Site prefixes in Neighbor Discovery

<draft-ietf-ipngwg-site-prefixes-03.txt>

Status of this Memo

Abstract

This document specifies extensions to IPv6 Neighbor Discovery to carry site prefixes. The site prefixes are used to reduce the effect of site renumbering by ensuring that the communication inside a site uses site-local addresses.

This protocol requires that all IPv6 implementations, even those that do not implement this protocol, ignore all site-local addresses that they retrieve from the DNS when the AAAA or A6 RRset contain both global and site-local addresses. If the RRset contains only site-local addresses those addresses can be used.

Contents

1. INTRODUCTION AND MOTIVATION

In order to maintain the aggregation of the global Internet routing tables it might be necessary for whole sites to renumber to use different prefixes for their global IPv6 addresses. Such renumbering would not directly benefit the renumbered sites but instead be necessary for the scaling of the Internet as a whole.

In order to increase the probability that such renumbering is viewed favorably by the sites themselves, which see little or no direct benefit, it is critical that both the effort of renumbering is kept at a minimum and also that the risk associated with renumbering is as small as possible.

The Stateless address autoconfiguration [ADDRCONF] and support for router renumbering [ROUTER-RENUM] make it easier to renumber a site. However, these protocols do not by themselves address long-running TCP connections or cases where IP addresses have been stored in some configuration file. Thus additional measures are needed to reduce the cost of renumbering.

For many sites it is much more critical to maintain the internal communication than the inter-site communication over the Internet. Based on that observation this proposal tries to limit the effect of a site renumbering one or more of its global prefixes by ensuring that intra-site communication can use site-local addresses which would not be affected by the site renumbering. With this proposal it is possible to maintain internal long-running TCP connections or otherwise store IPv6 addresses for longer time than would have been possible without it.

As specified in [ADDR-TODAY] IP addresses are no longer temporally unique. This implies, among other things, that applications should not store IPv6 addresses without a mechanism for honoring the DNS time-to-live and refreshing the IPv6 address. This protocol is not intended to deter from that recommendation but is merely based on the observation that the applications today might assume that IPv4 addresses are temporally unique and it is likely that some applications might not be corrected in their behavior as they are moved to IPv6. It would be unfortunate if such application "brokenness" would lead sites to view site renumbering as a too risky or a too costly operation.

This document does not address the general issues of renumbering such as renumbering a single host or a subnet. It is targeted at site renumbering. The proposal does not attempt to address how long-running TCP connections going outside a site will survive the site renumbering.

The author would like to acknowledge the contributions the IPNGWG working group and in particular Mike O'Dell who pointed out the importance of the problem, and Robert Elz who suggested this approach to solving the problem.

2. TERMINOLOGY

This documents uses the terminology defined in [IPv6] and [DISCOVERY] and in addition:

Multi-sited node
A node that has interfaces in multiple sites.

2.1. What is a Site?

This document does not attempt to define the concept of a "site", but it does place some assumptions on such a definition:

- A site is an administratively controlled piece of topology that is well-connected. It can be connected using tunnels including the special form of tunnels (using routing headers and home address options) defined in [MOBILE-IPv6].

- A link can belong to zero or one site. This implies that an interface can belong to at most one site.

- A node can have interfaces belonging to different sites. Such a node is said to be multi-sited.

- A mobile node [MOBILE-IPv6] which has been assigned one or more site-local addresses and moves outside the site which contains its home address (its "home site") is considered to have one interfaces which is part of the "home site".

2.2. Requirements

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this

document, are to be interpreted as described in [KEYWORDS].

3. OVERVIEW

The goal of this extension to Neighbor Discovery is to make communication that is local to a single site use the site-local addresses instead of the global addresses. If all communication internal to a site uses site-local addresses then the site's global addresses can be renumbered without having any affect on the internal communication. Thus the risk associated with site renumbering is lowered - applications that store IPv6 addresses and long-running TCP connections will, as long as the communication is local to the site, continue to operate across the renumbering of the site.

