

CHAPTER 2

Protocol Characteristics Overview

This chapter compares connection-oriented and connectionless protocols and identifies commonly used protocols as being one or the other. It also focuses on the operation of Ethernet/IEEE 802.3, Token Ring/IEEE 802.5, Fiber Data Distributed Interface (FDDI), and serial protocols such as High-Level Data Link Control (HDLC), as well as routed protocols and routing protocol families such as Transmission Control Protocol/Internet Protocol (TCP/IP), Novell Internetwork Packet Exchange (IPX), and AppleTalk.

This chapter is divided into two main sections:

- Basic protocol characteristics
- Detailed protocol characteristics

The first section provides an introduction to protocols as a foundation or core set of communication elements. The second section provides in-depth details on the data link, network, and upper-layer protocols.

To become a successful troubleshooter, it is important that you understand how protocols behave under normal circumstances. This will help you recognize abnormal behavior when it occurs. Understanding the complexities of protocols will help you understand what is likely to break or not perform optimally.

Most protocols have common characteristics. For example, they are either connection oriented or connectionless and reliable or unreliable. Understanding in theory how these functions work helps you understand all connection-oriented protocols. Even though you may not currently work with some of the protocols in this chapter, in the future that might change. New protocols might be added to your internetwork or you might move to another job that requires you to understand new protocols. Consider, for example, the number of IPX NetWare networks that are now pure TCP/IP.

Basic Protocol Characteristics

As previously mentioned, protocols can be distinguished as either connection oriented or connectionless and reliable or unreliable.

A reliable protocol is a protocol that has error correction, flow control, and retransmission functionality built into it. Unreliable protocols depend on higher-layer protocols to provide reliability. An example of sending data reliably is sending a letter via certified mail with a return receipt. Unreliable data delivery would simply be dropping the letter in the mail box and having best-effort delivery.

Most connection-oriented protocols are reliable, and most connectionless protocols are unreliable, but there are exceptions. For example, Frame Relay is a connection-oriented protocol. It requires virtual circuit connectivity before data can be sent, but there are no reliability mechanisms built into Frame Relay. Likewise, Open Shortest Path First (OSPF) is connectionless. When OSPF sends out routing updates, it sends multicast packets (which are by definition connectionless), yet it still expects to see acknowledgments from its neighbors.

All the connection-oriented protocols discussed in this section are reliable protocols.

An entity can transmit data to another entity in an unplanned fashion and without prior coordination. This is known as *connectionless data transfer*, in which each packet stands alone. The receiving software entity that is part of the Open System Interconnection (OSI) reference model protocol stack sees that the packets received are complete, but a higher-layer application must put them in the proper order, manage timeout counters, and request the retransmission of missing packets.

Connection-Oriented Services

A connection-oriented data transfer is conceptually similar to a telephone call in which the caller initiates the call, knows when the connection is made because a person is heard at the other end of the line, and hangs up and terminates the call when the information exchange is complete.

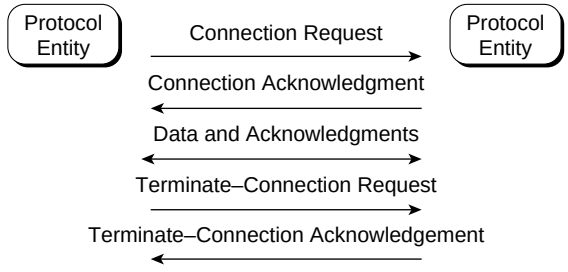
A connection-oriented protocol has some method for connection establishment, flow control, error control, and session termination. TCP and Asynchronous Transfer Mode (ATM) are examples of connection-oriented protocols.

A connection-oriented protocol ensures that packets are in order and manages timeout counters. That connection-oriented protocol also requests retransmission of missing packets.

With connection-oriented data transfer, a logical association, or connection, is established between protocol entities. As shown in Figure 2-1, this type of data transfer has a clearly distinguishable lifetime, consisting of three distinct phases:

- Connection establishment
- Data transfer
- Connection termination

Figure 2-1 *The connection establishment phase (which occurs first in connection-oriented services) is often referred to as a* handshake.



Depending on the protocol, a specific protocol entity might always establish and terminate the connection. However, in some protocols, either side can initiate these functions.

Connection-oriented data transfer is preferable to connectionless data transfer when stations require a lengthy exchange of data or certain details of their protocol must be worked out dynamically. TCP is an example of a connection-oriented protocol, and Asynchronous Transfer Mode (ATM) is another.

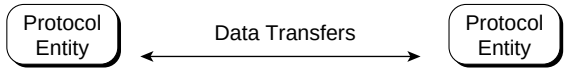
When troubleshooting connection-oriented protocols, check whether there are multiple retransmissions of segments of data. If there are, determine why the higher-layer protocol is requesting them. Verify that sequence numbers, acknowledgments, window sizes, and other parameters associated with this type of protocol are appropriate and are being incremented or managed correctly.

NOTE Refer to the section "Detailed Protocol Characteristics" later in this chapter for more specific examples of connection-oriented communications.

Connectionless Services

With connectionless services, there is no connection setup between the two communicating protocol entities (see Figure 2-2). Each data unit is transmitted independently of previous and subsequent data units. A connectionless data transfer is conceptually similar to a set of postcards, each postcard handled separately. So, too, each packet stands alone. An entity can transmit data to another entity in an unplanned fashion and without prior coordination.

Figure 2-2 *Connectionless communications, shown here, are much simpler than the three-part connection-oriented communications. Notice that there is no connection made before data transfer begins.*



The receiving software entity that is part of the OSI reference model protocol stack sees that the packets received are complete, but a higher-layer application must put them in the proper order and request the retransmission of missing packets. A connectionless data transfer is efficient, is simple to implement, and has relatively low demands for network traffic.

When troubleshooting connectionless data transfer, look for problems where data is not acknowledged, where errors in the data are not reported to the sender, where data may arrive out of order (because it is not sequenced), and where there is no flow control.

When connectionless protocols are used at lower layers, upper layers are usually connection oriented to ensure reliable transfer of data. For example, in the TCP/IP protocol suite, IP is a connectionless Layer 3 protocol, but TCP is a connection-oriented protocol up at Layer 4.

Sample Connection Sequences

After you have identified and corrected any physical-layer or data link-layer troubleshooting targets, you can proceed to consider higher-level troubleshooting targets.

Failures at these lower layers propagate to the higher layers. For example, a bad Ethernet data link affects the connected router Ethernet interface, which in turn affects the TCP/IP routing tables, which in turn affects the connection made by an application.

Sometimes there can be multiple potential causes for a network problem, especially where there are multiple technologies, protocols, and data-link hops that the application will use.

NOTE For the protocol suites covered next, the sending host on a network cannot get higher-layer data to a destination side until a network connection is made.

If users are having trouble making connections on the network, check the specific network and upper-layer protocol troubleshooting targets.

At the network layer are two types of protocols. Rout*ed* protocols send user data between hosts. Rout*ing* protocols communicate information between routers about the network paths to use. The connection sequences that protocol suites use involve both of these types of protocols as well as higher-layer protocols within the protocol suites. This chapter describes the following concepts:

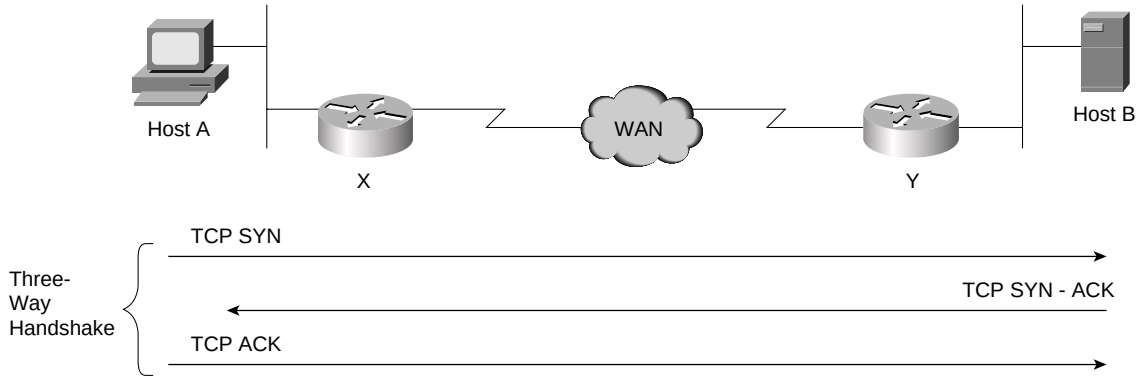
- TCP/IP using Address Resolution Protocol (ARP) and synchronization (SYN) packets
- Novell NetWare Core Protocol (NCP) using Routing Information Protocol (RIP), Service Advertising Protocol (SAP), and Get Nearest Server (GNS) requests
- AppleTalk using several of the protocols in its protocol suite, including Routing Table Maintenance Protocol (RTMP), Zone Information Protocol (ZIP), Name Binding Protocol (NBP), and AppleTalk Transaction Protocol (ATP).

Let's take a look at the different connection sequences that occur with TCP/IP, NetWare, and AppleTalk. Each protocol set has a unique connection establishment routine.

The TCP Connection Sequence

TCP is an example of a connection-oriented protocol. The TCP connection establishment sequence is often called a *three-way handshake*, as shown in Figure 2-3.

Figure 2-3 *The three-way handshake must be completed successfully before data can be exchanged.*



A three-way handshake consists of three stages:

- The host that wants to initiate the session sends a TCP synchronization (SYN) packet.
- The receiving host acknowledges the SYN and in the same packet sends its own SYN.
- The original host acknowledges the receipt of the SYN from the other host.

Figure 2-3 also shows Host A sending an ARP request and receiving a reply. The ARP frames are not part of a TCP session establishment, but are necessary for Host A to reach Host B.

Until Host A receives an ARP reply, it only knows the IP address of Host B. A MAC address must be determined in order to transmit data. Our example assumes that Host A's software sends ARP frames even when the destination is not local and the router responds to ARP frames for hosts on the other side. (The router runs proxy ARP, which is the default for Cisco Systems routers.)

A different, but correct, interpretation of the ARP frames shown in Figure 2-3 is that Host A is configured with Router X as its default router and must send an ARP to learn the MAC address of Router X.

ARP establishes correspondences between network addresses (an IP address, for example) and LAN hardware addresses (Ethernet addresses). A record of each correspondence is kept in a cache for a predetermined amount of time and then discarded.

When you are troubleshooting, use the **show ip arp** command to check for any anomalies (such as duplicate routes) and to see whether the host(s) in question is appearing in the ARP table.

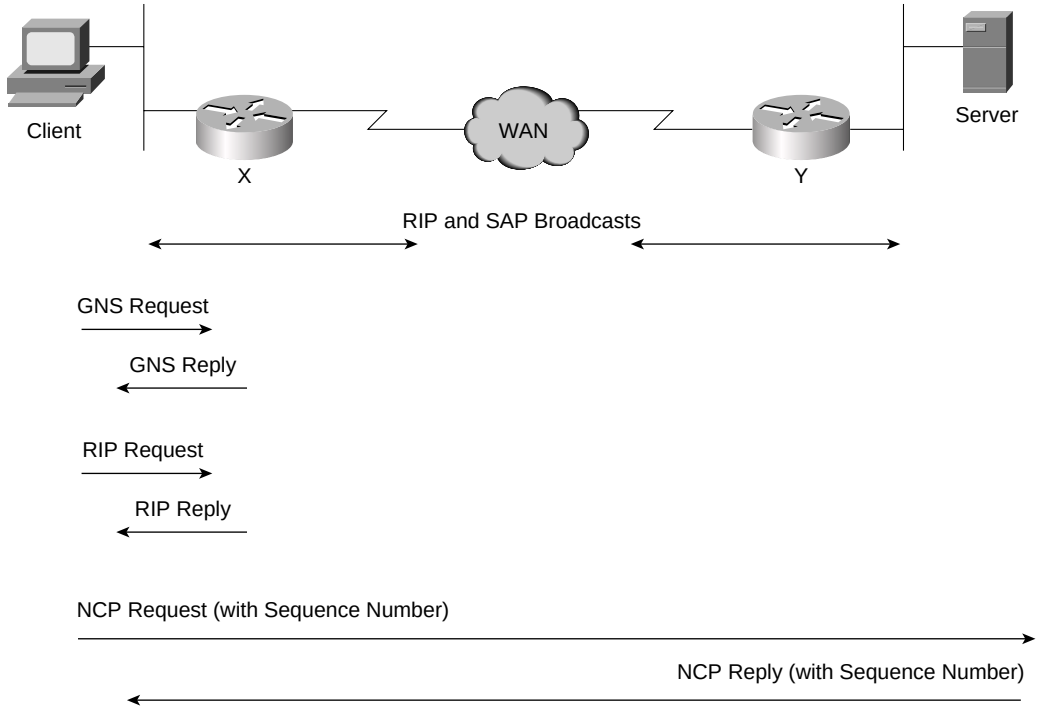
NOTE See the section "Detailed Protocol Characteristics" later in this chapter for more information on the TCP/IP protocol suite.

The NetWare Connection Sequence

NetWare has two types of connection sequences: Sequenced Packet Exchange (SPX) and NCP connections. NetWare uses SPX to provide transport-layer, connection-oriented services. In a manner similar to TCP, an SPX client requests synchronization with a peer SPX device. An acknowledgment reply completes SPX's two-way handshake process. SPX ensures the guaranteed delivery of data in order, just as TCP does.

Novell's NCP is also a connection-oriented protocol used by file servers and clients. Before a client can log in to an NCP file server and start requesting or sending data, the client first broadcasts a SAP GNS request, as shown in Figure 2-4.

Figure 2-4 *NetWare clients must perform service discovery before connecting to a server.*



A router or server can respond to the request. Routers learn about services by listening to SAP broadcasts from servers and other routers. After a client learns about a server, it sends a RIP request to find a route to the server. Finally, the client can send NCP requests to log in and send and retrieve data.

NCP requests and replies contain a sequence number field to ensure that packets arrive in order and that no packets are missing.

NetWare versions 3.x and 4.x use IPX for Layer 3 connectivity, addressing, and routing. Interconnecting NetWare clients on LANs running IPX face a variety of problems that can impair connecting to a server over a routed Internet. These can include mismatched network numbers and encapsulation types. It is important to proceed with a systematic troubleshooting method to obtain facts to help you isolate the problem or problems. Among the several Cisco Internetwork Operating System (IOS) tools to help you isolate problems is the **show ipx interface** command. This command can be used to verify network and encapsulation configurations as well as the status of the router interfaces. The **show ipx traffic** command can be used for verifying the sending and receiving of packets from the protocols described previously in this chapter.

The interfaces should indicate that the interface is up and that the protocol is up.

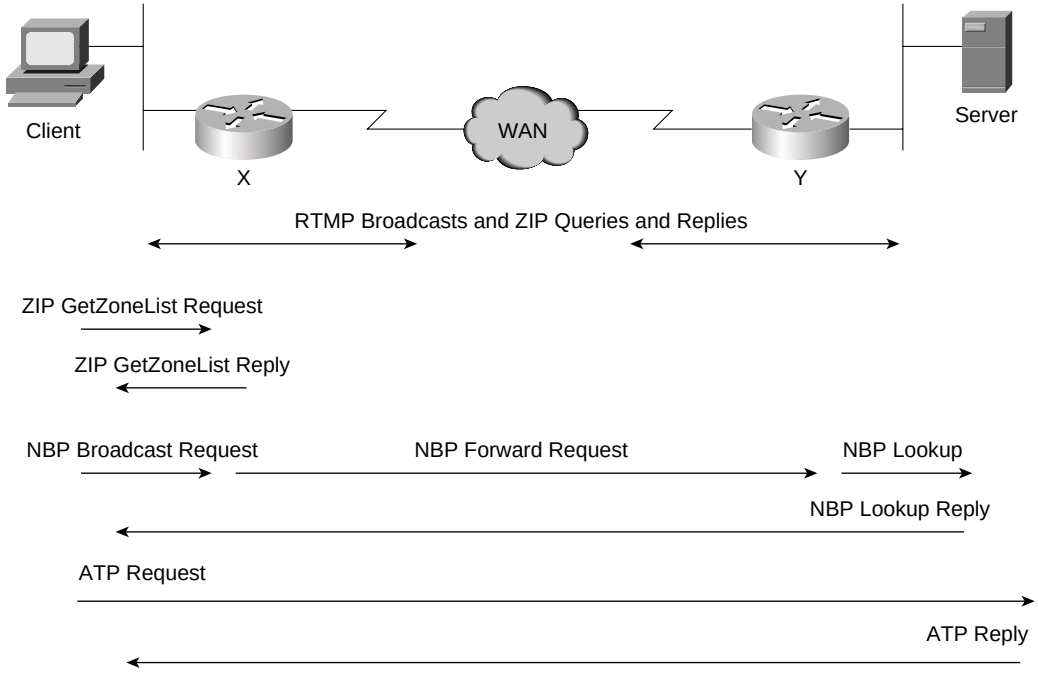
NOTE See the section "Detailed Protocol Characteristics" later in this chapter for more information on the Novell NetWare protocol suite.

The AppleTalk Connection Sequence

ATP is a transport-layer, connection-oriented protocol that is used by both AppleTalk Filing Protocol (AFP) and Printer Access Protocol (PAP). It provides reliable transport of data when a client attaches to an AppleShare file server or printer.

Before a client can attach to a file server (or printer), it must locate the server through a process called *service discovery.* A Macintosh computer sends a GetZoneList request to the local router to fill the Chooser window with zone names. When the user clicks on a zone name and type of service, the Macintosh sends an NBP broadcast request, as shown in Figure 2-5.

Figure 2-5 *AppleTalk uses a combination of protocols to locate and connect to a service.*



The router looks in its zone information table to determine which networks are in the requested zone. The router forwards the request to a router for each network in the zone. The receiving router then sends an NBP lookup request onto its local network. All servers of the requested service type respond to the original requesting Macintosh. (The address of the original requesting Macintosh is carried inside the NBP frames.)

When the client Macintosh knows how to reach a file server, it sets up a connection using ATP and AFP. It may also send an Echo frame in order to determine how much time it takes to reach the server. This information can be used for setting ATP timeout values.

A variety of problems can block access to servers and services on an AppleTalk network. For example, the Cisco IOS software command **show appletalk traffic** provides a good high-level overview of the AppleTalk activity on the router.

Another source of troubleshooting information is to look for an AppleTalk configuration mismatch. A mismatch occurs when the following rule of AppleTalk is violated: "All routers on a given cable must agree on the configuration of that cable (meaning that all must have matching network numbers, cable ranges, zone names, or zone lists)." The router declares a port configuration mismatch in the output of the **show appletalk interface** command.

NOTE See the following section "Detailed Protocol Characteristics" for more information on the AppleTalk protocol suite.

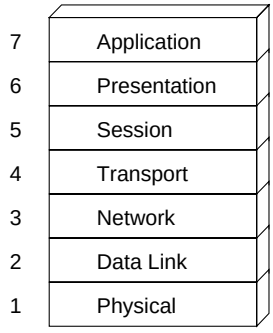
Detailed Protocol Characteristics

Although this section starts in quite general terms, it quickly delves into the complexities of the protocol sets and operation of 802.3, 802.5, FDDI, Point-to-Point Protocol (PPP), Synchronous Data Link Control (SDLC), Frame Relay, ISDN, TCP/IP, NetWare IPX/SPX, and AppleTalk networks.

Let's start this discussion with the infamous OSI reference model. When discussing communication between or among computer systems, it is helpful to adopt a common set of standards or conventions. The International Organization for Standardization (ISO) developed the OSI reference model as a framework for defining standards for linking heterogeneous computers.

The OSI reference model includes a vertical set of layers. Each layer performs the functions required to communicate with the associated layer of another system. Each layer relies on the next lower layer to provide a set of services and at the same time provides services to the next higher layer. ISO finalized a reference model that has seven layers, and then defined each layer and the services to be performed there. Figure 2-6 shows the seven layers of the OSI model.

Figure 2-6 *The OSI model has seven layers.*



The OSI layers and their functions are as follows:

- Layer 7, the application layer, specifies the user interface and the application program interface and is concerned with the meaning of data, job management, and data exchange.
- Layer 6, the presentation layer, accepts data from the application layer and negotiates data syntax, representation, compression, and encryption with peer systems.
- Layer 5, the session layer, adds control mechanisms such as checkpoints, terminations, and restarts to establish, maintain, and synchronize communication between applications.
- Layer 4, the transport layer, provides end-to-end accountability and ensures reliable data delivery using acknowledgments, sequence numbers, and flow control mechanisms.
- Layer 3, the network layer, specifies network routing (using symbolic addressing) and communication between network segments.
- Layer 2, the data link layer, is responsible for data transfer over a communication channel by consolidating data into frames, detecting and recovering from transmission errors, and defining physical device addresses.
- Layer 1, the physical layer, deals with the mechanical, electrical, functional, and procedural interface between user equipment and the network communications system—in other words, “getting bits onto the wire.”

The intent of the OSI model is that protocols be developed to perform the functions of each layer to enable communication between corresponding (peer) entities at the same layer in two different systems.

Regardless of the nature of the applications or entities attempting to exchange data, there is usually a requirement that data be exchanged reliably; that is, all the data should arrive at the destination process in the same order in which it was sent.

Connection-Oriented Versus Connectionless Protocols

In order for applications or entities in different systems to communicate with each other, some form of protocol must define a set of rules (that is, the syntax and semantics) governing the exchange of data between the two.

A fundamental aspect of any communications architecture is that one or more protocols operate at each layer of the architecture, and that two peer protocols at the same layer but in different systems cooperate to achieve the communication function.

A set of functions forms the basis of most protocols. These functions might be present in protocols at the different conceptual layers. Although we are looking at the concept of protocols within the seven-layer OSI model, some vendors' protocols are proprietary. However, these protocols can be viewed as fitting within one or more of the OSI model's layers.

Protocol functions can be grouped into the following categories:

- Data segmentation and reassembly
- Data encapsulation
- Connection control
- Ordered delivery of data
- Flow control
- Error control
- Multiplexing

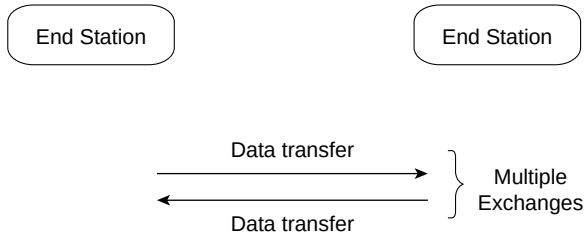
This section focuses on aspects of a protocol's connection characteristics that cover connection control, ordered delivery, flow control, and error control. Earlier in this chapter we talked about two types of protocols—connection oriented and connectionless. The connection-oriented protocols are likened to a telephone call, where we make certain there is a user at the other end before sending data, whereas connectionless services are more like a postcard in the mail, with no guarantee of delivery.

Connectionless Data Transfer

An entity can transmit data to another entity in an unplanned fashion and without prior coordination. This is known as *connectionless data transfer*, in which each packet stands alone. The receiving software entity that is part of the OSI protocol stack sees that the packets received are complete, but a higher-layer application must put them in the proper order and request the retransmission of missing packets. Although this mode can be useful, it is less common than connection-oriented transfer.

With connectionless data transfer, no logical connection is established and each PDU is transmitted independently of any previous or subsequent data unit. Figure 2-7 illustrates this concept.

Figure 2-7 *Connectionless data transfer has low overhead compared to connection-oriented services.*



The stations make an *a priori* agreement with each other, but not with the lower-layer services, about transmission of protocol data units (PDUs).

Because no connection is established, each data unit is transmitted with a single service access, and this access must contain all the information necessary for the unit to be delivered to its destination (destination address, services required, and so on). As a result of this single-service access, no negotiation of parameters occurs. Also, because the PDUs are transported by the service provider, which knows no relationship between current, previous, or subsequent data units, they do not include ordered delivery, flow control, or error control.

The following are examples of connectionless protocols:

- IEEE 802.2 (often referred to as Logical Link Control [LLC]) offers three types of service, two of which are connectionless: Type 1 provides unacknowledged connectionless service, and Type 3 provides acknowledged connectionless service.
- OSI offers both a connectionless and connection-oriented network-layer service. The connectionless service is described in ISO 8473 (usually referred to as Connectionless Network Protocol [CLNP]).
- Internet Protocol (IP) is the network-layer connectionless protocol in the TCP/IP suite. User Datagram Protocol (UDP) is the connectionless transport-layer protocol in the TCP/IP suite.
- AppleTalk's primary network-layer protocol is Datagram Delivery Protocol (DDP). DDP provides connectionless service between network sockets.
- Novell's IPX is a connectionless network-layer datagram protocol for NetWare networks functioning on IPX/SPX. NetWare 5 can run directly over TCP/IP and replaces IPX functionality with UDP in that case.
- DECnet uses CLNP and Connectionless Network Service (CLNS) for connectionless service.
- Fast Sequenced Transport (FST) is a connectionless, sequenced transport protocol that runs on top of IP. FST uses the IP header to implement sequencing without violating the IP specification. FST transports remote source-route bridging (RSRB) packets to peers without TCP or UDP header or processor overhead.

Connection-Oriented Data Transfer

As mentioned earlier in this chapter, connection-oriented data transfer is preferable if stations require a lengthy exchange of data or certain details of their protocol must be worked out dynamically.

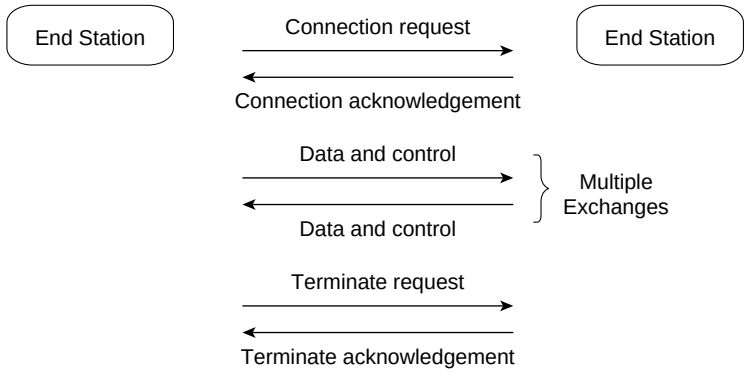
With connection-oriented data transfer, a logical association, or *connection*, is established between the entities. This type of data transfer has a clearly distinguishable lifetime, consisting of three distinct phases:

- Connection establishment
- Data transfer
- Connection termination

Connection establishment begins when one of the stations issues a connection request to the other. The two end stations might be the only devices involved, or a central authority might be involved as well.

As shown in Figure 2-8, during the connection establishment phase, the two end stations must make an agreement with each other and with the underlying protocol layer to exchange data. While the connection is in place, the parties agree on the acceptance of every error-free data unit that is transferred. There might be negotiation of parameters and options that will affect the subsequent data transfer (such as the maximum data or window size). The negotiation might result in the rejection of the connection, if one of the parties is unable to provide the connection configuration that has been requested.

Figure 2-8 *Connection-oriented data transfer has three phases.*



After the logical connection has been established, the data transfer phase is entered, when data and any relevant control information such as flow control or error control are exchanged between the end stations. A connection identifier can be used to identify that data and control information are associated with the logical connection.

A logical connection also allows for the use of sequencing, which means that each PDU is sequentially numbered. The end stations maintain a list of locally generated outgoing numbers and incoming numbers. Sequencing allows for ordered delivery, flow control, and error control. These concepts can be explained as follows:

- Ordered delivery—In a connection-oriented environment, because each PDU is identified by a connection identifier and sequence number, correct ordering of PDUs can be maintained. A unique sequential numbering scheme is determined during the connection establishment phase. The receiving station can use the sequential numbering scheme to easily reorder received PDUs, even if they arrive out of order.
- Flow control—With flow control, the transmitting station does not send data or information faster than the receiving station or any intermediate device can handle it. Again, with a connection-oriented protocol, the sequence numbers provide this function. The fundamental approach used with most protocols is a windowing technique in which the receiving station specifies how much buffer space is available for incoming PDUs. The sender sends that amount of data in PDUs and then waits to receive acknowledgment from the receiving station that it has received all or some of these PDUs and is ready to receive more.
- Error control—An end station detects damaged PDUs by performing a calculation on the bits received. If a PDU is damaged or lost, the end station requests retransmission of the affected PDU based on its record of sequence numbers.

The final phase of a connection-oriented operation is the termination of the connection. This can be initiated by any of the parties involved. Either of the end stations or the central authority, if one is involved, can send a termination request.

The following are examples of connection-oriented protocols:

- IEEE 802.2 (LLC) offers three types of service, one of which, Type 2, is connection oriented. LLC Type 2 (LLC2) is a connection-oriented OSI data link layer protocol that is widely used in local-area network (LAN) environments, particularly among IBM communication systems connected by Token Ring.
- ATM is a connection-oriented environment. All traffic to or from an ATM network is prefaced with a virtual path identifier (VPI) and virtual channel identifier (VCI). A VPI/VCI pair is considered a single virtual circuit (VC). Each VC is a private connection to another node on the ATM network. Each ATM node must establish a separate connection to every other node in the ATM network that it wants to communicate with.

- OSI offers a connection-oriented service sometimes called Connection-Mode Network Service (CMNS) as described in ISO 8208 and ISO 8878. (ISO 8208 is the X.25 packet-level protocol [PLP], sometimes referred to as Connection-Oriented Network Protocol [CONP]. ISO 8878 describes how to use ISO 8208 to provide OSI connection-oriented service.) OSI uses the X.25 packet-level protocol over IEEE 802 LANs for connection-oriented data movement and error indications.

  As with the OSI network layer, the OSI transport layer provides connection-oriented services. There are five connection-oriented OSI transport protocols: TP0, TP1, TP2, TP3, and TP4.

- In the TCP/IP protocol suite, TCP is a reliable, full-duplex, connection-oriented protocol that specifies the format of data and control packets that two computer systems exchange to transfer data.
- Novell's SPX is a connection-oriented NetWare transport protocol. Novell derived this protocol from the Xerox Network Systems (XNS) Sequenced Packet Protocol (SPP). As with TCP and many other transport protocols, SPX is a reliable, connection-oriented protocol that supplements the datagram service provided by Layer 3 protocols.
- AppleTalk supports several upper-layer protocols that are connection oriented. ATP is a popular transport-layer protocol that allows reliable request-response exchanges between two socket clients.
- DECnet's network layer uses the X.25 packet-level protocol, which is also known as X.25 Level 3, and CONP.

You will find that all protocol suites offer a combination of connection-oriented and connectionless services. If you spend most of your time learning one protocol, you will be pleasantly surprised to learn how fast you can pick up another protocol suite's connectionless and connection-oriented transport characteristics.

The next section discusses the OSI data link layer specifically. It's important to spend time focusing on your network's Layer 2 features in order to quickly recognize Layer 2 faults and misconfigurations.

OSI Layer 2 Characteristics

The OSI Layer 2, the data link layer, is responsible for arranging data into frames that are locally addressed. Specifically, this section covers LAN and WAN Layer 2 framing, such as Ethernet/IEEE 802.3, Token Ring/802.5, FDDI, and PPP.

Ethernet/IEEE 802.3

This section focuses on the most popular data link layer protocols: Ethernet/IEEE 802.3, Token Ring, FDDI, PPP, SDLC, X.25, Frame Relay, and ISDN.

Ethernet was developed by Xerox Corporation's Palo Alto Research Center (PARC) in the 1970s. Digital Equipment Corporation, Intel Corporation, and Xerox Corporation jointly developed and released an Ethernet specification (Version 2.0) that is substantially compatible with IEEE 802.3. IEEE 802.3 was first approved by the Institute of Electrical and Electronic Engineers (IEEE) board in 1983. Together, Ethernet and IEEE 802.3 currently maintain the greatest market share of any LAN protocol. Today, the term *Ethernet* is often used to refer to all carrier-sense multiple access collision detection (CSMA/CD) LANs that generally conform to Ethernet specifications, including IEEE 802.3.

Ethernet was developed to fill the middle ground between long-distance, low-speed networks and specialized computer-room networks carrying data at high speeds for very limited distances. Ethernet is well suited to applications where a local communication medium must carry bursty, occasionally heavy traffic at high peak data rates.

Ethernet and IEEE 802.3 specify similar technologies. Both are CSMA/CD LANs. Stations on a CSMA/CD LAN can access the network at any time. Before sending data, CSMA/CD stations listen to the network to see if it is already in use. If it is, the station that wants to transmit waits, in a sort of "deferral process." If the network is not in use, the station transmits. A collision occurs when two stations listen for network traffic, hear none and transmit simultaneously. In this case, both transmissions are damaged and the stations must retransmit later. A backoff algorithm determines when the colliding stations retransmit. CSMA/CD stations can detect collisions, so they know when they must retransmit.

Both Ethernet and IEEE 802.3 hubbed and repeated LANs are broadcast networks. In other words, all stations see all frames, regardless of whether they represent an intended destination. Each station must examine received frames to determine whether the station is a destination. If it is, the frame is passed to a higher protocol layer for appropriate processing. Layer 2 bridges and switches dynamically change this model by limiting unnecessary traffic propagation.

Differences between Ethernet and IEEE 802.3 LANs are subtle. Ethernet provides services corresponding to Layers 1 and 2 of the OSI reference model, and IEEE 802.3 specifies the physical layer (Layer 1) and only a portion of Layer 2. IEEE 802.2 specifies the remaining portion of Layer 2 (the LLC portion). Both Ethernet and IEEE 802.3 are implemented in hardware as either an interface card in a host computer or circuitry on a primary circuit board within a host computer.

IEEE 802.3 specifies several different physical layers, whereas Ethernet defines only one. Each IEEE 802.3 physical layer protocol has a name that summarizes its characteristics. The coded components of an IEEE 802.3 physical layer name are shown in Figure 2-9.

Figure 2-9 *The IEEE 802.3 physical layer name components describe its functionality.*

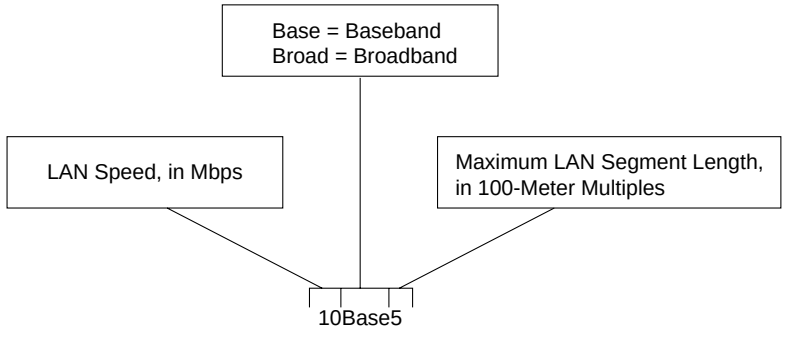


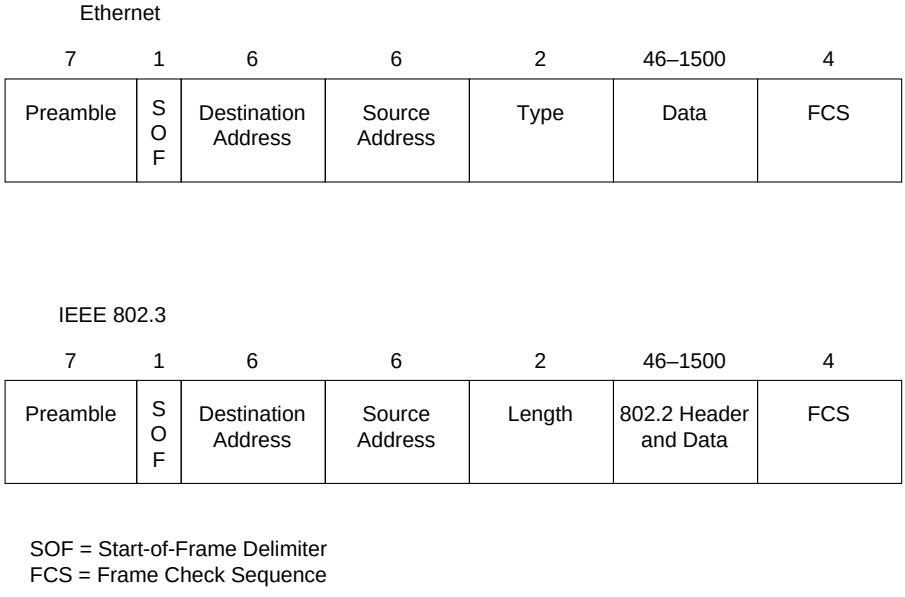
Table 2-1 is a summary of Ethernet Version 2 and IEEE 802.3 characteristics.

