Network Working Group
Request for Comments: 4219
Category: Informational

E. Lear
Cisco Systems

October 2005

Things Multihoming in IPv6 (MULTI6) Developers Should Think About

Status of this Memo

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

Copyright Notice

Abstract

This document specifies a set of questions that authors should be prepared to answer as part of a solution to multihoming with IPv6. The questions do not assume that multihoming is the only problem of interest, nor do they demand a more general solution.

Table of Contents

1. Introduction

At the time of this writing there are quite a number of proposed solutions to the problem of multihoming within IPv6, and related problems such as the locator/identifier split.

This document contains several sets of questions that attempt to focus these solutions on operational problems. This document does not suggest methods to solve the problem. Rather, we simply want to ensure that while solving a problem the medicine is not worse than the cure. We focus on practical operational problems that both single-homed and multihomed deployments may face.

It is the hope of the author that perhaps the authors of other proposed solutions will use this document to identify gaps in their solutions, and cooperate to close those gaps.

1.1. Reading this Document

The questions are organized along the following lines:

- changes to on the wire behavior;
- routing system interactions;
- identifier/mapping split;
- application concerns and backward compatibility;
- name service interactions;
- legal concerns; and
- security considerations.

In reality many questions cut across all of these concerns. For instance, the identifier / locator split has substantial application implications, and every area has security considerations.

Unless it is blatantly obvious, each question contains some reasoning as to why it is being asked. It is envisioned that no solution will answer every question with completeness, but that there will be tradeoffs to be made. The answers by the various designers of solutions will hopefully shed some light on which tradeoffs we as a community wish to make.

It would seem silly for people who have written detailed answers to these questions to have to repeat the exercise. Therefore, a simple reference to existing documents will suffice, so long as the answer is complete. If it is not complete, then feel free to reference it and add what text is necessary to make the answer complete.

This document presumes a familiarity with RFC 3582 [2], and does not attempt to repeat the requirements work gathered there.

2. On the Wire Behavior

2.1. How will your solution solve the multihoming problem?

Please scope the problem you are attempting to solve and what you are not attempting to solve.

2.2. At what layer is your solution applied, and how?

Is it applied in every packet? If so, what fields are used?

2.3. Why is the layer you chose the correct one?

Each layer has its benefits and tradeoffs. For instance, transport layer solutions would require that EVERY transport be modified, while IP layer solutions may entail expansion of the packet or a change to the pseudo-header (thus requiring changes to the transport layer).

2.4. Does your solution address mobility?

If so, how are rendezvous handled? Can your solution handle both locators changing at the same time? If so, please explain. Should it? If not, how will your solution interact with MOBILEIP-V6 [3] (MIPv6)

2.5. Does your solution expand the size of an IP packet?

Expanding the size of an IP packet may cause excessive fragmentation in some circumstances.

2.6. Will your solution add additional latency?

Latency is an important factor in many applications, including voice. Any substantial amount of additional latency, including session initiation would be highly undesirable.

2.7. Can multihoming capabilities be negotiated end-to-end during a connection?

If the proposal introduces additional overhead, can the information be somehow piggybacked on messages that are already used? This would be useful in order to keep connection setup constant. Please also indicate any drawbacks that might apply due to this piggybacking.

2.8. Do you change the way fragmenting is handled?

If you use a shim approach, do you fragment above or below the shim? How are fragments identified, so that they can be reassembled? If you use any additional names, do they need to be associated with fragments? If not, why not? If so, how will that happen?

2.9. Are there any layer 2 implications to your proposal?

While IPv6 has a simplified approach to layer 2, perhaps you unsimplified it. If so, please provide details.

3. Identifiers and Locators

3.1. Uniqueness

3.2. Does your solution provide for a split between identifiers and locators?

3.3. What is the lifetime of a binding from an identifier to a locator?

3.4. How is the binding updated?

Will transport connections remain up when new paths become available or when old ones become unavailable? How does the end node discover these events?

3.5. How does a host know its identity?

If you are establishing a new identity, how does the host learn it?

3.6. Can a host have multiple identifiers?

If so, how does an application choose an identity?

