AppleTalk

Background

AppleTalk, a protocol suite developed by Apple Computer in the early 1980s, was developed in conjunction with the Macintosh computer. AppleTalk's purpose was to allow multiple users to share resources, such as files and printers. The devices that supply these resources are called *servers*, while the devices that make use of these resources (such as a user's Macintosh computer) are referred to as *clients*. Hence, AppleTalk is one of the early implementations of a distributed client-server networking system. This chapter provides a summary of AppleTalk's network architecture.

AppleTalk was designed with a transparent network interface. That is, the interaction between client computers and network servers requires little interaction from the user. In addition, the actual operations of the AppleTalk protocols are invisible to end users, who see only the result of these operations. Two versions of AppleTalk exist: AppleTalk Phase 1 and AppleTalk Phase 2.

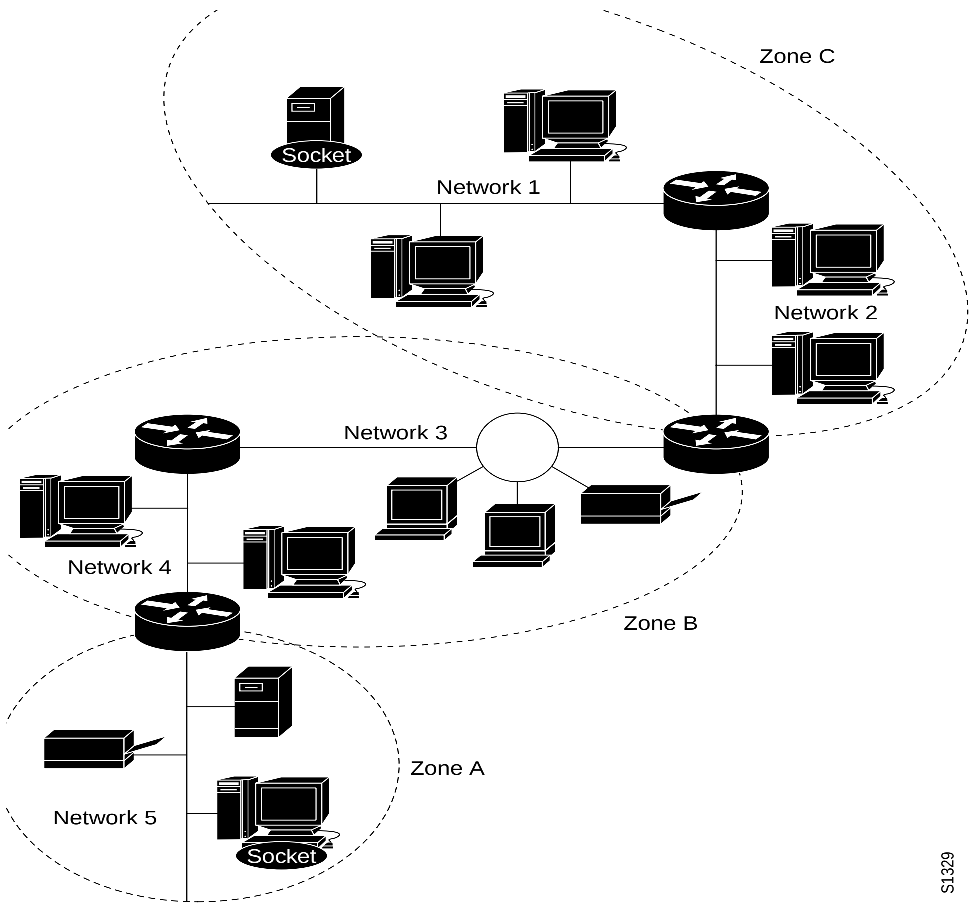
AppleTalk Phase 1, which is the first AppleTalk specification, was developed in the early 1980s strictly for use in local workgroups. Phase 1 therefore has two key limitations: its network segments can contain no more than 127 hosts and 127 servers, and it can support only nonextended networks. Extended and nonextended networks will be discussed in detail later in this chapter.

AppleTalk Phase 2, which is the second enhanced AppleTalk implementation, was designed for use in larger internetworks. Phase 2 addresses the key limitations of AppleTalk Phase 1 and features a number of improvements over Phase 1. In particular, Phase 2 allows any combination of 253 hosts or servers on a single AppleTalk network segment and supports both nonextended and extended networks.

AppleTalk Network Components

AppleTalk networks are arranged hierarchically. Four basic components form the basis of an AppleTalk network: *sockets*, *nodes*, *networks*, and *zones*. Figure 27-1 illustrates the hierarchical organization of these components in an AppleTalk internetwork. Each of these concepts is summarized in the sections that follow.

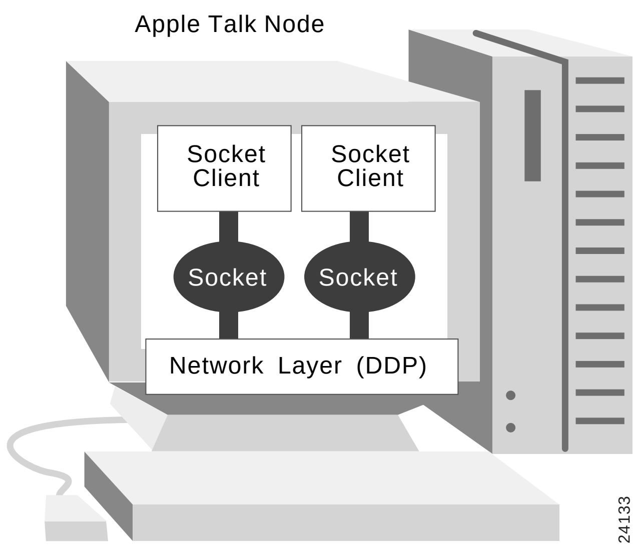
Figure 27-1 The AppleTalk internetwork consists of a hierarchy of components.



Sockets

An AppleTalk socket is a unique, addressable location in an AppleTalk node. It is the logical point at which upper-layer AppleTalk software processes and the network-layer Datagram-Delivery Protocol (DDP) interact. These upper-layer processes are known as *socket clients*. Socket clients own one or more sockets, which they use to send and receive datagrams. Sockets can be assigned statically or dynamically. Statically assigned sockets are reserved for use by certain protocols or other processes. Dynamically assigned sockets are assigned by DDP to socket clients upon request. An AppleTalk node can contain up to 254 different socket numbers. Figure 27-2 illustrates the relationship between the sockets in an AppleTalk node and DDP at the network layer.

Figure 27-2 Socket clients use sockets to send and receive datagrams.



Nodes

An AppleTalk node is a device that is connected to an AppleTalk network. This device might be a Macintosh computer, a printer, an IBM PC, a router, or some other similar device. Within each AppleTalk node exist numerous software processes called sockets. As discussed earlier, the function of these sockets is to identify the software processes running in the device. Each node in an AppleTalk network belongs to a single network and a specific zone.

