Internet Engineering Task Force
INTERNET-DRAFT
Expires: December 28, 2003

Jun-ichiro itojun Hagino
IIJ Research Laboratory
K. Ettikan
Intel ASG, Malaysia
June 28, 2003

An analysis of IPv6 anycast
draft-ietf-ipngwg-ipv6-anycast-analysis-02.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ``work in progress.''

To view the list Internet-Draft Shadow Directories, see http://www.ietf.org/shadow.html.

Distribution of this memo is unlimited.

The internet-draft will expire in 6 months. The date of expiration will be December 28, 2003.

Abstract

The memo tries to identify the problems and issues in the use of IPv6 anycast [Hinden, 1998] defined as of today. The goals of the draft are

(1) to understand the currently-defined IPv6 anycast better,

(2) to provide guidelines for people trying to deploy anycast services, and

(3) to suggest updates to IPv6 anycast protocol specification.

The document was made possible by input from ipngwg DNS discovery design team.

1. IPv6 anycast

"Anycast" is a communication model for IP, just like unicast and multicast are.

Anycast can be understood best by comparing with unicast and multicast. IP unicast allows a source node to transmit IP datagrams to a single destination node. The destination node is identified by a unicast address. IP multicast allows a source node to transmit IP datagrams to a group of destination nodes. The destination nodes are identfied by a multicast group, and we use a multicast address to identify the multicast group.

IP anycast allows a source node to transmit IP datagrams to a single destination node, out of a group of destination nodes. IP datagrams will reach the closest destination node in the set of destination nodes, based on the routing measure of distance. The source node does not need to care about how to pick the closest destination node, as the routing system will figure it out (in other words, the source node has no control over the selection). The set of destination nodes is identified by an anycast address.

Anycast was adopted by IPv6 specification suite. RFC2373 [Hinden, 1998] defines the IPv6 anycast address, and its constraints in the usage. The following sections try to analyze RFC2373 rules, and understand limitations with them. At the end of the draft we compile a couple of suggestions to exisitng proposals, for extending the usage of the IPv6 anycast.

2. Existing practices

There are multiple examples of anycast in IPv4. The section tries to summarize those practices.

2.1. RFC1546 anycast

RFC1546 [Partridge, 1993] defines an experimental anycast service for IPv4. With RFC1546, anycast address is distinguishable from unicast address (unlike RFC2373 anycast), as they are allocated from separate range. The authors have no knowledge whether RFC1546 anycast is widely practiced or not; our bet is that it is not.

2.2. Shared-unicast address: multiple hosts with single unicast address

There are existing practices of using a single unicast address at multiple different locations, for load balancing purposes, for DNS servers and web servers (1992 Olympic games) [Hardie, 2002] . We call the technique "shared-unicast address" for clarity in this document. The shared-unicast address technique works as follows:

- o A provider-independent IPv4 address prefix is allocated from an RIR.
- o The address prefix is configured at multiple distant locations on the Internet.
- o A host route, or a route that covers the address prefix is advertised from all of the locations.
- o Clients will reach the nearest location based on the routing table setup.

Shared-anycast address technique must not be confused with the one we discuss in the document (RFC2373 anycast), as the problem domain is different.

Shared-unicast address technique tries to replicate unicast servers. The distribution of servers is worldwide. Shared-unicast address is being used for specific upper-layer protocols only, like DNS and HTTP. There is no consideration given for the cases when a client contacts multiple servers by chance (transport layer protocol will get confused), since it is unlikely to see routing table changes during the short lifetime of the upper-layer protocols being used.

RFC2373 anycast is defined in more generic manner, and does not limit the routing infrastructure nor upper-layer protocol. Therefore, RFC2373 imposes certain limitation to the packet header contents (like IPv6 source address), to prevent confusions due to routing changes during the lifetime of a transport-layer connection.

This document tries to analyze RFC2373 anycast to understand if we can use it for site-scoped server replication, upper-layer protocols other than DNS or HTTP, and such. Still, it is possible to apply shared-unicast address technique to IPv6. Issues with shared-unicast address on IPv6 are outside of the scope of the document.

