Network Working Group
Request For Comments: 1931
Category: Informational

D. Brownell
Sun Microsystems, Inc.

April 1996

Dynamic RARP Extensions for Automatic Network Address Acquisition

Status of this Memo

This memo provides information for the Internet community. This memo does not define an Internet standard of any kind. Distribution of this memo is unlimited.

1. Introduction

This memo describes extensions to the Reverse Address Resolution Protocol (RARP [2]) and called Dynamic RARP (DRARP, pronounced D-RARP). The role of DRARP, and to some extent the configuration protocol used in conjunction with it, has subsequently been addressed by the DHCP protocol [9]. This memo is being published now to document this protocol for the record.

DRARP is used to acquire (or allocate) a protocol level address given the fixed hardware address for a host. Its clients are systems being installed or reconfigured, and its servers are integrated with other network administration services. The protocol, along with adjunct protocols as briefly described here, supports several common styles of "Intranet" administration including networks which choose not to support the simplified installation and reconfiguration features enabled by DRARP.

The rest of this introductory section summarizes the system design of which the DRARP protocol was a key part. The second section presents the DRARP protocol, and the third section discusses requirements noted for an "Address Authority" managing addresses in conjunction with one or more cooperating DRARP servers.

1.1 Automatic System Installation

Dynamic RARP was used by certain Sun Microsystems platforms beginning in 1988. (These platforms are no longer sold by Sun.) In conjunction with other administrative protocols, as summarized in the next subsection, it was part of a simplified network and domain administration framework for SunOS 4.0. Accordingly, there was a product requirement to extend (rather than replace) the RARP/TFTP two phase booting model [3], in order to leverage the existing system infrastructure. This is in contrast to the subsequent DHCP [9] work,

which extended BOOTP.

The "hands-off" installation of all kinds of systems (including diskless workstations, and servers) was required, as supported by LocalTalk networks [8]. However, Internet administrative models are not set up to allow that: there is no way to set up a completely functional IP network by just plugging machines into a cable and powering them up. That procedure doesn't have a way to input the network number (and class) that must be used, or to bootstrap the host naming system. An approach based on administered servers was needed for IP-based "Intranet" systems, even though that unfortunately called for networks to be initially set up by knowledgeable staff before any "hands-off" installations could be performed.

1.2 System Overview

DRARP was used by systems in the first phase of joining a network, to acquire a network address without personal intervention by a network administrator. Once a system was given a network address, it would perform whatever network operations it desired, subject to a site's access control policies. During system installation, those network operations involved a (re)configuration protocol ("Plug'n'Play", or PNP). Diskless sytems used TFTP to download code which could speak the PNP protocol.

The PNP protocol would register the names of newly installed hosts in the naming service, using the address which was acquired using DRARP. These names could be chosen by users installing the system, but could also be assigned automatically. Diskless systems used the PNP protocol to assign booting resources (e.g. filesystem space) on servers. All systems were assigned public and private keys, also initial (quasi-secret) "root" passwords, so that they could use what was then the strongest available ONC RPC authentication system.

Servers for DRARP and for the configuration protocol (as well as other administrative tools) needed to consult an authoritative database of which Internet addresses which were allocated to which hosts (as identified by hardware addresses). This "address authority" role was implemented using a name service (NIS) and an RPC-based centralized IP address allocation protocol ("IPalloc"). Address allocation could be performed only by authorized users, including network administrators and DRARP servers.

Most systems used DRARP and PNP each time they started, to automatically reconfigure applicable system and network policies. For example, network addresses and numbers were changed using these protocols; host names changed less often. The naming service (NIS)

held most information, such as the locations of printers and users' home directories.

2. Dynamic RARP Extensions

Dynamic RARP (DRARP) service is provided by any of a small active set of cooperating server systems on a network segment (network or subnetwork). Those servers are contacted through link level procedures, normally a packet broadcast. One or more servers may respond to a given request. It was intended that network segments will be administered together in domains [5] consisting of one or more network segments. Domains sharing a network segment need to share information about network addresses, both hardware level and protocol level, so an address authority (see section 3) can avoid reallocating protocol addresses which are already allocated or in use.

Dynamic RARP benefits from link layer addresses which are scoped more widely than just the local network segment. It takes advantage of such scoping to detect hosts which move between network segments. Such scoping is provided by IEEE 802 48-bit addresses [7], but not by all other kinds of network address. Without such a widely scoped ID, the case of systems roaming between networks can't be detected by Dynamic RARP.

2.1 Mixing RARP and DRARP Servers

DRARP is an extension to RARP, so that all Dynamic RARP servers are also RARP servers. However, DRARP provides a more manageable service model than RARP does: while RARP allows multiple servers to respond to RARP requests, it does not expect all those servers to be able to respond, or to respond identically. A given RARP server can not be relied upon to know whether a given link level address can be mapped into a protocol address, and some other RARP server may have a different answer.

