Network Working Group
Request for Comments: 1546
Category: Informational

C. Partridge
T. Mendez
W. Milliken
BBN

November 1993

Host Anycasting Service

Status of this Memo

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

Abstract

This RFC describes an internet anycasting service for IP. The primary purpose of this memo is to establish the semantics of an anycasting service within an IP internet. Insofar as is possible, this memo tries to be agnostic about how the service is actually provided by the internetwork. This memo describes an experimental service and does not propose a protocol. This memo is produced by the Internet Research Task Force (IRTF).

Motivation

There are a number of situations in networking where a host, application, or user wishes to locate a host which supports a particular service but, if several servers support the service, does not particularly care which server is used. Anycasting is a internetwork service which meets this need. A host transmits a datagram to an anycast address and the internetwork is responsible for providing best effort delivery of the datagram to at least one, and preferably only one, of the servers that accept datagrams for the anycast address.

The motivation for anycasting is that it considerably simplifies the task of finding an appropriate server. For example, users, instead of consulting a list of archie servers and choosing the closest server, could simply type:

```
telnet archie.net
```

and be connected to the nearest archie server. DNS resolvers would no longer have to be configured with the IP addresses of their servers, but rather could send a query to a well-known DNS anycast address. Mirrored FTP sites could similarly share a single anycast address, and users could simply FTP to the anycast address to reach the nearest server.

Architectural Issues

Adding anycasting to the repertoire of IP services requires some decisions to be made about how to balance the architectural requirements of IP with those of anycasting. This section discusses these architectural issues.

The first and most critical architectural issue is how to balance IP's stateless service with the desire to have an anycast address represent a single virtual host. The best way to illustrate this problem is with a couple of examples. In both of these examples, two hosts (X and Y) are serving an anycast address and another host (Z) is using the anycast address to contact a service.

In the first example, suppose that Z sends a UDP datagram addressed to the anycast address. Now, given that an anycast address is logically considered the address of a single virtual host, should it be possible for the datagram to be delivered to both X and Y? The answer to this question clearly has to be yes, delivery to both X and Y is permissible. IP is allowed to duplicate and misroute datagrams so there clearly are scenarios in which a single datagram could be delivered to both X and Y. The implication of this conclusion is that the definition of anycasting in an IP environment is that IP anycasting provides best effort delivery of an anycast datagram to one, but possibly more than one, of the hosts that serve the destination anycast address.

In the second example, suppose that Z sends two datagrams addressed to the anycast address. The first datagram gets delivered to X. To which host (X or Y) does the second datagram get delivered? It would be convenient for stateful protocols like TCP if all of a connection's datagrams were delivered to the same anycast address. However, because IP is stateless (and thus cannot keep track of where earlier datagrams were delivered) and because one of the goals of anycasting is to support replicated services, it seems clear that the second datagram can be delivered to either X or Y. Stateful protocols will have to employ some additional mechanism to ensure that later datagrams are sent to the same host. Suggestions for how to accomplish this for TCP are discussed below.

After considering the two examples, it seems clear that the correct definition of IP anycasting is a service which provides a stateless best effort delivery of an anycast datagram to at least one host, and preferably only one host, which serves the anycast address. This definition makes clear that anycast datagrams receive the same basic type of service as IP datagrams. And while the definition permits delivery to multiple hosts, it makes clear that the goal is delivery to just one host.

Anycast Addresses

There appear to be a number of ways to support anycast addresses, some of which use small pieces of the existing address space, others of which require that a special class of IP addresses be assigned.

The major advantage of using the existing address space is that it may make routing easier. As an example, consider a situation where a portion of each IP network number can be used for anycasting. I.e., a site, if it desires, could assign a set of its subnet addresses to be anycast addresses. If, as some experts expect, anycast routes are treated just like host routes by the routing protocols, the anycast addresses would not require special advertisement outside the site -- the host routes could be folded in with the net route. (If the anycast addresses is supported by hosts outside the network, then those hosts would still have be advertised using host routes). The major disadvantages of this approach are (1) that there is no easy way for stateful protocols like TCP to discover that an address is an anycast address, and (2) it is more difficult to support internet-wide well-known anycast address. The reasons TCP needs to know that an address is an anycast address is discussed in more detail below. The concern about well-known anycast addresses requires a bit of explanation. The idea is that the Internet might establish that a particular anycast address is the logical address of the DNS server. Then host software could be configured at the manufacturer to always send DNS queries to the DNS anycast address. In other words, anycasting could be used to support autoconfiguration of DNS resolvers.

The major advantages of using a separate class of addresses are that it is easy to determine if an address is an anycast address and well-known anycast addresses are easier to support. The key disadvantage is that routing may be more painful, because the routing protocols may have to keep track of more anycast routes.

