IPv6 Maintenance Working Group
Internet-Draft
Intended status: Informational
Expires: August 2, 2008

A. Matsumoto
T. Fujisaki
NTT
R. Hiromi
K. Kanayama
Intec Netcore
June 18, 2008

Solution approaches for address-selection problems
draft-ietf-6man-addr-select-sol-01.txt

Abstract

In response to address selection problem statement and requirement documents, this document describes approaches to solutions and evaluates proposed solution mechanisms in line with requirements. It also examines the applicability of each solution mechanism from the viewpoint of practical application.

Table of Contents

1. Introduction 3
2. Solution Design 3
  2.1. Proactive approaches 3
  2.2. Reactive approaches 4
3. Solution approaches 4
  3.1. Obtain all information prior to communication (Most Proactive) 4
    3.1.1. Overview 4
    3.1.2. Requirements correspondence analysis 5
    3.1.3. Other issues 6
  3.2. Routing system assistance for address selection (Proactive) 6
    3.2.1. Overview 6
    3.2.2. Requirements correspondence analysis 6
    3.2.3. Other issues 7
  3.3. Trial-and-error approach (Reactive) 7
    3.3.1. Overview 7
    3.3.2. Requirement correspondence analysis 8
    3.3.3. Other issues 9
  3.4. All-by-oneself approach (Most Reactive) 9
    3.4.1. Overview 9
    3.4.2. Requirement correspondence analysis 9
    3.4.3. Other issues 10
4. Applicability Comparison 11
  4.1. Dynamic-static and managed-unmanaged 11
  4.2. Deployment Difficulty 12
5. Security Considerations 13
6. IANA Considerations 13
7. Conclusions 13
8. References 13
  8.1. Normative References 13
  8.2. Informative References 14
Authors' Addresses 14
Intellectual Property and Copyright Statements 16

1. Introduction

One physical network can have multiple logical networks. In that case, an end-host has multiple IP addresses. (e.g, in the IPv4-IPv6 dual-stack environment, in a site that uses both ULA [RFC4193] and global scope addresses or in a site connected to multiple upstream IPv6 networks.) For such a host, RFC 3484 [RFC3484] defines default address-selection rules for the source and destination addresses.

Today, the RFC 3484 mechanism is widely implemented in major OSs. However, many people, including us, have found that in many sites the default address-selection rules are not appropriate for the network structure. PS [I-D.ietf-v6ops-addr-select-ps] lists problematic cases that resulted from incorrect address selection.

Though RFC 3484 made the address-selection behavior of a host configurable, typical users cannot make use of that because of the complexity of the mechanism and their lack of knowledge about their network topologies. Therefore, an address-selection autoconfiguration mechanism is necessary, especially for the unmanaged hosts of typical users.

REQ [I-D.ietf-v6ops-addr-select-req] document enumerates requirements for address-selection mechanisms that enable hosts to perform appropriate address selection automatically.

In the IETF mailing lists and in the internet-draft archives, some mechanisms for solving address-selection problems have already been proposed. This document describes possible design approaches for solving address selection problems. After that, we try to put together an overview as well as an analysis of how well the method corresponds with the requirements.

2. Solution Design

There are two types of approaches that can control the behavior of hosts in terms of the selection of destination address and source address. The first type is proactive, where the host is given the necessary information to decide the destination and source addresses before the beginning of transmission. The other type is reactive, where the host decides appropriate destination address and source addresses through trial and error.

2.1. Proactive approaches

There can be two types of proactive approaches. One gives hosts all the information for selecting destination and address and source addresses beforehand. Under some circumstances, a lot of information could be stored in hosts.

The other type informs hosts about which prefixes should be used in the source address for the different destinations every time before

starting each connection.

2.2. Reactive approaches

In these approaches, the host does not have initial information for address selection. It will try using different pairs of destination and source addresses until the connection is established. When an outage occurs, the host must detect it and try again with a new pair of destination address and source address. Some reactive solutions may use some kind of control message that enables the gateway to indicate the outage.

