AppleTalk Protocols Communications services

The **AppleTalk Manager** provides a variety of services that allow Macintosh programs to interact with programs in devices connected to an AppleTalk network. This interaction, achieved through the exchange of variable-length blocks of data (known as packets) over AppleTalk, follows well-defined sets of rules known as protocols.

Although most programmers using AppleTalk need not understand the details of these protocols, they should understand the information in this section--what the services provided by the different protocols are, and how the protocols are interrelated. Detailed information about AppleTalk protocols is available in *Inside AppleTalk.*

The AppleTalk system architecture consists of a number of protocols arranged in layers. Each protocol in a specific layer provides services to higher-level layers (know as the protocol's clients) by building on the services provided by lower-level layers. A Macintosh program can use services provided by any of the layers in order to construct more sophisticated or more specialized services.

The **AppleTalk Manager** contains the following protocols:

- AppleTalk Link Access Protocol
- Datagram Delivery Protocol
- Routing Table Maintenance Protocol
- Name-Binding Protocol
- AppleTalk Transaction Protocol

The AppleTalk Link Access Protocol (ALAP) provides the lowest-level services of the AppleTalk system. Its main function is to control access to the AppleTalk network among various competing devices. Each device connected to an AppleTalk network, known as a node, is assigned an eight-bit node ID number that identifies the node. ALAP ensures that each node on an AppleTalk network has a unique node ID, assigned dynamically when the node is started up. ALAP provides its clients with node-to-node delivery of data frames on a single AppleTalk network. An ALAP frame is a variable-length packet of data preceded and followed by control information referred to as the ALAP frame header and frame trailer, respectively. The ALAP frame header includes the node ID's of the frame's destination and source nodes. The AppleTalk hardware uses the destination node ID to deliver the frame. The frame's source node ID allows a program in the receiving node to determine the identity of the source. A sending node can ask ALAP to send a frame to all nodes on the AppleTalk network; this broadcast service is obtained by specifying a destination node ID of 255.

ALAP can have multiple clients in a single node. When a frame arrives at a node, ALAP determines which client it should be delivered to by reading the frame's ALAP protocol type. The ALAP protocol type is an eight-bit quantity, contained in the frame's header, that identifies the ALAP client to whom the frame will be sent. ALAP calls the client's protocol handler, which is a software process in the node that reads in and then services the frames. The protocol handlers for a node are listed in a protocol handler table.

An ALAP frame trailer contains a 16-bit frame check sequence generated by the AppleTalk hardware. The receiving node uses the frame check sequence to detect transmission errors, and discards frames with errors. In effect, a frame with an error is "lost" in the AppleTalk network, because ALAP does not attempt to recover from errors by requesting the sending node to retransmit such frames. Thus ALAP is said to make a "best effort" to deliver frames, without any guarantee of delivery.

An ALAP frame can contain up to 600 bytes of client data. The first two bytes must be an integer equal to the length of the client data (including the length bytes themselves).

Datagram Delivery Protocol (DDP) provides the next-higher level protocol in the AppleTalk architecture, managing socket-to-socket delivery of datagrams over AppleTalk internets. DDP is an ALAP client, and uses the node-to-node delivery service provided by ALAP to send and receive datagrams. Datagrams are packets of data transmitted by DDP. A DDP datagram can contain up to 586 bytes of client data. Sockets are logical entities within the nodes of a network; each socket within a given node has a unique eight-bit socket number.

On a single AppleTalk network, a socket is uniquely identified by its AppleTalk address--its socket number together with its node ID. To identify a socket in the scope of an AppleTalk internet, the socket's AppleTalk Address and network number are needed. Internets are formed by interconnecting AppleTalk networks via intelligent nodes called bridges. A network number is a 16-bit number that uniquely identifies a network in an internet. A socket's AppleTalk address together with its network number provide an internet-wide unique socket identifier called an internet address.

