo de servicios de red de la estación 1 mediante el protocolo proceso-proceso. Por último, aunque el módulo de servicios de red de la estación 2 puede controlar su flujo de datos, la aplicación de la estación 2 puede ser vulnerable a un flujo excesivo. Por ejemplo, la aplicación podría bloquearse esperando un acceso a disco. Por tanto, el control de flujo es también necesario para protocolos orientados a aplicaciones.

Control de errores

Otra función estudiada con anterioridad es el control de errores. Es necesario el uso de técnicas para gestionar la pérdida o los errores de datos e información de control. La mayor parte de las técnicas incluyen detección de errores, basada en el uso de una secuencia de comprobación de trama, y retransmisión de PDU. La retransmisión se consigue a veces mediante el uso de un temporizador. Si una entidad emisora no recibe una confirmación de una PDU dentro de un período de tiempo especificado, retransmitirá los datos.

Como en el caso de control de flujo, el control de errores es una función que debe ser realizada en varios niveles del protocolo. Consideremos de nuevo la Figura 1.6. El protocolo de acceso a la red debería incluir control de errores para asegurar que los datos se intercambian correctamente entre la estación y la red. Sin embargo, un paquete de datos puede perderse en la red, debiendo ser capaz el protocolo proceso-proceso de recuperar esta pérdida.

Direccionamiento

El concepto de direccionamiento en una arquitectura de comunicaciones es complejo y abarca un gran número de cuestiones. Al menos cuatro de ellas deben ser discutidas:

- Nivel de direccionamiento
- Ámbito del direccionamiento
- Identificadores de conexión
- Modo de direccionamiento

A lo largo de esta discusión ilustraremos los conceptos haciendo uso de la Figura 15.4, que muestra una configuración empleando la arquitectura de protocolo TCP/IP. Los conceptos son esencialmente los mismos para la arquitectura OSI o cualquier otra arquitectura de comunicaciones.

El *nivel de direccionamiento* hace mención al nivel en que se llama a una entidad dentro de la arquitectura de comunicaciones. Generalmente, en una configuración se asocia una única dirección a cada sistema final (por ejemplo, estación o terminal) y a cada sistema intermedio (por ejemplo, un dispositivo de encaminamiento). Esta dirección es, en general, una dirección del nivel de red. En el caso de la arquitectura TCP/IP ésta se conoce como una dirección IP, o simplemente dirección internet. En el caso de la arquitectura OSI, esta dirección se conoce como punto de acceso al servicio de red (NSAP, "Network Service Access Point"). La dirección del nivel de red se usa para encaminar una PDU a través de una o varias redes hacia un sistema especificado por la dirección del nivel de red en la PDU.

Una vez que se reciben los datos en el sistema de destino, deben ser encaminados hacia algún proceso o aplicación en el sistema. Generalmente un sistema admitirá múltiples aplica-

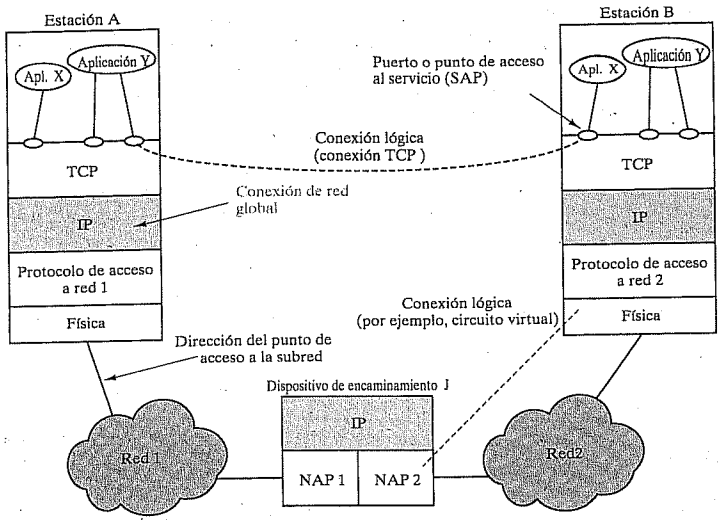


FIGURA 15.4 Conceptos de direccionamiento.

ciones y una aplicación puede admitir múltiples usuarios. A cada aplicación y, quizá, a cada usuario concurrente de una aplicación se le asigna un único identificador, denominado puerto en la arquitectura TCP/IP y punto de acceso de servicio (SAP, "Service Access Point") en la arquitectura OSI. Por ejemplo, un sistema estación podría soportar tanto una aplicación de correo electrónico como una de transferencia de ficheros. Como mínimo, cada aplicación tendría un número de puerto o SAP que es único en el sistema. Además, la aplicación de transferencia de ficheros podría admitir múltiples transferencias simultáneas, en cuyo caso, a cada transferencia se le asigna dinámicamente un único número de puerto o SAP.

La Figura 15.4 ilustra dos niveles de direccionamiento en un sistema. Éste es el caso general en la arquitectura TCP/IP. Sin embargo, puede existir direccionamiento en cada nivel de una arquitectura. Por ejemplo, se puede asignar un único SAP a cada nivel de la arquitectura OSI.

