Network Working Group
Internet-Draft
Expires: December 16, 2006

M. Bagnulo
UC3M

June 14, 2006

Updating RFC 3484 for multihoming support
draft-bagnulo-rfc3484-update-00

Status of this Memo

Copyright Notice

Abstract

This note describes the limitations of RFC 3484 in multihomed environments and proposes possible updates to the default address selection mechanisms in order to cope with the identified limitations.

Table of Contents

1. Introduction

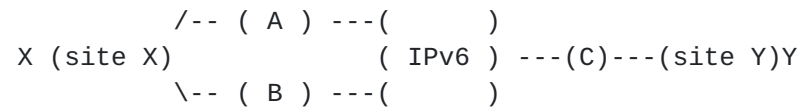
A way to solve the issue of site multihoming is to have a separate site prefix for each connection of the site, and to derive as many addresses for each hosts. This approach to multi-homing has the advantage of minimal impact on the inter-domain routing fabric, since each site prefix can be aggregated within the larger prefix of a specific provider; however, it opens a number of issues, that have to be addressed in order to provide a multihoming solution compatible with such addressing scheme.

In this memo we will present the issues that such multihoming configuration presents with respect to the address selection mechanisms. In particular, in section 2 of this memo, we describe the limitations of current source and destination address selection mechanisms specified in RFC 3484 in the described multihoming configuration. In section 3 we describe possible modifications to RFC 3484 to cope with the identified limitations.

2. Limitations of RFC 3484 in multihomed environments

2.1. Reference topology

In the following discussion, we will use this reference topology:



The topology features two hosts, X and Y. The site of X is multihomed while the site of Y is single homed. Host X has two global IPv6 addresses, which we will note "A:X" and "B:X", formed by combining the prefixes allocated by ISP A and B to "site X" with the host identifier of X. Y has only one address "C:Y".

We assume that Y, when it starts engaging communication with X, has learned the addresses A:X and B:X, for example because they were published in the DNS. We do not assume that the DNS is dynamic: there will be situations in which both A:X and B:X are published, while in fact only one is reachable. We assume that X, when it receives packets from Y, has only access to information contained in the packet coming from Y, e.g. the source address; we do not assume that X can retrieve by external means the set of addresses associated

to Y. similar assumptions are made when X is initiating the communication, only that in this case, a single address i.e. C:Y is published in the DNS

In this scenario, both ISPA and ISPB are performing ingress filtering and have not relaxed the source address checks. So, we assume that an ingress filtering compatibility mechanism [2] is available in the multihomed site (Site X) so that packets are forwarded through the ISP that corresponds to the source address prefix included in the packet by the host.

2.1.1. RFC 3484 and the shim6 protocol

The shim6 working group is developing a shim protocol to preserve established communications through outages. Through the shim protocol a pair of shim enabled communicating peers will be able to survive outages affecting the path used for the communication using alternative addresses to exchange packets. Communications will be preserved because even though a different address pair is being used for the communications, exchanged packets are presented to the upper layers as containing the addresses used initially. In order to perform this function, shim protocol support from both peers involved in the communication is required.

The problem addressed in this note is somehow different, since the goal of considered mechanisms is to enable the establishment of a new communication after an outage. In this case, the communication has not yet been established and the address pair to be used for exchanging packet is being determined at this very moment. It is possible then to try with different source and destination addresses until a working address pair is discovered. Another difference is that in this case, the mechanisms are located only in the multihomed end of the communication and no special support other than regular IPv6 is required from the non-multihomed peer. Essentially , the proposed mechanisms are aimed to allow a node in a multihomed site that implements them to be able to establish a new communication after an outage with an external host that does not have any multihoming specific support mechanism. (In the reference topology depicted above, the mechanisms reside in the Host X and no multihoming mechanisms are located in Host Y)

It is also possible to use the mechanisms described in this note to establish communications between two shim enabled peers. However, whether this is the best approach to follow in this case will be determined by the merits of the modifications to current address selection mechanisms proposed to overcome the limitations that current mechanisms exhibits in multihomed environments. It may well be that in the case of two shim enabled communicating peers, it makes

more sense to define special mechanisms that require cooperation from both nodes to establish new communications after an outage.