3. Solution approaches

This section describes the evaluation of the four approaches to finding solutions. The evaluation value has a 3-point scale for each of 8 requirements in the requirement document. The meaning of the points is as follows.

1 : bad
2 : fair
3 : good

About "Effectiveness", the score is 1 if the approach solves no problematic cases described in the problem statement document, 2 if it can handle at least one, and 3 if it solves every case.

3.1. Obtain all information prior to communication (Most Proactive)

3.1.1. Overview

In this approach, a host obtains everything needed to select addresses at once prior to communication. A host receives all policy information from a server beforehand. It then sets up communication whenever it wants to. DHCPv6 and RA fall into this category as known protocols. There is a reference document [I-D.fujisaki-dhc-addr-select-opt] in which DHCPv6 is used for this purpose.

This approach can take advantage of the RFC 3484 Policy Table, which is already widely deployed. By distributing policies for the Policy Table, you can auto-configure a host's address selection policy.

3.1.2. Requirements correspondence analysis

1. Effectiveness: 3
   It can support all cases by using the policy table.

2. Timing: 3
   All information for communication is in a host in advance. Communication starts at once when it is necessary and the communication process refers to local policy information, so it exhibits good usability. Moreover, this leads to fewer overheads than per-connection mechanisms.

3. Dynamic update: 3
   Though it depends on what protocol is used to distribute the policies, some mechanisms support information updates from the server. Moreover, it is difficult to support dynamic network changes and real-time updates in some specific protocols.

4. Node-specific behavior: 3
   For distribution to individual hosts in the same segment, DHCPv6 can be used.

5. Application-specific behavior: 2
   The policy table itself doesn't support application-specific address selection. It can be done using the address selection API. [RFC5014]

6. Multiple interfaces: 2
   If all interfaces belong to the same administration domain, it is possible for the address-selection information to be controlled by administrators of that domain. However, if not, routing information and address selection policies are not always equivalent between domains, and it is not possible to handle them.

7. Central control: 3
   It can support central control. A site administrator or a service provider can determine users' policy tables.

8. Route selection: 2
   Current solutions, such as DHCPv6 and RA, do not have a mechanism for cooperation with routing protocols. This could be done with other techniques such as "source address based routing" or "Default Router Preferences and More-Specific Routes" RFC4191. [RFC4191]

9. Compatibility with RFC 3493: 3
This approach is able to coexist with any kind of applications (socket API). In detail, any types of function such as getaddrinfo(), getsockname(), connect() or other typical system calls will work without alterations if this mechanism is applied to a host.

10. Compatibility and Interoperability with RFC 3484: 3
The basic idea of this approach has a compatibility with RFC3484. This approach make RFC3484 policy table configurable to put some hints related with it's individual network case.

11. Security: 2
This approach has a weakness on hijacking.
A combination of Layer 2 securing techniques and this mechanism will be able to be effective against security concerns.
DHCP and RA protocol have own security measures and they also protect from them.

3.1.3. Other issues

- The traffic volume will be equal to the number of policies.

- Hosts and servers need to support this function.

3.2. Routing system assistance for address selection (Proactive)

3.2.1. Overview

Fred Baker proposed this approach. A host asks the DMZ routers or the local router which is the best pair of source and destination addresses when the host has a set of addresses A and the destination host has a set of addresses B. Then, the host uses the policy provided by the server/routing system as a guide in applying the response. He also proposed a mechanism that utilizes the ICMP error message to change the source address of the existing session. This point resembles Section 3.3 3484update mechanism, so the following evaluation is based on only the first part of his proposal.

3.2.2. Requirements correspondence analysis

1. Effectiveness: 3
A routing system knows about information about paths toward the destination and information about which of their prefixes should be used. Therefore, it is possible to select an appropriate pair of source and destination addresses.