Table 2-1 *Ethernet Version 2 and IEEE 802.3 physical characteristics.*

Characteristic	Ethernet Value	802.3 Values					
		10Base5	10Base2	1Base5	10BaseT	100BaseT	10Broad36
Data rate (Mbps)	10	10	10	1	10	100	10
Signaling method	Baseband	Baseband	Baseband	Baseband	Baseband	Baseband	Broadband
Maximum segment length (m)	500	500	185	250	100	100	1800
Media	50-ohm coax (thick)	50-ohm coax (thick)	50-ohm coax (thin)	Unshielded twisted pair	Unshielded twisted pair	Unshielded twisted pair	75-ohm coax
Topology	Bus	Bus	Bus	Star	Star	Star	Bus

Ethernet is most similar to IEEE 802.3 10Base5. Both protocols specify a bus-topology network with a connecting cable between the end stations and the actual network medium. In the case of Ethernet, that cable is called a *transceiver cable*. The transceiver cable connects to a transceiver device attached to the physical network medium. The IEEE 802.3 configuration is similar, except that the connecting cable is referred to as an *attachment unit interface (AUI)*, and the transceiver is called a *medium attachment unit (MAU)*. In both cases, the connecting cable attaches to an interface board (or interface circuitry) within the end station. Figure 2-10 shows the Ethernet and IEEE 802.3 frame formats.

Figure 2-10 *The type and length fields differentiate Ethernet_II and IEEE 802.3 frame formats.*



Both Ethernet and IEEE 802.3 frames begin with an alternating pattern of ones and zeros called a *preamble*. The preamble tells receiving stations that a frame is coming.

The byte before the destination address in both an Ethernet and an IEEE 802.3 frame ends with two consecutive 1 bits, which synchronize the frame reception portions of all stations on the LAN. The IEEE 802.3 specification defines this as the start-of-frame delimiter (SFD), following a 7-byte preamble, whereas the Ethernet specifications simply include this pattern at the end of the 8-byte preamble.

Immediately following the preamble/SFD in both Ethernet and IEEE 802.3 frames are the destination and source address fields. Both Ethernet and IEEE 802.3 addresses are 6 bytes long. Addresses are contained in hardware on the Ethernet and IEEE 802.3 interface cards. The first 3 bytes of the addresses are specified by the IEEE as a vendor code, and the last 3 bytes are specified by the interface card vendor. The source address is always a unicast (that is, single-node) address, and the destination address can be unicast, multicast (group), or broadcast (all nodes).

In Ethernet frames, the 2-byte field following the source address is a type field. This field specifies the upper-layer protocol to receive the data after Ethernet processing is complete.

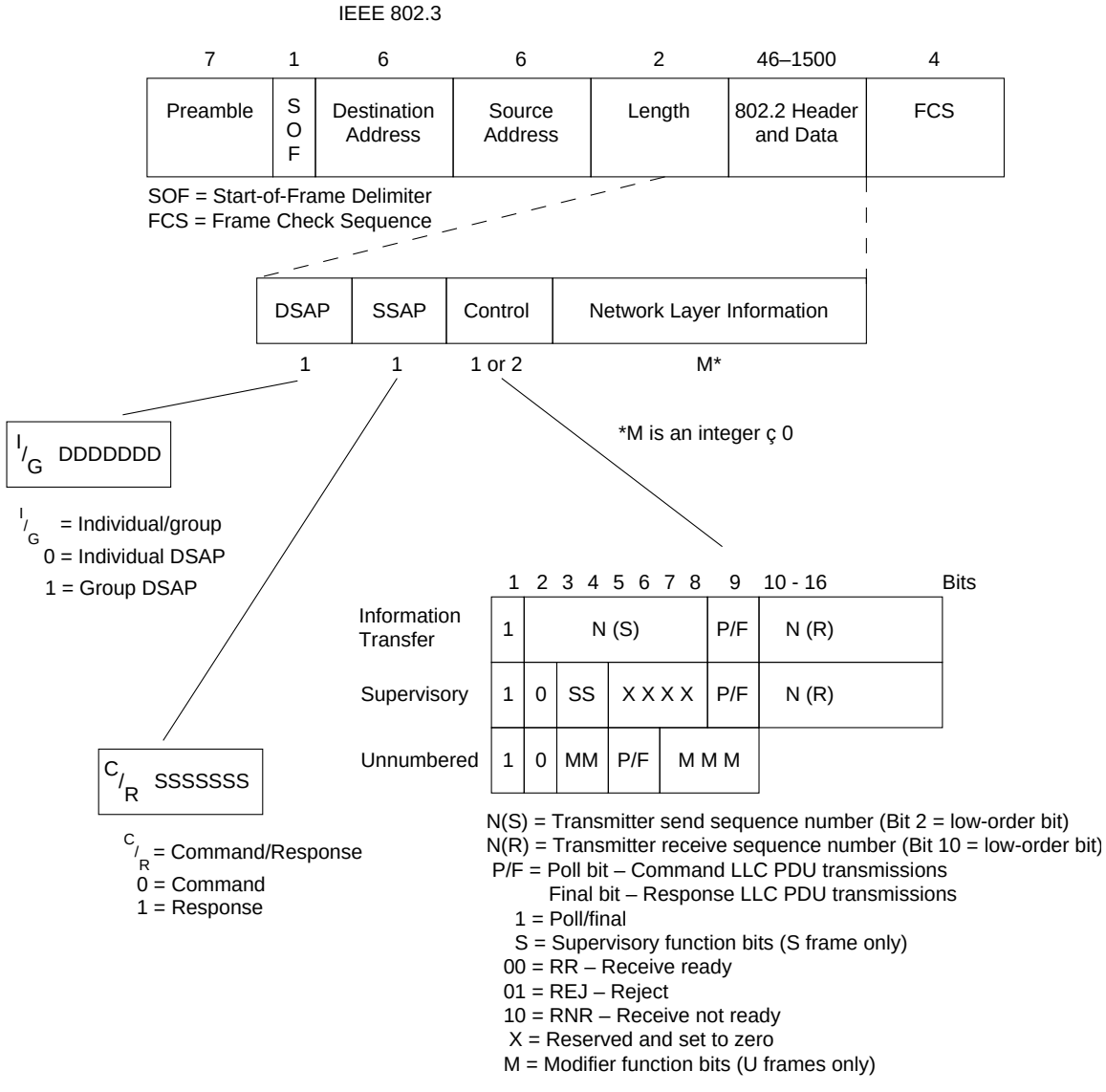
In IEEE 802.3 frames, the 2-byte field following the source address is a length field, which is a value in the range 0001 to 05DC (hexadecimal) and indicates the number of bytes of data that follow this field and precede the frame check sequence (FCS) field.

Following the type/length field is the actual data contained in the frame. After physical-layer and link-layer processing is complete, this data is eventually sent to an upper-layer protocol. In the case of Ethernet, the upper-layer protocol is identified in the type field. In the case of IEEE 802.3, the upper-layer protocol must be defined within the data portion of the frame. If data in the frame is insufficient to fill the frame to its minimum 64-byte size, padding bytes are inserted to ensure at least a 64-byte frame.

After the data field is a 4-byte FCS field that contains a cyclic redundancy check (CRC) value. The CRC is created by the sending device and recalculated by the receiving device to check for damage that might have occurred to the frame in transit.

Figure 2-11 shows the 802.2 header and data frame format. (This format is also referred to as the LLC PDU.)

Figure 2-11 *The 802.2 header and data frame format is also referred to as the LLC PDU.*



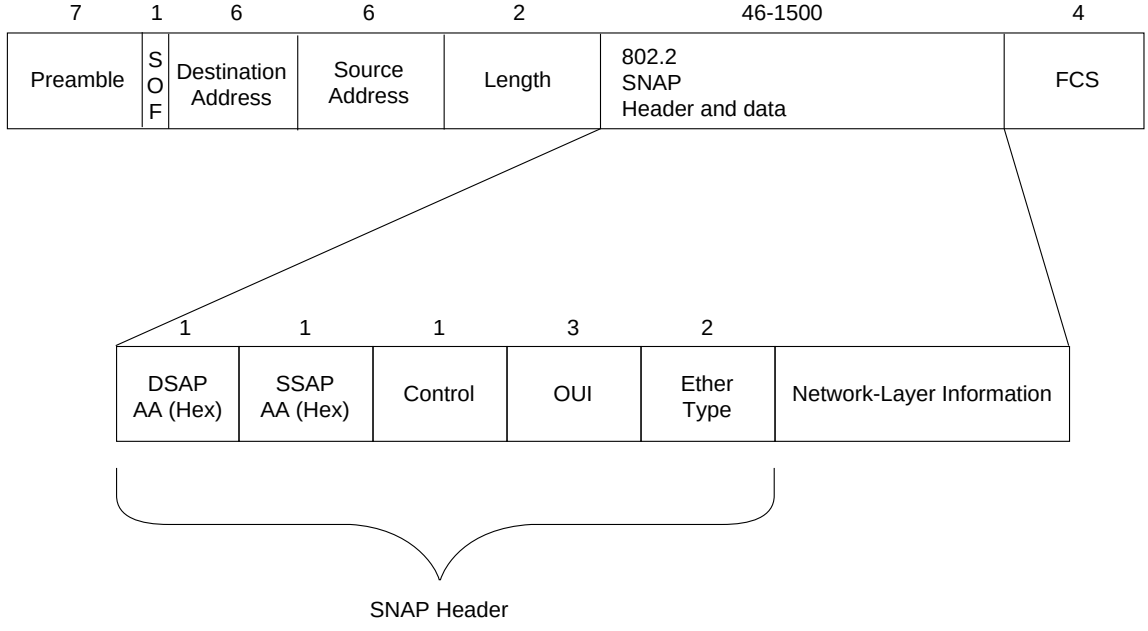
The 1-byte destination service access point (DSAP) and source service access point (SSAP) fields specify the destination and sending upper-layer protocols. The SAP fields have the same function as the type field in an Ethernet Version 2 frame.

The control field identifies the particular PDU and specifies various control functions. It is 1 or 2 bytes long, depending on the type of PDU. The PDU can be either a command or a response and is either an information transfer, a supervisory, or an unnumbered information PDU.

The network-layer information field carries the upper-layer protocol data.

An extension to the LLC was defined by the Internet community in RFC 1042, primarily to encapsulate IP datagrams and ARP requests and replies within IEEE 802.3, IEEE 802.4 (Token Bus), IEEE 802.5 (Token Ring), and FDDI networks. Doing this allows IP datagrams, which have historically been tied to Ethernet frames, to be transported across non-Ethernet networks through a Subnetwork Access Protocol (SNAP) mechanism. Figure 2-12 shows the SNAP frame format.

Figure 2-12 *SAP AA fields indicate a SNAP frame.*



The first 3 bytes of the 802.2 portion of the frame are the same as for the LLC frame. The DSAP and SSAP fields have the special value AA (hex, indicating that this is a SNAP-format frame).

Following the control field is a 5-byte protocol identifier; the first 3 bytes represent the organizational unit identifier (OUI), a unique value that is assigned to an organization by the IEEE. The remaining 2 bytes contain information that is similar to the type field used in an Ethernet frame.

For more information on Ethernet, IEEE 802.3, or other network types, refer to Appendix C, “References and Recommended Reading.”

Token Ring/IEEE 802.5

Token Ring was originally developed by IBM in the early 1980s. It is still IBM's primary LAN technology, and is second only to Ethernet/IEEE 802.3 in general LAN popularity. The IEEE 802.5 specification is almost identical to IBM's Token Ring network. In fact, the IEEE 802.5 specification was modeled after IBM Token Ring and continues to shadow IBM's Token Ring development. The term *Token Ring* is generally used to refer to both IBM's Token Ring networks and IEEE 802.5 networks.

Token Ring and IEEE 802.5 networks are compatible, although the specifications differ in relatively minor ways. IBM's Token Ring network specifies a star, with all end stations attached to a device called a multistation access unit (MSAU), whereas IEEE 802.5 does not specify a topology (although virtually all IEEE 802.5 implementations are also based on a star). There are other differences, including media type (IEEE 802.5 does not specify a media type, and IBM Token Ring networks primarily use twisted pair) and routing information field size. In some cases, the specifications are identical. Both IBM Token Ring and IEEE 802.5 specify the following:

- Baseband signaling
- Token passing
- Data rates of either 4 or 16 Mbps

Token Ring and IEEE 802.5 are the primary examples of token-passing networks. Token-passing networks move a small frame, called a *token*, around the network. Possession of the token grants the right to transmit. If a node receiving the token has no information to send, it simply passes the token to the next end station. Each station has a limitation on the amount of time that it can hold the token.

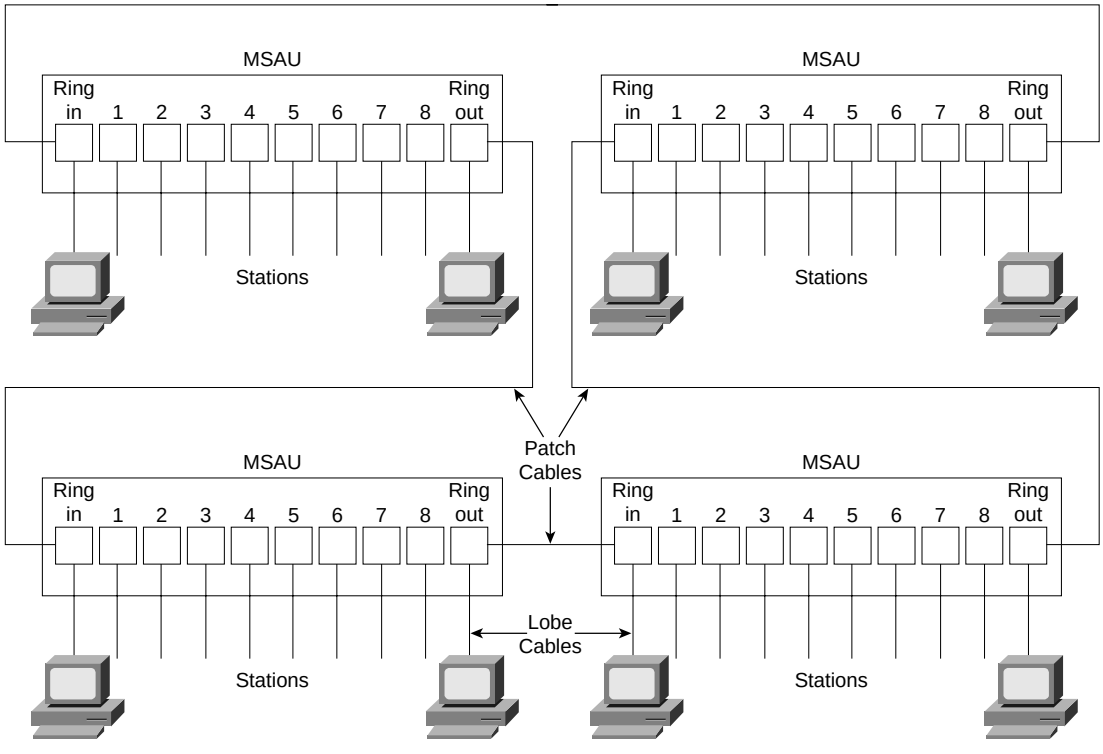
If a station that possesses the token has information to transmit, it alters 1 bit of the token (which turns the token into a start-of-frame sequence), appends the information it wants to transmit, performs a CRC calculation on the contents of the frame, and sends this information to the next station on the ring. While the information frame is circling the ring, there is no token on the network (unless the ring supports early token release), so other stations that want to transmit must wait. Therefore, collisions cannot occur in Token Ring networks. If early token release is supported, a new token can be released when frame transmission is completed. Although there may be multiple frames on the ring, there can be only one token on the ring at any time.

The information frame circulates the ring until it reaches the intended destination station, which copies the information for further processing. The information frame continues to circle the ring and is finally removed when it reaches the sending station. The sending station can check the returning frame to see whether the frame was seen and subsequently copied by the destination. The sending station is responsible for stripping the frame off the ring once it has returned.

Unlike CSMA/CD networks (such as Ethernet), token-passing networks are deterministic. In other words, it is possible to calculate the maximum time that will pass before any end station will be able to transmit. This feature and several reliability features make Token Ring networks ideal for applications where delay must be predictable, and robust network operation is important. Factory automation environments are examples of such applications.

Token Ring network stations are directly connected to MSAUs, which can be wired together to form one large ring (as shown in Figure 2-13). Patch cables connect MSAUs to adjacent MSAUs. Lobe cables connect MSAUs to stations. MSAUs include bypass relays for removing stations from the ring.

Figure 2-13 *Token Ring devices connect through MSAUs.*



Token Ring networks use several mechanisms for detecting and compensating for network faults. For example, one station in the Token Ring network is selected to be the active monitor. This station, which can potentially be any station on the network, acts as a centralized source of timing information for other ring stations and performs a variety of ring maintenance functions. One of these functions is the removal of continuously circulating frames from the ring. When a sending device fails, its frame might continue to circle the ring. This can prevent other stations from transmitting their own frames and essentially lock up the network. The active monitor can detect such frames, remove them from the ring, and generate a new token.

The Token Ring network's star topology also contributes to overall network reliability. Because all information in a Token Ring network is seen by active MSAUs, intelligent versions of these devices can be programmed to check for problems and selectively remove stations from the ring if necessary.

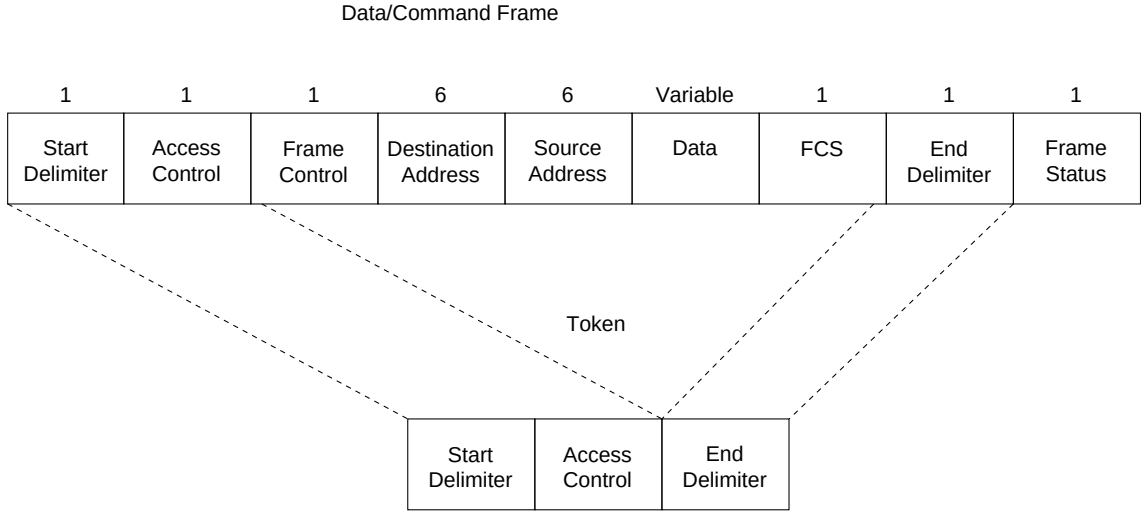
A Token Ring process called *beaconing* detects and tries to repair certain network faults. Whenever a station detects a serious problem with the network (such as a cable break), it sends a beacon frame. The beacon frame defines a fault domain, which includes the station reporting the failure, its nearest active upstream neighbor (NAUN), and everything in between. Beaconing initiates a process called *autoreconfiguration*, where nodes within the fault domain automatically perform diagnostics in an attempt to reconfigure the network around the failed areas. Physically, the MSAU can accomplish this through electrical reconfiguration.

Token Ring networks support a sophisticated priority system that permits high-priority applications to use the network more frequently than lower-priority applications. Token Ring frames have two fields that control priority: the priority field and the reservation field.

Only stations with a priority equal to or higher than the priority value contained in a token can seize that token. After the token is seized and changed to an information frame, only stations with a priority value higher than that of the transmitting station can reserve the token for the next pass around the network. When the next token is generated, it includes the higher priority of the reserving station. Stations that raise a token's priority level must reinstate the previous priority after their transmission is complete.

Token Ring networks define two frame types: tokens and data/command frames. Figure 2-14 shows both formats.

Figure 2-14 *The IEEE 802.5 and IBM Token Ring frame formats are identical.*



Tokens

A token is 3 bytes long and consists of a start delimiter, an access control byte, and an end delimiter.

The start delimiter alerts each station that a token (or data/command frame) has arrived. This field includes signals that distinguish the byte from the rest of the frame by violating the encoding scheme used elsewhere in the frame.

The access control byte contains the priority and reservation fields (used to allow some devices more frequent access to the token); a token bit (used to differentiate a token from a data/command frame); and a monitor bit (used by the active monitor to determine whether a frame is circling the ring endlessly).

Finally, the end delimiter signals the end of the token or data/command frame. It also contains bits to indicate a damaged frame and a frame that is the last in a logical sequence.

Data/Command Frames

Data/command frames vary in size, depending on the size of the data (information) field. Data frames carry information for upper-layer protocols; command frames contain control information and have no data for upper-layer protocols.

In data/command frames, a frame control byte follows the access control byte. The frame control byte indicates whether the frame contains data or control information. In control frames, this byte specifies the type of control information.

Following the frame control byte are the two address fields that identify the destination and source stations. As with IEEE 802.3, addresses are 6 bytes long.

The data field follows the address fields. The length of this field is limited by the token holding timer, which defines the maximum time a station can hold the token.

Following the data field is the FCS field. This field is filled by the source station with a calculated value dependent on the frame contents. The destination station recalculates the value to determine whether the frame was damaged in transit. If it was damaged, the frame is discarded.

As with the token, the end delimiter signifies the end of the data/command frame. However, unlike the token, the frame status byte follows the end delimiter in a data/command frame. The frame status field includes the address recognized and frame-copied indicators. The address recognized indicator allows the destination station to indicate that it recognized the destination address. The frame-copied indicator allows the destination station to indicate that it copied the frame data.

For more information on Token Ring technology, refer to Appendix C, “References and Recommended Reading.”

FDDI

The FDDI standard was produced by the ANSI X3T9.5 standards committee in the mid-1980s. During this period, high-speed engineering workstations were beginning to tax the capabilities of existing LANs (primarily Ethernet and Token Ring). A new LAN technology was needed that could easily support these workstations and their new distributed applications. At the same time, network reliability was becoming an increasingly important issue as system managers began to migrate mission-critical applications from large computers to networks. FDDI was developed to fill these needs.

After completing the FDDI specification, American National Standards Institute (ANSI) submitted FDDI to the ISO. ISO has created an international version of FDDI that is completely compatible with the ANSI standard version.

FDDI has gained a substantial following that continues to increase as the cost of FDDI interfaces decreases. FDDI is frequently used as a backbone technology as well as a means to connect high-speed computers in a local area.

FDDI specifies a 100-Mbps, token-passing, dual-ring LAN using a fiber-optic transmission medium. It defines the physical layer and the media-access portion of the data link layer and is roughly analogous to IEEE 802.3 and IEEE 802.5 in its relationship to the OSI reference model.

FDDI is similar to Token Ring, although it operates at faster speeds. The two networks share many features, including topology (ring), media-access technique (token passing), and reliability features (beaconing, for example).

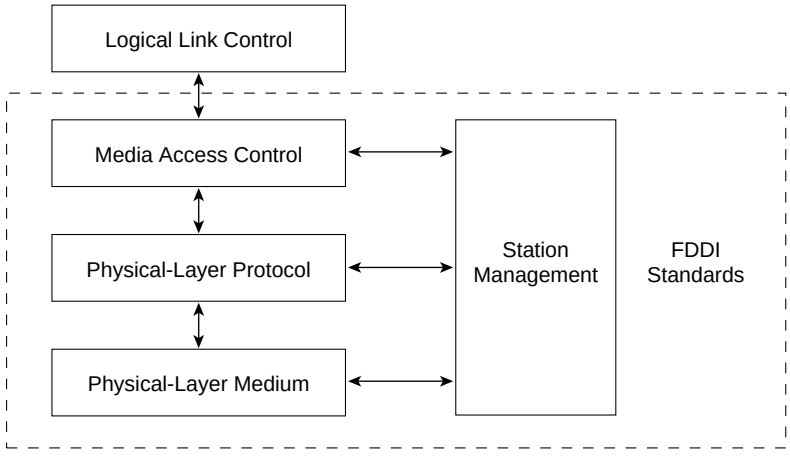
One of FDDI's most important characteristics is its use of optical fiber as a transmission medium. Optical fiber offers several advantages over traditional copper wiring:

- Security—Fiber does not emit electrical signals that can be tapped.
- Reliability—Fiber is immune to electrical interference.
- Speed—Fiber has much higher throughput potential than copper cable.

FDDI defines the use of two types of fiber: single mode (sometimes called monomode) and multimode. Modes can be thought of as bundles of light rays entering the fiber at a particular angle. Single-mode fiber allows only one mode of light to propagate through the fiber, and multimode fiber allows multiple modes of light to propagate through the fiber. Because multiple modes of light propagating through the fiber might travel different distances (depending on the entry angles), causing them to arrive at the destination at different times (a phenomenon called *modal dispersion*), single-mode fiber is capable of higher bandwidth and greater cable run distances than multimode fiber. Due to these characteristics, single-mode fiber is often used for campus backbones, and multimode fiber is often used for workgroup connectivity. Multimode fiber uses light-emitting diodes (LEDs) as the light-generating devices, and single-mode fiber generally uses lasers.

FDDI is defined by four separate specifications (see Figure 2-15).

Figure 2-15 *There are several FDDI standards for Layer 1 and 2 connectivity.*



The following list explains these specifications in more detail:

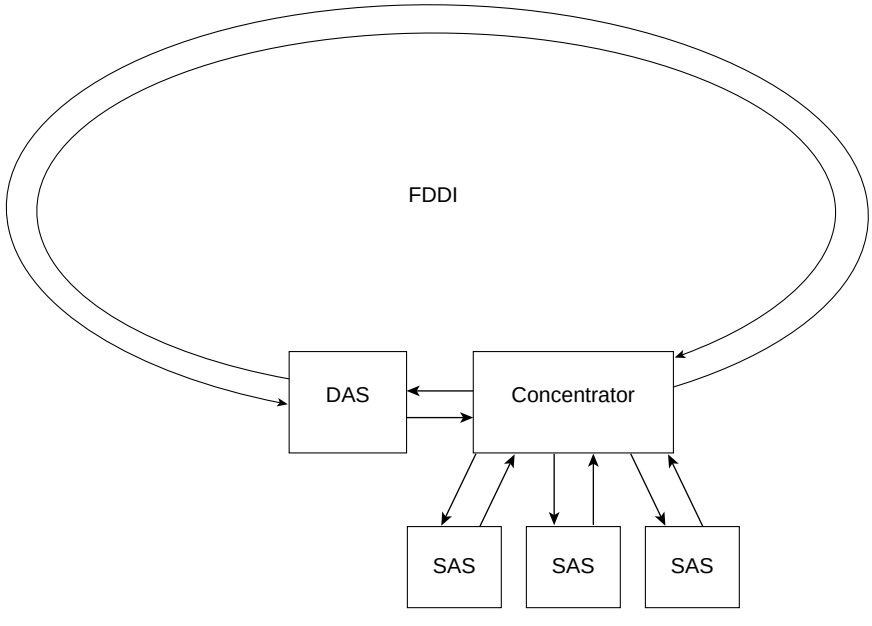
- Physical-Layer Medium Dependent (PMD)—Defines the characteristics of the transmission medium, including the fiber-optic link, power levels, bit error rates, optical components, and connectors.
- Physical-Layer Protocol (PHY)—Defines data encoding/decoding procedures, clocking requirements, framing, and other functions.

- Media Access Control (MAC)—Defines how the medium is accessed, including frame format, token handling, addressing, calculating a cyclic redundancy check value, and recovering from errors.
- Station Management (SMT)—Defines the FDDI station configuration, ring configuration, and ring control features, including station insertion and removal, initialization, fault isolation and recovery, scheduling, and statistics collection.

FDDI specifies the use of dual rings. Traffic on these rings travels in opposite directions. Physically, the rings consist of two or more point-to-point connections between adjacent stations. One of the two FDDI rings is called the *primary ring*; the other is called the *secondary ring*. The primary ring is used for data transmission, and the secondary ring is generally used as a backup.

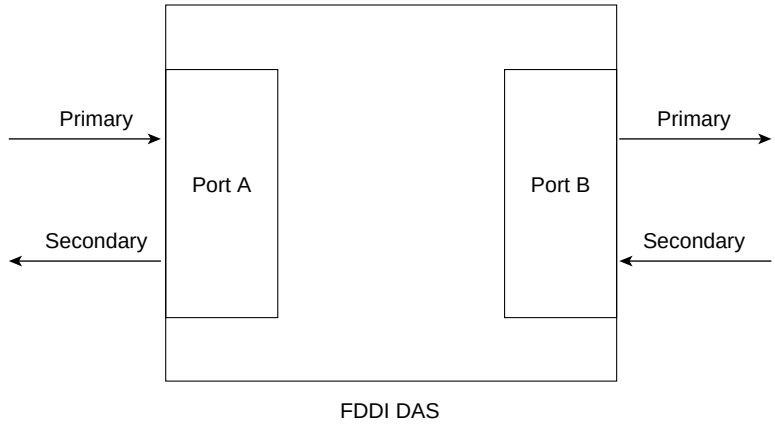
There are two classes that define device attachment to the FDDI network. Class B stations, or single attachment stations (SASs), attach to one ring; Class A stations, or dual attachment stations (DASs), attach to both rings. SASs are attached to the primary ring through a concentrator that provides connections for multiple SASs. The concentrator ensures that failure or power down of any given SAS does not interrupt the ring (see Figure 2-16). This is particularly useful when PCs or similar devices that power on and off frequently connect to the ring.

Figure 2-16 *A typical FDDI configuration with both DASs and SASs includes the FDDI node types DAS, SAS, and concentrator.*



Each FDDI DAS has two ports designated A and B. These ports connect the station to the dual FDDI ring. Therefore, each port provides a connection for both the primary and the secondary ring, as Figure 2-17 shows.

Figure 2-17 *FDDI DASs connect Port A to Port B.*



FDDI supports real-time allocation of network bandwidth, making it ideal for a variety of application types. FDDI provides this support by defining two types of traffic: synchronous and asynchronous. Synchronous traffic can consume a portion of the 100-Mbps total bandwidth of an FDDI network, and asynchronous traffic can consume the rest. Synchronous bandwidth is allocated to stations that require continuous transmission capability. This capability is useful for transmitting voice and video information, for example. Other stations use the remaining bandwidth asynchronously. FDDI's SMT specification defines a distributed bidding scheme to allocate FDDI bandwidth.

Asynchronous bandwidth is allocated using an eight-level priority scheme. Each application can be assigned an asynchronous priority level. FDDI also permits extended dialogues, where stations can temporarily use all asynchronous bandwidth by using a reserved token. FDDI's reserved token mechanism can essentially lock out stations that cannot use synchronous bandwidth.

FDDI provides several fault-tolerant features. The primary fault-tolerant feature is the dual ring. If a station on the dual ring fails or is powered down, or if the cable is damaged, the dual ring is automatically "wrapped" (that is, doubled back onto itself) into a single ring, as shown in Figure 2-18. In this figure, when Station 3 fails, the dual ring is automatically wrapped in Stations 2 and 4, forming a single ring. Although Station 3 is no longer on the ring, network operation continues for the remaining stations.

Figure 2-18 *FDDI station failures result in ring recovery via ring wrapping.*

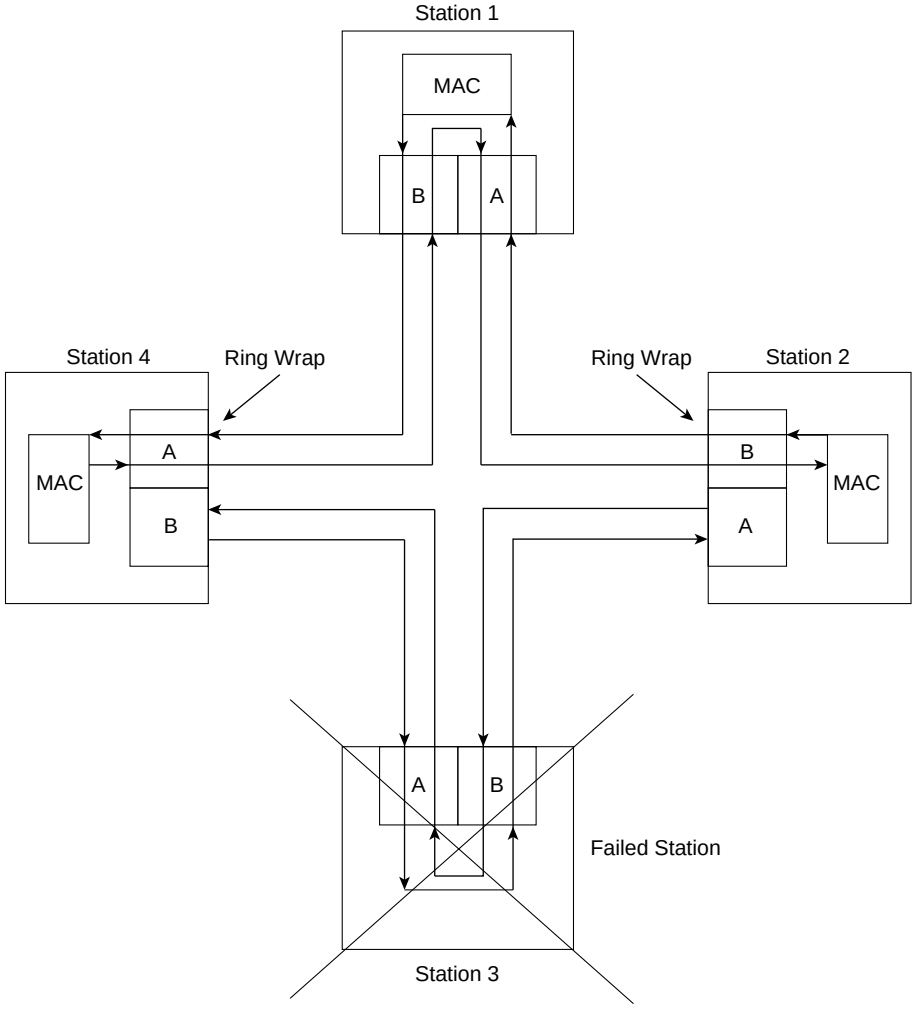
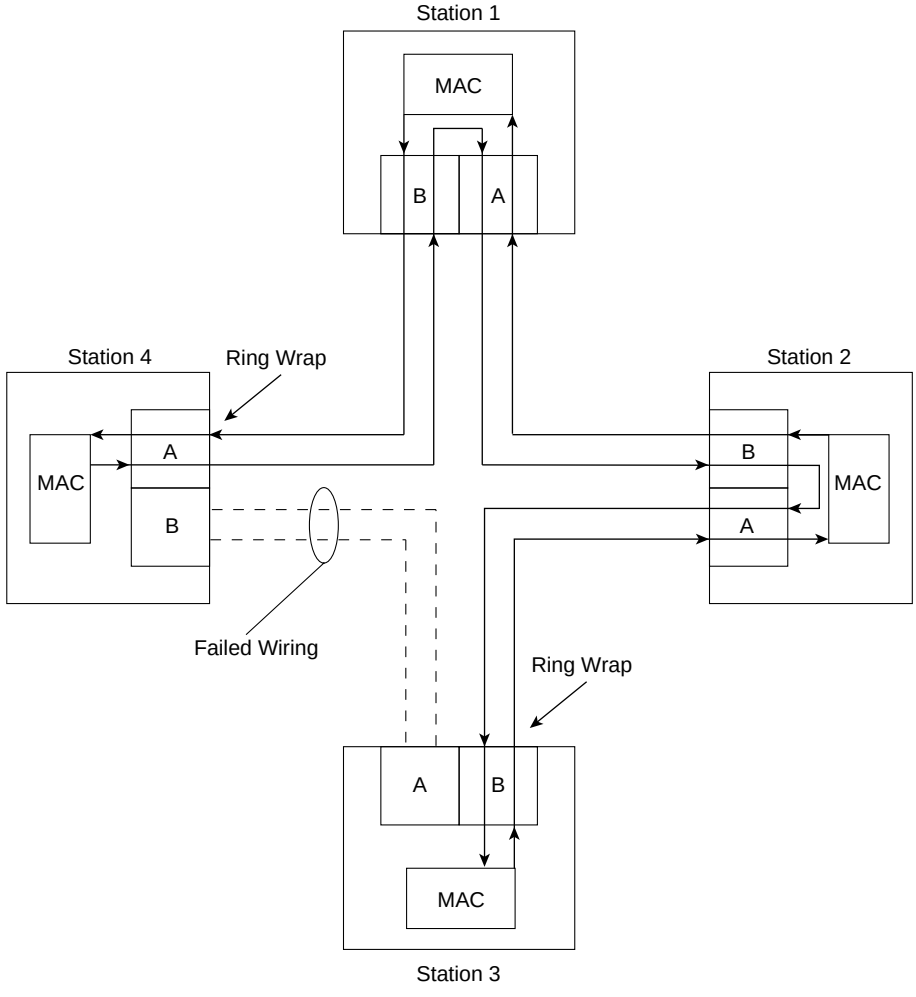


Figure 2-19 shows how FDDI compensates for a wiring failure. Stations 3 and 4 wrap the ring within themselves when wiring between them fails.

