Network Working Group
Request for Comments: 922
Jeffrey Mogul
Computer Science Department
Stanford University
October 1984

BROADCASTING INTERNET DATAGRAMS IN THE PRESENCE OF SUBNETS

Status of this Memo

We propose simple rules for broadcasting Internet datagrams on local networks that support broadcast, for addressing broadcasts, and for how gateways should handle them.

This RFC suggests a proposed protocol for the ARPA-Internet community, and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

Acknowledgement

This proposal here is the result of discussion with several other people, especially J. Noel Chiappa and Christopher A. Kent, both of whom both pointed me at important references.

1. Introduction

The use of broadcasts, especially on high-speed local area networks, is a good base for many applications. Since broadcasting is not covered in the basic IP specification [12], there is no agreed-upon way to do it, and so protocol designers have not made use of it. (The issue has been touched upon before, e.g. [6], but has not been the subject of a standard.)

We consider here only the case of unreliable, unsequenced, possibly duplicated datagram broadcasts (for a discussion of TCP broadcasting, see [10].) Even though unreliable and limited in length, datagram broadcasts are quite useful [1].

We assume that the data link layer of the local network supports efficient broadcasting. Most common local area networks do support broadcast; for example, Ethernet [7, 5], ChaosNet [9], token ring networks [2], etc.

We do not assume, however, that broadcasts are reliably delivered. (One might consider providing a reliable datagram broadcast protocol as a layer above IP.) It is quite expensive to guarantee delivery of broadcasts; instead, what we assume is that a host will receive most of the broadcasts that are sent. This is important to avoid excessive use of broadcasts; since every host on the network devotes at least some effort to every broadcast, they are costly.

When a datagram is broadcast, it imposes a cost on every host that hears it. Therefore, broadcasting should not be used indiscriminately, but rather only when it is the best solution to a problem.

2. Terminology

Because broadcasting depends on the specific data link layer in use on a local network, we must discuss it with reference to both physical networks and logical networks.

The terms we will use in referring to physical networks are, from the point of view of the host sending or forwarding a broadcast:

Local Hardware Network

The physical link to which the host is attached.

Remote Hardware Network

A physical network which is separated from the host by at least one gateway.

Collection of Hardware Networks

A set of hardware networks (transitively) connected by gateways.

The IP world includes several kinds of logical network. To avoid ambiguity, we will use the following terms:

Internet

The DARPA Internet collection of IP networks.

IP Network

One or a collection of several hardware networks that have one specific IP network number.

Subnet

A single member of the collection of hardware networks that compose an IP network. Host addresses on a given subnet share an IP network number with hosts on all other subnets of that IP network, but the local-address part is divided into subnet-number and host-number fields to indicate which subnet a host is on. We do not assume a particular division of the local-address part; this could vary from network to network.

The introduction of a subnet level in the addressing hierarchy is at variance with the IP specification [12], but as the use of addressable subnets proliferates it is obvious that a broadcasting scheme should support subnetting. For more on subnets, see [8].

In this paper, the term "host address" refers to the host-on-subnet address field of a subnetted IP network, or the host-part field otherwise.

An IP network may consist of a single hardware network or a collection of subnets; from the point of view of a host on another IP network, it should not matter.

3. Why Broadcast?

Broadcasts are useful when a host needs to find information without knowing exactly what other host can supply it, or when a host wants to provide information to a large set of hosts in a timely manner.

When a host needs information that one or more of its neighbors might have, it could have a list of neighbors to ask, or it could poll all of its possible neighbors until one responds. Use of a wired-in list creates obvious network management problems (early binding is inflexible). On the other hand, asking all of one's neighbors is slow if one must generate plausible host addresses, and try them until one works. On the ARPANET, for example, there are roughly 65 thousand plausible host numbers. Most IP implementations have used wired-in lists (for example, addresses of "Prime" gateways.) Fortunately, broadcasting provides a fast and simple way for a host to reach all of its neighbors.

A host might also use a broadcast to provide all of its neighbors with some information; for example, a gateway might announce its presence to other gateways.

