Network Working Group
Internet-Draft
Intended status: BCP
Expires: April 10, 2011

B. Carpenter
Univ. of Auckland
S. Amante
Level 3

October 7, 2010

Using the IPv6 flow label for equal cost multipath routing and link aggregation in tunnels
draft-carpenter-flow-ecmp-03

Abstract

The IPv6 flow label has certain restrictions on its use. This document describes how those restrictions apply when using the flow label for load balancing by equal cost multipath routing, and for link aggregation, particularly for tunneled traffic.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on April 10, 2011.

Table of Contents

1. Introduction

When several network paths between the same two nodes are known by the routing system to be equally good (in terms of capacity and latency), it may be desirable to share traffic among them. Two such techniques are known as equal cost multipath routing (ECMP) and link aggregation (LAG) [IEEE802.1AX]. There are of course numerous possible approaches to this, but certain goals need to be met:

- Roughly equal share of traffic on each path.
- Work-conserving method (no idle time when queue is non-empty).
- Minimize or avoid out-of-order delivery for individual traffic flows.

There is some conflict between these goals: for example, strictly avoiding idle time could cause a small packet sent on an idle path to overtake a bigger packet from the same flow, causing out-of-order delivery.

One lightweight approach to ECMP or LAG is this: if there are N equally good paths to choose from, then form a modulo(N) hash [RFC2991] from a consistent set of fields in each packet header, and use the resulting value to select a particular path. If the hash function is chosen so that the hash values have a uniform statistical distribution, this method will share traffic roughly equally between the N paths. If the header fields included in the hash are consistent, all packets from a given flow will generate the same hash, so out-of-order delivery will not occur. Assuming a large number of unique flows are involved, it is also probable that the method will be work-conserving, since the queue for each link will remain non-empty.

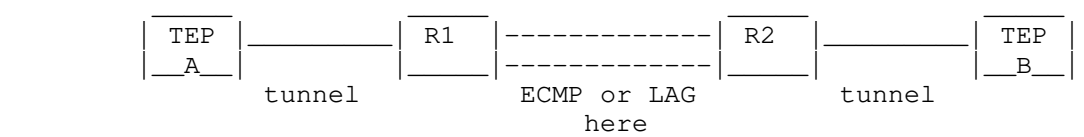
The question with such a method is which IP header fields are chosen to identify a flow and, consequently, are used as input keys to a modulo(N) hash algorithm.

In the remainder of this document, we will use the term "flow" to represent a sequence of packets that may be identified by either the source and destination IP addresses alone {2-tuple} or the source and destination IP addresses, protocol and source and destination port numbers {5-tuple}. It should be noted that the latter is more specifically referred to as a "microflow" in [RFC2474], but this term is not used in connection with the flow label in [RFC3697].

The question with such a method, then, is which IP header fields to include to identify a flow. A minimal choice in the routing system is simply to use a hash of the source and destination IP addresses, i.e., the 2-tuple. This is necessary and sufficient to avoid out-of-order delivery, and with a wide variety of sources and destinations,

as one finds in the core of the network, sometimes sufficient to achieve work-conserving load sharing. In practice, implementations often use the 5-tuple {dest addr, source addr, protocol, dest port, source port} as input keys to the hash function, to maximize the probability of evenly sharing traffic over the equal cost paths. However, including transport layer information as input keys to a hash may be a problem for IPv4 fragments [RFC2991]. In addition, protocol and destination port numbers in the hash will not only make the hash slightly more expensive to compute, but will not particularly improve the hash distribution, due to the prevalence of well known port numbers and popular protocol numbers. Ephemeral ports, on the other hand, are quite well distributed [Lee10]. In the case of IPv6, protocol numbers are particularly inconvenient due to the variable placement of and variable length of next-headers. In addition, [RFC2460] recommends that all next-headers, except hop-by-hop options, should not be inspected by intermediate nodes in the network, presumably to make introduction of new next-headers more straightforward.

The situation is different in tunneled scenarios. Identifying a flow inside the tunnel is more complicated, particularly because nearly all hardware can only identify flows based on information contained in the outermost IP header. Assume that traffic from many sources to many destinations is aggregated in a single IP-in-IP tunnel from tunnel end point (TEP) A to TEP B (see figure). Then all the packets forming the tunnel have outer source address A and outer destination address B. In all probability they also have the same port and protocol numbers. If there are multiple paths between routers R1 and R2, and ECMP or LAG is applied to choose a particular path, the 5-tuple and its hash will be constant and no load sharing will be achieved. If there is much tunnel traffic, this will result in a high probability of congestion on one of the paths between R1 and R2.



Also, for IPv6, the total number of bits in the 5-tuple is quite large (296), as well as inconvenient to extract due to the next-header placement. This may be challenging for some hardware implementations, raising the potential that network equipment vendors might sacrifice the length of the fields extracted from an IPv6 header. The question therefore arises whether the 20-bit flow label in IPv6 packets would be suitable for use as input to an ECMP or LAG hash algorithm. If it could be used in place of the port numbers and

protocol number in the 5-tuple, the hash calculation would be simplified.

The flow label is left experimental by [RFC2460] but is better defined by [RFC3697]. We quote three rules from that RFC:

1. "The Flow Label value set by the source MUST be delivered unchanged to the destination node(s)."
2. "IPv6 nodes MUST NOT assume any mathematical or other properties of the Flow Label values assigned by source nodes."
3. "Router performance SHOULD NOT be dependent on the distribution of the Flow Label values. Especially, the Flow Label bits alone make poor material for a hash key."

These rules, especially the last one, have caused designers to hesitate about using the flow label in support of ECMP or LAG. The fact is today that most nodes set a zero value in the flow label, and the first rule definitely forbids the routing system from changing the flow label once a packet has left the source node. Considering normal IPv6 traffic, the fact that the flow label is typically zero means that it would add no value to an ECMP or LAG hash. But neither would it do any harm to the distribution of the hash values. If the community at some stage agrees to set pseudo-random flow labels in the majority of traffic flows, this would add to the value of the hash.

However, in the case of an IP-in-IPv6 tunnel, the TEP is itself the source node of the outer packets. Therefore, a TEP may freely set a flow label in the outer IPv6 header of the packets it sends into the tunnel. In particular, it may follow the [RFC3697] suggestion to set a pseudo-random value.

The second two rules quoted above need to be seen in the context of [RFC3697], which assumes that routers using the flow label in some way will be involved in some sort of method of establishing flow state: "To enable flow-specific treatment, flow state needs to be established on all or a subset of the IPv6 nodes on the path from the source to the destination(s)." The RFC should perhaps have made clear that a router that has participated in flow state establishment can rely on properties of the resulting flow label values without further signaling. If a router knows these properties, rule 2 is irrelevant, and it can choose to deviate from rule 3.

In the tunneling situation sketched above, routers R1 and R2 can rely on the flow labels set by TEP A and TEP B being assigned by a known method. This allows a safe ECMP or LAG method to be based on the flow label without breaching [RFC3697].

2. Normative Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Guidelines

We assume that the routers supporting ECMP or LAG (R1 and R2 in the above figure) are unaware that they are handling tunneled traffic. If it is desired to include the IPv6 flow label in an ECMP or LAG hash in the tunneled scenario shown above, the following guidelines apply:

- Inner packets MUST be encapsulated in an outer IPv6 packet whose source and destination addresses are those of the tunnel end points (TEPs).
- The flow label in the outer packet SHOULD be set by the sending TEP to a pseudo-random 20-bit value in accordance with [RFC3697]. The same flow label value MUST be used for all packets in a single user flow, as determined by the IP header fields of the inner packet.
  - Note that this rule is a SHOULD rather than a MUST, to permit individual implementers to take an alternative approach if they wish to do so. Such an alternative MUST conform to [RFC3697].
- The sending TEP MUST classify all packets into flows, once it has determined that they should enter a given tunnel, and then write the relevant flow label into the outer IPv6 header. A user flow could be identified by the ingress TEP most simply by its {destination, source} address pair (coarse) or by its 5-tuple {dest addr, source addr, protocol, dest port, source port} (fine). This is an implementation detail in the sending TEP.
  - It might be possible to make this classifier stateless, by using a suitable 20 bit hash of the inner IP header's 2-tuple or 5-tuple as the pseudo-random flow label value.
- At intermediate router(s) that perform load distribution of tunneled packets whose source address is a TEP, the hash algorithm used to determine the outgoing component-link in an ECMP and/or LAG toward the next-hop MUST minimally include the triple {dest addr, source addr, flow label} to meet the [RFC3697] rules.
  - Intermediate router(s) MAY also include {protocol, dest port, source port} as input keys to the ECMP and/or LAG hash algorithms, to provide sufficient entropy in cases where the flow-label is currently set to zero.

4. Security Considerations

The flow label is not protected in any way and can be forged by an on-path attacker. Off-path attackers are unlikely to guess a valid flow label if a pseudo-random value is used. In either case, the worst an attacker could do against ECMP or LAG is to attempt to selectively overload a particular path. For further discussion, see [RFC3697].

5. IANA Considerations

This document requests no action by IANA.

6. Acknowledgements

This document was suggest by corridor discussions at IETF76. Joel Halpern made crucial comments on an early version. We are grateful to Qinwen Hu for general discussion about the flow label. Valuable comments and contributions were made by Jarno Rajahalme, Brian Haberman, Sheng Jiang, and others.

This document was produced using the xml2rfc tool [RFC2629].

7. Change log

draft-carpenter-flow-ecmp-03: clarifications after further comments, 2010-10-07

draft-carpenter-flow-ecmp-02: updated after IETF77 discussion, especially adding LAG, changed to BCP language, added second author, 2010-04-14

draft-carpenter-flow-ecmp-01: updated after comments, 2010-02-18

draft-carpenter-flow-ecmp-00: original version, 2010-01-19

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6

(IPv6) Specification", RFC 2460, December 1998.