A few alternative solutions have been explored. An early proposal was to place the site-local addresses in the name service (e.g., the DNS) and make sure they are returned first in the list of addresses returned to an application (to make it likely that the application will use that address). That proposal has the disadvantage that the name service must return different addresses depending on who asks the question; if a node inside the site asks for an address it should return the site-local address(es) but if a node outside the site asks it must not return a site-local address. This is referred to as the two-faced DNS. While some sites use a two-faced DNS today as part of their firewall solution it would be rather unfortunate if each and every site had to deploy such a solution. See [GSE-EVAL] for more discussion.

An earlier version of this proposal took a different approach. The name service would only contain global addresses and the routers would advertise the global address prefixes assigned to the site which the nodes would use to derive site-local addresses corresponding to the global addresses returned from the name service. That approach had the disadvantage that all nodes in a site would be required to respond to their automatically derived site local address. For instance, it was not possible to have certain mobile nodes that would only be reachable using global addresses.

3.1. Protocol Overview

This version of the document takes a middle ground. The site-local addresses, as well as the global addresses, are stored in the DNS without requiring a "two-faced" DNS. All nodes are required to ignore any site-local addresses retrieved from the DNS unless:

1) the DNS returned only site-local addresses (used in sites that are not connected to the Internet i.e. where all the addresses

are site-local), or

2) they can determine that they are in the same site as the peer. This determination is done by verifying that the retrieved AAAA/A6 RRset for the peer includes one or more global addresses that match the site prefixes advertised by the routers.

This protocol assumes that the routing infrastructure will be used to distribute information about which prefixes belong to the local site. This document only specifies how the site prefixes are distributed from the routers to the hosts on each link. However, other protocols such as [ROUTER-RENUM] might be extended to carry the site prefixes to all routers in a site. The use of the routing infrastructure to carry the site prefixes avoids the "two-faced" issue above - the routers know which part of the network is inside the site thus they can naturally prevent this information from being distributed outside the site.

The protocol is based on each host maintaining a list of all the currently active site prefixes. The site prefixes are periodically advertised in Neighbor Discovery Router Advertisement messages and each prefix has an associated lifetime.

Once a host has a list of prefixes that apply to its site it uses this information to determine if the global addresses contained in a AAAA/A6 RRset is part of its site. If this is the case then the host can use any site-local addresses contained in that AAAA/A6 RRset. Otherwise any site-local addresses contained in that RRset must be ignored. A node should prefer the site-local addresses over the global addresses e.g. by having the applications try the site-local addresses before or instead of the global addresses.

The reverse lookup (from an IPv6 address to a host name) is handled by mapping a site-local address to the corresponding global addresses as a fallback. Thus, if the address being looked up is a site-local address and the reverse lookup for that address fails the host constructs the corresponding global addresses using the list of site prefixes and performs a reverse lookup on those addresses until a match is found.

It is expected that both the forward and reverse lookup rules can be hidden from applications by implementing them as part of the library that handles host name lookups.

3.2. Mobile IP Implications

A mobile node which moves outside its "home site" must maintain the "home site-local addresses" for continued communication with nodes in its "home site". This implies that such a mobile node conceptually will have one interface (for the traffic destined to and from its home site) which is assigned the home site-local addresses in addition to its other interfaces which might be part of the visited site.

A mobile node may choose to autoconfigure site-local addresses in the visited site. However, such addresses add complexity to the mobile node with little or no benefit. Thus it is recommended that mobile nodes only autoconfigure global addresses when moving to links outside its home site.

A mobile node needs to be able to detect when it has moved to a different site. Thus in addition to the regular movement detection in [MOBILE-IPv6] it should inspect the site prefixes in the Router Advertisement messages to determine when it is outside its home site.

The remainder of this section specifies the operation of Mobile IP when the mobile node is outside its home site.