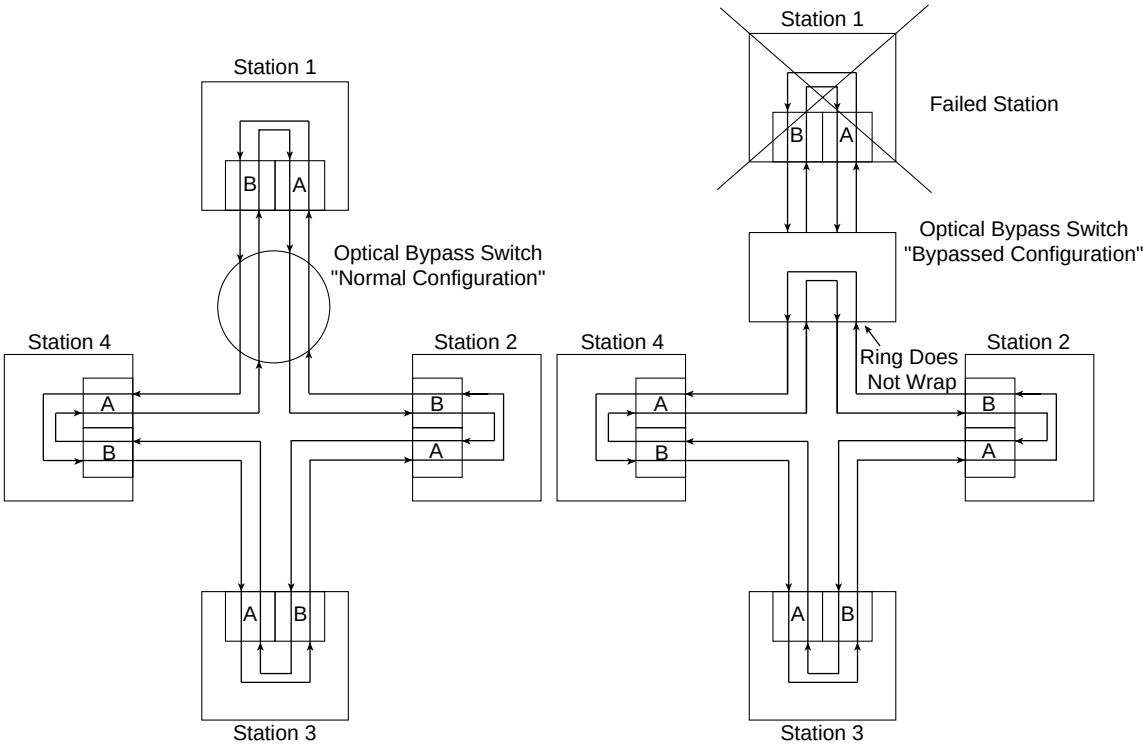
Figure 2-19 *Failed wiring also results in ring recovery via ring wrapping.*



As FDDI networks grow, the possibility of multiple ring failures grows. When two ring failures occur, the ring is wrapped in both cases, which might segment the ring into two separate rings that cannot communicate with each other. Subsequent failures might cause additional ring segmentation.

Optical bypass switches can be used to prevent ring segmentation by eliminating failed stations from the ring. Figure 2-20 shows that station 1 has been completely removed from the ring.

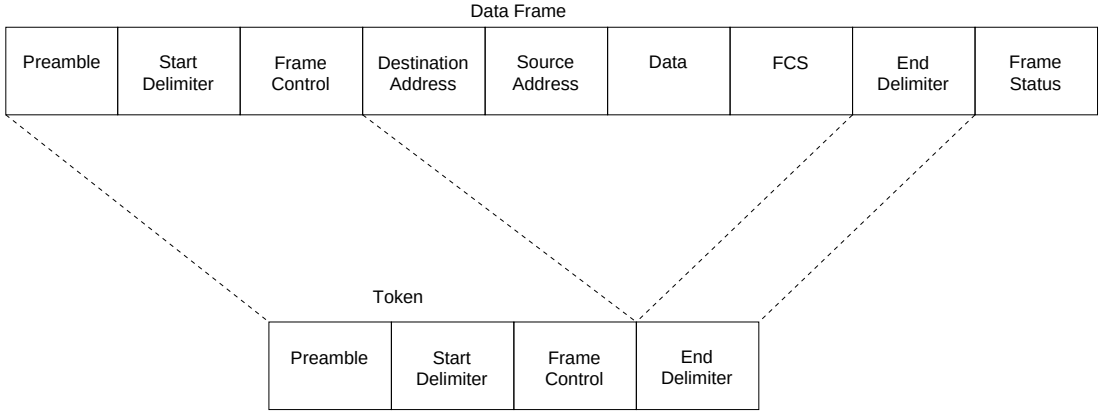
Figure 2-20 *Optical bypass switches can be used for ring recovery.*



Critical devices such as routers or mainframe hosts can use another fault-tolerant technique called *dual homing* to provide additional redundancy and help guarantee operation. In dual-homing situations, the critical device is attached to two concentrators. One concentrator link is declared the active link; the other is declared passive. The passive link stays in backup mode until the primary link (or the concentrator to which it is attached) is determined to have failed. When this occurs, the passive link is automatically activated.

FDDI frame formats (shown in Figure 2-21) are similar to those of Token Ring.

Figure 2-21 *The FDDI frame format is similar to the Token Ring format.*



The preamble prepares each station for the upcoming frame. The start delimiter indicates the beginning of the frame. It consists of signaling patterns that differentiate it from the rest of the frame.

The frame control field indicates the size of the address fields, whether the frame contains asynchronous or synchronous data, and other control information.

As with Ethernet and Token Ring, FDDI addresses are 6 bytes. The destination address field can contain a unicast (singular), multicast (group), or broadcast (every station) address, and the source address identifies the single station that sent the frame.

The data field contains either information destined for an upper-layer protocol or control information.

As with Token Ring and Ethernet, the FCS field is filled by the source station with a calculated CRC value dependent on the frame contents. The destination station recalculates the value to determine whether the frame was damaged in transit. If it was damaged, the frame is discarded.

The end delimiter contains nondata symbols that indicate the end of the frame.

The frame status field allows the source station to determine if an error occurred and if the frame was recognized and copied by a receiving station.

For more information on FDDI, refer to Appendix C, “References and Recommended Reading.”

PPP

In the late 1980s, the Internet began to experience explosive growth in the number of hosts supporting IP. The vast majority of these hosts were connected to LANs of various types, Ethernet being the most common. Most of the other hosts were connected through wide-area networks (WANs) such as X.25 public data networks (PDNs). Relatively few of these hosts were connected with simple point-to-point (that is, serial) links. Yet point-to-point links are among the oldest methods of data communications, and almost every host supports point-to-point connections.

One reason for the small number of point-to-point IP links was the lack of a standard Internet encapsulation protocol. PPP was designed to solve this problem. In addition to solving the problem of standardized Internet encapsulation of IP over point-to-point links, PPP was also designed to address other issues, including assignment and management of IP addresses, asynchronous (start/stop) and bit-oriented synchronous encapsulation, network protocol multiplexing, link configuration, link quality testing, error detection, and option negotiation for such capabilities as network-layer address negotiation and data compression negotiation. PPP addresses these issues by providing an extensible Link Control Protocol and a family of Network Control Protocols to negotiate optional configuration parameters and facilities. Today, PPP supports other protocols besides IP, including IPX and DECnet.

PPP provides a method for transmitting datagrams over serial point-to-point links. It has three main components:

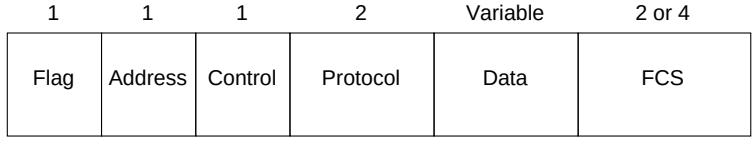
- A method for encapsulating datagrams over serial links—PPP uses HDLC as a basis for encapsulating datagrams over point-to-point links.
- An extensible Link Control Protocol to establish, configure, and test the data-link connection.
- A family of Network Control Protocols for establishing and configuring different network-layer protocols. PPP is designed to allow the simultaneous use of multiple network-layer protocols.

In order to establish communications over a point-to-point link, the originating PPP station first sends LCP frames to configure and (optionally) test the data link. After the link has been established and optional facilities have been negotiated as needed by the LCP, the originating PPP sends Network Control Protocol frames to choose and configure one or more network-layer protocols. When each of the chosen network-layer protocols has been configured, packets from each network-layer protocol can be sent over the link. The link remains configured for communications until explicit LCP or NCP frames close the link or until some external event occurs (for example, an inactivity timer expires or a user intervenes).

PPP can operate across any data terminal equipment (DTE)/data circuit-terminating equipment (DCE) interface (for example, EIA/TIA-232, EIA/TIA-422, EIA/TIA-423, or ITU-T V.35). The only absolute requirement imposed by PPP is the provision of a duplex circuit, either dedicated or switched, that can operate in either an asynchronous or synchronous bit-serial mode,

transparent to PPP link-layer frames. PPP does not impose any restrictions regarding transmission rate, other than those imposed by the particular DTE/DCE interface in use. Figure 2-22 shows the PPP frame format.

Figure 2-22 *The PPP frame format allows for both synchronous and asynchronous connections.*



The flag sequence is a single byte and indicates the beginning or end of a frame. The flag sequence consists of the binary sequence 01111110.

The address field is a single byte and contains the binary sequence 11111111, the standard broadcast address. PPP does not assign individual station addresses because it supports only a single connection between two devices.

The control field is a single byte and contains the binary sequence 00000011, which calls for transmission of user data in an unsequenced frame. A connectionless link service similar to that of LLC Type 1 is provided.

The protocol field is 2 bytes, and its value identifies the protocol encapsulated in the information field of the frame. The most up-to-date values of the protocol field are specified in the most recent assigned RFC numbers (RFC 1700).

NOTE An up-to-date list of assigned numbers can be found at www.iana.org.

The data field is 0 or more bytes long and contains the datagram for the protocol specified in the protocol field. The end of the information field is found by locating the closing flag sequence and allowing 2 bytes for the FCS field. The default maximum length of the information field is 1,500 bytes. By prior agreement, consenting PPP implementations can use other values for the maximum information field length.

The FCS field is normally 2 bytes. By prior agreement, consenting PPP implementations can use a 4-byte FCS for improved error detection.

The LCP can negotiate modifications to the standard PPP frame structure. However, modified frames are always clearly distinguishable from standard frames.

The PPP LCP provides a method of establishing, configuring, maintaining and terminating the point-to-point connection. LCP goes through four distinct phases:

Step 1 Link establishment and configuration negotiation—Before any network-layer datagrams (for example, IP) can be exchanged, LCP must first open the connection and negotiate configuration parameters. This phase is complete when a configuration acknowledgment frame has been both sent and received.

Step 2 Link quality determination—LCP allows an optional link quality determination phase following the link establishment and configuration negotiation phase. In this phase, the link is tested to determine whether the link quality is sufficient to bring up network-layer protocols. This phase is optional. LCP can delay transmission of network-layer protocol information until this phase is completed.

Step 3 Network-layer protocol configuration negotiation—When LCP has finished the link quality determination phase, network-layer protocols can be separately configured by the appropriate NCP and can be brought up and taken down at any time. If LCP closes the link, it informs the network-layer protocols so that they can take appropriate action.

Step 4 Link termination—LCP can terminate the link at any time. This is usually done at the request of a user, but can happen because of a physical event such as the loss of carrier or the expiration of an idle-period timer.

There are three classes of LCP frames:

- Link establishment frames—Used to establish and configure a link.
- Link termination frames—Used to terminate a link.
- Link maintenance frames—Used to manage and debug a link.

These frames are used to accomplish the work of each of the LCP phases.

For more information on PPP, refer to Appendix C, “References and Recommended Reading.”

SDLC and Derivatives

IBM developed the SDLC protocol in the mid-1970s for use in Systems Network Architecture (SNA) environments. SDLC was the first of an important new type of link-layer protocols based on synchronous, bit-oriented operation. Compared to synchronous character-oriented

(for example, Bisync from IBM) and synchronous byte-count-oriented protocols (for example, Digital Data Communications Message Protocol [DDCMP] from Digital Equipment Corporation), bit-oriented synchronous protocols are more efficient, more flexible, and often faster.

After developing SDLC, IBM submitted it to various standards committees. The ISO modified SDLC to create the HDLC protocol. The Consultative Committee for International Telegraph and Telephone (CCITT), which is now the Telecommunication Standardization Sector of the ITU (the ITU-T), subsequently modified HDLC to create Link Access Procedure (LAP) and then Link Access Procedure, Balanced (LAPB). The IEEE modified HDLC to create IEEE 802.2. Each of these protocols has become important in its own domain. SDLC remains SNA's primary link-layer protocol for WAN links.

SDLC supports a variety of link types and topologies. It can be used with point-to-point and multipoint links, half-duplex and full-duplex transmission facilities, and circuit-switched and packet-switched networks.

SDLC identifies two types of network nodes:

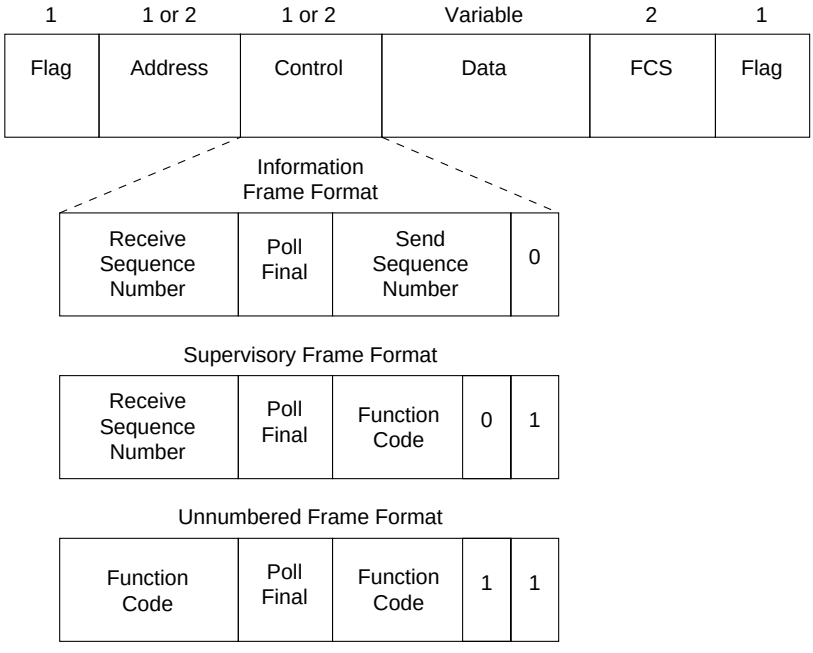
- Primary—Controls the operation of other stations (called secondaries). The primary polls the secondaries in a predetermined order. Secondaries can then transmit if they have outgoing data. The primary also sets up and tears down links and manages the link while it is operational.
- Secondary—Secondaries are controlled by a primary. Secondaries can only exchange information with primary, but they cannot send unless the primary gives permission.

SDLC primaries and secondaries can be connected in four basic configurations:

- Point-to-point—Involves only two nodes, one primary and one secondary.
- Multipoint—Involves one primary and multiple secondaries.
- Loop—Involves a loop topology, with the primary connected to the first and last secondaries. Intermediate secondaries pass messages through one another as they respond to the primary's requests.
- Hub go-ahead—Involves an inbound and an outbound channel. The primary uses the outbound channel to communicate with the secondaries. The secondaries use the inbound channel to communicate with the primary. The inbound channel is daisy-chained back to the primary through each secondary.

Figure 2-23 shows the SDLC frame.

Figure 2-23 *SDLC frames are used for SNA connectivity.*



As Figure 2-23 shows, SDLC frames are bounded by flag fields that indicate the start and end of the SDLC frame. The address field always contains the address of the secondary involved in the current communication. Because the primary is either the communication source or destination, there is no need to include the primary's address—it is already known by all secondaries.

The control field uses three different formats, depending on the type of SDLC frame used. The three SDLC frames are described as follows:

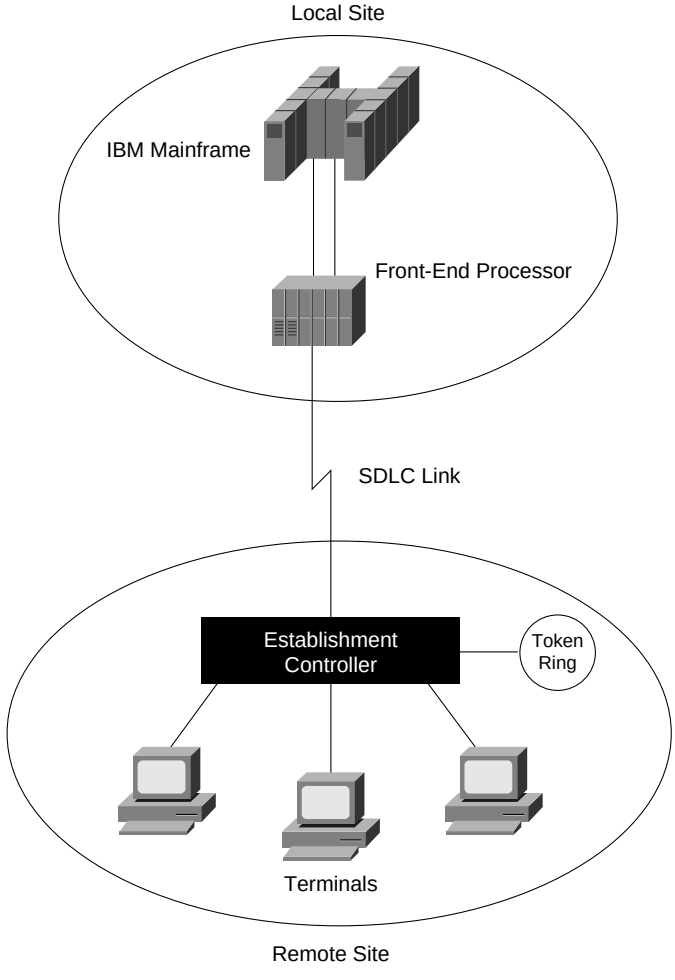
- Information (I) frames—These frames carry upper-layer information and some control information. Send and receive sequence numbers and the poll final (P/F) bit perform flow and error control. The send sequence number refers to the number of the frame now being sent. The receive sequence number provides the number of the frame to be received next. Both the sender and the receiver maintain send and receive sequence numbers. The primary uses the P/F bit to tell the secondary whether it requires an immediate response. The secondary uses this bit to tell the primary whether the current frame is the last in its current response.
- Supervisory (S) frames—These frames provide control information. They request and suspend transmission, report on status, and acknowledge the receipt of I frames. They do not have a data field.

- Unnumbered (U) frames—These frames, as the name suggests, are not sequenced. They are used for control purposes. For example, they can specify either a 1- or 2-byte control field, initialize secondaries, and do other similar functions. They can have a data field.

The FCS precedes the ending flag delimiter. The FCS is usually a CRC calculation remainder. The CRC calculation is redone in the receiver. If the result differs from the value in the sender's frame, an error is assumed.

Figure 2-24 shows a typical SDLC-based network configuration. As illustrated, an IBM establishment controller (formerly called a *cluster controller*) in a remote site connects to dumb terminals and to a Token Ring network. In a local site, an IBM host connects (via channel attach techniques) to an IBM front-end processor (FEP), which can also have links to local Token Ring LANs and an SNA backbone. The two sites are connected through an SDLC-based 56-kbps leased line.

Figure 2-24 *Typical SDLC-based network configuration connects users to the data center.*



Although it omits several features used in SDLC, HDLC is generally considered to be a compatible superset of SDLC. LAP is a subset of HDLC. LAPB was created to ensure ongoing compatibility with HDLC, which was modified in the early 1980s. IEEE 802.2 is a modification of HDLC for LAN environments. Qualified Logical Link Control (QLLC) is a link-layer protocol defined by IBM that allows SNA data to be transported across X.25 networks.

HDLC

HDLC shares SDLC's frame format, and HDLC fields have the same purpose as those in SDLC. Also, like SDLC, HDLC supports synchronous, full-duplex operation.

HDLC differs from SDLC in several minor ways. First, HDLC has an option for a 4-byte checksum. Also, unlike SDLC, HDLC does not support the loop or hub go-ahead configurations.

Also, HDLC has a 2-byte control field, allowing for a 7-bit window size (128 values) rather than the 3-bit (8 values) that SDLC allows.

The major difference between HDLC and SDLC is that SDLC supports only one transfer mode, and HDLC supports three. The three HDLC transfer modes are as follows:

- Normal response mode (NRM)—This transfer mode is used by both HDLC and SDLC. In this mode, secondaries cannot communicate with a primary until the primary has given permission.
- Asynchronous response mode (ARM)—This transfer mode is used by HDLC only and allows secondaries to initiate communication with a primary without receiving permission.
- Asynchronous balanced mode (ABM)—ABM introduces the combined node. A combined node can act as a primary or a secondary, depending on the situation. All ABM communication is between multiple combined nodes. In ABM environments, any combined station can initiate data transmission without permission from any other.

LAPB

LAPB is best known for its presence in the X.25 protocol stack. LAPB shares the same frame format, frame types, and field functions as SDLC and HDLC. Unlike either of these, however, LAPB is restricted to the ABM transfer mode, and so is appropriate only for combined stations. Also, LAPB circuits can be established by either the DTE or the DCE. The station initiating the call is determined to be the primary, and the responding station is the secondary. Finally, LAPB use of the P/F bit is somewhat different from that of the other protocols.

IEEE 802.2

IEEE 802.2 is often referred to as LLC. It is extremely popular in LAN environments, where it interoperates with protocols such as IEEE 802.3, IEEE 802.4, and IEEE 802.5. IEEE 802.2

offers three types of service. Type 1 provides unacknowledged connectionless service, Type 2 provides connection-oriented service, and Type 3 provides acknowledged connectionless service.

As an unacknowledged connectionless service, LLC Type 1 does not confirm data transfers. Because many upper-layer protocols such as TCP offer reliable data transfer that can compensate for unreliable lower-layer protocols, Type 1 is a commonly used service.

LLC2 service establishes logical connections between sender and receiver, and is therefore connection oriented. LLC2 acknowledges data upon receipt. It is commonly found in IBM communication systems.

Although LLC Type 3 service supports acknowledged data transfer, it does not establish logical connections. As a compromise between the other two LLC services, LLC Type 3 is useful in factory automation environments where error detection is important (through acknowledgment) but virtual connection capability is extremely limited (due to memory constraints).

End stations can support multiple LLC service types. A Class I device supports only Type 1 service. A Class II device supports both Type 1 and Type 2 services. Class III devices support both Type 1 and Type 3 services, and Class IV devices support all three types of service.

Upper-layer processes use IEEE 802.2 services through service access points (SAPs). The IEEE 802.2 header begins with a DSAP field, which identifies the receiving upper-layer process. In other words, after the receiving node's IEEE 802.2 implementation completes its processing, the upper-layer process identified in the DSAP field receives the remaining data. Following the DSAP address is the SSAP address, which identifies the sending upper-layer process. As an example, the value 06 (hexadecimal) indicates the upper-layer process is IP. The value E0 (hexadecimal) represents IPX.

QLLC

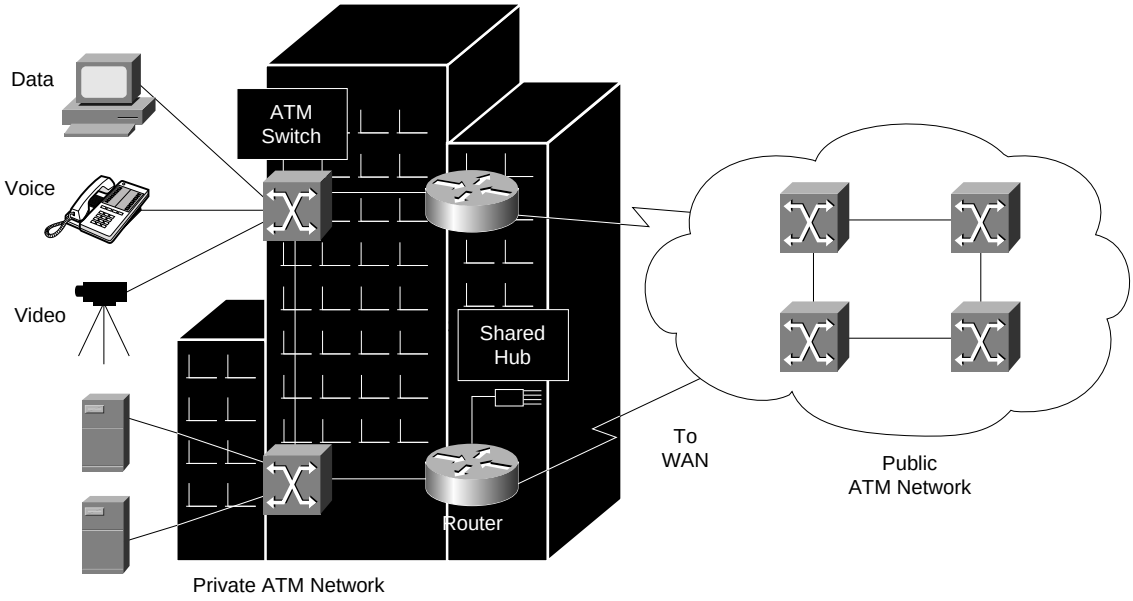
QLLC provides the data link control capabilities that are required to transport SNA data across X.25 networks. Together, QLLC and X.25 replace SDLC in the SNA protocol stack.

QLLC uses the packet-level protocol (Layer 3) of the X.25 protocol stack. To indicate that a Layer 3 X.25 packet must be handled by QLLC, a special bit, called the qualifier bit, in the general format identifier (GFI) of the Layer 3 X.25 packet-level header is set to 1. The SNA data is carried as user data in Layer 3 X.25 packets.

ATM Switching

ATM is an ITU standard for cell relay wherein information for multiple service types, such as voice, video, or data, is conveyed in small, fixed-size cells. ATM networks are connection-oriented. This chapter provides summaries of ATM protocols, services, and operation. Figure 2-25 illustrates a private ATM network and a public ATM network carrying voice, video, and data traffic.

Figure 2-25 *A private ATM network and a public ATM network can both carry voice, video, and data traffic.*



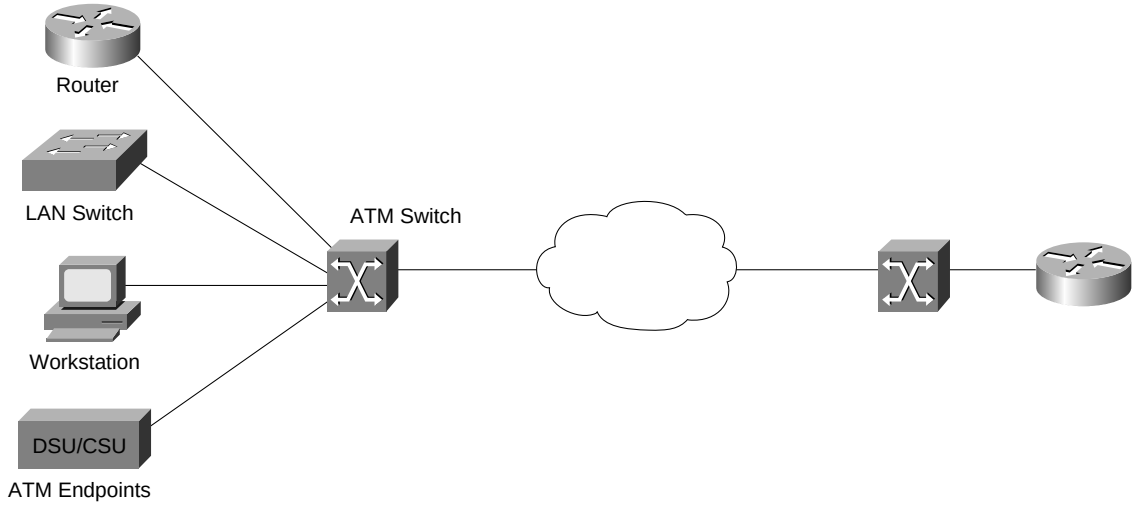
ATM combines the benefits of circuit switching (guaranteed capacity and constant transmission delay) with those of packet switching (flexibility and efficiency for intermittent traffic). It provides scalable bandwidth from a few megabits per second (Mbps) to many gigabits per second (Gbps). Because of its asynchronous nature, ATM is more efficient than synchronous technologies, such as time-division multiplexing (TDM).

With TDM, users are assigned to time slots, and no other station can send in that time slot. If a station has a lot of data to send, it can send only when its time slot comes up, even if all other time slots are empty. If, however, a station has nothing to transmit when its time slot comes up, the time slot gets sent empty and is wasted. Because ATM is asynchronous, time slots are available on demand with information identifying the source of the transmission contained in the header of each ATM cell.

ATM Devices

An ATM network is made up of an ATM switch and ATM endpoints. An ATM switch is responsible for cell transit through an ATM network. The job of an ATM switch is well defined: It accepts the incoming cell from an ATM endpoint or another ATM switch. It then reads and updates the cell-header information and quickly switches the cell to an output interface towards its destination. An ATM endpoint (or end system) contains an ATM network interface adapter. Examples of ATM endpoints are workstations, routers, data-service units (DSUs), LAN switches, and video coder-decoders (CODECs). Figure 2-26 illustrates an ATM network made up of ATM switches and ATM endpoints.

Figure 2-26 *An ATM network comprises ATM switches and endpoints.*



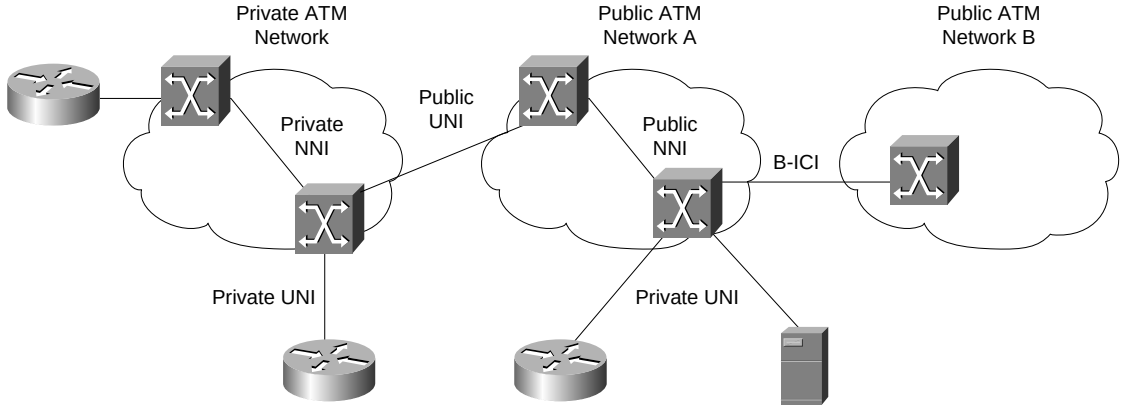
ATM Network Interfaces

An ATM network consists of a set of ATM switches interconnected by point-to-point ATM links or interfaces. ATM switches support two primary types of interfaces: User-Network Interface (UNI) and Network-to-Network Interface (NNI). The UNI connects ATM end-systems (such as hosts and routers) to an ATM switch. The NNI connects two ATM switches.

Depending on whether the switch is owned and located at the customer's premises or publicly owned and operated by the telephone company, UNI and NNI can be further subdivided into public and private UNIs and NNIs. A private UNI connects an ATM endpoint and a private ATM switch. Its public counterpart connects an ATM endpoint or private switch to a public switch. A private NNI connects two ATM switches within the same private organization. A public one connects two ATM switches within the same public organization.

Figure 2-27 illustrates the ATM interface specifications for private and public networks.

Figure 2-27 *ATM interface specifications differ for private and public networks.*



ATM Cell-Header Format

ATM transfers information in fixed-size units called *cells*. Each cell consists of 53 octets, or bytes. The first 5 bytes contain cell-header information, and the remaining 48 contain the "payload" (that is, the user information). Small fixed-length cells are well suited to transferring voice and video traffic because such traffic is intolerant of delays that result from having to wait for a large data packet to download, among other things.

An ATM cell header can be one of two formats: UNI or NNI. The UNI header is used for communication between ATM endpoints and ATM switches in private ATM networks. The NNI header is used for communication between ATM switches. Figure 2-28 depicts the basic ATM cell format, the ATM UNI cell-header format, and the ATM NNI cell-header format.