[RFC3697] Rajahalme, J., Conta, A., Carpenter, B., and S. Deering, "IPv6 Flow Label Specification", RFC 3697, March 2004.

8.2. Informative References

[IEEE802.1AX] Institute of Electrical and Electronics Engineers, "Link Aggregation", IEEE Standard 802.1AX-2008, 2008.

[Lee10] Lee, D., Carpenter, B., and N. Brownlee, "Observations of UDP to TCP Ratio and Port Numbers", Fifth International Conference on Internet Monitoring and Protection ICIMP 2010, May 2010, <http://www.cs.auckland.ac.nz/~brian/udptcp-paper-cam-submit.pdf>.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC2991] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, November 2000.

Authors' Addresses

Brian Carpenter
Department of Computer Science
University of Auckland
PB 92019
Auckland, 1142
New Zealand

Email: brian.e.carpenter@gmail.com

Shane Amante
Level 3 Communications, LLC
1025 Eldorado Blvd
Broomfield, CO 80021
USA

Email: shane@level3.net

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 1, 2011

W. Dec, Ed.
Cisco Systems
T. Mrugalski
Gdansk University of Technology
T. Sun
China Mobile
B. Sarikaya
Huawei USA

September 28, 2010

DHCPv6 Route Option
draft-dec-dhcpv6-route-option-05

Abstract

This document describes DHCPv6 Route Options for provisioning IPv6 routes on nodes with DHCPv6 clients. This is expected to improve the ability of an operator to configure and influence a node's ability to pick an appropriate route to a destination when this node is multi-homed and where other means of route configuration may be impractical.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on April 1, 2011.

Copyright Notice

Table of Contents

1. Introduction

The Neighbor Discovery (ICMPv6) protocol [RFC4861] provides a mechanism for hosts to discover one or more default routers on a directly connected network segment. Extensions to the protocol defined in [RFC4191] allow hosts to discover the preferences for multiple default routers on a given link, as well as any specific routes advertised by these routers. This allows network administrators to better handle multi-homed host topologies and influence the route selection by the host. This ND based mechanism however is sub optimal or impractical in some multi-homing scenarios, where DHCPv6 is seen to be more viable.

This draft defines the DHCPv6 Route Option for provisioning IPv6 routes on DHCPv6 clients. The proposed option is primarily envisaged for use by DHCPv6 client nodes that are capable of making basic IP routing decisions and maintaining an IPv6 routing table, broadly in line with the capabilities of a generic host as described in [RFC4191].

Throughout the document the words node and client are used as a reference to the device with such routing capabilities, hosting the DHCPv6 client software. The route information is taken to be equivalent to static routing, and limited in the number of required routes to a handful.

2. Problem overview

The following scenario is used to illustrate the problem as found in multi-homed residential access networks. It is duly noted that the problem is not specific to IPv6, occurring also with IPv4, where it is today solved by means of DHCPv4 classless route information option [RFC3442], or alternative configuration mechanisms.

In multi-homed networks, a given user’s node may be connected to more than one gateways. Such connectivity may be realized by means of dedicated physical or logical links that may also be shared with other users nodes. In such multi-homed networks it is quite common for the network operator to offer the delivery of a particular type of IP service via a particular gateway, where the service can be characterised by means of specific destination IP network prefixes. Thus, from an IP routing perspective in order for the user node to select the appropriate gateway for a given destination IP prefix, recourse needs to be made to classic longest destination match IP routing, with the node acquiring such prefixes into its routing table. This is typically the remit of dynamic Internal Gateway Protocols (IGPs), which however are rarely used by operators in

residential access networks. This is primarily due to operational costs and a desire to contain the complexity of user nodes and IP Edge devices to a minimum. While, IP Route configuration may be achieved using the ICMPv6 extensions defined in [RFC4191], this mechanism does not lend itself to other operational constraints such as the desire to control the route information on a per node basis, the ability to determine whether a given node is actually capable of receiveing/processing such route information. A preferred mechanism, and one that additionally also lends itself to centralized management independent of the management of the gateways, is that of using the DHCP protocol for conveying route information to the nodes.

3. DHCPv6 Based Solution

A DHCPv6 based solution allows an operator an on demand and node specific means of configuring static routing information. Such a solution also fits into network environments where the operator prefers to manage RG configuration information from a centralized DHCP server. [I-D.troan-multihoming-without-nat66] provides additional background to the need for a DHCPv6 solution to the problem.

In terms of the high level operation of the solution defined in this draft, a DHCPv6 client interested in obtaining routing information request the route option using the DHCPv6 Option Request Option (ORO) sent to a server. A Server, when configured to do so, provides the requested route information as part of a nested options structure covering; the next-hop address; the destination prefix; the route metric; any additional options applicable to the destination or next-hop. The overall DHCPv6 design follow a similar approach to that used in the design of the IA_NA, IA_TA and IA_PD options in [RFC3633]

4. DHCPv6 Route Option

A DHCPv6 client interested in obtaining routing information includes the OPTION_IA_RT in its DHCPv6 Option Request Option (ORO) sent to a server. A Server, when configured to do so, provides the requested route information using the OPTION_IA_RT option. So as to allow the route option to be both extensible, as well as conveying detailed info for routes, use is made of a nested options structure. An IA_RT conveys one or more OPTION_NEXT_HOP options that specify the IPv6 next hop addresses. Each OPTION_NEXT_HOP conveys in turn one or more OPTION_RT_PREFIX options that represents the IPv6 destination prefixes reachable via the given next hop. The Formats of the OPTION_IA_RT, OPTION_NEXT_HOP and OPTION_RT_PREFIX are defined in the following sub-sections

The DHCPv6 Route Option format borrows from the principles of the Route Information Option defined in [RFC4191]. One notable exception with respect to [RFC4191] is however that a Route Lifetime element is not defined. The information conveyed by the DHCPv6 Route Option is considered valid until changed or refreshed by general events that trigger DHCPv6 or route table state changes on a node, thus not requiring a specific route lifetime. In the event that it is desired for the client to request a refresh of the route information (and other stateless DHCPv6 options), use of the generic DHCPv6 Information Refresh Time Option, as specified in [RFC4242] is envisaged.

4.1. DHCPv6 Route Option Format

To separate routing information from other options conveyed in a DHCPv6 message, the DHCPv6 Route Option is defined and is used to convey to a client one or more IPv6 routes. Each IPv6 route consists of an IPv6 next hop address, an IPv6 destination prefix (a.k.a. The destination subnet), and a host preference value for the route. Elements of such route (e.g. Next hops and prefixes associated with them) are conveyed in IA_RT's options, rather than in the IA_RT option itself.

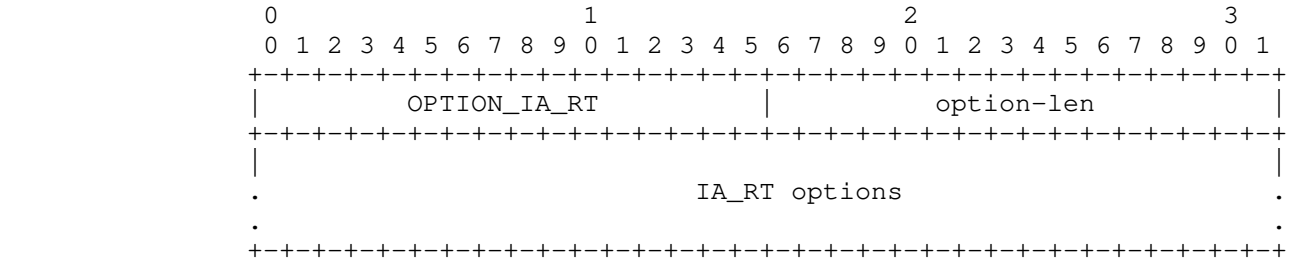


Figure 1: IPv6 Routes Option Format

option-code: OPTION_IA_RT (TBD).

option-len: Length of the IA_RT options field.

IA_RT options: Options associated with this IA_RT. This includes, but is not limited to, OPTION_NEXT_HOP options that specify next hop addresses.

The Route option MUST NOT appear in the following DHCPv6 messages: Solicit, Request, Renew, Rebind, Information-Request. The Route Option MAY appear in ADVERTISE and REPLY messages.

Discussion: Traditionally, grouping options (IA_NA, IA_TA and IA_RD) contain an identifier field (IAID) that must be unique among

identifiers generated by one client. It is used to differentiate between several options of the same type (e.g. several IA_NA options) that may be used simultaneously. However, it is assumed that client will never use more than one IA_RT option therefore such an identifier is not needed.

4.2. Next Hop Option Format

The Next Hop Option defines the IPv6 address of the next hop, usually corresponding to a specific next-hop router. For each next hop address there are one or more prefixes reachable via that next hop.

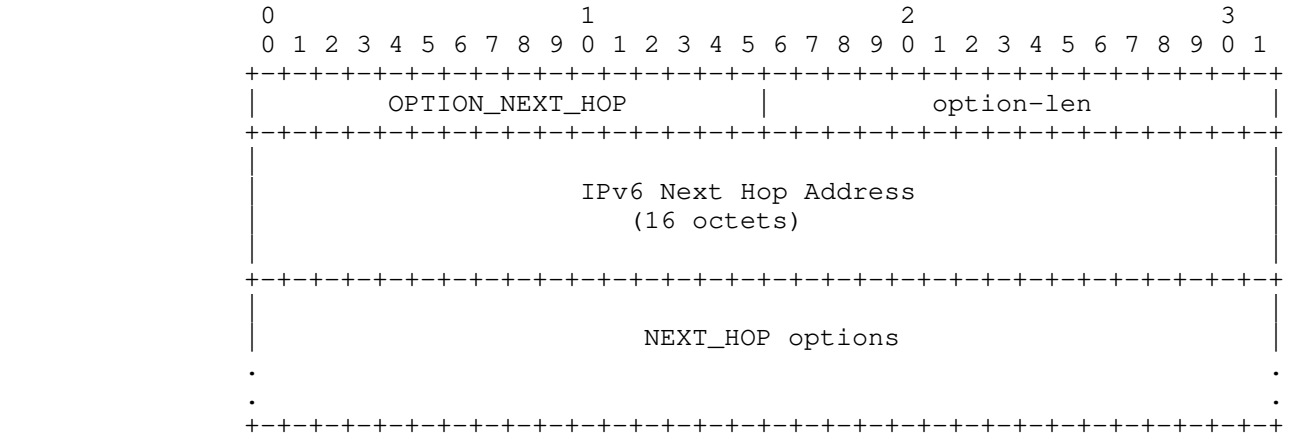


Figure 2: IPv6 Route Option Format

option-code: OPTION_NEXT_HOP (TBD).

option-len: 16 + Length of NEXT_HOP options field.