An intermediate approach is to take part of the current address space (say 256 Class C addresses) and make the network addresses into anycast addresses (and ignore the host part of the class C address). The advantage of this approach is that it makes anycast routes look

like network routes (which are easier for some routing protocols to handle). The disadvantages are that it uses the address space inefficiently and so more severely limits the number of anycast addresses that can be supported.

In the balance it seems wiser to use a separate class of addresses. Carving anycast addresses from the existing address space seems more likely to cause problems in situations in which either applications mistakenly fail to recognize anycast addresses (if anycasts are part of each site's address space) or use the address space inefficiently (if network addresses are used as anycast addresses). And the advantages of using anycast addresses for autoconfiguration seem compelling. So this memo assumes that anycast addresses will be a separate class of IP addresses (not yet assigned). Since each anycast address is a virtual host address and the number of anycasting hosts seems unlikely to be larger than the number of services offered by protocols like TCP and UDP, the address space could be quite small, perhaps supporting as little as 2**16 different addresses.

Transmission and Reception of Anycast Datagrams

Historically, IP services have been designed to work even if routers are not present (e.g., on LANs without routers). Furthermore, many in the Internet community have historically felt that hosts should not have to participate in routing protocols to operate. (See, for instance, page 7 of STD 3, RFC 1122). To provide an anycasting service that is consistent with these traditions, the handling of anycast addresses varies slightly depending on the type of network on which datagrams with anycast addresses are sent.

On a shared media network, such as an Ethernet and or Token Ring, it must be possible to transmit an anycast datagram to a server also on the same network without consulting a (possibly non-existent) router. There are at least two ways this can be done.

One approach is to ARP for the anycast address. Servers which support the anycast address can reply to the ARP request, and the sending host can transmit to the first server that responds. This approach is reminiscent of the ARP hack (RFC 1027) and like the ARP hack, requires ARP cache timeouts for the anycast addresses be kept small (around 1 minute), so that if an anycast server goes down, hosts will promptly flush the ARP entry and query for other servers supporting the anycast address.

Another approach is for hosts to transmit anycast datagrams on a link-level multicast address. Hosts which serve an anycast address would be expected to listen to the link-level multicast address for

datagrams destined for their anycast address. By multicasting on the local network, there is no need for a router to route the anycast datagrams. One merit of this approach is that if there are multiple servers and one goes down, the others will still receive any requests. Another possible advantage is that, because anycast ARP entries must be quickly timed out, the multicasting approach may be less traffic intensive than the ARP approach because in the ARP approach, transmissions to an anycast address are likely to cause a broadcast ARP, while in the multicast approach, transmissions are only to a select multicast group. An obvious disadvantage is that if there are multiple servers on a network, they will all receive the anycast message, when delivery to only one server was desired.

On point-to-point links, anycast support is simpler. A single copy of the anycast datagram is forwarded along the appropriate link towards the anycast destination.

When a router receives an anycast datagram, the router must decide if it should forward the datagram, and if so, transmits one copy of the datagram to the next hop on the route. Note that while we may hope that a router will always know the correct next hop for an anycast datagram and will not have to multicast anycast datagrams on a local network, there are probably situations in which there are multiple servers on a local network, and to avoid sending to one that has recently crashed, routers may wish to send anycast datagrams on a link-level multicast address. Because hosts may multicast any datagrams, routers should take care not to forward a datagram if they believe that another router will also be forwarding it.

Hosts which wish to receive datagrams for a particular anycast address will have to advertise to routers that they have joined the anycast address. On shared media networks, the best mechanism is probably for a host to periodically multicast information about the anycast addresses it supports (possibly using an enhanced version of IGMP). The multicast messages ensure that any routers on the network hear that the anycast address is supported on the local subnet and can advertise that fact (if appropriate) to neighboring routers. Note that if there are no routers on the subnet, the multicast messages would simply simply ignored. (The multicasting approach is suggested because it seems likely to be simpler and more reliable than developing a registration protocol, in which an anycast server must register itself with each router on its local network).

On point-to-point links, a host can simply advertise its anycast addresses to the router on the other end of the link.

Observe that the advertisement protocols are a form of routing protocol and that it may make sense to simply require anycast servers

to participate (at least partly) in exchanges of regular routing messages.

When a host receives an IP datagram destined for an anycast address it supports, the host should treat the IP datagram just as if it was destined for one of the host's non-anycast IP addresses. If the host does not support the anycast address, it should silently discard the datagram.

Hosts should accept datagrams with an anycast source address, although some transport protocols (see below) may refuse to accept them.

How UDP and TCP Use Anycasting

It is important to remember that anycasting is a stateless service. An internetwork has no obligation to deliver two successive packets sent to the same anycast address to the same host.

Because UDP is stateless and anycasting is a stateless service, UDP can treat anycast addresses like regular IP addresses. A UDP datagram sent to an anycast address is just like a unicast UDP datagram from the perspective of UDP and its application. A UDP datagram from an anycast address is like a datagram from a unicast address. Furthermore, a datagram from an anycast address to an anycast address can be treated by UDP as just like a unicast datagram (although the application semantics of such a datagram are a bit unclear).