2.3. Route scaling issues

The use of anycast addresses has route scalaing issues. If anycast addresses are drawn from the unicast address space (as is the case in RFC2373 anycast and the shared-unicast address used for anycast DNS servers) the routing scaling impact can potentially be limited by aggregating the anycast addresses as part of the regular unicast routing prefixes. But this aggregation can only be applied when all members in the anycast group remaing within the piece of toplogy whose routes get aggregated. For instance having an anycast address where all the members belong to one ISP means it the anycast address can be drawn from the ISPs address space and be aggregated at the ISP boundary with no effect on route scaling outside that domain. But having e.g. anycast groups with members across the whole Internet will have effect on the size of the routing tables globally.

3. Limitations/properties of RFC2373 anycast

The section tries to list limitations and properties of RFC2373 anycast. Some of the properties listed herein applies to IPv4/v6 shared-unicast address technique as well.

3.1. Identifying anycast destination

For anycast addresses, RFC2373 uses the same address format as unicast addresses. Therefore, without other specific configurations, a sender usually cannot identify if the sender is sending a packet to anycast destination, or unicast destination. This is different from RFC1546 IPv4 anycast, where anycast address is distinguishable from unicast addresses.

This item applies to shared-unicast address technique as well.

3.2. Nondeterministic packet delivery

If multiple packets carry an anycast address in IPv6 destination address header, these packets may not reach the same destination node, depending on stability of the routing table. This property leads to a couple of interesting symptoms.

If we can assume that the routing table is stable enough during a protocol exchanges, multiple packets (with anycast address in destination address field) will reach the same destination node just fine. However, there is no such guarantee.

If routing table is not stable enough or you would like to take a more strict approach, a client can only send one packet with anycast address in the destination address field. For example, consider the following packet exchange. The following exchange can lead us to failure, as we are not sure if the 1st and 2nd anycast packet have reached the same destination.

```
query: client unicast (Cu) -> server anycast (Sa)
reply: server unicast (Su) -> client unicast (Cu)
query: client unicast (Cu) -> server anycast (Sa)
          It may reach a different server!
reply: server unicast (Su) -> client unicast (Cu)
```

Because of the non-determinism, if we take a strict approach, we can use no more than 1 packet with anycast destination address, in a set of protocol exchange. If we use more than 2 packets, 1st and 2nd packet may reach different server and may cause unexpected results. If the protocol is completely stateless, and we can consider the first roundtrip and second roundtrip separate, it is okay. For stateful protocols, it is suggested to use anycast for the first packet in the exchange, to discover unicast address of the (nearest) server. After we

have discovered the unicast address of the server, we should use the server's unicast address for the protocol exchange (note that there is

some security implication here - see Security Consideration section).

Also because of non-determinism, if we are to assign an IPv6 anycast address to servers, those servers must provide uniform services. For example, if server A and server B provide different services, and people wants to differentiate between A and B, we cannot use single IPv6 anycast address to identify both A and B.

Note that, this is not a bad feature of anycast; this property lets us use anycast addresses for load balancing. Also, packets will automatically be delivered to the nearest node with anycast address assigned. Anaycast will ease service locating problem by pusing the task to network layer rather than handled by upper layers.

Here are situations where multiple packets with anycast destination address can lead us to problems:

o Fragmented IPv6 packets. Fragments may reach multiple different destinations, and will prevent reassembly.

Because the sending node cannot differentiate between anycast addresses and unicast addresses, it is hard for the sending node to control the use of fragmentation.

This item applies to shared-unicast address technique as well.

3.3. Anycast address assignment to hosts

RFC2373 suggests to assign anycast addresses to a node, only when the node is a router. This is because there was no standard way for hosts to announce their intention to accept packets toward anycast addresses. If no hosts have anycast address on them, it is easier for us to route an IP datagram to anycast destination. We just need to follow existing routing entries, and we will eventually hit a router that has the anycast address. If we follow RFC2373 restriction strictly, we could only assign anycast addresses onto routers.

However, note that there is no inherent difference between IPv4 and IPv6 with respect to the use of anycast address on hosts; we can certainly run shared-unicast address technique with IPv6, with certain limitation/caveats as described previously.

3.4. Anycast address in source address

Under RFC2373, IPv6 anycast address can not be put into IPv6 source address. This is basically because an IPv6 anycast address does not identify a single source node.

o Incorrect reassembly of fragmented packets due to multiple anycast members sending packets with the same fragment ID to the same

destination at about the same time; the same the source IP address, destination IP address, nextheader, and fragment ID numbers might be

accidentally used at the same time by different senders.

o Errors and other response packets might be delivered to a different anycast member than sent the packet. This might be very likely since asymmetric routing is rather prevalent on the Internet.