The mobile node needs to retain any site-local addresses it was assigned in its home site, but those site-local addresses should only be used when communicating with nodes in its home site.

The binding updates must use a global address as the care-of-address.

There are no changes needed to Home Agents. The home agent needs to select a proper source address when sending to a global address as is expected of all IPv6 implementations - it should not use a site-local source address when sending to a global destination address.

The only change needed to the Correspondent Nodes is to not use a site-local source address when sending to a global destination: When using a Routing Header to communicate with a mobile node that has a global Care-of-Address the correspondent needs to include a Home Address Option to carry its site-local source address and set the IP source address field to one of its global addresses.

This additional use of the Home Address Option from the correspondents ensures that all traffic to and from the mobile node will have global source addresses. Thus the site-local addresses will be "hidden" in 1) encapsulated headers, 2) routing headers, or 3) home address option.

Packets encapsulated to the mobile node will look like this:

Outer IP header destination address:
Registered care-of-address. A global address.

Outer IP header source address:
Global address assigned to home agent

Inner IP header destination address:
One of the mobile node's home addresses. Likely to be a site-local address.

Inner IP header source address:
Sender of original packet. Likely to be a site-local address.

Packets sent to the mobile node using routing headers:

IP header destination address:
Registered care-of-address. A global address.

IP header source address:
A global address is needed to match the scope of the destination address. (This requirement is added by this specification.)

Routing header:
The mobile node's home address which has been used for this communication e.g. which identifies the TCP connection. Likely to be a site-local address.

Home address option (This requirement is added by the specification.):
The correspondent node's address which has been used for this communication e.g. which identifies the TCP connection. Likely to be a site-local address.

Packets sent from the mobile node to a site-local correspondent address:

IP header destination address:
The correspondent node's global address. If this is not known then the packet must be instead be encapsulated and sent to the (global address of) the home agent which can deliver the packet to the site-local destination address.

IP header source address:
Mobile node's care-of-address. A global address.

Routing header:
The correspondent node's address which has been used for this communication e.g. which identifies the TCP connection. Likely to be a site-local address.

Home address option:
The mobile node's address which has been used for this communication e.g. which identifies the TCP connection. Likely to be a site-local address.

Packets sent from the mobile node to a global correspondent:

IP header destination address:
The correspondent node's global address.

IP header source address:
Mobile node's care-of-address. A global address.

Home address option:
The mobile node's address which has been used for this communication e.g. which identifies the TCP connection. Likely to be a site-local address.