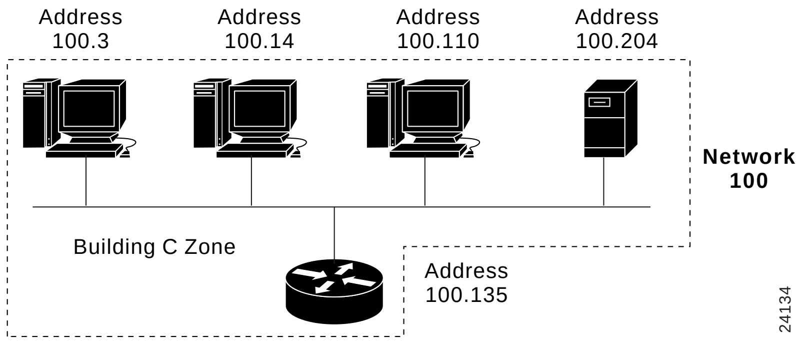
Networks

An AppleTalk network consists of a single logical cable and multiple attached nodes. The logical cable is composed of either a single physical cable or multiple physical cables interconnected by using bridges or routers. AppleTalk networks can be nonextended or extended. Each is discussed briefly in the following sections.

Nonextended Networks

A *nonextended* AppleTalk network is a physical-network segment that is assigned only a single network number, which can range between 1 and 1,024. Network 100 and Network 562, for example, are both valid network numbers in a nonextended network. Each node number in a nonextended network must be unique, and a single nonextended network segment cannot have more than one AppleTalk Zone configured on it. (A zone is a logical group of nodes or networks.) AppleTalk Phase 1 supports only nonextended networks, but as a rule, nonextended network configurations are no longer used in new networks because they have been superseded by extended networks. Figure 27-3 illustrates a nonextended AppleTalk network.

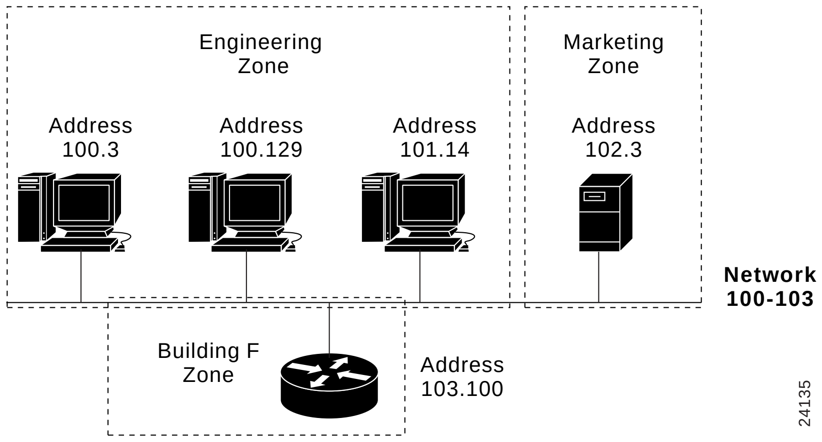
Figure 27-3 A nonextended network is assigned only one network number.



Extended Networks

An *extended* AppleTalk network is a physical-network segment that can be assigned multiple network numbers. This configuration is known as a *cable range*. AppleTalk cable ranges can indicate a single network number or multiple consecutive network numbers. The cable ranges Network 3-3 (unary) and Network 3-6, for example, are both valid in an extended network. Just as in other protocol suites, such as TCP/IP and IPX, each combination of network number and node number in an extended network must be unique, and its address must be unique for identification purposes. Extended networks can have multiple AppleTalk zones configured on a single network segment, and nodes on extended networks can belong to any single zone associated with the extended network. Extended network configurations have, as a rule, replaced nonextended network configurations. Figure 27-4 illustrates an extended network.

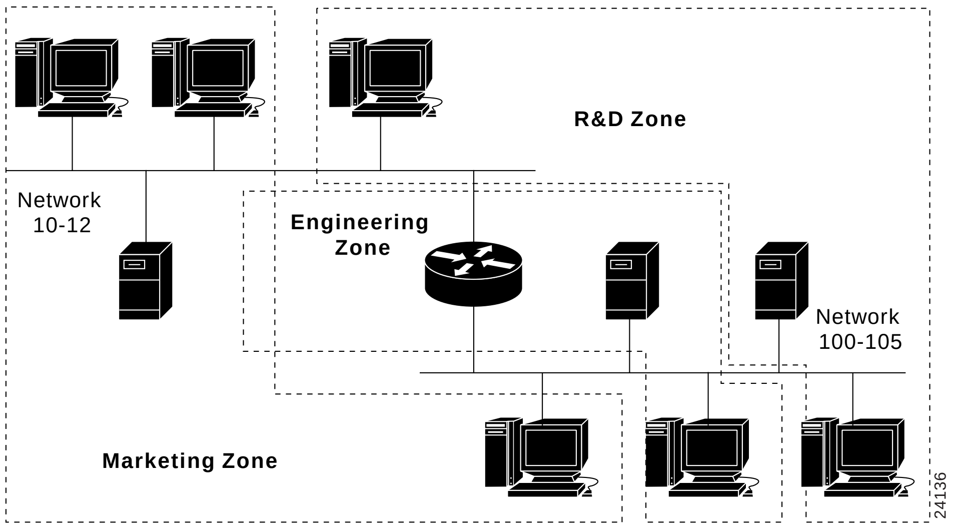
Figure 27-4 An extended network can be assigned multiple network numbers.



Zones

An AppleTalk *zone* is a logical group of nodes or networks that is defined when the network administrator configures the network. The nodes or networks need not be physically contiguous to belong to the same AppleTalk zone. Figure 27-5 illustrates an AppleTalk internetwork composed of three noncontiguous zones.

Figure 27-5 Nodes or networks in the same zone need not be physically contiguous.

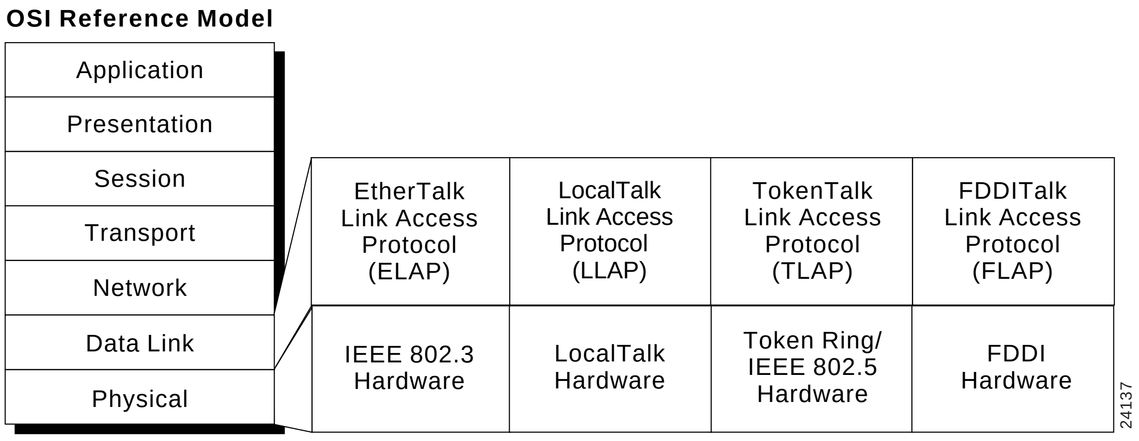


AppleTalk Physical and Data Link Layers

As with other popular protocol suites, such as TCP/IP and IPX, the AppleTalk architecture maintains media-access dependencies on such lower-layer protocols as Ethernet, Token Ring, and FDDI. Four main media-access implementations exist in the AppleTalk protocol suite: EtherTalk, LocalTalk, TokenTalk, and FDDITalk.