3.7. If you have separate locators and identifiers, how will they be mapped?

Does the mapping work in both directions? How would someone debugging a network determine which end stations are involved?

3.8. Does your solution create an alternate "DNS-like" service?

If you use mechanisms other than DNS, first, why was DNS not appropriate? Also, how will this other mechanism interact with DNS? What are its scaling properties?

3.9. Please describe authentication/authorization

How are bindings authenticated and authorized. What technology do you build on for this mechanism?

3.10. Is your mechanism hierarchical?

Please describe the hierarchical breakdown.

3.11. Middlebox interactions

What are the implications for firewalls? What are the interactions with Network Address Translation (NAT)? What are the interactions with web caches? What complications are introduced with your solution? For instance, are there implication for ingress filters? If so, what are they?

When considering this question, there are really two issues. First, will middleboxes impede your solution by rewriting headers in some way, as NATs do for IP addresses, and web caches do at higher layers? Second, is there a way in which middleboxes are actually part of your solution? In particular, are they required? This would be the case, for example, with Generalized Structure Element (GSE) (8+8).

3.12. Are there any implications for scoped addressing?

Please see RFC 3513 [1]. How does your mechanism interact with multicast?

How does your solution interact with link-local addressing

How does your solution interact with Son-Of-Sitelocal (whatever that will be)?

4. Routing System Interactions

4.1. Does your solution change existing aggregation methods?

Routing on the Internet scales today because hosts and networks can be aggregated into a relatively small number of entries. Does your solution change the way in which route aggregation occurs?

4.2. Does the solution solve traffic engineering requirements?

One of the significant goals of IPv4 multihoming solutions has been the ability to perform traffic engineering based on appropriately adjusting the BGP advertisements. If the prefixes used by such sites was be aggregated (particularly beyond the site"s border), the site"s ability to perform traffic engineering would be diminished.

4.3. Does the solution offer ways for the site to manage its traffic flows?

If so, how? Is this controllable on a per-host basis, or on a per-site basis?

4.4. If you introduce any new name spaces, do they require aggregation?

Is it desirable or required that, in order to scale distribution of any mapping information, an aggregation method be introduced?

4.5. Does your solution interact with Autonomous System numbering?

If your solution involves address prefixes distributed using BGP4+, does it interact with the use of AS numbers and, if so, how? Will it require additional AS numbers?

4.6. Are there any changes to ICMP error semantics?

Do you create new codes? If so, why and what do they mean? Will a host that is not aware of your scheme see them?

5. Name Service Interactions

5.1. Please explain the relationship of your solution to DNS

If your solution uses new names for identifiers, please explain what mappings are defined, and how they are performed?

If there are any additional administrative requirements, such as new zones or RR types to manage, please explain them as well.

5.2. Please explain interactions with "2-faced" DNS

2-faced DNS is used so that hosts behind a NAT get one address for internal hosts, while hosts outside the NAT get another. Similar mechanisms are used for application layer gateways, such as SOCKS [5].

5.3. Does your solution require centralized registration?

For instance, if you are using the DNS, what will be the top level domain, and how will the name space distribute through it?

Also, how will the centralized registration be managed?

5.4. Have you checked for DNS circular dependencies?

If you are using the DNS in your solution, is it required for connectivity? What happens if the DNS fails? Can communication between the DNS resolver and the server make use of your solution? What about between the application and the resolver?

5.5. What if a DNS server itself is multihomed?

If a link fails or a service is dropped, how will it impact DNS? Again, are there any dependency loops? Perhaps diagram out your dependencies to make sure.

5.6. What additional load will be placed on DNS servers?

Can the load be distributed? Remember that DNS is optimized for READ operations.

5.7. Any upstream provider support required?

If so, please describe. For instance, currently reverse mappings are delegated down from upstream providers. How would this work with your solution?

5.8. How do you debug connectivity?

How would tools like ping and traceroute need to be enhanced? What additional tools would prove useful or necessary? For instance, if there is an id/locator split, can one ping an identifier? If so, what gets returned?

6. Application Concerns and Backward Compatibility

6.1. What application/API changes are needed?

Will old code work with the new mechanism? For instance, what about code that uses gethostbyname()?