  Particular cases of such errors that are known to cause protocol problems are (1) ICMP packet too big making path MTU discovery impossible. (2) (could be more) The misdelivery of other errors might cause operational problems - making the network harder to trouble-shoot when anycast source addresses are used.

The above items apply to shared-unicast address technique as well. In the actual shared-unicast IPv4 operation, path MTU discovery is avoided by setting DF bit to 0 for the packets from servers that share the shared-unicast address (it is not possible to do the same thing for IPv6, however, as DF bit is no longer present).

3.5. IPsec

IPsec and IKE identify nodes by using source/destination address pairs. Due to the combination of issues presented above, it is difficult to use IPsec on packets with anycast address in source address, destination address, or both.

Even with manual keying, IPsec trust model with anycast address is confusing. As IPsec uses IPv6 destination address to identify which IPsec key to be used, we need to use the same IPsec key for all of the anycast destinations that share an anycast address.

IPsec protocol has replay protection mechanism. If IPsec is used with an anycast address, it will not work well as replay counter will not be updated consistently due to the anycast packet delivery.

Dynamic IPsec key exchange (like IKE) is almost impossible. First of all, to run IKE session between two nodes, the two nodes need to be able to communicate with each other stably. With nondeterministic packet delivery provided by anycast, it is not quite easy. Even if we could circumvent the issue with IKE, we have exactly the same problem as manual keying case for actual communication.

This item applies to shared-unicast address technique as well.

4. Possible improvements and protocol changes

4.1. Assigning anycast address to hosts (non-router nodes)

Under RFC2373 rule, we can only assign anycast addresses to routers, not to hosts. The restriction was put into the RFC because it was felt

insecure to permit hosts to inject host routes to anycast address.

If we try to ease the restriction and assign anycast addresses to IPv6 hosts (non-routers), we would need to inject host routes for the anycast addresses, with prefix length set to /128, into the IPv6 routing system. We will inject host routes from each of the nodes with anycast addresses, to make packets routed to a topologically-closest node. Or, we may be able to inject host routes from routers adjacent to the servers (not from the servers themselvers).

Here are possible ways to allow anycast addresses to be assigned to hosts. We would need to diagnose each of the following proposals carefully, as they have different pros and cons. The most serious issue would be the security issue with service blackhole attack (malicious party can blackhole packets toward anycast addresses, by making false advertisement).

o Let the host with anycast address to participate into routing information exchange. The host does not need to fully participate; it only needs to announce the anycast address to the routing system. To secure routing exchange, administrators need to configure secret information that protects the routing exchange to the host, as well as other routers.

o Develop a protocol for a router, to discover hosts with anycast address on the same link. The router will then advertise the anycast address to the routing system. This could be done by an extension to IPv6 Neighbor Discovery or an extension to IPv6 Multicast Listener Discovery [Haberman, 2002] .

As described in section 2.3, the impact of host routes depends on the scope of the anycast address usage. If we deploy nodes with an RFC2373 anycast address worldwide, host route for RFC2373 anycast would impact the global routing table. If we deploy nodes with an RFC2373 anycast address in a single aggregatable address domain (such as a /48 site), host route will impact the aggregatable address domain only.

4.2. Anycast address in destination address

By using anycast in IPv6 layer, upper-layer protocols may be able to enjoy redundancy and higher availability of servers. However, for stateful upper-layer protocols, a client may need to specify a single node out of nodes that share an anycast address. Suppose a client C would like to communicate a specific server with anycast address, Si. Si shares the same anycast address with other servers, S1 to Sn. It is hard for C to selectively communicate with Si.

One possible workaround is to use IPv6 routing header. By specifying a unicast address of Si as an intermediate hop, C can deliver the packet to Si, not to other Sn.

Note that, however, by specifying Si explicitly, C now have lost the server redundancy provided by the use of anycast address in IPv6 layer. If Si goes down, the communication between C and Si will be lost. C

cannot enjoy the failure resistance provided by redundant servers, S1 to Sn. Protocol designers should carefully diagnose if any state is managed by C and/or Si, and decide how the protocol should take advantage of anycast addresses and their characteristics.