These data link-layer implementations perform address translation and other functions that allow proprietary AppleTalk protocols to communicate over industry-standard interfaces, which include IEEE 802.3 (using EtherTalk), Token Ring/IEEE 802.5 (using TokenTalk), and FDDI (using FDDITalk). In addition, AppleTalk implements its own network interface, known as LocalTalk. Figure 27-6 illustrates how the AppleTalk media-access implementations map to the OSI reference model.

Figure 27-6 AppleTalk media-access maps to the bottom two layers of the OSI reference model.


OSI Reference Model
Application
Presentation
Session
Transport
Network
Data Link
Physical
EtherTalk Link Access Protocol (ELAP)
LocalTalk Link Access Protocol (LLAP)
TokenTalk Link Access Protocol (TLAP)
FDDITalk Link Access Protocol (FLAP)
IEEE 802.3 Hardware
LocalTalk Hardware
Token Ring/ IEEE 802.5 Hardware
FDDI Hardware
24137

EtherTalk

EtherTalk extends the data link layer to enable the AppleTalk protocol suite to operate atop a standard IEEE 802.3 implementation. EtherTalk networks are organized exactly as IEEE 802.3 networks, supporting the same speeds and segment lengths, as well as the same number of active network nodes. This allows AppleTalk to be deployed over any of the thousands of Ethernet-based networks in existence today. Communication between the upper-layer protocols of the AppleTalk architecture and the Ethernet protocols is handled by the EtherTalk Link-Access Protocol (ELAP).

EtherTalk Link-Access Protocol

The *EtherTalk Link-Access Protocol* (ELAP) handles the interaction between the proprietary AppleTalk protocols and the standard IEEE 802.3 data link layer. Upper-layer AppleTalk protocols do not recognize standard IEEE 802.3 hardware addresses, so ELAP uses the *Address-Mapping Table* (AMT) maintained by the *AppleTalk Address-Resolution Protocol* (AARP) to properly address transmissions.

ELAP handles the interaction between upper-layer protocols of AppleTalk and the data link layer by encapsulating or enclosing the data inside the protocol units of the 802.3 data link layer. ELAP performs three levels of encapsulation when transmitting DDP packets:

- Subnetwork-Access Protocol (SNAP) header
- IEEE 802.2 Logical Link-Control (LLC) header
- IEEE 802.3 header

This process of encapsulation performed by the ELAP is detailed in the following section.

ELAP Data-Transmission Process

ELAP uses a specific process to transmit data across the physical medium. First, ELAP receives a DDP packet that requires transmission. Next, it finds the protocol address specified in the DDP header and checks the AMT to find the corresponding IEEE 802.3 hardware address. ELAP then prepends three different headers to the DDP packet, beginning with the SNAP and 802.2 LLC headers. The third header is the IEEE 802.3 header. When prepending this header to the packet, the hardware address taken from the AMT is placed in the Destination Address field. The final result, an IEEE 802.3 frame, is placed on the physical medium for transmission to the destination.

LocalTalk

LocalTalk, which is a proprietary data link-layer implementation developed by Apple Computer for its AppleTalk protocol suite, was designed as a cost-effective network solution for connecting local workgroups. LocalTalk hardware typically is built into Apple products, which are easily connected by using inexpensive twisted-pair cabling. LocalTalk networks are organized in a bus topology, which means that devices are connected to each other in series. Network segments are limited to a 300-meter span with a maximum of 32 active nodes, and multiple LocalTalk networks can be interconnected by using routers or other similar intermediate devices. The communication between the data link-layer protocol LocalTalk and upper-layer protocols is the LocalTalk Link-Access Protocol (LLAP).

LocalTalk Link-Access Protocol

The *LocalTalk Link-Access Protocol* (LLAP) is the media-access protocol used in LocalTalk networks to provide best-effort, error-free delivery of frames between AppleTalk nodes. This means that delivery of datagrams is not guaranteed by the LLAP; such a function is performed only by higher-layer protocols in the AppleTalk architecture. LLAP is responsible for regulating node access to the physical media and dynamically acquiring data link-layer node addresses.

Regulating Node Access to the Physical Media

LLAP implements a media-access scheme known as *carrier-sense multiple access, collision avoidance* (CSMA/CA), whereby nodes check the link to see if it is in use. The link must be idle for a certain random period of time before a node can begin transmitting data. LLAP uses data exchanges known as *handshakes* to avoid *collisions* (that is, simultaneous transmissions by two or more nodes). A successful handshake between nodes effectively reserves the link for their use. If two nodes transmit a handshake simultaneously, the transmissions collide. In this case, both transmissions are damaged, causing the packets to be discarded. The handshake exchange is not completed, and the sending nodes infer that a collision occurred. When the collision occurs, the device will remain idle for a random period of time and then retry its transmission. This process is similar to the access mechanism used with Ethernet technology.

Acquiring Node Addresses

LLAP acquires data link-layer node addresses dynamically. The process allows a unique data link-layer address to be assigned without permanently assigning the address to the node. When a node starts up, LLAP assigns the node a randomly chosen node identifier (node ID). The uniqueness of this node ID is determined by the transmission of a special packet that is addressed to the randomly chosen node ID. If the node receives a reply to this packet, the node ID is not unique. The node therefore is assigned another randomly chosen node ID and will send out another packet addressed to that node until no reply returns. If the acquiring node does not receive a reply to the first query, it makes a number of subsequent attempts. If there is still no reply after these attempts, the node ID is considered unique, and the node uses this node ID as its data link-layer address.

TokenTalk

TokenTalk extends the data link layer to allow the AppleTalk protocol suite to operate atop a standard IEEE 802.5/Token Ring implementation. TokenTalk networks are organized exactly as IEEE 802.5/Token Ring networks, supporting the same speeds and the same number of active network nodes. Communication between the data link-layer protocols used with Token Ring and upper-layer protocols is the TokenTalk Link-Access Protocol (TLAP).