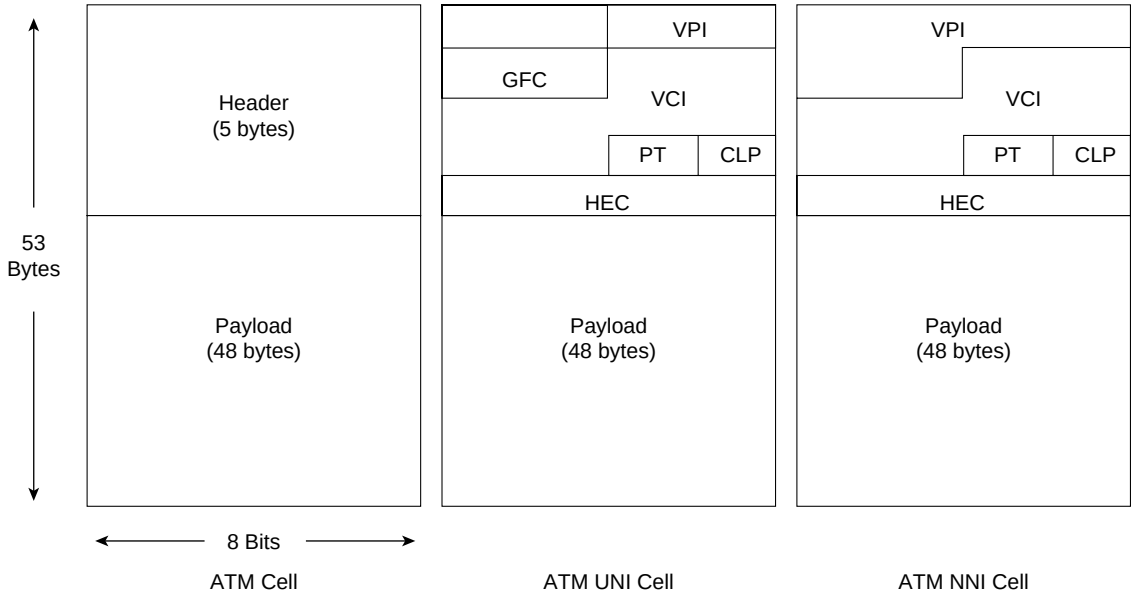
Unlike the UNI, the NNI header does not include the generic flow control (GFC) field. Additionally, the NNI header has a VPI field that occupies the first 12 bits, allowing for larger trunks between public ATM switches.

In addition to GFC and VPI header fields, several others are used in ATM cell-header fields. The following descriptions summarize the ATM cell-header fields illustrated in Figure 2-28:

- GFC—Provides local functions, such as identifying multiple stations that share a single ATM interface. This field is typically not used and is set to its default value.
- VPI—In conjunction with the VCI, identifies the next destination of a cell as it passes through a series of ATM switches on the way to its destination.
- VCI—In conjunction with the VPI, identifies the next destination of a cell as it passes through a series of ATM switches on the way to its destination.

- Payload type (PT)—Indicates in the first bit whether the cell contains user data or control data. If the cell contains user data, the second bit indicates congestion, and the third bit indicates whether the cell is the last in a series of cells that represent a single AAL5 frame.
- Congestion loss priority (CLP)—Indicates whether the cell should be discarded if it encounters extreme congestion as it moves through the network. If the CLP bit equals 1, the cell should be discarded in preference to cells with the CLP bit equal to zero.
- Header error control (HEC)—Calculates checksum only on the header itself.

Figure 2-28 *An ATM cell, ATM UNI cell, and ATM NNI cell header each contain 48 bytes of payload.*



ATM Services

Three types of ATM services exist: permanent virtual circuits (PVCs), switched virtual circuits (SVCs), and connectionless service (which is similar to Switched Multimegabit Data Service [SMDS]).

A PVC allows direct connectivity between sites. In this way, a PVC is similar to a leased line. Among its advantages, a PVC guarantees availability of a connection and does not require call setup procedures between switches. Disadvantages of PVCs include static connectivity and manual setup.

An SVC is created and released dynamically and remains in use only as long as data is being transferred. In this sense, it is similar to a telephone call. Dynamic call control requires a signaling protocol between the ATM endpoint and the ATM switch. The advantages of SVCs include connection flexibility and call setup that can be handled automatically by a networking device. Disadvantages include the extra time and overhead required to set up the connection.

ATM networks are fundamentally connection-oriented, which means that a virtual channel must be set up across the ATM network prior to any data transfer. (A virtual channel is roughly equivalent to a virtual circuit.)

Two types of ATM connections exist: *virtual paths (VPs)*, which are identified by virtual path identifiers, and *virtual channels (VCs)*, which are identified by the combination of a VPI and a VCI.

A virtual path is a bundle of virtual channels, all of which are switched transparently across the ATM network on the basis of the common VPI. All VCIs and VPIs, however, have only local significance across a particular link and are remapped, as appropriate, at each switch.

A transmission path is a bundle of VPs. Figure 2-29 illustrates how VCs concatenate to create VPs, which, in turn, concatenate to create a transmission path.

Figure 2-29 *Multiple VCs are bundled together to create VPs.*



ATM Switching Operation

The basic operation of an ATM switch is straightforward: The cell is received across a link on a known VCI or VPI value. The switch looks up the connection value in a local translation table to determine the outgoing port (or ports) of the connection and the new VPI/VCI value of the connection on that link. The switch then retransmits the cell on that outgoing link with the appropriate connection identifiers. Because all VCIs and VPIs have only local significance across a particular link, these values are remapped, as necessary, at each switch.

The ATM Reference Model

The ATM architecture uses a logical model to describe the functionality it supports. ATM functionality corresponds to the physical layer and part of the data link layer of the OSI reference model.

The ATM reference model is composed of the following planes, which span all layers:

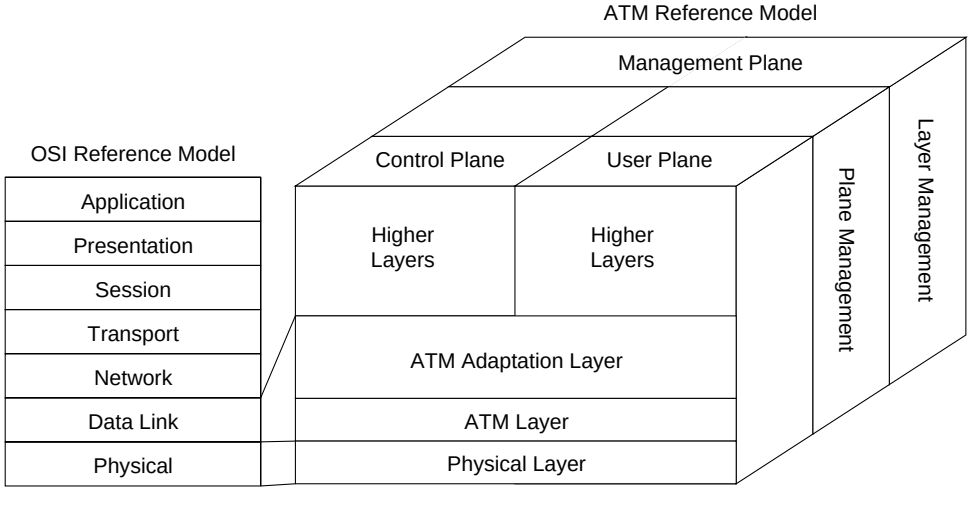
- Control—This plane is responsible for generating and managing signaling requests.
- User—This plane is responsible for managing the transfer of data.
- Management—This plane contains two components:
  - — Layer management manages layer-specific functions, such as the detection of failures and protocol problems.
  - — Plane management manages and coordinates functions related to the complete system.

The ATM reference model is composed of the following ATM layers:

- Physical layer—Analogous to the physical layer of the OSI reference model, the ATM physical layer manages the medium-dependent transmission.
- ATM layer—Combined with the ATM adaptation layer, the ATM layer is roughly analogous to the data link layer of the OSI reference model. The ATM layer is responsible for establishing connections and passing cells through the ATM network. To do this, it uses information in the header of each ATM cell.
- ATM adaptation layer (AAL)—Combined with the ATM layer, the AAL is roughly analogous to the data link layer of the OSI model. The AAL is responsible for isolating higher-layer protocols from the details of the ATM processes.

Finally, the higher layers residing above the AAL accept user data, arrange it into packets, and hand it to the AAL. Figure 2-30 illustrates the ATM reference model.

Figure 2-30 *The ATM reference model relates to the lowest two layers of the OSI reference model.*



The ATM physical layer has four functions: Bits are converted into cells; the transmission and receipt of bits on the physical medium are controlled; ATM cell boundaries are tracked; and cells are packaged into the appropriate type of frame for the physical medium.

AAL1, a connection-oriented service, is suitable for handling circuit-emulation applications, such as voice and video conferencing. Circuit-emulation service also accommodates the attachment of equipment currently using leased lines to an ATM backbone network. AAL1 requires timing synchronization between the source and destination. For this reason, AAL1 depends on a medium, such as SONET, that supports clocking.

AAL3/4 supports both connection-oriented and connectionless data. It was designed for network service providers and is closely aligned with SMDS. AAL3/4 is used to transmit SMDS packets over an ATM network.

AAL5 is the primary AAL for data and supports both connection-oriented and connectionless data. It is used to transfer most non-SMDS data, such as classical IP over ATM and LAN Emulation (LANE).

ATM Addressing

The ITU-T standard is based on the use of E.164 addresses (similar to telephone numbers) for public ATM (BISDN) networks. The ATM Forum extended ATM addressing to include private networks. ITU-T decided on the subnetwork or overlay model of addressing, in which the ATM layer is responsible for mapping network-layer addresses to ATM addresses. This subnetwork model is an alternative to using network-layer protocol addresses (such as IP and IPX) and existing routing protocols (such as IGRP and RIP). The ATM Forum defined an address format based on the structure of the OSI network service access point (NSAP) addresses.

The Subnetwork Model of Addressing The subnetwork model of addressing decouples the ATM layer from any existing higher-layer protocols, such as IP or IPX. As such, it requires an entirely new addressing scheme and routing protocol. All ATM systems must be assigned an ATM address, in addition to any higher-layer protocol addresses. This requires an ATM address resolution protocol (ATM_ARP) to map higher-layer addresses to their corresponding ATM addresses.

NSAP-Format ATM Addresses The 20-byte NSAP-format ATM addresses are designed for use within private ATM networks, whereas public networks typically use E.164 addresses, which are formatted as defined by ITU-T. The ATM Forum did specify an NSAP encoding for E.164 addresses, which will be used for encoding E.164 addresses within private networks, but this address also can be used by some private networks.

Such private networks can base their own (NSAP format) addressing on the E.164 address of the public UNI to which they are connected and can take the address prefix from the E.164 number, identifying local nodes by the lower-order bits.

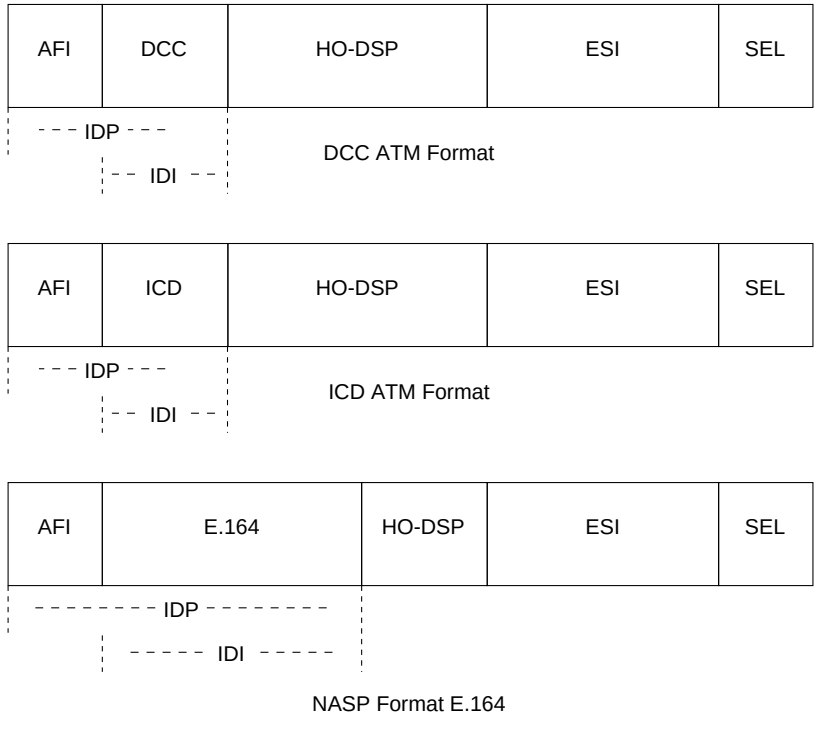
All NSAP-format ATM addresses consist of three components: the authority and format identifier (AFI), the initial domain identifier (IDI), and the domain-specific part (DSP). The AFI identifies the type and format of the IDI, which, in turn, identifies the address allocation and administrative authority. The DSP contains actual routing information.

Three formats of private ATM addressing differ by the nature of the AFI and IDI. In the NSAP-encoded E.164 format, the IDI is an E.164 number. In the DCC format, the IDI is a data country code (DCC), which identifies particular countries, as specified in ISO 3166. Such addresses are administered by the ISO National Member Body in each country. In the ICD format, the IDI is an international code designator (ICD), which is allocated by the ISO 6523 registration authority (the British Standards Institute). ICD codes identify particular international organizations.

The ATM Forum recommends that organizations or private-network service providers use either the DCC or ICD formats to form their own numbering plan.

Figure 2-31 illustrates the three formats of ATM addresses used for private networks.

Figure 2-31 *Three formats of ATM addresses are used for private networks: DCC and ICD ATM formats and NASP E.164.*



The following descriptions summarize the fields illustrated in Figure 2-31:

- AFI—Identifies the type and format of the address (E.164, ICD, or DCC).
- DCC—Identifies particular countries.
- High-order domain-specific part (HO-DSP)—Combines the Routing Domain (RD) and Area Identifier (AREA) of the NSAP addresses. The ATM Forum combined these fields to support a flexible, multi-level addressing hierarchy for prefix-based routing protocols.
- End system identifier (ESI)—Specifies the 48-bit MAC address, as administered by the IEEE.
- Selector (SEL)—Used for local multiplexing within end stations and has no network significance.
- ICD—Identifies particular international organizations.
- E.164—Indicates the BISDN E.164 address.

ATM Connections

ATM supports two types of connections: point-to-point and point-to-multipoint. Point-to-point connects two ATM end systems and can be unidirectional (one-way communication) or bidirectional (two-way communication). Point-to-multipoint connects a single-source end-system (known as the root node) to multiple destination end-systems (known as leaves). Such connections are unidirectional only. Root nodes can transmit to leaves, but leaves cannot transmit to the root or each other on the same connection. Cell replication is done within the ATM network by the ATM switches where the connection splits into two or more branches.

It would be desirable in ATM networks to have bidirectional multipoint-to-multipoint connections. Such connections are analogous to the broadcasting or multicasting capabilities of shared-medium LANs, such as Ethernet and Token Ring. A broadcasting capability is easy to implement in shared-medium LANs, where all nodes on a single LAN segment must process all packets sent on that segment. Unfortunately, a multipoint-to-multipoint capability cannot be implemented by using AAL5, which is the most common AAL to transmit data across an ATM network. Unlike AAL3/4, with its message identifier (MID) field, AAL5 does not provide a way within its cell format to interleave cells from different AAL5 packets on a single connection. This means that all AAL5 packets sent to a particular destination across a particular connection must be received in sequence; otherwise, the destination reassembly process will be unable to reconstruct the packets. This is why ATM AAL5 point-to-multipoint connections can be only unidirectional. If a leaf node were to transmit an AAL5 packet onto the connection, for example, it would be received by both the root node and all other leaf nodes. At these nodes, the packet sent by the leaf could be interleaved with packets sent by the root and possibly other leaf nodes, precluding the reassembly of any of the interleaved packets.

ATM QOS

ATM supports *quality of service* (QOS) guarantees comprising traffic contract, traffic shaping, and traffic policing.

A *traffic contract* specifies an envelope that describes the intended data flow. This envelope specifies values for peak bandwidth, average sustained bandwidth, and burst size, among others. When an ATM end system connects to an ATM network, it enters a "contract" with the network based on QOS parameters.

Traffic shaping is the use of queues to constrain data bursts, limit peak data rate, and smooth jitters so that traffic will fit within the promised envelope. ATM devices are responsible for adhering to the contract by means of traffic shaping. ATM switches can use methods known as traffic policing to enforce the contract. The switch can measure the actual traffic flow and compare it against the agreed-upon traffic envelope. If the switch finds that traffic is outside the agreed-upon parameters, it can set the cell-loss priority (CLP) bit of the offending cells. Setting the CLP bit makes the cell "discard eligible," which means that any switch handling the cell is allowed to drop the cell during periods of congestion.

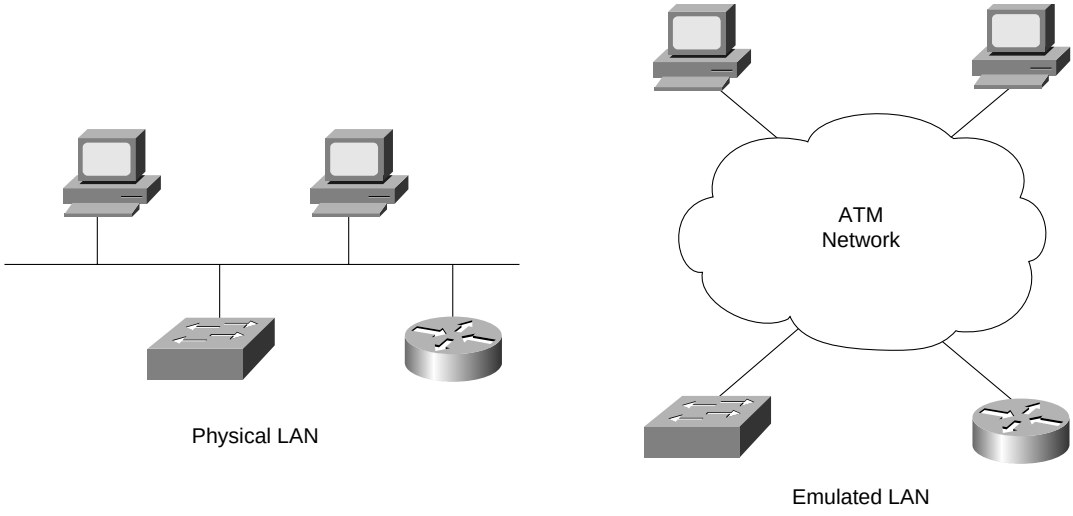
LANE

LANE is a standard defined by the ATM Forum that provides to stations attached via ATM the same capabilities they normally obtain from legacy LANs, such as Ethernet and Token Ring. As the name suggests, the function of the LANE protocol is to emulate a LAN on top of an ATM network. Specifically, the LANE protocol defines mechanisms for emulating either an IEEE 802.3 Ethernet or an 802.5 Token Ring LAN. The current LANE protocol does not define a separate encapsulation for FDDI. (FDDI packets must be mapped into either Ethernet or Token Ring emulated LANs [ELANs] by using existing translational bridging techniques.) Fast Ethernet (100BaseT) and IEEE 802.12 (100VG-AnyLAN) both can be mapped unchanged because they use the same packet formats. Figure 2-32 compares a physical LAN and an ELAN.

The LANE protocol defines a service interface for higher-layer (that is, network-layer) protocols that is identical to that of existing LANs. Data sent across the ATM network is encapsulated in the appropriate LAN MAC packet format. Simply put, the LANE protocols make an ATM network look and behave like an Ethernet or Token Ring LAN—albeit one operating much faster than an actual Ethernet or Token Ring LAN network.

It is important to note that LANE does not attempt to emulate the actual MAC protocol of the specific LAN concerned (that is, CSMA/CD for Ethernet or token passing for IEEE 802.5). LANE requires no modifications to higher-layer protocols to enable their operation over an ATM network. Because the LANE service presents the same service interface of existing MAC protocols to network-layer drivers (such as an NDIS- or ODI-like driver interface), no changes are required in those drivers.

Figure 2-32 *An ATM network can emulate a physical LAN.*



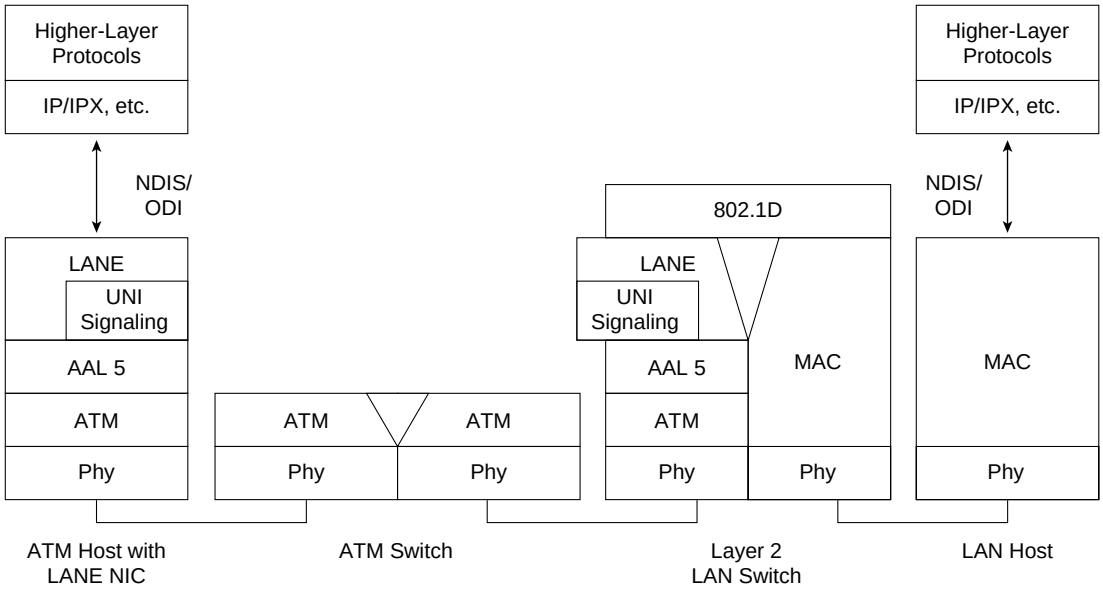
LANE Protocol Architecture The basic function of the LANE protocol is to resolve MAC addresses to ATM addresses. The goal is to resolve such address mappings so that LANE end systems can set up direct connections between themselves and then forward data. The LANE protocol is deployed in two types of ATM-attached equipment: ATM network interface cards (NICs) and internetworking and LAN-switching equipment.

ATM NICs implement the LANE protocol and interface to the ATM network but present the current LAN service interface to the higher-level protocol drivers within the attached end system. The network-layer protocols on the end system continue to communicate as if they were on a known LAN, by using known procedures. However, they are able to use the vastly greater bandwidth of ATM networks.

The second class of network gear that implements LANE consists of ATM-attached LAN switches and routers. These devices, together with directly attached ATM hosts equipped with ATM NICs, are used to provide a virtual LAN (VLAN) service in which ports on the LAN switches will be assigned to particular VLANs independent of physical location. Figure 2-33 shows the LANE protocol architecture implemented in ATM network devices.

NOTE The LANE protocol does not directly affect ATM switches. LANE, as with most of the other ATM internetworking protocols, builds on the overlay model. As such, the LANE protocols operate transparently over and through ATM switches, using only standard ATM signaling procedures.

Figure 2-33 *LANE protocol architecture can be implemented in ATM network devices.*



X.25

In the 1970s, a set of protocols was needed to provide users with WAN connectivity across PDNs. PDNs such as Telnet and Tymnet had achieved remarkable success, but it was thought that protocol standardization would increase subscription to PDNs by providing improved equipment compatibility and lower cost. The result of the ensuing development effort was a group of protocols, the most popular of which is X.25.

X.25 was developed by the common carriers (phone companies, essentially) rather than any single commercial enterprise. The specification is therefore designed to work well regardless of a user's system type or manufacturer. Users contract with the common carriers to use their packet-switched networks (PSNs) and are charged based on PSN use. Services offered and charges levied are regulated by the Federal Communications Commission (FCC) in the United States.

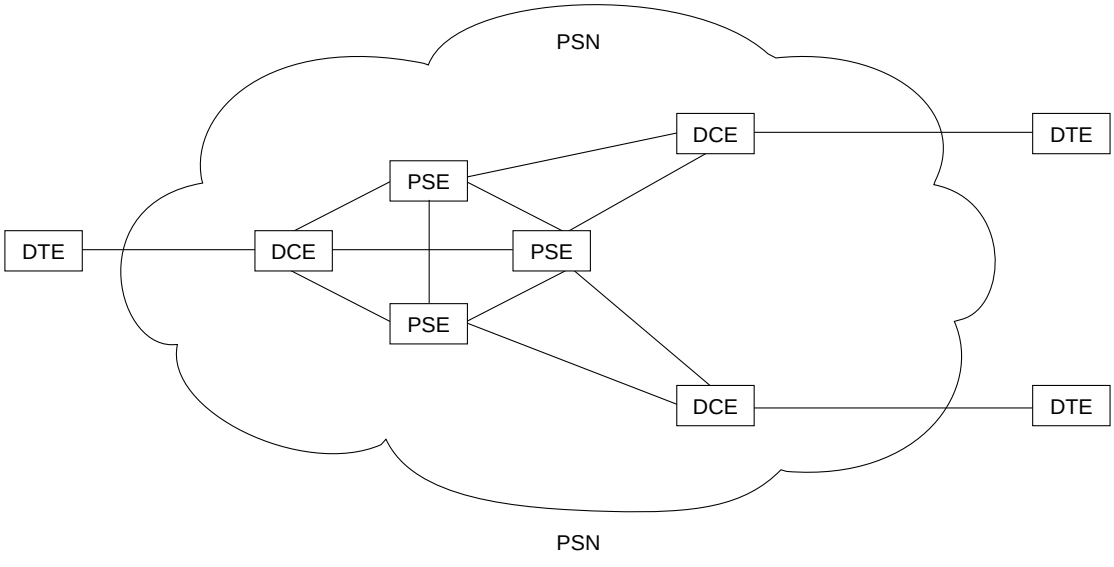
One of X.25's unique attributes is its international nature. X.25 and related protocols are administered by an agency of the United Nations called the ITU. The ITU-T carries out the functions of the former CCITT. Union members include the FCC, the European Postal Telephone and Telegraph organizations, the common carriers, and many computer and data communications companies. As a result, X.25 is truly a global standard.

NOTE ITU can be reached at www.itu.ch.

X.25 defines a telephone network for data communications. To begin communication, one computer calls another to request a communication session. The called computer can accept or refuse the connection. If the call is accepted, the two systems can begin full-duplex information transfer. Either side can terminate the connection at any time.

The X.25 specification defines a point-to-point interaction between DTE and DCE. Terminals and hosts in the user's facilities connect to modems or packet switches generally located in the carrier's facilities, which connect to packet-switching exchanges (PSE) and other DCEs inside a PSN and, ultimately, to another DTE device. Figure 2-34 shows the relationship between the entities in an X.25 network.

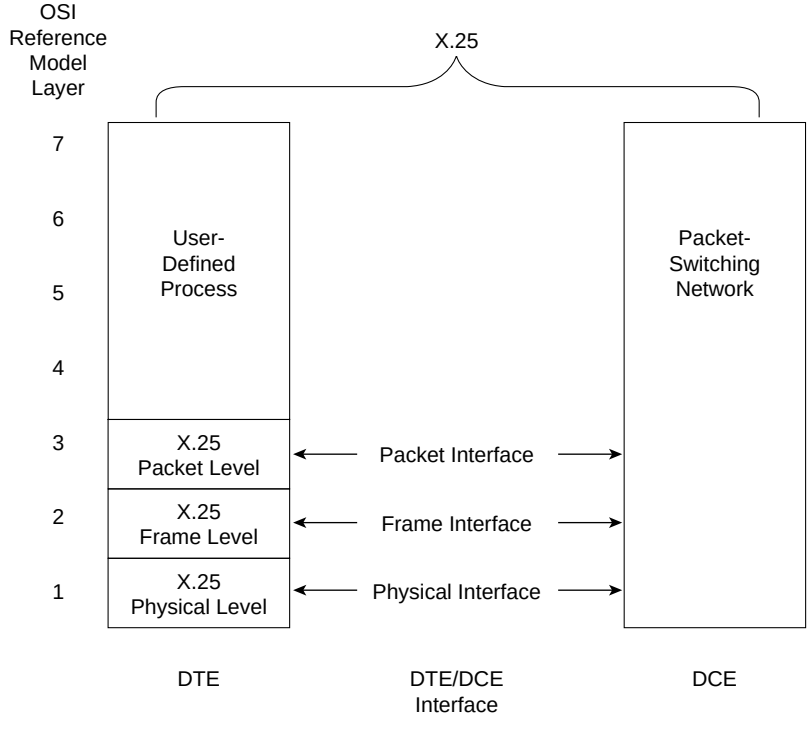
Figure 2-34 *PSE and DCE devices make up the X.25 service cloud.*



A DTE can be a terminal that does not implement the complete X.25 functionality. A DTE is connected to a DCE through a translation device called a packet assembler/disassembler (PAD). The operation of the terminal-to-PAD interface, the services offered by the PAD, and the interaction between the PAD and the host are defined by ITU-T Recommendations X.28, X.3, and X.29, respectively.

The X.25 specification maps to Layers 1 through 3 of the OSI reference model. Layer 3 X.25 describes packet formats and packet exchange procedures between peer Layer 3 entities. Layer 2 X.25 is implemented by LAPB. LAPB defines packet framing for the DTE/DCE link. Layer 1 X.25 defines the electrical and mechanical procedures for activating and deactivating the physical medium connecting the DTE and the DCE. This relationship is shown in Figure 2-35. Note that Layers 2 and 3 are also referred to as the ISO standards ISO 7776 (LAPB) and ISO 8208 (the X.25 packet layer).

Figure 2-35 *X.25 has references to the lower three layers of the OSI reference model.*

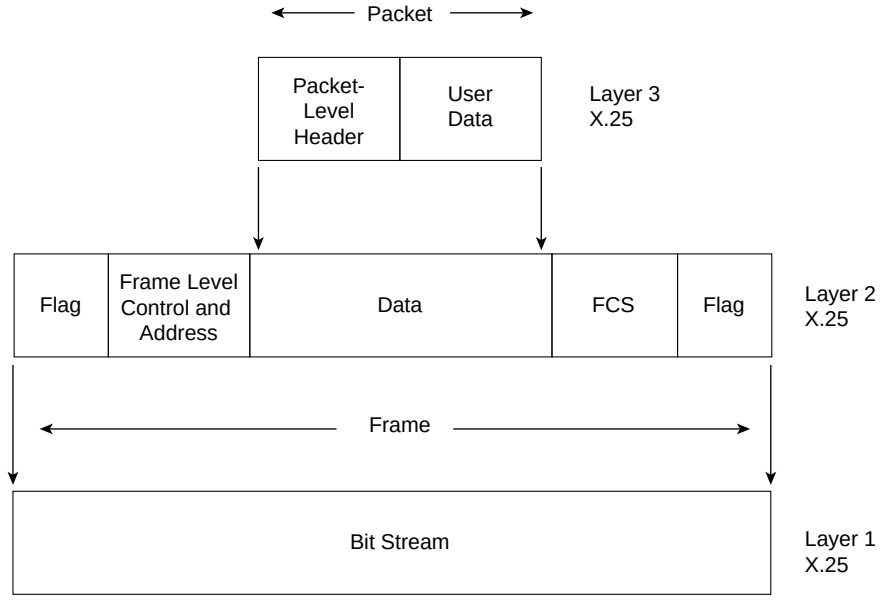


End-to-end communication between DTEs is accomplished through a bidirectional association called a virtual circuit. Virtual circuits permit communication between distinct network elements through any number of intermediate nodes without the dedication of portions of the physical medium that characterizes physical circuits. Virtual circuits can either be permanent or switched (temporary). Permanent virtual circuits are commonly called PVCs; switched virtual circuits are commonly called SVCs. PVCs are typically used for frequent data transfers, and SVCs are used for sporadic data transfers. Layer 3 X.25 is concerned with end-to-end communication involving both PVCs and SVCs.

After a virtual circuit is established, the DTE sends a packet to the other end of the connection by sending it to the DCE using the proper virtual circuit. The DCE looks at the virtual circuit number to determine how to route the packet through the X.25 network. The Layer 3 X.25 protocol multiplexes between all the DTEs served by the DCE on the destination side of the network, and the packet is delivered to the destination DTE.

An X.25 frame is composed of a series of fields, as shown in Figure 2-36. Layer 3 X.25 fields make up an X.25 packet and include a header and user data. Layer 2 X.25 (LAPB) fields include frame-level control and addressing fields, the embedded Layer 3 packet, and an FCS.

Figure 2-36 *X.25 frames contain control information for reliability and windowing.*

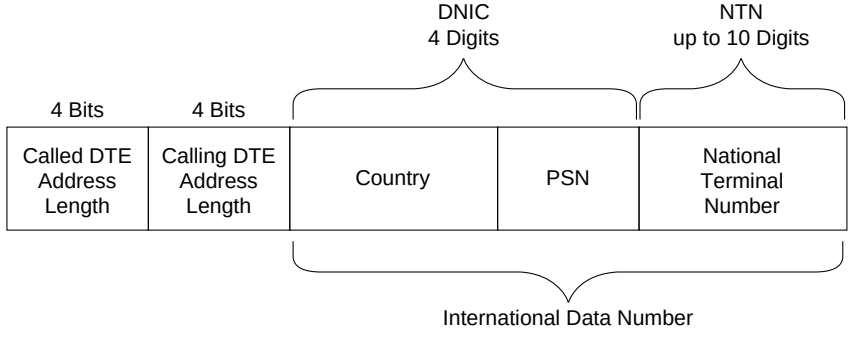


Layer 3 X.25

The Layer 3 X.25 header is made up of a GFI, a logical channel identifier (LCI), and a packet type identifier (PTI). The GFI is a 4-bit field that indicates the general format of the packet header. The LCI is a 12-bit field that identifies the virtual circuit. The LCI is locally significant at the DTE/DCE interface. In other words, the PDN connects two logical channels, each with an independent LCI, on two DTE/DCE interfaces to establish a virtual circuit. The PTI field identifies one of X.25's 17 packet types.

Addressing fields in call setup packets provide source and destination DTE addresses. These are used to establish the virtual circuits that constitute X.25 communication. ITU-T Recommendation X.121 specifies the source and destination address formats. X.121 addresses (also referred to as international data numbers, or IDNs) vary in length and can be up to 14 decimal digits long. Byte 4 in the call setup packet specifies the source DTE and destination DTE address lengths. The first four digits of an IDN are called the data network identification code (DNIC). The DNIC is divided into two parts: the first (three digits) specifying the country and the last specifying the PSN itself. The remaining digits are called the national terminal number (NTN) and are used to identify the specific DTE on the PSN. Figure 2-37 shows the X.121 address format.

Figure 2-37 *X.25 uses the X.121 address format.*



The addressing fields that make up the X.121 address are necessary only when an SVC is used, and then only during call setup. After the call is established, the PSN uses the LCI field of the data packet header to specify the particular virtual circuit to the remote DTE.

Layer 3 X.25 uses three virtual circuit operational procedures:

- Call setup
- Data transfer
- Call clearing

Execution of these procedures depends on the virtual circuit type being used. For a PVC, Layer 3 X.25 is always in data transfer mode because the circuit has been permanently established. If an SVC is used, all three procedures are used.

Data transfer is affected by data packets. Layer 3 X.25 segments and reassembles user messages if they are too long for the circuit's maximum packet size. Each data packet is given a sequence number so error and flow control can occur across the DTE/DCE interface.

Layer 2 X.25

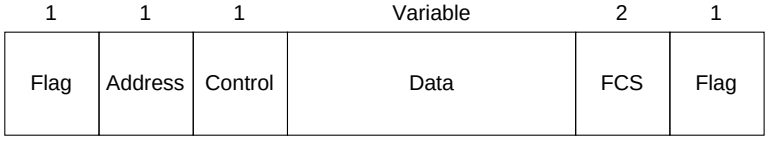
Layer 2 X.25 is implemented by LAPB. LAPB allows both sides—the DTE and the DCE—to initiate communication with the other. During information transfer, LAPB checks that the frames arrive at the receiver in the correct sequence and error free.

