Network Working Group
Internet-Draft
Expires: December 2002

Dave Thaler
Christian Huitema
Microsoft
29 June 2002

Multi-link Subnet Support in IPv6
<draft-ietf-ipv6-multilink-subnets-00.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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Copyright Notice

Abstract

Bridging multiple links into a single entity has several operational advantages. A single subnet prefix is sufficient to support multiple physical links. There is no need to allocate subnet numbers to the different networks, simplifying management. This document introduces the concept of a "multilink subnet", defined as a collection of independent links, connected by routers, but sharing a common subnet prefix. It then specifies the behavior of multilink subnet routers so that no changes to host behavior are needed.

1. Introduction

Not all link-layer media can be easily bridged. Classic IEEE 802 bridging technology fails when the media does not naturally support IEEE 802 addressing. Furthermore, the operation becomes problematic when the different links don't support the same MTU size. Finally, bridging cannot be easily implemented when the network interface cannot be easily placed in "promiscuous" mode.

We define a "multilink subnet" as a collection of independent links, connected by routers, but sharing a common subnet prefix. This might be used, for example, in a home network where a router connects a wired and a wireless link together to form a single subnet.

A multilink subnet can be implemented in either of two ways. In a simple scenario, a router can act as an asymmetric Neighbor Discovery proxy, connecting links in a loop-free topology. In a more complex scenario, an arbitrary topology exists, and routers within the subnet communicate using some means of exchanging host routes. In this draft, we will only specify the behavior of a router in the simple scenario, while the behavior of hosts in either scenario is the same (indeed hosts need not even know which scenario they are in, or even whether they are on a multilink subnet). Appendix A will discuss some of the issues affecting routers in the complex scenario.

2. Terminology

multilink subnet:
a collection of independent links, connected by routers, but

sharing a common subnet prefix.

subnet scope:
multicast SCOP value 3, as specified in [ADDRARCH], which covers a (potentially multilink) subnet. This is the next larger multicast scope above link scope.

multilink-subnet router (MSR):
a router which has interfaces attached to different links in a multilink subnet, and which implements the rules in this document. Such interfaces are boundaries for the link scope, but not for the subnet scope.

3. Design Goals

Multilink subnet functionality is designed with the following goals in mind:

- Existing IPv6 end hosts should continue to work when connected to a multilink subnet, without requiring any change to their behavior. For example, the host behavior parts of Router Discovery, Neighbor Discovery [ND], and Multicast Listener Discovery [MLD], must be supported.

- Leave link-local address behavior unchanged. Link-local behavior continues to function only within a link, not across a multilink subnet. That is, sending and receiving unicast, anycast, and multicast traffic within the link should be supported in the normal fashion.

- Also support sending and receiving unicast and anycast traffic at the site and global scopes.

- Also support sending and receiving multicast traffic at the subnet scope and above.

- Prevent routing loops.

- Support nodes moving between links within the subnet, with a reasonably fast convergence time (on the same order as Neighbor Unreachability Detection).

4. Overview

This section gives an overview of multilink subnets. We describe the behavior of hosts (which is normal IPv6 host behavior with no changes), and the resulting requirements for routers.

4.1. Router Discovery

Router Discovery continues to work on a per-link basis, as specified in [ND]. When sending Router Advertisements (RAs) with a Prefix Information Option, there are two possibilities for how a multilink subnet router can influence the Neighbor Discovery procedure used.

4.1.1. Making hosts not use ND

If the MSR sets the A (autonomous address-configuration) flag on, and the L (on-link) flag off, then hosts on the link will attempt stateless address configuration [ADDRCONF] in the given prefix, but will not treat the prefix as being on-link. As a result, neighbor discovery is effectively disabled and packets to new destinations always go to the router first, which will then either forward them if the destination is off-link, or redirect them if the destination is on-link.

In the remainder of this document, we will refer to this model as the "off-link" model, since hosts initially treat all addresses in the subnet as being off-link.

4.1.2. Making hosts use ND

If the MSR sets both the A and the L flags, then hosts on the link will perform stateless address configuration and neighbor discovery as usual. However, since Neighbor Solicitations (NSs) from existing hosts are sent to a link-scoped solicited-node multicast address, they will never reach nodes on other links within the subnet. Instead, MSRs must either know the location of the destination a priori, or else be able to relay such NS's to other links. We will return to this discussion in Section 5.

In the remainder of this document, we will refer to this model as the "on-link" model, since hosts treat all addresses in the subnet

as being on-link.

4.1.3. Effects on Duplicate Address Detection

In either approach above, existing nodes will still do Duplicate Address Detection using the link-scoped solicited-node multicast address.