One way to view broadcasting is as an imperfect substitute for multicasting, the sending of messages to a subset of the hosts on a network. In practice, broadcasts are usually used where multicasts are what is wanted; datagrams are broadcast at the hardware level, but filtering software in the receiving hosts gives the effect of multicasting.

For more examples of broadcast applications, see [1, 3].

4. Broadcast Classes

There are several classes of IP broadcasting:

- Single-destination datagrams broadcast on the local hardware net: A datagram is destined for a specific IP host, but the sending host broadcasts it at the data link layer, perhaps to avoid having to do routing. Since this is not an IP broadcast, the IP layer is not involved, except that a host should discard datagram not meant for it without becoming flustered (i.e., printing an error message).

- Broadcast to all hosts on the local hardware net: A distinguished value for the host-number part of the IP address denotes broadcast instead of a specific host. The receiving IP layer must be able to recognize this address as well as its own. However, it might still be useful to distinguish at higher levels between broadcasts and non-broadcasts, especially in gateways. This is the most useful case of broadcast; it allows a host to discover gateways without wired-in tables, it is the basis for address resolution protocols, and it is also useful for accessing such utilities as name servers, time servers, etc., without requiring wired-in addresses.

- Broadcast to all hosts on a remote hardware network: It is occasionally useful to send a broadcast to all hosts on a non-local network; for example, to find the latest version of a hostname database, to bootload a host on a subnet without a bootserver, or to monitor the timeservers on the subnet. This case is the same as local-network broadcasts; the datagram is routed by normal mechanisms until it reaches a gateway attached to the destination hardware network, at which point it is broadcast. This class of broadcasting is also known as "directed broadcasting", or quaintly as sending a "letter bomb" [1].

- Broadcast to all hosts on a subnetted IP network (Multi-subnet broadcasts): A distinguished value for the subnet-number part of the IP address is used to denote "all subnets". Broadcasts to all hosts of a remote subnetted IP network are done just as directed broadcasts to a single subnet.

- Broadcast to the entire Internet: This is probably not useful, and almost certainly not desirable.

For reasons of performance or security, a gateway may choose not to forward broadcasts; especially, it may be a good idea to ban broadcasts into or out of an autonomous group of networks.

5. Broadcast Methods

A host's IP receiving layer must be modified to support broadcasting. In the absence of broadcasting, a host determines if it is the recipient of a datagram by matching the destination address against all of its IP addresses. With broadcasting, a host must compare the destination address not only against the host's addresses, but also against the possible broadcast addresses for that host.

The problem of how best to send a broadcast has been extensively discussed [1, 3, 4, 13, 14]. Since we assume that the problem has already been solved at the data link layer, an IP host wishing to send either a local broadcast or a directed broadcast need only specify the appropriate destination address and send the datagram as usual. Any sophisticated algorithms need only reside in gateways.

The problem of broadcasting to all hosts on a subnetted IP network is apparently somewhat harder. However, even in this case it turns out that the best known algorithms require no additional complexity in non-gateway hosts. A good broadcast method will meet these additional criteria:

- No modification of the IP datagram format.

- Reasonable efficiency in terms of the number of excess copies generated and the cost of paths chosen.

- Minimization of gateway modification, in both code and data space.

- High likelihood of delivery.

The algorithm that appears best is the Reverse Path Forwarding (RPF) method [4]. While RPF is suboptimal in cost and reliability, it is quite good, and is extremely simple to implement, requiring no additional data space in a gateway.

6. Gateways and Broadcasts

Most of the complexity in supporting broadcasts lies in gateways. If a gateway receives a directed broadcast for a network to which it is not connected, it simply forwards it using the usual mechanism. Otherwise, it must do some additional work.

6.1. Local Broadcasts

When a gateway receives a local broadcast datagram, there are several things it might have to do with it. The situation is unambiguous, but without due care it is possible to create infinite loops.

The appropriate action to take on receipt of a broadcast datagram depends on several things: the subnet it was received on, the destination network, and the addresses of the gateway.

- The primary rule for avoiding loops is "never broadcast a datagram on the hardware network it was received on". It is not sufficient simply to avoid repeating datagram that a gateway has heard from itself; this still allows loops if there are several gateways on a hardware network.

