Tech Info Library

AppleTalk and IBM 8209 Bridges Don't Mix

Article Created: 23 January 1992
Article Last Reviewed:
Article Last Updated:

TOPIC ---

I have a lab with 24 Macintosh LCs and one Macintosh IIci on Ethernet. The file server is attached to an IBM 8209 bridge which is a hardware component that bridges Token Ring networks and Ethernet networks. Some of the Macintosh LC computers randomly won't show the file server in the Chooser, but the Macintosh IIci always shows the file server.

If we restart the IBM 8209 bridge, the Macintosh LCs show the file server again. All the workstations, Macintosh LCs, and Macintosh IIci run system software version 6.0.7. The file server on the other side of the 8209 bridge is a Macintosh IIci running System 7.0 and AppleShare 3.0. We tried System 7.0 on all the computers, and AppleShare 2.0 with system software version 6.0.7 on the server.

DISCUSSION --

Here is some information that should help you understand why the IBM 8209 bridge and AppleTalk aren't compatible.

IBM 8209 and AppleTalk

Ideally, with the 8209 providing Token Ring to Ethernet bridging services, Macintosh computers on both transports would see the bridged network as a single logical network. Unfortunately, for a number of reasons, this isn't true.

Phase 1 EtherTalk

The first problem stems from the 8209 bridge's generosity in forwarding all datalink (MAC) broadcasts and multicasts. When you run ANY Phase 1 EtherTalk, adding an 8209 bridge into the network automatically generates Phase 1 TokenTalk. This is useless traffic, because the TokenTalk drivers expect Phase 2 protocols. As far as we can tell, it isn't dangerous traffic; it's safely ignored by all nodes. But it does take up bandwidth, and if the Phase 1 Ethernet is large and has many routers, there can be significant broadcast traffic forwarded onto the Token Ring. The answer here is to filter Phase 1 EtherTalk at the 8209 bridge.

Address Bit-Flipping

Token Ring and Ethernet don't agree on the order of bits. So, the 8209 bridge flips the bits of the datalink (MAC) level addresses when it forwards the packets. Normally, this affects only the destination and source addresses in the packet header. However, address resolution protocols such as TCP/IP ARP and AppleTalk's AARP also carry MAC level addresses as a part of the data portion of the packet. The 8209 bridge is specially programmed to look for TCP's ARP, and to go down inside the data, flipping the bits of the embedded MAC address. Later, when a Token Ring host wants to talk with an Ethernet node, it uses the flipped version of the MAC address. When encountered by the 8209 bridge, the address is flipped back and thus delivered correctly.

The 8209 bridge ISN'T specially programmed to assist AARP. Thus, the AARP address is stored in the Token Ring AppleTalk node in its "unflipped" form. Then, should it be used later to address a packet, the 8209 bridge will flip the bits when forwarding back to Ethernet. Now the packet is incorrectly addressed, and can't be delivered.

In short, when using an IBM 8209 to bridge Token Ring and Ethernet, AppleTalk nodes on the ring and Ethernet can't communicate because they can never use AARP correctly.

Configuration Workarounds

Curiously, if the you use Token Ring only as a backbone, and a Macintosh on Ethernet wants to talk to a Macintosh on another Ethernet, with both Ethernets attached by 8209 bridges, this connection WILL work. In fact, as long as you go through an EVEN number of 8209 bridges, connection succeeds. When there's an odd number, connection fails. Note that this makes some extremely bizarre network configurations possible, such as a system of alternating Token Rings and Ethernets connected by 8209 bridges. In this case, all the Ethernets would have AppleTalk connectivity, and all the Token Rings would have AppleTalk connectivity, but the rings and Ethernets would have AppleTalk connectivity between them!

Address Ranges

You may remember that under AppleTalk Phase 2, we use multicast addressing for all protocol "overhead" packets, such as RTMPs, NBPs, and ZIPs. You may also recall that we have a range of 254 addresses used in Ethernet, and 19 different addresses in Token Ring available for multicasts.

Our use of different address ranges, in addition to the bit flipping described previously, ensures that AppleTalk nodes can't interoperate across an 8209 bridging Ethernet and Token Ring. Just look at the startup process.

A node begins by sending an AARP probe for a protocol address (multicast). The 8209 bridge forwards this AARP message, but because the bits are flipped, a response to the AARP probe gets lost on the return trip. So, the node assumes that its protocol address is unique, and then proceeds to check for a router by multicasting (protocol broadcasting) a ZIP GetNetInfo request. If there is an AppleTalk router present on the cable (whether on Ethernet or Token Ring), it will work. The router hears the request, provides the needed information about the cable, and sends another AARP if necessary.

However, if the router is on the other side of the 8209 bridge, the multicast is forwarded. Even if the bit-flipping wasn't a problem (that is, if IBM were to change the 8209 software to support AARP), you will still NOT MAKE CONNECTION with the router. This is because a TokenTalk router expects multicast addresses in a different range than an EtherTalk node can send, and vice versa. The only answer is to use an AppleTalk router, which manages protocol and zone multicast addresses for each data link appropriately.

In short, AppleTalk and 8209 bridges don't mix. We advise you NOT to use bridging strategies between AppleTalk Token Ring and Ethernet networks.

Keywords: <None>

===

This information is from the Apple Technical Information Library.

19960215 11:05:19.00

Tech Info Library Article Number: 9694