2.2. The problem: address selection after failures

In case that a failure occurs in one of the ISPs of the multihomed site, it may not be possible to establish a new communication towards a destination outside the site using the addresses derived from the prefix of the ISP affected by the failure. For instance, in the case that the link between ISPA and the Internet fails, it will not be possible to establish a communication between X and Y using address A:X. In this case, any communication involving this address will fail because:

- If Y tries to establish a communication with X using A:X as a destination address, packets would be discarded because there is no path available from the Internet to ISPA.
- If X tries to communicate with Y using A:X as a source address, packets will be routed through ISPA in order to comply with ingress filters, and because ISPA has no link available with the rest of the Internet, the packet will be discarded (it should be noted that even if the packet could make it to Y, reply packets from Y to X would contain A:X as a destination address, which is unreachable from Y).

So, in order to establish a communication between X and Y when a failure has occurred in ISPA, the address derived from ISPA block i.e. A:X, must not be used for the communication.

The solution for this problem has to be provided by the address selection mechanisms. In particular, when the communication is established from the host Y to the host X, the solution has to be provided by the destination address selection mechanism at host Y and when the communication is established from the host X to the host Y, the solution has to be provided by the source address selection mechanism at host X. Default address selection for IPv6 hosts is specified in RFC 3484 [1]

We will next analyze the support provided by RFC 3484 when the communication is established from host Y to host X. In this case, host Y has two possible destination addresses A:X and B:X. Without any additional knowledge, both addresses are equivalent to host Y, so the default destination address selection mechanism will return a list of the two addresses ordered as they were returned by the resolver. It may occur that A:X is first. In this case, host Y will use A:X to reach host X and it will fail. At this point, RFC 3484 states that if there are other destination addresses available, the application should retry to establish the communication, using the next address in the list. If the application retries with address

B:X, the communication will be established successfully.

In conclusion, the current destination address selection mechanism is enough to deal with this situation (as long as applications retry with all the addresses).

Next, we will analyze the support provided by RFC 3484 when the communication is established from host X to host Y. In this case, destination address selection performed in host X is trivial, since there is only one address available for Y (C:Y). Source address selection mechanism as specified in RFC 3484 will not prefer any of the two source addresses if no additional information is available, so any of the addresses can be selected as source address. In the case that address A:X is selected, the communication will fail. In this case there are no alternative destination address to retry with, so the communication will definitely fail.

In conclusion, the source address selection mechanism defined in RFC 3484 is not enough to support this scenario. This memo defines mechanisms to provide a solution for this case.

3. Updates to RFC 3484

RFC 3484 essentially performs two functions:

- It provides an ordered list of destination addresses to the application that are used to initiate a communication. In addition RFC 3484 states that the application should iterate through all the addresses contained in the list until they find a working address
- In case that the application does not select a source address, the source address selection mechanism describes how the IP layer selects the source address for a given destination address.

However, RFC 3484 does not provides support for the following situations:

- When the source address is specified by the application, the source address selection mechanism does not provide any guidance to the application about how to select the source address for communicating with a destination address. In particular, RFC 3484 does not recommend that the application should iterate through all available source addresses until a working address pair is found.
- When the source address is unspecified by the application and it is selected by the IP layer, the source address selection mechanism does not take into account that a given destination address may be reachable when using a certain source address and unreachable when using another source address.

The result is that when an outage occurs, current source address selection mechanisms specified in RFC 3484 may not be able to find a working source and destination address pair, even though, one exists. In this section, we describe some modifications to RFC 3484 that are aimed to cope with these issues, and enable the source address selection mechanism to discover the available working pairs.

Accordingly to the structure of RFC 3484, the proposed modifications are divided in two components:

- A set of rules that provide guidance to the application when it decides to select the source address itself, similar to those already available for the selection of the destination address.
- A modification to the source address selection mechanism performed by the IP layer, so that unreachable source and destination address pairs are detected and alternative address pairs are tried for establishing a communication.

3.1. Providing guidance to the applications for selecting source addresses

3.1.1. Considered scenario

In this case the application selects the source address to use when sending packet to a given destination address (e.g. using bind()). The stack and the source address selection mechanisms should honour this choice. The goal of the proposed mechanisms is to provide guidance to the application in order to perform this source address selection. Current RFC 3484 specification is silent in this case. In order to fill this void, we propose two changes as described in the following sections.

3.1.2. Retrying with different source addresses

Current RFC 3484 states that when more than one destination address are available, the application should iterate through them until a working address is found. However, RFC 3484 is silent with respect to the case where multiple source addresses are available and the application decides to select the source address to be used.

So, the proposed change is to update RFC 3484 to include that:

> In the case that the application decides to select the source address used in the communication (e.g. using bind()) the application should iterate through all the source and destination address pairs available until a working pair is found.

In addition an additional rule must be added to the source address selection algorithm:

Rule 0: Avoid unreachable source addresses.