IPv6 Next Hop Address: 16 octet long field that specified IPv6 address of the next hop.

NEXT_HOP options: Options associated with this Next Hop. This includes, but is not limited to, OPTION_RT_PREFIX options that specify prefixes available via specified next hop.

4.3. Route Prefix Option Format

The Route Prefix Option is used to convey information about a single prefix that represents the destination network. The Route Prefix Option is used as a sub-option in the previously defined Next Hop Option.

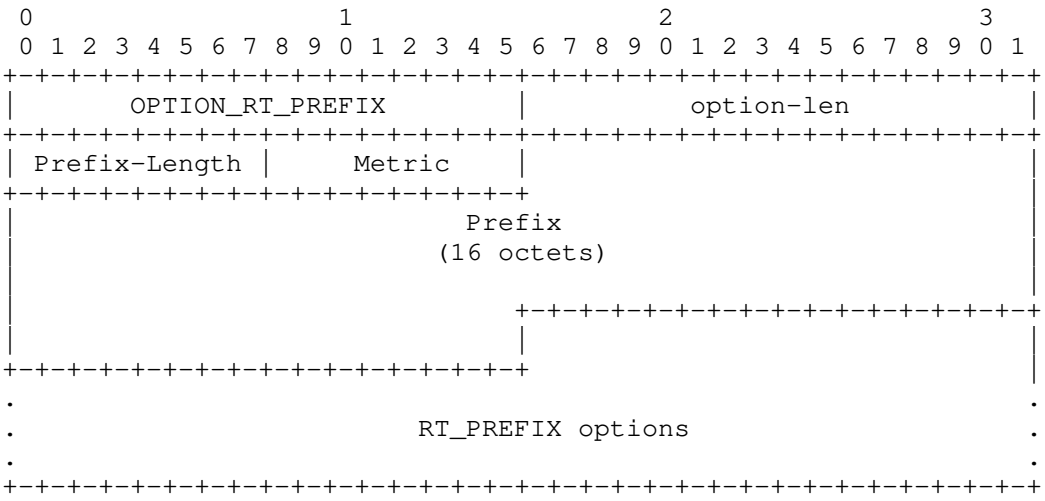


Figure 3: Route Prefix Option Format

option-code: OPTION_RT_PREFIX (TBD).

option-len: 18 + length of RT_PREFIX options.

Prefix Length: 8-bit unsigned integer. The length in bits of the IP Prefix. The value ranges from 0 to 128. This field represents the number of valid leading bits in the prefix.

Metric: Route Metric. 8-bit signed integer. The Route Metric indicates whether to prefer the next hop associated with this prefix over others, when multiple identical prefixes (for different next hops) have been received.

Prefix: Fixed length 16 octet field containing an IPv6 prefix.

RT_PREFIX options: Options specific to this particular prefix.

5. DHCPv6 Server Behavior

When configured to do so s DHCPv6 server shall provide the Routes Option in ADVERTISE and REPLY messages sent to a client that requested the route option. Each Next Hop Option sent by the server must convey at least one Route Prefix Option.

Servers SHOULD NOT send Route Option to clients that did not explicitly requested it, using the ORO.

Servers MUST NOT send Route Option in messages other than ADVERTISE or REPLY.

Servers MAY also include Status Code Option, defined in Section 22.13 of the [RFC3315] to indicate the status of the operation.

Servers MUST include the Status Code Option, if the requested routing configuration was not successful and SHOULD use status codes as defined in [RFC3315] and [RFC3633].

Discussion: How should server indicate that there are no specific routes for this particular client? The reasonable behavior is to return empty IA_RT option, possibly with Status Code indicating Success. Another approach could be to simply not return any IA_RT option.

6. DHCPv6 Client Behavior

A DHCPv6 client compliant with this specification MUST request the Route Option (option value TBD) in an Option Request Option (ORO) in the following messages: Solicit, Request, Renew, Rebind, Information-Request or Reconfigure. The messages are to be sent as and when specified by [RFC3315].

When processing a received Route Option a client MUST substitute a received 0::0 value in the Next Hop Option with the source IPv6 address of the received DHCPv6 message. It MUST also associate a received Link Local next hop addresses with the interface on which the client received the DHCPv6 message containing the route option. Such a substitution and/or association is useful in cases where the DHCPv6 server operator does not directly know the IPv6 next-hop address, other than knowing it is that of a DHCPv6 relay agent on the client LAN segment. DHCPv6 Packets relayed to the client are sourced by the relay using this relay's IPv6 address, which could be a link local address.

The Client MAY refresh assigned route information periodically. The generic DHCPv6 Information Refresh Time Option, as specified in [RFC4242], can be used when it is desired for the client to periodically refresh of route information.

The routes conveyed by the Route Option should be considered as complimentary to any other static route learning and maintenance mechanism used by, or on the client with one modification: The client MUST flush DHCPv6 installed routes following a link flap event on the DHCPv6 client interface over which the routes were installed. This requirement is necessary to automate the flushing of routes for

clients that may move to a different network.

7. IANA Considerations

A DHCPv6 option number of TBD for the introduced Route Option. IANA is requested to allocate three DHCPv6 option codes referencing this document: OPTION_IA_RT, OPTION_NEXT_HOP and OPTION_RT_PREFIX.

8. Security Considerations

The overall security considerations discussed in [RFC3315] apply also to this document. The Route option could be used by malicious parties to misdirect traffic sent by the client either as part of a denial of service or man-in-the-middle attack. An alternative denial of service attack could also be realized by means of using the route option to overflowing any known memory limitations of the client, or to exceed the client's ability to handle the number of next hop addresses.

Neither of the above considerations are new and specific to the proposed route option. The mechanisms identified for securing DHCPv6 as well as reasonable checks performed by client implementations are deemed sufficient in addressing these problems.

9. Contributors and .Acknowledgements

This document would not have been possible without the significant support and contribution to its development provided by: Arifumi Matsumoto, Hui Deng, Richard Johnson, Zhen Cao.

The authors would like to thank Alfred Hines, Ralph Droms, Ted Lemon, Ole Troan, Dave Oran and Dave Ward for their comments and useful suggestions.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

10.2. Informative References

[I-D.troan-multihoming-without-nat66] Troan, O., Miles, D., Matsushima, S., Okimoto, T., and D. Wing, "IPv6 Multihoming without Network Address Translation", draft-troan-multihoming-without-nat66-01 (work in progress), July 2010.

[RFC3442] Lemon, T., Cheshire, S., and B. Volz, "The Classless Static Route Option for Dynamic Host Configuration Protocol (DHCP) version 4", RFC 3442, December 2002.

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

[RFC4242] Venaas, S., Chown, T., and B. Volz, "Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 4242, November 2005.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

Authors' Addresses

Wojciech Dec (editor)
Cisco Systems
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands

Email: wdec@cisco.com

Tomasz Mrugalski
Gdansk University of Technology
Storczykowa 22B/12
Gdansk 80-177
Poland

Phone: +48 698 088 272
Email: tomasz.mrugalski@eti.pg.gda.pl

Tao Sun
China Mobile
Unit2, 28 Xuanwumenxi Ave
Beijing, Xuanwu District 100053
China

Phone:
Email: suntao@chinamobile.com

Behcet Sarikaya
Huawei USA
1700 Alma Dr. Suite 500
Plano, TX 75075
United States

Phone: +1 972-509-5599
Fax:
Email: sarikaya@ieee.org
URI:

IPv6 Maintenance
Internet-Draft
Intended status: Informational
Expires: October 03, 2013

T.J. Chown, Ed.
University of Southampton
A.M. Matsumoto, Ed.
NTT

April 01, 2013

Considerations for IPv6 Address Selection Policy Changes draft-ietf-6man-addr-select-considerations-05

Abstract

This ducument is intended to capture the address selection design team's considerations about the address selection issues mainly raised in [RFC5220]. This considerations led to the revision of RFC 3484 [RFC6724], and Address Selection DHCP option. Although it does not perfectly match the current state, this document captures the past discussion and considerations for the historical record.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on October 03, 2013.

Table of Contents

1. Introduction

This ducument is intended to capture the past discussions and considerations about the address selection issues mainly raised in [RFC5220]. This considerations led to the revision of RFC 3484 [RFC6724], and Address Selection DHCP option [I-D.ietf-6man-addr-select-opt]. Although it does not necessarily match the current state, this document captures the past discussion and considerations for the historical record.

Where the source and/or destination node of an IPv6 communication is multi-addressed, a mechanism is required for the initiating node to select the most appropriate address pair for the communication. RFC 3484 (IPv6 Default Address Selection) [RFC3484] defines such a mechanism for nodes to perform source and destination address selection. While RFC 3484 recognised the need for implementations to be able to change the policy table, it did not define how this could be achieved. Requirements have now emerged for administrators to be able to configure and potentially dynamically change RFC 3484 policy from a central control point, and for (nomadic) hosts to be able to obtain the policy for the network that they are currently attached to without manual user intervention. This text discusses considerations for such policy changes, including examples of cases where a change of policy is required, and the likely frequency of such policy changes. This text also includes some discussion on the need to also update RFC 3484, where default policies are currently defined.