As with similar link-layer protocols, LAPB uses three frame format types:

- Information (I) frame—These frames carry upper-layer information and some control information (necessary for full-duplex operation). Send and receive sequence numbers and the P/F bit perform flow control and error recovery. The send sequence number refers to the number of the current frame. The receive sequence number records the number of the frame to be received next. In full-duplex conversation, both the sender and the receiver keep send and receive sequence numbers. The poll bit is used to force a final bit message in response; this is used for error detection and recovery.
- Supervisory (S) frames—These frames provide control information. They request and suspend transmission, report on status, and acknowledge the receipt of I frames. They do not have an information field.
- Unnumbered (U) frames—These frames, as the name suggests, are not sequenced. They are used for control purposes. For example, they can initiate a connection using standard or extended windowing (modulo 8 versus 128), disconnect the link, report a protocol error, or perform other similar functions.

Figure 2-38 shows the LAPB frame.

Figure 2-38 *LAPB is used by X.25 for Layer 2 encapsulation.*



The flag fields delimit the LAPB frame. Bit stuffing is used to ensure that the flag pattern does not occur within the body of the frame.

The address field indicates whether the frame carries a command or a response.

The control field provides further qualifications of command and response frames, and also indicates the frame format (U, I, or S), frame function (for example, receiver ready or disconnect), and the send/receive sequence number.

The data field carries upper-layer data. Its size and format vary depending on the Layer 3 packet type. The maximum length of this field is set by agreement between a PSN administrator and the subscriber at subscription time. The FCS field ensures the integrity of the transmitted data.

Layer 1 X.25

Layer 1 X.25 uses the X.21bis physical layer protocol, which is roughly equivalent to EIA/TIA-232-C. The X.21bis protocol was derived from ITU-T Recommendations V.24 and V.28, which identify the interchange circuits and electrical characteristics (respectively) of a DTE-to-DCE interface. X.21 bis supports point-to-point connections, speeds of up to 19.2 kbps, and synchronous, full-duplex transmission over four-wire media (with speeds of up to 64 kbps). The maximum distance between DTE and DCE is 15 meters.

For more information on X.25, refer to Appendix C, "References and Recommended Reading."

Frame Relay

Frame Relay was originally conceived as a protocol for use over ISDN interfaces, and initial proposals to this effect were submitted to the CCITT in 1984. Work on Frame Relay was also undertaken in the ANSI-accredited T1S1 standards committee in the United States.

There was a major development in Frame Relay's history in 1990 when Cisco Systems, StrataCom, Northern Telecom, and Digital Equipment Corporation formed a consortium to focus Frame Relay technology development and accelerate the introduction of interoperable Frame Relay products. This consortium developed a specification conforming to the basic Frame Relay protocol being discussed in T1S1 and CCITT but extended it with features that provide additional capabilities for complex internetworking environments. These Frame Relay extensions are referred to collectively as the Local Management Interface (LMI).

Frame Relay exists at Layer 2 of the OSI model. It provides a packet-switching data communications capability that is used across the interface between user devices (for example, routers, bridges, host machines) and network equipment (for example, switching nodes). User devices are often referred to as DTEs, and network equipment that interfaces to a DTE is often referred to as a DCE. The network providing the Frame Relay interface can be either a carrier-provided public network or a network of privately owned equipment serving a single enterprise.

As an interface to a network, Frame Relay is the same type of protocol as X.25. However, Frame Relay differs significantly from X.25 in its functionality and format. In particular, Frame Relay is a more streamlined protocol, facilitating higher performance and greater efficiency.

As an interface between user and network equipment, Frame Relay provides a means for statistically multiplexing many logical data conversations (referred to as virtual circuits) over a single physical transmission link. This contrasts with systems that use only time-division multiplexing (TDM) techniques for supporting multiple data streams. Frame Relay's statistical multiplexing provides more flexible and efficient use of available bandwidth. It can be used without TDM techniques or on top of channels provided by TDM systems.

Another important characteristic of Frame Relay is that it exploits the recent advances in WAN transmission technology. Earlier WAN protocols such as X.25 were developed when analog transmission systems and copper media were predominant. These links are much less reliable than the fiber media/digital transmission links available today. Over links such as these,

link-layer protocols can forego time-consuming error correction algorithms, leaving these to be performed at higher protocol layers. Greater performance and efficiency are therefore possible without sacrificing data integrity. Frame Relay is designed with this approach in mind. It includes a CRC algorithm for detecting corrupted bits (so the data can be discarded), but it does not include any protocol mechanisms for correcting bad data (for example, by retransmitting it at this level of protocol).

Another difference between Frame Relay and X.25 is the absence of explicit, per-virtual-circuit flow control in Frame Relay. Now that many upper-layer protocols are effectively executing their own flow control algorithms, the need for this functionality at the data link layer has diminished. Frame Relay, therefore, does not include explicit flow control procedures that duplicate those in higher layers. Instead, very simple congestion notification mechanisms are provided to allow a network to inform a user device that the network resources are close to a congested state. This notification can alert higher-layer protocols that flow control may be needed.

Frame Relay provides connection-oriented data link-layer communication. This service is implemented using virtual circuits. A Frame Relay virtual circuit is a logical connection created between two DTE devices across a Frame Relay PSN. Virtual circuits provide a bidirectional communications path from one DTE device to another. They are uniquely identified by the data link connection identifier (DLCI). A virtual circuit can pass through any number of intermediate DCE devices (switches) located within the Frame Relay PSN. A number of virtual circuits can be multiplexed into a single physical circuit for transmission across the network. Frame Relay virtual circuits can be either SVCs or PVCs.

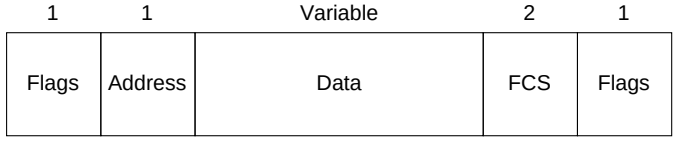
In addition to the basic Frame Relay protocol functions for transferring data, the consortium Frame Relay specification includes LMI extensions that make supporting large, complex internetworks easier. Some LMI extensions are referred to as “common” and are expected to be implemented by everyone who adopts the specification. Other LMI functions are referred to as “optional.” A summary of the LMI extensions follows:

- Virtual circuit status messages (common)—Provide communication and synchronization between the network and the user device, periodically reporting the existence of new PVCs and the deletion of already existing PVCs, and generally providing information about PVC integrity. Virtual circuit status messages prevent the sending of data into black holes, that is, over PVCs that no longer exist.
- Multicasting (optional)—Allows a sender to transmit a single frame but have it delivered by the network to multiple recipients. Thus, multicasting supports the efficient conveyance of routing protocol messages and address resolution procedures that typically must be sent to many destinations simultaneously.
- Global addressing (optional)—Gives connection identifiers global rather than local significance, allowing them to be used to identify a specific interface to the Frame Relay network. Global addressing makes the Frame Relay network resemble a LAN in terms of addressing; address resolution protocols therefore perform over Frame Relay exactly as they do over a LAN.

- Simple flow control (optional)—Provides for an XON/XOFF flow control mechanism that applies to the entire Frame Relay interface. It is intended for those devices whose higher layers cannot use the congestion notification bits and that need some level of flow control.

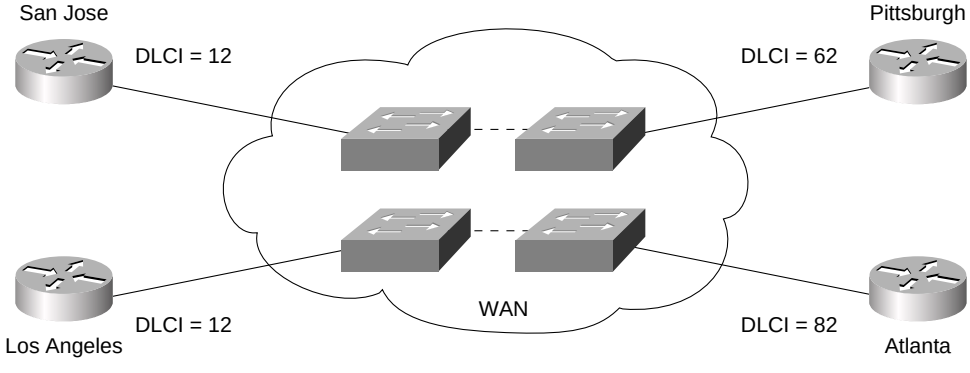
Figure 2-39 shows the Frame Relay frame. Flags delimit the frame's beginning and end. Following the leading flags are 2 bytes of address information. Ten bits of these 2 bytes make up the actual circuit ID (called the DLCI, or data link connection identifier).

Figure 2-39 *The Frame Relay frame format was defined by the Internet Engineering Task Force.*



The 10-bit DLCI value is the heart of the Frame Relay address field in the header. It identifies the logical connection that is multiplexed into the physical channel. In the basic (that is, not extended by the LMI) mode of addressing, DLCIs have local significance; that is, the end devices at two different ends of a connection may use a different DLCI to refer to that same connection. Figure 2-40 provides an example of the use of DLCIs in nonextended Frame Relay addressing.

Figure 2-40 *DLCIs are used for addressing PVCs in Frame Relay networks.*



In Figure 2-40, assume that there are two PVCs, one between Atlanta and Los Angeles, and one between San Jose and Pittsburgh. Los Angeles may refer to its PVC with Atlanta using DLCI = 12, and Atlanta refers to the same PVC with DLCI = 82. Similarly, San Jose may refer to its PVC with Pittsburgh using DLCI = 12. The network uses internal proprietary mechanisms to keep the two locally significant PVC identifiers distinct.

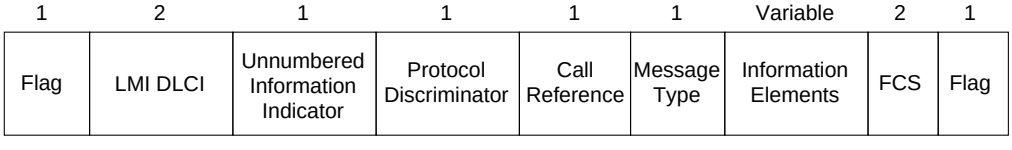
At the end of each DLCI byte is an extended address (EA) bit. If this bit is 1, the current byte is the last DLCI byte. All implementations currently use a 2-byte DLCI, but the presence of the EA bits means that longer DLCIs may be agreed upon and used in the future.

Three bits in the 2-byte DLCI are fields related to congestion control. The forward explicit congestion notification (FECN) bit is set by the Frame Relay network in a frame to tell the DTE receiving that frame that congestion was experienced in the path from source to destination. The backward explicit congestion notification (BECN) bit is set by the Frame Relay network in frames traveling in the opposite direction from frames encountering a congested path. The notion behind both of these bits is that the FECN or BECN indication can be promoted to a higher-level protocol that can take flow control action as appropriate. (FECN bits are useful to higher-layer protocols that use receiver-controlled flow control, and BECN bits are significant to those that depend on "emitter controlled" flow control.)

The discard eligibility (DE) bit is set by the DTE to tell the Frame Relay network that a frame has lower importance than other frames and should be discarded before other frames if the network becomes short of resources. Thus, it represents a very simple priority mechanism. This bit is usually set only when the network is congested.

The previous section described the basic Frame Relay protocol format for carrying user data frames. The consortium's Frame Relay specification also includes the LMI procedures. LMI messages are sent in frames distinguished by an LMI-specific DLCI (defined in the consortium specification as DLCI = 1023). Figure 2-41 shows the LMI message format.

Figure 2-41 *LMI messages are used for signaling between Frame Relay switches and end devices.*



In LMI messages, the basic protocol header is the same as in normal data frames. The actual LMI message begins with four mandatory bytes, followed by a variable number of information elements (IEs). The format and encoding of LMI messages is based on the ANSI T1S1 standard.

The first of the mandatory bytes (unnumbered information indicator) has the same format as the LAPB unnumbered information (UI) frame indicator with the poll/final bit set to zero. The next byte is referred to as the protocol discriminator, which is set to a value that indicates LMI. The third mandatory byte (call reference) is always filled with zeros.

The final mandatory byte is the message type field. Two message types have been defined:

- Status-inquiry messages allow the user device to inquire about network status. Status messages respond to status-inquiry messages.

- Keepalives (messages sent through a connection to ensure that both sides will continue to regard the connection as active) and PVC status messages are examples of these messages and are the common LMI features that are expected to be a part of every implementation that conforms to the consortium specification.

Together, status and status-inquiry messages help verify the integrity of logical and physical links. This information is critical in a routing environment because routing algorithms make decisions based on link integrity.

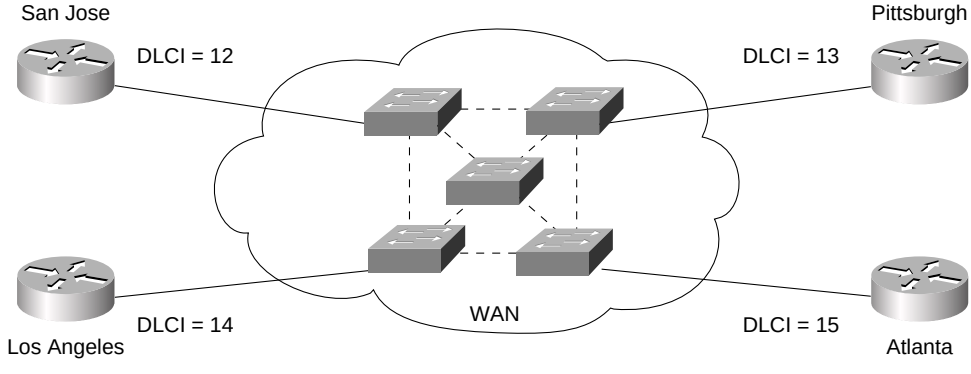
Following the message type field is some number of IEs. Each IE consists of a single-byte IE identifier, an IE length field, and one or more bytes containing actual data.

Frame Relay Global Addressing

In addition to the common LMI features, several optional LMI extensions are useful in an internetworking environment. The first important optional LMI extension is global addressing. As noted earlier, the basic (nonextended) Frame Relay specification supports only values of the DLCI field that identify PVCs with local significance. In this case, there are no addresses that identify network interfaces or nodes attached to these interfaces. Because these addresses do not exist, they cannot be discovered by traditional address resolution and discovery techniques. This means that with normal Frame Relay addressing, static maps must be created to tell routers which DLCIs to use to find a remote device and its associated internetwork address.

The global addressing extension permits node identifiers. With this extension, the values inserted in the DLCI field of a frame are globally significant addresses of individual end-user devices (for example, routers). This is implemented as shown in Figure 2-42.

Figure 2-42 *Globally significant DLCIs must be unique across the network.*



In Figure 2-42, note that each interface has its own identifier. Suppose Pittsburgh must send a frame to San Jose. San Jose's identifier is 12, so Pittsburgh places the value 12 in the DLCI field and sends the frame into the Frame Relay network. At the exit point, the DLCI field contents are changed by the network to 13 to reflect the frame's source node. As each router's interface has a distinct value as its node identifier, individual devices can be distinguished. This permits adaptive routing in complex environments.

Global addressing provides significant benefits in a large, complex internetwork. The Frame Relay network now appears to the routers on its periphery like a typical LAN. No changes to higher-layer protocols are needed to take full advantage of higher-layer protocol capabilities.

Frame Relay Multicasting

Multicasting is another valuable optional LMI feature. Multicast groups are designated by a series of four reserved DLCI values (1019 to 1022). Frames sent by a device using one of these reserved DLCIs are replicated by the network and sent to all exit points in the designated set. The multicasting extension also defines LMI messages that notify user devices of the addition, deletion, and presence of multicast groups.

In networks that take advantage of dynamic routing, routing information must be exchanged among many routers. Routing messages can be sent efficiently by using frames with a multicast DLCI. This allows messages to be sent to specific groups of routers.

Frame Relay can be used as an interface to either a publicly available carrier-provided service or to a network of privately owned equipment. A typical means of private network implementation is to equip traditional T1 multiplexers with Frame Relay interfaces for data devices, as well as non-Frame Relay interfaces for other applications such as voice and videoteleconferencing. Figure 2-43 shows this configuration.

A public Frame Relay service is deployed by putting Frame Relay switching equipment in the central offices of a telecommunications carrier. In this case, users can gain economic benefits from traffic-sensitive charging rates and are relieved from the work necessary to administer and maintain the network equipment and service.

In either type of network, the lines that connect user devices to the network equipment can operate at a speed selected from a broad range of data rates. Speeds between 56 kbps and 2 Mbps are typical, although Frame Relay can support lower and higher speeds.

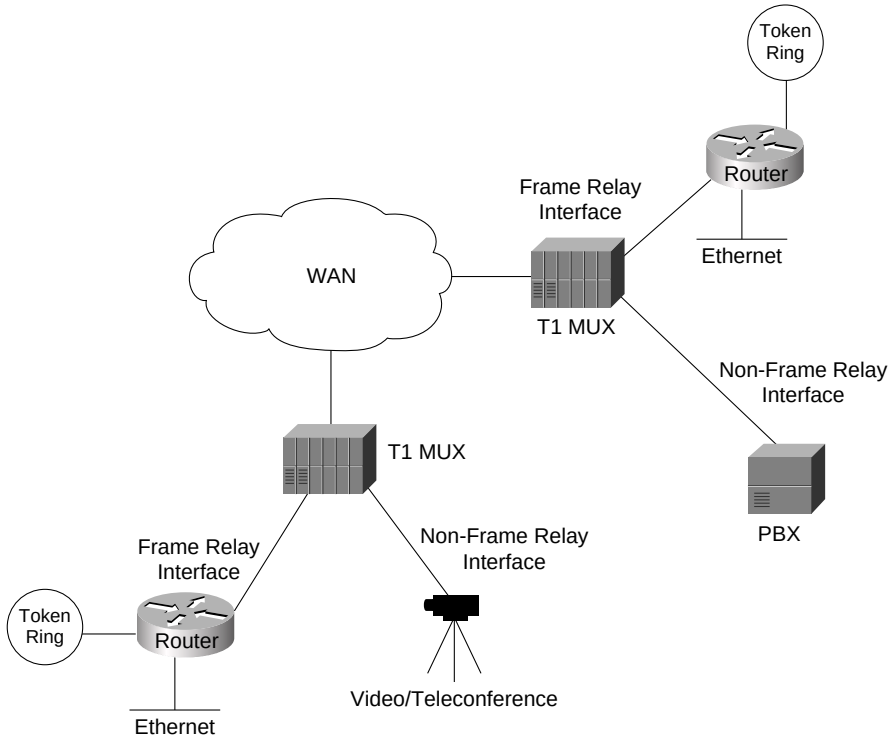
ISDN

ISDN refers to a set of communication protocols implemented by telephone companies to permit telephone networks to carry digitized voice, data, text, graphics, music, and video to end users over existing telephone systems. ISDN services are offered by many carriers under tariff.

ISDN is generally viewed as an alternative to Frame Relay and T1 wide-area telephone services (WATS). In practical terms, ISDN has evolved into one of the leading technologies for facilitating telecommuting arrangements and internetworking small, remote offices into corporate campuses.

ISDN is addressed by a suite of ITU-T standards, spanning the physical, data link, and network layers of the seven-layer OSI reference model.

Figure 2-43 *Hybrid Frame Relay networks combine data, voice, and video services.*



The ISDN BRI service provides two bearer channels and one data channel. The BRI B-channel service operates at 64 kbps and carries data, and the BRI D-channel service operates at 16 kbps and usually carries control and signaling information. The total bit rate is 144 kbps.

The ISDN Primary Rate Interface (PRI) service delivers 23 B channels and one 64-kbps D channel in North America and Japan for a total bit rate of up to 1.544 Mbps. In Europe, Australia, and other parts of the world, ISDN provides 30 B channels and one 64-kbps D channel, for a total bit rate of up to 2.048 Mbps.

There are three principal categories of ISDN network components:

- ISDN terminal equipment
- ISDN termination devices
- ISDN reference points

ISDN Terminal Equipment

ISDN specifies two basic terminal equipment types:

- Terminal Equipment Type 1 (TE1)—A TE1 is a specialized ISDN terminal, including computer equipment or telephones. It is used to connect to ISDN through a four-wire, twisted-pair digital link.
- Terminal Equipment Type 2 (TE2)—A TE2 is a non-ISDN terminal such as DTE that predates the ISDN standards. A TE2 connects to ISDN through a terminal adapter (TA). An ISDN TA can be either a standalone device or a board inside the TE2.

ISDN Network Termination Devices

ISDN specifies a type of intermediate equipment called a network termination (NT) device. NTs connect the four-wire subscriber wiring to two-wire local loops. There are three supported NT types:

- NT Type 1 (NT1) device—An NT1 device is treated as customer premises equipment (CPE) in North America, but is provided by carriers elsewhere.
- NT Type 2 (NT2) device—An NT2 device is typically found in digital private branch exchanges (PBXs). An NT2 performs Layer 2 and 3 protocol functions and concentration services.
- NT Type 1/2 (NT1/2) device—An NT1/2 device provides combined functions of separate NT1 and NT2 devices. An NT1/2 is compatible with NT1 and NT2 devices and is used to replace separate NT1 and NT2 devices.

ISDN Reference Points

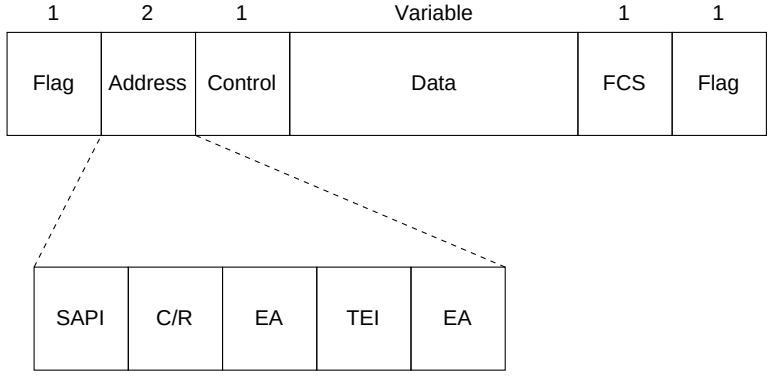
ISDN reference points define logical interfaces. Four reference points are defined in ISDN:

- R reference point—Defines the reference point between non-ISDN equipment and a TA.
- S reference point—Defines the reference point between user terminals and an NT2.

- T reference point—Defines the reference point between NT1 and NT2 devices.
- U reference point—Defines the reference point between NT1 devices and line-termination equipment in a carrier network. (This is only in North America, where the NT1 function is not provided by the carrier network.)

The data link layer of the ISDN signaling protocol is Link Access Procedure on the D channel (LAPD). LAPD is similar to the HDLC and LAPB specifications. LAPD is used to ensure that control and signaling information flows and is received properly. The LAPD frame format uses supervisory, information, and unnumbered frames. Figure 2-44 illustrates the fields associated with the ISDN data link-layer frame.

Figure 2-44 *ISDN uses LAPD for data link-layer signaling.*



The LAPD flag, control, and FCS fields are identical to those of HDLC. The LAPD address field can be either 1 or 2 bytes long. If the EA bit of the first byte is set, the address is 1 byte; if it is not set, the address is 2 bytes. The first address field byte contains the service access point identifier (SAPI), which is a 6-bit field that identifies the point at which LAPD services are provided to Layer 3. The C/R bit indicates whether the frame contains a command or a response. The terminal endpoint identifier (TEI) field identifies either a single terminal or multiple terminals. A TEI of all ones indicates a broadcast.

For more information on ISDN, refer to Appendix C, “References and Recommended Reading.”

OSI Layers 3 to 7 Characteristics—Routed Protocols and Routing Protocols

OSI Layers 3 through 7 (network, transport, session, presentation, and application layers) are defined as the *upper-layers* of the OSI reference model. This section covers the following protocol suites:

- TCP/IP
- NetWare
- AppleTalk

TCP/IP

In the mid-1970s, the Defense Advanced Research Projects Agency (DARPA) became interested in establishing a packet-switched network to provide communications between research institutions in the United States. DARPA and other government organizations understood the potential of packet-switched technology and were just beginning to face the problem virtually all companies with networks now have—communication between dissimilar computer systems.

With the goal of heterogeneous connectivity in mind, DARPA funded research by Stanford University and Bolt, Beranek, and Newman (BBN) to create a series of communication protocols. The result of this development effort, completed in the late 1970s, was the Internet protocol suite, of which TCP and IP are the two best known.

The Internet protocols can be used to communicate across any set of interconnected networks. They are equally well suited for LAN and WAN communications. The Internet suite includes not only lower-layer specifications (such as TCP and IP), but also specifications for such common applications as mail, terminal emulation, and file transfer. Figure 2-45 shows some of the most important Internet protocols and their relationship to the OSI reference model.

Creation and documentation of the Internet protocols closely resembles an academic research project. The protocols are specified in RFCs, which are published and then reviewed and analyzed by the Internet community. Protocol refinements are published in new RFCs. Taken together, the RFCs provide a colorful history of the people, companies, and trends that shaped the development of what is today the world's most popular open system protocol suite.

The TCP/IP Network Layer

IP is the primary Layer 3 protocol in the Internet suite. In addition to internetwork routing, IP provides fragmentation and reassembly of datagrams and error reporting. Along with TCP, IP represents the heart of the Internet protocol suite. Figure 2-46 shows the IP packet format.

Figure 2-45 *The TCP/IP stack combines the upper three layers of the OSI reference model into one application layer.*

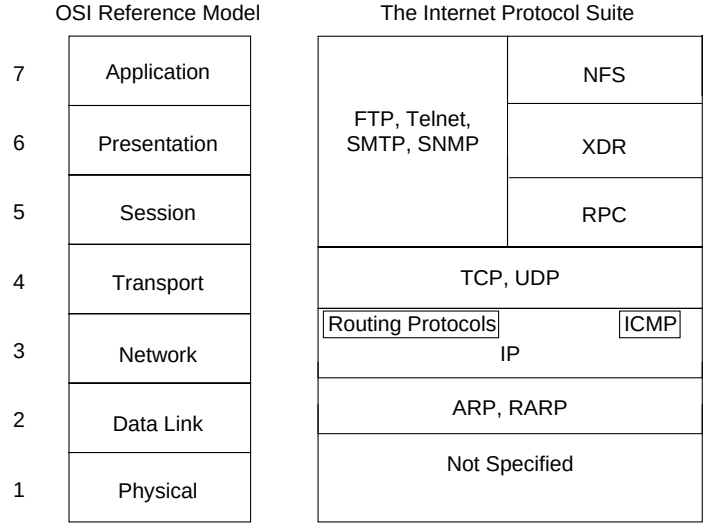
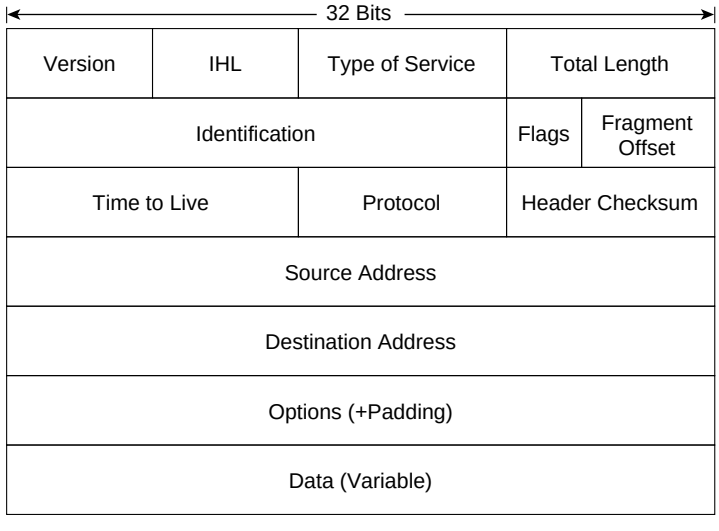


Figure 2-46 *The IP packet header includes the source and destination addresses required for routing IP packets.*



The IP header begins with a 4-bit version number, which indicates the version of IP currently used. The current version of IP is 4, so IP is sometimes called IPv4.

NOTE IPv6 is the next generation of the IP protocol. (Actually, there is an entire set of protocols to support the version 6 protocol level.) The key changes seen in IPv6 are in the following areas:

- IP address—IPv6 addresses are 16 bytes long and are not class based.
- Configuration—IPv6 has many of the autoconfiguration characteristics of IPX.
- Security—IPv6 has added levels of security to fix some of the inadequacies of IPv4.

This section focuses on IPv4, not IPv6.

The 4-bit IP header length (IHL) field indicates the datagram header length in 32-bit words.

The type-of-service (TOS) field is an 8-bit field that specifies how a particular upper-layer protocol wants the current datagram to be handled. Datagrams can be assigned various levels of importance through this field.

The 16-bit total length field specifies the length of the entire IP packet, including data and header, in bytes.

The 16-bit identification field contains an integer that identifies the current datagram. This field is used to help piece together datagram fragments.

The flags field is a 3-bit field of which the low-order 2 bits control fragmentation. The high-order bit is set to 0. The first low-order bit specifies whether the packet can be fragmented (0 = may fragment, 1 = don't fragment); the second bit specifies whether the packet is the last fragment in a series of fragmented packets (0 = last fragment, 1 = more fragments).

The 13-bit fragment offset field indicates where in the entire datagram this fragment belongs and is measured in 64-bit units from the beginning of the datagram.

The 8-bit time-to-live field maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.

The protocol field indicates which upper-layer protocol receives incoming packets after IP processing is complete. Examples include those listed in Table 2-2.

Table 2-2 *IP protocol field values.*

Decimal	Keyword	Description
1	ICMP	Internet Control Message Protocol
2	IGMP	Internet Group Management Protocol
6	TCP	Transmission Control Protocol
8	EGP	Exterior Gateway Protocol
9	IGRP	Interior Gateway Routing Protocol
16	CHAOS	Chaos
17	UDP	User Datagram Protocol
22	XNS-IDP	XNS Internetwork Datagram Protocol
29	ISO-TP4	ISO Transport Protocol Class 4
80	ISO-IP	ISO Internet Protocol
83	VINES	Virtual Integrated Network Service

The 16-bit header checksum field helps ensure IP header integrity.

The 32-bit source and destination address fields specify the IP address of sending and receiving hosts.

The variable-length options field allows IP to support options such as security and record route. The options field must end on a 32-bit boundary. Padding can be added to ensure this.

The data field contains upper-layer information.

TCP/IP Addressing

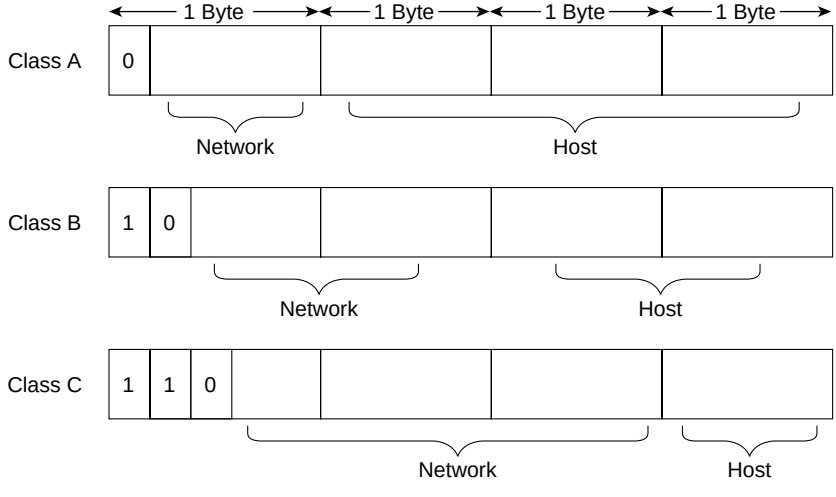
As with all network-layer protocols, IP's addressing scheme is integral to the process of routing IP datagrams through an internetwork. An IP address is 32 bits in length, divided into either two or three parts. The first part designates the network address, the second part (if present) designates the subnet address, and the final part designates the host address. Subnet addresses are only present if the network administrator has decided that the network should be divided into subnetworks. The lengths of the network, subnet, and host fields are all variable.

IP addressing supports five different network classes (the leftmost bits indicate the network class):

- Class A networks are intended mainly for use with a few very large networks, because they provide only 8 bits for the network address field.
- Class B networks allocate 16 bits for the network address field and 16 bits for the host address field. This address class offers a good compromise between network and host address space.
- Class C networks allocate 24 bits for the network address field. Class C networks provide only 8 bits for the host field, however, so the number of hosts per network may be a limiting factor.
- Class D addresses are reserved for multicast groups, as described formally in RFC 1112. In Class D addresses, the 4 highest-order bits are set to 1, 1, 1, and 0.
- Class E addresses are also defined by IP but are reserved for future use. In Class E addresses, the 4 highest-order bits are all set to 1.

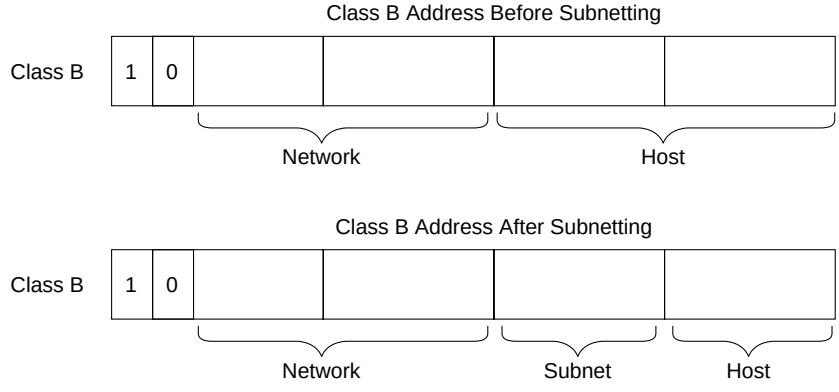
IP addresses are written in dotted-decimal format—for example, 34.10.2.1. Figure 2-47 shows the address formats for Class A, B, and C IP networks.

Figure 2-47 *The leftmost bits of an IP address determine its class.*



IP networks can also be divided into smaller units, called *subnets*. Subnets provide extra flexibility for network administrators. For example, assume that a network has been assigned a Class B address and all the nodes on the network currently conform to a Class B address format. Then, assume that the dotted-decimal representation of this network's address is 128.10.0.0 (all zeros in the host field of an address specifies the entire network). Rather than change all the addresses to some other basic network number, the administrator can subdivide the network by using subnetting. This is done by borrowing bits from the host portion of the address and using them as a subnet field, as shown in Figure 2-48.

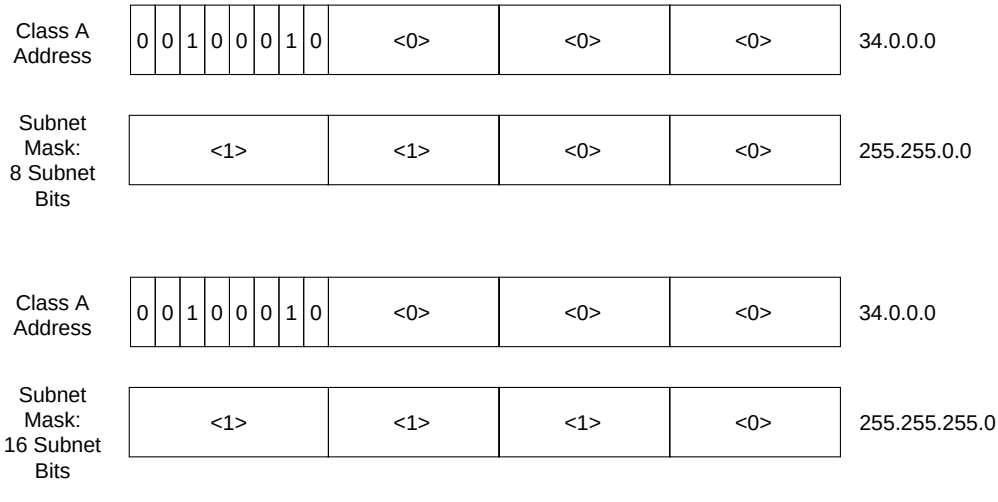
Figure 2-48 *With subnetting, host bits are turned into subnet bits and become part of the subnet addresses.*



If a network administrator has chosen to use 8 bits of subnetting, the third octet of a Class B IP address provides the subnet number. In our example, address 128.10.1.0 refers to network 128.10, subnet 1; address 128.10.2.0 refers to network 128.10, subnet 2; and so on.

