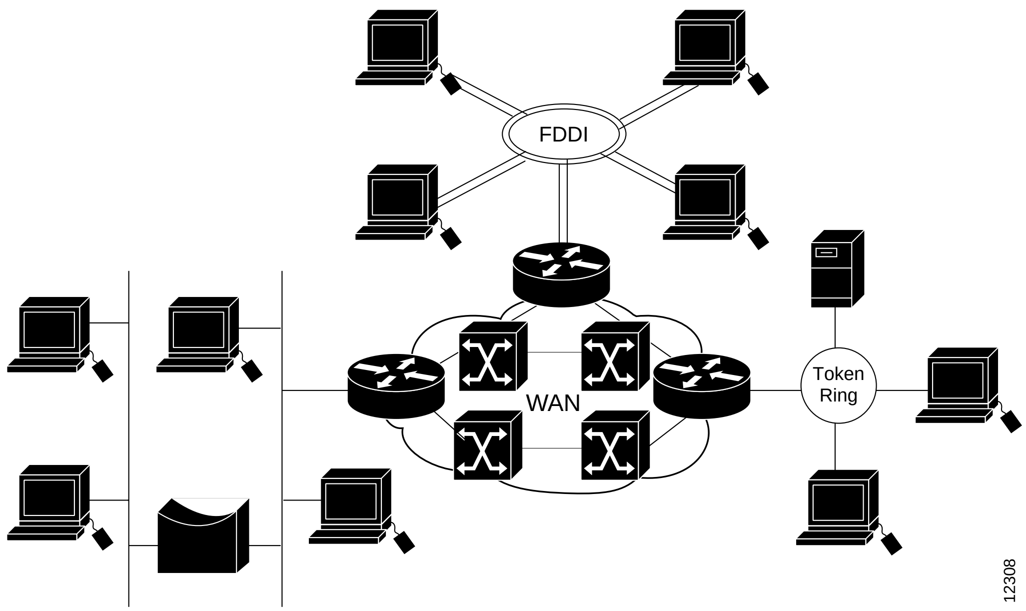
Introduction to Internetworking

This chapter works with the next six chapters to act as a foundation for the technology discussions that follow. In this chapter, some fundamental concepts and terms used in the evolving language of internetworking are addressed. In the same way that this book provides a foundation for understanding modern networking, this chapter summarizes some common themes presented throughout the remainder of this book. Topics include flow control, error checking, and multiplexing, but this chapter focuses mainly on mapping the *Open Systems Interconnect* (*OSI*) model to networking/internetworking functions and summarizes the general nature of addressing schemes within the context of the OSI model.

What is an Internetwork?

An internetwork is a collection of individual networks, connected by intermediate networking devices, that functions as a single large network. Internetworking refers to the industry, products, and procedures that meet the challenge of creating and administering internetworks. Figure 1-1 illustrates some different kinds of network technologies that can be interconnected by routers and other networking devices to create an internetwork:

Figure 1-1 Different network technologies can be connected to create an internetwork.



History of Internetworking

The first networks were time-sharing networks that used mainframes and attached terminals. Such environments were implemented by both IBM's System Network Architecture (SNA) and Digital's network architecture.

Local area networks (LANs) evolved around the PC revolution. LANs enabled multiple users in a relatively small geographical area to exchange files and messages, as well as access shared resources such as file servers.

Wide area networks (WANs) interconnect LANs across normal telephone lines (and other media), thereby interconnecting geographically dispersed users.

Today, high-speed LANs and switched internetworks are becoming widely used, largely because they operate at very high speeds and support such high-bandwidth applications as voice and video conferencing.

Internetworking evolved as a solution to three key problems: isolated LANs, duplication of resources, and a lack of network management. Isolated LANS made electronic communication between different offices or departments impossible. Duplication of resources meant that the same hardware and software had to be supplied to each office or department, as did a separate support staff. This lack of network management meant that no centralized method of managing and troubleshooting networks existed.

Internetworking Challenges

Implementing a functional internetwork is no simple task. Many challenges must be faced, especially in the areas of connectivity, reliability, network management, and flexibility. Each area is key in establishing an efficient and effective internetwork.

The challenge when connecting various systems is to support communication between disparate technologies. Different sites, for example, may use different types of media, or they might operate at varying speeds.

Another essential consideration, reliable service, must be maintained in any internetwork. Individual users and entire organizations depend on consistent, reliable access to network resources.

Furthermore, network management must provide centralized support and troubleshooting capabilities in an internetwork. Configuration, security, performance, and other issues must be adequately addressed for the internetwork to function smoothly.

Flexibility, the final concern, is necessary for network expansion and new applications and services, among other factors.

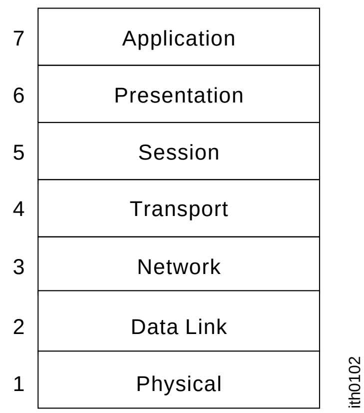
Open Systems Interconnection (OSI) Reference Model

The Open Systems Interconnection (OSI) reference model describes how information from a software application in one computer moves through a network medium to a software application in another computer. The OSI reference model is a conceptual model composed of seven layers, each specifying particular network functions. The model was developed by the International Organization for Standardization (ISO) in 1984, and it is now considered the primary architectural model for intercomputer communications. The OSI model divides the tasks involved with moving information between networked computers into seven smaller, more manageable task groups. A task or group of tasks is then assigned to each of the seven OSI layers. Each layer is reasonably self-contained, so that the tasks assigned to each layer can be implemented independently. This enables the solutions offered by one layer to be updated without adversely affecting the other layers. The following list details the seven layers of the Open System Interconnection (OSI) reference model:

- Layer 7—Application layer
- Layer 6—Presentation layer
- Layer 5—Session layer
- Layer 4—Transport layer
- Layer 3—Network layer
- Layer 2—Data Link layer
- Layer 1—Physical layer

Figure 1-2 illustrates the seven-layer OSI reference model.

Figure 1-2 The OSI reference model contains seven independent layers.



Characteristics of the OSI Layers

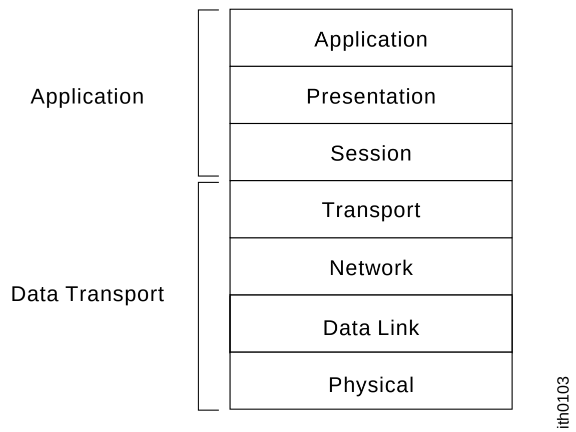
The seven layers of the OSI reference model can be divided into two categories: *upper layers* and *lower layers*.