Will getaddrinfo() need to change?

What about other API calls?

There are several possible approaches. For instance, a multihoming service could attempt to require no changes to the API, in which case it is possible that IP addresses might become opaque blobs that work with the API, but might break operational assumptions that applications make about addresses. Consider the case of a web server that wants to log IP addresses. How will it accomplish this task?

Another approach is to have some sort of compatibility library for legacy applications, but also provide a richer calling interface for transparency.

Yet another approach would be to only provide the new functionality to those applications that make use of a new calling interface.

One useful exercise would be to provide code fragments that demonstrate any API changes.

6.2. Is this solution backward compatible with "old" IP version 6?

Can it be deployed incrementally? Please describe how.

Does your solution impose requirements on non-multihomed/non-mobile hosts?

What happens if someone plugs in a normal IPv6 node?

6.3. Is your solution backward compatible with IPv4?

How will your mechanism interact with 6to4 gateways and IPv4 hosts?

6.4. Can IPv4 devices take advantage of this solution?

Can the same mechanism somehow be used on the existing network? N.B. this is NOT a primary consideration, but perhaps a side benefit of a particular solution.

6.5. What is the impact of your solution on different types of sites?

What will the impact of your solution be on the following types of systems?

- single homed sites
- small multihomed sites
- large multihomed sites
- ad-hoc sites
- short lived connections (think aggregator wireless ISPs)

In particular, consider ongoing administration, renumbering events, and mobile work forces.

6.6. How will your solution interact with other middleboxes?

6.7. Referrals

How will your solution handle referrals, such as those within FTP or various conferencing or other peer to peer systems?

Referrals exist within various other protocols, such as so-called "peer to peer" applications. Note that referrals might suffer three types of failure:

firewall and NAT - Is there failure just as what FTP active mode experiences today with relatively simple firewalls?

time-based - Is there something ephemeral about the nature of the solution that might cause a referral (such as a URL) to fail over time, more so than what we have today?

location-based - If the binding varies based on where the parties are in the network, and if one moves, will they no longer be able to find each other?

6.8. Demonstrate use with a real life complex application

Provide a detailed walk-through of SIP + Real Time Streaming Protocol (SIP+RTSP) when one or several of the peers are multihomed. How does your analysis change when encrypted RTSP is used or when SIP with S/MIME end-to-end (e2e) signalling is used?

7. Legal Concerns

Are you introducing a namespace that might involve mnemonics? Doing so might introduce trademark concerns. If so, how do you plan to address such concerns?

Are there any organizations required to manage a new name space? If so, please describe what they are and how the method will scale.

8. Security Considerations

How secure should a multi6 solution be? This is a reasonable question for each solution to answer. The author opines that the worst case should be no worse than what we have today. For example, would a multi6 solution open up a host, on either end of a communication, to a time-based attack? Any such risks should be clearly stated by the authors. Considerable time should be spent on threat analysis. Please see [4] for more details.

As IP addresses can often be tied to individuals, are there any auditing or privacy concerns introduced by your solution?

9. Acknowledgements

The author wishes to acknowledge everyone in the multi6 group and elsewhere that is putting forward proposals. It is easy to ask questions like the ones found in this document. It is quite a bit harder to develop running code to answer them. Marcelo Bagnulo, Kurt Erik Lindqvist, Joe Touch, Patrik Faltstrom, Brian Carpenter, and Iljitsch van Beijnum provided input to this document.

10. References

10.1. Normative References

[1] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.

[2] Abley, J., Black, B., and V. Gill, "Goals for IPv6 Site-Multihoming Architectures", RFC 3582, August 2003.

[3] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[4] Nordmark, E., "Threats Relating to IPv6 Multihoming Solutions", RFC 4218, October 2005.

10.2. Informative References

[5] Kitamura, H., "A SOCKS-based IPv6/IPv4 Gateway Mechanism", RFC 3089, April 2001.

Author's Address

Eliot Lear
Cisco Systems GmbH
Glatt-com, 2nd Floor
CH-8301 Glattzentrum ZH
Switzerland

EMail: lear@cisco.com

Full Copyright Statement

Intellectual Property

Acknowledgement

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