TCP's use of anycasting is less straightforward because TCP is stateful. It is hard to envision how one would maintain TCP state with an anycast peer when two successive TCP segments sent to the anycast peer might be delivered to completely different hosts.

The solution to this problem is to only permit anycast addresses as the remote address of a TCP SYN segment (without the ACK bit set). A TCP can then initiate a connection to an anycast address. When the SYN-ACK is sent back by the host that received the anycast segment, the initiating TCP should replace the anycast address of its peer, with the address of the host returning the SYN-ACK. (The initiating TCP can recognize the connection for which the SYN-ACK is destined by treating the anycast address as a wildcard address, which matches any incoming SYN-ACK segment with the correct destination port and address and source port, provided the SYN-ACK's full address, including source address, does not match another connection and the sequence numbers in the SYN-ACK are correct.) This approach ensures that a TCP, after receiving the SYN-ACK is always communicating with only one host.

Applications and Anycasting

In general, applications use anycast addresses like any other IP address. The only worrisome application use of anycasting is applications which try to maintain stateful connections over UDP and applications which try to maintain state across multiple TCP connections. Because anycasting is stateless and does not guarantee delivery of multiple anycast datagrams to the same system, an application cannot be sure that it is communicating with the same peer in two successive UDP transmissions or in two successive TCP connections to the same anycast address.

The obvious solutions to these issues are to require applications which wish to maintain state to learn the unicast address of their peer on the first exchange of UDP datagrams or during the first TCP connection and use the unicast address in future conversations.

Anycasting and Multicasting

It has often been suggested that IP multicasting can be used for resource location, so it is useful to compare the services offered by IP multicasting and IP anycasting.

Semantically, the difference between the two services is that an anycast address is the address of a single (virtual) host and that the internetwork will make an effort to deliver anycast datagrams to a single host. There are two implications of this difference. First, applications sending to anycast addresses need not worry about managing the TTLs of their IP datagrams. Applications using multicast to find a service must balance their TTLs to maximize the chance of finding a server while minimizing the chance of sending datagrams to a large number of servers it does not care about. Second, making a TCP connection to an anycast address makes perfectly good sense, while the meaning of making a TCP connection to a multicast address are unclear. (A TCP connection to a multicast address is presumably trying to establish a connection to multiple peers simultaneously, which TCP is not designed to support).

From a practical perspective, the major difference between anycasting and multicasting is that anycasting is a special use of unicast addressing while multicasting requires more sophisticated routing support. The important observation is that multiple routes to an anycast address appear to a router as multiple routes to a unicast destination, and the router can use standard algorithms to choose to the best route.

Another difference between the two approaches is that resource location using multicasting typically causes more datagrams to be sent. To find a server using multicasting, an application is expected to transmit and retransmit a multicast datagram with successively larger IP TTLs. The TTL is initially kept small to try to limit the number of servers contacted. However, if no servers respond, the TTL must be increased on the assumption that the available servers (if any) were farther away than was reachable with the initial TTL. As a result, resource location using multicasting causes one or more multicast datagrams to be sent towards multiple servers, with some datagrams' TTLs expiring before reaching a server. With anycasting, managing the TTL is not required and so (ignoring the case of loss) only one datagram need be sent to locate a server. Furthermore, this datagram will follow only a single path.

A minor difference between the two approaches is that anycast may be less fault tolerant than multicast. When an anycast server fails, some datagrams may continue to be mistakenly routed to the server, whereas if the datagram had been multicast, other servers would have received it.

Related Work

The ARPANET AHIP-E Host Access Protocol described in RFC 878 supports logical addressing which allows several hosts to share a single logical address. This scheme could be used to support anycasting within a PSN subnet.

Security Considerations

There are at least two security issues in anycasting, which are simply mentioned here without suggested solutions.

First, it is clear that malevolent hosts could volunteer to serve an anycast address and divert anycast datagrams from legitimate servers to themselves.

Second, eavesdropping hosts could reply to anycast queries with inaccurate information. Since there is no way to verify membership in an anycast address, there is no way to detect that the eavesdropping host is not serving the anycast address to which the original query was sent.

Acknowledgements

This memo has benefitted from comments from Steve Deering, Paul Francis, Christian Huitema, Greg Minshall, Jon Postel, Ram Ramanathan, and Bill Simpson. However, the authors are solely responsible for any dumb ideas in this work.

Authors' Addresses

Craig Partridge
Bolt Beranek and Newman
10 Moulton St
Cambridge MA 02138

EMail: craig@bbn.com

Trevor Mendez
Bolt Beranek and Newman
10 Moulton St
Cambridge MA 02138

EMail: tmendez@bbn.com

Walter Milliken
Bolt Beranek and Newman
10 Moulton St
Cambridge MA 02138

EMail: milliken@bbn.com