The *upper layers* of the OSI model deal with application issues and generally are implemented only in software. The highest layer, application, is closest to the end user. Both users and application-layer processes interact with software applications that contain a communications component. The term upper layer is sometimes used to refer to any layer above another layer in the OSI model.

The *lower layers* of the OSI model handle data transport issues. The physical layer and data-link layer are implemented in hardware and software. The other lower layers generally are implemented only in software. The lowest layer, the physical layer, is closest to the physical network medium (the network cabling, for example, and is responsible for actually placing information on the medium.

Figure 1-3 illustrates the division between the upper and lower OSI layers.

Figure 1-3 Two sets of layers make up the OSI layers.



Protocols

The OSI model provides a conceptual framework for communication between computers, but the model itself is not a method of communication. Actual communication is made possible by using communication protocols. In the context of data networking, a *protocol* is a formal set of rules and conventions that governs how computers exchange information over a network medium. A protocol implements the functions of one or more of the OSI layers. A wide variety of communication protocols exist, but all tend to fall into one of the following groups: *LAN protocols*, *WAN protocols*, *network protocols*, and *routing protocols. LAN protocols* operate at the physical and data-link layers of the OSI model and define communication over the various LAN media. *WAN protocols* operate at the lowest three layers of the OSI model and define communication over the various wide-area media. *Routing protocols* are network-layer protocols that are responsible for path determination and traffic switching. Finally, *network protocols* are the various upper-layer protocols that exist in a given protocol suite.

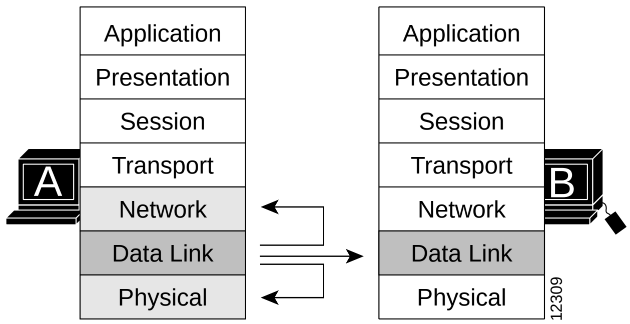
OSI Model and Communication Between Systems

Information being transferred from a software application in one computer system to a software application in another must pass through each of the OSI layers. If, for example, a software application in System A has information to transmit to a software application in System B, the application program in System A will pass its information to the application layer (Layer 7) of System A. The application layer then passes the information to the presentation layer (Layer 6), which relays the data to the session layer (Layer 5), and so on down to the physical layer (Layer 1). At the physical layer, the information is placed on the physical network medium and is sent across the medium to System B.The physical layer of System B removes the information from the physical medium, and then its physical layer passes the information up to the data link layer (Layer 2), which passes it to the network layer (Layer 3), and so on until it reaches the application layer (Layer 7) of System B. Finally, the application layer of System B passes the information to the recipient application program to complete the communication process.

Interaction Between OSI Model Layers

A given layer in the OSI layers generally communicates with three other OSI layers: the layer directly above it, the layer directly below it, and its peer layer in other networked computer systems. The data link layer in System A, for example, communicates with the network layer of System A, the physical layer of System A, and the data link layer in System B. Figure 1-4 illustrates this example.

Figure 1-4 OSI model layers communicate with other layers.



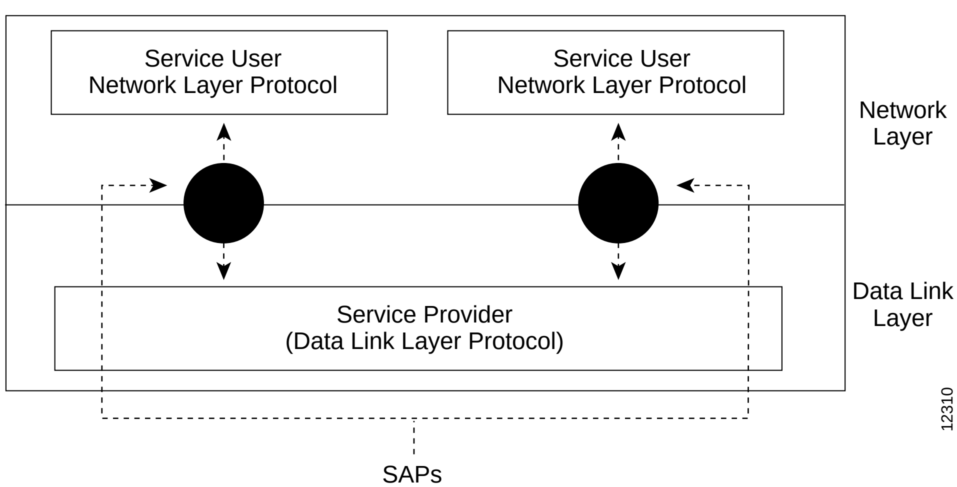
OSI-Layer Services

One OSI layer communicates with another layer to make use of the services provided by the second layer. The services provided by adjacent layers help a given OSI layer communicate with its peer layer in other computer systems. Three basic elements are involved in layer services: the service user, the service provider, and the service access point (SAP).

In this context, the service *user* is the OSI layer that requests services from an adjacent OSI layer. The service *provider* is the OSI layer that provides services to service users. OSI layers can provide services to multiple service users. The *SAP* is a conceptual location at which one OSI layer can request the services of another OSI layer.

Figure 1-5 illustrates how these three elements interact at the network and data link layers.

Figure 1-5 Service users, providers, and SAPs interact at the network and data link layers.



OSI Model Layers and Information Exchange

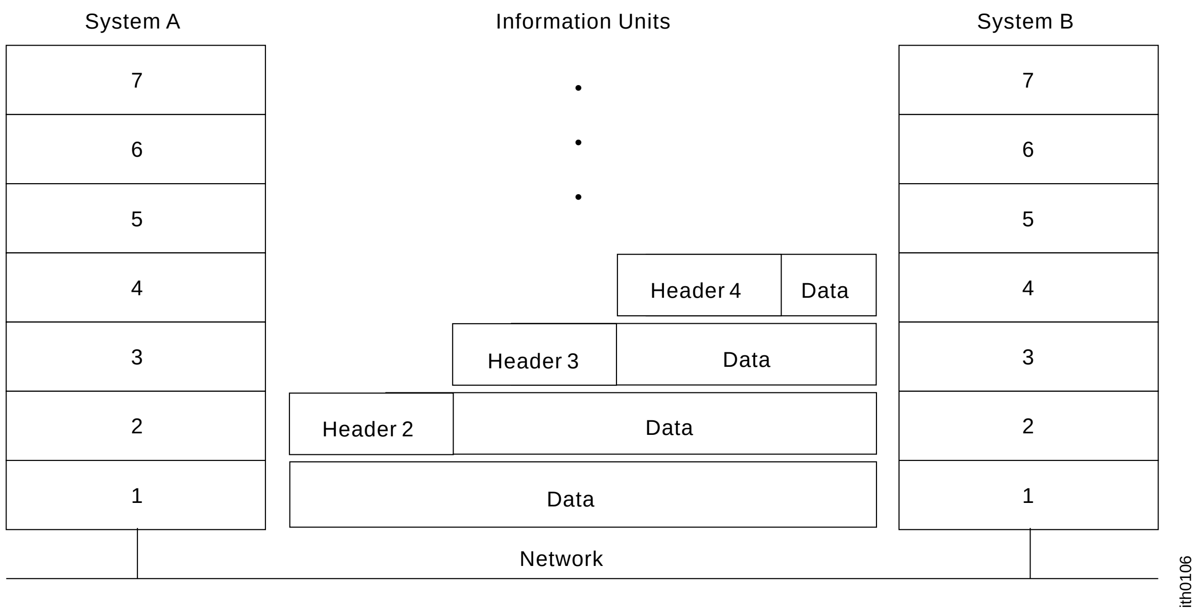
The seven OSI layers use various forms of control information to communicate with their peer layers in other computer systems. This *control information* consists of specific requests and instructions that are exchanged between peer OSI layers.