2. Timing: 3
   A routing system always has up-to-date routing information, so it will be possible to provide suitable information whenever requests come. However, the amount of information that the system must handle is huge, so there will be cases where it takes time to answer the request because appropriate information must be retrieved from a huge database.
   If any server or routing trouble occurs, the requester cannot get the answer, and address selection will fail. This point is the same in all systems that depend on other servers.

3. Dynamic update: 3
   A routing system always has up-to-date routing information, and it will be possible to provide suitable information whenever requests come.

4. Node-specific behavior: 3
   Node-specific information can be provided if a server recognizes individual nodes.

5. Application-specific behavior: 2
   A routing system does not care about applications. Using address selection API allows nodes to behave in an application-specific way.

6. Multiple Interfaces: 2
   If all interfaces belong to the same administration domain, it is possible for the address-selection information to be controlled by administrators of that domain. However, if not, routing information and address selection policies are not always equivalent between domains, and it is not possible to handle them.

7. Central Control: 3
   It is possible to provide address selection information from one source. However, because routing information changes dynamically, it is difficult to control it in the way that administrators want.

8. Route Selection: 3
   It is possible to give next-hop selection advice to a host. As routers have routing information, it would seem to be easier for routers to implement this function.

9. Compatibility with RFC 3493: 3
   This approach is able to coexist with any kind of applications (socket API). In detail, any types of function such as getaddrinfo(), getsockname(), connect() or other typical system calls will work without alterations if this mechanism is applied to a host.
   In the existing TCP/IP protocol stack implementation, destination address selection is mainly the role of the application and not that of the kernel unlike source address selection. Therefore,

implementing this model without affecting applications is not so easy.

10. Compatibility and Interoperability with RFC 3484: 2
Currently it just proposed and there is no implementation.
Therefore, it depends on how to implement with this requirement
and it can be coexistence with RFC3484.

11. Security: 2
This approach has a weakness on hijacking.
Currently it just proposed and there is no implementation.
Therefore, it depends on how to define security protection
mechanism and how to implement it.

3.2.3. Other issues

- A host must consult the routing system every time it starts a connection if the host does not have address selection information for the destination host or if the information lifetime has expired. This could be a possible scalability problem.

- The existing host/router OS implementation must be changed a lot.

3.3. Trial-and-error approach (Reactive)

3.3.1. Overview

M. Bagnulo proposed a new address selection method in his draft. When the host notices that a network failure has occurred or packets have been dropped somewhere in the network by, for example, an ingress filter, the host changes the source address of the connection to another source address.

Hosts may use some kinds of error messages, e.g, ICMP error messages, from a network to detect that sent packets did not reach the destination quickly.

The host stores a cache of address selection information so that the host can select an appropriate source address for new connections.

For source address selection by the application that initiated a communication, this method provides an ordered list of source addresses for the destination address to the application.

3.3.2. Requirement correspondence analysis

1. Effectiveness: 2
This solution is not effective for the problem about IPv4 or IPv6 prioritization described in the problem statement document.

2. Timing: 2
   Hosts should try to use all the available source addresses to the maximum to find an appropriate source address. If the host tries the next source address after the previous trial using another source address has failed, it may take a long time because this trial-and-error process lasts until the connection succeeds. If the host does not use an error message from a network to detect a connection error, it takes longer to wait for a time-out.

3. Dynamic update: 3
   If hosts detect a connection failure using some reliable mechanism, such like TCP or ICMP error messages, a connection failure caused by some changes in the network will be detected immediately by the hosts.

4. Node-specific behavior: 2
   This solution does not have a function for node-specific behavior. However, it is not impossible to implement by setting a packet filter for each node at the gateways through which the packets from nodes pass.

5. Application-specific behavior: 2
   This solution does not have a function for application-specific behavior. However, the mechanism of this approach does not exclude address selection by each application.

6. Multiple interfaces: 3
   If the protocol-stack or an application supports interface selection and it tries to establish a connection by changing addresses and also interfaces, it can find a working combination of addresses and interface.