4.3. Anycast address in source address

Under RFC2373 rule, anycast address cannot be put into source address. Here is a possible workaround, however, it did not win a consensus in the past ipngwg meetings:

o When we try to use anycast address in the source address, use a (non-anycast) unicast address as the IPv6 source address, and attach home address option with anycast address. In ipngwg discussions, however, there seem to be a consensus that the home address option should have the same semantics as the source address in the IPv6 header, so we cannot put anycast address into the home address option.

5. Upper layer protocol issues

5.1. Use of UDP with anycast

Many of the UDP-based protocols use source and destination address pair to identify the traffic. One example would be DNS over UDP; most of the DNS client implementation checks if the source address of the reply is the same as the destination address of the query, in the fear of the fabricated reply from a bad guy.

```
query: client unicast (Cu) -> server unicast (Su*)
reply: server unicast (Su*) -> client unicast (Cu)

addresses marked with (*) must be equal.
```

If we use server's anycast address as the destination of the query, we cannot meet the requirement due to RFC2373 restriction (anycast address cannot be used as the source address of packets). Effectively, client will consider the reply is fabricated (hijack attempt), and drops the packet.

```
query: client unicast (Cu) -> server anycast (Sa)
reply: server unicast (Su) -> client unicast (Cu)
```

Note that, however, bad guys can still inject fabricated results to the client, even if the client checks the source address of the reply. The check does not improve security of the exchange at all.

If we check the existing protocol descriptions, in many cases, it is not possible to perform sanity checks against IP source address for UDP exchanges. Either they are not specified on the protocol documents, or

it is an implementation mistake to check the IP source address. For example, from RFC2181 [Elz, 1997] section 4.1, the source address of

response could be different from the destination address of the query if the destination address of the query is an anycast address. Therefore, we cannot check IP source address matches for UDP DNS packets. There is no wording available on the selection of source address, in TFTP protocol specification [Sollins, 1992] .

So, regarding to this issue, we conclude as follows:

- o There is an issue with the use of anycast addresses with UDP traffic due to the prohibitation against using anycast as a source address. New application protocols which use UDP and want to take advantage of anycast should take this into account by not identifying the response based on the source IP address, without compromising any security that might be present from verifying that the source IP address of the response is the same as the destination address in the request.

- o If you need to secure UDP protocol exchange, it is suggested to verify the authenticity of the reply, by using upper-layer security mechanisms like DNSSEC (note that we cannot use IPsec with anycast).

- o Existing UDP-based protocols that perform IP address verification between requests and responses, such as DNS lookups, are more problematic. If the transaction is covered by higher-layer security mechanisms it makes sense exploring protocol modifications in IETF working groups to relax or remove the the IP address checks. In other cases it makes sense to explore whether the IP address checks provide any real security in the appropriate IETF working groups.

5.2. Use of TCP with anycast

We cannot simply use anycast for TCP exchanges, as we identify a TCP connection by using address/port pair for the source/destination node. It is desired to implement some of the following, to enable the use of IPv6 anycast in TCP. Note, however, security requirement is rather complicated for the following protocol modifications. In particular, we are unsure about the relationship between the anycast address we contact first, and the unicast address we subsequently contact.

- o Define a TCP option which lets us to switch peer's address from IPv6 anycast address, to IPv6 unicast address. There have been a couple of proposals in the past.

- o Define an additional connection setup protocol that resolves IPv6 unicast address from IPv6 anycast address. We first resolve IPv6 unicast address using the new protocol, and then, make a TCP connection using the IPv6 unicast address. IPv6 node information query/reply [Crawford, 2002] could be used for this.

5.3. Use of SCTP with Anycast

SCTP [Stewart, 2000] is a bit more interesting. An SCTP endpoint is defined as:

> The logical sender/receiver of SCTP packets.
> On a multi-homed host, an SCTP endpoint is represented to its peers as a combination of a set of eligible destination transport addresses to which SCTP packets can be sent and a set of eligible source transport addresses from which SCTP packets can be received.
> All transport addresses used by an SCTP endpoint must use the same port number, but can use multiple IP addresses.

Therefore, it is legal to send packets to a unicast address of an SCTP peer endpoint, as long as the SCTP peer endpoint replies using a unicast address which is part of the association.