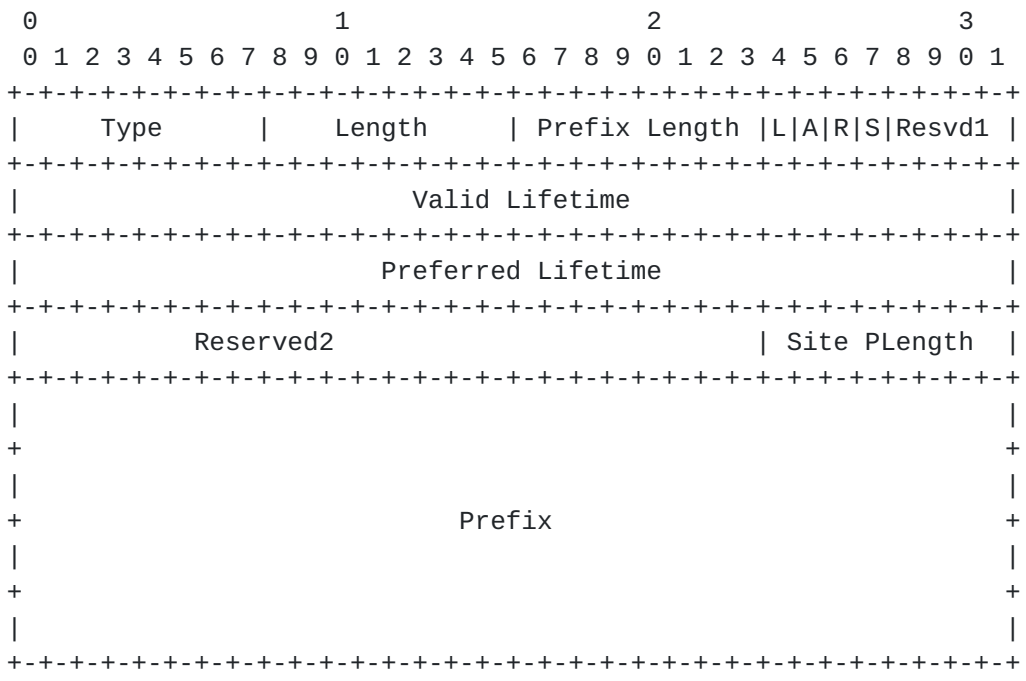
3.3. Assumptions

The protocol assumes that the site uses a consistent subnet numbering scheme across all its global addresses and its site-local addresses.

Thus, for every subnet in the site that uses both global and site-local addresses, the 16-bit subnet ID field [ADDR-ARCH] for the site-local address must have the same value as the Site-Local Aggregator(s) field in the global addresses. However, it is possible that some hosts (or whole subnets) only be configured with site-local addresses in which case they will only be reachable from nodes within the site. Is it also possible that some hosts (or subnets) only be configured with global addresses in which case they will not benefit from use of site locals.

4. UPDATED PREFIX OPTION FORMAT

The protocol adds two new fields using previously reserved parts of the Prefix Information Option defined in [DISCOVERY].



New fields:

- S — 1-bit site prefix flag. When set indicates that this prefix, in addition to what might be specified by the L and A flags, should be used to create site-local addresses when an address matches the first Site PLength part of the prefix.

- Site PLength — 8-bit unsigned integer. This Site Prefix Length is only valid when the S flag is set. The number of leading bits in the Prefix that are valid. The value ranges from 0 to 128.

The defined format above allows a single Prefix Information option to carry a subnet prefix used for on-link and/or stateless address autoconfiguration [ADDRCONF] together with a site prefix since the site prefix(es) are normally sub-prefixes of the subnet prefixes.

For example, if the subnet prefix is
        2000:1:2:653a::0/64
and the site prefix is:
        2000:1:2::0/48
this can be encoded in a single Prefix Information option with Prefix

Length being 64, Site PLength being 48, the Prefix being 2000:1:2:653a::0, and the S flag being set.

5. CONCEPTUAL VARIABLES

This document makes use of internal conceptual variables to describe protocol behavior and external variables that an implementation must allow system administrators to change. The specific variable names, how their values change, and how their settings influence protocol behavior are provided to demonstrate protocol behavior. An implementation is not required to have them in the exact form described here, so long as its external behavior is consistent with that described in this document.

Hosts will need to maintain the following pieces of information. Like the prefix related information specified in [DISCOVERY] this information recorded per interface but, except for multi-sited nodes, used as a global list being the union of the information over all interfaces. Multi-sited nodes need to use the information separately for each site i.e. form the union over all interfaces that are attached to a particular site. See Section 9 how multi-sited nodes operate.

Site Prefix List

A list of the site prefixes that have been received in Router Advertisement messages that have not yet timed out. Each entry has an associated invalidation timer value (extracted from the advertisement) used to expire site prefixes when they become invalid. A special "infinity" timer value specifies that a prefix remains valid forever, unless a new (finite) value is received in a subsequent advertisement.

Note that the Site Prefix List is separate from the list of on-link prefixes called Prefix List in [DISCOVERY].