7. Central control: 2
   The only way that a central administrator has to control the node behavior is switching a filter on/off on the network. Therefore, advanced control such as traffic engineering and QoS is almost impossible.

8. Route Selection: 2
   This solution does not refer to next-hop selection for the transmission of a packet. So, it should be used with some routing function such as RFC 4191 on the nodes.

9. Compatibility with RFC 3493: 1
This approach has possibility to interfere with coexistence with applications(socket APIs). The returen value of functions would be changed for its meaning. A case is suspected that the return value of connect() system call will change its state from "non-blocking" to "blocking" and this will bring alteration to the application behaviours. Because of this suspicion, this approach scores 1.

10. Compatibility and Interoperability with RFC 3484: 2
It depends on how to implement with this requirement. But there will be possible conflict which a result will be overwrite the 3484 policy table without any permission of domain administrators.

11. Security: 2
This approach has a weakness on hijacking.
Currently it just proposed and there is no implementation.
Therefore, it depends on how to define security protection mechanism and how to implement it.

3.3.3. Other issues

- A host must learn address selection information for each destination host. Therefore, the number of cache entries could be very large.

- The existing host/router OS implementation must be changed a lot. In particular, changing the source address of the existing connection is not so easy and has a big impact on the existing TCP/IP protocol stack implementation.

3.4. All-by-oneself approach (Most Reactive)

3.4.1. Overview

shim6 was designed for site-multihoming. This mechanism introduces a new address selection method for session initiation and session survivability; it is documented in two drafts:
[I-D.ietf-shim6-locator-pair-selection] and
[I-D.ietf-shim6-failure-detection].

The shim6 host detects connection failures and changes the destination and source addresses during the session.

In this document, we focus on address selection issues in the connection initiation phase of shim6 and not on any other functions, such as session survivability.

3.4.2. Requirement correspondence analysis

1. Effectiveness: 2
   This solution is not effective for the problem about IPv4 or IPv6 prioritization described in the problem statement document.

2. Timing: 2
   Hosts should try to use all the available source addresses to the maximum to find an appropriate source address. If the host tries the next source address after the previous trial using another source address has failed, it may take a long time because this trial-and-error process lasts until the connection succeeds. If the host does not use error messages from a network to detect a connection error, it takes longer to wait for a time-out.

3. Dynamic update: 3
   It can reflect dynamically changing network, as far as it always tries all possible addresses and next-hops.

4. Node-specific behavior: 2
   This solution does not have a function for node-specific behavior. However, it is not impossible to implement by setting a packet filter for each node on the gateways through which the packets from nodes pass.

5. Application-specific behavior: 2
   The use of shim6 API [I-D.ietf-shim6-multihome-shim-api] allows applications to override address selection behavior.

6. Multiple interfaces: 3
   If the protocol-stack supports interface selection and it tries to establish a connection by changing addresses and also interfaces, it can find a working combination of addresses and interface.

7. Central control: 2
   The only way that a central administrator has to control the node behavior is switching a filter on/off on the network. Therefore, advanced control such as traffic engineering and QoS is almost impossible.

8. Route Selection: 2
   This solution does not refer to next-hop selection for the transmission of a packet. Therefore, it should be used with some routing function such as RFC 4191 on the nodes.

9. Compatibility with RFC 3493: 3
   This approach is able to coexist with any kind of applications (socket API). In detail, any types of function such as getaddrinfo(), getsockname(), connect() or other typical system calls will work without alterations if this mechanism is applied to a host.

10. Compatibility and Interoperability with RFC 3484: 1

shim6 has different framework and coordination with RFC 3484. The shim6 host performs address selection that reflects network failures that have occurred between the source and destination host. This may lead some interference with RFC3484 policy table.

11. Security: 1
    This approach has a weakness on Denial of Service attack. It will be concerned that the malicious users can abuse of failure detection and make the network falling into critical condition. However, it depends on a situation how shim6 operate with ICMPv6.

3.4.3. Other issues

- End hosts themselves can avoid network failure. There is no need to modify or reconfigure routers in the path.