There have been various operational issues observed with Default Address Selection for IPv6 (RFC 3484) [RFC3484], as described in RFC 5220 [RFC5220]. As as a result, there has been some demand for hosts to be able to have their policy tables, and potentially the rules described in RFC 3484, modified dynamically. Such changes may apply to 'static' hosts in a network where policies or topologies change, or different default policy to that described in RFC 3484 is required, or for nomadic hosts within a network for which policies may vary depending on their location within the network.

2. Issues to Consider

There are a number of aspects to consider in the context of such address selection policy updates.

First is the frequency for which such updates are likely to be required; this can be determined largely from identifying the scenarios in which policy changes will be required. This may include overriding default operating system policies on startup, as well as changes while a system is running. We discuss this topic in Section 4.

Second, by understanding how dynamic the policy update mechanism needs to be we should be better placed to determine what types of update approaches best meet those needs. There may be other considerations of course, e.g. whether the systems are in managed or unmanaged environments, and whether the solution should be proactive or automated. Section 5 covers these issues.

Third, if we assume some policy update mechanism is defined we should consider how hosts and systems may become aware that a policy change has happened, and how policy can be disseminated in a timely fashion. Thus we need to understand what kind of triggers can be identified that can be used for invoking the policy table update mechanism, e.g. address re-obtainment, address lifetime expiration, or perhaps policy lifetime expiration. We also need to consider what other factors may come into play, e.g. potential policy conflicts. This is discussed in Section 6.

After analysing these issues, we can make some initial comments regarding the potential solution spaces, and what models may be well suited, e.g. push vs pull models, and what other methods might assist us, e.g. hints from local routing tables. This is covered in Section 7.

Finally, we should assess whether these update solutions require or need RFC 3484 to be updated. In some instances, we might envision solutions that simply use RFC 3484 as guidelines and provide sufficient controls to address the current limitations in the RFC. However, as noted in RFC 5220 [RFC5220], not all the operational issues observed to date can be remedied by updating RFC 3484 alone.

3. Other Related Work

We note that there is some existing work in defining Requirements for Address Selection Mechanisms [RFC5221], and some initial work has been done in the solution space (for a DHCP-based method) [I-D.ietf-6man-addr-select-opt], but these are not discussed here. While RFC 5221 assumes that a dynamic policy update mechanism of some form is available, this draft is primarily aimed at understanding the scenarios and triggers for policy changes, to better inform future detailed solution discussions.

A draft discussing methods for multihoming without IPv6 NAT [I-D.ietf-v6ops-multihoming-without-nat66] has been published recently. This draft includes a requirement for a method to distribute address selection policy to support IPv6 multihoming.

4. Drivers for Policy Changes

If we wish to determine how frequent address selection policy changes are likely to be, we need to understand why such policies might need to be changed, for particular sites or networks.

One reference text for potential drivers for policy change is RFC 5220, in which operational issues with the existing policies described in RFC 3484 are listed. Each subsection of this document gives a reason why the existing rules or policy tables in RFC 3484 may not be sufficient in certain cases. There have been some significant changes to IPv6 since RFC 3484 was drafted which have impacted the RFC, e.g. the introduction of Unique Local Addresses (ULAs), and concerns about the impact of using longest prefix matching on (DNS) round-robin load balancing.

In summary, the issues raised in RFC 5220 were:

- Multiple Routers on a Single Interface
- Ingress Filtering
- Half-Closed Network Problem (*)
- Combined Use of Global and ULA addresses (*)
- Site Renumbering (*)
- Multicast Source Address Selection (*)
- Temporary Address Selection
- IPv4 or IPv6 Prioritization (*)
- ULA and IPv4 Dual-Stack Environment (*)
- ULA or Global Prioritization (*)

The authors of RFC 5220 noted which of these issues can be solved just by changes to the RFC 3484 policy table, marked (*) above, and which cannot. It is interesting to note that issues largely related to internal networking and (administrative) policy decisions can be handled this way. However some issues need changes beyond just policy table updates.

4.1. Internal vs External Triggers

When considering drivers or triggers that may lead to a requirement for the policy to change, we can divide the problem space into those drivers that are external to a site or network and those internal to it. In the case of the first two examples above, a dynamic policy table update may be required by externally driven routing changes, assuming the site uses a dynamic routing protocol intra-site and the routing protocol is configured to reflect changes of extra-site routing topology.

If a site is multihomed using BGP and advertising a single prefix upstream, then no policy table manipulation is required for global address preferences. However where a site is multihomed by receiving a prefix from each upstream provider, each host will have multiple addresses and many need policy table manipulation. In such a case, the policy table of hosts may need to be updated according to the routing policy.

It should be noted that we have other mechanisms for dynamic routing topology change, for example deprecating one of the advertised prefixes, e.g. when one of the upstream links has a problem. But such mechanisms may only help in some cases, and do not remove the need for agility in the RFC 3484 policy.

Other examples of external factors include a new transition mechanism being defined (e.g. as with the emergence of Teredo using 2001::/32 as assigned by IANA) and its inclusion being required in the policy table (at the time of writing Teredo is not included in RFC 3484, though some operating systems have added it), a new address block being defined, or a site renumbering event that could be triggered by an upstream provider's actions.

4.2. Administratively Triggered Changes

The other examples above are, in the general case, where the site administrator chooses to change a local policy and in doing so triggers the need for policy table updates. Some of these changes one might assume to be set once, and to change rarely, for example:

- Setting priority use of IPv6 over IPv4 (or vice versa).
- Setting priority use of ULAs over globals (or vice versa).
- Setting priority of Teredo over native IPv4 (or vice versa).
- Setting priority use of privacy addresses over DNS-published globals (or vice versa).

- An internal network renumbering occurs, perhaps due to a site expanding.

- The nature of the external connectivity through multiple ISPs requires specific additional information (policy) to be delivered to certain hosts (as discussed in 2.1.3 in RFC 5220).

- Disabling longest-prefix match functions to facilitate round-robin load balancing.

However it may be the case that different parts of a site have different policies, or policies are changed in a rolling fashion across a site over time as IPv6 and/or ULAs are introduced (for example). This may happen where the administrator prefers a gradual introduction of new policy in a phased operation across a site, rather than changing policy across the whole site in one operation.

Other administrative changes may occur more frequently, e.g.:

- Routing tables and forwarding tables change dynamically.

- A different provider (link) is preferred for a given destination.

It's possible that provider links may vary on a daily basis, or by time of day. The frequency of such policy changes will depend on the frequency that the administrator wishes to change the implied traffic engineering policies.

4.3. Start-up vs Running Changes

When a host starts up it may be configured with the default RFC 3484 policies. At this stage a number of addresses may be configured on a number of interfaces on the host. At this time it may be desirable for the host to be able to receive the site-specific policy updates as a start-up override from the RFC 3484 defaults.

Other policy changes may later be required while the host is running. Ideally the same protocol should be used for the start-up and running state update mechanism.

4.4. Nomadic Nodes

A host may be nomadic within a site and as a result it may see the preferred policy change depending on the host's topological location within that site. Such a host should be capable of receiving policy updates in a timely fashion as it migrates within the network.

While this may be one case of 'running changes' described above, the policy changes are required due to the host's new point of attachment, not changes of policy to the current point of attachment. The frequency of updates are thus depend ant on the frequency of host mobility to parts of the network that have differing policies.

It is worth noting that the point at which a nomadic host configures its network settings would be an appropriate time for it to also receive any specific address selection policy for its point of attachement.

4.5. Multiple Interface Nodes

In considering scenarios where hosts may be multi-addressed and require policy to assist in address selection, the issue of hosts with multiple interfaces arises.

A host may have a variety of reasons to have multiple interfaces. It may for example have WiFi and 3G interfaces, and be capable of sending or receiving data over either interface. In some cases these interfaces may fall within the same administrative domain (ISP) and in some cases they may not. Another example would be the case of a host with a VPN connection established, where address selection may be affected by the choice of whether the VPN connection is used or not. In this case it is interesting to note the choice to use the VPN tunnel for all, or just VPN home site traffic, is often left as a choice for the user via a tickbox selection. In addition, initiating the VPN typically changes several related settings, which is reasonable behaviour given the user chose to initiate the VPN connection.

Handling multiple interface nodes, and the possibility of conflicting policy being retrieved via each, is clearly an important problem today, but we note that RFC 3484 is currently defined as a per-node, not per-interface, mechanism (at least in the context of destination address selection). However, for RFC 3484, and its potential update mechanisms, to be applicable to typical 'real world' usage patterns, we should consider the multiple interface scenarios.

In the case where a host has multiple interfaces there are two likely scenarios:

- Wired and wireless interfaces - in this case the operating system just needs to pick one interface and use it.

- Normal and VPN interfaces - here the default should be the normal interface; the VPN interface should only be used for destinations associated with the VPN.

It has been suggested that an RFC 3484 policy table is required on a per-interface basis, though the choice of interface may itself be determined by the (destination) address selection process. As stated above, RFC 3484's policy table is currently defined to be node-wide. The node-wide problem is destination address selection when the source address is implied from a selected interface.

We note that there are some new, initial drafts published recently on the multiple interface problem [RFC6418], and on a number of possible DHCPv6 extensions, e.g. to inform hosts about routing information to assist the selection process., to inform hosts about DNS server selection policy, [RFC6731]. These drafts fall within the remit of the new IETF mif WG. We note that the mif WG may produce relevant work with respect to the analysis of RFC 3484 policy changes, but at this stage no such output exists for inclusion.

5. How Dynamic?

The discussion above suggests that many of the potential triggers for policy table changes are 'one-off' in nature, i.e. a site makes a one-time policy change. It is thus unlikely that such administrative changes will be frequent.