- If the datagram is received on the hardware network to which it is addressed, then it should not be forwarded. However, the gateway should consider itself to be a destination of the datagram (for example, it might be a routing table update.)

- Otherwise, if the datagram is addressed to a hardware network to which the gateway is connected, it should be sent as a (data link layer) broadcast on that network. Again, the gateway should consider itself a destination of the datagram.

- Otherwise, the gateway should use its normal routing procedure to choose a subsequent gateway, and send the datagram along to it.

6.2. Multi-subnet broadcasts

When a gateway receives a broadcast meant for all subnets of an IP network, it must use the Reverse Path Forwarding algorithm to decide what to do. The method is simple: the gateway should forward copies of the datagram along all connected links, if and only if the datagram arrived on the link which is part of the best route between the gateway and the source of the datagram. Otherwise, the datagram should be discarded.

This algorithm may be improved if some or all of the gateways exchange among themselves additional information; this can be done transparently from the point of view of other hosts and even other gateways. See [4, 3] for details.

6.3. Pseudo-Algol Routing Algorithm

This is a pseudo-Algol description of the routing algorithm a gateway should use. The algorithm is shown in figure 1. Some definitions are:

RouteLink(host)

A function taking a host address as a parameter and returning the first-hop link from the gateway to the host.

RouteHost(host)

As above but returns the first-hop host address.

ResolveAddress(host)

Returns the hardware address for an IP host.

IncomingLink

The link on which the packet arrived.

OutgoingLinkSet

The set of links on which the packet should be sent.

OutgoingHardwareHost

The hardware host address to send the packet to.

Destination.host

The host-part of the destination address.

Destination.subnet

The subnet-part of the destination address.

Destination.ipnet

The IP-network-part of the destination address.

```
BEGIN
   IF Destination.ipnet IN AllLinks THEN
      BEGIN
         IF IsSubnetted(Destination.ipnet) THEN
            BEGIN
               IF Destination.subnet = BroadcastSubnet THEN
                  BEGIN      /* use Reverse Path Forwarding algorithm */
                     IF IncomingLink = RouteLink(Source) THEN
                        BEGIN IF Destination.host = BroadcastHost THEN
                              OutgoingLinkSet <- AllLinks -
                           IncomingLink;
                           OutgoingHost <- BroadcastHost;
                           Examine packet for possible internal use;
                        END
                     ELSE  /* duplicate from another gateway, discard */
                        Discard;
                  END
               ELSE
                  IF Destination.subnet = IncomingLink.subnet THEN
                     BEGIN           /* forwarding would cause a loop */
                        IF Destination.host = BroadcastHost THEN
                           Examine packet for possible internal use;
                        Discard;
                     END
                  ELSE BEGIN    /* forward to (possibly local) subnet */
                        OutgoingLinkSet <- RouteLink(Destination);
                        OutgoingHost <- RouteHost(Destination);
                     END
            END
         ELSE BEGIN         /* destined for one of our local networks */
               IF Destination.ipnet = IncomingLink.ipnet THEN
                  BEGIN              /* forwarding would cause a loop */
                     IF Destination.host = BroadcastHost THEN
                        Examine packet for possible internal use;
                     Discard;
                  END
               ELSE BEGIN                     /* might be a broadcast */
                     OutgoingLinkSet <- RouteLink(Destination);
                     OutgoingHost <- RouteHost(Destination);
                  END
            END
      END
   ELSE BEGIN                    /* forward to a non-local IP network */
         OutgoingLinkSet <- RouteLink(Destination);
         OutgoingHost <- RouteHost(Destination);
      END
   OutgoingHardwareHost <- ResolveAddress(OutgoingHost);
END
```

Figure 1: Pseudo-Algol algorithm for routing broadcasts by gateways

7. Broadcast IP Addressing - Conventions

If different IP implementations are to be compatible, there must be convention distinguished number to denote "all hosts" and "all subnets".

Since the local network layer can always map an IP address into data link layer address, the choice of an IP "broadcast host number" is somewhat arbitrary. For simplicity, it should be one not likely to be assigned to a real host. The number whose bits are all ones has this property; this assignment was first proposed in [6]. In the few cases where a host has been assigned an address with a host-number part of all ones, it does not seem onerous to require renumbering.