Control information typically takes one of two forms: headers and trailers. Headers are prepended to data that has been passed down from upper layers.Trailers are appended to data that has been passed down from upper layers. An OSI layer is not required to attach a header or trailer to data from upper layers.

Headers, trailers, and data are relative concepts, depending on the layer that analyzes the information unit. At the network layer, an information unit, for example, consists of a Layer 3 header and data. At the data link layer, however, all the information passed down by the network layer (the Layer 3 header and the data) is treated as data.

In other words, the data portion of an information unit at a given OSI layer potentially can contain headers, trailers, and data from all the higher layers. This is known as *encapsulation*. Figure 1-6 shows how the header and data from one layer are encapsulated into the header of the next lowest layer.

Figure 1-6 Headers and data can be encapsulated during information exchange.



Information Exchange Process

The information exchange process occurs between peer OSI layers. Each layer in the source system adds control information to data and each layer in the destination system analyzes and removes the control information from that data.

If System A has data from a software application to send to System B, the data is passed to the application layer. The application layer in System A then communicates any control information required by the application layer in System B by prepending a header to the data. The resulting information unit (a header and the data) is passed to the presentation layer, which prepends its own header containing control information intended for the presentation layer in System B.

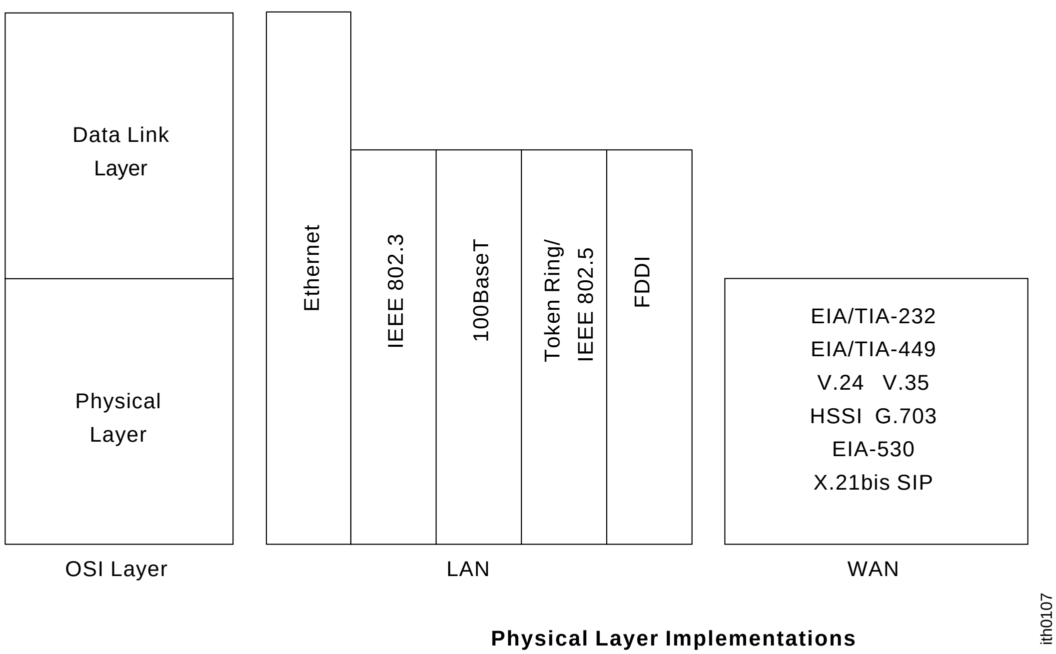
The information unit grows in size as each layer prepends its own header (and in some cases a trailer) that contains control information to be used by its peer layer in System B. At the physical layer, the entire information unit is placed onto the network medium.

The physical layer in System B receives the information unit and passes it to the data-link layer. The data link layer in System B then reads the control information contained in the header prepended by the data link layer in System A. The header is then removed, and the remainder of the information unit is passed to the network layer. Each layer performs the same actions: The layer reads the header from its peer layer, strips it off, and passes the remaining information unit to the next highest layer. After the application layer performs these actions, the data is passed to the recipient software application in System B, in exactly the form in which it was transmitted by the application in System A.

OSI Model Physical Layer

The physical layer defines the electrical, mechanical, procedural, and functional specifications for activating, maintaining, and deactivating the physical link between communicating network systems. Physical layer specifications define characteristics such as voltage levels, timing of voltage changes, physical data rates, maximum transmission distances, and physical connectors. Physical-layer implementations can be categorized as either LAN or WAN specifications. Figure 1-7 illustrates some common LAN and WAN physical-layer implementations.

Figure 1-7 Physical-layer implementations can be LAN or WAN specifications.

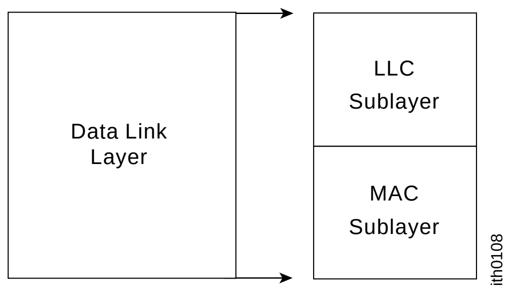


OSI Model Data-Link Layer

The data link layer provides reliable transit of data across a physical network link. Different data link layer specifications define different network and protocol characteristics, including physical addressing, network topology, error notification, sequencing of frames, and flow control. Physical addressing (as opposed to network addressing) defines how devices are addressed at the data link layer. Network topology consists of the data-link layer specifications that often define how devices are to be physically connected, such as in a bus or a ring topology. Error notification alerts upper-layer protocols that a transmission error has occurred, and the sequencing of data frames reorders frames that are transmitted out of sequence. Finally, flow control moderates the transmission of data so that the receiving device is not overwhelmed with more traffic than it can handle at one time.

The Institute of Electrical and Electronics Engineers (IEEE) has subdivided the data-link layer into two sublayers: Logical Link Control (LLC) and Media Access Control (MAC). Figure 1-8 illustrates the IEEE sublayers of the data-link layer.

Figure 1-8 The data link layer contains two sublayers.



The Logical Link Control (LLC) sublayer of the data-link layer manages communications between devices over a single link of a network. LLC is defined in the IEEE 802.2 specification and supports both connectionless and connection-oriented services used by higher-layer protocols. IEEE 802.2 defines a number of fields in data-link layer frames that enable multiple higher-layer protocols to share a single physical data link. The Media Access Control (MAC) sublayer of the data link layer manages protocol access to the physical network medium. The IEEE MAC specification defines MAC addresses, which enable multiple devices to uniquely identify one another at the data link layer.