- A host must learn address selection information for each destination host. Therefore, the number of cache entries can be very large.

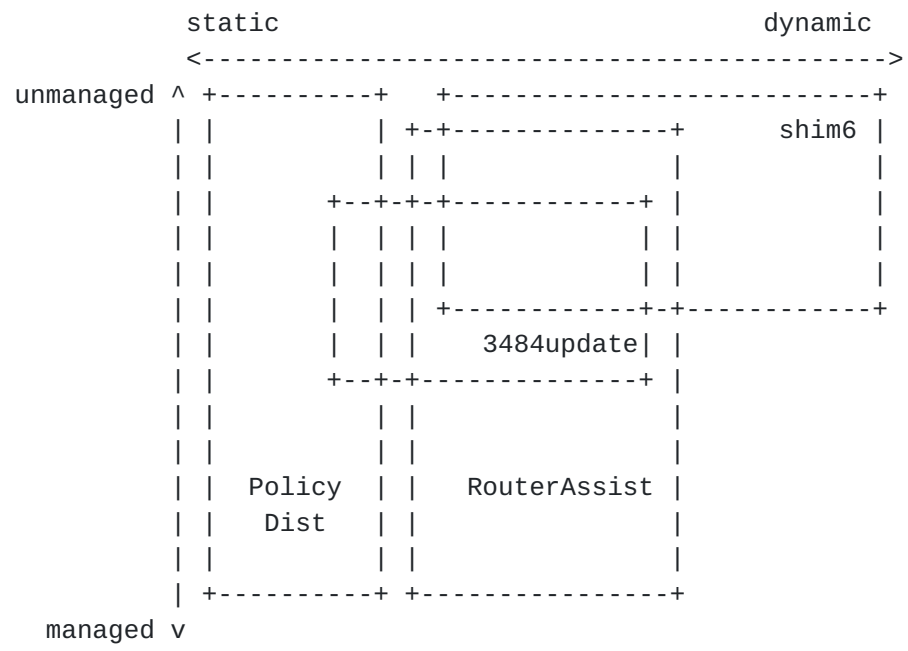
- The existing host OS implementation must be changed significantly.

4. Applicability Comparison

In the previous section, every approach scored "fair" or better for every requirement. This means that every approach can meet the demands of address selection. However, if you actually want to choose one mechanism to solve your address selection problem, it is important to figure out which approach is best suited to your situation. This section tries to evaluate the applicability of each approach from several aspects.

4.1. Dynamic-static and managed-unmanaged

First, we use two axes to evaluate the applicability of the four approaches. One axis shows whether or not the network structure changes dynamically and the other axis shows whether the site is managed or unmanaged. In a managed network, by our definition, a network administrator manages his or her network, routers, and hosts. For example, an enterprise network is managed, whereas a home network and a SOHO network are unmanaged.



PolicyDist:
- In a dynamic site, the policy table must be updated accordingly and traffic for policy table distribution increases.

3484update:
- This is a slightly manageable than shim6 in that 3484update does not change the paths of established connections dynamically.

- In a very dynamic site, the use of an address selection information cache does not have a good effect. This results in connection failure and may degrade usability badly.

- Even in a very static site, a host may try inappropriate addresses or next-hops and experience connection failures.

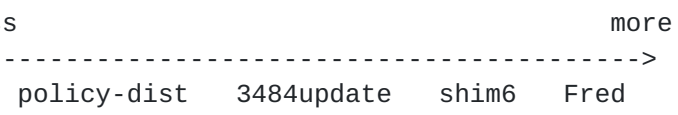
RouterAssist:
- A host must send at least as many queries as the number destination hosts. Therefore, in a static site, this method is not optimal.

- In a very dynamic site, address selection information cache is no help. If the cache function is not used, then connection failures do not occur.

shim6:
- In a static site, shim6 is not desirable because of its connection sequence overhead and timeout-wait for path exploration.