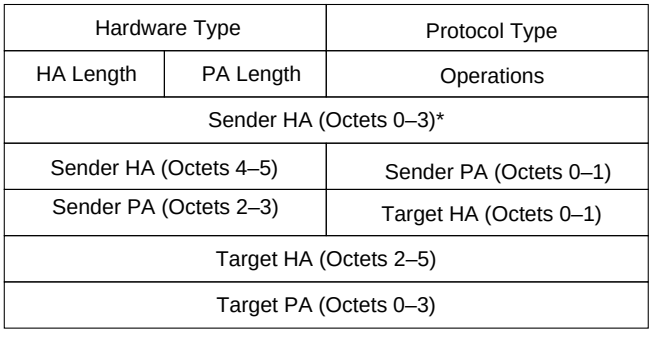
The number of bits borrowed for the subnet address is variable. To specify how many bits are used, IP provides the subnet mask. Subnet masks use the same format and representation technique as IP addresses. Subnet masks have ones in all bits except those bits that specify the host field. For example, the subnet mask that specifies 8 bits of subnetting for Class A address 34.0.0.0 is 255.255.0.0. The subnet mask that specifies 16 bits of subnetting for Class A address 34.0.0.0 is 255.255.255.0. Both of these subnet masks are shown in Figure 2-49.

Figure 2-49 *Ones in the subnet mask indicate the network portion of the address.*



On some media (such as IEEE 802 LANs), media addresses and IP addresses are dynamically discovered through the use of two other members of the Internet protocol suite: ARP and RARP. ARP uses broadcast messages to determine the hardware MAC-layer address corresponding to a particular internetwork address. ARP is sufficiently generic to allow use of IP with virtually any type of underlying media-access mechanism. RARP uses broadcast messages to determine the Internet address associated with a particular hardware address. RARP is particularly important to diskless nodes, which may not know their internetwork address when they boot. Figure 2-50 shows the ARP/RARP packet structure.

Figure 2-50 *ARP/RARP packets are used for MAC/IP address resolution.*



*Field Lengths Assume HA = 6 Octets and PA = 4 Octets

HA = Hardware Address
PA = Protocol Address (IP Address)

The 16-bit hardware type field indicates the hardware type for which the request is being made. Examples include the following:

Value	Description
1	Ethernet (10 Mb)
4	Proteon ProNET Token Ring
6	IEEE 802 networks
7	ARCNET
11	LocalTalk

The 16-bit protocol type field is the protocol code or Ethertype. The hexadecimal value 0800 means IP.

The 8-bit hardware address (HA) length and protocol address (PA) length fields are the lengths (in octets), of the hardware and protocol addresses, respectively.

The 16-bit operation field indicates the operation code for this message:

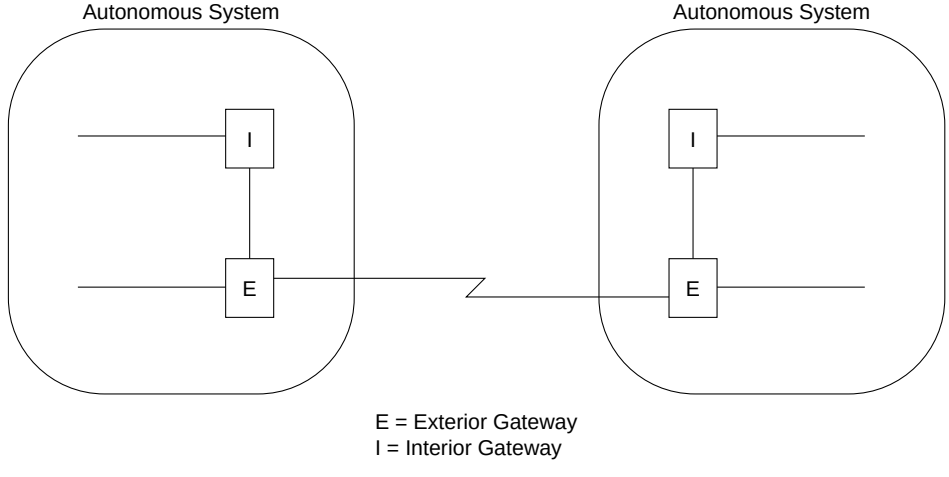
Value	Description
1	ARP request
2	ARP reply
3	RARP request
4	RARP reply

The sender HA and target HA are the 48-bit (for Ethernet) hardware addresses, and the sender PA and target PA are the 32-bit (for IP) protocol addresses.

TCP/IP Internet Routing

Routing devices in the Internet have traditionally been called *gateways*—an unfortunate term because, elsewhere in the industry, the term applies to a device with somewhat different functionality. Gateways (which we call *routers* from this point on) within the Internet are organized hierarchically. Some routers are used to move information through one particular group of networks under the same administrative authority and control. A group of networks and routers under the same administrative control is called an *autonomous system (AS)*. Routers used for information exchange within autonomous systems are called *interior routers*, and they use a variety of IGPs to accomplish this purpose. Routers that move information between autonomous systems are called *exterior routers*, and they use an EGP for this purpose. Figure 2-51 shows the location of interior and exterior gateways.

Figure 2-51 *Exterior gateways are for routing between autonomous systems.*



IP routing protocols are dynamic. Dynamic routing calls for routes to be calculated at regular intervals by software in the routing devices. This contrasts with static routing, where routes are established by the network administrator and do not change until the network administrator changes them. An IP routing table consists of destination address/next hop pairs. A sample entry, shown in Figure 2-52, is interpreted as meaning "to get to Network 34.1.0.0 (Subnet 1 on Network 34), the next stop is the node at address 54.34.23.12."

Figure 2-52 *Routers store the next hops to destination networks in the IP routing table.*

Destination Address	Next Hop
34.1.0.0	54.34.23.12
78.2.0.0	54.34.23.12
147.9.5.0	.
17.12.0.0	.
.	54.32.12.10
.	54.32.12.10
.	.
.	.

IP routing specifies that IP datagrams travel through internetworks one hop at a time. The entire route is not known at the outset of the journey. Instead, at each stop, the next destination is calculated by matching the destination address within the datagram with an entry in the current node's routing table. Each node's involvement in the routing process consists only of forwarding packets based on internal information, regardless of whether the packets get to their final destination. In other words, IP does not provide for error reporting back to the source when routing anomalies occur. This task is left to another Internet protocol, Internet Control Message Protocol (ICMP).

ICMP

ICMP performs a number of tasks within an IP internetwork. The principal reason it was created was for reporting routing failures back to the source. In addition, ICMP provides helpful messages such as the following:

- Echo and reply messages to test node reachability across an internetwork (via the **ping** command)
- Traceroute messages to test node reachability in a step-by-step fashion across an internetwork (via the **trace** command)
- Redirect messages to stimulate more efficient routing

- Time exceeded messages to inform sources that a datagram has exceeded its allocated time to exist within the internetwork
- Router advertisement and router solicitation messages to determine the addresses of routers on directly attached subnetworks

A more recent addition to ICMP provides a way for new nodes to discover the subnet mask currently used in an internetwork. All in all, ICMP is an integral part of any IP implementation, particularly those that run in routers.

NOTE You can use ICMP as a diagnostic tool. By analyzing all ICMP traffic crossing the network, you can isolate routing faults, misconfigurations, security attacks, and more, so it's a good idea to learn ICMP in depth. Refer to RFC 792 for more details on ICMP.

IRDP

ICMP Router Discovery Protocol (IRDP) uses router advertisement and router solicitation messages to discover addresses of routers on directly attached subnets.

With IRDP, each router periodically multicasts router advertisement messages from each of its interfaces. Hosts discover the addresses of routers on the directly attached subnet by listening for these messages. Hosts can use router solicitation messages to request immediate advertisements, rather than wait for unsolicited messages.

IRDP offers several advantages over other methods of discovering addresses of neighboring routers. Primarily, it does not require hosts to recognize routing protocols, nor does it require manual configuration by an administrator.

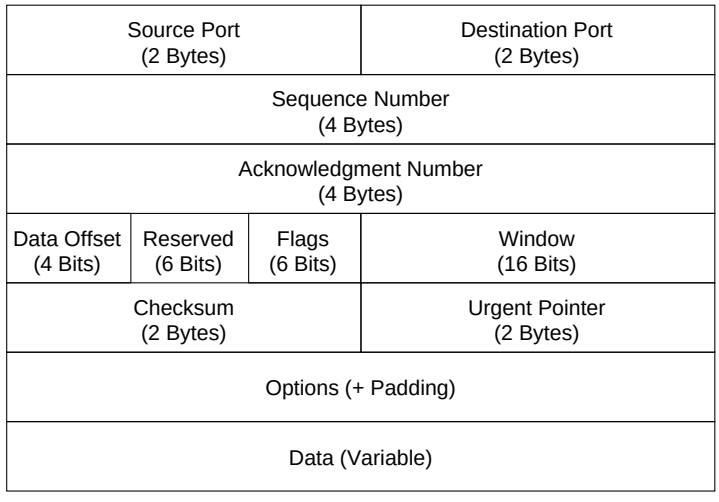
Router advertisement messages allow hosts to discover the existence of neighboring routers, but not which router is best to reach a particular destination. If a host uses a poor first-hop router to reach a particular destination, it receives a redirect message identifying a better choice.

The TCP/IP Transport Layer

The Internet transport layer is implemented by TCP and UDP. TCP provides connection-oriented data transport, and UDP operation is connectionless. This section focuses on TCP first.

TCP TCP provides full-duplex, acknowledged, and flow-controlled service to upper-layer protocols. It moves data in a continuous, unstructured byte stream where bytes are identified by sequence numbers. TCP can also support numerous simultaneous upper-layer conversations. Figure 2-53 shows the TCP packet format.

Figure 2-53 *The TCP segment header fields provide reliability, windowing, and connection orientation.*



The 16-bit source port and destination port fields identify the points at which upper-layer source and destination processes receive TCP services. Examples of TCP and UDP ports include those listed in Table 2-3.

Table 2-3 *TCP and UDP application port numbers.*

Decimal	Keyword	Description
7	ECHO	Echo
9	DISCARD	Discard
11	USERS	Active users
13	DAYTIME	Daytime
17	QUOTE	Quote
20	FTP-DATA	File transfer (data)
21	FTP	File transfer (control)
23	TELNET	Telnet
25	SMTP	Simple Mail Transfer Protocol
37	TIME	Time
42	NAMESERVER	Host name server
43	NICNAME	Who is

Table 2-3 *TCP and UDP application port numbers. (Continued)*

Decimal	Keyword	Description
53	DOMAIN	Domain name server
67	BOOTPS	BOOTP server/DHCP server
68	BOOTPC	BOOTP client/DHCP client
69	TFTP	Trivial File Transfer Protocol
79	FINGER	Finger
80	HTTP	Hypertext Transfer Protocol
102	ISO-TSAP	ISO-TSAP
103	X400	X400
111	SUNRPC	Sun remote-procedure call
137	NETBIOS—NS	NetBIOS name service
138	NETBIOS—DGM	NetBIOS datagram service
139	NETBIOS—SSSN	NetBIOS session service
146	ISO-TP0	ISO-TP0
147	ISO-IP	ISO-IP
161	SNMP	Simple Network Management Protocol
162	SNMPTRAP	SNMPTRAP
163	CMIP-Manage	CMIP/TCP manager
164	CMIP-Agent	CMIP/TCP agent
201	AT-RTMP	AppleTalk routing maintenance
202	AT-NBP	AppleTalk name binding
204	AT-ECHO	AppleTalk echo
206	AT-ZIS	AppleTalk zone information
520	RIP	Routing Information Protocol
524	NetWare over IP	NetWare 5’s pure IP

The 32-bit sequence number field usually specifies the number assigned to the first byte of data in the current message. During session establishment, it can also be used to identify an initial sequence number to be used in the upcoming transmission.

The 32-bit acknowledgment number field contains the sequence number of the next byte of data the sender of the packet expects to receive.

The 4-bit data offset field indicates the number of 32-bit words in the TCP header.

The 6-bit reserved field is reserved for future use by protocol designers, and is set to 0.

The 6-bit flags field carries a variety of control information, such as the setup and termination of a session, expedited or urgent flow, or reset of a connection. The six flags, starting with the most significant bit, are as follows:

Flag	Description
URG	Urgent pointer field significant
ACK	Acknowledgment field significant
PSH	Push function
RST	Reset connection
SYN	Synchronize a new session
FIN	No more data from sender

The 16-bit window field specifies the size of the sender's receive window (that is, buffer space available for incoming data).

The 16-bit checksum field indicates whether the header or data was damaged in transit.

The 16-bit urgent pointer field points to the last byte of urgent data and allows the receiver to determine how much urgent data is coming. (In some implementations, the urgent pointer points to the byte following the last byte of urgent data.)

The variable-length options field specifies various TCP options. It must end on a 32-bit boundary. A padding field can be added to ensure this.

The data field contains upper-layer information.

UDP UDP is a much simpler protocol than TCP and is useful in situations where TCP's powerful reliability mechanisms are not necessary. The UDP header (shown in Figure 2-54) has only four fields: 16-bit source port and destination port, 16-bit length, and 16-bit UDP checksum. The source and destination port fields serve the same functions as they do in the TCP header. The length field specifies the length of the UDP header and data, and the checksum field allows packet integrity checking. The UDP checksum is optional.

Figure 2-54 *The UDP header reflects the low overhead of connectionless services.*



The 16-bit source port and destination port fields identify the points at which upper-layer source and destination processes receive UDP services. The source port value is optional, and when used, it specifies the port to which replies should be sent; if not used, it should be 0. Port numbers for UDP are the same as those for TCP (refer to Table 2-3).

The 16-bit length field contains a count of the total bytes in the UDP datagram, including the header and the user data.

The 16-bit checksum is optional; a value of 0 means that the checksum has not been computed.

TCP/IP Upper-Layer Protocols

The Internet Protocol suite includes many upper-layer protocols representing a wide variety of applications, including network management, file transfer, distributed file services, terminal emulation, and electronic mail. Table 2-4 maps the best-known Internet upper-layer protocols to the applications they support.

Table 2-4 *Internet protocol/application mapping.*

Application	Supported Protocols
File Transfer	File Transfer Protocol
Terminal emulation	Telnet, X Window system
Electronic mail	Simple Mail Transfer Protocol
Network management	Simple Network Management Protocol
Distributed file services	Network File System, external data representation, and remote-procedure call

FTP provides a way to move files between computer systems. Telnet allows virtual terminal emulation. The X Window system is a popular protocol that permits intelligent terminals to communicate with remote computers as if they were directly attached. Simple Mail Transfer Protocol (SMTP) provides an electronic mail transport mechanism. Simple Network Management Protocol (SNMP) is a network management protocol used for reporting

anomalous network conditions and setting network threshold values. Network File System (NFS), external data representation (XDR), and remote-procedure call (RPC) combine to allow transparent access to remote network resources. These and other network applications use the services of TCP/IP and other lower-layer Internet protocols to provide users with basic network services.

RIP

RIP is a routing protocol originally designed for Xerox PARC Universal Protocol (where it was called GWINFO) and used in the XNS protocol suite. RIP became associated with both UNIX and TCP/IP in 1982, when the Berkeley Software Distribution (BSD) version of UNIX began shipping with a RIP implementation referred to as *routed* (pronounced "route-dee"). RIP, which is still a very popular routing protocol in the TCP/IP community, is formally defined in the XNS Internet Transport Protocols publication (1981) and in RFC 1058 (1988).

RIP has been widely adopted by PC manufacturers for use in networking products. For example, AppleTalk's routing protocol RTMP is a modified version of RIP. RIP was also the basis for the routing protocols of network architectures from Novell and Banyan.

Each entry in a RIP routing table provides a variety of information, including the ultimate destination, the next hop on the way to that destination, and a metric. The metric indicates the distance in number of hops to the destination. Other information can also be present in the routing table, including various timers associated with the route. Figure 2-55 shows a typical RIP routing table.

Figure 2-55 *RIP routing tables display distance information based on hop counts.*

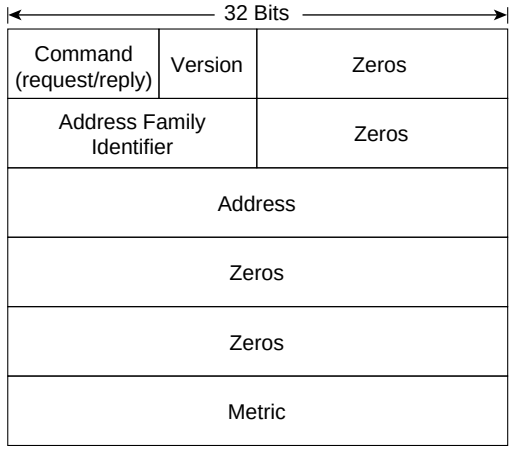
Destination	Next hop	Distance	Timers	Flags
Network A	Router 1	3	t1, t2, t3	x, y
Network B	Router 2	5	t1, t2, t3	x, y
Network C	Router 1	2	t1, t2, t3	x, y
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

RIP maintains only the best route to a destination. When new information provides a better route, this information replaces old route information. Network topology changes can provoke changes to routes, causing, for example, a new route to become the best route to a particular destination. When network topology changes occur, they are reflected in routing update

messages. For example, when a router detects a link failure or a router failure, it recalculates its routes and sends routing update messages. Each router that receives a routing update message that includes a change updates its tables and propagates the change.

Figure 2-56 shows the RIP packet format for IP implementations, as specified by RFC 1058.

Figure 2-56 *The RIP packets carry information on the networks advertised throughout the network.*



The first field in an IP RIP packet is the command field. This field carries an integer indicating either a request (value = 1) or a response (value = 2). The request command requests the responding system to send all or part of its routing table. Destinations for which a response is requested are listed later in the packet. The response command represents a reply to a request or, more frequently, an unsolicited regular routing update. In the response packet, a responding system includes all or part of its routing table. Regular routing update messages include the entire routing table.

The version number field specifies the RIP version being implemented (currently either 1 or 2). With the potential for many RIP implementations in an internetwork, this field can be used to signal different, potentially incompatible, implementations.

Following a 16-bit field of all zeros is the address family identifier field, which specifies the particular address family being used. This is typically IP (value = 2), but other network types can also be represented.

After another 16-bit field of zeros is a 32-bit address field. This field typically contains an IP address.

Following two more 32-bit fields of zeros is the RIP metric, which is a hop count. It indicates how many internetwork hops (routers) must be traversed before the destination can be reached.

Up to 25 occurrences of the address family identifier through metric fields are permitted to occur in any single IP RIP packet. In other words, up to 25 destinations can be listed in any single RIP packet. Multiple RIP packets are used to convey information from larger routing tables.

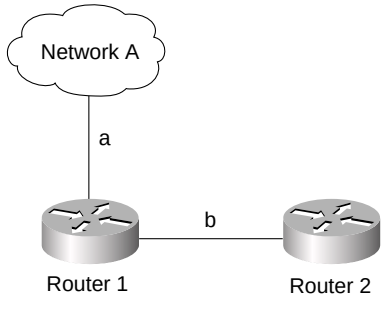
Like other routing protocols, RIP uses certain timers to regulate its performance. The RIP routing update timer is generally set to 30 seconds, ensuring that each router will send a complete copy of its routing table to all neighbors every 30 seconds. The route invalid timer determines how much time must expire without a router having heard about a particular route before that route is considered invalid. When a route is marked invalid, neighbors are notified of this fact. This notification must occur prior to expiration of the route flush timer, which indicates when the route is removed from the routing table. Typical initial values for these timers are 90 seconds for the route invalid timer and 270 seconds for the route flush timer.

RIP implementations can use several features to make operation more stable in the face of rapid network topology changes. These include a hop-count limit, holddowns, triggered updates, split horizon, and poison reverse updates.

RIP Hop-Count Limit RIP permits a maximum hop count of 15. Any destination greater than 15 hops away is tagged as unreachable. RIP's maximum hop count greatly restricts its use in large internetworks but prevents a problem called count to infinity from causing endless network routing loops.

In Figure 2-57, consider what will happen if the link from Router 1 to Network A fails. Router 1 will remove the entry for Network A from its routing table, but in the meantime, Router 2 will advertise that it has a link to Network A. This will cause Router 1 to begin routing all traffic for Network A through Router 2. This creates a routing loop, because Router 2 has in its table that the next hop to Network A is Router 1, and Router 1 has in its table that the next hop to Network A is Router 2. A frame destined for Network A would continue to loop indefinitely except that the IP time-to-live field will finally decrement to 0, causing the frame to be stopped.

Figure 2-57 *Even small networks can experience routing loops.*



The looping frame is not the only problem. The other problem is that when Router 2 advertises a one-hop link to Network A, Router 1 changes its own routing table to show that it has a two-hop path to Network A. Router 1 advertises this path in its next update. This will cause Router 2 to advertise a three-hop path, and so on. This problem will continue indefinitely unless some external boundary condition is imposed. That boundary condition is RIP's hop-count maximum. When the hop count exceeds 15, the route is marked unreachable. Over time, the route is removed from the table.

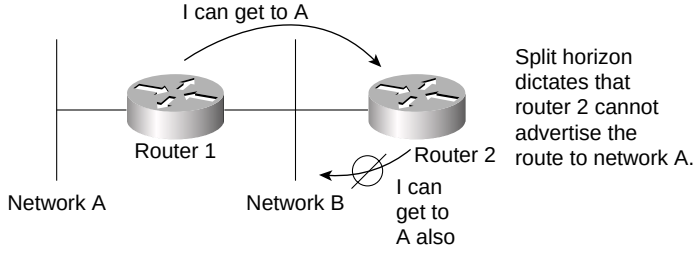
RIP Holddowns and Triggered Updates Holddowns can be used to prevent regular update messages from inappropriately reinstating a route that has gone bad. Holddowns tell routers to hold down any changes that might affect recently removed routes for some period of time. The hold-down period is usually calculated to be just greater than the period of time necessary to update the entire network with a routing change. Holddowns prevent the count-to-infinity problem.

When a route goes down, neighboring routers detect this. Triggered updates allow routers to inform their neighbors of the route change immediately, without waiting for a regular update period. Triggered updates cause a wave of routing updates that travel through the network.

Triggered updates do not instantly arrive at every network device. It is therefore possible that a device that has yet to be informed of a network failure may send a regular update message (indicating that a route that has just gone down is still good) to a device that has just been notified of the network failure. In this case, the latter device now contains (and potentially advertises) incorrect routing information. Holddowns prevent this.

RIP Split Horizon The split-horizon rule derives from the fact that it is usually not useful to send information about a route back in the direction from which it came. For example, consider Figure 2-58.

Figure 2-58 *Split horizon prevents Router 2 from advertising Network A back to Router 1.*



Router 1 initially advertises that it has a route to Network A. There is no reason for Router 2 to include this route in its update back to Router 1, because Router 1 is closer to Network A. The split-horizon rule says that Router 2 should strike this route from any updates it sends to Router 1.

The split-horizon rule helps prevent two-node routing loops. For example, consider the case discussed earlier, where Router 1's interface to Network A goes down. Without split horizon, Router 2 informs Router 1 that it can get to Network A. Router 1 picks up Router 2's route as an alternative to its failed direct connection, causing a routing loop. Although holddowns should prevent this, split horizon provides extra algorithm stability.

RIP Poison Reverse Updates Poison reverse allows a router to include in its routing updates networks with a hop count set to infinity. This can be used for unreachable networks or another way of implementing split horizon. When used as an alternative to a split-horizon implementation, a router can include networks reachable via the link onto which the update is being sent but specify the hop count as infinity. This immediately kills two-node routing loops.

IGRP and Enhanced IGRP

Interior Gateway Routing Protocol (IGRP) is a routing protocol developed in the mid-1980s by Cisco Systems, Inc. Cisco's principal goal in creating IGRP was to provide a robust protocol for routing within an AS that has an arbitrarily complex topology and media with diverse bandwidth and delay characteristics. An AS is a collection of networks under common administration that share a common routing strategy.

In the mid-1980s, the most popular intra-AS routing protocol was RIP. Although RIP was quite useful for routing within small to moderate-sized, relatively homogeneous internetworks, its limits were being pushed by network growth. In particular, RIP's small hop-count limit (15) restricted the size of internetworks, and its single metric (hop count) did not allow for much routing flexibility in complex environments. The popularity of Cisco routers and the robustness of IGRP have encouraged many organizations with large internetworks to replace RIP with IGRP.

Cisco's initial IGRP implementation worked in IP networks. IGRP was designed to run in any network environment, however, and Cisco soon ported it to run in OSI CLNP networks.

Cisco developed Enhanced IGRP in the early 1990s to improve the operating efficiency of IGRP. Enhanced IGRP is discussed in detail later in this chapter.

IGRP is a distance-vector IGP. Distance-vector routing protocols call for each router to send all or a portion of its routing table in a routing update message at regular intervals to each of its neighboring routers. As routing information proliferates through the network, routers can calculate distances to all nodes within the internetwork.

Distance-vector routing protocols are often contrasted with link-state routing protocols, which send local connection information to all nodes in the internetwork.

IGRP uses a combination of metrics. Internetwork delay, bandwidth, reliability, and load are all factored in to the routing decision. Network administrators can set the weighting factors for each of these metrics. IGRP uses either the administrator-set or the default weightings to automatically calculate optimal routes.

IGRP provides a wide range for its metrics. For example, reliability and load can take on any value between 1 and 255, bandwidth can take on values reflecting speeds from 1,200 bps to 10 Gbps, and delay is measured in tens of microseconds and can take on any value from 1 to 2^{24}. Wide metric ranges allow satisfactory metric setting in internetworks with widely varying performance characteristics. Most importantly, the metric components are combined in a user-definable algorithm. As a result, network administrators can influence route selection in an intuitive fashion.

To provide additional flexibility, IGRP permits multipath routing. Dual equal-bandwidth lines may run a single stream of traffic in round-robin fashion, with automatic switchover to the second line if one line goes down. Also, multiple paths can be used even if the metrics for the paths are different. For example, if one path is three times better than another because its metric is three times lower, the better path will be used three times as often. Only routes with metrics that are within a certain range of the best route are used as multiple paths.

IGRP provides a number of features that are designed to enhance its stability. These include holddowns, split horizon, triggered updates, and poison reverse updates. These work in a similar fashion to the stability options discussed earlier in this chapter, in the section "RIP."

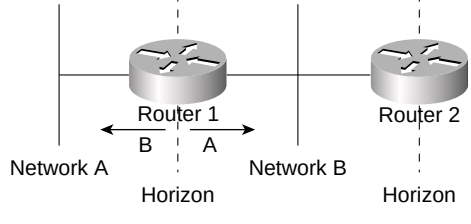
IGRP Holddowns and Triggered Updates Holddowns are used to prevent regular update messages from inappropriately reinstating a route that may have gone bad. When a route goes down, neighboring routers detect this via the lack of regularly scheduled update messages. These routers then calculate new routes and send routing update messages to inform their neighbors of the route change. This activity begins a wave of triggered updates that filter through the network.

These triggered updates do not instantly arrive at every network device. It is therefore possible that a device that has yet to be informed of a network failure might send a regular update message (indicating that a route that has just gone down is still good) to a device that has just been notified of the network failure. In this case, the latter device would now contain (and potentially advertise) incorrect routing information.

Holddowns tell routers to hold down any changes that might affect routes for some period of time. The hold-down period is usually calculated to be just greater than the period of time necessary to update the entire network with a routing change.

IGRP Split Horizon The split-horizon rule derives from the fact that it is usually not useful to send information about a route back in the direction from which it came. For example, consider Figure 2-59.

Figure 2-59 *Split horizon supplies the same benefit for IGRP as it does for RIP.*



Router 1 initially advertises that it has a route to Network A. There is no reason for Router 2 to include this route in its update back to Router 1, because Router 1 is closer to Network A. The split-horizon rule says that Router 2 should strike this route from any updates it sends to Router 1.

The split-horizon rule helps prevent routing loops. For example, consider the case where the link from Router 1 to Network A goes down. Without split horizon, Router 2 informs Router 1 that it can get to Network A. If Router 1 does not have sufficient intelligence, it may actually pick up Router 2's route as an alternative to its failed direct connection, causing a routing loop. Although holddowns should prevent this, split horizon is implemented in IGRP because it provides extra algorithm stability.

IGRP Poison Reverse Updates Whereas split horizon should prevent routing loops between adjacent routers, poison reverse updates are intended to defeat larger routing loops. Increases in routing metrics generally indicate routing loops. Poison reverse updates are sent to remove the route and place it in holddown. In Cisco's implementation of IGRP, poison reverse updates are sent if a route metric has increased by a factor of 1.1 or greater.

IGRP Timers IGRP maintains a number of timers and variables containing time intervals. These include an update timer, an invalid timer, a hold-time period, and a flush timer. The update timer specifies how frequently routing update messages should be sent. The IGRP default for this variable is 90 seconds. The invalid timer specifies how long a router should wait, in the absence of routing update messages about a specific route, before declaring that route invalid. The IGRP default for this variable is three times the update period. The hold-time variable specifies the hold-down period. The IGRP default for this variable is three times the update timer period plus 10 seconds. Finally, the flush timer indicates how much time should pass before a route should be flushed from the routing table. The IGRP default is seven times the routing update period.

Enhanced IGRP Enhanced IGRP combines the advantages of link-state protocols with the advantages of distance-vector protocols. Enhanced IGRP incorporates the Diffusing Update Algorithm (DUAL) developed at SRI International by Dr. J.J. Garcia-Luna-Aceves. Enhanced IGRP includes the following features:

- Fast convergence—Enhanced IGRP uses DUAL to achieve convergence quickly. A router running Enhanced IGRP stores all of its neighbors' routing tables so that it can quickly adapt to alternate routes. If no appropriate route exists, Enhanced IGRP queries its neighbors to discover an alternate route. These queries propagate until an alternate route is found.
- Variable-length subnet masks—Enhanced IGRP includes full support for variable length subnet masks. Subnet routes are automatically summarized on a network number boundary. In addition, Enhanced IGRP can be configured to summarize on any bit boundary at any interface.

- Partial, bounded updates—Enhanced IGRP does not make periodic updates. Instead, it sends partial updates only when the metric for a route changes. Propagation of partial updates is automatically bounded so that only those routers that need the information are updated. As a result of these two capabilities, Enhanced IGRP consumes significantly less bandwidth than IGRP.
- Multiple network-layer support—Enhanced IGRP includes support for AppleTalk, IP, and Novell NetWare. The AppleTalk implementation uses RTMP to redistribute routes. The IP implementation can use OSPF, RIP, Intermediate System-to-Intermediate System (IS-IS), EGP, or BGP to redistribute routes. The Novell implementation can use Novell RIP and Service Advertising Protocol (SAP) to redistribute routes.

Enhanced IGRP provides compatibility and seamless interoperation with IGRP routers. An automatic redistribution mechanism allows IGRP routes to be imported into Enhanced IGRP and Enhanced IGRP routes to be imported into IGRP, so it is possible to add Enhanced IGRP gradually into an existing IGRP network. Because the metrics for both protocols are directly translatable, they are as easily comparable as if they were routes that originated in their own ASs. In addition, Enhanced IGRP treats IGRP routes as external routes and provides a way for the network administrator to customize them.

Enhanced IGRP consists of the following components:

- Neighbor discovery/recovery
- Reliable Transport Protocol (RTP)
- DUAL finite state machine
- Protocol-dependent modules

Neighbor discovery/recovery is the process that routers use to dynamically learn about other routers on their directly attached networks. Routers must also discover when their neighbors become unreachable or inoperative. This process is achieved with low overhead by periodically sending small hello packets. As long as a router receives hello packets from a neighboring router, it assumes that the neighbor is functioning and that they can exchange routing information.

RTP is responsible for guaranteed, ordered delivery of Enhanced IGRP packets to all neighbors. It supports intermixed transmission of multicast or unicast packets. For efficiency, only certain Enhanced IGRP packets are transmitted reliably. For example, on a multiaccess network that has multicast capabilities, such as Ethernet, it is not necessary to send hello packets reliably to all neighbors individually. For that reason, Enhanced IGRP sends a single multicast hello packet containing an indicator that informs the receivers that the packet need not be acknowledged. Other types of packets, such as updates, indicate in the packet that acknowledgment is required. RTP has a provision for sending multicast packets quickly when unacknowledged packets are pending, which helps ensure that convergence time remains low in the presence of varying speed links.

The DUAL finite state machine embodies the decision process for all route computations. It tracks all routes advertised by all neighbors. DUAL uses distance information to select efficient, loop-free paths and selects routes for insertion in a routing table based on feasible successors. A feasible successor is a neighboring router used for packet forwarding that is a least-cost path to a destination guaranteed not to be part of a routing loop. When a neighbor changes a metric or when a topology change occurs, DUAL tests for feasible successors. If one is found, DUAL uses it to avoid recomputing the route unnecessarily. When there are no feasible successors but there are neighbors advertising the destination, a recomputation (also known as a diffusing computation) must occur to determine a new successor. Although recomputation is not processor intensive, it affects convergence time, so it is advantageous to avoid unnecessary recomputations.

The protocol-dependent modules are responsible for network-layer, protocol-specific requirements. For example, the IP Enhanced IGRP module is responsible for sending and receiving Enhanced IGRP packets that are encapsulated in IP. IP Enhanced IGRP is also responsible for parsing Enhanced IGRP packets and informing DUAL of the new information that has been received. IP Enhanced IGRP asks DUAL to make routing decisions, the results of which are stored in the IP routing table. IP Enhanced IGRP is responsible for redistributing routes learned by other IP routing protocols.

Enhanced IGRP Packet Types Enhanced IGRP uses five packet types:

- Hello/acknowledgment
- Update
- Query
- Reply
- Request

Hello packets are multicast for neighbor discovery/recovery and do not require acknowledgment. An acknowledgment packet is a hello packet that has no data. Acknowledgment packets contain a nonzero acknowledgment number, and they are always sent using a unicast address.

Update packets are used to convey reachability of destinations. When a new neighbor is discovered, unicast update packets are sent so the neighbor can build up its topology table. In other cases, such as a link cost change, updates are multicast. Updates are always transmitted reliably.

Query and reply packets are sent when a destination has no feasible successors. Query packets are always multicast. Reply packets are sent in response to query packets to indicate to the originator that the originator does not need to recompute the route because there are feasible successors. Reply packets are unicast to the originator of the query. Both query and reply packets are transmitted reliably.

Request packets are used to get specific information from one or more neighbors. Request packets are used in route server applications and can be multicast or unicast. Request packets are transmitted unreliably.

Enhanced IGRP Neighbor Tables When a router discovers a new neighbor, it records the neighbor's address and interface as an entry in the neighbor table. There is one neighbor table for each protocol-dependent module. When a neighbor sends a hello packet, it advertises a hold time, which is the amount of time a router treats a neighbor as reachable and operational. If a hello packet is not received within the hold time, the hold time expires and DUAL is informed of the topology change.

The neighbor table entry also includes information required by RTP. Sequence numbers are employed to match acknowledgments with data packets. The last sequence number received from the neighbor is recorded so that out-of-order packets can be detected. A transmission list is used to queue packets for possible retransmission on a per-neighbor basis. Round-trip timers are kept in the neighbor table entry to estimate an optimal retransmission interval.