The "all subnets" number is also all ones; this means that a host wishing to broadcast to all hosts on a remote IP network need not know how the destination address is divided up into subnet and host fields, or if it is even divided at all. For example, 36.255.255.255 may denote all the hosts on a single hardware network, or all the hosts on a subnetted IP network with 1 byte of subnet field and 2 bytes of host field, or any other possible division.

The address 255.255.255.255 denotes a broadcast on a local hardware network that must not be forwarded. This address may be used, for example, by hosts that do not know their network number and are asking some server for it.

Thus, a host on net 36, for example, may:

- broadcast to all of its immediate neighbors by using 255.255.255.255

- broadcast to all of net 36 by using 36.255.255.255

without knowing if the net is subnetted; if it is not, then both addresses have the same effect. A robust application might try the former address, and if no response is received, then try the latter. See [1] for a discussion of such "expanding ring search" techniques.

If the use of "all ones" in a field of an IP address means "broadcast", using "all zeros" could be viewed as meaning "unspecified". There is probably no reason for such addresses to appear anywhere but as the source address of an ICMP Information Request datagram. However, as a notational convention, we refer to networks (as opposed to hosts) by using addresses with zero fields. For example, 36.0.0.0 means "network number 36" while 36.255.255.255 means "all hosts on network number 36".

7.1. ARP Servers and Broadcasts

The Address Resolution Protocol (ARP) described in [11] can, if incorrectly implemented, cause problems when broadcasts are used on a network where not all hosts share an understanding of what a broadcast address is. The temptation exists to modify the ARP server so that it provides the mapping between an IP broadcast address and the hardware broadcast address.

This temptation must be resisted. An ARP server should never respond to a request whose target is a broadcast address. Such a request can only come from a host that does not recognize the broadcast address as such, and so honoring it would almost certainly lead to a forwarding loop. If there are N such hosts on the physical network that do not recognize this address as a broadcast, then a datagram sent with a Time-To-Live of T could potentially give rise to T**N spurious re-broadcasts.

8. References

1. David Reeves Boggs. Internet Broadcasting. Ph.D. Th., Stanford University, January 1982.

2. D.D. Clark, K.T. Pogran, and D.P. Reed. "An Introduction to Local Area Networks". Proc. IEEE 66, 11, pp1497-1516, November 1978.

3. Yogan Kantilal Dalal. Broadcast Protocols in Packet Switched Computer Networks. Ph.D. Th., Stanford University, April 1977.

4. Yogan K. Dalal and Robert M. Metcalfe. "Reverse Path Forwarding of Broadcast Packets". Comm. ACM 21, 12, pp1040-1048, December 1978.

5. The Ethernet, A Local Area Network: Data Link Layer and Physical Layer Specifications. Version 1.0, Digital Equipment Corporation, Intel, Xerox, September 1980.

6. Robert Gurwitz and Robert Hinden. IP - Local Area Network Addressing Issues. IEN-212, BBN, September 1982.

7. R.M. Metcalfe and D.R. Boggs. "Ethernet: Distributed Packet Switching for Local Computer Networks". Comm. ACM 19, 7, pp395-404, July 1976. Also CSL-75-7, Xerox Palo Alto Research Center, reprinted in CSL-80-2.

8. Jeffrey Mogul. Internet Subnets. RFC-917, Stanford University, October 1984.

9. David A. Moon. Chaosnet. A.I. Memo 628, Massachusetts Institute of Technology Artificial Intelligence Laboratory, June 1981.

10. William W. Plummer. Internet Broadcast Protocols. IEN-10, BBN, March 1977.

11. David Plummer. An Ethernet Address Resolution Protocol. RFC-826, Symbolics, September 1982.

12. Jon Postel. Internet Protocol. RFC-791, ISI, September 1981.

13. David W. Wall. Mechanisms for Broadcast and Selective Broadcast. Ph.D. Th., Stanford University, June 1980.

14. David W. Wall and Susan S. Owicki. Center-based Broadcasting. Computer Systems Lab Technical Report TR189, Stanford University, June 1980.