There are some cases where updates may be required to be more frequent. In the example of a site which is implementing the gradual introduction of new policy across its network, while the frequency of changes may be relatively high, there is still probably only one or a small number of changes per host.

There may be a higher rate of policy changes within a site if there are nomadic hosts within the site, and these are roaming frequently to parts of the network where differing policies are in effect. In such cases it may be useful for a host to know whether or not the default RFC 3484 (or soon to be 3484bis) policies are in effect or not, and for there to be a 'cheap' way for the host to discover this.

Perhaps the biggest cause of policy change lies where the preferred links or paths for certain destinations change frequently over time as (typically) traffic engineering requirements change. In some networks this may be a daily change, or change between states at different times of day. It is not clear how common these cases are, and thus further input is welcomed here. Our belief is that cases where dynamic changes are used heavily are rare.

So, unless a site or network has rapidly changing traffic engineering requirements, or includes a high number of mobile nodes where the nodes are roaming to areas of the network with differing address selection related policies, the frequency of updates is likely to be

relatively low. Most update requests will simply occur when a host starts up, and such requests for policy will be little different in frequency to other configuration requests. Other types of network change that may require a host to change its RFC 3484 policy behaviour are probably also likely to have associated changes with other host configuration data.

6. Considerations when Obtaining Policy

When a policy change is made, or a host migrates to a part of the network with different policies, that change of policy needs to be conveyed to the host. It needs to be made available and applied without restarting every affected host.

6.1. Changes in Available Address(es)

One might assume at first that when a host observes a change in its addresses, it should re-obtain the selection policy, but this may not always be the case. Not all policy changes are tied to a host changing one or more addresses, though it may be acceptable to query regardless for new policy (if a pull model is used) when address information changes.

As described above, it may be sufficient for a host to know when a policy is changed, or that perhaps the default policy is - or is not - in effect in its current locale.

6.2. Timeliness

In many, but not all, cases a policy change will need to be synchronised across a network. Thus there is a general issue of timely and synchronised dissemination of new policy. If the policy is distributed via the same mechanism that informs a host of a change of address(es), the application of the policy should be synchronised sufficiently with the address change. However, not all hosts may receive the update information at the same time, e.g. where new address assignments may be dependent on DHCP lease timers.

Where hosts use DHCPv6 for address information, in the absence of some form of Reconfigure message, a host may see a delay in policy changes being notified. One possible tool to help here is the DHCPv6 Lifetime Option (RFC4242) [RFC4242], which was originally introduced to assist with network renumbering events.

7. Solution Space

In this section we make some initial observations on the possible solution space.

7.1. Is default policy used?

There could be some mechanism to indicate to a host that the local network has a modified RFC 3484 policy in use, and thus that a revised policy table is available (and should be used). Alternatively a host could simply always attempt to obtain local RFC 3484 policy on startup. Regardless, it should also be possible for a host to detect that policy has changed (whether 'around' the host, or due to the host being nomadic). The method to convey this chnage to a host would depend on whether a push or pull configuration method is used.

It is assumed by 'default' policy here we refer to the revised/ updated RFC3484 specification, when that is produced.

7.2. Pull model

One potential solution is that a host uses a similar mechanism for RFC 3484 policy updates as is used for obtaining other configuration data, for example DHCPv6 [RFC3315]. For hosts using stateless autoconfiguration, policy could be made available via stateless DHCPv6 [RFC3736].

There are also already some initial proposals from the IETF mif WG on using DHCPv6 to deliver (mainly routing oriented) information to hosts, e.g. DHCPv6 route option and [RFC6731]. These methods assume entities that have timely knowledge of routing information can provide equally timely hints to hosts on address selection, via DHCPv6. At this stage we believe that distributing RFC 3484 policy, as configured by an administrator, is a more practical use of DHCPv6.

The DHCP model allows individual nodes to potentially have differing policy, even when on the same subnet.

7.3. Push model

For hosts only using stateless autoconfiguration, in environments without stateless DHCPv6, it may be argued that since the network is not managed, there is not likely to be any managed policy to push to the hosts. In such environments hosts may perhaps more usefully use techniques such as router hints to make informed selections, as discussed later in this text.

It may of course be possible to piggy back policy information to a host in a Router Advertisement message, though initial consensus seems to be that this is a less attractive approach.

7.4. Routing Hints

As mentioned above, if a host has routing hints available, it may be able to make more informed selections. For example, a protocol could be specified for a node to query an on-link or remote (e.g. edge) router for 'hints'. For example, a new ICMPv6 message could be defined that queried a site edge router or route server for address pairs to use for a given destination address.

However, having hosts themselves participate in routing is generally not desirable. At this stage we can simply note that address selection might be simplified when some hint based on routing state is provided to the end system, but such mechanisms are out of scope for this text.

It is noted in [RFC5887] that:

"In an environment where a site has more than one upstream link to the outside world, the site might have more than one valid routing prefix. In such cases, typically all valid routing prefixes within a site will have the same prefix length. Also in such cases, it might be desirable for hosts that obtain their addresses using DHCPv6 to learn about the availability of upstream links dynamically, by deducing from periodic IPv6 RA messages which routing prefixes are currently valid. This application seems possible within the IPv6 Neighbour Discovery architecture, but does not appear to be clearly specified anywhere."

The same thought seems relevant to address selection. There's no point selecting a source address whose prefix is not being advertised in RAs.

While routing and prefix hints may help a host make selection decisions, we should consider to what extent we wish to 'burden' a host with holding such information. If a host is to determine and cache routing hints, this may require an update of RFC 3484 policy table syntax to support preference for address pairs.

7.5. Policy Conflicts

In the case of a host operating in a single administrative domain, consistent policy should be available from whichever policy distribution mechanism provides the information. In such cases the network should not distribute policy sets from multiple entities (or by multiple mechanisms). However, in scenarios where a host is multi-addressed from multiple providers (e.g. a SOHO network with differing DSL and cable providers, or a user in a coffee shop initiating a VPN connection to their home network), multiple RFC 3484

policies may be received and there is likely to be some conflicts in the received policy information.

There are scenarios where a host may wish to ignore a conveyed policy. For example, the manager of a mobile node may not want to have its preferences changed by a visited network. In such a case one might argue that the mobile node should use MIPv6 with whatever its home network policies are.

The question then is whether the policy update mechanism itself needs to handle such potential conflicts, choosing one or ther other or merging by some set of heuristics, or whether the policy update mechansism should be viewed independently of the conflict handling. The view of the design team was that distributing policy is a network problem, while handling conflicts is a host problem.

7.6. Policy Merging

For whatever mechanism is used to distribute RFC 3484 policy, it is not yet clear whether entire policy tables will be made available, or simply differences to the 'default', and thus whether policies may need to be merged, or overridden. Some policy conflicts will be unresolvable, e.g. one prefers IPv4 over IPv6, the other vice-versa. It may be simpler, though less efficient, for whole policy tables to be distributed, to avoid the merger problem.

One option may be to split the policy table into destination address selection and source address selection tables, with the policy distribution only updating the source address selection. Whether this might make merging policies simpler or in fact more complex would require further study.

It may also be possible to indicate some priority value for a policy, e.g. the priority of the interface it is received on, or perhaps to convey a unique identifier for the policy provider. Alternatibely, if there are multiple policies in conflict, a host could simply choose to fall back to use the default RFC 3484 policy.

A host also needs to know how to decide when to accept a policy. We could simplify the discussion by assuming a host is located in and only nomadic within a single site with one administrative controlling entity.

8. On RFC3484 Default Policies

RFC 3484 includes text about mechanisms for changing policy, having 'policy hooks' and having a configurable policy table. The implication is that defaults can be changed, and the text gives examples of this in Section 10. However, issues with RFC 3484 are broader that just policy table updates - it remains the case that some operational issues with RFC 3484 are not just related to the table, but on rules themselves, e.g. longest prefix match (affecting DNS round robin as described in [RFC5220]).

While discussing default policy, we noted that the word 'default' has to be carefully defined, and also what the scope of this 'default' is. The default policy should be whatever RFC 3484, or its -bis version, states. At present some operating systems have already modified their default, based on operational feedback (e.g. on ULAs, on Teredo prefixes, or on the DNS round-robin problem). Currently we assume RFC3484 and changes to it will remain node-specific.

It certainly seems the case that the issues raised in RFC 5220, and problems about RFC 3484 revision mean that an update of RFC 3484 is required, if only because some of the issues (as highlighted earlier) cannot be addressed by updating the policy table alone. An update would also give us some hope that all operating systems might have a common 'default'.

We do not note any specific comments here on how RFC 3484 should be updated. Other drafts have made suggestions. There are some discussions on ideas however, e.g. on the semantics of labels, and in adding ULAs explicitly to the default policy table.

There have also been new issues identified, e.g. on how one differentiates between IPv4+NAT access or IPv6 transitional access (e.g. via Teredo) to a dual-stack destination (the IPv4 private address inside the NAT is implicitly global, although its explicit scope is local) [I-D.denis-v6ops-nat-addrsel]. This illustrates that new issues may continue to be identified through growing IPv6 operational experience.

It is hard to predict exactly what features people will want to add to address selection algorithms in the future. Ideally we should not preclude future flexibility. It seems clear that any RFC 3484 update has two aspects: one that uses the existing policy table capability, and one that might change associated algorithms.

9. Conclusions

We believe a key outcome of this text should be progression of a solution to allow an enterprise network manager to configure their hosts with address selection policies that may differ from the RFC 3484 default, across all or part of their network, and possibly changing polciy with time. The general scope of this text applies to site and enterprise networks, where an administrator may need to change policies over time. It also includes nomadic nodes within the site, which may migrate to different parts of the site where different policies are required.

