Multi6 Working Group
Internet-Draft
Expires: December 22, 2003

I. van Beijnum
June 23, 2003

Two Prefixes in One Address
draft-van-beijnum-multi6-2pi1a-00

Status of this Memo

Copyright Notice

Abstract

This memo presents a possible solution to the multihoming in IPv6 problem. It borrows from earlier 8+8 and GSE proposals but it diverts from these approaches in order to be more readily deployable.

A source host that wants to initiate communications looks up the IPv6 addresses that will function as the transport-layer identifier and the IP-level locators for the remote end in the DNS and collapses both of its own addresses into a single one. This makes it possible to communicate without prior negotiation, but without opening the door to trivial identity theft that must be repaired by cryptographic means.

1. Clients and servers

Note that the use of the words "client" and "server" indicate the role a host fulfills at a particular time communication with a particular correspondent. A single host can operate as both a client and a server (for different sessions) concurrently.

2. Client to multihomed server

When a client connects to a multihomed server, it looks up all the addresses for the server in the DNS. The DNS returns one address that will be used as an identifier (i.e., this address will be presented to upper layers, even if the actual address used for transmitting and receiving packets changes over the course of a session) and two or more addresses that will be used to locate the server.

The identifier address may or may not do double duty as a locator address. The wisdom of using one address both as a multihomed identifier and a generic IPv6 is under debate, so this shouldn't be the default.

The different locator addresses may be bound to the same physical interface, or to different interfaces or a combination. For instance, a host with two network interfaces could have a single locator address for each interface, while a host with one network interface would have two locator addresses tied to this one interface. A host with two interfaces could also have two addresses for each interface, for a total of four, if there are compelling reasons for this.

The client sets up mapping state that allows incoming packets from one of the locator addresses to be mapped back to the identifier address, and for outgoing packets the identifier address to be mapped to one of the locators.

Note that despite the fact that this happens at session setup time, this mechanism works at the IP level. So if the mapping has been set up it is re-used on subsequent sessions that are directed to the same identifier address. The mapping state is flushed when there haven't been any active higher layer sessions for some time. Since removing the mapping state will break all higher layer sessions that are still active, and hinder applications that may want to set up new sessions to systems they have communicated with earlier, the system must not

be too aggressive in this regard.

The client may implement several heuristics to determine wich locator address should be used when transmitting packets. If one of the heuristics is the use of ICMP packets to check reachability, the number of those may not exceed one per minute unless the client has positive knowledge the server is prepared to handle a larger number.

In the absense of better knowledge, the client is expected to return packets to the locator address last used by the server. However, since this locator may become unreachable for the client at some point, the client must be prepared to make its own judgement of the reachability of this locator and switch to an alternative locator if

this is warranted.

3. Multihomed server to client

When a multihomed server receives a packet on one of its locator addresses, it maps this packet to its identifier address and continues to process the packet as if it had been addressed to the identifier address.

In order for the multihomed server to recognize whether a packet was sent to a locator address by a multihoming-aware client, or to a generic address by a non-multihoming-aware client, locator address may not be the same as regular addresses that are published using AAAA records. The identifier address may be the same as one of those addresses, however. Note that in this case, the server must keep per-peer-address state in order to know whether the correspondent host is multihoming-capable.

When a multihomed server transmits a packet back to a multihoming-capable client it maps the identifier address to one of the locator addresses. The server may implement several heuristics to determine wich locator address should be used when transmitting packets. If one of the heuristics is the use of ICMP packets to check

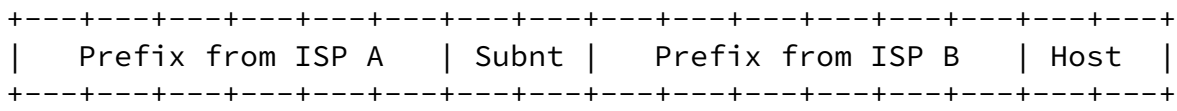
reachability, the number of those may not exceed one per minute unless the server has positive knowledge the client is prepared to handle a larger number.

