Internet Router and Novell Token Ring VAP Problem

Article Created: 15 February 1991

Article Change History

12/9/92 - RETITLED
- To more accurately describe the article.

TOPIC ---

I have been having problems with what looks like the Novell Token Ring VAP and the Apple Internet Router. The following is a description of the problem, the observations of the packet traffic by an engineer from Novell, and myself, with a shot at a conclusion. I know that the description of the packet traffic is meaningless without the actual trace data in front of you, but sending the trace data was impractical due to size (5MB). If you are interested in looking at the collected data, I can get it for you.

If there are known problems with the Internet Router that manifest themselves by the following description, I need to know as soon as possible.

Observations: The Router Problem

Our company is seeing a problem where our 286 server drops connections. Our printers fade in and out of the Chooser. Our mail servers appear and disappear. Our network is nearly unusable. In the office we visited, there is a main token ring with four routers. Two of those routers connect the main ring to smaller rings that contain users. One of those routers connects to a LocalTalk segment. The other router routes to an Ethernet that passes back to the company's enterprise-wide net, and contains some large servers.

In general, we are worried about users on the two smaller rings and the LocalTalk segment talking to servers on the main ring. There are no "users" on the main ring, just routers and servers. The two servers involved are called "who" and "who2". All of the routers involved are Apple Internet Routers, running on some fast II-class Macintoshes.

To look at this problem we attached two LANalyzers to the network, one (LA1) on one of the smaller rings, and one (LA2) on the main ring. On the

smaller ring, we caught all traffic in and out of the router. On the main ring we caught all broadcast traffic, all traffic in and out of the "who" server, and all traffic in and out of that router. We told engineers to halt the LANalyzers whenever they had a report of a dropped connection, and we would look at the problem in the morning.

The next morning they had very few reports of the network failing. They had two traces for us. One was a failure on the subring we were watching. Two were failures from the LocalTalk segment, whose router we were not specifically watching. In the case of the failure on the subring, we were watching, the traffic on the main ring was so high that we couldn't get enough information about the history before the crash.

We did see that the server (who2) had sent an ASP close connection request to the workstation, and that the workstation had received it. The server issues a close connection usually only when there aren't enough ASP tickle packets to convince the server that the connection is still up. But we could not look far enough back in time to see the cause of this close.

We looked at one of the LocalTalk failure traces, expecting to find little of interest, because we weren't specifically targeting that router. However, the LANalyzer on the main ring told an interesting story. What follows is my interpretation of the trace, showing the important points. There is probably other information to be gleaned from the trace.

Packet #186 the first traffic between WHO and the LocalTalk connected router about the ASP connection in question (called 17103_air) was a tickle packet. Instead of sending the tickle directly over the Ethernet, the packet was sent to a different router (let's call it MYSTERY) on the ring, one that we couldn't identify. Time 7:43:31.

Packet #36517103_air starts sending AARP request broadcasts, looking for the WHO node. It sends one every 200 millisec, and about 3 millisec later the WHO server responds. It seems that the AARP responses are well formed, but 17103_air continues to rebroadcast. This continues for approximately 15 seconds, and during that time no traffic between 17103_air and WHO takes place. We see some, but not much, traffic between 17103_air and the other router we were monitoring. Time 7:43:42.

Packet #64017103_air stops sending AARP requests to WHO, and starts sending packets. Several retries of an ASP Request are sent, all within a second, and the server responds back to 17103_air correctly in each case. The connection will continue correctly for a while, with 17103_air sending directly to WHO and vice versa. Time 7:43:57

Packet #95317103_air sends and ASP Req packet to WHO, and WHO sends the response to a new and different Token Ring address (MYSTERY2). We don't see this node forward the packet, but 17103_air sends the ASP release directly to WHO, so the connection is still up, and the packet did get to the end Macintosh. The connection continues, with WHO sending most of its packets on this connection (destined eventually for 17103_air) to MYSTERY2.

Packet #1161 WHO sends its tickle packet to MYSTERY2 instead of

17103_air.

