Network Working Group
Request For Comments: 1868
Category: Experimental

G. Malkin
Xylogics, Inc.
November 1995

ARP Extension - UNARP

Status of this Memo

This memo defines an Experimental Protocol for the Internet community. This memo does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Abstract

The Address Resolution Protocol allows an IP node to determine the hardware (datalink) address of a neighboring node on a broadcast network. The protocol depends on timers to age away old ARP entries. This document specifies a trivial modification to the ARP mechanism, not the packet format, which allows a node to announce that it is leaving the network and that all other nodes should modify their ARP tables accordingly.

Acknowledgements

Thanks to James Carlson/Xylogics for reviewing this document and proposing the backwards compatibility mechanism.

1. Introduction

The primary purpose of the Address Resolution Protocol, as defined in [1], is to determine a node's hardware address based on its network address (protocol address in ARPspeak). The ARP protocol specifically states that nodes should not periodically advertise their existence for two reasons: first, this would generate a lot of network traffic and table maintenance overhead; second, it is highly unlikely that all nodes will need to communicate to all other nodes. Since a node does not advertise its existence, neither does it advertise its imminent departure. This is not a serious problem since most ARP implementations maintain timers to age away old entries, and departing nodes seldom depart gracefully in any case.

Over time, an additional use has been found for ARP: Proxy ARP. While there are those who believe Proxy ARP is an evil thing, it does serve a purpose; that is, it allows for communication in ways never considered in the original IP architecture. For example, allows dial-in hosts to connect to a network without consuming a large amount of the IP address space (i.e., all of the hosts contain addresses on the same subnet, even though they are not directly attached to the physical network associated with that subnet address. It is this use of Proxy ARP which produces the problem addressed by this document.

2. The Problem

Consider the following topology:

```
                 +--------+
                 | Host A |
                 +--------+
                     |
========================================= LAN
    |                             |
+--------+                    +--------+
|  CS1   |   comm. servers    |  CS2   |
+--------+                    +--------+
  |    |                        |    |
 +-+  +-+                      +-+  +-+
 | |  | |       modems         | |  | |
 +-+  +-+                      +-+  +-+
```

Assume that all of the modems are on the same rotary; that is, when a remote host dials in, it may be assigned a modem on either of the communication servers. Further assume that all of the remote hosts' IP addresses have the same subnet address as the servers and Host A, this in order to conserve address space.

To begin, a remote host dials into CS1 and attempts to communicate with Host A. Host A will assume, based on the subnet mask, that the remote host is actually attached to the LAN and will issue an ARP Request to determine its hardware address. Naturally, the remote host will not hear this request. CS1, knowing this, will respond in the remote host's place with its own hardware address. Host A, on receiving the ARP Reply, will then communicate with the remote host, transparently through CS1. So far everything is just fine.

Now, the remote host disconnects and, before Host A can age its ARP cache, reconnects through CS2. Herein lies the problem. Whenever Host A attempts to send a packet to the remote host, it will send it to CS1 because it cannot know that its ARP cache entry is invalid. If, when the remote host disconnects, the server to which it was attached could inform other nodes on the LAN that the protocol address/hardware address mapping was no longer valid, the problem would not occur.

3. The Solution

When a server, as described above, disconnects from a remote host for which it has responded to a Proxy ARP, it broadcasts an UNARP. An UNARP is an unsolicited ARP Reply with the following field values:

Hardware Address Space	as appropriate
Protocol Address Space	0x800 (IP)
Hardware Address Length	0 (see Backwards Compatibility)
Protocol Address Length	4 (length of an IP address)
Opcode	2 (Reply)
Source Hardware Address	Not Included
Source Protocol Address	IP address of detaching host
Target Hardware Address	Not Included
Target Protocol Address	255.255.255.255 (IP broadcast)

NOTE: this is a 16-byte packet (not including MAC header)

On receiving an UNARP, a node deletes the ARP cache entry associated with the IP address.

It is not strictly necessary that a server keep state information about whether or not it has actually sent a Proxy ARP Reply; it would be sufficient if a server always sends an UNARP when a remote host disconnects.

Of course, there is no reason why a host which gracefully detaches from a LAN cannot also send an UNARP for itself. This would be especially useful if, upon re-attaching, it might have a different hardware address.

4. Backwards Compatibility

The modifications to support UNARP are trivial, so there is every expectation that it will be widely supported. Of course, there will be a period of time during which nodes which support UNARP will coexist with nodes which do not support UNARP. To prevent unenlightened nodes from adding spurious ARP cache entries with hardware addresses of zero, UNARP packets specify a hardware address length of zero. This should be rejected by nodes which do not support UNARP. As a consequence of this, the source and target hardware address fields do not exist in UNARP packets (as previously described).

It is recommended that implementors include a configuration switch to disable UNARP in the event that some vendor’s ARP implementation might take offense at the abbreviated UNARP packet format.

5. Security Considerations

Security issues are not discussed in this memo.

References

[1] Plummer, D., "An Ethernet Address Resolution Protocol", STD 37, RFC 826, MIT, November 1982.

Author's Address

Gary Scott Malkin
Xylogics, Inc.
53 Third Avenue
Burlington, MA 01803

Phone: (617) 272-8140
EMail: gmalkin@xylogics.com