- In a managed site, shim6 is not easy to manage in terms of node-specific address selection control and central control.

4.2. Deployment Difficulty



PolicyDist:
- What must be implemented is a distribution mechanism. The existing protocols, such as RA and DHCP, can be used for this purpose.

3484update:
- The protocol stack or applications on a host must be modified. Routers in a site must be configured to return error messages to the sender of inappropriately addressed packets.
- In RFC3484, precedences and labels are configurable, but not scopes. Those of issues with ULA prefix or non routable global prefix still be left behind even if this RFC would be updated.

RouterAssist:
- The protocol stack and applications on a host must be modified. Furthermore, routers must be modified.

shim6:
- The protocol stack must be modified. For this address selection purpose, corresponding nodes need not support shim6. Basically, there is no need to change the router implementation or configuration.

5. Security Considerations

Incorrect address selection can lead to serious security problems, such as session hijacking. However, we should note that address-selection is ultimately decided by nodes and their users. There are no means to enforce a specific address-selection behavior upon every end-host from outside the host. Therefore, a network administrator must take countermeasures against unexpected address selection.

6. IANA Considerations

This document has no actions for IANA.

7. Conclusions

In this document, we examined solutions to address selection problems in the IPv6 multi-prefix environment. Although almost all solutions examined in this document could be applied to any environment and situation, a solution with a mechanism that is suitable for the situation should be selected.

8. References

8.1. Normative References

[I-D.ietf-v6ops-addr-select-ps]
Matsumoto, A., Fujisaki, T., Hiromi, R., and K. Kanayama, "Problem Statement of Default Address Selection in Multi-prefix Environment: Operational Issues of RFC3484 Default Rules", draft-ietf-v6ops-addr-select-ps-06 (work in progress), May 2008.

[I-D.ietf-v6ops-addr-select-req]
Matsumoto, A., "Requirements for address selection mechanisms", draft-ietf-v6ops-addr-select-req-07 (work in progress), May 2008.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

8.2. Informative References

[I-D.fujisaki-dhc-addr-select-opt]
Fujisaki, T., "Distributing Address Selection Policy using DHCPv6", draft-fujisaki-dhc-addr-select-opt-06 (work in progress), June 2008.

[I-D.ietf-shim6-failure-detection]
Arkko, J. and I. Beijnum, "Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming", draft-ietf-shim6-failure-detection-10 (work in progress), January 2008.

[I-D.ietf-shim6-locator-pair-selection]
Bagnulo, M., "Default Locator-pair selection algorithm for the SHIM6 protocol", draft-ietf-shim6-locator-pair-selection-02 (work in progress), July 2007.

[I-D.ietf-shim6-multihome-shim-api]
Komu, M., "Socket Application Program Interface (API) for Multihoming Shim", draft-ietf-shim6-multihome-shim-api-03 (work in progress), July 2007.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

[RFC5014] Nordmark, E., Chakrabarti, S., and J. Laganier, "IPv6 Socket API for Source Address Selection", RFC 5014, September 2007.

Authors' Addresses

Arifumi Matsumoto
NTT PF Lab
Midori-Cho 3-9-11
Musashino-shi, Tokyo 180-8585
Japan

Phone: +81 422 59 3334
Email: arifumi@nttv6.net

Tomohiro Fujisaki
NTT PF Lab
Midori-Cho 3-9-11
Musashino-shi, Tokyo 180-8585
Japan

Phone: +81 422 59 7351
Email: fujisaki@syce.net

Ruri Hiromi
Intec Netcore, Inc.
Shinsuna 1-3-3
Koto-ku, Tokyo 136-0075
Japan

Phone: +81 3 5665 5069
Email: hiromi@inetcore.com

Ken-ichi Kanayama
INTEC Systems Institute, Inc.
Shimoshin-machi 5-33
Toyama-shi, Toyama 930-0804
Japan

Phone: +81 76 444 8088
Email: kanayama_kenichi@intec-si.co.jp

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).