Dynamic RARP addresses this problem by requiring that all Dynamic RARP servers on a network segment must communicate with the same address authority. That address authority controls name and address bindings, records bindings between host identifiers and addresses, makes decisions about how to allocate addresses, and keeps records about addresses in use.

This means that in effect there may be a number of independent RARP services offered along with a single DRARP service. DRARP service may well be offered through multiple servers, and the persistent address bindings it serves will be accessible as from a set of coordinated RARP servers.

Not all networks want to support dynamic address allocation services. Even those that do support it will need control over implementation of the address authority. So DRARP servers need policy controls such as "restricting" them from assigning addresses (applied to an entire network segment) as well as disabling use of DRARP entirely. (One may need to disable servers that would otherwise allocate new addresses, in order to enable ones which can speak to the "correct" address authority. Standards do not exist for protocols and security options used to talk to address authorities.)

2.2 Packet Format

The packet format is identical to RARP and is encapsulated using RARP frames, with the same Ethernet/SNAP type field. [1, 2, 6]. That is, a DRARP packet looks like a RARP packet, but it uses opcodes which are ignored by RARP servers; DRARP servers must also support RARP requests, and hence ARP requests [1].

2.2.1 RARP Packets

The two RARP opcodes are described here, in order to clarify the overall presentation. The name "REVARP", used in the opcode descriptions, is a synonym for "RARP".

REVARP_REQUEST (3)
: REVARP_REQUEST packets are sent to RARP servers as a request to map the target hardware address (tha) into the corresponding target protocol address (tpa), sending the response to the source hardware address (sha) as encoded in the packet. The source hardware address will usually be the same as the target hardware address, that of the system sending the packet. RARP servers will consult their name and address databases, and return a REVARP_REPLY packet if they can perform the reverse address resolution as requested.

REVARP_REPLY (4)
: This packet is sent by RARP servers in response to REVARP_REQUEST packets. The target protocol address (tpa) is filled in as requested, and the source hardware and protocol addresses (sha, spa) correspond to the RARP server. The target hardware address (tha) is from the corresponding REVARP_REQUEST packet, and the packet is sent to the source hardware address (sha) from that packet.

  This packet is also sent by Dynamic RARP servers in response to DRARP_REQUEST packets, if the protocol address returned was not a temporary one, but was instead what it would have returned given an otherwise identical REVARP_REQUEST packet.

2.2.2 Dynamic RARP Packets

There are three opcodes defined for DRARP, in addition to the two already defined for RARP:

DRARP_REQUEST (5)
: DRARP_REQUEST packets have the same format as REVARP_REQUEST packets, except for the operation code. The semantics are similar, except that in cases where a REVARP_REQUEST would produce no REVARP_REPLY (no persistent address mapping is stored in an addressing database) a DRARP_REQUEST will normally return a temporary address allocation in a DRARP_REPLY packet. A DRARP_ERROR packet may also be returned; a Dynamic RARP server will always provide a response, unlike a REVARP server.

DRARP_REPLY (6)
: DRARP_REPLY packets have the same format, opcode excepted, as REVARP_REPLY packets. The interpretation of the fields is the same.

  There are semantic differences between the two packet types. First, the protocol address bindings returned in DRARP_REPLY packets are temporary ones, which will be recycled after some period (e.g. an hour). Those bindings returned in REVARP_REPLY packets are "persistent" addresses which typically change much more slowly. Second, it is explicitly a protocol error for DRARP_REPLY packets to be sent which differ except in the sender address fields. Also, DRARP_REPLY packets are generated only in response to DRARP_REQUEST packets.

  These temporary addresses may be reallocated to another system after some time period. A configuration protocol is normally used to ensure that reallocation does not occur.

DRARP_ERROR (7)
: DRARP_ERROR packets may also be sent in response to DRARP_REQUESTs. The format is identical to REVARP_REPLY, except for the opcode and that the target protocol address (tpa) field is replaced by an error code field. The error code field must be at least one byte long, and the first byte is used to encode an error status describing why no target protocol address (tpa) is being returned. The status values are:

  DRARPERR_RESTRICTED (1)
  : This network does not support dynamic address allocation. The response is definitive; the network is controlled so that no other DRARP_REPLY (for this hardware address) is legal until the network policy on dynamic address

allocation is changed, or until the client is otherwise assigned a persistent address binding. A REVARP_REQUEST might yield a REVARP_REPLY, however; non-cooperating RARP servers could be the very reason that dynamic address allocation was disabled.