In summary, anycast should work with SCTP, as long as the SCTP endpoint contains a valid unicast address.

6. Summary

The draft tried to diagnose the limitation in currently-specified IPv6 anycast, and explored couple of ways to extend its use. Some of the proposed changes affects IPv6 anycast in general, some are useful in certain use of IPv6 anycast. To take advantage of anycast addresses, protocol designers would need to diagnose their requirements to anycast address, and introduce some of the tricks described in the draft.

Use of IPsec with anycast address still needs a great amount of analysis.

7. Security consideration

The document should introduce no new security issues.

When we use an anycast address to discover a server and then switch to unicast adddress for the server, upper-layer protocols need to make sure that the two addresses actually belong to the same node. Otherwise, there could be a chance for malicious nodes to hijack the communciation. One possible way to achieve this is to use public-key based authentication in the upper-layer protocol.

For secure anycast operation, we may need to enable security mechanisms in other protocols. For example, if we were to inject /128 routes from end hosts by using a routing protocol, we may need to configure the routing protocol to exchange routes securely, to prevent malicious parties from injecting bogus routes.

References

Hinden, 1998.

R. Hinden and S. Deering, "IP Version 6 Addressing Architecture" in RFC2373 (July 1998). ftp://ftp.isi.edu/in-notes/rfc2373.txt.

Partridge, 1993.
C. Partridge, T. Mendez, and W. Milliken, "Host Anycasting Service" in RFC1546 (November 1993). ftp://ftp.isi.edu/in-notes/rfc1546.txt.

Hardie, 2002.
T. Hardie, "Distributing Authoritative Name Servers via Shared Unicast Addresses" in RFC3258 (April 2002). ftp://ftp.isi.edu/in-notes/rfc3258.txt.

Haberman, 2002.
B. Haberman and D. Thaler, "Host-based Anycast using MLD" in draft-haberman-ipngwg-host-anycast-01.txt (May 2002). work in progress material.

Elz, 1997.
R. Elz and R. Bush, "Clarifications to the DNS Specification" in RFC2181 (July 1997). ftp://ftp.isi.edu/in-notes/rfc2181.txt.

Sollins, 1992.
K. Sollins, "The TFTP Protocol (Revision 2)" in RFC1350 (July 1992). ftp://ftp.isi.edu/in-notes/rfc1350.txt.

Crawford, 2002.
Matt Crawford, "IPv6 Node Information Queries" in draft-ietf-ipngwg-icmp-name-lookups-09.txt (May 2002). work in progress material.

Stewart, 2000.
R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream Control Transmission Protocol" in RFC2960 (October 2000). ftp://ftp.isi.edu/in-notes/rfc2960.txt.

Change history

individual submission, 00 -> 01
Improve security considerations section. Remove an invalid use of home address option from UDP section. Improve wording on IPsec.

individual submission, 01 -> 02
Split sections for current status analysis, and future protocol design suggestions.

02 -> 00
Distinguish RFC2373 anycast and BGP anycast. Ettikan's new address.

00 -> 01
Reflect IESG comments. BGP anycast is now called "pseudo anycast" for clarity. SCTP section contributed by John Loughney.

01 -> 02

Typo and grammar fixes from Richard Dawe. Many comments from Erik Nordmark, including: (1) use "shared-unicast address" isntead of "pseudo anycast", (2) remove mention of BGP as RFC3258 is not specifically tied to BGP, and (3) clarify the impact of host route injection. IIJ office have moved. Some additional notes wrt difference between shared-unicast approach.

Authors' addresses

Jun-ichiro itojun Hagino
Research Laboratory, Internet Initiative Japan Inc.
Jinbocho Mitsui Buildling, 1-105,
Kanda-Jinbocho, Chiyoda-ku, Tokyo 101-0051 JAPAN
Tel: +81-3-5205-6464
Fax: +81-3-5205-6465
E-mail: itojun@iijlab.net

Ettikan Kandasamy Karupiah
ASG Penang & Shannon Operations,
Intel Microelectronis (M) Sdn. Bhd.,
Bayan Lepas Free Trade Zone III,
Penang, Malaysia.
Tel: +60-4-859-2591
Fax: +60-4-859-7899
Email: ettikan.kandasamy.karuppiah@intel.com