OSI Model Network Layer

The network layer provides routing and related functions that enable multiple data links to be combined into an internetwork. This is accomplished by the logical addressing (as opposed to the physical addressing) of devices. The network layer supports both connection-oriented and connectionless service from higher-layer protocols. Network-layer protocols typically are routing protocols, but other types of protocols are implemented at the network layer as well. Some common routing protocols include Border Gateway Protocol (BGP), an Internet interdomain routing protocol; Open Shortest Path First (OSPF), a link-state, interior gateway protocol developed for use in TCP/IP networks; and Routing Information Protocol (RIP), an Internet routing protocol that uses hop count as its metric.

OSI Model Transport Layer

The transport layer implements reliable internetwork data transport services that are transparent to upper layers. Transport-layer functions typically include flow control, multiplexing, virtual circuit management, and error checking and recovery.

Flow control manages data transmission between devices so that the transmitting device does not send more data than the receiving device can process. Multiplexing enables data from several applications to be transmitted onto a single physical link. Virtual circuits are established, maintained, and terminated by the transport layer. Error checking involves creating various mechanisms for detecting transmission errors, while error recovery involves taking an action, such as requesting that data be retransmitted, to resolve any errors that occur.

Some transport-layer implementations include Transmission Control Protocol, Name Binding Protocol, and OSI transport protocols. Transmission Control Protocol (TCP) is the protocol in the TCP/IP suite that provides reliable transmission of data. \Name Binding Protocol (NBP) is the protocol that associates AppleTalk names with addresses. OSI transport protocols are a series of transport protocols in the OSI protocol suite.

OSI Model Session Layer

The session layer establishes, manages, and terminates communication sessions between presentation layer entities. Communication sessions consist of service requests and service responses that occur between applications located in different network devices. These requests and responses are coordinated by protocols implemented at the session layer. Some examples of session-layer implementations include Zone Information Protocol (ZIP), the AppleTalk protocol that coordinates the name binding process; and Session Control Protocol (SCP), the DECnet Phase IV session-layer protocol.

OSI Model Presentation Layer

The presentation layer provides a variety of coding and conversion functions that are applied to application layer data. These functions ensure that information sent from the application layer of one system will be readable by the application layer of another system. Some examples of presentation-layer coding and conversion schemes include common data representation formats, conversion of character representation formats, common data compression schemes, and common data encryption schemes.

Common data representation formats, or the use of standard image, sound, and video formats, enable the interchange of application data between different types of computer systems. Conversion schemes are used to exchange information with systems by using different text and data representations, such as EBCDIC and ASCII. Standard data compression schemes enable data that is compressed at the source device to be properly decompressed at the destination. Standard data encryption schemes enable data encrypted at the source device to be properly deciphered at the destination.

Presentation-layer implementations are not typically associated with a particular protocol stack. Some well-known standards for video include QuickTime and Motion (MPEG). QuickTime is an Apple Computer specification for video and audio, and MPEG is a standard for video compression and coding.

Among the well-known graphic image formats are Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), and Tagged Image File Format (TIFF). GIF is a standard for compressing and coding graphic images. JPEG is another compression and coding standard for graphic images, and TIFF is a standard coding format for graphic images.

OSI Model Application Layer

The application layer is the OSI layer closest to the end user, which means that both the OSI application layer and the user interact directly with the software application.

This layer interacts with software applications that implement a communicating component. Such application programs fall outside the scope of the OSI model. Application-layer functions typically include identifying communication partners, determining resource availability, and synchronizing communication.

When identifying communication partners, the application layer determines the identity and availability of communication partners for an application with data to transmit. When determining resource availability, the application layer must decide whether sufficient network resources for the requested communication exist. In synchronizing communication, all communication between applications requires cooperation that is managed by the application layer.

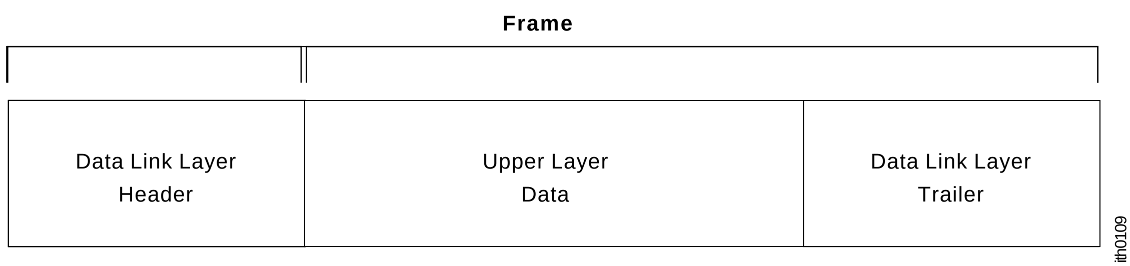
Two key types of application-layer implementations are TCP/IP applications and OSI applications. TCP/IP applications are protocols, such as lTelnet, File Transfer Protocol (FTP), and Simple Mail Transfer Protocol (SMTP), that exist in the Internet Protocol suite. OSI applications are protocols, such as File Transfer, Access, and Management (FTAM),F Virtual Terminal Protocol (VTP), and Common Management Information Protocol (CMIP), that exist in the OSI suite.

Information Formats

The data and control information that is transmitted through internetworks takes a wide variety of forms. The terms used to refer to these information formats are not used consistently in the internetworking industry but sometimes are used interchangeably. Common information formats include frame, packet, datagram, segment, message, cell, and data unit.

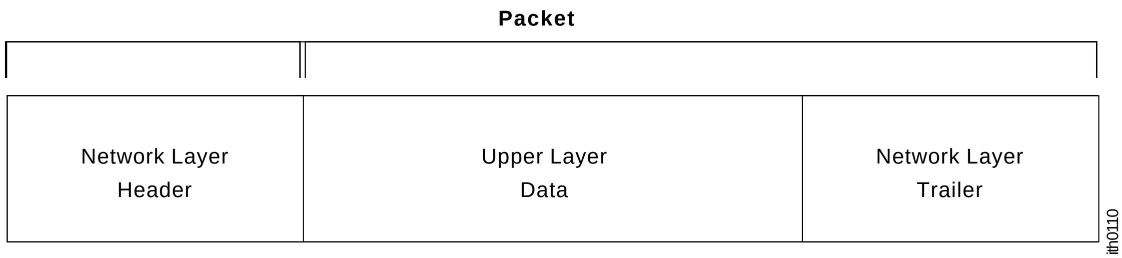
A frame is an information unit whose source and destination are data link layer entities. A frame is composed of the data-link layer header (and possibly a trailer) and upper-layer data. The header and trailer contain control information intended for the data-link layer entity in the destination system. Data from upper-layer entities is encapsulated in the data-link layer header and trailer. Figure 1-9 illustrates the basic components of a data-link layer frame.

Figure 1-9 Data from upper-layer entities makes up the data-link layer frame.



A packet is an information unit whose source and destination are network-layer entities. A packet is composed of the network-layer header (and possibly a trailer) and upper-layer data. The header and trailer contain control information intended for the network-layer entity in the destination system. Data from upper-layer entities is encapsulated in the network-layer header and trailer. Figure 1-10 illustrates the basic components of a network-layer packet.