The conceptual Router variable called AdvPrefixList in [DISCOVERY] is extended to also contain site prefixes. Conceptually this can be done by having each prefix both contain a AdvSubnetPrefixLength (the length of the AdvPrefix as specified in [DISCOVERY]) and a AdvSitePrefixLength field. If one of the length fields is zero the prefix is not used as a on-link and/or addrconf prefix or a site prefix, respectively. The same lifetime values will apply to both

the subnet and site prefix aspects of a prefix in the AdvPrefixList.

The above are conceptual variables; Implementations are free to implement the router variables as a separate list for the site prefixes and the existing Neighbor Discovery AdvPrefixList for subnet prefixes. However, it is desirable that such implementations still use a single Prefix Information option to encode both a site and a subnet prefix when the site prefix is just a sub-prefix of the subnet prefix (unless the lifetimes need to be different for the subnet and site prefixes).

6. SENDING RULES

When a router is sending Prefix options as part of sending Router Advertisement messages, in addition to the rules in [DISCOVERY], the router performs the following operations:

- If the AdvSitePrefixLength field in the AdvPrefixList entry is non-zero set the S flag in the Prefix option to one and set the Site PLength to the AdvSitePrefixLength.

- Only if the AdvSubnetPrefixLength field is non-zero should the L-bit and the A-bit be set from the AdvOnLinkFlag and the AdvAutonomousFlag fields, respectively.

- The Prefix field and the lifetime fields are set is specified in [DISCOVERY].

7. RECEIVING RULES

The host receiving a valid Router Advertisement follows the rules as specified in [DISCOVERY] with the following additions when processing each received Prefix Information option. For each prefix that has the S-flag set:

- If the Site PLength is zero then do nothing further.

- If the prefix is a link-local or a site-local prefix then do nothing further.

- If the prefix is a multicast address then do nothing further.

- If the prefix is not already present in the Site Prefix List and the Valid Lifetime is zero, then do nothing further.

- If the prefix is not already present in the Site Prefix List and

the Valid Lifetime is non-zero, then create a new entry for the prefix in the Site Prefix List and initialize its invalidation timer to the Valid Lifetime value in the Prefix Information option.

- o If the prefix is already present in the host's Site Prefix List as the result of a previously-received advertisement, then reset its invalidation timer to the Valid Lifetime value in the Prefix Information option. If the new Lifetime value is zero, then immediately remove the prefix from the Site Prefix List.

The bits in the Prefix after the first Site PLength bits MUST be ignored when the prefix is entered in the Site Prefix List and/or when it is compared against other site prefixes. These bits might be non-zero when the Prefix option carries a subnet prefix in addition to a site prefix.

Timing out a site prefix from the Site Prefix List SHOULD NOT affect any existing communication. New communication will use the updated Site Prefix List after performing a host name lookup.

8. USING THE SITE PREFIXES

The following rules apply when a node looks up host names and addresses in a name service such as DNS.

8.1. Host Name Lookups

The node will inspect the AAAA/A6 RRset returned from DNS to check if one or more of the global addresses belong to the same site as itself. This is done by comparing all the global addresses against all the prefixes in the Site Prefix List. If there are no matches then the site-local addresses in the RRset must not be used. If there are one or more matches then the node should prefer using the site-local address(es) over the global addresses. This can be done by sorting the addresses before they are returned to the application and excluding the addresses that are subsumed by the site-local addresses.

It is important that the site-local addresses are first in the sorted list so that the applications try the site-local addresses before any global address. Also, the matched global addresses are removed from the list in order to prevent the applications from using global addresses for communication that is local to the site.

A possible algorithm for doing these comparisons is as follows:

1) Assume the name service returns the global addresses G1, G2, G3, ... Gn and the site-local addresses SL1, SL2, ... SLk. Assume the prefixes in the Site Prefix List are SP1, SP2, ... SPm. The Site PLength of each of the prefixes is Length(SPj).

2) If n is zero (i.e. no global addresses were returned) just hand all the site local addresses SL1, .. SLk to the application.