Two important issues arise that must be addressed:

1) If two nodes on different links in the subnet simultaneously attempt DAD for the same address, care must be taken to so that the collision is detected correctly.

2) If a node moves from one link to another link in the same subnet, and performs DAD in its new location, care must be taken so that MSRs can distinguish between such a move, and a legitimate duplicate, so that after the move, the node can retain its address.

Because of these issues, routers MUST NOT use cached information to respond on behalf of off-link nodes.

Another problem arises from the statement in [ND] that: "the link-local address MUST be tested for uniqueness, and if no duplicate address is detected, an implementation MAY choose to skip Duplicate Address Detection for additional addresses derived from the same interface identifier".

Collisions would result if the interface identifier were unique on the link, but not across the entire multilink subnet. To avoid this, MSRs must get involved in duplicate address detection even for link-local addresses, to ensure that all addresses are unique across a multilink subnet.

4.2. Neighbor Discovery

Neighbor Discovery is used differently, depending on whether the on-link or off-link model is used, as described in the previous section.

Off-link model
: If the subnet is treated as being off-link, all packets are sent to a default router. It is then the default router's responsibility to figure out the next-hop of the packets. If the next-hop is on-link, it sends a Redirect to the source.

On-link model
: If the subnet is treated as being on-link, nodes will send NS's to the solicited node multicast address. (If a node has interfaces attached to multiple links in the subnet, NS's MAY be sent on each link.) If the next-hop is off-link, a router will respond with a proxy Neighbor Advertisement (NA) containing its own link-layer address.

In either case, it is the router's responsibility to determine whether a destination in the subnet is on-link.

5. ND-Proxy Behavior

A simple ND-Proxy router will have one or more interfaces on which it acts as a router, and optionally an interface on which it acts as a "host", proxying for all nodes on its router interfaces. We will hereafter refer to an interface in this latter mode as a "proxy-mode" interface. (This configuration is consistent with that of an IGMP/MLD Proxy [IGMPPROXY].)

An ND-Proxy MAY support automatic configuration as follows. First, it starts out as a normal host, discovering routers (if any) on each interface. Then, it switches to router-mode on all interfaces on which no routers were discovered. If any interfaces with routers were found, it chooses one and acts in proxy-mode on that interface. In Router Advertisements sent on its router-mode interfaces, the ND-Proxy advertises itself as a default router, and includes copies of the Prefix Information options it learned on its proxy-mode interface, possibly after changing the parity of the L flag, depending on whether it wants nodes to use the on-link or the off-link model.

5.1. Basic Unicast

In this section, we step through an example of basic unicast communication, assuming that address configuration has already completed, and the router's routing table and neighbor cache

already have any required information. Section 5.2 will then explain how it obtains the required information when a cache miss occurs.

In the simple scenario depicted in Figure 1 below, two links, (1) and (2) on a common subnet with global prefix G, are connected by an ND-Proxy B. Node A has link-layer address a on link 1, and has acquired global IPv6 address Ga. ND-Proxy B is in router-mode on link 1, where it has link-layer address b1, and IPv6 address Gb1. It is in either router-mode or proxy-mode (e.g., if C is a router) on link 2, where it has link-layer address b2 and IPv6 address Gb2. Node C has link-layer address c on link 2, and IPv6 address Gc. Node D has link-layer address d on link 1, and IPv6 address Gd.

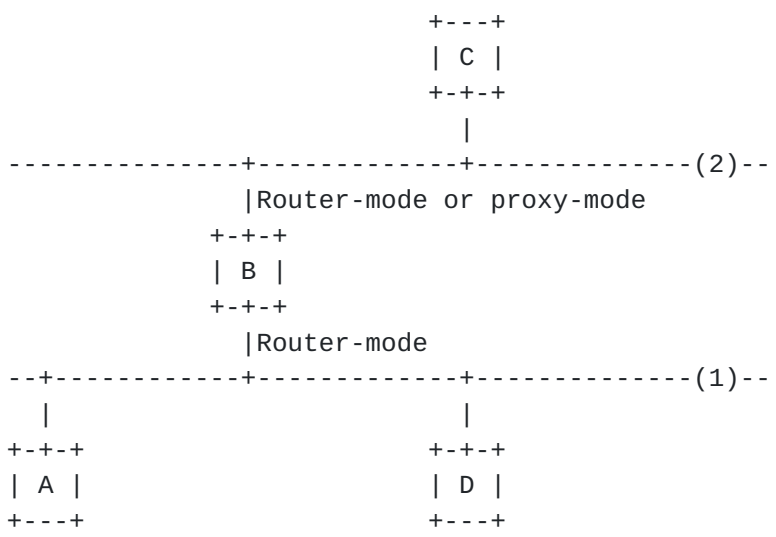

Figure 1: Simple ND-Proxy Scenario

Off-link model

When A wants to start communication with Gc, it finds that the destination address matches no on-link prefix, and so sends the packet directly to its default router B. B first applies its usual packet validation rules (including decrementing the Hop Count in the IPv6 header). B knows that C is on-link to link 2, with link-layer address c, and so it forwards the packet to C.