Figure 1-10 Three basic components make up a network-layer packet.



The term *datagram* usually refers to an information unit whose source and destination are network-layer entities that use connectionless network service.

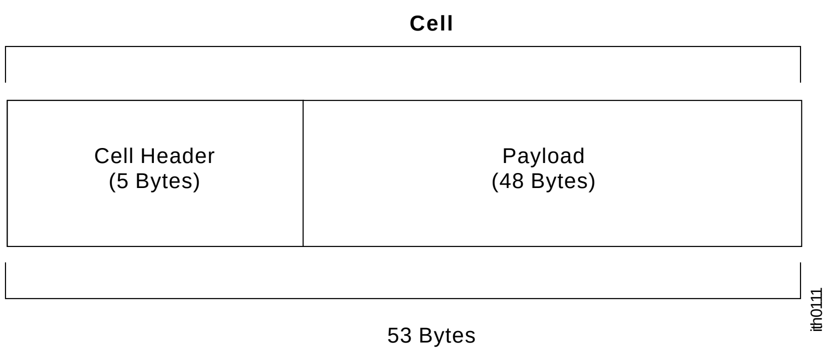
The term *segment* usually refers to an information unit whose source and destination are transport-layer entities.

A *message* is an information unit whose source and destination entities exist above the network layer (often the application layer).

A *cell* is an information unit of a fixed size whose source and destination are data-link layer entities. Cells are used in switched environments, such as Asynchronous Transfer Mode (ATM) and Switched Multimegabit Data Service (SMDS) networks. A cell is composed of the header and payload. The header contains control information intended for the destination data-link layer entity and is typically 5 bytes long. The payload contains upper-layer data that is encapsulated in the cell header and is typically 48 bytes long.

The length of the header and the payload fields always are exactly the same for each cell. Figure 1-11 depicts the components of a typical cell.

Figure 1-11 Two components make up a typical cell.



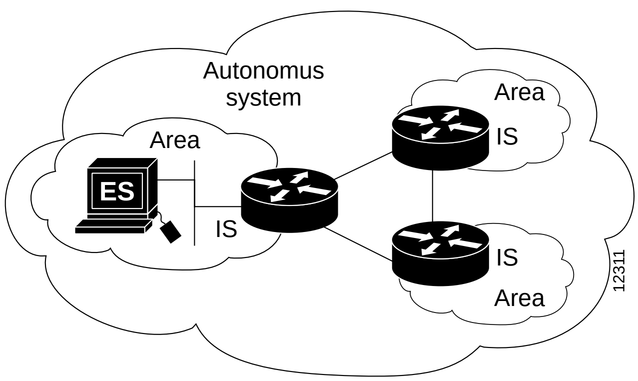
Data unit is a generic term that refers to a variety of information units. Some common data units are service data units (SDUs), protocol data units, and bridge protocol data units (BPDUs). SDUs are information units from upper-layer protocols that define a service request to a lower-layer protocol. PDU is OSI terminology for a packet. BPDUs are used by the spanning-tree algorithm as hello messages.

ISO Hierarchy of Networks

Large networks typically are organized as hierarchies. A hierarchical organization provides such advantages as ease of management, flexibility, and a reduction in unnecessary traffic. Thus, the International Organization for Standardization (ISO) has adopted a number of terminology conventions for addressing network entities. Key terms, defined in this section, include *end system* (ES), *intermediate system* (IS), *area*, and *autonomous system* (AS).

An *ES* is a network device that does not perform routing or other traffic-forwarding functions. Typical ESs include such devices as terminals, personal computers, and printers. An IS is a network device that performs routing or other traffic-forwarding functions. Typical *ISs* include such devices as routers, switches, and bridges. Two types of IS networks exist: intradomain IS and interdomain IS. An intradomain IS communicates within a single autonomous system, while an interdomain IS communicates within and between autonomous systems. An *area* is a logical group of network segments and their attached devices. Areas are subdivisions of autonomous systems. An AS is a collection of networks under a common administration that share a common routing strategy. Autonomous systems are subdivided into areas, and an AS is sometimes called a *domain*. Figure 1-12 illustrates a hierarchical network and its components.

Figure 1-12 A hierarchical network contains numerous components.



Connection-Oriented and Connectionless Network Services

In general, networking protocols and the data traffic that they support can be characterized as being either connection-oriented or connectionless. In brief, connection-oriented data handling involves using a specific path that is established for the duration of a connection. Connectionless data handling involves passing data through a permanently established connection.

Connection-oriented service involves three phases: connection establishment, data transfer, and connection termination.

During the connection-establishment phase, a single path between the source and destination systems is determined. Network resources typically are reserved at this time to ensure a consistent grade of service, such as a guaranteed throughput rate.

In the data-transfer phase, data is transmitted sequentially over the path that has been established. Data always arrives at the destination system in the order in which it was sent.

During the connection-termination phase, an established connection that is no longer needed is terminated. Further communication between the source and destination systems requires that a new connection be established.

Connection-oriented network service carries two significant disadvantages over connectionless, static-path selection and the static reservation of network resources. Static-path selection can create difficulty because all traffic must travel along the same static path. A failure anywhere along that path causes the connection to fail. Static reservation of network resources causes difficulty because it requires a guaranteed rate of throughput and, thus, a commitment of resources that other network users cannot share. Unless the connection uses full, uninterrupted throughput, bandwidth is not used efficiently.

Connection-oriented services, however, are useful for transmitting data from applications that don't tolerate delays and packet resequencing. Voice and video applications are typically based on connection-oriented services.

As another disadvantage, connectionless network service does not predetermine the path from the source to the destination system, nor are packet sequencing, data throughput, and other network resources guaranteed. Each packet must be completely addressed because different paths through the network may be selected for different packets, based on a variety of influences. Each packet is transmitted independently by the source system and is handled independently by intermediate network devices.

Connectionless service, however, offers two important advantages over connection-oriented service: dynamic-path selection and dynamic-bandwidth allocation. Dynamic-path selection enables traffic to be routed around network failures because paths are selected on a packet-by-packet basis. With dynamic-bandwidth allocation, bandwidth is used more efficiently because network resources are not allocated a bandwidth that they will not use.

Connectionless services are useful for transmitting data from applications that can tolerate some delay and resequencing. Data-based applications typically are based on connectionless service.