Otra cuestión relacionada con la dirección de un sistema final o intermedio es el *ámbito de direccionamiento*. La dirección internet o dirección NSAP mencionada anteriormente es una dirección global. Las características más importantes de una dirección global son:

- *Sin ambigüedad global*: una dirección global identifica un único sistema, permitiéndose el uso de sinónimos; es decir, un sistema puede tener más de una dirección global.
- *Aplicabilidad global*: en una dirección global es posible identificar cualquier otra dirección global, en cualquier sistema, mediante la dirección global del otro sistema.

Debido a que una dirección global es única y globalmente aplicable, se posibilita a una internet el encaminamiento de datos desde cualquier sistema conectado en una red hacia otro sistema conectado en otra red.

La Figura 15.4 ilustra la necesidad de otro nivel de direccionamiento. Cada subred debe mantener una única dirección para cada dispositivo de interfaz con la subred. Algunos ejemplos son la dirección MAC en una red IEEE 802 y una dirección DTE X.25. Esta dirección permite a la subred encaminar unidades de datos (por ejemplo, tramas MAC, paquetes X.25) a través de la subred y hacer entrega de ellas al sistema conectado de destino. A una dirección así la podemos llamar dirección de punto de conexión a la subred.

El ámbito de direccionamiento es generalmente relevante sólo para direcciones del nivel de red. Un puerto o SAP por encima del nivel de red es único en un sistema específico, pero no necesita ser único globalmente. Por ejemplo, en la Figura 15.4, puede existir un puerto 1 en el sistema A y un puerto 1 en el sistema B. La designación completa de estos dos puertos se podría expresar como A.1 y B.1, lo que los identifica unívocamente.

El concepto de *identificadores de conexión* entra en juego cuando se consideran transferencias de datos orientadas a conexión (por ejemplo, circuito virtual) en lugar de transferencias no orientadas a conexión (por ejemplo, datagrama). Para transferencias de datos no orientadas a conexión se usa un nombre global para cada transmisión de datos. Para transferencias orientadas a conexión es deseable a veces el uso de un único nombre de conexión durante la fase de transferencia de datos. La situación es la siguiente: la entidad 1 en el sistema A solicita una conexión a la entidad 2 en el sistema B, haciendo uso quizá de la dirección global B.2. Cuando B.2 acepta la conexión, se proporciona un nombre de conexión (generalmente un número) y se usa por ambas entidades en transmisiones futuras. El uso de un nombre de conexión presenta varias ventajas:

- *Coste suplementario reducido*: los nombres de conexión son generalmente más cortos que los nombres globales. Por ejemplo, en el protocolo X.25 (discutido en el Capítulo 9) usado en redes de conmutación de paquetes, los paquetes de petición de conexión contienen los campos de dirección origen y de destino, cada uno de ellos con una longitud definida en el sistema, que puede ser de varios octetos. Tras el establecimiento de un circuito virtual, los paquetes de datos contienen sólo un número de circuito virtual de 12 bits.
- *Encaminamiento*: en el establecimiento de una conexión se puede definir una ruta fija, estática. El nombre de conexión le sirve a los sistemas intermedios, tales como nodos de conmutación de paquetes, para identificar la ruta con el fin de gestionar futuras PDU.
- *Multiplexación*: esta función se trata más adelante en términos más generales. Aquí indicaremos sólo el hecho de que una entidad puede hacer uso de más de una conexión de forma simultánea. De este modo, las PDU entrantes deben identificarse por un nombre de conexión.
- *Uso de información de estado*: una vez que se ha establecido una conexión, los sistemas finales pueden mantener información de estado relacionada con la conexión. Esto posibilita funciones tales como control de flujo y control de errores, empleando números de secuencia. Ejemplos de esto se muestran con HDLC (Capítulo 6) y X.25 (Capítulo 9).

La Figura 15.4 muestra varios ejemplos de conexiones. La conexión lógica entre el dispositivo de encaminamiento J y la estación B se establece a nivel de red. Por ejemplo, si la red 2 es de conmutación de paquetes usando X.25, esta conexión lógica sería un circuito virtual. En un nivel superior, varios protocolos del nivel de transporte, tales como TCP, admiten conexiones lógicas entre usuarios del servicio de transporte. Así, TCP puede mantener una conexión entre dos puertos de sistemas diferentes.

Otro concepto importante en direccionamiento es el *modo de direccionamiento*. Una dirección se refiere generalmente a un único sistema o puerto; en este caso se le denomina *dirección indivi-*

TABLA 15.1 Modos de direccionamiento.