When A wants to communicate with Dc, it again finds that the

destination address matches no on-link prefix, and so sends the packet directly to its default router B. B knows that D is on-link to the same link as A, and so responds with a Redirect.

On-link model

When A wants to start communication with Gc, it finds that the destination address matches an on-link prefix, and so sends an NS to the solicited-node multicast address Sc constructed from Gc. The NS message is received by the ND-Proxy B, which listens on all multicast groups. B knows that C is on-link to link 2, and responds to A with an NA containing its own link-layer address b1 as the Target Link-Layer Address.

After this, A can send packets to the address Gc. The packets will be sent to the link address b1; they will be received by B, which will apply its usual validation rules (including decrementing the Hop Count in the IPv6 header), and forward them to the address c on link 2.

When A wants to communicate with Gd, it again finds that the destination address matches an on-link prefix, and so sends an NS to its solicited-node multicast address. D receives the NS and responds. B also receives the NS, but knows that D is on the same link as A, and so does not respond.

Note that we did not assume that the links had to use IEEE 802 addresses, or in fact any form of consistent link-layer addressing. B can also handle MTU discovery procedures, returning an ICMP messages if either A or C sends a packet that is too long.

5.2. Router Configuration

The previous section assumed that the router's routing table and neighbor cache already had any required information. We now describe how this can be done.

Like any other router, an MSR can acquire routes (including the subnet prefix) by using manual configuration or a routing protocol. An MSR with all interfaces in the same subnet MAY acquire its information solely based on RAs received from another

router (which is not an MSR), in the same way a host would. It can then advertise the same prefix/route information on other links in the subnet, using either the on-link or off-link model.

To receive Neighbor Solicitations for nodes for which it is proxying, the MSR must be able to receive NS's sent to any multicast group on any of its links within the subnet (including its proxy-mode interface).

When a Neighbor Solicitation is received, and a corresponding entry is not found in its neighbor cache, the MSR will attempt to resolve by sending a Neighbor Solicitation on each attached link in the subnet, except that in the on-link model one is not sent back on the link from which an NS was just received. After sending an NS, the router SHOULD suppress sending of any other NS's for the same target address for a short interval (which must be less than ND's RetransTimer), and while it is resolving a next-hop, remember each node sending an NS for the same target address. If a Neighbor Advertisement is received, a proxy NA is sent to each nodes from which the MSR received an NS.

A Neighbor Advertisement would be sent in response to an NS only by (a) the actual node with the target address, or (b) an MSR which has received an NA in response to a relayed NS it sent as a result of receiving the first NS. Specifically, a Neighbor Advertisement MUST NOT be sent just because the MSR has a neighbor cache entry for the target. When an MSR receives an NA, it sends an NA to all nodes from which it received NSs above.

As specified in [ND], proxy Neighbor Advertisements sent by MSR's on behalf of remote targets always have the Override bit clear.

5.3. Multicast

Most current multicast routing protocols are based on a "Reverse-Path Forwarding" check. That is, they drop a packet if the packet does not arrive on the link towards a given address (e.g., the source address, or a Rendezvous Point address associated with the group address). Thus, sourcing multicast will work as long as a router can tell which link is towards any address within the subnet. Note that in particular, simply using the subnet route is not sufficient in a multilink subnet. If an MSR's longest-match RPF lookup matches the subnet route for the multilink subnet, it means the source is in the subnet, and the neighbor cache is

consulted (as for unicast) to find the link towards the source.

To allow hosts to receive multicast data, an ND-Proxy acts as an MLD-Proxy using the rules described in [IGMPPROXY].

5.4. Disabling ND-Proxying

The above rules assume only one ND-proxy is acting as a router on each link. Consequently, an ND-Proxy MUST stop acting as a router on a given interface if it receives an RA from another device on that interface.

6. Connecting ND-Proxies

ND-Proxies can be connected, either in parallel (connecting different leaf links) or in series (subject to the constraints below), using the same rules specified above. The following figure illustrates this.