Packet #1387 WHO sends an ASP Response to MYSTERY (remember him?).

Packet #1843 17103_air starts sending AARP request packets, looking for node WHO. Like in the sequence starting in packet 356, AARP requests are sent every 200 microsec, and WHO responds correctly to each and every request. 17103_air continues to request. We see no further packets from 17103_air to WHO. WHO continues to send tickle packets destined for the end workstation, but sends them to all kinds of routers (MYSTERY, MYSTERY2). Time 7:45:20.

Packet #4516 WHO sends a close connection packet to the workstation through 17102_air (the router we were watching in the first place). 17103_air is still AARPing. Time 7:47:09.

Packet #5043 17103_air stops AARPing. It does not send any further packets to WHO. The connection is dead. Time 7:47:26.

Questions and Problems

The first and most interesting question is why does the Apple Internet Router continue to AARP over and over again? Another question is, why did it stop AARPing in the first case, in so little time as to let the connection stay alive? Looking at the trace we see another router, 17102_air, AARP over and over again for the server WHO2.

This is obviously not an isolated problem, and it is the cause of the destruction of the connections. Possibly, it also backs up the router queues, and thus prevents NBP lookups, accounting for our NBP problems. The router statistics are showing a large number of "overflows" whatever that is.

The second question is why the 286 VAPs start sending to other routers besides the correct router (MYSTERY, MYSTERY2). Secondly, the VAPs do not seem to redirect back to the correct router after getting a packet from it. However, although "Inside AppleTalk" suggests the caching of routers and building up of a RTMP table via the RTMP stub, nothing particularly bad will happen except loss of bandwidth and overburdening of routers.

In conclusion, the company's problem is a sporadic loss of routing through our Apple Internet Routers, caused by an illegal activity of the AIRs (not accepting AARP responses correctly). The cause of this failure remains unknown.

DISCUSSION ----------

The central problem is that printers, file servers, and mail servers drop in and out of the Chooser. This brings up a couple of questions. How many of each type of device do you have in each zone? Could you do an Inter•Poll of the affected zones for each type of device in question and send us the information? Here is some information and a possible answer:

This is how the Chooser really works in a nutshell:

The Chooser sends out a Name Binding Protocol (NBP) packet looking for all devices of type XXXXXX (for example, type LaserWriter). It sets up a buffer of 512 bytes for the responses. The responses look like:

device name length	1 byte	
device name	variable bytes	e.g. MyLaser-Hands off
type name length	1 byte	
type name	variable bytes	e.g. LaserWriter
zone field length	1 byte	
zone name field	variable bytes	probably *

The Chooser gets such a packet back for each device, i.e., each LaserWriter. When the 512-byte buffer is full of these packets, it stops looking for device names to display. This means that some LaserWriters might not be displayed immediately. If you leave the Chooser window open, however, the Chooser continues to send out NBP lookups every 1.47 seconds. Different LaserWriters could respond more quickly each time. In this case, you may see the Chooser show and hide various devices.

This means that the number of devices the Chooser can show really depends on how long the type name (like "LaserWriter") is and how long the device names are.

The number 18 is an average number, based on device names being about 13- or 14 characters long and the device name being about 10 or 11 characters long.

In System 7.0, the buffer size for the Chooser is increased to 1024. This means, on an average, about 36 devices will be able to be displayed.

There is a way to affect the manner in which the lookup is done, which could help in some environments, especially in wide-area-network environments where slow data links may be used. If you modify the GNRL resource in the Chooser document (AppleShare, for example), it will affect every NBP lookup that is done from the Chooser for that type of device.

The Chooser uses these values to determine the NBP lookup interval and retry values for the current NBP transaction. The default of 0705 tells the Chooser to send five NBP lookup requests at an interval of 7/8ths of a second. This process is repeated in an infinite loop, until the user closes the Chooser.