Destino	Dirección de red	Dirección de sistema	Dirección puerto/SAP
Individual	Individual	Individual	Individual
Multidestino	Individual	Individual	Grupo
	Individual	Todas	Grupo
	Todas	Todas	Grupo
Difusión	Individual	Individual	Todas
	Individual	Todas	Todas
	Todas	Todas	Todas

dual. Es posible también que una dirección se refiera a más de una entidad o puerto. Este tipo de dirección identifica simultáneamente múltiples receptores de datos. Por ejemplo, un usuario podría desear enviar una nota a varios individuos. El centro de control de red puede querer notificar a todos los usuarios que la red se va a desactivar. Una dirección con múltiples receptores o destinos puede ser de difusión, dirigida a todas las entidades en un dominio, o multidestino, dirigida a un subconjunto específico de entidades. La Tabla 15.1 muestra las distintas posibilidades.

Multiplexación

Relacionado con el concepto de direccionamiento se encuentra el de multiplexación. En un sistema individual se admite una forma de multiplexación mediante múltiples conexiones. Por ejemplo, con X.25 pueden existir múltiples circuitos virtuales que terminan en el mismo sistema final; podemos decir que estos circuitos virtuales están multiplexados sobre la interfaz física entre el sistema final y la red. La multiplexación también se puede realizar mediante el uso de nombres de puerto, que también permiten múltiples conexiones simultáneas. Por ejemplo, pueden existir varias conexiones TCP que terminan en un sistema específico, donde cada conexión contiene un par diferente de puertos.

La multiplexación se usa también en otro contexto, a saber, la transformación de conexiones de un nivel en otro. Consideremos de nuevo la Figura 15.4. La red A podría ofrecer un servicio de circuito virtual. Se podría crear un circuito virtual en el nivel de acceso a la red para una conexión proceso-proceso establecida en el nivel de servicios de red. Ésta es una relación uno a uno, pero no es necesario que sea así. La multiplexación se puede usar en una o en dos direcciones (Figura 15.5). La multiplexación hacia arriba ("upward") tiene lugar cuando se multiplexan, o comparten, varias conexiones del nivel superior a través de una sola conexión del nivel inferior. Esto puede resultar necesario para hacer un uso más eficiente de los servicios del nivel inferior o para proporcionar varias conexiones del nivel superior en un entorno en el que existe una sola conexión del nivel inferior. La Figura 15.5 muestra un ejemplo de multiplexación hacia arriba. La multiplexación hacia abajo ("downward"), o división, implica que se crea una sola conexión del nivel superior encima de múltiples conexiones del nivel inferior, dividiéndose el tráfico de la conexión superior entre las distintas conexiones inferiores. Esta técnica se puede usar para proporcionar fiabilidad, prestaciones o eficiencia.

Servicios de transmisión

Un protocolo puede ofrecer una gran variedad de servicios adicionales a las entidades que hagan uso de él. Aquí mencionamos tres ejemplos comunes:

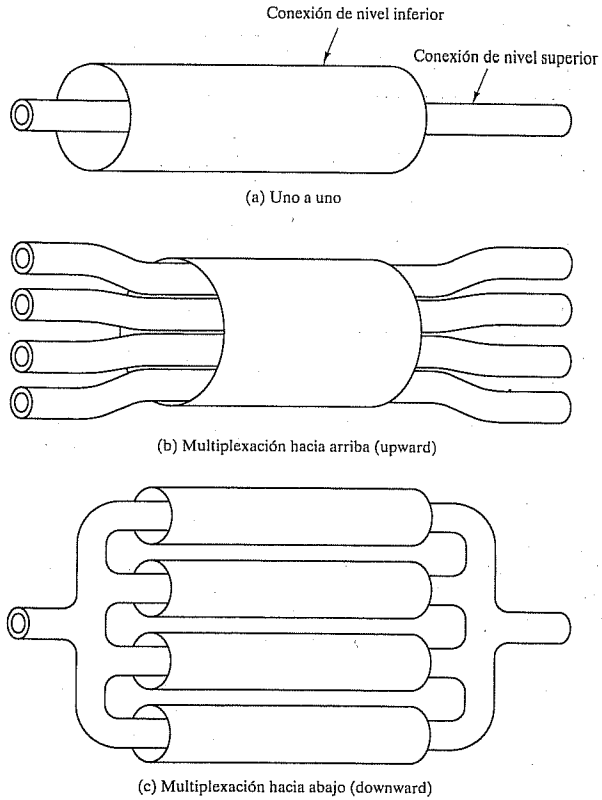


FIGURA 15.5 Multiplexación y conexiones de protocolo.

- *Prioridad*: determinados mensajes, tales como los mensajes de control, pueden necesitar ir hacia la entidad de destino con un retardo mínimo. Un ejemplo podría ser una petición de cierre de conexión. De este modo, la prioridad se podría asignar tomando en consideración el mensaje. Adicionalmente, ésta se podría asignar en base a una conexión.
- *Grado de servicio*: ciertas clases de datos pueden necesitar un umbral de rendimiento mínimo u otro de retardo máximo.
- *Seguridad*: se pueden utilizar mecanismos de seguridad, de acceso restringido.

Todos estos servicios dependen del sistema de transmisión subyacente y de cualesquiera entidades de nivel inferior que intervengan. Si es posible ofrecer estos servicios hacia abajo, puede usarse el protocolo por las dos entidades para llevarlos a cabo.