Enhanced IGRP Topology Tables The topology table contains all destinations advertised by neighboring routers. The protocol-dependent modules populate the table, and the table is acted on by the DUAL finite state machine. Each entry in the topology table includes the destination address and a list of neighbors that have advertised the destination. For each neighbor, the entry records the advertised metric, which the neighbor stores in its routing table. An important rule that distance-vector protocols must follow is that if the neighbor is advertising this destination, it must be using the route to forward packets.

The metric that the router uses to reach the destination is also associated with the destination. The metric that the router uses in the routing table and to advertise to other routers is the sum of the best advertised metric from all neighbors plus the link cost to the best neighbor.

Enhanced IGRP Route States A topology table entry for a destination can be in one of two states: active or passive. A destination is in the passive state when the router is not performing a recomputation and in the active state when the router is performing a recomputation. If feasible successors are always available, a destination never has to go into the active state, thereby avoiding a recomputation.

A recomputation occurs when a destination has no feasible successors. The router initiates the recomputation by sending a query packet to each of its neighboring routers. The neighboring router can send a reply packet, indicating that it has a feasible successor for the destination, or it can send a query packet, indicating that it is participating in the recomputation. While a destination is in the active state, a router cannot change the destination's routing table information. Once the router has received a reply from each neighboring router, the topology table entry for the destination returns to the passive state and the router can select a successor.

Enhanced IGRP Route Tagging Enhanced IGRP supports internal and external routes. Internal routes originate within an Enhanced IGRP AS. Therefore, a directly attached network that is configured to run Enhanced IGRP is considered an internal route and is propagated with this information throughout the Enhanced IGRP AS. External routes are learned by another routing protocol or reside in the routing table as static routes. These routes are tagged individually with the identity of their origin.

External routes are tagged with the following information:

- The router ID of the Enhanced IGRP router that redistributed the route
- The AS number of the destination
- A configurable administrator tag
- The ID of the external protocol
- The metric from the external protocol
- Bit flags for default routing

Route tagging allows the network administrator to customize routing and maintain flexible policy controls. Route tagging is particularly useful in transit ASs where Enhanced IGRP typically interacts with an interdomain routing protocol that implements global policies, resulting in scalable, policy-based routing.

OSPF

OSPF (Open Shortest Path First) is a routing protocol developed for IP networks by the IGP working group of the Internet Engineering Task Force (IETF). The working group was formed in 1988 to design an IGP based on the shortest path first (SPF) algorithm for use in the Internet. Like IGRP, OSPF was created because RIP was, in the mid-1980s, increasingly unable to serve large, heterogeneous internetworks.

OSPF was derived from several research efforts, including the following:

- BBN's SPF algorithm developed in 1978 for the ARPANET (a landmark packet-switching network developed in the early 1970s by BBN)
- Dr. Radia Perlman's research on fault-tolerant broadcasting of routing information (1988)
- BBN's work on area routing (1986)
- An early version of the OSI IS-IS routing protocol

As indicated by its abbreviation, OSPF has two primary characteristics. First, it is open, in that its specification is in the public domain and was originally described in RFC 1131. The most recent version, known as OSPF 2, is described in RFC 1583. Second, it is based on the SPF algorithm, which is sometimes referred to as the Dijkstra algorithm, named for the person credited with its creation.

OSPF is a link-state routing protocol. As such, it calls for the sending of link-state advertisements (LSAs) to all other routers within the same hierarchical area. Information on attached interfaces, metrics used, and other variables is included in OSPF LSAs. As OSPF routers accumulate link-state information, they use the SPF algorithm to calculate the shortest path to each node.

As a link-state algorithm, OSPF contrasts with RIP and IGRP, which are distance-vector routing protocols. Routers running the distance-vector algorithm send all or a portion of their routing tables in routing update messages, but only to their neighbors.

Unlike RIP, OSPF can operate within a hierarchy. The largest entity within the hierarchy is the AS. An AS is a collection of networks under a common administration, sharing a common routing strategy. OSPF is an intra-AS (interior gateway) routing protocol, although it is capable of receiving routes from and sending routes to other ASs.

An AS can be divided into a number of areas, which are groups of contiguous networks and attached hosts. Routers with multiple interfaces can participate in multiple areas. These routers, which are called *area border routers*, *maintain separate topological databases for each area.*

A topological database is essentially an overall picture of networks in relationship to routers. The topological database contains the collection of LSAs received from all routers in the same area. Because routers within the same area share the same information, they have identical topological databases.

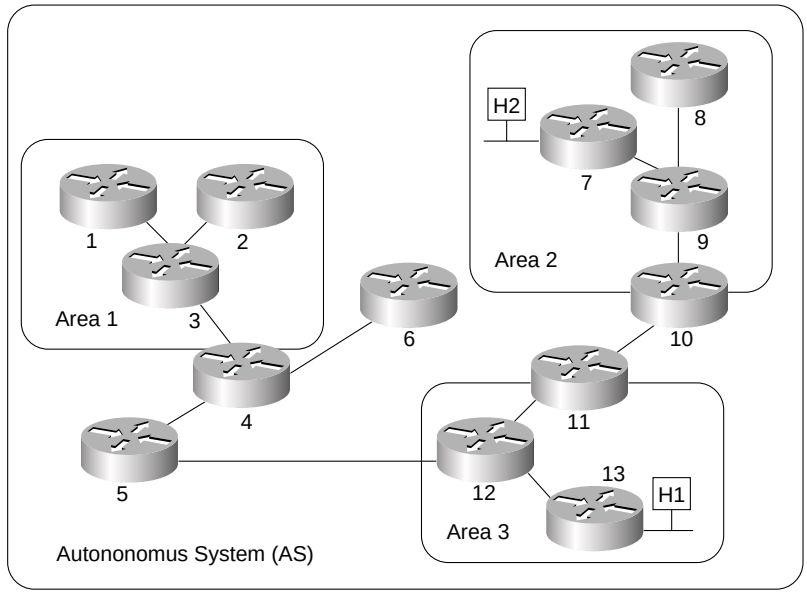
The term *domain* is sometimes used to describe a portion of the network in which all routers have identical topological databases.

An area's topology is invisible to entities outside the area. By keeping area topologies separate, OSPF passes less routing traffic than it would if the AS were not partitioned.

Area partitioning creates two different types of OSPF routing, depending on whether the source and destination are in the same or different areas. *Intra-area routing* occurs when the source and destination are in the same area; *interarea routing* occurs when they are in different areas.

An OSPF backbone is responsible for distributing routing information between areas. It consists of all area border routers, networks not wholly contained in any area, and their attached routers. Figure 2-60 shows an example of an internetwork with several areas.

Figure 2-60 *OSPF uses hierarchical internetwork design to optimize routing.*



In Figure 2-60, Routers 4, 5, 6, 10, 11, and 12 make up the backbone. If Host 1 in Area 3 wants to send a packet to Host 2 in Area 2, the packet is sent to Router 13, which forwards the packet to Router 12, which sends the packet to Router 11. Router 11 forwards the packet along the backbone to area border router Router 10, which sends the packet through two intra-area routers (Router 9 and Router 7) to be forwarded to Host 2.

The backbone itself is an OSPF area (Area 0), so all backbone routers use the same procedures and algorithms to maintain routing information within the backbone that any area router would. The backbone topology is invisible to all intra-area routers, as are individual area topologies to the backbone.

AS border routers running OSPF learn about exterior routes through exterior gateway protocols such as EGP or BGP, or through configuration information.

The SPF Algorithm The SPF routing algorithm is the basis for OSPF operations. When an SPF router is powered up, it initializes its routing protocol data structures and then waits for indications from lower-layer protocols that its interfaces are functional.

When a router is assured that its interfaces are functioning, it uses the OSPF Hello protocol to acquire neighbors, which are routers with interfaces to a common network. The router sends Hello packets to its neighbors and receives their Hello packets. In addition to helping acquire neighbors, Hello packets also act as keepalives to let routers know that other routers are still functional.

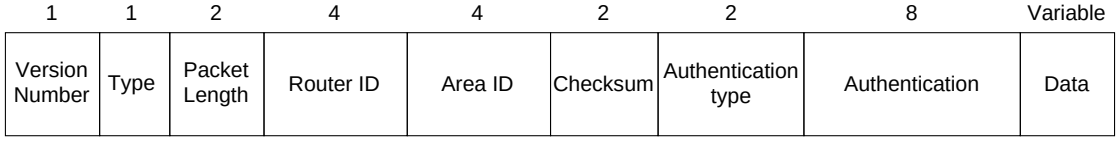
On multiaccess networks (that is, networks supporting more than two routers), the Hello protocol elects a designated router and a backup designated router. The designated router is responsible, among other things, for generating LSAs for the entire multiaccess network. Designated routers allow a reduction in network traffic and in the size of the topological database.

When the link-state databases of two neighboring routers are synchronized, the routers are said to be adjacent. On multiaccess networks, the designated router determines which routers should become adjacent. Topological databases are synchronized between pairs of adjacent routers. Adjacencies control the distribution of routing protocol packets. These packets are sent and received only on adjacencies.

Each router periodically sends an LSA. LSAs are also sent when a router's state changes. LSAs include information on a router's adjacencies. By comparing established adjacencies to link states, failed routers can be quickly detected and the network's topology altered appropriately. From the topological database generated from LSAs, each router calculates a shortest-path tree, with itself as root. The shortest-path tree, in turn, yields a routing table.

OSPF Packet Format Figure 2-61 shows the OSPF packet header. The first field in the OSPF header is the OSPF version number. The version number identifies the particular OSPF implementation being used.

Figure 2-61 *Each OSPF packet begins with a 24-byte header.*



Following the version number is the type field. There are five OSPF packet types:

- Hello—Establishes and maintains neighbor relationships. These messages are sent at regular intervals (typically every 10 seconds on a LAN).
- Database description—Describes the contents of the topological database. These messages are exchanged when an adjacency is being initialized.
- Link-state request—Requests pieces of the topological database from neighbor routers. These messages are exchanged after a router discovers (through examination of database description packets) that parts of its topological database are out of date.
- Link-state update—Responses to link-state request packets. These messages are also used for the regular dispersal of LSAs. Several LSAs can be included within a single link-state update packet.
- Link-state acknowledgment—Acknowledges link-state update packets. Link-state update packets must be explicitly acknowledged to ensure that link-state flooding throughout an area is a reliable process.

Each LSA in a link-state update packet contains a type field. There are four LSA types:

- Router-links advertisements (RLAs)—Describe the collected states of the router's links to a specific area. A router sends an RLA for each area to which it belongs. RLAs are flooded throughout the entire area, and no further.
- Network-links advertisements (NLAs)—Sent by the designated routers. They describe all the routers that are attached to a multiaccess network, and are flooded throughout the area containing the multiaccess network.
- Summary-links advertisements (SLAs)—Summarize routes to destinations outside an area, but within the AS. They are generated by area border routers, and are flooded throughout the area. Only intra-area routes are advertised into the backbone. Both intra-area and interarea routes are advertised into the other areas.
- AS external-links advertisements—Describe a route to a destination that is external to the AS. AS external-links advertisements are originated by AS boundary routers. This type of advertisement is the only type that is forwarded everywhere in the AS; all others are forwarded only within specific areas.

Following the OSPF packet header's type field is a packet length field, which provides the packet's length, including the OSPF header, in bytes.

The router ID field identifies the packet's source.

The area ID field identifies the area to which the packet belongs. All OSPF packets are associated with a single area.

A standard IP checksum field checks the entire packet contents for potential damage suffered in transit.

The authentication type field contains an authentication type. "Simple password" is an example of an authentication type. All OSPF protocol exchanges are authenticated. The authentication type is configurable on a per-area basis.

The authentication field is 64 bits in length and contains authentication information.

Additional OSPF Features Additional OSPF features include equal-cost, multipath routing and routing based on upper-layer TOS requests. TOS-based routing supports upper-layer protocols that can specify particular types of service. For example, an application might specify that certain data is urgent. If OSPF has high-priority links at its disposal, these can be used to transport the urgent datagram.

OSPF supports one or more metrics. If only one metric is used, it is considered to be arbitrary, and TOS is not supported. If more than one metric is used, TOS is optionally supported through the use of a separate metric (and, therefore, a separate routing table) for each of the eight combinations created by the three IP TOS bits (the delay, throughput, and reliability bits). For example, if the IP TOS bits specify low delay, low throughput, and high reliability, OSPF calculates routes to all destinations based on this TOS designation.

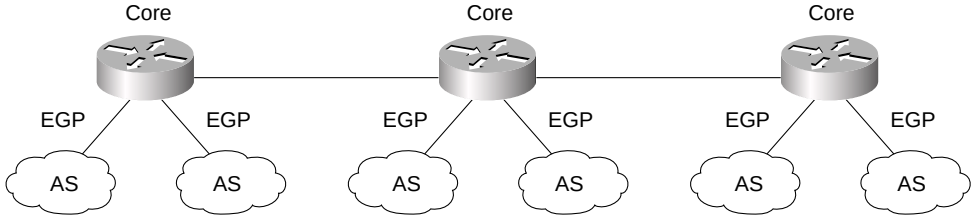
IP subnet masks are included with each advertised destination, enabling variable-length subnet masks. With variable-length subnet masks, an IP network can be broken into many subnets of various sizes. This provides network administrators with extra network configuration flexibility.

EGP

EGP is an interdomain reachability protocol. EGP is documented in RFC 904, published in April 1984. As the first exterior gateway protocol to gain widespread acceptance in the Internet, EGP served a valuable purpose. EGP's weaknesses became more apparent as the Internet grew and matured. Because of these weaknesses, EGP is being phased out of the Internet, replaced mainly by BGP.

EGP was originally designed to communicate reachability to and from the ARPANET core routers. Information was passed from individual source nodes in distinct Internet ASs up to the core routers, which passed the information through the backbone until it could be passed down to the destination network within another AS. This relationship between EGP and other ARPANET components is shown in Figure 2-62.

Figure 2-62 *EGP was used for routing between autonomous systems in the ARPANET.*



Although EGP is a dynamic routing protocol, it uses a very simple design. It does not use metrics and therefore cannot make intelligent routing decisions. EGP routing updates contain network reachability information. In other words, they specify that certain networks are reachable through certain routers.

EGP has three primary functions. First, routers running EGP establish a set of neighbors. These neighbors are simply routers with which an EGP router wishes to share reachability information; there is no implication of geographic proximity. Second, EGP routers poll their neighbors to see whether they are alive. Third, EGP routers send update messages containing information about the reachability of networks within their ASs.

EGP Packet Format The EGP packet format is shown in Figure 2-63. The first field in the EGP packet header is the EGP version number field. This field identifies the current EGP version and is checked by recipients to determine whether there is a match between the sender and recipient version numbers.

Figure 2-63 *The EGP packet header consists of eight fields.*

1	1	2	1	2	2	2	Variable
EGP Version Number	Type	Code	Status	Checksum	Autonomous System Number	Sequence Number	Data

The code field distinguishes among message subtypes.

The status field contains message-dependent status information. Status codes include insufficient resources, parameter problem, and protocol violation.

The checksum field is used to detect possible problems that may have developed with the packet in transit.

The autonomous system number field identifies the AS to which the sending router belongs.

The sequence number field is the last field in the EGP packet header. This field allows two EGP routers exchanging messages to match requests with replies. The sequence number is initialized to zero when a neighbor is established and incremented by one with each request-response transaction.

EGP Message Types Additional fields follow the EGP header. The contents of these fields vary depending on the message type (as specified by the type field):

- Neighbor acquisition—The neighbor acquisition message includes a hello interval field and a poll interval field. The hello interval field specifies the interval period for testing whether neighbors are alive. The poll interval field specifies the routing update frequency.
- Neighbor reachability—The neighbor reachability message adds no extra fields to the EGP header. These messages use the code field to indicate whether the message is a hello message or a response to a hello message. Separating the reachability assessment function from the routing update function reduces network traffic because network reachability changes usually occur more often than routing parameter changes. Only after a specified percentage of reachability messages have not been received does an EGP node declare a neighbor to be down.
- Poll—To provide correct routing between ASs, EGP must know the relative location of remote hosts. The poll message allows EGP routers to acquire reachability information about the networks on which these hosts reside. These messages only have one field beyond the common header—the IP source network field. This field specifies the network to be used as a reference point for the request.
- Routing update—Routing update messages provide a way for EGP routers to indicate the locations of various networks within their ASs. In addition to the common header, these messages include many additional fields. The number of interior gateways field indicates the number of interior gateways appearing in the message. The number of exterior gateways field indicates the number of exterior gateways appearing in the message. The IP source network field provides the IP address of the network from which reachability is measured. Following this field is a series of gateway blocks. Each gateway block provides the IP address of a gateway and a list of networks and distances associated with reaching those networks.

  Within the gateway block, EGP lists networks by distances. In other words, at distance three, there may be four networks. These networks are then listed by address. The next group of networks may be those that are distance four away, and so on.

  EGP does not interpret the distance metrics that are contained within the routing update messages. In essence, EGP uses the distance field to indicate whether a path exists; the distance value can only be used to compare paths if those paths exist wholly within a particular AS. For this reason, EGP is more a reachability protocol than a routing protocol. This restriction also places topology limitations on the structure of the Internet. Specifically, an EGP portion of the Internet must be a tree structure in which a core gateway is the root, and there are no loops among other ASs within the tree. This restriction is a primary limitation of EGP, and provides an impetus for its gradual replacement by other, more capable exterior gateway protocols.

- Error—Error messages identify various EGP error conditions. In addition to the common EGP header, EGP error messages provide a reason field, followed by an error message header. Typical EGP errors (reasons) include bad EGP header format, bad EGP data field format, excessive polling rate, and the unavailability of reachability information. The error message header consists of the first three 32-bit words of the EGP header.

BGP

BGP performs interdomain routing. BGP is an exterior gateway protocol, which means that it performs routing between multiple ASs and exchanges routing and reachability information with other BGP systems.

BGP was developed to replace its predecessor, EGP, as the standard exterior gateway routing protocol used in the global Internet. BGP solves serious problems with EGP and scales to Internet growth more efficiently.

BGP is specified in several RFCs:

- RFC 1771—This RFC describes BGP4, the current version of BGP.
- RFC 1654—This RFC describes the first BGP4 specification.
- RFC 1105, RFC 1163, and RFC 1267—These RFCs describe versions of BGP prior to BGP4.

Although BGP was designed as an inter-AS protocol, it can be used both within and between ASs. Two BGP neighbors communicating between ASs must reside on the same physical network. BGP routers within the same AS communicate with one another to ensure that they have a consistent view of the AS and to determine which BGP router within that AS will serve as the connection point to or from certain external ASs.

Some ASs are merely pass-through channels for network traffic. In other words, some ASs carry network traffic that did not originate within and is not destined for them. BGP must interact with whatever intra-AS routing protocols exist within these pass-through ASs.

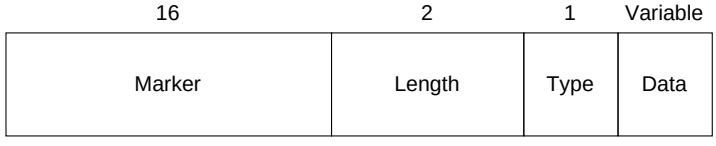
BGP update messages consist of network number/AS path pairs. The AS path contains the string of ASs through which the specified network may be reached. These update messages are sent over the TCP transport mechanism to ensure reliable delivery.

The initial data exchange between two routers is the entire BGP routing table. Incremental updates are sent out as the routing tables change. Unlike some other routing protocols, BGP does not require periodic refresh of the entire routing table. Instead, routers running BGP retain the latest version of each peer routing table. Although BGP maintains a routing table with all feasible paths to a particular network, it only advertises the primary (optimal) path in its update messages.

The BGP metric is an arbitrary unit number specifying the degree of preference of a particular path. These metrics are typically assigned by the network administrator through configuration files. Degree of preference may be based on any number of criteria, including AS count (paths with a smaller AS count are generally better) and stability and speed of the link.

BGP Packet Format The BGP packet format is shown in Figure 2-64.

Figure 2-64 *The BGP packet header consists of four fields.*



BGP packets have a common 19-byte header consisting of three fields:

- The marker field is 16 bytes long and contains a value that the receiver of the message can predict. This field is used for authentication.
- The length field contains the total length of the message, in bytes.
- The type field specifies the message type.

BGP Message Types Four message types are specified in BGP:

- Open—After a transport protocol connection is established, the first message sent by each side is an open message. If the open message is acceptable to the recipient, a keepalive message confirming the open message is sent back. Upon successful confirmation of the open message, updates, keepalives, and notifications may be exchanged.

  In addition to the common BGP packet header, open messages define several fields. The version field provides a BGP version number and allows the recipient to check that it is running the same version as the sender. The autonomous system field provides the AS number of the sender. The hold-time field indicates the maximum number of seconds that may elapse without receipt of a message before the transmitter is assumed to be dead. The authentication code field indicates the authentication type being used (if any). The authentication data field contains actual authentication data (if any).

- Update—BGP update messages provide routing updates to other BGP systems. Information in these messages is used to construct a graph describing the relationships of the various ASs. In addition to the common BGP header, update messages have several additional fields. These fields provide routing information by listing path attributes corresponding to each network. BGP currently defines five attributes:

  - Origin—Can take on one of three values: IGP, EGP, and incomplete. The IGP attribute means that the network is part of the AS. The EGP attribute means that the information was originally learned from the EGP. BGP implementations would be inclined to prefer IGP routes over EGP routes, since EGP fails in the presence of routing loops. The incomplete attribute is used to indicate that the network is known via some other means.
  - AS path—Provides the actual list of ASs on the path to the destination.
  - Next hop—Provides the IP address of the router that should be used as the next hop to the networks listed in the update message.
  - Unreachable—If present, indicates that a route is no longer reachable.
  - Inter-AS metric—Provides a way for a BGP router to advertise its cost to destinations within its own AS. This information can be used by routers external to the advertiser's AS to select an optimal route into the AS to a particular destination.
- Notification—Notification messages are sent when an error condition has been detected and one router wants to tell another why it is closing the connection between them. Aside from the common BGP header, notification messages have an error code field, an error subcode field, and error data. The error code field indicates the type of error, and can be one of the following:
  - Message header error—Indicates a problem with the message header such as an unacceptable message length, an unacceptable marker field value, or an unacceptable message type.
  - Open message error—Indicates a problem with an open message such as an unsupported version number, an unacceptable AS number or IP address, or an unsupported authentication code.
  - Update message error—Indicates a problem with the update message. Examples include a malformed attribute list, an attribute list error, and an invalid next-hop attribute.
  - Hold time expired—Indicates a hold-time expiration, after which a BGP node will be declared dead.
- Keepalive—Keepalive messages do not contain any additional fields beyond those in the common BGP header. These messages are sent often enough to keep the hold-time timer from expiring.

NetWare Protocols

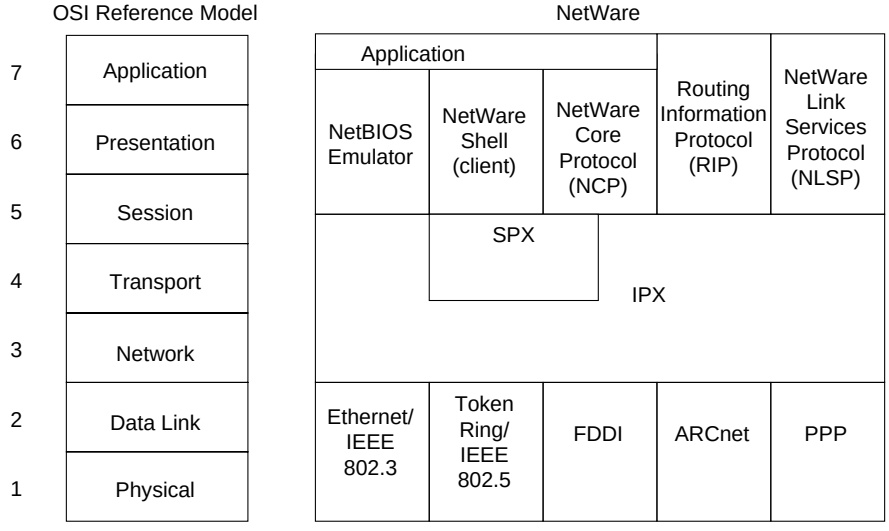
NetWare is a network operating system (NOS) that provides transparent remote file access and numerous other distributed network services, including printer sharing, electronic mail transfer, and database access. Much of NetWare's networking technology was derived from XNS, a networking system created by Xerox Corporation in the late 1970s.

Novell introduced NetWare to the market in the early 1980s. By the early 1990s, NetWare's NOS market share had risen to between 50% and 75%. With more than 500,000 NetWare networks installed worldwide and an accelerating movement to connect networks to other networks, NetWare and its supporting protocols often coexist on the same physical channel with many other popular protocols, including TCP/IP and AppleTalk.

As a NOS environment, NetWare specifies the upper five layers of the OSI reference model. Like other NOSs such as NFS from Sun Microsystems, Inc., and NT from Microsoft Corporation, NetWare is based on a client/server architecture. In such architectures, clients (sometimes called workstations) request certain services such as file and printer access from servers.

Figure 2-65 illustrates a simplified view of NetWare's best-known protocols and their relationship to the OSI reference model. With appropriate drivers, NetWare can run on any media-access protocol. The figure lists the media-access protocols currently supported with NetWare drivers.

Figure 2-65 *The NetWare protocol suite maps to all OSI reference model layers.*

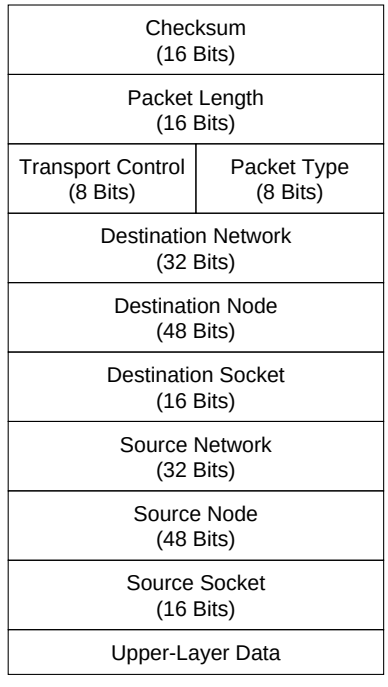


NetWare also works over synchronous WAN links using PPP.

The NetWare Network Layer

IPX is Novell's network-layer protocol. When a device to be communicated with is located on a different network, IPX routes the information to the destination through any intermediate networks. Figure 2-66 shows the IPX packet format.

Figure 2-66 *IPX uses 32-bit network and 48-bit node addresses.*



The IPX packet begins with a 16-bit checksum field that is set to ones by default. (FFFF in hexadecimal.) NetWare 4.x and 5.x enable you to turn on IPX checksumming.

A 16-bit packet length field specifies the length, in bytes, of the complete IPX datagram (including the IPX header and valid data, but excluding any data-link padding). IPX packets can be any length up to the media maximum transmission unit (MTU) size. There is no packet fragmentation in IPX.

The 8-bit transport control field indicates the number of routers the packet has passed through. This field is set to 0 by IPX before packet transmission and is incremented by each router that forwards the packet. When the value of this field reaches 15, the packet is discarded under the assumption that a routing loop might be occurring. This is always a silent discard—no discard notification process is available for IPX.

The 8-bit packet type field specifies the upper-layer protocol to receive the packet's information. Common values for this field are 0, which specifies unknown packet; 1, which specifies RIP; 5, which specifies SPX; and 17, which specifies NCP.

The destination network, destination node, and destination socket fields specify destination information. The source network, source node, and source socket fields specify source information.

The network number is a 32-bit number assigned by the network administrator, and the node number is a 48-bit number that identifies the LAN hardware address. The socket number is a 16-bit hexadecimal number that identifies the higher-layer process. Values are as follows:

Value	Description
0451	NetWare Core Protocol
0452	Service Advertising Protocol
0453	Routing Information Protocol
0455	NetBIOS
0456	Diagnostics
0457	Serialization
4000–8000	Dynamic sockets

The upper-layer data field contains information for upper-layer processes.

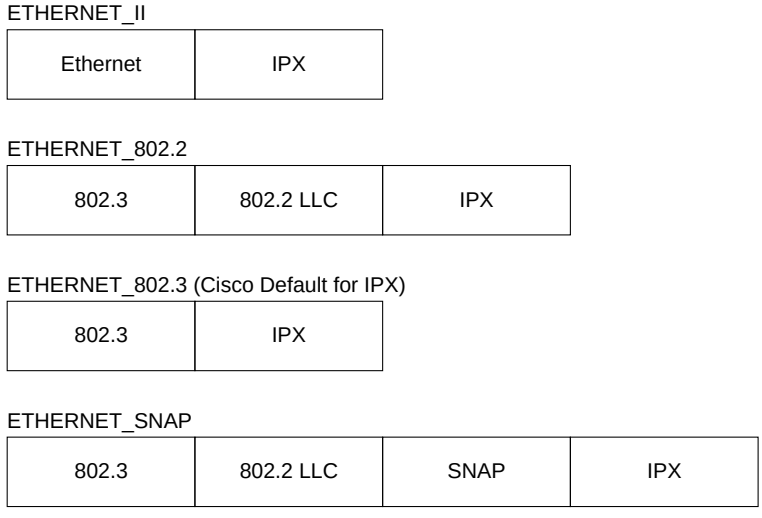
NetWare Encapsulation Types *Encapsulation* is the process of packaging upper-layer protocol information and data into a frame. Novell supports multiple encapsulation schemes on Ethernet/802.3 networks. A Cisco router supports multiple encapsulation schemes on a single router interface, provided that multiple network numbers are assigned.

NetWare supports the Ethernet/IEEE 802.3 encapsulation schemes listed in Table 2-5 and shown in Figure 2-67.

Table 2-5 *NetWare Ethernet encapsulation types.*

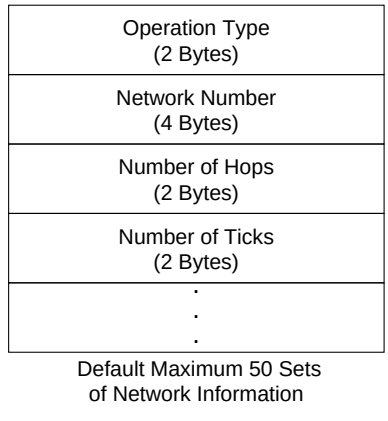
Common Term	Novell Term	Cisco Term	Characteristics
Ethernet V. 2	ETHERNET_II	arpa	Includes Ethertype
IEEE 802.3	ETHERNET_802.2	sap	Includes 802.3 length and 802.2 SAPs
Novell 802.3 raw	ETHERNET_802.3	novell-ether	Includes 802.3 length with no 802.2 SAPs
SNAP	ETHERNET_SNAP	snap	Includes 802.2 SAPs and SNAP header

Figure 2-67 *There are four IPX encapsulation types for Ethernet LANs.*



To route packets in an internetwork, IPX uses the dynamic routing protocol RIP. IPX RIP is similar but not identical to IP RIP. Figure 2-68 shows the IPX RIP packet format.

Figure 2-68 *Novell RIP uses ticks and hops as routing metrics.*



The operation field specifies the packet operation, with value 1 indicating RIP request and 2 indicating RIP response.

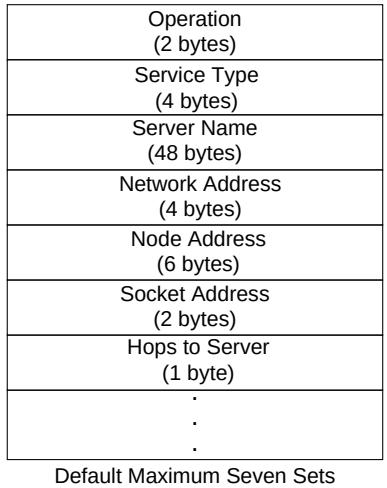
The network number is the 32-bit address of the specified network.

The hops field indicates the number of routers that must be passed through to reach the specified network.

The ticks field is a measure of time needed to reach the specified network (18.21 ticks/second).

Novell uses another protocol, Service Advertising Protocol (SAP), to broadcast and discover network services. SAP allows nodes that provide services (such as file servers and print servers) to advertise their addresses and the services they provide. Note that this is a completely different concept than an LLC service access point (SAP). Figure 2-69 shows the IPX SAP packet format.

Figure 2-69 *SAP packets are advertised every 60 seconds.*



The operation field specifies the operation that the packet will perform:

Value	Description
1	General service request
2	General service reply
3	Nearest service request
4	Nearest service reply

NOTE NetWare 5 also includes the ability to send SAP requests and include server names or addresses in the lookup (instead of just denoting the service type desired). This technology, called specific SAP technology, uses SAP types 12, 13, 14, and 15.

The service type field specifies the service performed. Values are given in hexadecimal and examples include the following:

Value	Description
0004	File Server
0007	Print Server
0047	Advertising Print Server
0107	NetWare Remote Console
0278	NDS Server

The server name is a 48-byte field containing the server's name. If the server name is shorter than 48 bytes, the field is zero-padded.

The network address and the node address contain the server's 32-bit network and 48-bit node numbers, respectively. The socket address field is the server's 16-bit socket number which identifies the source process (similar to a TCP/IP port number).

The hops field is the number of routers that must be passed through to reach the specified server.

NLSP NetWare Link Services Protocol (NLSP) is a link-state routing protocol from Novell designed to overcome some of the limitations associated with IPX RIP and its companion protocol, SAP. Compared to RIP and SAP, NLSP provides better reliability, improved routing, reduced network overhead, and increased scalability.

NLSP facilitates improved routing decisions by specifying that NLSP-based routers store a complete map of the network instead of only next-hop information, which RIP-based routers store. Routing information is transmitted only when the topology has changed. (RIP sends routing information every 60 seconds, regardless of whether the topology has changed.) Additionally, NLSP-based routers send service information updates only when services change. (SAP sends service information every 60 seconds, regardless of whether there have been any changes.)

To further reduce the effects of network traffic, NLSP supports multicast addressing so that routing information is sent only to other NLSP routers. (RIP sends routing information to all devices.)

NLSP supports load balancing across parallel paths. If there are two or more equal-cost paths between two network nodes, the traffic is automatically divided among them to make efficient use of the internetwork. NLSP periodically checks links for connectivity and for the data integrity of routing information. If a link fails, NLSP switches to an alternate link and updates the network topology databases stored in each router in the routing area.

Other useful features of NLSP are its support for up to 127 hops (RIP supports only 15 hops) and its support for hierarchical addressing of network nodes, allowing networks to contain thousands of LANs and servers. NLSP is based on the OSI IS-IS protocol and is designed to be used in a hierarchical routing environment, where routing areas can be linked into routing domains and domains can be linked into a global internetwork. Hierarchical routing simplifies the process of enlarging a network by reducing the information that every router must store and process to route packets within and between areas and domains.

NLSP was designed to replace RIP, which is Novell's original routing protocol, designed when internetworks were local and relatively small. NLSP is better suited to today's large, global internetworks. NLSP-based routers are backward compatible with RIP-based routers. Any combination of NLSP-based and RIP-based routers can be used in the same internetwork during the migration from RIP to NLSP.

The NetWare Transport Layer

SPX is a commonly used NetWare transport protocol. Novell derived this protocol from XNS's SPP. As with TCP and many other transport protocols, SPX is a reliable, connection-oriented protocol that supplements the datagram service provided by Layer 3 protocols.

NOTE With the release of NetWare 5, Novell began supporting NCP with TCP/IP (or UDP/IP) used for Layer 3 and Layer 4 protocol services as an alternative to IPX.

NetWare Upper-Layer Protocols

NetWare supports a wide variety of upper-layer protocols, but several are somewhat more popular than others. The NetWare client software runs in clients (often called *workstations* in the NetWare community) and intercepts application I/O calls to determine whether they require network access. If so, the NetWare client software packages the requests and sends them to lower-layer software for processing and network transmission. If not, they are simply passed to local I/O resources. Client applications are unaware of any network access required for completion of application calls.