DRARPERR_NOADDRESSES (2)
: This network supports dynamic address allocation, but all available protocol addresses in the local segment are in use, so none can be allocated now.

DRARPERR_SERVERDOWN (3)
: The service providing access to the address authority is temporarily unavailable. May also be returned if an address allocation was required and the required response took a "long time" to generate; this distinguishes the case of a network that didn't support DRARP from the case of one that does, but is slow.

DRARPERR_MOVED (4)
: Analogous to the DRARPERR_RESTRICTED status in that no address was dynamically allocated. This provides the additional status that this client was recognized by the administration software for the domain as being on a different network segment than expected; users will be able to remedy the problem by connecting the system to the correct network segment.

DRARPERR_FAILURE (5)
: For some reason, no address could be returned. No defined status code known to the server explained the reason.

More opcodes for the Address Resolution Protocol (ARP) family could be defined in the future, so unrecognized opcodes (and error codes) should be ignored rather than treated as errors.

2.3 Protocol Exchanges

This section describes typical protocol exchanges using RARP and Dynamic RARP, and common fault modes of each exchange.

2.3.1. RARP Address Lookup

To determine a previously published ("persistent") protocol address for itself or another system, a system may issue a REVARP_REQUEST packet. If a REVARP_REPLY packet arrives in response, then the target protocol address listed there should be used.

If no REVARP_REPLY response packet arrives within some time interval, a number of errors may have occurred. The simplest one is that the request or reply packet may never have arrived: most RARP client implementations retransmit requests to partially account for this error. There is no clear point at which to stop retransmitting a request, so many implementations apply an exponential backoff to the retransmit interval, to reduce what is typically broadcast traffic.

Otherwise there are many different errors which all have the same failure mode, including: the system might not have a published protocol address; it might be on the wrong network segment, so its published address is invalid; the RARP servers which can supply the published address may be unavailable; it might even be on a network without any RARP servers at all.

2.3.2 Dynamic RARP Address Lookup

Dynamic RARP may be used to determine previously published protocol addresses by clients who issue DRARP_REQUEST packets. If the client has a published protocol address on the network segment on which the DRARP_REQUEST packet was issued, it is returned in a REVARP_REPLY packet.

If the client has a published protocol address only on some other network segment, then two basic responses are possible. In the case where dynamic address reallocation is enabled, a temporary protocol address may be allocated and returned in a DRARP_REPLY packet. Otherwise if dynamic address reallocation is disabled, a DRARP_ERROR packet is returned with the status DRARPERR_MOVED. Detection of host movement can be provided only with link level addresses that are unique over the catenet, such as are provided with IEEE 802 48 bit addresses. Without such uniqueness guarantees, this case looks like a request for a new address as described in the next section.

2.3.3 Dynamic RARP Address Allocation

Dynamic RARP clients who issue DRARP_REQUEST packets may acquire newly allocated protocol addresses. If the client has no published protocol address, there are three responses:

(a) When dynamic address allocation is enabled, a temporary protocol address is allocated and returned in a DRARP_REPLY packet.

(b) Errors or delays in the allocation process (with dynamic address allocation enabled) are reported in DRARP_ERROR packets with error codes such as DRARPERR_SERVERDOWN, DRARPERR_NOADDRESSES, DRARPERR_MOVED, or even DRARPERR_FAILURE.

(c) When dynamic address allocation is disabled (or "restricted"), a DRARP_ERROR packet with status DRARPERR_RESTRICTED is returned.

DRARP_REQUESTS are normally retransmitted until an address is returned, using backoff-style algorithms to minimize needless network traffic. When DRARP_ERROR responses are received, they should be reported to the user. For example, knowing that the server is busy could indicate it's time for a cup of Java, but if the network is restricted then it might be time to contact a network administrator for help instead.

2.3.4 Discovering Other DRARP Servers

The existence of a DRARP server can be discovered by the fact that it puts its addressing information in all DRARP_REPLY packets that it sends. DRARP servers can listen for such packets, as well as announcing themselves by sending such a packet to themselves.

It can be important to discover other DRARP servers. Users make mistakes, and can inappropriately set up DRARP servers that do not coordinate their address allocation with that done by the other DRARP servers on their network segment. That causes significant administrative problems, which can all but be eliminated by DRARP servers which politely announce themselves, and when they detect an apparently spurious server, report this fact before entering a "restricted" mode to avoid creating any problems themselves.

As no further server-to-server protocol is defined here, some out-of-band mechanism, such as communication through the address authority, must be used to help determine which servers are in fact spurious.

2.4 Network Setup Concerns

Some internetwork environments connect multiple network segments using link level bridges or routers. In such environments, a given broadcast accessible "local" area network will have two problems worth noting.