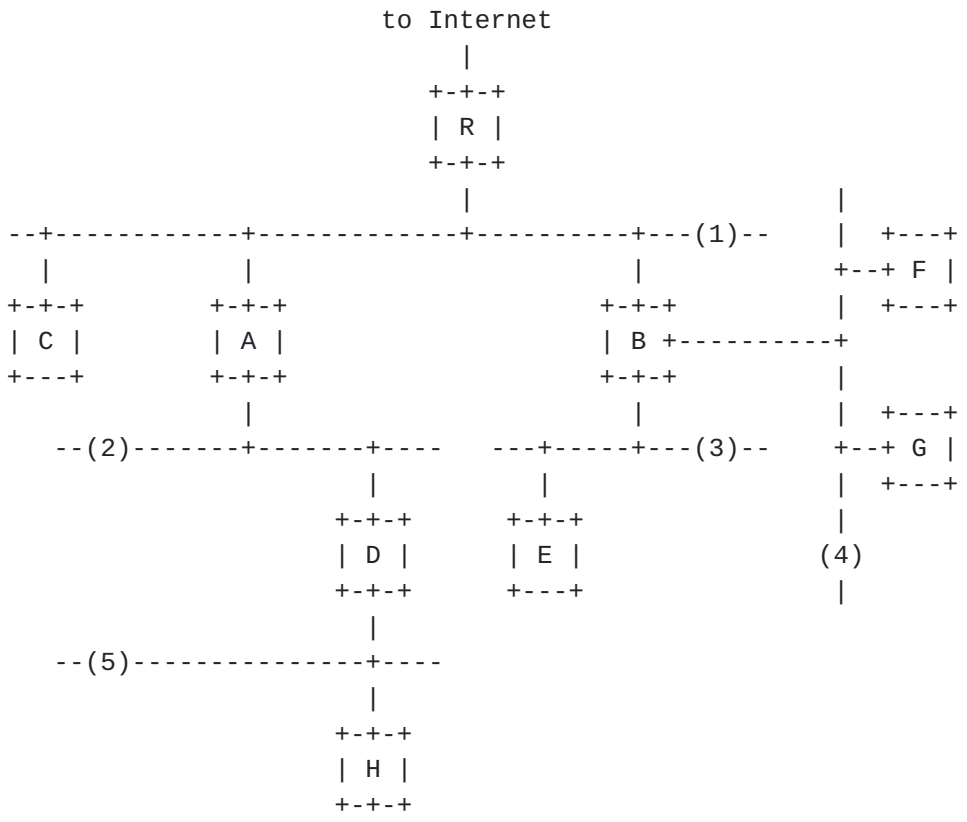

Figure 2: Multiple ND-Proxy Scenario

Figure 2 shows a complex tree topology with MSRs A, B, and D connecting five links into a single subnet with hosts C, D, E, F, and G. R is a normal router that provides connectivity to an internet, and sends RAs on link 1.

MSR's A and B have both discovered router R on link 1, and are acting in proxy-mode on that link, while acting in router-mode on their other links.

MSR D is acting in proxy-mode on link 2, and in router-mode on link 5. In this configuration, all hosts can communicate with each other and with the Internet. However, the automatic configuration mechanism described in section X above does not always work for D. If D comes up before A, then it would act in router-mode on link 2, preventing A from doing so. As a result, chaining ND-Proxies should only be done where some means of preventing this exists. For example, D could either:

- be manually configured to be in proxy-mode on link 2, or
- could have one of its ports permanently labelled as being its proxy-mode port, so the correct orientation is obvious either when connecting cables (if homogenous media types are used on different interfaces) or when first obtaining the device (if its purpose is to connect heterogeneous media types).

7. Security Considerations

In general, the security considerations issues and recommendations are the same as those described in Section 11 of [ND]. In addition, the following points are worth noting.

In the off-link model, the source address of the relevant ICMP messages is the address of the actual source. As a result, messages can be authenticated, if some means of securing Router Discovery is used.

In the on-link model, Neighbor Discovery messages are proxied, and hence Neighbor Discovery is harder to secure. As a result, when secure Neighbor Discovery is required, the off-link model should be used.

8. Appendix A: Complex Scenario Issues

In the network depicted in Figure 2, we have now three links, and also three multilink-subnet routers (MSRs), B, E, and F.