Sockets are owned by socket clients, which typically are software processes in the node. Socket clients include code called the socket listener, which receives and services datagrams addressed to that socket. Socket clients must open a socket before datagrams can be sent or received through it. Each node contains a socket table that lists the listener for each open socket.

A datagram is sent from its source socket through a series of AppleTalk networks, being passed on from bridge to bridge, until it reaches its destination network. The ALAP in the destination network then delivers the datagram to the node containing the destination socket. Within that node the datagram is received by ALAP calling the DDP protocol handler, and by the DDP protocol handler in turn calling the destination socket listener, which for most applications will be a higher-level protocol such as the AppleTalk Transaction Protocol. You cannot send a datagram between two sockets in the same node.

Bridges on AppleTalk internets use the Routing Table Maintenance Protocol to maintain routing tables for routing datagrams through the internet. In addition, nonbridge nodes use RTMP to determine the number of the network to which they're connected and the node ID of one bridge on their network. The RTMP code in nonbridge nodes contains only a subset of RTMP (the RTMP stub), and is a DDP client owning socket number 1 (the RTMP socket).

Socket clients are also known as network-visible entities, because they're the primary accessible entities on an internet. Network-visible entities can choose to identify themselves by an entity name, an identifier of the form:

object:type@zone

Each of the three fields of this name is an alphanumeric string of up to 32 characters. The object and type fields are arbitrary identifiers assigned by a socket client to provide itself with a name and type descriptor (for example, abs:Mailbox). The zone field identifies the zone in which the socket client is located; a zone is an arbitrary subset of AppleTalk networks in an internet. A socket client can identify itself by as many different names as it chooses. These aliases are all treated as independent identifiers for the same socket client.

The Name-Binding Protocol (NBP) maintains a names table in each node that contains the name and internet address of each entry in *that* node. These name-address pairs are called NBP tuples. The collection of names tables in an internet is known as the names directory.

NBP allows its clients to add or delete their name-address tuples from the node's names table. It also allows its clients to obtain the internet addresses of entities from their names. This latter operation, known as name lookup (in the names directory), requires that NBP install itself as a DDP client and broadcast special name-lookup packets to the nodes in a specified zone. These datagrams are sent by NBP to the names information socket--socket number 2 in every node using NBP.

NBP clients can use special meta-characters in place of one or more of the three fields of the name of an entity it wishes to look up. The character "=" in the object or type field signifies "all possible values". The zone field can be replaced by "*", which signifies "this zone"--the zone in which the NBP client's node is located. For example, an NBP client performing a lookup with the name:

=:Mailbox@*

will obtain in return the entity names and internet addresses of all mailboxes in the client's zone (excluding the client's own names and addresses). The client can specify whether one or all of the matching names should be returned.

There is also a metacharacter in AppleTalk phase 2: the "≈" character matches a partial name; for example, matching "Mailbox≈" would match "Mailbox", "Mailbox37" and "Mailboxing".

NBP clients specify how thorough a name lookup should be by providing NBP with the number of times (retry count) that NBP should broadcast the lookup packets and the time interval (retry interval) between these retries.

As noted above, ALAP and DDP provide "best effort" delivery services with no recovery mechanism when packets are lost or discarded because of errors. Although from many situations such a service suffices, the AppleTalk Transaction Protocol (ATP) provides a reliable loss-free transport service. ATP uses transactions, consisting of a transaction request and a transaction response, to deliver data reliably. Each transaction is assigned a 16-bit transaction ID number to distinguish it from other transactions. A transaction request is retransmitted by ATP until a complete response has been received, thus allowing for recovery from packet-loss situations. The retry interval and retry count are specified by the ATP client sending the request.

Although transaction requests must be contained in a single datagram, transaction responses can consist of as many as eight datagrams. Each datagram in a response is assigned a sequence number from 0 to 7, to indicate its ordering within the response.

ATP is a DDP client, and uses the services provided by DDP to transmit requests and responses. ATP supports both at-least-once and exactly-once transactions. Four of the bytes in an ATP header, called the user bytes, are provided for use by ATP's clients--they're ignored by ATP.