TokenTalk Link-Access Protocol

The *TokenTalk Link-Access Protocol* (TLAP) handles the interaction between the proprietary AppleTalk protocols and the standard IEEE 802.5 data-link layer. Upper-layer AppleTalk protocols do not recognize standard IEEE 802.5 hardware addresses, so TLAP uses the AMT maintained by the AARP to properly address transmissions. TLAP performs three levels of encapsulation when transmitting DDP packets:

- Subnetwork-Access Protocol (SNAP) header
- IEEE 802.2 Logical Link-Control (LLC) header
- IEEE 802.5 header

TLAP Data Transmission Process

TLAP data transmission involves a number of steps to transmit data across the physical medium. When TLAP receives a DDP packet that requires transmission, it finds the protocol address specified in the DDP header and then checks the AMT to find the corresponding IEEE 802.5/Token Ring hardware address. Next, TLAP prepends three different headers to the DDP packet, beginning with the SNAP and 802.2 LLC headers. When the third header, IEEE 802.5/Token Ring, is prepended to the packet, the hardware address received from the AMT is placed in the Destination Address field. The final result, an IEEE 802.5/Token Ring frame, is placed on the physical medium for transmission to the destination.

FDDITalk

FDDITalk extends the data link layer to allow the AppleTalk protocol suite to operate atop a standard ANSI FDDI implementation. FDDITalk networks are organized exactly as FDDI networks, supporting the same speeds and the same number of active network nodes.

FDDITalk Link-Access Protocol

The *FDDITalk Link-Access Protocol* (FLAP) handles the interaction between the proprietary AppleTalk protocols and the standard FDDI data link layer. Upper-layer AppleTalk protocols do not recognize standard FDDI hardware addresses, so FLAP uses the AMT maintained by the AARP to properly address transmissions. FLAP performs three levels of encapsulation when transmitting DDP packets:

- Subnetwork-Access Protocol (SNAP) header
- IEEE 802.2 Logical Link Control (LLC) header
- FDDI header

FLAP Data-Transmission Process

As with TLAP, FLAP involves a multi-stage process to transmit data across the physical medium. When FLAP receives a DDP packet requiring transmission, it finds the protocol address specified in the DDP header and then checks the AMT to find the corresponding FDDI hardware address. FLAP then prepends three different headers to the DDP packet, beginning with the SNAP and 802.2 LLC headers. When the third header, the FDDI header, is prepended to the packet, the hardware address received from the AMT is placed in the Destination Address field. The final result, an FDDI frame, is placed on the physical medium for transmission to the destination.

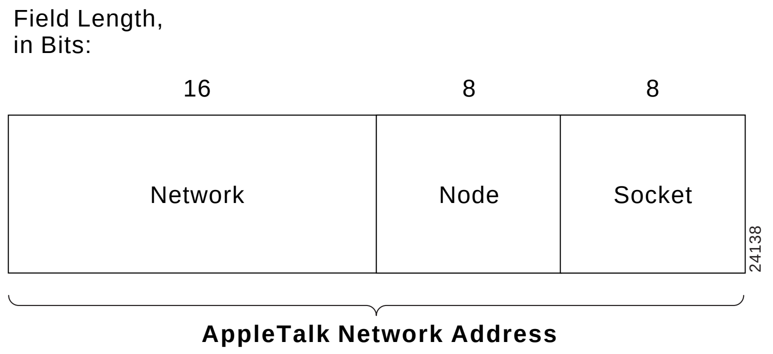
Network Addresses

AppleTalk utilizes addresses to identify and locate devices on a network in a manner similar to the process utilized by such common protocols as TCP/IP and IPX. These addresses, which are assigned dynamically as discussed in the following section, are composed of three elements:

- *Network number*—A 16-bit value that identifies a specific AppleTalk network (either nonextended or extended).
- *Node number*—An 8-bit value that identifies a particular AppleTalk node attached to the specified network.
- *Socket number*—An 8-bit number that identifies a specific socket running on a network node.

AppleTalk addresses usually are written as decimal values separated by a period. For example, 10.1.50 means Network 10, Node 1, Socket 50. This also might be represented as 10.1, socket 50. Figure 27-7 illustrates the AppleTalk network address format.

Figure 27-7 The AppleTalk network address consists of three distinct numbers.



Network-Address Assignment

One of the unique characteristics of AppleTalk is the dynamic nature of device addresses. It is not necessary to statically define an address to an AppleTalk device. Rather, AppleTalk nodes are assigned addresses dynamically when they first attach to a network.

When an AppleTalk network node starts up, it receives a provisional network-layer address. The network portion of the provisional address (the first 16 bits) is selected from the *startup range,* which is a reserved range of network addresses (values 65280 to 65534). The node portion (the next 8 bits) of the provisional address is chosen randomly.

Using the *Zone-Information Protocol (ZIP)*, the node communicates with a router attached to the network. The router replies with the valid cable range for the network to which the node is attached. Next, the node selects a valid network number from the cable range supplied by the router and then randomly chooses a node number. A broadcast message is used to determine whether the selected address is in use by another node.

If the address is not being used (that is, no other node responds to the broadcast within a specific period of time), the node has successfully been assigned an address. If, however, another node is using the address, that node responds to the broadcast with a message indicating that the address is in use. The new node must choose another address and repeat the process until it selects an address that is not in use.

AppleTalk Address-Resolution Protocol (AARP)

AppleTalk Address-Resolution Protocol (AARP) is a network-layer protocol in the AppleTalk protocol suite that associates AppleTalk network addresses with hardware addresses. AARP services are used by other AppleTalk protocols. When an AppleTalk protocol has data to transmit, for example, it specifies the network address of the destination. It is the job of AARP to find the hardware address that is associated with the device using that network address.

AARP uses a request-response process to learn the hardware address of other network nodes. Because AARP is a media-dependent protocol, the method used to request a hardware address from a node varies depending on the data link-layer implementation. Typically, a broadcast message is sent to all AppleTalk nodes on the network.

Address-Mapping Table

Each AppleTalk node contains an Address-Mapping Table (AMT), where hardware addresses are associated with network addresses. Each time AARP resolves a network and hardware address combination, the mapping is recorded in the AMT.

Over time, the potential for an AMT entry to become invalid increases. For this reason, each AMT entry typically has a timer associated with it. When AARP receives a packet that verifies or changes the entry, the timer is reset.

If the timer expires, the entry is deleted from the AMT. The next time an AppleTalk protocol wants to communicate with that node, another AARP request must be transmitted to discover the hardware address.

Address Gleaning

In certain implementations, incoming DDP packets are examined to learn the hardware and network addresses of the source node. DDP then can place this information in the AMT. This is one way in which a device, such as a router, workstation, or server, can discover devices within an AppleTalk network.

This process of obtaining address mappings from incoming packets is known as *address gleaning*. Address gleaning is not widely used, but in some situations it can reduce the number of AARP requests that must be transmitted.

AARP Operation

The AppleTalk Address-Resolution Protocol (AARP) maps hardware addresses to network addresses. When an AppleTalk protocol has data to send, it passes the network address of the destination node to AARP. It is the job of AARP to supply the hardware address associated with that network address.

AARP checks the AMT to see whether the network address is already mapped to a hardware address. If the addresses are already mapped, the hardware address is passed to the inquiring AppleTalk protocol, which uses it to communicate with the destination. If the addresses are not mapped, AARP transmits a broadcast requesting that the node using the network address in question supply its hardware address.