Internetwork Addressing

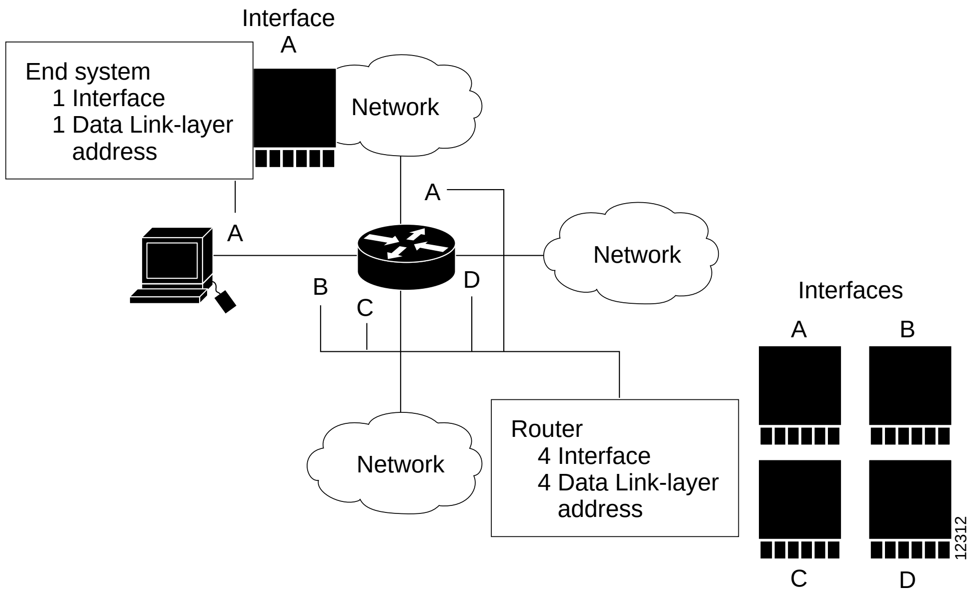
Internetwork addresses identify devices separately or as members of a group. Addressing schemes vary depending on the protocol family and the OSI layer. Three types of internetwork addresses are commonly used: data-link layer addresses, Media Access Control (MAC) addresses, and network-layer addresses.

Data Link Layer

A data-link layer address uniquely identifies each physical network connection of a network device. Data-link addresses sometimes are referred to as *physical* or *hardware* addresses. Data-link addresses usually exist within a flat address space and have a pre-established and typically fixed relationship to a specific device.

End systems generally have only one physical network connection, and thus have only one data-link address. Routers and other internetworking devices typically have multiple physical network connections and therefore also have multiple data-link addresses. Figure 1-13 illustrates how each interface on a device is uniquely identified by a data-link address.

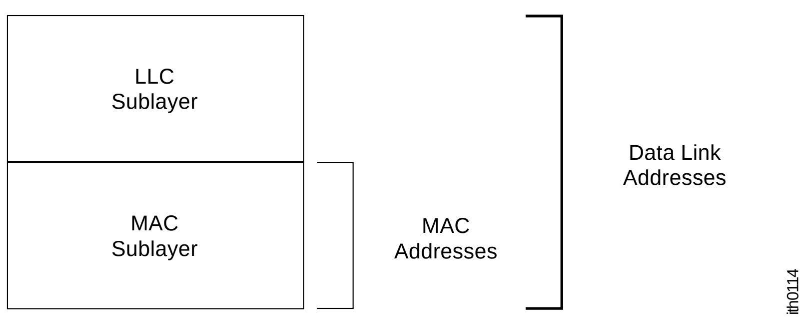
Figure 1-13 Each interface on a device is uniquely identified by a data-link address.



MAC Addresses

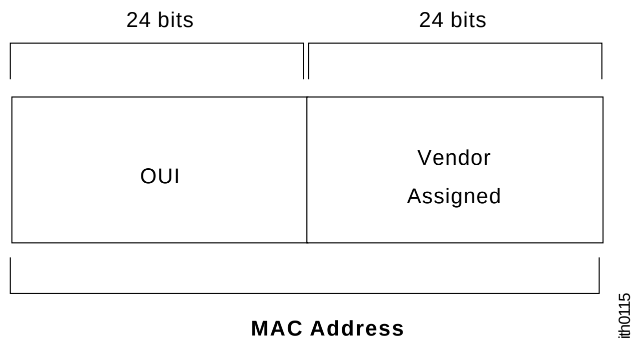
Media Access Control (MAC) addresses consist of a subset of data-link layer addresses. MAC addresses identify network entities in LANs that implement the IEEE MAC addresses of the data-link layer. As with most data-link addresses, MAC addresses are unique for each LAN interface. Figure 1-14 illustrates the relationship between MAC addresses, data-link addresses, and the IEEE sublayers of the data-link layer.

Figure 1-14 MAC addresses, data-link addresses, and the IEEE sublayers of the data-link layer are all related.



MAC addresses are 48 bits in length and are expressed as 12 hexadecimal digits. The first 6 hexadecimal digits, which are administered by the IEEE, identify the manufacturer or vendor and thus comprise the Organizational Unique Identifier (OUI). The last 6 hexadecimal digits comprise the interface serial number, or another value administered by the specific vendor. MAC addresses sometimes are called *burned-in addresses* (BIAs) because they are burned into read-only memory (ROM) and are copied into random-access memory (RAM) when the interface card initializes. Figure 1-15 illustrates the MAC address format.

Figure 1-15 The MAC address contains a unique format of hexadecimal digits.



Different protocol suites use different methods for determining the MAC address of a device. The following three methods are used most often: Address Resolution Protocol (ARP) maps network addresses to MAC addresses. Hello protocol enables network devices to learn the MAC addresses of other network devices. MAC addresses are either embedded in the network-layer address or are generated by an algorithm.

Address resolution is the process of mapping network addresses to Media Access Control (MAC) addresses. This process is accomplished by using the Address Resolution Protocol (ARP), which is implemented by many protocol suites.When a network address is successfully associated with a MAC address, the network device stores the information in the ARP cache. The ARP cache enables devices to send traffic to a destination without creating ARP traffic because the MAC address of the destination is already known.

The process of address resolution differs slightly, depending on the network environment. Address resolution on a single LAN begins when End System A broadcasts an ARP request onto the LAN in an attempt to learn the MAC address of End System B. The broadcast is received and processed by all devices on the LAN, although only End System B replies to the ARP request by sending an ARP reply containing its MAC address to End System A. End System A receives the reply and saves the MAC address of End System B in its ARP cache. (The ARP cache is where network addresses are associated with MAC addresses.)Whenever End System A must communicate with End System B, it checks the ARP cache, finds the MAC address of System B, and sends the frame directly without first having to use an ARP request.

Address resolution works differently, however, when source and destination devices are attached to different LANs that are interconnected by a router. End System Y broadcasts an ARP request onto the LAN in an attempt to learn the MAC address of End System Z. The broadcast is received and processed by all devices on the LAN, including Router X, which acts as a proxy for End System Z by checking its routing table to determine that End System Z is located on a different LAN. Router X then replies to the ARP request from End System Y, sending an ARP reply containing its *own* MAC address as if it belonged to End System Z. End System Y receives the ARP reply and saves the MAC address of Router X in its ARP cache in the entry for End System Z. When End System Y must communicate with End System Z, it checks the ARP cache, finds the MAC address of Router X, and sends the frame directly without using ARP requests. Router X receives the traffic from End System Y and forwards it to End System Z on the other LAN.

The Hello protocol is a network-layer protocol that enables network devices to identify one another and indicate that they are still functional. When a new end system powers up, for example, it broadcasts Hello messages onto the network. Devices on the network then return Hello replies, and Hello messages are also sent at specific intervals to indicate that they are still functional. Network devices can learn the MAC addresses of other devices by examining Hello-protocol packets.