3) Otherwise; for each Gi compare it against all the SPj. If the first Length(SPj) bits of Gi are equal to the first Length(SPj) bits of SPj then we have a match. If there is a match then suppress Gi (do not hand it to the application).

3a) If there is one or more matches then give the application the site-local addresses SL1, SL2, ... SLk inserted before the Gi addresses that were not suppressed by rule 2)

3b) If there is no match the result is that the application only gets the global addresses G1, ... Gn.

For example, if the name service returns these addresses for a multihomed node:

```
2837:a:b:987:X:Y:W:Z1
2000:1:2:987:X:Y:W:Z1
fec0::987:X:Y:W:Z1
2837:a:b:34:X:Y:W:Z2
2000:1:2:34:X:Y:W:Z2
fec0::34:X:Y:W:Z2
2abc:77:66:23:X:Y:W:Z3
```

and the prefixes in the Site Prefix List are:

```
2837:a:b::0/48
2000:1:2::0/48
```

The resulting list that the application should use should be:

```
fec0::987:X:Y:W:Z1
fec0::34:X:Y:W:Z2
2abc:77:66:23:X:Y:W:Z3
```

If there is no match (e.g., the Site Prefix List is empty) the resulting list that the application should use should be:

```
2837:a:b:987:X:Y:W:Z1
2000:1:2:987:X:Y:W:Z1
2837:a:b:34:X:Y:W:Z2
2000:1:2:34:X:Y:W:Z2
2abc:77:66:23:X:Y:W:Z3
```

8.2. IPv6 Address Lookups

It is not sufficient to handle the forward lookup. For instance, the node that receives packets and/or connections from a site-local address might have the desire to perform a reverse lookup to get a host name. Thus these rules allow such a reverse lookup to succeed as long as the Site Prefix List contains all the prefixes that apply to the site.

A possible algorithm for doing this is as follows:

1) Assume the site-local address is SL and the prefixes in the Site Prefix List are SP1, SP2, ... SPm. The Site PLength of each of the prefixes is Length(SPj).

2) First perform a regular reverse lookup of the IPv6 address. If the lookup succeeds return success to the application. If the lookup fails and the IPv6 address is not a site-local address report the failure to the application.

3) When the reverse lookup of a site-local address fails use the Site Prefix List to construct global addresses corresponding to the site-local address. This is done by taking each entry in the Site Prefix List and using it to construct a global address. For each of the SPj concatenate the first Length(SPj) bits from SPj and the last (128 - Length(SPj)) bits from SL to form a new address. Look up each of the resulting addresses until a match is found.

For example, if the site-local address is:

```
fec0::987:X:Y:W:Z1
```

and the prefixes in the Site Prefix List are:

```
2837:a:b::0/48
2000:1:2::0/48
```

The addresses that should be tried in the reverse lookup are:

```
fec0::987:X:Y:W:Z1
2837:a:b:987:X:Y:W:Z1
2000:1:2:987:X:Y:W:Z1
```

9. MULTI-SITED NODES

A node potentially connected to multiple sites needs to be able to

- o Detect that is it multi-sited.
- o Be configured with the appropriate AAAA/A6 records in the DNS.
- o Be able to distinguish between the different sites when originating applications.

An alternative to multi-sited nodes is to not use any site-local addresses for the node "close" to the site boundary (i.e. and not list any site-local addresses in the DNS for that node). This will force all traffic to and from that node to use global addresses (except those few cases where link-local addresses are used).

9.1. Detecting that a Node is Multi-sited

A possible algorithm for detecting when a node is multi-sited is as follows:

1) Inspect the Site Prefix List for all interfaces.
2) If an interface has no Site Prefix List entry ignore that interface.
3) If two or more interfaces have one or more common Site Prefix List entries group those interfaces together.
4) If the result is more than one group of interfaces the node is considered to be multi-sited.

If the node detects that it is multi-sited and does not contain support for site-local addresses in this environment it must at a minimum log an event. It may also attempt to remove any site-local addresses assigned to it from the DNS to avoid communication failure should other nodes attempt to communicate with it using site-local addresses.