When the request reaches the node using the network address, that node replies with its hardware address. If no node exists with the specified network address, no response is sent. After a specified number of retries, AARP assumes that the protocol address is not in use and returns an error to the inquiring AppleTalk protocol. If a response is received, the hardware address is associated to the network address in the AMT. The hardware address then is passed to the inquiring AppleTalk protocol, which uses it to communicate with the destination node.

Datagram Delivery Protocol (DDP) Overview

The Datagram Delivery Protocol (DDP) is the primary network-layer routing protocol in the AppleTalk protocol suite that provides a best-effort connectionless datagram service between AppleTalk sockets. As with protocols such as TCP, no virtual circuit or connection is established between two devices. The function of guaranteeing delivery instead is handled by upper-layer protocols of the AppleTalk protocol suite. These upper-layer protocols will be discussed later in this chapter.

DDP performs two key functions: packet transmission and receipt.

- *Transmission of packets*—DDP receives data from socket clients, creates a DDP header by using the appropriate destination address, and passes the packet to the data link-layer protocol.
- *Reception of packets*—DDP receives frames from the data link layer, examines the DDP header to find the destination address, and routes the packet to the destination socket.

DDP maintains the cable range of the local network and the network address of a router attached to the local network in every AppleTalk node. In addition to this information, AppleTalk routers must maintain a routing table by using the Routing Table Maintenance Protocol (RTMP).

DDP Transmission Process

DDP operates much like any routing protocol. Packets are addressed at the source, passed to the data link layer, and transmitted to the destination. When DDP receives data from an upper-layer protocol, it determines whether the source and destination nodes are on the same network by examining the network number of the destination address. If the destination network number is within the cable range of the local network, the packet is encapsulated in a DDP header and is passed to the data link layer for transmission to the destination node. If the destination network number is not within the cable range of the local network, the packet is encapsulated in a DDP header and is passed to the data link layer for transmission to a router. Intermediate routers use their routing tables to forward the packet toward the destination network. When the packet reaches a router attached to the destination network, the packet is transmitted to the destination node.

AppleTalk Transport Layer

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

Five key implementations exist at the transport layer of the AppleTalk protocol suite:

- Routing Table Maintenance Protocol (RTMP)
- Name-Binding Protocol (NBP)
- AppleTalk Update-Based Routing Protocol (AURP)
- AppleTalk Transaction Protocol (ATP)
- AppleTalk Echo Protocol (AEP)

Each of these protocol implementations are addressed briefly in the discussions that follow.

Routing Table Maintenance Protocol (RTMP) Overview

The *Routing Table Maintenance Protocol (RTMP)* is a transport-layer protocol in the AppleTalk protocol suite that establishes and maintains routing tables in AppleTalk routers.

RTMP is based on the Routing Information Protocol (RIP), and as with RIP, RTMP uses hop count as a routing metric. *Hop count* is calculated as the number of routers or other intermediate nodes through which a packet must pass to travel from the source network to the destination network.

RTMP Routing Tables

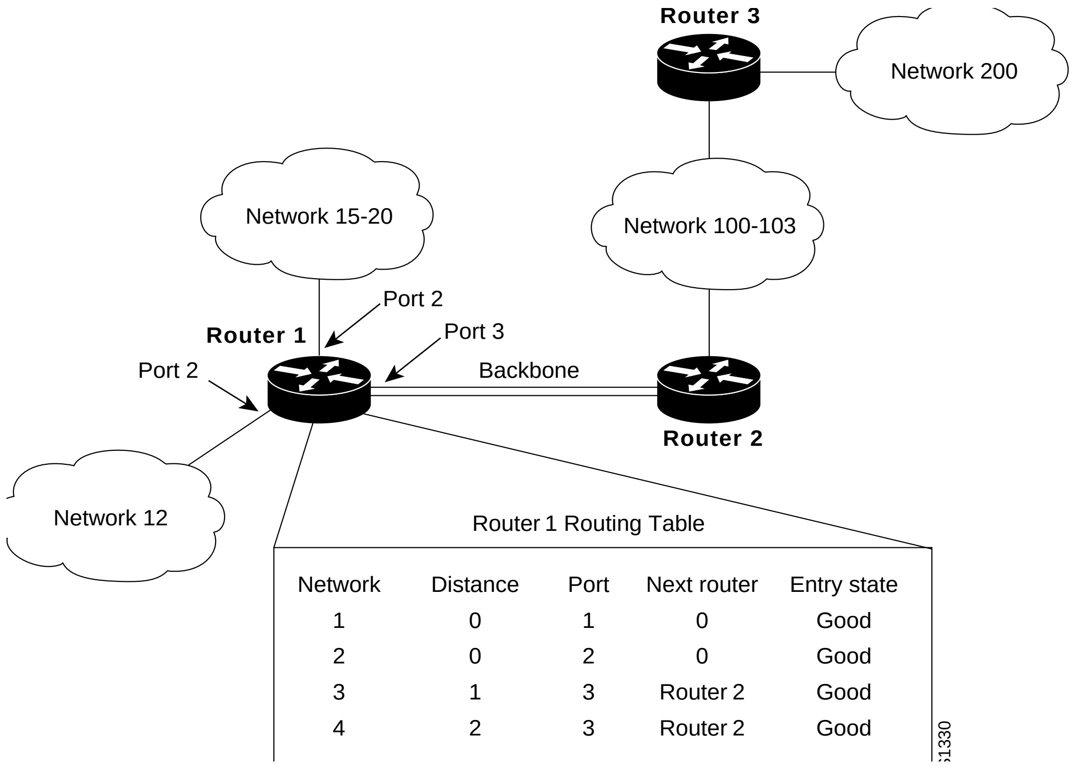
RTMP is responsible for establishing and maintaining routing tables for AppleTalk routers. These routing tables contain an entry for each network that a packet can reach.

Routers periodically exchange routing information to ensure that the routing table in each router contains the most current information, and that the information is consistent across the internetwork. An RTMP routing table contains the following information about each of the destination networks known to the router:

- Network cable range of the destination network
- Distance in hops to the destination network
- Router port that leads to the destination network
- Address of the next hop router
- Current state of the routing-table entry (good, suspect, or bad)

Figure 27-8 illustrates a typical RTMP routing table.

Figure 27-8 An RTMP routing table contains information about each destination network known to the router.



Network	Distance	Port	Next router	Entry state
1	0	1	0	Good
2	0	2	0	Good
3	1	3	Router 2	Good
4	2	3	Router 2	Good

Name-Binding Protocol (NBP) Overview

The *Name-Binding Protocol (NBP)* is a transport-layer protocol in the AppleTalk protocol suite that maps the addresses used at lower layers to AppleTalk names. Socket clients within AppleTalk nodes are known as *Network-Visible Entities* (NVEs). An NVE is a network-addressable resource, such as a print service, that is accessible over the internetwork. NVEs are referred to by character strings known as *entity names*. NVEs also have a zone and various attributes, known as *entity types*, associated with them.

Two key reasons exist for using entity names rather than addresses at the upper layers. First, network addresses are assigned to nodes dynamically and, therefore, change regularly. Entity names provide a consistent way for users to refer to network resources and services, such as a file server. Second, using names instead of addresses to refer to resources and services preserves the transparency of lower-layer operations to end-users.

