**About the AppleTalk Manager-1** Introduction to communications

The **AppleTalk Manager** is divided into three parts.

- A lower-level driver called ".MPP" that contains code to implement ALAP, DDP, NBP and the RTMP stub; this includes separate code resources loaded in when an NBP name is registered or looked up.

- A higher-level driver called ".ATP" that implements ATP.

- A high-level language interface to these two drivers, which is a set of high-level language data types and routines to aid high-level language programmers in calling the **AppleTalk Manager**.

The two drivers and the interface to them are not in ROM; your application must link to the appropriate object files.

Pascal and C programmers make calls to the **AppleTalk Manager**'s high-level language interface, which in turn makes **Device Manager Control** calls to the two drivers. Assembly-language programmers make **Control** calls directly to the drivers.

> **Note**: High-level language programmers can, of course make **PBControl** calls directly if they wish.

The **AppleTalk Manager** provides ALAP routines that allow a program to:

- send a frame to another node

- receive a frame from another node

- add a protocol handler to the protocol handler table

- remove a protocol handler from the protocol handler table

Each node may have up to four protocol handlers in its protocol handler table, two of which are currently used by DDP.

By calling DDP, socket clients can:

- send a datagram via a socket

- receive a datagram via a socket

- open a socket and add a socket listener to the socket table

- close a socket and remove a socket listener from the socket table

Each node may have up to 12 open sockets in its socket table

Programs cannot access RTMP directly via the **AppleTalk Manager**; RTMP exists solely for the purpose of providing DDP with routing information.

The NBP code allows a socket client to:

- register the name and socket number of an entity in the node's names table

- determine the address (and confirm the existence) of an entity
- delete the name of an entity from the node's names table

The **AppleTalk Manager**'s .ATP driver allows a socket client to do the following:

- open a responding socket to receive requests
- send a request to another socket and get back a response
- receive a request via a responding socket
- send a response via a responding socket
- close a responding socket

> **Note:**Although the **AppleTalk Manager** provides four different protocols for your use, you're not bound to use all of them. In fact, most programmers will use only the NBP and ATP protocols.

AppleTalk communicates via channel B of the Serial Communications Controller (SCC). When the Macintosh is started up with a disk containing the AppleTalk code, the status of the network connection is checked. If the network connection is not being used by another device driver, and is available for use by AppleTalk, the .MPP driver is loaded into the system heap. On a Macintosh 128K, only the MPP code is loaded at system startup; the .ATP driver and NBP code are read into the application heap when the appropriate commands are issued. On any Macintosh later than the 128K, all AppleTalk code is loaded into the system heap at system startup.

After loading the AppleTalk code, the .MPP driver installs its own interrupt handlers, installs a task into the vertical retrace queue, and prepares the SCC for use. It then chooses a node ID for the Macintosh and confirms that the node ID is not already being used by another node on the network.

> **Warning:** For this reason it's imperative that the Macintosh be connected to the AppleTalk network *before* being switched on.

The **AppleTalk Manager** also provides high-level language routines for opening and closing the .MPP and . ATP drivers. The open calls allow a program to load AppleTalk code at times other than system startup. The close calls allow a program to remove the AppleTalk code from the Macintosh; the use of close calls is high discouraged, since other co-resident programs are then "disconnected" from AppleTalk. Both sets of calls are described in detail under **Calling the AppleTalk Manager**.

> **Warning:** If, at system startup, the network connection is not available for use by AppleTalk, the .MPP driver will not open. However, a driver does not return an error message when it fails to open. High-level language programmers must ensure the proper opening of AppleTalk by calling one of the two routines for opening the AppleTalk drivers (either **MPPOpen** or **ATPLoad**). If AppleTalk was successfully loaded at system startup, these calls will have no effect; otherwise they'll check the availability of the network connection, attempt to load the AppleTalk code, and return an appropriate result code.

**Assembly-language note:** Assembly-language programmers can use the high-level language routines for opening AppleTalk. They can also check the availability of the network connection themselves and then decide whether to open MPP or ATP.