It is clear there may be environments which might introduce conflicting policies from different administrative domains, e.g. a SOHO network with two ISP links, or an enterprise node running a VPN to a remote network. We conclude that the policy distribution mechanism is a network task, while policy conflict handling is a host task. Within this text, we do not present a solution for policy conflict handling, because at this time there is no perfect or practical solution. We thus recommend that we should progress the policy distribution solution while analysing conflict handling (which is not unique to this domain) in a separate text.

The scope of this text includes issues affecting the design of a protocol to allow a host's RFC 3484 policy table to be updated. From discussion of update triggers/scenarios, we believe rapid updates are unlikely to be required unless a node is in a network which has (very) dynamic external traffic engineering, or many nodes are mobile between parts of the network with differing policy. It's thus generally appropriate to use a similar method to obtain RFC 3484 policy as to obtain other configuration data.

In terms of obtaining policy, a pull-based solution, such as DHCPv6, may be more appropriate in managed environments (where managed non-default policies are most likely to be in effect), which would assure that hosts only gain policy information from a single entity (the DHCPv6 service). Use of DHCPv6 is also preferable if individual hosts on a subnet require different policies. In unmanaged networks, without stateless DHCPv6, use of routing hints may be an approach worth exploring.

Finally, there is a clear need to revise RFC 3484, to create a new default policy table for address selection, and to improve non policy table algorithms. This should be expedited.

10. Security Considerations

There are no extra Security consideration for this document.

11. IANA Considerations

There are no extra IANA consideration for this document.

12. Acknowledgements

The design team working on this draft is: Marcelo Bagnulo Braun, Marc Blanchet, Tim Chown, Francis Dupont, Tim Enos, TJ Evans, Brian Haberman, Tony Hain, Ruri Hiromi, Suresh Krishnan, Arifumi Matsumoto, Janos Mohacsi, Sebastien Roy, Teemu Savolainen, Fujisaki Tomohiro, and John Zhao.

We also acknowledge comments received from IETF WG mail lists, including those by Brian Carpenter and Dave Thaler.

13. Informative References

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

[RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004.

[RFC4242] Venaas, S., Chown, T., and B. Volz, "Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 4242, November 2005.

[RFC5220] Matsumoto, A., Fujisaki, T., Hiromi, R., and K. Kanayama, "Problem Statement for Default Address Selection in Multi-Prefix Environments: Operational Issues of RFC 3484 Default Rules", RFC 5220, July 2008.

[RFC5221] Matsumoto, A., Fujisaki, T., Hiromi, R., and K. Kanayama, "Requirements for Address Selection Mechanisms", RFC 5221, July 2008.

[RFC5887] Carpenter, B., Atkinson, R., and H. Flinck, "Renumbering Still Needs Work", RFC 5887, May 2010.

[RFC6418] Blanchet, M. and P. Seite, "Multiple Interfaces and Provisioning Domains Problem Statement", RFC 6418, November 2011.

[RFC6731] Savolainen, T., Kato, J., and T. Lemon, "Improved Recursive DNS Server Selection for Multi-Interfaced Nodes", RFC 6731, December 2012.

[I-D.ietf-6man-addr-select-opt] Matsumoto, A., Fujisaki, T., and T. Chown, "Distributing Address Selection Policy using DHCPv6", draft-ietf-6man-addr-select-opt-08 (work in progress), January 2013.

[I-D.ietf-mif-dhcpv6-route-option] Dec, W., Mrugalski, T., Sun, T., Sarikaya, B., and A. Matsumoto, "DHCPv6 Route Options", draft-ietf-mif-dhcpv6-route-option-05 (work in progress), August 2012.

[I-D.denis-v6ops-nat-addrsel] Denis-Courmont, R., "Problems with IPv6 source address selection and IPv4 NATs", draft-denis-v6ops-nat-addrsel-00 (work in progress), February 2009.

[I-D.ietf-v6ops-multihoming-without-nat66] Troan, O., Miles, D., Matsushima, S., Okimoto, T., and D. Wing, "IPv6 Multihoming without Network Address Translation", draft-ietf-v6ops-multihoming-without-nat66-00 (work in progress), December 2010.

Authors' Addresses

Tim Chown (editor)
University of Southampton
Southampton , Hampshire SO17 1BJ
United Kingdom

Email: tjc@ecs.soton.ac.uk

Arifumi Matsumoto (editor)
NTT NT Lab
Midori-Cho 3-9-11
Musashino-shi, Tokyo 180-8585
Japan

Phone: +81 422 59 3334
Email: matsumoto.arifumi@lab.ntt.co.jp

Internet Engineering Task Force
Internet-Draft
Obsoletes: 4294 (if approved)
Intended status: Informational
Expires: December 2, 2011

E. Jankiewicz
SRI International, Inc.
J. Loughney
Nokia
T. Narten
IBM Corporation

May 31, 2011

IPv6 Node Requirements
draft-ietf-6man-node-req-bis-11.txt

Abstract

This document defines requirements for IPv6 nodes. It is expected that IPv6 will be deployed in a wide range of devices and situations. Specifying the requirements for IPv6 nodes allows IPv6 to function well and interoperate in a large number of situations and deployments.

This document obsoletes RFC4294.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on December 2, 2011.

Table of Contents

1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Introduction

This document defines common functionality required from both IPv6 hosts and routers. Many IPv6 nodes will implement optional or additional features, but this document collects and summarizes requirements from other published Standards Track documents in one place.

This document tries to avoid discussion of protocol details, and references RFCs for this purpose. This document is intended to be an Applicability Statement and provide guidance as to which IPv6 specifications should be implemented in the general case, and which specification may be of interest to specific deployment scenarios. This document does not update any individual protocol document RFCs.

Although the document points to different specifications, it should be noted that in many cases, the granularity of a particular requirement will be smaller than a single specification, as many specifications define multiple, independent pieces, some of which may not be mandatory. In addition, most specifications define both client and server behavior in the same specification, while many implementations will be focused on only one of those roles.

This document defines a minimal level of requirement needed for a device to provide useful internet service and considers a broad range of device types and deployment scenarios. Because of the wide range of deployment scenarios, the minimal requirements specified in this document may not be sufficient for all deployment scenarios. It is perfectly reasonable (and indeed expected) for other profiles to define additional or stricter requirements appropriate for specific usage and deployment environments. For example, this document does not mandate that all clients support DHCP, but some deployment scenarios may deem it appropriate to make such a requirement. For example, government agencies in the USA have defined profiles for specialized requirements for IPv6 in target environments [DODv6] and [USGv6].

As it is not always possible for an implementer to know the exact usage of IPv6 in a node, an overriding requirement for IPv6 nodes is that they should adhere to Jon Postel’s Robustness Principle:

Be conservative in what you do, be liberal in what you accept from others [RFC0793].

2.1. Scope of This Document

IPv6 covers many specifications. It is intended that IPv6 will be deployed in many different situations and environments. Therefore, it is important to develop the requirements for IPv6 nodes to ensure interoperability.

This document assumes that all IPv6 nodes meet the minimum requirements specified here.

2.2. Description of IPv6 Nodes

From the Internet Protocol, Version 6 (IPv6) Specification [RFC2460], we have the following definitions:

Description of an IPv6 Node

- a device that implements IPv6.

Description of an IPv6 router

- a node that forwards IPv6 packets not explicitly addressed to itself.

Description of an IPv6 Host

- any node that is not a router.

3. Abbreviations Used in This Document

ATM Asynchronous Transfer Mode
AH Authentication Header
DAD Duplicate Address Detection
ESP Encapsulating Security Payload
ICMP Internet Control Message Protocol
IKE Internet Key Exchange
MIB Management Information Base
MLD Multicast Listener Discovery
MTU Maximum Transfer Unit
NA Neighbor Advertisement

NBMA Non-Broadcast Multiple Access
ND Neighbor Discovery
NS Neighbor Solicitation
NUD Neighbor Unreachability Detection
PPP Point-to-Point Protocol
PVC Permanent Virtual Circuit
SVC Switched Virtual Circuit

4. Sub-IP Layer

An IPv6 node must include support for one or more IPv6 link-layer specifications. Which link-layer specifications an implementation should include will depend upon what link-layers are supported by the hardware available on the system. It is possible for a conformant IPv6 node to support IPv6 on some of its interfaces and not on others.

As IPv6 is run over new layer 2 technologies, it is expected that new specifications will be issued. In the following, we list some of the link-layers for which an IPv6 specification has been developed. It is provided for information purposes only, and may not be complete.

- Transmission of IPv6 Packets over Ethernet Networks [RFC2464]
- IPv6 over ATM Networks [RFC2492]
- Transmission of IPv6 Packets over Frame Relay Networks Specification [RFC2590]
- Transmission of IPv6 Packets over IEEE 1394 Networks [RFC3146]
- Transmission of IPv6, IPv4, and Address Resolution Protocol (ARP) Packets over Fibre Channel [RFC4338]
- Transmission of IPv6 Packets over IEEE 802.15.4 Networks [RFC4944]
- Transmission of IPv6 via the IPv6 Convergence Sublayer over IEEE 802.16 Networks [RFC5121]
- IP version 6 over PPP [RFC5072]

In addition to traditional physical link-layers, it is also possible to tunnel IPv6 over other protocols. Examples include:

- Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs) [RFC4380]
- Section 3 of "Basic IPv6 Transition Mechanisms" [RFC4213]

5. IP Layer

5.1. Internet Protocol Version 6 - RFC 2460

The Internet Protocol Version 6 is specified in [RFC2460]. This specification MUST be supported.

Any unrecognized extension headers or options MUST be processed as described in RFC 2460.

The node MUST follow the packet transmission rules in RFC 2460.

Nodes MUST always be able to send, receive, and process fragment headers. All conformant IPv6 implementations MUST be capable of sending and receiving IPv6 packets; the forwarding functionality MAY be supported. Overlapping fragments MUST be handled as described in [RFC5722].

RFC 2460 specifies extension headers and the processing for these headers.