Name Binding

Name binding is the process of mapping NVE entity names with network addresses. Each AppleTalk node maps the names of its own NVEs to their network addresses in a names table. The combination of all the names tables in all internetwork nodes is known as the *names directory*, which is a distributed database of all name-to-address mappings. Name binding can occur when a node is first started up or dynamically, immediately before the named entity is accessed.

NBP performs the following four functions: name lookup, name recognition, name confirmation, and name deletion. *Name lookup* is used to learn the network address of an NVE before the services in that NVE are accessed. NBP checks the names directory for the name-to-address mapping. *Name registration* allows a node to create its names table. NBP confirms that the name is not in use and then adds the name-to-address mappings to the table. *Name confirmation* is used to verify that a mapping learned by using a name lookup is still accurate. *Name deletion* is used to remove an entry from the names table in such instances as when the node is powered off.

AppleTalk Update-Based Routing Protocol (AURP)

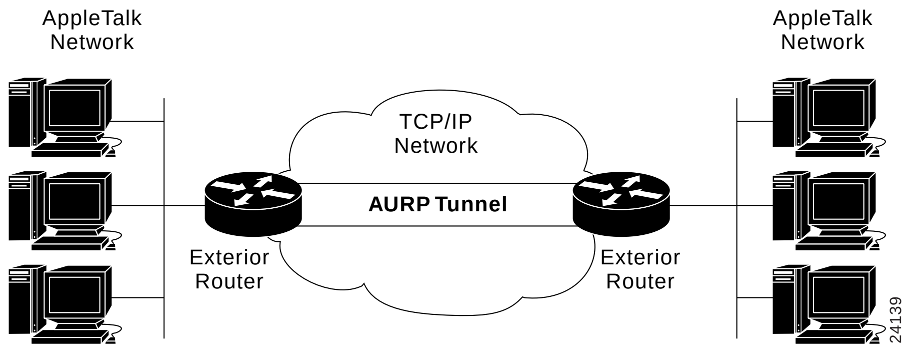
The *AppleTalk Update-Based Routing Protocol* (AURP) is a transport-layer protocol in the AppleTalk protocol suite that allows two or more AppleTalk internetworks to be interconnected through a *Transmission-Control Protocol/Internet Protocol* (TCP/IP) network to form an AppleTalk WAN. AURP encapsulates packets in User-Datagram Protocol (UDP) headers, allowing them to be transported transparently through a TCP/IP network. An AURP implementation has two components: exterior routers and AURP tunnels.

Exterior routers connect a local AppleTalk internetwork to an AURP tunnel. Exterior routers convert AppleTalk data and routing information to AURP and perform encapsulation and de-encapsulation of AppleTalk traffic. An exterior router functions as an AppleTalk router in the local network and as an end node in the TCP/IP network. When exterior routers first attach to an AURP tunnel, they exchange routing information with other exterior routers. Thereafter, exterior routers send routing information only under the following circumstances:

- When a network is added to or removed from the routing table
- When the distance to a network is changed
- When a change in the path to a network causes the exterior router to access that network through its local internetwork rather than through the tunnel, or through the tunnel rather than through the local internetwork

An AURP tunnel functions as a single, virtual data link between remote AppleTalk internetworks. Any number of physical nodes can exist in the path between exterior routers, but these nodes are transparent to the AppleTalk networks. Two kinds of AURP tunnels exist: *point-to-point tunnels* and *multipoint tunnels*. A point-to-point AURP tunnel connects only two exterior routers. A multipoint AURP tunnel connects three or more exterior routers. Two kinds of multipoint tunnels also exist. A fully connected multipoint tunnel enables all connected exterior routers to send packets to one another. With a partially connected multipoint tunnel, one or more exterior routers are aware only of some, not all, the other exterior routers. Figure 27-9 illustrates two AppleTalk LANs connected via a point-to-point AURP tunnel.

Figure 27-9 An AURP tunnel acts as a virtual link between remote networks.



AURP Encapsulation

When exchanging routing information or data through an AURP tunnel, AppleTalk packets must be converted from RTMP, ZIP, and (in the Cisco implementation) Enhanced IGRP to AURP. The packets then are encapsulated in User-Datagram Protocol (UDP) headers for transport across the TCP/IP network. The conversion and encapsulation are performed by exterior routers, which receive AppleTalk routing information or data packets that must be sent to a remote AppleTalk internetwork. The exterior router converts the packets to AURP packets, and these packets then are encapsulated in UDP headers and sent into the tunnel (that is, the TCP/IP network).

The TCP/IP network treats the packets as normal UDP traffic. The remote exterior router receives the UDP packets and removes the UDP header information. The AURP packets then are converted back into their original format, whether as routing information or data packets. If the AppleTalk packets contain routing information, the receiving exterior router updates its routing tables accordingly. If the packets contain data destined for an AppleTalk node on the local network, the traffic is sent out the appropriate interface.

AppleTalk Transaction Protocol (ATP)

The *AppleTalk Transaction Protocol* (ATP) is a transport-layer protocol in the AppleTalk protocol suite that handles transactions between two AppleTalk sockets. A transaction consists of transaction requests and transaction responses, which are exchanged by the involved socket clients.

The requesting socket client sends a transaction request asking that the receiving client perform some action. Upon receiving the request, the client performs the requested action and returns the appropriate information in a transaction response. In transmitting transaction requests and responses, ATP performs most of the important transport-layer functions, including acknowledgment and retransmission, packet sequencing, and segmentation and reassembly.

Several session-layer protocols run over ATP, including the *AppleTalk Session Protocol* (ASP) and the *Printer-Access Protocol* (PAP). These two upper-layer AppleTalk protocols are discussed later in this chapter.

Responding devices behave differently depending on which of two types of transaction services is being used: *At-Least-Once* (ALO) or *Exactly-Once* (XO) transactions. ALO transactions are used when repetition of the transaction request is the same as executing it once. If a transaction response is lost, the source will retransmit its request. This does not adversely affect protocol operations, because repetition of the request is the same as executing it once. XO transactions are used when repetition of the transaction request might adversely affect protocol operations. Receiving devices keep a list of every recently received transaction so that duplicate requests are not executed more than once.

AppleTalk Echo Protocol (AEP)

The *AppleTalk Echo Protocol* (AEP) is a transport-layer protocol in the AppleTalk protocol suite that generates packets that test the reachability of network nodes. AEP can be implemented in any AppleTalk node and has the statically assigned socket number 4 (the Echoer socket).

To test the reachability of a given node, an AEP request packet is passed to the DDP at the source. DDP addresses the packet appropriately, indicating in the Type field that the packet is an AEP request. When the packet is received by the destination, DDP examines the Type field and sees that it is an AEP request. In this process, the packet is copied, changed to an AEP reply (by changing a field in the AEP packet), and returned to the source node.