If the address pair with source address SA and destination address D is known to be not working, then prefer SB

3.1.3. Providing an ordered list of source address

In addition to recommending that when the application selects the source address, it should try with all available address pairs to establish the communication, it would also make sense to provide some guidance about which addresses to try first. It should be noted that RFC 3484 does provides an ordered list of destination addresses so that the application can try with the multiple available destination addresses in the suggested order. A similar approach is here suggested for the source addresses

Currently, there are several ways for an application to retrieve the list of available source addresses i.e. the addresses available in the local host. A possibility would be to let the source address selection mechanism order that list before it is returned to the application. The problem with this approach is that available calls to retrieve the source address set have no destination address information associated, and the problem being dealt here is the selection of a source address to use with a given destination address.

A possible approach then is to define a new function to retrieve an ordered list of available source addresses for a given destination address. In this case, the application would have an ordered list of destination addresses and for each of them the application would retrieve an ordered list of potential source addresses. It should be noted that current RFC 3484 already provides an algorithm to order the set of source addresses, but instead of returning the ordered list it just uses the "best" one. This basically means that the algorithm for sorting the source addresses for a given destination address is already available in RFC 3484. In this case, only the new function that returns the ordered list of source addresses for a given destination address needs to be defined (of course, applications need to be modified so that the new function is used)

In the approach described in the previous paragraph an application would obtain an ordered list of destination addresses and for each destination address an ordered list of source addresses. This option is attractive because it does not requires major changes in the way source and destination address selection mechanisms described in RFC 3484 operate (the only change required is a new function call). However, such approach has the drawback that the resulting order of address pairs to try may not be the optimal, since for each

destination address, all the available source address would be tried before moving on to the next destination address. A possible workaround for this limitation would be that an ordered list of address pairs is returned instead of an ordered list of destination addresses and for each destination address, an ordered list of source addresses. The drawback of this approach is that not only a major change in the source and destination address selection is required to produce a list of ordered source and destination address pairs instead of a list of source addresses and a list of destination addresses, but also application should use a new function that returns the ordered list of address pairs instead of the function currently used to retrieve the destination address list.

3.2. Modifications to the IP layer source address selection mechanism

3.2.1. Considered scenario

In this case the application that is communicating has not selected the source address to be used (i.e. no bind() to a specific source address). In this case, it is up to the IP layer to select the proper source address to include in the outgoing packets. The source address selection is then performed at the connect() time for connected sockets or when each packet is sent for non-connected sockets. We will next consider two different cases: TCP sockets, and UDP sockets.

3.2.2. TCP sockets

In this case, the application has selected a destination address and it has open a TCP socket. Then it performs a connect(). At this point in time, the 3-way handshake of TCP is executed. Normally, a source address is selected before performing the handshake and the SYN packet is sent using this selected source address. In order to deal with unreachable source addresses in this case, the proposed approach is that if the 3-way handshake can not be completed using one of the source addresses, the IP layer should iterate through the rest of the available source addresses until a working source address is found and the 3-way handshake of TCP is completed. The list of source addresses to try with is ordered using the source address selection algorithm described in the current RFC3484.

3.2.3. UDP sockets

In this case, it is not possible to use a similar approach to the one described for TCP, because there is no way to determine if a given source address is working or not, because there is no connection establishment packet exchange as in the case of TCP. So, in this case, the basic action that can be performed would be to keep trace

of the source address that has been used and some hints if those have worked or not. In particular, the proposed approach is to keep trace of incoming packets with a given address pair as a possible hint of a working address pair. A detailed description of how this would work is included in [3]. In any case, the goal here is to keep track of the source addresses tried for each destination address and whether these have worked or not (according to the previous definition of "working"). If they have not worked, then they should be avoided as long as alternative addresses are available. If they have worked, they should be preferred over other potential source addresses for that destination address.

4. Acknowledgments

Thanks to Pierre Baume for reviewing this document and providing feedback

5. References

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

[2] Huitema, C. and M. Bagnulo, "Ingress Filtering compatibility for IPv6 multihomed sites", ID draft-huitema-shim6-ingress-filtering-00.txt, October 2005.

[3] Bagnulo, M., "Address selection in multihomed environments", ID draft-bagnulo-shim6-addr-selection-00.txt, October 2005.

Author's Address

Marcelo Bagnulo
Universidad Carlos III de Madrid
Av. Universidad 30
Leganes, Madrid 28911
SPAIN

Phone: 34 91 6248814
Email: marcelo@it.uc3m.es
URI: http://www.it.uc3m.es

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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