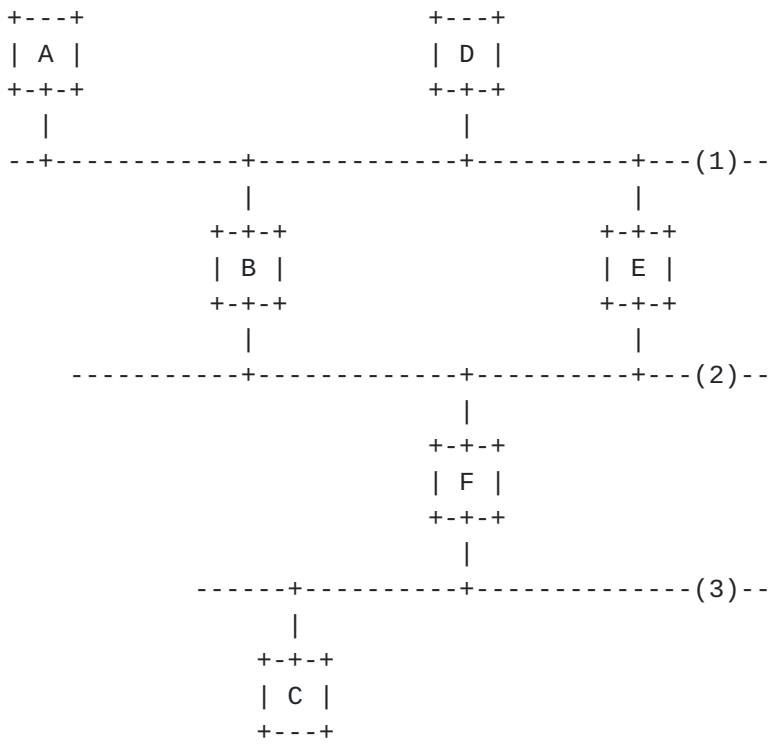

Figure 3: Arbitrary Topology Scenario

The network is sufficiently complex to expose several problems:

- If A sends an NS packet, that packet is received by both B and E. Depending on the inter-router communication mechanism, this could lead to duplicate transmissions on link 2, and possibly to random behaviors, or to loops.

- If A sends a multicast packet, and that packet is relayed by both B and E, it would lead to duplicate traffic, or even potential loops. It may not be relayed at all, if neither B nor E realize there is a group member hidden behind F.

There are multiple possible approaches to solving the above problems which might meet our design goals. We discuss two examples below.

8.1. Method 1: Flooding Neighbor Solicitations

Neighbor Soliticitations and Advertisements could be proxied out all interfaces and hence flooded across the subnet. To prevent loops, some mechanism such as a new "Local Distance" option in NA's would be needed. The use of Neighbor Discovery would then be equivalent to a simple means of exchanging host routes between MSRs, and could be used regardless of which routing protocol is used.

To route actual packets, an MSR's route lookup would determine that the longest matching route is on-link to multiple links. The router would consult its (conceptual) neighbor cache, and use the next-hop with the lowest Local Distance. The same procedure would apply to multicast packets as well, when the router would look up the RPF address.

8.2. Method 2: Proactively populate host routes

MSR's could inject host routes into a routing protocol used within (at least) the subnet upon detecting a new node on a directly-connected link (e.g., when DAD completes on an Ethernet, or when IPv6CP completes on a PPP link).

Once host routes exist, either the off-link or the on-link model could be used. In addition, multicast works with no changes, since host routes would be used for RPF checks.

Another advantage is that since all resolution is done by MSR's "a priori", no additional delay is incurred when A wants to communicate with A. If the on-link model is used, no neighbor discovery delay exists at all. Packets are immediately forwarded along the correct path. This approach avoids bursty-source problems.

Since host routes are cached state, they cannot, however, be used for duplicate address detection, due to the issues described in Section 4.1.3. That is, the presence of a host route does not imply a duplicate, since the node may have just moved. The lack of a host route does not imply uniqueness, since another node may be simultaneously choosing the same address. As a result, DAD requires additional mechanisms, such as flooding neighbor discovery messages as in Method 1, or provided by a specialized routing protocol.

Work in progress in the Mobile Ad-hoc Networks (manet) WG may provide solutions to the above problems in the future.

9. Acknowledgements

Steve Deering, Brian Zill, Hesham Soliman, and Karim El-Malki participated in discussions that led to this draft. The term "multilink subnet" was coined by Steve Deering.

10. Authors' Addresses

Dave Thaler
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
Phone: +1 425 703 8835
EMail: dthaler@microsoft.com

Christian Huitema
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
EMail: huitema@microsoft.com

11. References

[ADDRARCH]
Hinden, R., and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

[ADDRCONF]
Thomson, S., and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

[IGMPPROXY]
Fenner, B., He, H., Haberman, B., and H. Sandick, "IGMP-based Multicast Forwarding (IGMP Proxying)", draft-ietf-magma-igmp-proxy-00.txt, November, 2001.

[MLD]
Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.

[ND] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

12. Full Copyright Statement

Table of Contents