AppleTalk Upper-Layer Protocols

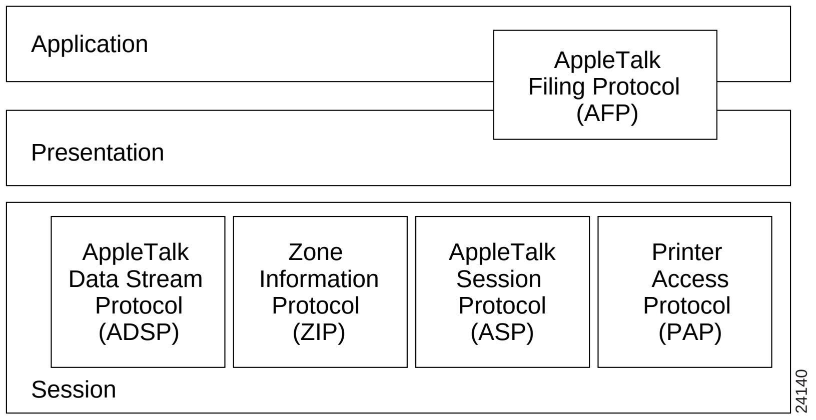
AppleTalk implements services at the session, presentation, and application layers of the OSI model. Four key implementations at the session layer are included in the AppleTalk protocol suite. (The session layer establishes, manages, and terminates communication sessions between presentation-layer entities).

Communication sessions consist of service requests and service responses that occur between applications located in different network devices. These requests and responses are coordinated by protocols implemented at the session layer.

The session-layer protocol implementations supported by AppleTalk include the *AppleTalk Data-Stream Protocol (ADSP)*, *Zone-Information Protocol (ZIP)*, *AppleTalk Session Protocol (ASP)*, and *Printer-Access Protocol (PAP)*.

The *AppleTalk Filing Protocol (AFP)* is implemented at the presentation and application layers of the AppleTalk protocol suite. In general, the presentation layer provides a variety of coding and conversion functions that are applied to application-layer data. The application layer interacts with software applications (which are outside the scope of the OSI model) that implement a communicating component. Application-layer functions typically include identifying communication partners, determining resource availability, and synchronizing communication. Figure 27-10 illustrates how the upper layers of the AppleTalk protocol suite map to the OSI model.

Figure 27-10 AppleTalk upper layer protocols map to three layers of the OSI model.


Application
AppleTalk Filing Protocol (AFP)
Presentation
AppleTalk Data Stream Protocol (ADSP)
Zone Information Protocol (ZIP)
AppleTalk Session Protocol (ASP)
Printer Access Protocol (PAP)
Session
24140

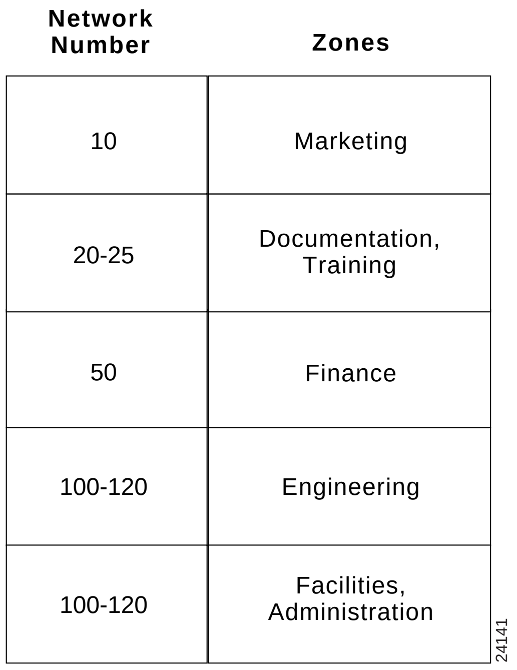
AppleTalk Data-Stream Protocol (ADSP)

The *AppleTalk Data-Stream Protocol (ADSP)* is a session-layer protocol in the AppleTalk protocol suite that establishes and maintains full-duplex communication between two AppleTalk sockets. ADSP guarantees that data is correctly sequenced and that packets are not duplicated. ADSP also implements a flow-control mechanism that allows a destination to slow source transmissions by reducing the size of the advertised receive window. ADSP runs directly on top of the DDP.

Zone-Information Protocol (ZIP)

The Zone-Information Protocol (ZIP) is a session-layer protocol in the AppleTalk protocol suite that maintains network number-to-zone name mappings in AppleTalk routers. ZIP is used primarily by AppleTalk routers. Other network nodes, however, use ZIP services at startup to choose their zone. ZIP maintains a *zone-information table* (ZIT) in each router. ZITs are lists maintained by ZIP that map specific network numbers to one or more zone names. Each ZIT contains a network number-to-zone name mapping for every network in the internetwork. Figure 27-11 illustrates a basic ZIT.

Figure 27-11 Zone-information tables assist in zone identification.



Network Number	Zones
10	Marketing
20-25	Documentation, Training
50	Finance
100-120	Engineering
100-120	Facilities, Administration

AppleTalk Session Protocol (ASP)

The *AppleTalk Session Protocol (ASP)* is a session-layer protocol in the AppleTalk protocol suite that establishes and maintains sessions between AppleTalk clients and servers. ASP allows a client to establish a session with a server and to send commands to that server. Multiple client sessions to a single server can be maintained simultaneously. ASP uses many of the services provided by lower-layer protocols, such as ATP and NBP.

Printer-Access Protocol (PAP) Overview

The *Printer-Access Protocol (PAP)* is a session-layer protocol in the AppleTalk protocol suite that allows client workstations to establish connections with servers, particularly printers. A session between a client workstation and a server is initiated when the workstation requests a session with a particular server. PAP uses the NBP to learn the network address of the requested server and then opens a connection between the client and server. Data is exchanged between client and server using the ATP. When the communication is complete, PAP terminates the connection. Servers implementing PAP can support multiple simultaneous connections with clients. This allows a print server, for example, to process jobs from several different workstations at the same time.