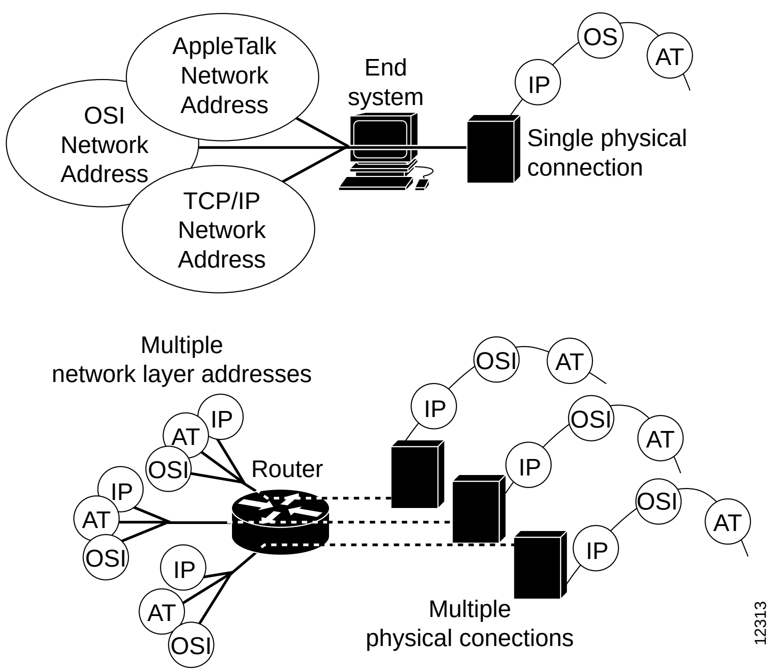
Three protocols use predictable MAC addresses. In these protocol suites, MAC addresses are predictable because the network layer either embeds the MAC address in the network-layer address or uses an algorithm to determine the MAC address. The three protocols are Xerox Network Systems (XNS), Novell Internetwork Packet Exchange (IPX), and DECnet Phase IV.

Network-Layer Addresses

A network-layer address identifies an entity at the network layer of the OSI layers. Network addresses usually exist within a hierarchical address space and sometimes are called *virtual* or *logical* addresses.

The relationship between a network address and a device is logical and unfixed; it typically is based either on physical network characteristics (the device is on a particular network segment) or on groupings that have no physical basis (the device is part of an AppleTalk zone). End systems require one network-layer address for each network-layer protocol they support. (This assumes that the device has only one physical network connection.) Routers and other internetworking devices require one network-layer address per physical network connection for each network-layer protocol supported. A router, for example, with three interfaces each running AppleTalk, TCP/IP, and OSI must have three network-layer addresses for each interface. The router therefore has nine network-layer addresses. Figure 1-16 illustrates how each network interface must be assigned a network address for each protocol supported.

Figure 1-16 Each network interface must be assigned a network address for each protocol supported.

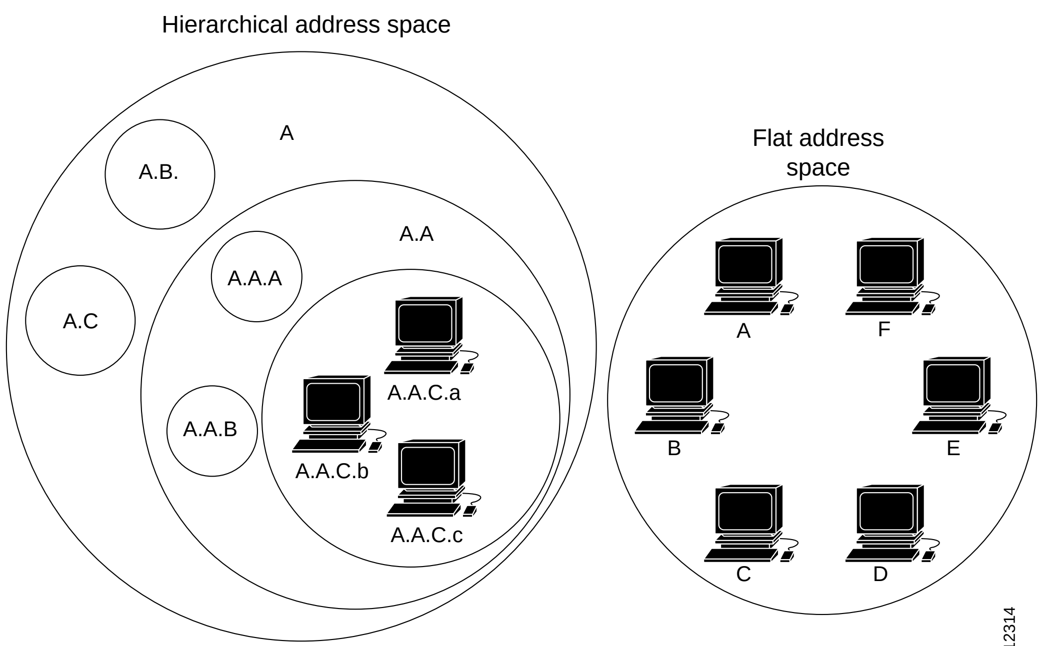


Hierarchical Versus Flat Address Space

Internetwork address space typically takes one of two forms: hierarchical address space or flat address space. A hierarchical address space is organized into numerous subgroups, each successively narrowing an address until it points to a single device (in a manner similar to street addresses). A flat address space is organized into a single group (in a manner similar to U.S. Social Security numbers).

Hierarchical addressing offers certain advantages over flat-addressing schemes. Address sorting and recall is simplified through the use of comparison operations. Ireland, for example, in a street address eliminates any other country as a possible location. Figure 1-17 illustrates the difference between hierarchical and flat-address spaces.

Figure 1-17 Hierarchical and flat address spaces differ in comparison operations.



Address Assignments

Addresses are assigned to devices as one of three types: *static*, *dynamic*, or *server* addresses. *Static addresses* are assigned by a network administrator according to a preconceived internetwork addressing plan. A static address does not change until the network administrator manually changes it. *Dynamic addresses* are obtained by devices when they attach to a network, by means of some protocol-specific process. A device using a dynamic address often has a different address each time it connects to the network. Addresses assigned by a server are given to devices as they connect to the network. Server-assigned addresses are recycled for reuse as devices disconnect. A device is therefore likely to have a different address each time it connects to the network.

Addresses Versus Names

Internetwork devices usually have both a name and an address associated with them. Internetwork names typically are location-independent and remain associated with a device wherever that device moves (for example, from one building to another). Internetwork addresses usually are location-dependent and change when a device is moved (although MAC addresses are an exception to this rule). Names and addresses represent a logical identifier, which may be a local system administrator or an organization, such as the Internet Assigned Numbers Authority (IANA).

Flow-Control Basics

Flow control is a function that prevents network congestion by ensuring that transmitting devices do not overwhelm receiving devices with data. Countless possible causes of network congestion exist. A high-speed computer, for example, may generate traffic faster than the network can transfer it, or faster than the destination device can receive and process it. The three commonly used methods for handling network congestion are buffering, transmitting source-quench messages, and windowing.

Buffering is used by network devices to temporarily store bursts of excess data in memory until they can be processed. Occasional data bursts are easily handled by buffering. Excess data bursts can exhaust memory, however, forcing the device to discard any additional datagrams that arrive.

Source-quench messages are used by receiving devices to help prevent their buffers from overflowing. The receiving device sends source-quench messages to request that the source reduce its current rate of data transmission. First, the receiving device begins discarding received data due to overflowing buffers. Second, the receiving device begins sending source-quench messages to the transmitting device at the rate of one message for each packet dropped. The source device receives the source-quench messages and lowers the data rate until it stops receiving the messages. Finally, the source device then gradually increases the data rate as long as no further source-quench requests are received.