9.2. Address Records for Multi-sited Nodes

A given AAAA/A6 RRset can only contain site-local addresses for one site, since the site is implicit in the association between the global and site-local addresses contained in the same RRset.

This implies that a multi-sited node that have a single domain name with AAAA/A6 records for interfaces in multiple sites can not have site-local addresses in that RRset.

The multi-sited node can have a different domain name for each site to which it is connected, in order to enter site-local AAAA/A6 records in the DNS.

For example, a multi-sited node connected to two sites:

```
Site1:
        Address 2000:1:2:987:X:Y:W:Z1   (A1)
        Address fec0::987:X:Y:W:Z1      (A2)
        Address 2000:1:2:34:X:Y:W:Z2    (A3)
        Address fec0::34:X:Y:W:Z2       (A4)
        Site prefix 2000:1:2::0/48
Site2:
        Address 4444:a:b:34:X1:Y1:W1:Z3 (A5)
        Address fec0:::34:X1:Y1:W1:Z3   (A6)
        Site prefix 4444:a:b::0/48
```

This node could have 3 different host names:

foo.bar.site1.tla which list the AAAA/A6 records A1 through A4

foo.site2.tla which list the AAAA/A6 records A5 through A6

foo.net which list the global AAAA/A6 records A1, A3 and A5.

9.3. Distinguishing Between Different Sites

A multi-sited node needs to take additional care in applications and in the protocol stack to qualify any site-local addresses with the site unless all applications always associate an interface with each IP address. (For instance, the use of getaddrinfo() as specified in [BSD-API] allows the transparent passing of a site-id to the TCP/IP stack in the sin6_scope_id without modifying the applications.) It is recommended (but not required) that implementations which operate at site boundaries have such support.

If the implementation does not have such support it must not pass any site-local addresses to the applications since it will not be

possible for the IP layer to determine which site (i.e., the set of interfaces attached to a site) to originate the packet on.

10. SECURITY CONSIDERATIONS

Router Advertisements are not required to be authenticated and even if they are authenticated it is unclear whether or not there would be a mechanisms to verify the authority of a particular node to send Router Advertisements.

Neighbor Discovery uses the rule of HopCount 255 (set to 255 on transmit and verified to be 255 on reception) to drop any Neighbor Discovery packets that are sent non-neighboring nodes. This limits any attack using ND to the neighbors.

Without authentication and authorization this new mechanisms introduces a new type of denial of service attack. A node on the link can send a router advertisement listing site prefixes that are in fact not part of the site. For instance, it could advertise some other sites prefix as a site prefix. Such an attack would result in all nodes on the link to fail initiate any new communication with any node in that site since they would accept the site-local AAAA/A6 records.

Also there is the possibility to return incorrect information for the reverse lookup of IPv6 addresses. A node on the link can send a router advertisement listing site prefixes that are in fact not part of the site. For instance, it could advertise an incorrect site prefix (e.g. a:b::0/48) which would make the reverse lookup of the site local address fec0::X lookup a:b::X.

This could be viewed as allowing some form of indirect spoofing of the addresses returned by the DNS independent whether or not the DNS itself is secure. Thus introducing a secure DNS [DNSsec] would not remove this form of "address spoofing". However, it seems like this threat is no worse than the other threats in [DISCOVERY] where any node on the link can intercept all packets sent on the link.

The packets used to discover site prefixes, just like all other Neighbor Discovery protocol packet exchanges, can be authenticated using the IP Authentication Header [IPv6-AUTH]. A node SHOULD include an Authentication Header when sending Neighbor Discovery packets if a security association for use with the IP Authentication Header exists for the destination address. The security associations may have been created through manual configuration or through the operation of some key management protocol.

Received Authentication Headers in these packets, just like all Neighbor Discovery packets, MUST be verified for correctness and packets with incorrect authentication MUST be ignored.