NCP consists of a series of server routines designed to satisfy application requests coming from, for example, the NetWare client software. Services provided by NCP include file access, name management, accounting, security, and file synchronization.

NetWare also supports the Network Basic Input/Output System (NetBIOS) session-layer interface specification from IBM and Microsoft. NetWare's NetBIOS emulation software allows programs written to the industry-standard NetBIOS interface to run within the NetWare system.

NetWare application-layer services are most often provided through NetWare Loadable Modules (NLMs). NLMs are implemented as add-on modules that attach into the NetWare system. NLMs for alternate protocol stacks, communication services, database services, and many other services are currently available from Novell and third parties.

AppleTalk

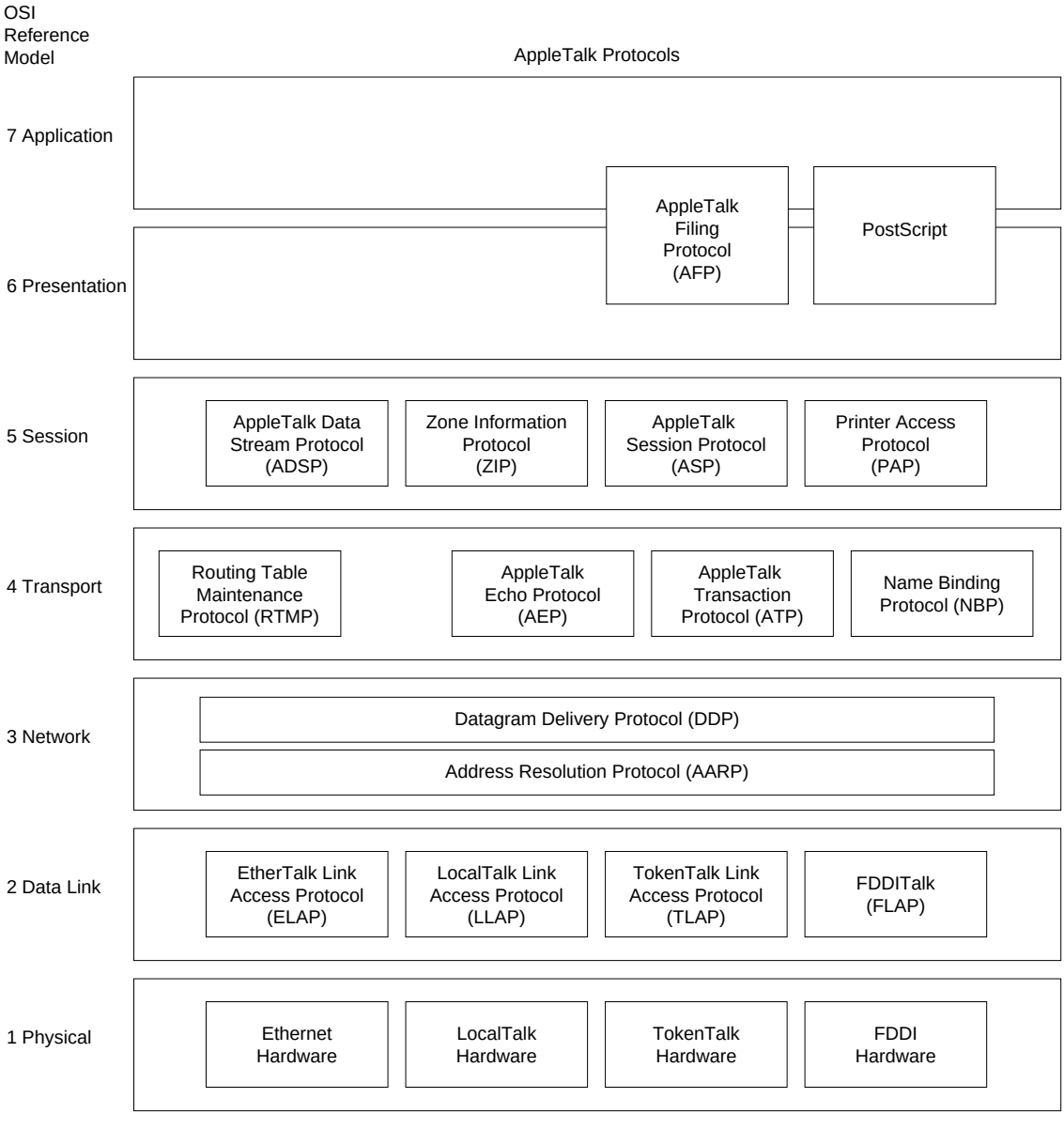
In the early 1980s, as Apple Computer, Inc., was preparing to introduce the Macintosh computer, Apple engineers knew that networks would become a critical need. They wanted to ensure that a Macintosh-based network was a seamless extension of the revolutionary Macintosh user interface. With these two goals in mind, Apple decided to build a network interface into every Macintosh and to integrate that interface into the desktop environment. Apple's new network architecture was called AppleTalk.

The original implementation of AppleTalk, which was designed for local workgroups, is now commonly referred to as AppleTalk Phase 1. With the installation of more than 1.5 million Macintosh computers in the first five years of the product's life, however, Apple found that some large corporations were exceeding the built-in limits of AppleTalk Phase 1, so they enhanced the protocols. The new protocols, known as AppleTalk Phase 2, enhanced AppleTalk's routing capabilities and allowed AppleTalk to run successfully in larger networks.

AppleTalk was designed as a client/server distributed network system. In other words, users share network resources (such as files and printers) with other users. Computers supplying these network resources are called *servers*; computers using a server's network resources are called *clients*. Interaction with servers is essentially transparent to the user because the computer itself determines the location of the requested material and accesses it without further information from the user. In addition to their ease of use, distributed systems also enjoy an economic advantage over peer-to-peer systems because important materials can be located in a few, rather than many, locations.

AppleTalk corresponds relatively well to the OSI reference model. In Figure 2-70, AppleTalk protocols are shown adjacent to the OSI layers to which they map.

Figure 2-70 *The AppleTalk protocol suite maps to every layer of the OSI reference model.*



Apple constructed AppleTalk to be link-layer independent. In other words, it can theoretically run on top of any link-layer implementation. Apple supports a variety of link-layer implementations, including Ethernet, Token Ring, FDDI, and LocalTalk. Apple refers to AppleTalk over Ethernet as EtherTalk, to AppleTalk over Token Ring as TokenTalk, and to AppleTalk over FDDI as FDDITalk.

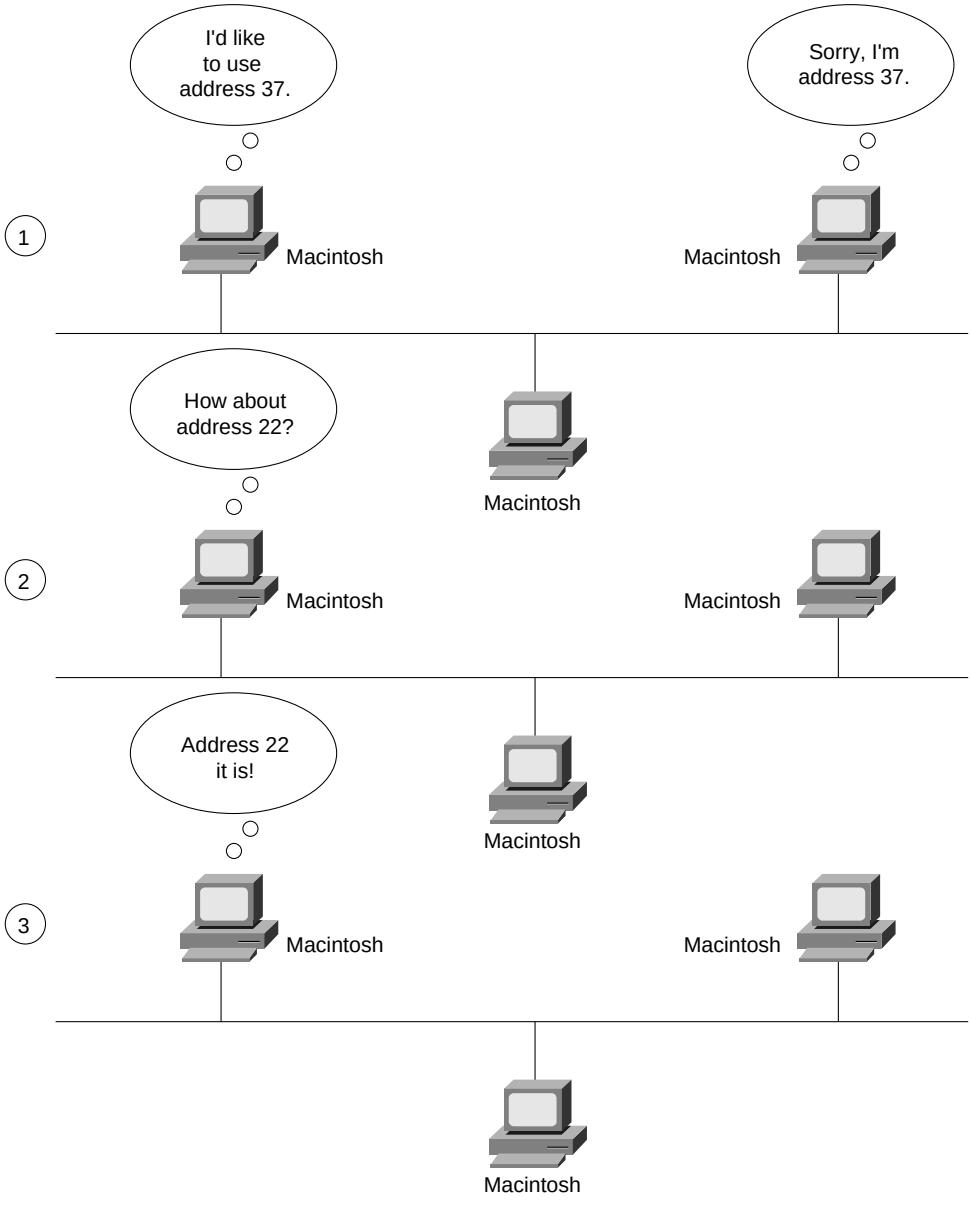
LocalTalk is Apple's proprietary media-access system. It is based on contention access, bus topology, and baseband signaling, and runs on shielded twisted-pair media at 230.4 kbps. The physical interface is EIA/TIA-422, a balanced electrical interface supported by EIA/TIA-449. LocalTalk segments can span up to 300 meters and support a maximum of 32 nodes.

The AppleTalk Network Layer

This section describes AppleTalk network-layer concepts and protocols. It includes discussion of protocol address assignment, network entities, and AppleTalk protocols that provide OSI reference model Layer 3 functionality.

AppleTalk Protocol Address Assignment To ensure minimal network administrator overhead, AppleTalk node addresses are assigned dynamically. When a Macintosh running AppleTalk starts up, it chooses a protocol (that is, network-layer) address and checks to see whether that address is currently in use. If not, the new node has successfully assigned itself an address. If the address is currently in use, the node with the conflicting address sends a message indicating a problem, and the new node chooses another address and repeats the process. Figure 2-71 shows the AppleTalk address selection process.

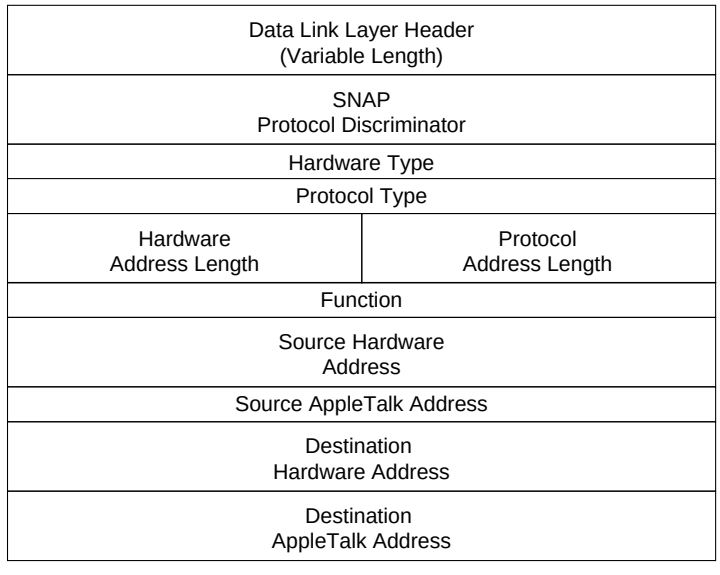
Figure 2-71 *AppleTalk clients dynamically acquire network and node addresses.*



The actual mechanics of AppleTalk address selection are media dependent. AppleTalk Address Resolution Protocol (AARP) is used to associate AppleTalk addresses with particular media addresses. When AppleTalk must send a packet to another network node, the protocol address is passed to AARP. AARP first checks an address cache to see whether the relationship between

the protocol and the hardware address is already known. If it is known, that relationship is passed up to the inquiring protocol stack. If it is not known, AARP initiates a broadcast or multicast message inquiring about the hardware address for the protocol address in question. If the broadcast reaches a node with the specified protocol address, that node replies with its hardware address. This information is passed up to the inquiring protocol stack, which uses the hardware address in communications with that node. Figure 2-72 shows the AARP packet format.

Figure 2-72 *AARP packets use SNAP encapsulation.*



The SNAP protocol discriminator field defined for AARP is the hexadecimal value 80F3.

The hardware type field has either the value 1 indicating Ethernet or 2 indicating Token Ring as the data link.

The protocol type field indicates the AppleTalk protocol family and has the value 809B.

The 1-byte hardware address length, with the value 6, indicates the length in bytes of the field containing the Ethernet or Token Ring address.

The 1-byte protocol address length, with the value 4, indicates the length in bytes of the field containing the AppleTalk protocol address. (The high byte of the address field must be set to 0, followed by the 2-byte network number, and then the 1-byte node ID.)

The function field indicates the type of AARP packet, where 1 = request, 2 = response, and 3 = probe.

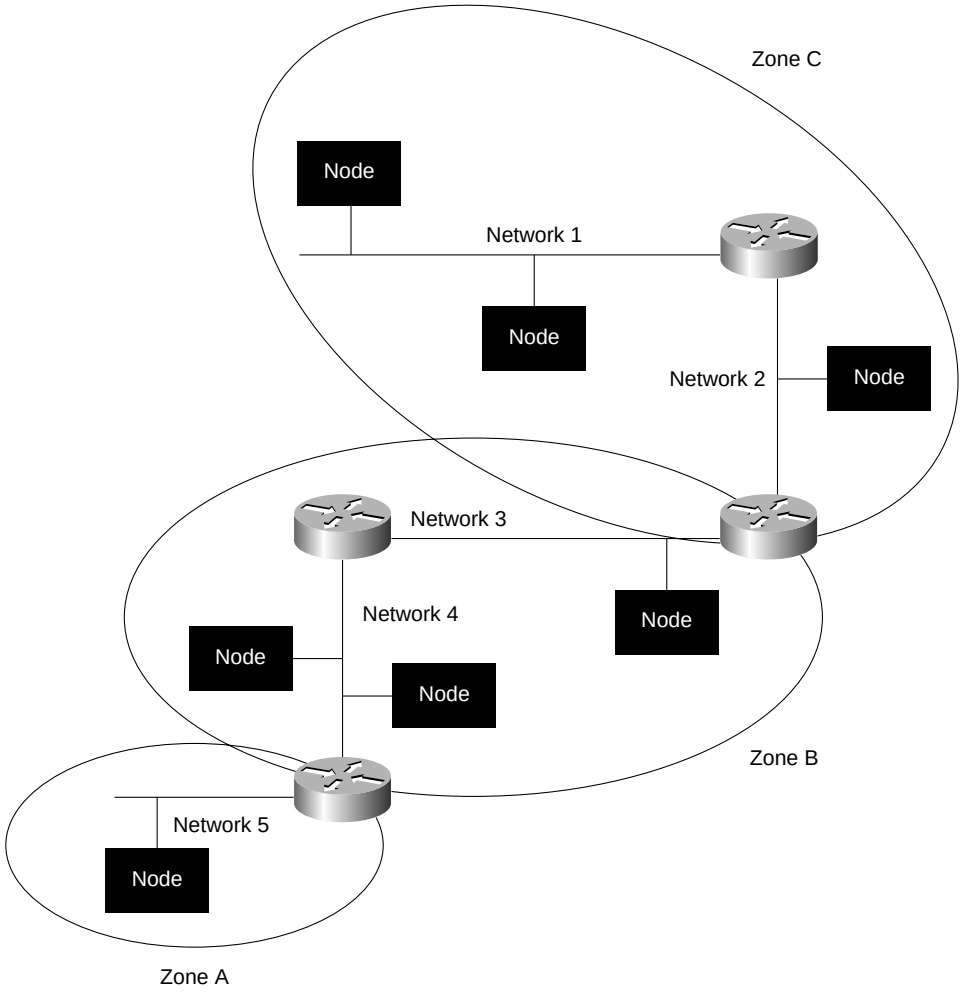
The remainder of the AARP packet contains the source and destination hardware and AppleTalk addresses, the latter always in a 4-byte field with the upper byte set to 0.

AppleTalk Network Entities AppleTalk identifies several network entities. The most elemental is a *node*, which is any device connected to an AppleTalk network. The most common nodes are Macintosh computers and laser printers, but many other types of computers are also capable of AppleTalk communication, including IBM PCs, Digital Equipment Corporation VAX computers, and a variety of workstations.

The next entity defined by AppleTalk is the *network*. An AppleTalk network is a single logical cable. Although the logical cable is frequently a single physical cable, some sites use bridges to interconnect several physical cables.

Finally, an AppleTalk *zone* is a logical group of (possibly noncontiguous) nodes. Figure 2-73 shows these AppleTalk entities.

Figure 2-73 *An AppleTalk network consists of a hierarchy of components.*



DDP AppleTalk's primary network-layer protocol is Datagram Delivery Protocol (DDP). DDP provides connectionless service between network sockets. Sockets can be assigned either statically or dynamically.

AppleTalk addresses, which are administered by the DDP, consist of two components: a 16-bit network number and an 8-bit node number. The two components are usually written as decimal numbers, separated by a period (for example, 10.1 means network 10, node 1). When an 8-bit socket identifying a particular process is added to the network number and node number, a unique process on a network is specified.

AppleTalk Phase 2 distinguishes between nonextended and extended networks. In a nonextended network such as LocalTalk, each AppleTalk node number is unique. Nonextended networks were the sole network type defined in AppleTalk Phase 1. In an extended network such as EtherTalk or TokenTalk, each network number/node number combination is unique.

Zones are defined by the AppleTalk network manager during the router configuration process. Each node in an AppleTalk network belongs to a single specific zone. Extended networks can have multiple zones associated with them. Nodes on extended networks can belong to any single zone associated with the extended network.

Figure 2-74 shows the format of the short header DDP packet and the extended header DDP packet.

The short DDP header is used on nonextended networks when source and destination sockets have the same network number. The extended DDP header is used on extended networks. An extended DDP header is also used on nonextended networks between sockets with different network numbers.

The first 2 bytes contain the 10-bit datagram length field (length of the header plus data), with the most significant bits in the first byte. In the short DDP header, the upper 6 bits of the first byte are not significant and are set to 0. In the extended header, 4 of these bits are used for a hop count field. The source node of the datagram sets the hop count field to 0 before sending the datagram. Each intermediate router increments this field, up to a maximum of 15.

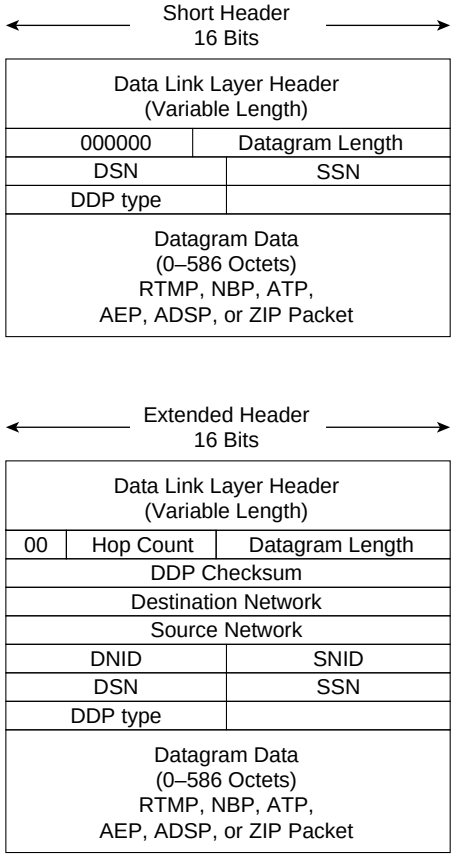
The following fields—DDP checksum, destination and source network numbers, and destination and source node IDs—are applicable to the extended header only. Calculation of the DDP checksum on the header and data is optional.

The DDP checksum is followed by 16-bit destination and source network numbers and 8-bit destination and source network node IDs.

In both the short and extended headers, the next two 8-bit fields are the destination and source socket numbers, respectively. Values of DDP sockets are 1 = RTMP, 2 = names information, 4 = echoer, and 6 = zone information.

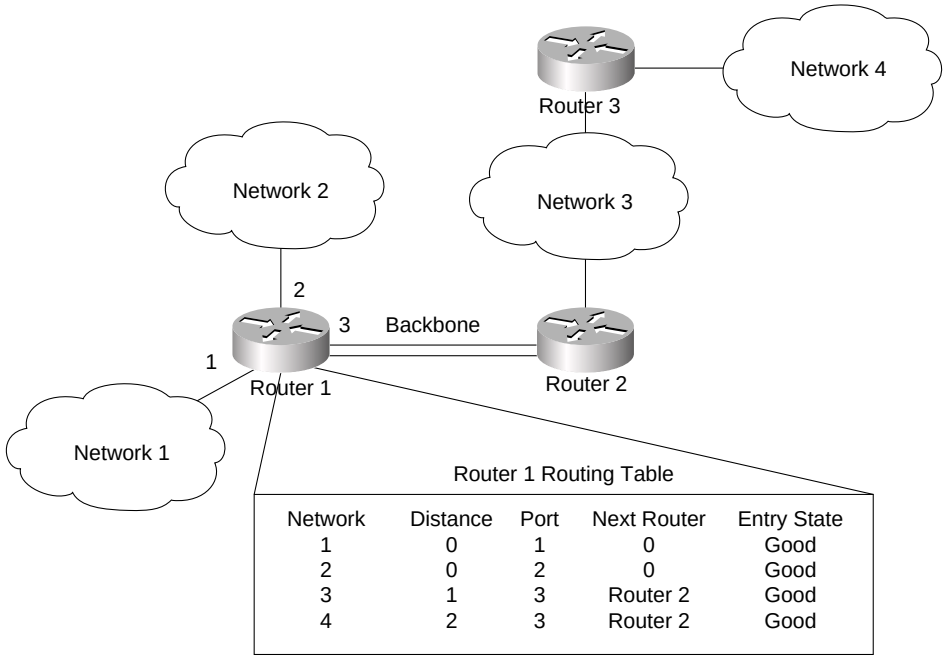
The 8-bit DDP type values are 1 = RTMP response or data packet, 2 = NBP packet, 3 = ATP packet, 4 = AEP packet, 5 = RTMP request packet, 6 = ZIP packet, and 7 = ADSP packet.

Figure 2-74 *DDP is the Layer 3 routed protocol of the AppleTalk suite.*



RTMP The protocol that establishes and maintains AppleTalk routing tables is called RTMP. An AppleTalk routing table contains an entry for each network that a datagram can reach. Each entry includes the router port that leads to the destination network, the node ID of the next router to receive the packet, the distance in hops to the destination network, and the current state of the entry (good, suspect, or bad). Periodic exchange of routing tables allows the routers in an internet to ensure that they supply current and consistent information. Figure 2-75 shows a sample routing table and the corresponding network architecture, and Figure 2-76 shows the RTMP packet format for both nonextended and extended networks.

Figure 2-75 *RTMP uses hop count as its routing metric.*



The first 2-byte field is the router's network number, that is, the sender's network number.

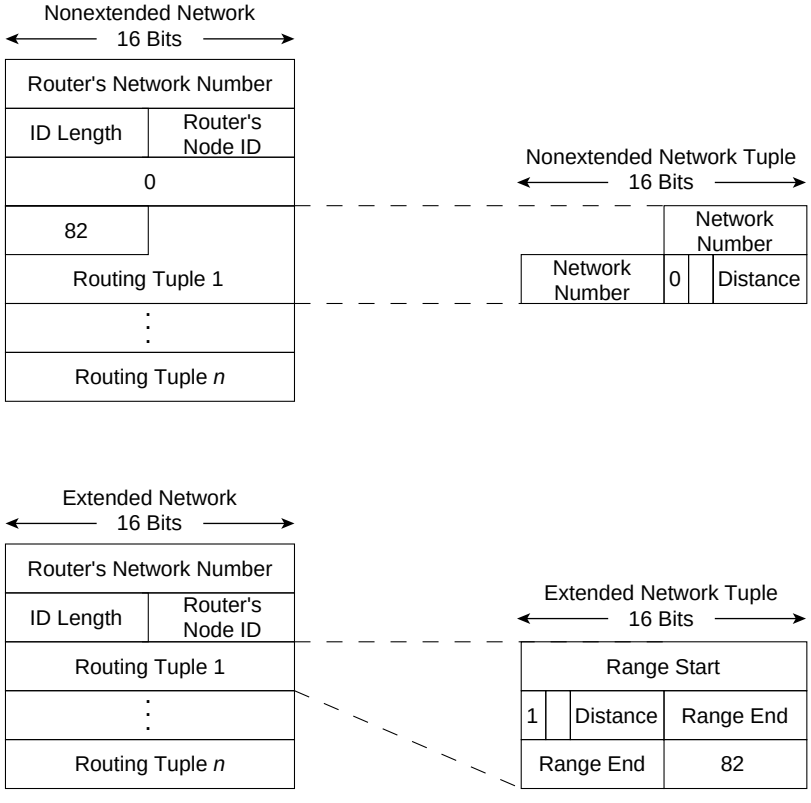
The ID length is a 1-byte field that indicates the length of the sender's node address.

The 1-byte router's node ID field indicates the sender's node address.

Following the sender's node ID in a nonextended packet is a 3-byte field indicating the version number of the packet, with a hexadecimal value of 000082. The version number of a packet sent on an extended network is specified in the first tuple.

There are two types of routing tuples. Nonextended network tuples are of the form <network number (2-bytes), distance (1-byte)>, and extended tuples are of the form <network number range start (2-bytes), distance (1-byte), network number end range (2-bytes), unused byte (value = 82)>. An extended tuple is differentiated by having the high bit of its distance field set.

Figure 2-76 *RTMP uses tuples to advertise network addresses and metrics.*

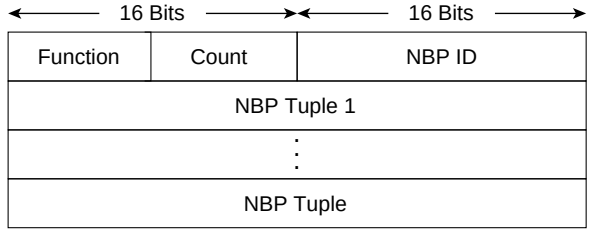


NBP AppleTalk's NBP associates AppleTalk names (expressed as network-visible entities, or NVEs) with addresses. An NVE is an AppleTalk network-addressable service, such as a socket. NVEs are associated with one or more entity names and attribute lists. Entity names are character strings such as printer@net1, and attribute lists specify NVE characteristics.

Named NVEs are associated with network addresses through the process of name binding. Name binding can be done when the user node is first started up, or dynamically, immediately before first use. NBP orchestrates the name binding process, which includes name registration, name confirmation, name deletion, and name lookup.

Figure 2-77 shows the NBP packet format.

Figure 2-77 *NBP packets carry information about network services.*



The high-order 4 bits of the first byte indicate the function of the type of NBP packet. The values are 1 = broadcast request (BrRq), 2 = lookup (LkUp), 3 = lookup reply (LkUp-Reply), and 4 = forward request (FwdReq).

The low-order 4 bits contain a count of the number of NBP tuples in that packet.

The NBP ID is used to associate LkUp-Reply packets with BrRq or LkUp packets.

The NBP tuples, the name/address pairs, consist of the entity's socket address, enumerator, and entity name.

Zones allow name lookup in a group of logically related nodes. To look up names within a zone, an NBP lookup request is sent to a local router, which sends a broadcast request to all networks that have nodes belonging to the target zone.

ZIP maintains network number to zone name mappings in zone information tables (ZITs). ZITs are stored in routers, which are the primary users of ZIP, but end nodes also use ZIP during the startup process to choose their zones and to acquire internetwork zone information. RTMP uses ZIP to maintain a list of zones for known networks. When RTMP learns of a new network it sends a ZIP query to request the zone names. Figure 2-78 shows a sample ZIT.

Figure 2-78 *The zone information table maps zone names to cable ranges.*

Network Number	Zone
1–1	My
2–2	Your
3–3	Marketing
4–4	Documentation
5–5	Sales

The AppleTalk Transport Layer

AppleTalk's transport layer is implemented by two primary Apple protocols: ATP and AppleTalk Data Stream Protocol (ADSP). ATP is transaction oriented, and ADSP is data-stream oriented.

ATP ATP is one of AppleTalk's transport-layer protocols. ATP is suitable for transaction-based applications such as those found in banks or retail stores.

ATP transactions consist of requests (from clients) and replies (from servers). Each request/reply pair has a particular transaction ID. Transactions occur between two socket clients. ATP uses exactly once (XO) and at-least-once (ALO) transactions. XO transactions are required in situations where accidentally performing the transaction more than once is unacceptable. Bank transactions are examples of such nonidempotent situations (that is, situations where repeating a transaction causes problems by invalidating the data involved in the transaction).

ATP is capable of most important transport-layer functions, including data acknowledgment and retransmission, packet sequencing, and fragmentation and reassembly. ATP limits message segmentation to eight packets, and ATP packets cannot contain more than 578 data bytes.

ADSP ADSP is another important AppleTalk transport-layer protocol. As its name implies, ADSP is data-stream oriented rather than transaction oriented. It establishes and maintains full-duplex data streams between two sockets in an AppleTalk internetwork.

ADSP is a reliable protocol in that it guarantees that data bytes will be delivered in the same order in which they were sent and that they are not duplicated. ADSP numbers each data byte to keep track of the individual elements of the data stream.

ADSP also specifies a flow-control mechanism. The destination can essentially slow source transmissions by reducing the size of its advertised receive window.

ADSP also provides an out-of-band control message mechanism. Attention packets are used as the vehicle for movement of out-of-band control messages between two AppleTalk entities. These packets use a separate sequence number stream to differentiate them from normal ADSP data packets.

AppleTalk Upper-Layer Protocols

AppleTalk supports several upper-layer protocols. AppleTalk Session Protocol (ASP) establishes and maintains sessions (logical conversations) between an AppleTalk client and a server. AppleTalk's PAP is a connection-oriented protocol that establishes and maintains connections between clients and print servers. AppleTalk Echo Protocol (AEP) is an extremely simple protocol that generates packets that can be used to test the reachability of various network nodes. Finally, AFP helps clients share server files across a network.

Summary

This chapter focuses on the most popular media access types, such as IEEE 802.3, IEEE 802.5, and FDDI, as well as the most popular protocol suites, such as TCP/IP, Novell's IPX/SPX, and AppleTalk. Knowing the basic technical details of these communication methods will help you understand how the network should function and recognize when the communication "rules" are being broken.

As you use your systematic method for troubleshooting network problems, narrow the scope of facts by knowing the targets that you pay closest attention to as possible problem causes.

If the problem has affected your users' ability to make a connection, the sequence of gathering facts and considering possibilities begins by targeting the physical-layer prerequisites.

The relevant hardware and physical connections must be working before it is meaningful to check for data-link problem causes. Also, if the problem has interrupted the transfer of data, check first to see if the physical connection has been broken.

Data-link checking on Cisco routers focuses on the direct connection linking network devices across an interface. The Cisco IOS software **show interface** commands provide facts relevant to problem-solving efforts. In particular, a quick output line to check is to see if the interface is up (for example, "Ethernet 0 is up") and to see if the data link is up (for example, "line protocol is up"). We will explore the **show interface** command in Chapter 3, "Cisco Routing and Switching Processes."

With the data link layer working, you can troubleshoot targets at the layers above the data link layer. Begin by checking the basic connection sequence for the protocol suite in question. Cisco IOS commands that focus on a specific protocol suite can help you check.

Various **show** commands available in the Cisco IOS software provide you with aggregate information about routed and routing protocols. Output from these commands may contain high-level facts that can point the way to other targets as you narrow the likely cause of the problem. These commands are introduced in later chapters.

You can augment your problem-solving resources by having handy access to the breakdown of explanations for the output fields and other related Cisco IOS commands that can provide further information.

One handy method is for you to have a copy of the Cisco IOS documentation available on a CD-ROM. Check the troubleshooting guides available over your CCO session or refer to your paper-based manuals. If necessary, extend your efforts by using a third-party protocol analyzer.

Chapter 3, "Cisco Routing and Switching Processes," focuses specifically on router processing and functionality.

Chapter 2 Test
Protocol Characteristics Overview

Estimated Time: 15 minutes

Complete all the exercises to test your knowledge of the materials contained in this chapter. Answers are listed in Appendix A, "Chapter Test Answer Key."

Use the information contained in this chapter to answer the following questions.

Question 2.1

Define the two types of protocols that are used to communicate between devices.

a. _____

b. _____

Question 2.2

List at least two connection-oriented protocols.

a. _____

b. _____

Question 2.3

Which OSI reference model layer defines flow-control functionality?

_____

Question 2.4

What field begins both Ethernet and 802.3 frame types?

_____

Question 2.5

What Token Ring technology supports multiple frames on the cable at one time?

_____

Question 2.6

What other media access type is FDDI similar to?

_____

Question 2.7

Differentiate the following types of 802.2 service from one another:

a. LLC1: _____

b. LLC2: _____

c. LLC3: _____

Question 2.8

What is the purpose of a Frame Relay DLCI?

_____

_____

Question 2.9

What TCP/IP protocol is used to obtain a hardware address for an IP device?

_____

Question 2.10

What TCP/IP protocol set is Novell's IPX most closely related to?

_____

Question 2.11

What are the link-state routing protocols used by TCP/IP and IPX/SPX?

_____

Question 2.12

What is the service discovery mechanism used with AppleTalk?

_____

Question 2.13

What are the three phases typically seen on connection-oriented communications?

a. _____

b. _____

c. _____

Question 2.14

List at least two key advantages that connectionless protocols have over connection-oriented protocols:

a. _____

b. _____

Question 2.15

What connection-oriented and connectionless transport-layer protocols are used in TCP/IP communications?

a. Connection-oriented: _____

b. Connectionless: _____

Question 2.16

Match the following terms with their definitions:

a. Ethernet	**1.** 4- or 16-Mbps bandwidth
b. Token Ring	**2.** Similar to 802.3
c. FDDI	**3.** Contains DSAP/SSAP fields
d. 802.2	**4.** Dual-ring LAN

Question 2.17

Which fiber mode is capable of higher bandwidth and greater cable runs?

a. Single mode

b. Multimode

Question 2.18

What TCP/IP protocol reports routing errors back to the source of a communication?

_____

Question 2.19

What technologies help prevent the count-to-infinity problem?

_____

_____

Question 2.20

Is BGP an interior gateway protocol or an exterior gateway protocol?

_____

Question 2.21

List and define Novell's two routing protocols.

a. _____

b. _____

Question 2.22

Which three of the following attributes fit the definition of ATM networking?

a. Uses variable-length cells

b. Built to carry voice, video, and data traffic

c. Primarily a connection-oriented network

d. Uses efficient synchronous communications

e. Supports virtual paths and virtual channels