A multihomed server is not required to keep per-correspondent state about which of its locators is best used as a source address to reach a certain correspondent. A server may make this determination based on the contents of the routing table, or it may select a single locator to be used for all multihoming-aware correspondents. The latter isn't advised as it provides only partial multihoming benefits.

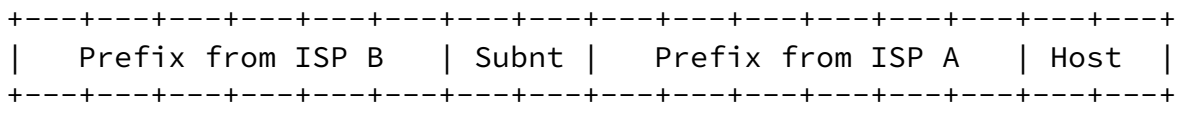
4. Multihomed client

In order to avoid the use of discovery or negotiation mechanisms when setting up sessions to multihomed servers, multihomed clients encode two prefixes (and host- and subnet identifiers) into two equivalent 128 bit IPv6 addresses that are to be used as locators. These addresses are structured as follows:

Locator 1:



Locator 2:



Locator 2 is an alternate form of locator 1 and vice versa.

One of the locators is also used as the identifier. In order to determine which, two 48-bit values are derived from a locator. The first value (V1) has the bits 32 - 47 as the first 16 bits, bits 16 - 31 as the second 16 bits and bits 0 - 15 as the last 16 bits. The second value (V2) has bits 96 - 111 as the first 16 bits, bits 80 - 95 as the second 16 bits and bits 64 - 79 as the last 16 bits. V1 and V2 are now compared, as are the 16 bit subnet and host values. There are now four possibilities:

1. subnet >= host and V1 >= V2: the current form of the address is the identifier

2. subnet >= host and V1 < V2: the alternate form of the address is the identifier

3. subnet < host and V1 >= V2: the alternate form of the address is the identifier

4. subnet < host and V1 < V2: the current form of the address is the identifier

When setting up a session, both the multihomed client and the server set up the necessary mapping state to map the identifier to the alternate locator and from the alternate locator back to the identifier, when required. The subsequent handling of transmitting and receiving packets is equivalent to the multihomed server case.

5. Single homed, but multihoming-aware client to multihomed server

When a multihoming-aware, but not multihomed client wants to communicate with a multihomed server, it must avoid being classified as an actual multihomed client. This is accomplished by using a value in bits 64 - 111 of its address that is easily recognizable as not being a valid 48 bit prefix of a routable IPv6 unicast address. At

present, setting bits 64 - 66 to any other binary value than 001 will accomplish this, but single homed, multihoming-aware clients are advised to use the value 0 in bits 64 - 71 in order to avoid being mistaken for a multihomed client when the global unicast address space is increased.

Alternatively, the client may create an address for itself the same way a multihomed client would, but with the same prefix in both bits 0 - 47 and bits 64 - 111.

An implementation must support setting the entire address by user-configurable means such as manual configuration or DHCP.

6. Address discovery through the DNS

In order for multihoming-aware clients to set up sessions to multihomed servers, they must connect to the right locator addresses and create mappings using the identifier address, but these addresses must remain invisible to legacy IPv6 hosts; those should connect to different addresses for which no multihoming processing is done. This is accomplished by publishing identifiers and locators in the DNS using a new resource record type. The name "A8" seems appropriate for this purpose.

The exact structure of the A8 RR will be determined at a later date. The A8 record, or the full set of A8 records, for a host should provide the following information about the set of addresses the host wants to use in multihoming:

1. Whether the address is an identifier only, and identifier and locator, or a locator only.

2. Whether a locator may always be used or it is just a backup.

3. A value that indicates the degree of preference given to a locator. All else being equal, a locator with a twice as high preference value should receive twice the number of connection attempts and/or packets during established sessions as another locator with half that preference value.

Note that this document doesn't provide for multihomed communication without involvement from the DNS. Alternative mechanisms to discover the locators based on an identifier so the identifier can be used by applications to set up sessions the same way a literal IP address can be used may be developed later.

