Tech Info Library

Open Transport and Compatibility Mode (9/95)

Revised: 1/4/96
Security: Everyone

Open Transport and Compatibility Mode (9/95)

==

Article Created: 20 June 1995
Article Reviewed/Updated: 19 September 1995

TOPIC ---

This article describes how the new Open Transport (OT) networking architecture maintains compatibility with older networking schemes.

DISCUSSION --

In addition to its standards-based APIs (XTI, TPI, and DLPI), Open Transport includes support for many of the existing ("classic") networking and communications APIs to provide backwards-compatibility for existing AppleTalk and MacTCP applications, and for select other networking software.

AppleTalk backwards compatibility

AppleTalk backwards compatibility is accomplished by trapping all networking calls at the ".ddp" driver level. Above the DDP protocol, applications written to the classic AppleTalk APIs are actually running the classic implementations of AppleTalk. When the packets reach the DDP protocol layer, calls to the .ddp driver are intercepted, translated to the corresponding XTI calls, and are passed to Open Transport for processing. The process is reversed for incoming packets (OT receives the packet, native DDP handles the packet, the data is repackaged into an AppleTalk parameter block API, and control is passed back to the upper layer as if it had come from the .ddp driver).

The primary advantage of this approach is that backwards compatibility is very robust. Since the classic implementations of ADSP, ASP, ATP, NBP, ZIP, and so on are all used, we didn't have to attempt a "warts and all" emulation layer. It also turns out that this decreases our overall footprint, as compared to an implementation strategy where we would have tried to trap all packets at the top of the protocol and remap to their OT native counterparts -- that is, there is only the one DDP emulation shim, rather than an ADSP shim, an ASP shim, an ATP shim, a ZIP shim, an NBP shim, a PAP shim, and so on.

The primary disadvantage of this approach is that backwards compatibility, especially on the Power Macintosh, does not gain any meaningful performance increase for applications running in backward compatibility. Afterall, most of the stack is still running in emulation - only DDP is actually native. And add on top of that the increased cost of the mixed mode context switch from 680x0 to native (or native to 680x0) for each packet handled.

TCP/IP backwards compatibility

TCP/IP backwards compatibility is accomplished by trapping all networking calls at the ".ipp" driver level. Since MacTCP was implemented essentially as a single monolithic driver, this means that there was only one place where networking calls could be intercepted and emulated. Calls to .ipp are trapped, mapped to appropriate XTI calls, and the OT/TCP stack handles the rest. The process is reversed on incoming packets.

The primary advantage of this approach (not that there really was any technical alternative) is performance. The entire handling of the packet happens in the new OT implementation. Especially on Power Macintosh, that means networking goes native as soon as the application makes the call to the driver (well, technically as soon as the emulation shim maps the driver call to the XTI call. The emulation shim still runs as 680x0 code).

The primary disadvantage of this approach is that backward compatibility is potentially less robust. It is a completely new code base for TCP and developers may be depending on idiosyncrasies of MacTCP -- if so, they will probably break.

Other networking compatibility

- DLPI to .enet conversion. Current NuBus Macintosh computers (680x0 and DLPI) have ethernet implementations that rely on the '.enet' driver architecture and API specification. Open Transport, however, expects to communicate only with DLPI drivers. To allow OT to run on NuBus Macintosh computers without requiring the creation of new drivers (that is, to allow existing cards and drivers to be used with OT), we've created an adapter that accepts DLPI calls (from the 'top) and converts them to .enet calls (out the bottom). This is believed to be quite robust (or will be when the next version of OT ships).

- .enet to DLPI conversion. Certain other networking products, including protocols like MacIPX and PATHWORKS DECnet, and certain applications software products, such as Insignia SoftWindows, have been written to bypass AppleTalk and MacTCP, and talk directly to the .enet driver. This is a problem, because early PCI-based Power Macintosh computers will have ONLY DLPI drivers for networking -- no .enet will exist on the Power Macintosh 9500 with the first release of Open Transport.

Later versions of Open Transport will allow these protocols and applications to run on early PCI-based Power Macintosh computers, a special adapter has been created that accepts .enet calls (from the 'top') and converts them to DLPI calls (out the bottom). This software is designed as a stop-gap compatibility measure until new versions of such software can be created for early PCI-based Power Macintosh computers. Because of this, it has two key limitations: it only supports the built-in ethernet adapter (which means that you can't run the current versions of SoftWindows or MacIPX or PATHWORKS on a PCI based networking card); and it does not support promiscuous mode (which means that you can't use an early PCI-based Power Macintosh with the current version of NetMinder or EtherPeek).

There are no other API changes or backward compatibility issues in OT v1.0.

Article Change History:
19 Sep 1995 - Changed access privileges to Everyone.
07 Sep 1995 - Made editing corrections.
14 Jul 1995 - Clarified version numbers mentioned.

Support Information Services

Tech Info Library Article Number:17989