An IPv6 node MUST be able to process these headers. An exception is Routing Header type 0 (RH0) which was deprecated by [RFC5095] due to security concerns, and which MUST be treated as an unrecognized routing type.

5.2. Neighbor Discovery for IPv6 - RFC 4861

Neighbor Discovery is defined in [RFC4861] and was updated by [RFC5942]. Neighbor Discovery SHOULD be supported. RFC4861 states:

> Unless specified otherwise (in a document that covers operating IP over a particular link type) this document applies to all link types. However, because ND uses link-layer multicast for some of its services, it is possible that on some link types (e.g., NBMA links) alternative protocols or mechanisms to implement those services will be specified (in the appropriate document covering the operation of IP over a particular link type). The services described in this document that are not directly dependent on multicast, such as Redirects, Next-hop determination, Neighbor Unreachability Detection, etc., are expected to be provided as specified in this document. The details of how one uses ND on NBMA links is an area for further study.

Some detailed analysis of Neighbor Discovery follows:

Router Discovery is how hosts locate routers that reside on an attached link. Hosts MUST support Router Discovery functionality.

Prefix Discovery is how hosts discover the set of address prefixes

that define which destinations are on-link for an attached link. Hosts MUST support Prefix discovery.

Hosts MUST also implement Neighbor Unreachability Detection (NUD) for all paths between hosts and neighboring nodes. NUD is not required for paths between routers. However, all nodes MUST respond to unicast Neighbor Solicitation (NS) messages.

Hosts MUST support the sending of Router Solicitations and the receiving of Router Advertisements. The ability to understand individual Router Advertisement options is dependent on supporting the functionality making use of the particular option.

All nodes MUST support the Sending and Receiving of Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages. NS and NA messages are required for Duplicate Address Detection (DAD).

Hosts SHOULD support the processing of Redirect functionality. Routers MUST support the sending of Redirects, though not necessarily for every individual packet (e.g., due to rate limiting). Redirects are only useful on networks supporting hosts. In core networks dominated by routers, redirects are typically disabled. The sending of redirects SHOULD be disabled by default on backbone routers. They MAY be enabled by default on routers intended to support hosts on edge networks.

"IPv6 Host-to-Router Load Sharing" [RFC4311] includes additional recommendations on how to select from a set of available routers. RFC 4311 SHOULD be supported.

5.3. Default Router Preferences and More-Specific Routes - RFC 4191

"Default Router Preferences and More-Specific Routes" [RFC4191] provides support for nodes attached to multiple (different) networks each providing routers that advertise themselves as default routers via Router Advertisements. In some scenarios, one router may provide connectivity to destinations the other router does not and choosing the "wrong" default router can result in reachability failures. In such cases, RFC4191 can help.

Small Office/Home Office (SOHO) deployments supported by routers adhering to [RFC6204], use [RFC4191] to advertise routes to certain local destinations. Consequently, nodes that will be deployed in SOHO environments SHOULD implement [RFC4191].

5.4. SEcure Neighbor Discovery (SEND) - RFC 3971

SEND [RFC3971] and Cryptographically Generated Address (CGA) [RFC3972] provide a way to secure the message exchanges of Neighbor Discovery. SEND is a new technology, in that it has no IPv4 counterpart but it has significant potential to address certain classes of spoofing attacks. While there have been some implementations of SEND, there has been only limited deployment experience to date in using the technology. In addition, the IETF working group Cga & Send maIntenance (csi) is currently working on additional extensions intended to make SEND more attractive for deployment.

At this time, SEND is considered optional and IPv6 nodes MAY provide SEND functionality.

5.5. IPv6 Router Advertisement Flags Option - RFC 5175

Router Advertisements include an 8-bit field of single-bit Router Advertisement flags. The Router Advertisement Flags Option extends the number of available flag bits by 48 bits. At the time of this writing, 6 of the original 8 bit flags have been assigned, while 2 remain available for future assignment. No flags have been defined that make use of the new option, and thus strictly speaking, there is no requirement to implement the option today. However, implementations that are able to pass unrecognized options to a higher level entity that may be able to understand them (e.g., a user-level process using a "raw socket" facility), MAY take steps to handle the option in anticipation of a future usage.

5.6. Path MTU Discovery and Packet Size

5.6.1. Path MTU Discovery - RFC 1981

"Path MTU Discovery" [RFC1981] SHOULD be supported. From [RFC2460]:

> It is strongly recommended that IPv6 nodes implement Path MTU Discovery [RFC1981], in order to discover and take advantage of path MTUs greater than 1280 octets. However, a minimal IPv6 implementation (e.g., in a boot ROM) may simply restrict itself to sending packets no larger than 1280 octets, and omit implementation of Path MTU Discovery.

The rules in [RFC2460] and [RFC5722] MUST be followed for packet fragmentation and reassembly.

One operational issue with Path MTU discovery occurs when firewalls block ICMP Packet Too Big messages. Path MTU discovery relies on

such messages to determine what size messages can be successfully sent. Packetization Layer Path MTU Discovery [RFC4821] avoids having a dependency on Packet Too Big messages.

5.7. IPv6 Jumbograms - RFC 2675

IPv6 Jumbograms [RFC2675] are an optional extension that allow the sending of IP datagrams larger than 65.535 bytes. IPv6 Jumbograms make use of IPv6 hop-by-hop options and are only suitable on paths in which every hop and link are capable of supporting Jumbograms (e.g., within a campus or datacenter). To date, few implementations exist and there is essentially no reported experience from usage. Consequently, IPv6 Jumbograms [RFC2675] remain optional at this time.

5.8. ICMP for the Internet Protocol Version 6 (IPv6) - RFC 4443

ICMPv6 [RFC4443] MUST be supported. "Extended ICMP to Support Multi-Part Messages" [RFC4884] MAY be supported.

5.9. Addressing

5.9.1. IP Version 6 Addressing Architecture - RFC 4291

The IPv6 Addressing Architecture [RFC4291] MUST be supported.

5.9.2. IPv6 Stateless Address Autoconfiguration - RFC 4862

Hosts MUST support IPv6 Stateless Address Autoconfiguration as defined in [RFC4862]. Configuration of static address(es) may be supported as well.

Nodes that are routers MUST be able to generate link local addresses as described in RFC 4862 [RFC4862].

From 4862:

> The autoconfiguration process specified in this document applies only to hosts and not routers. Since host autoconfiguration uses information advertised by routers, routers will need to be configured by some other means. However, it is expected that routers will generate link-local addresses using the mechanism described in this document. In addition, routers are expected to successfully pass the Duplicate Address Detection procedure described in this document on all addresses prior to assigning them to an interface.

All nodes MUST implement Duplicate Address Detection. Quoting from Section 5.4 of RFC 4862:

> Duplicate Address Detection MUST be performed on all unicast addresses prior to assigning them to an interface, regardless of whether they are obtained through stateless autoconfiguration, DHCPv6, or manual configuration, with the following [exceptions noted therein].

"Optimistic Duplicate Address Detection (DAD) for IPv6" [RFC4429] specifies a mechanism to reduce delays associated with generating addresses via stateless address autoconfiguration [RFC4862]. RFC 4429 was developed in conjunction with Mobile IPv6 in order to reduce the time needed to acquire and configure addresses as devices quickly move from one network to another, and it is desirable to minimize transition delays. For general purpose devices, RFC 4429 remains optional at this time.

5.9.3. Privacy Extensions for Address Configuration in IPv6 - RFC 4941

Privacy Extensions for Stateless Address Autoconfiguration [RFC4941] addresses a specific problem involving a client device whose user is concerned about its activity or location being tracked. The problem arises both for a static client and for one that regularly changes its point of attachment to the Internet. When using Stateless Address Autoconfiguration [RFC4862], the Interface Identifier portion of formed addresses stays constant and is globally unique. Thus, although a node's global IPv6 address will change if it changes its point of attachment, the Interface Identifier portion of those addresses remain the same, making it possible for servers to track the location of an individual device as it moves around, or its pattern of activity if it remains in one place. This may raise privacy concerns as described in [RFC4862].

In such situations, RFC4941 SHOULD be implemented. In other cases, such as with dedicated servers in a data center, RFC4941 provides limited or no benefit.

Implementers of "RFC4941 should be aware that certain addresses are reserved and should not be chosen for use as temporary addresses. Consult "Reserved IPv6 Interface Identifiers" [RFC5453] for more details.

5.9.4. Default Address Selection for IPv6 - RFC 3484

The rules specified in the Default Address Selection for IPv6 [RFC3484] document MUST be implemented. IPv6 nodes will need to deal with multiple addresses configured simultaneously.

5.9.5. Stateful Address Autoconfiguration (DHCPv6) - RFC 3315

DHCPv6 [RFC3315] can be used to obtain and configure addresses. In general, a network may provide for the configuration of addresses through Router Advertisements, DHCPv6 or both. There will be a wide range of IPv6 deployment models and differences in address assignment requirements, some of which may require DHCPv6 for address assignment. Consequently all hosts SHOULD implement address configuration via DHCPv6.

In the absence of a router, IPv6 nodes using DHCP for address assignment MAY initiate DHCP to obtain IPv6 addresses and other configuration information, as described in Section 5.5.2 of [RFC4862].

5.10. Multicast Listener Discovery (MLD) for IPv6

Nodes that need to join multicast groups MUST support MLDv1 [RFC2710]. MLDv1 is needed by any node that is expected to receive and process multicast traffic. Note that Neighbor Discovery (as used on most link types -- see Section 5.2) depends on multicast and requires that nodes join Solicited Node multicast addresses.

MLDv2 [RFC3810] extends the functionality of MLDv1 by supporting Source-Specific Multicast. The original MLDv2 protocol [RFC3810] supporting Source-Specific Multicast [RFC4607] supports two types of "filter modes". Using an INCLUDE filter, a node indicates a multicast group along with a list of senders for that group it wishes to receive traffic from. Using an EXCLUDE filter, a node indicates a multicast group along with a list of senders it wishes to exclude receiving traffic from. In practice, operations to block source(s) using EXCLUDE mode are rarely used, but add considerable implementation complexity to MLDv2. Lightweight MLDv2 [RFC5790] is a simplified subset of the original MLDv2 specification that omits EXCLUDE filter mode to specify undesired source(s).