Windowing is a flow-control scheme in which the source device requires an acknowledgment from the destination after a certain number of packets have been transmitted. With a window size of three, the source requires an acknowledgment after sending three packets, as follows. First, the source device sends three packets to the destination device. Then, after receiving the three packets, the destination device sends an acknowledgment to the source. The source receives the acknowledgment and sends three more packets. If the destination does not receive one or more of the packets for some reason, such as overflowing buffers, it does not receive enough packets to send an acknowledgment. The source then retransmits the packets at a reduced transmission rate.

Error-Checking Basics

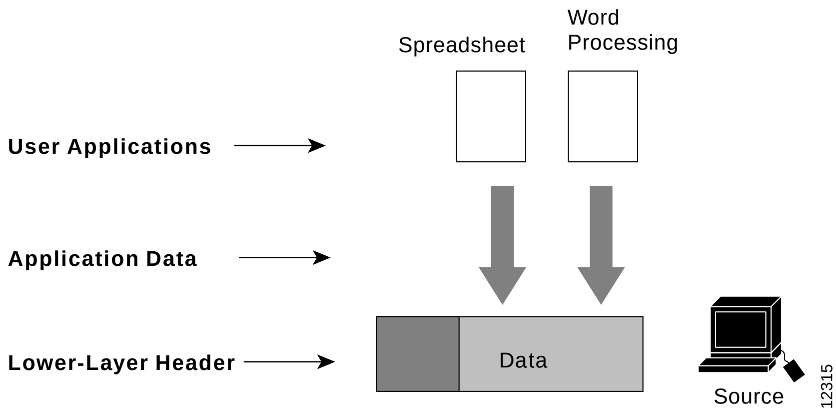
Error-checking schemes determine whether transmitted data has become corrupt or otherwise damaged while traveling from the source to the destination. Error-checking is implemented at a number of the OSI layers.

One common error-checking scheme is the cyclic redundancy check (CRC), which detects and discards corrupted data. Error-correction functions (such as data retransmission) are left to higher-layer protocols. A CRC value is generated by a calculation that is performed at the source device. The destination device compares this value to its own calculation to determine whether errors occurred during transmission. First, the source device performs a predetermined set of calculations over the contents of the packet to be sent. Then, the source places the calculated value in the packet and sends the packet to the destination. The destination performs the same predetermined set of calculations over the contents of the packet and then compares its computed value with that contained in the packet. If the values are equal, the packet is considered valid. If the values are unequal, the packet contains errors and is discarded.

Multiplexing Basics

Multiplexing is a process in which multiple data channels are combined into a single data or physical channel at the source. Multiplexing can be implemented at any of the OSI layers. Conversely, demultiplexing is the process of separating multiplexed data channels at the destination. One example of multiplexing is when data from multiple applications is multiplexed into a single lower-layer data packet. Figure 1-18 illustrates this example.

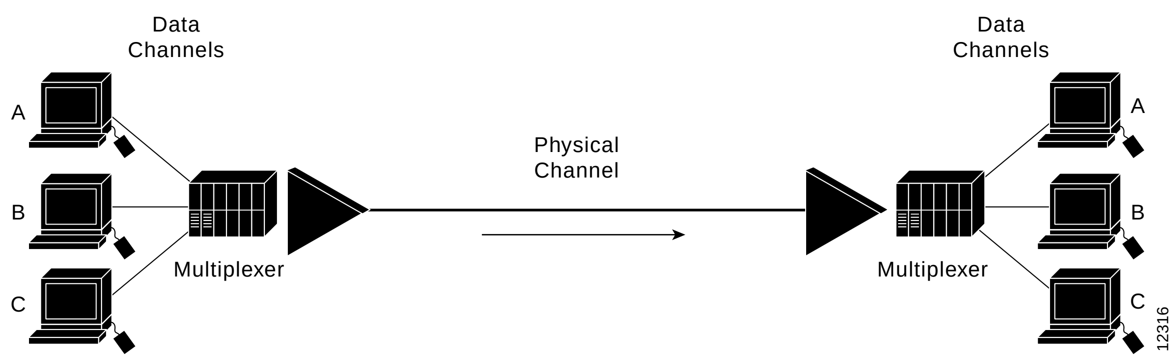
Figure 1-18 Multiple applications can be multiplexed into a single lower-layer data packet.



Another example of multiplexing is when data from multiple devices is combined into a single physical channel (using a device called a multiplexer). Figure 1-19 illustrates this example.

A multiplexer is a physical-layer device that combines multiple data streams into one or more output channels at the source. Multiplexers demultiplex the channels into multiple data streams at the remote end and thus maximize the use of the bandwidth of the physical medium by enabling it to be shared by multiple traffic sources.

Figure 1-19 Multiple devices can be multiplexed into a single physical channel.



Some methods used for multiplexing data are time-division multiplexing (TDM), asynchronous time-division multiplexing (ATDM), frequency-division multiplexing (FDM), and statistical multiplexing.

In TDM, information from each data channel is allocated bandwidth based on preassigned time slots, regardless of whether there is data to transmit. In ATDM, information from data channels is allocated bandwidth as needed, by using dynamically assigned time slots. In FDM, information from each data channel is allocated bandwidth based on the signal frequency of the traffic. In statistical multiplexing, bandwidth is dynamically allocated to any data channels that have information to transmit.

Standards Organizations

A wide variety of organizations contribute to internetworking standards by providing forums for discussion, turning informal discussion into formal specifications, and proliferating specifications after they are standardized.

Most standards organizations create formal standards by using specific processes: organizing ideas, discussing the approach, developing draft standards, voting on all or certain aspects of the standards, and then formally releasing the completed standard to the public.

Some of the best-known standards organizations that contribute to internetworking standards include:

- *International Organization for Standardization (ISO)*—ISO is an international standards organization responsible for a wide range of standards, including many that are relevant to networking. Their best-known contribution is the development of the OSI reference model and the OSI protocol suite.
- *American National Standards Institute (ANSI)*—ANSI, which is also a member of the ISO, is the coordinating body for voluntary standards groups within the United States. ANSI developed the Fiber Distributed Data Interface (FDDI) and other communications standards.
- *Electronic Industries Association (EIA)*—EIA specifies electrical transmission standards, including those used in networking. The EIA developed the widely used EIA/TIA-232 standard (formerly known as RS-232).
- *Institute of Electrical and Electronic Engineers (IEEE)*—IEEE is a professional organization that defines networking and other standards. The IEEE developed the widely used LAN standards IEEE 802.3 and IEEE 802.5.
- *International Telecommunication Union Telecommunication Standardization Sector (ITU-T)*—Formerly called the Committee for International Telegraph and Telephone (CCITT), ITU-T is now an international organization that develops communication standards. The ITU-T developed X.25 and other communications standards.
- *Internet Activities Board (IAB)*—IAB is a group of internetwork researchers who discuss issues pertinent to the Internet and set Internet policies through decisions and task forces. The IAB designates some Request For Comments (RFC) documents as Internet standards, including Transmission Control Protocol/Internet Protocol (TCP/IP) and the Simple Network Management Protocol (SNMP).