First, it will extend over several cable segments, and be subject to partitioning faults. Assigning one DRARP server to each segment (perhaps on systems acting as routers or bridges, to serve multiple segments) can reduce the cost of such faults. Assigning more than one such server can help reduce the cost of failure to any single network segment; these cooperate in the assignment of addresses through the address authority.

Second, those networks are sometimes shared by organizations which don't cooperate much on the management of protocol addresses, or perhaps aren't even collocated. A DRARP server might need help from link level bridges/routers in order to ensure that local clients are tied to local servers (rather than, for example, to servers across the country where they are prone to availability problems). Or the server might need to run in "restricted" mode so that a network administrator manually assigns address and other resources to each system.

3. The Address Authority

While not part of the DRARP protocol, the Address Authority used by the DRARP servers on a network segment is critical to providing the address allocation functionality. It manages the data needed to implement such service, which is required not just for dynamic address allocation tools. This section is provided to record one set of requirements for such an authority, ignoring implementation isssues such as whether protocol support for replication or partitioning is needed.

3.1 Basic Requirements

For each network segment under its control, an Address Authority maintains at least:

- persistent bindings between hardware and protocol addresses (for at least those hosts which are DRARP clients);
- temporary bindings between such addresses;
- protocol addresses available for temporary bindings;

The Address Authority is also responsible for presenting and managing those bindings. DRARP clients need it to support:

- creating temporary bindings initially,
- looking up bindings (the distinction between temporary and persistent bindings is not usually significant here),
- deleting temporary or persistent bindings on request,
- purging them automatically by noticing that a binding is now persistent or that the temporary address is available for reuse.

Those clients will frequently make concurrent requests, and should be required to pass some kind of authorization check before they create or change any bindings. They may also need to know about other clients, in order to determine (for example) if a given DRARP server is spurious.

3.2 Multiple Authorities and Segments

Note there is only a single address authority on a given network segment. It may be desirable to partition that authority, though that complicates implementation and administration of the authority substantially.

If detection of systems moving between network segments is to be provided, then the authorities for those two network segments must either be the same or (equivalently) must communicate with one another. Also, as noted earlier, hardware addresses must be scoped widely enough that the two segments do not assign the same link level address to different hosts.

3.3 Quality of Service

The records of temporary address bindings must be persistent for at least long enough to install a system and propagate its records through the site's administrative databases, even in the case of server or network faults. A timeout mechanism could be used to ensure that the limited address space was not used up too quickly. The initial implementation found that an hour's worth of caching, before deleting temporary bindings, was sufficient.

Experience has shown that many networks have addresses in use which are not listed in their name services (or other administrative databases). On such networks, the Address Authority should have a way to learn when an address which it thinks is available for allocation is instead being actively used. Probing the network for "the truth" before handing out what turns out to be a duplicate IP address is a worthwhile. Both ARPing for the address and ICMP echo request have been used for this.

4. Security Considerations

Security concerns are not addressed in this memo. They are recognized as significant, but they also interact with site-specific network administration policies. Those policies need to be addressed at higher levels before ramifications at this level can be understood.

5. References

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

[2] Finlayson, R., Mann, T., Mogul, J., and M. Theimer, "A Reverse Address Resolution Protocol", STD 38, RFC 903, Stanford, June 1984.

[3] Finlayson, R., "Bootstrap Loading using TFTP", RFC 906, Stanford, June 1984.

[4] Postel, J., "Multi-LAN Address Resolution", RFC 925, USC/Information Sciences Institute, October 1984.

[5] Mockapetris, P., "Domain Names -- Concepts and Facilities", STD 13, RFC 1034, USC/Information Sciences Institute, November 1987.

[6] Postel, J., and J. Reynolds, "A Standard for the Transmission of IP Datagrams over IEEE802 Networks", STD 43, RFC 1042, USC/Information Sciences Institute, February 1988.

[7] IEEE; "IEEE Standards for Local Area Networks: Logical Link Control" (IEEE 802.2); IEEE, New York, NY; 1985.

[8] United States Patent No. 4,689,786; "Local Area Network with Self Assigned Address Method"; Issued August 25, 1987; Inventors: Sidhu, et al.; Assignee: Apple Computer, Inc.

[9] Droms, R., "Dynamic Host Configuration Protocol", RFC 1541, Bucknell University, October 1993.

[10] Srinivasan, R., "RPC: Remote Procedure Call Protocol Specification, Version 2", RFC 1831, Sun Microsystems, August 1995.

Author’s Address:

David Brownell
SunSoft, Inc
2550 Garcia Way, MS 19-215
Mountain View, CA 94043

Phone: +1-415-336-1615
EMail: dbrownell@sun.com