



Tech Info Library

AppleTalk Phase 2: SNAP Encapsulation with OUI=0

This article last reviewed: 9 October 1990

TOPIC -----

My problem is that the AppleTalk Phase 2 ARP uses a SNAP encapsulation with an OUI=0 ("Organizational Unique Identifier"). Because Translating FDDI/Ethernet bridges use this OUI to pass Ethernet 2 packets to FDDI and back, an AppleTalk ARP gets converted to an Ethernet 2 packet when passed from FDDI to Ethernet. This packet format is not accepted by Apple host implementations. Therefore, it's not possible to use AppleTalk Phase 2 with FDDI with translating bridges between two AppleTalk hosts.

We need a solution.

DISCUSSION -----

The all-zeros "Organizational Unique Identifier" (OUI) was not specifically intended to be used for translating Ethernet frames to 802.2/SNAP. Its original intent was (and is) not completely clear; all that was generally agreed upon was:

- 1) The low-order 16 bits of the SNAP protocol discriminator indicated an Ethernet protocol type.
- 2) The format of the data part of the SNAP packet was the same as the format of the data part of the Ethernet packet of this protocol type.

This "general agreement," as far as we know, was never written down. AppleTalk Phase 2 and TCP/IP both adopted this convention. The unfortunate difference is that, on an Ethernet/802.3 data link, AppleTalk Phase 2 runs on top of 802.3 (and 802.2) and TCP/IP runs on top of Ethernet. Thus, for a TCP/IP Ethernet node to communicate with a TCP/IP 802.2 node (on Token Ring or FDDI, for example), some type of packet translation must be done. On the other hand, for an AppleTalk Phase 2 Ethernet node to communicate with any other AppleTalk Phase 2 802.2 node (on Ethernet, Token Ring, or FDDI), no such translation is necessary (in fact, such translation MUST NOT occur).

This makes it somewhat difficult to build a bridge that "transparently" allows both forms of communication. However, all that really has to be done is for the bridge to contain a small table of Ethernet protocol types that either should, or shouldn't, be translated in this manner. Obviously, the smaller the

table, and the less translation done, the better the overall performance. Any bridge between Ethernet and an 802 media that does not translate any OUI zero packets will prevent TCP/IP nodes (and any others using the same OUI convention) from communicating between the two media. Any bridge between Ethernet and an 802 media that translates all OUI zero packets will prevent AppleTalk nodes (and any others with an OUI of zero that don't expect translation) from communicating between the two media. In both cases, routers can, of course, be set up to accomplish the communication.

A somewhat equivalent problem could exist if two bridges are used to connect two Ethernet/802.3 data links across an 802 backbone (say, FDDI). In this case, some type of translation/encapsulation of Ethernet (non-802.3) packets must be performed to allow them to be passed across the 802 backbone. There are many ways to do this, one of which seems to be translating all Ethernet packets to OUI zero packets, shipping them across the backbone, and then converting them back. This, however, does not work for any OUI zero packets that start out on the Ethernet/802.3 data link. Such packets will not be translated on the way into the backbone, but will be incorrectly translated before being delivered on the destination Ethernet. Such packets should be forwarded entirely without translation.

In terms of actual product implementations, unfortunately, DEC is already testing an Ethernet-to-FDDI bridge that translates all OUI zero packets and, thus, does not work with AppleTalk Phase 2. They are aware of this issue. At the recent 802 meetings in Denver, it became clear that this type of translation is not part of the 802.1d (Macintosh Bridging) standard, but is something that many bridge manufacturers want to include as a non-802-standard option with their products. Most have said that they will have some type of translation (or non-translation) table to ensure that it works with all known protocols.

Copyright 1990 Apple Computer, Inc.

Keywords: <None>

=====

This information is from the Apple Technical Information Library.

19960215 11:05:19.00

Tech Info Library Article Number: 6202