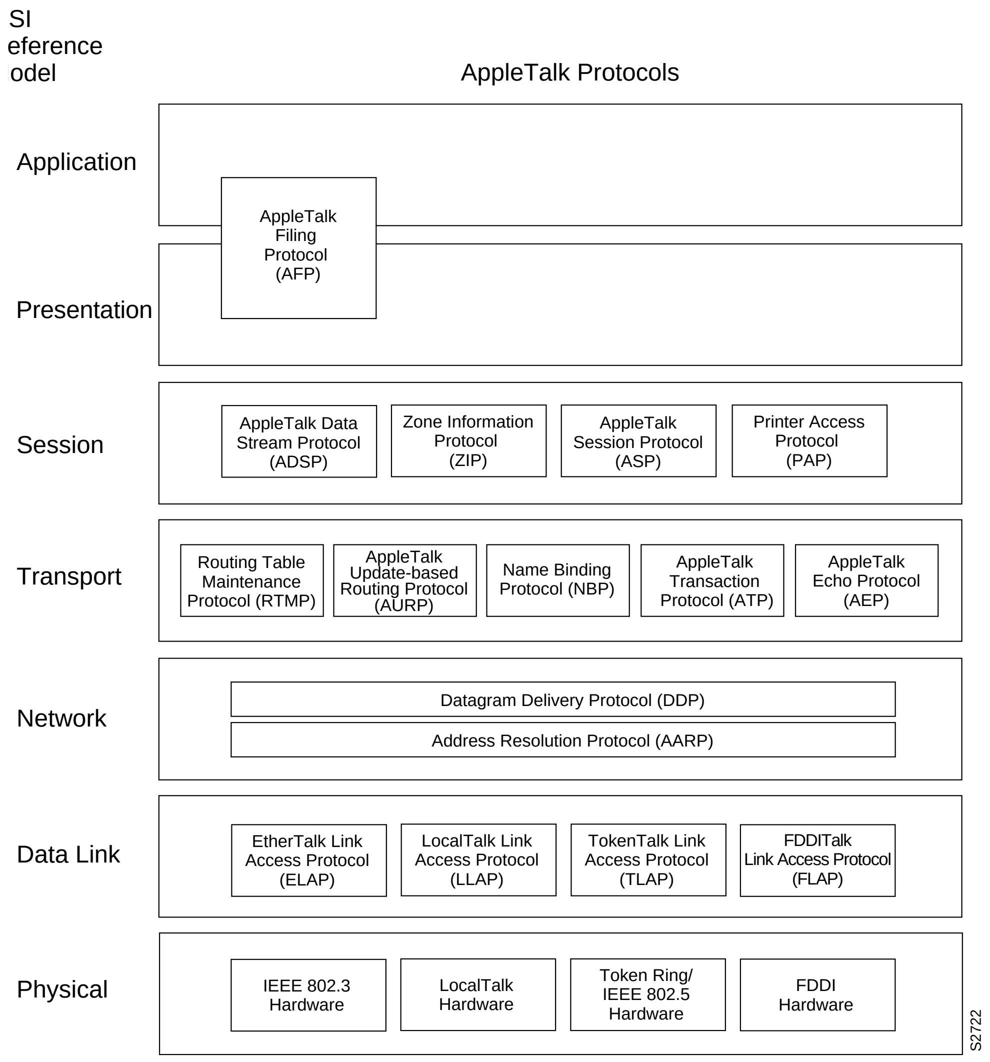
AppleTalk Filing Protocol (AFP)

The *AppleTalk Filing Protocol (AFP)* permits AppleTalk workstations to share files across a network. AFP performs functions at the presentation and application layers of the AppleTalk protocol suite. This protocol preserves the transparency of the network by allowing users to manipulate remotely stored files in exactly the same manner as locally stored files. AFP uses the services provided by the ASP, the ATP, and the AEP.

AppleTalk Protocol Suite

Figure 27-12 illustrates the entire AppleTalk protocol suite and how it maps to the OSI reference model.

Figure 27-12 The AppleTalk protocol suite maps to every layer of the OSI model.


SI
eference
odel
AppleTalk Protocols
Application
Presentation
AppleTalk Filing Protocol (AFP)
Session
AppleTalk Data Stream Protocol (ADSP)
Zone Information Protocol (ZIP)
AppleTalk Session Protocol (ASP)
Printer Access Protocol (PAP)
Transport
Routing Table Maintenance Protocol (RTMP)
AppleTalk Update-based Routing Protocol (AURP)
Name Binding Protocol (NBP)
AppleTalk Transaction Protocol (ATP)
AppleTalk Echo Protocol (AEP)
Network
Datagram Delivery Protocol (DDP)
Address Resolution Protocol (AARP)
Data Link
EtherTalk Link Access Protocol (ELAP)
LocalTalk Link Access Protocol (LLAP)
TokenTalk Link Access Protocol (TLAP)
FDDITalk Link Access Protocol (FLAP)
Physical
IEEE 802.3 Hardware
LocalTalk Hardware
Token Ring/ IEEE 802.5 Hardware
FDDI Hardware
S2722

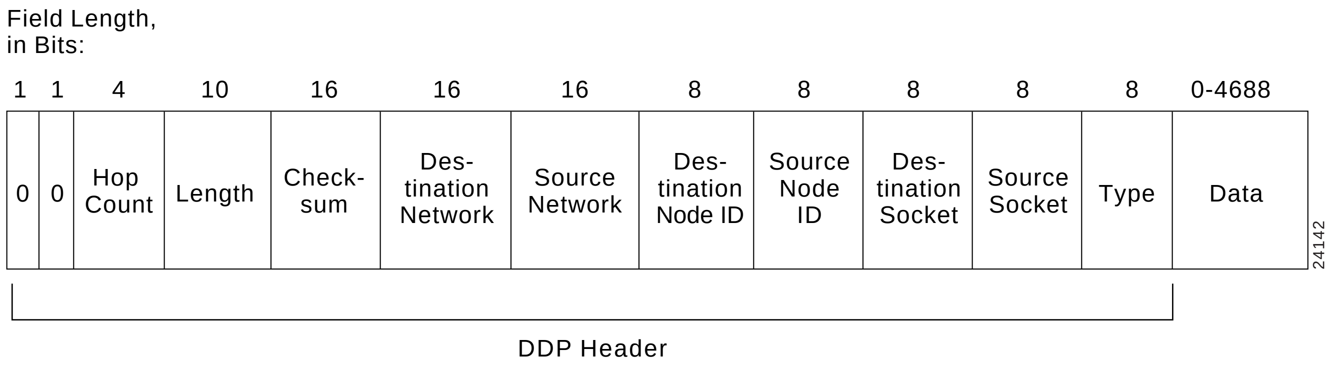
DDP Packet Format

The following descriptions summarize the fields associated with the DDP packets. This packet has two forms:

- *Short DDP Packet*—The short format is used for transmissions between two nodes on the same network segment in a nonextended network only. This format is seldom used in new networks.
- *Extended DDP Packet*—The extended format is used for transmissions between nodes with different network numbers (in a nonextended network), and for any transmissions in an extended network.

Figure 27-13 illustrates the format of the extended DDP packet.

Figure 27-13 An extended DDP packet consists of 13 fields.



The extended DDP packet fields illustrated in Figure 27-13 are summarized in the discussion the follows:

- *Hop Count*—Counts the number of intermediate devices through which the packet has passed. At the source, this field is set to zero. Each intermediate node through which the packet passes increases the value of this field by one. The maximum number of hops is 15.
- *Length*—Indicates the total length, in bytes, of the DDP packet.
- *Checksum*—Contains a checksum value used to detect errors. If no checksum is performed, the bits in this optional field are set to zero.
- *Destination Network*—Indicates the 16-bit destination network number.
- *Source Network*—Indicates the 16-bit source network number.
- *Destination Node ID*—Indicates the 8-bit destination node ID.
- *Source Node ID*—Indicates the 8-bit source node ID.
- *Destination Socket*—Indicates the 8-bit destination socket number.
- *Source Socket*—Indicates the 8-bit source socket number.
- *Type*—Indicates the upper-layer protocol to which the information in the data field belongs.
- *Data*—Contains data from an upper-layer protocol.