Chooser Event Flow Example:

```
User opens Chooser and selects the AppleShare CDEV
GNRL resource -4096 loaded value = (5002)
NBP lookup mechanism started

NBP Loop:
    Get NBP ID for this transaction
```

```
            (Note: All NBP request and replies for this loop will use
              this ID)

          Send first lookup (NBP ID = "New")
          Collect and display responses from the NBP lookup ID "New"
          Wait 10.6 seconds

          Send second lookup (NBP ID still = "New")
          Collect and display responses from the NBP lookup ID "New"
          Wait 10.6 seconds

          Discard all buffers and data associated with NBP ID "New"
            (Note: If a response is received for NBP lookup ID =
              "New" after this point the reply data would be discarded
              and the device would not be added to the list in the
              Chooser)
          Do some other misc. cleanup (approx. time 1 sec)

          goto NBP Loop

      End NBP Loop:
```

With the retry timer set to such a large value the multiple retry count is really not necessary. On the other hand, it doesn't hurt either, and it effectively increases the time we'll wait for NBP replies to over 20 seconds for the current transaction. The idea behind the retry count is to send several lookup requests out in quick succession (default < 1 sec.), in case there are devices which were unable to respond because they were busy or because the previous packet never reached them.

The reason that increasing the interval timer helps in the case of remote servers is directly tied to the way the NBP mechanism works. The Chooser maintains only 1 NBP lookup request at a time, tracking all replies to that request by way of the NBP ID mechanism. Replies that are received that do not match the current request ID are discarded.

The request ID is maintained only for the current NBP request, the interval and retry counters for this request can be tuned via the GNRL resource. In other words, if you set the retry counter to 10 and the interval timer to 50 the NBP ID would be maintained for 10 requests at an interval of 10.6 seconds. The GNRL resource is documented in "Inside Macintosh, Vol. 4", page 216.

AARP Issue

The symptoms you describe with the Apple Internet router AARPing for the same node over and over is probably attributed to the router being overloaded and not able to accept the response AARPs. You mention that the router is getting a fair amount of overflow errors, this would lead me to believe that the routers are indeed overloaded.

What ports on the routers are getting the overflow errors and how many are reported over the course of a day, a week? Overflow errors are caused by

the router being too busy to process all incoming packets on an interface. The network interface chip set can detect that a packet was available but that the processor was too busy to get the packet from the interface before the next packet arrived. There may be some other problems related to your environment, but this is a good starting point.

Novell VAPs Sending to Random Routers

AppleTalk Phase 2 does provide an enhancement that lets your node cache network to router pairs for use when determining where to route a packet destined for a remote network. When the AppleTalk protocol DDP receives a packet from a remote network, it strips off the data-link source address of the packet. This is the address of the last router on the route from the original network. This router should generally be the optimal router in terms of hop count back to the original network. You can then use that router for any future transactions to that network.

Now that I've explained all of that, you're saying, "Okay, that sounds right, but the Novell server is not doing what it's supposed to." The real story is that this enhancement is an optional, implementation-specific addition that is not required by the AppleTalk protocol. In this case, it would be normal for the Novell server to act the way that you described, if they did not implement the "Best Router" enhancement.

Conclusions

We first need to get a handle on the environment that you have, including numbers of devices per network and per zone. We need to take a close look at the statistics from the Apple Internet routers, average load on the various network segments as measured by the Internet Router, as measured by a network monitor/analyzer.

A close look at traffic patterns could also be helpful in determining where to best segment the network if it becomes necessary. It may be that the traffic on the main ring is so heavy that the Macintosh routers can't keep up. We don't really have any statistical, benchmark, or historic information that would tell us when the use of an AIR on a token ring is not going to offer the best performance. At this point, we just need to collect all the information and then take it one step at a time.

The LANalyzer traces wouldn't do us any good, because we have a Network General Sniffer. If we really need to see some trace data, we'll do some tests using a Sniffer on your end, or the LANalyzer data could be saved to an ASCII file and shipped to us on a tape. I don't think we need to worry about the trace data yet.

Keywords: <None>

==========

19960215 11:05:19.00

Tech Info Library Article Number: 7009