7. Source address handling

It is very important to select the right source address for every outgoing packet. Not only will return traffic often be directed at this address, using an unfortunate source address may also cause the packet to be lost due to egress or ingress filtering. It is strongly encouraged that hosts implement heuristics to determine that a source address doesn't work and that a different one should be used. An obvious example of such an heuristic is listening for ICMP administratively prohibited messages that are generated by the site's edge routers upon egress filtering. There are some security issues with this, but the unnecessary triggering of source address switching isn't fatal, while not switching source addresses when the wrong one is used often is (in the presence of ingress filtering by an ISP).

Additionaly, routers may route packets based on the source address. So when a packet contains a source address with a prefix from ISP A, the packet is forwarded to ISP A, even though the routing table may point to a different ISP as the best or working route for this destination. This feature works well with smart hosts that know how to manipulate the source address for optimum connectivity.

An alternative approach is to rewrite the address to conform with the ISP's ingress filtering when the packet leaves the site. Hosts may signal their desire to have addresses rewritten by setting one of two special source prefixes.

When a router encounters the first special prefix in a source address, it rewrites the upper 48 bits of the source address with the prefix received from the ISP the packet is fowarded to. This special prefix is to be used by multihomed servers that have identical lower 80 bits in all their locator addresses.

The second special prefix signals the router to perform the action indicated by the first special prefix, but also to rewrite bits 64 - 111 with the prefix received from the secondary ISP. Use of this special prefix is appropriate for multihomed clients.

Both servers and clients must support manual disabling of the rewriting feature and should automatically recover from a situation where rewriting is requested but not honored.

8. Stateless autoconfiguration

The requirements for bits 64 - 111 set forth in this document don't necessarily break stateless autoconfiguration, but they do limit the part of the IPv6 address that identifies a host to only 16 bits. Such a small value can't be realistically be expected to be chosen by the host without the strong possibility of address conflicts. This means autoconfiguration can't be depended upon to select the same address for a host repeatedly, which probably means servers must be manually configured with their addresses. For clients this shouldn't be much of an issue.

Both multihomed clients and servers may obtain additional addresses that are not used in multihoming using traditional stateless autoconfiguration or DHCP.

9. IANA considerations

The IANA is requested to assign a new resource record type code for A8 information in the DNS and two 48 bit values that can be used as a prefix in source addresses to signal routers that the source address should be rewritten upon egress.

10. Security considerations

This multihoming mechanism doesn't provide any more security than regular IPv6 or IPv4. As such, the use of additional security measures such as TLS, IPsec and/or DNSSEC is highly recommended for even slightly sensitive applications.

In order to be compatible with IPsec AH, both ends must either always first do IPsec processing and then multihoming processing, or first multihoming processing and then IPsec processing. A situation where one host does the former and the other host does the latter can't work. The implications of choosing the processing order are unknown at this time and should be subject of further study. In the mean time, implementations should by default perform all IPsec processing (not limited to AH) first and the required multihoming processing after that, but it should be configurable to do this the other way around.

This mechanism enables communication over different addresses than the addresses applications see. This breaks most security mechanisms that operate on addresses. It is recommended that services that depend on address-based access restrictions not be

multihoming-enabled by not binding the services to a multihomed identifier address and/or filtering the service on both the identifier and locator addresses.

In order to be able to enable multihoming on clients, the client must not run any applications that restrict access to certain server addresses. However, it shouldn't be a problem to have these same restrictions enforced by mechanisms working at the IP layer (i.e., a firewall), as long as these mechanisms take all possible addresses into account.

Hosts acting as multihomed clients can trick servers into sending traffic to third party networks by constructing a double prefix address with the target network's prefix in bits 64 - 111 and then ignore incoming packets from the server. However, if the original sender's ISP employs ingress filtering, this should be easy to track down as bits 64 - 111 in the packet that ends up at the target network points back to the original source network. Still, transport protocols and applications are encouraged to scale back their bandwidth use when there is a switch to a new destination locator. In TCP this event could trigger slow start.

Author's Address

Iljitsch van Beijnum
Karel Roosstraat 95
2571 BG The Hague
NL

email: iljitsch@muada.com

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the

IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.