Confidentiality issues are addressed by the IP Security Architecture and the IP Encapsulating Security Payload documents [IPv6-SA, IPv6-ESP].

REFERENCES

[KEYWORDS] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[IPv6] S. Deering, R. Hinden, Editors, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[ADDR-ARCH] S. Deering, R. Hinden, Editors, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

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

[ADDR-TODAY] B. Carpenter, J. Crowcroft, Y. Rekhter, "IPv4 Address Behavior Today", RFC 2101, February 1997.

[GSE-EVAL] M. Crawford, A. Mankin, T. Narten, J. Stewart, L. Zhang, "Separating Identifiers and Locators in Addresses: An Analysis of the GSE Proposal for IPv6", Internet Draft, draft-ietf-ipngwg-esd-analysis-04.txt.

[ROUTER-RENUM] M. Crawford, and R. Hinden, "Router Renumbering for IPv6", Internet Draft, draft-ietf-ipngwg-router-renum-08.txt.

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

[IPv6-SA] R. Atkinson. "Security Architecture for the Internet Protocol". RFC 2401, November 1998.

[IPv6-AUTH] R. Atkinson. "IP Authentication Header", RFC 2402,

November 1998.

[IPv6-ESP] R. Atkinson. "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[DNSsec] D. Eastlake, C. Kaufman, "Domain Name System Security Extensions", RFC 2535, March 1999.

[MOBILE-IPv6] D.B. Johnson, C. Perkins, "Mobility Support in IPv6", Internet Draft, draft-ietf-mobileip-ipv6-07.txt, March 1999.

[BSD-API] R. Gilligan, S. Thomson, J. Bound, W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 2553, March 19999.

AUTHOR'S ADDRESS

Erik Nordmark
Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
USA

phone: +1 650 786 5166
fax: +1 650 786 5896
email: nordmark@sun.com

APPENDIX A: CHANGES SINCE PREVIOUS DRAFT

The following changes have been made since version 02 of the draft.

- Moved the Site PLength field in the option format to make it easier for [ROUTER-RENUM] to include the field in its option format.
- Changed the rules about suppressing global addresses to only suppress the ones that match the site prefixes.
- Refined the rules and text to handle case when site not connected to the Internet i.e. when the DNS returns only site-local addresses.
- Specified that a multi-sited node that have a single domain name can use site-local addresses on at most one site. This is to ensure that one AAAA/A6 RRset contains site-local addresses for at

most one site. The multihomed node must have a different domain name for each site in it wants to use site-local addresses in all its sites.
- Defined "multi-sited node"
- Clarified that a multi-sited node can, instead of ignoring all site-locals, pass the full AAAA/A6 RRset (include the site-local addresses) for nodes in directly attached sites *IF* the applications and protocol stack can ensure that the communication will use the proper site. (For instance, using mechanisms like getaddrinfo() and the sin6_scope_id to pass the local site identifier to the protocol stack transparent to the application.)
- Changed references from AAAA to AAAA/A6.

The following changes have been made since version 01 of the draft.

- Stated the assumptions on what a "site" is and how it is configured.

- Changed the document to store site-local addresses in the DNS and use filtering do ignore site-local addresses unless the sender and receiver can be determined to belong to the same site.

- Added text describing interaction with mobile IP.

- Added rules for ignoring site-local entries from the DNS

- Make "turn off at site boundary" implementation dependent.

- Changed 'S' bit in prefix option not to conflict with [MOBILE-IPv6].

The following changes have been made since version 00 of the draft.

- Removed mention of routing protocols.

- Made the formed site-local addresses replace the global addresses in the list returned to the application. This change prevents the "accidental" use of a global address when the application tries all of the returned addresses and for whatever reason it could not reach the node when it tried the site-local address(es).

- Added text describing how to the mechanism is automatically disabled on nodes which are Multihomed to multiple sites.

- Updated list of open issues.