Nodes SHOULD implement either MLDv2 [RFC3810] or Lightweight MLDv2 [RFC5790]. Specifically, nodes supporting applications using Source-Specific Multicast that expect to take advantage of MLDv2's EXCLUDE functionality [RFC3810] MUST support MLDv2 as defined in [RFC3810], [RFC4604] and [RFC4607]. Nodes supporting applications that expect to only take advantage of MLDv2's INCLUDE functionality as well as Any-Source Multicast will find it sufficient to support MLDv2 as defined in [RFC5790].

If a node only supports applications that use Any-Source Multicast (i.e, they do not use source-specific multicast), implementing MLDv1 [RFC2710] is sufficient. In all cases, however, nodes are strongly

encouraged to implement MLDv2 or Lightweight MLDv2 rather than MLDv1, as the presence of a single MLDv1 participant on a link requires that all other nodes on the link operate in version 1 compatibility mode.

When MLDv1 is used, the rules in the Source Address Selection for the Multicast Listener Discovery (MLD) Protocol [RFC3590] MUST be followed.

6. DHCP vs. Router Advertisement Options for Host Configuration

In IPv6, there are two main protocol mechanisms for propagating configuration information to hosts: Router Advertisements and DHCP. Historically, RA options have been restricted to those deemed essential for basic network functioning and for which all nodes are configured with exactly the same information. Examples include the Prefix Information Options, the MTU option, etc. On the other hand, DHCP has generally been preferred for configuration of more general parameters and for parameters that may be client-specific. That said, identifying the exact line on whether a particular option should be configured via DHCP vs. an RA option has not always been easy. Generally speaking, however, there has been a desire to define only one mechanism for configuring a given option, rather than defining multiple (different) ways of configuring the same information.

One issue with having multiple ways of configuring the same information is that if a host chooses one mechanism, but the network operator chooses a different mechanism, interoperability suffers. For "closed" environments, where the network operator has significant influence over what devices connect to the network and thus what configuration mechanisms they support, the operator may be able to ensure that a particular mechanism is supported by all connected hosts. In more open environments, however, where arbitrary devices may connect (e.g., a WIFI hotspot), problems can arise. To maximize interoperability in such environments hosts would need to implement multiple configuration mechanisms to ensure interoperability.

Originally in IPv6, configuring information about DNS servers was performed exclusively via DHCP. In 2007, an RA option was defined, but was published as Experimental [RFC5006]. In 2010, "IPv6 Router Advertisement Options for DNS Configuration" [RFC6106] was published as a Standards Track Document. Consequently, DNS configuration information can now be learned either through DHCP or through RAs. Hosts will need to decide which mechanism (or whether both) should be implemented. Specific guidance regarding DNS server discovery is discussed in Section 7.

7. DNS and DHCP

7.1. DNS

DNS is described in [RFC1034], [RFC1035], [RFC3363], and [RFC3596]. Not all nodes will need to resolve names; those that will never need to resolve DNS names do not need to implement resolver functionality. However, the ability to resolve names is a basic infrastructure capability that applications rely on and most nodes will need to provide support. All nodes SHOULD implement stub-resolver [RFC1034] functionality, as in RFC 1034, Section 5.3.1, with support for:

- AAAA type Resource Records [RFC3596];
- reverse addressing in ip6.arpa using PTR records [RFC3596];
- EDNS0 [RFC2671] to allow for DNS packet sizes larger than 512 octets.

Those nodes are RECOMMENDED to support DNS security extensions [RFC4033], [RFC4034], and [RFC4035].

Those nodes are NOT RECOMMENDED to support the experimental A6 Resource Records [RFC3363].

7.2. Dynamic Host Configuration Protocol for IPv6 (DHCPv6) - RFC 3315

7.2.1. Other Configuration Information

IPv6 nodes use DHCP [RFC3315] to obtain address configuration information (See Section 5.8.5) and to obtain additional (non-address) configuration. If a host implementation supports applications or other protocols that require configuration that is only available via DHCP, hosts SHOULD implement DHCP. For specialized devices on which no such configuration need is present, DHCP may not be necessary.

An IPv6 node can use the subset of DHCP (described in [RFC3736]) to obtain other configuration information.

7.2.2. Use of Router Advertisements in Managed Environments

Nodes using the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) are expected to determine their default router information and on-link prefix information from received Router Advertisements.

7.3. IPv6 Router Advertisement Options for DNS Configuration - RFC 6106

Router Advertisements have historically limited options to those that are critical to basic IPv6 functioning. Originally, DNS

configuration was not included as an RA option and DHCP was the recommended way to obtain DNS configuration information. Over time, the thinking surrounding such an option has evolved. It is now generally recognized that few nodes can function adequately without having access to a working DNS resolver. RFC 5006 was published as an experimental document in 2007, and recently, a revised version was placed on the Standards Track [RFC6106].

Implementations SHOULD implement the DNS RA option [RFC6106].

8. IPv4 Support and Transition

IPv6 nodes MAY support IPv4.

8.1. Transition Mechanisms

8.1.1. Basic Transition Mechanisms for IPv6 Hosts and Routers - RFC 4213

If an IPv6 node implements dual stack and tunneling, then [RFC4213] MUST be supported.

9. Application Support

9.1. Textual Representation of IPv6 Addresses - RFC 5952

Software that allows users and operators to input IPv6 addresses in text form SHOULD support "A Recommendation for IPv6 Address Text Representation" [RFC5952].

9.2. Application Program Interfaces (APIs)

There are a number of IPv6-related APIs. This document does not mandate the use of any, because the choice of API does not directly relate to on-the-wire behavior of protocols. Implementers, however, would be advised to consider providing a common API, or reviewing existing APIs for the type of functionality they provide to applications.

"Basic Socket Interface Extensions for IPv6" [RFC3493] provides IPv6 functionality used by typical applications. Implementers should note that RFC3493 has been picked up and further standardized by POSIX [POSIX].

"Advanced Sockets Application Program Interface (API) for IPv6" [RFC3542] provides access to advanced IPv6 features needed by

diagnostic and other more specialized applications.

"IPv6 Socket API for Source Address Selection" [RFC5014] provides facilities that allow an application to override the default Source Address Selection rules of [RFC3484].

"Socket Interface Extensions for Multicast Source Filters" [RFC3678] provides support for expressing source filters on multicast group memberships.

"Extension to Sockets API for Mobile IPv6" [RFC4584] provides application support for accessing and enabling Mobile IPv6 features. [RFC3775]

10. Mobility

Mobile IPv6 [RFC3775] and associated specifications [RFC3776] [RFC4877] allow a node to change its point of attachment within the Internet, while maintaining (and using) a permanent address. All communication using the permanent address continues to proceed as expected even as the node moves around. The definition of Mobile IP includes requirements for the following types of nodes:

- mobile nodes
- correspondent nodes with support for route optimization
- home agents
- all IPv6 routers

At the present time, Mobile IP has seen only limited implementation and no significant deployment, partly because it originally assumed an IPv6-only environment, rather than a mixed IPv4/IPv6 Internet. Recently, additional work has been done to support mobility in mixed-mode IPv4 and IPv6 networks[RFC5555].

More usage and deployment experience is needed with mobility before any specific approach can be recommended for broad implementation in all hosts and routers. Consequently, [RFC3775], [RFC5555], and associated standards such as [RFC4877] are considered a MAY at this time.

11. Security

This section describes the specification for security for IPv6 nodes.

Achieving security in practice is a complex undertaking. Operational procedures, protocols, key distribution mechanisms, certificate

management approaches, etc. are all components that impact the level of security actually achieved in practice. More importantly, deficiencies or a poor fit in any one individual component can significantly reduce the overall effectiveness of a particular security approach.

IPsec provides channel security at the Internet layer, making it possible to provide secure communication for all (or a subset of) communication flows at the IP layer between pairs of internet nodes. IPsec provides sufficient flexibility and granularity that individual TCP connections can (selectively) be protected, etc.

Although IPsec can be used with manual keying in some cases, such usage has limited applicability and is not recommended.

A range of security technologies and approaches proliferate today (e.g., IPsec, TLS, SSH, etc.) No one approach has emerged as an ideal technology for all needs and environments. Moreover, IPsec is not viewed as the ideal security technology in all cases and is unlikely to displace the others.

Previously, IPv6 mandated implementation of IPsec and recommended the key management approach of IKE. This document updates that recommendation by making support of the IP Security Architecture [RFC 4301] a SHOULD for all IPv6 nodes. Note that the IPsec Architecture requires (e.g., Sec. 4.5 of RFC 4301) the implementation of both manual and automatic key management. Currently the default automated key management protocol to implement is IKEv2 [RFC5996].

This document recognizes that there exists a range of device types and environments where other approaches to security than IPsec can be justified. For example, special-purpose devices may support only a very limited number or type of applications and an application-specific security approach may be sufficient for limited management or configuration capabilities. Alternatively, some devices my run on extremely constrained hardware (e.g., sensors) where the full IP Security Architecture is not justified.

11.1. Requirements

"Security Architecture for the Internet Protocol" [RFC4301] SHOULD be supported by all IPv6 nodes. Note that the IPsec Architecture requires (e.g., Sec. 4.5 of RFC 4301) the implementation of both manual and automatic key management. Currently the default automated key management protocol to implement is IKEv2. As required in [RFC4301], IPv6 nodes implementing the IPsec Architecture MUST implement ESP [RFC4303] and MAY implement AH [RFC4302].

11.2. Transforms and Algorithms

The current set of mandatory-to-implement algorithms for the IP Security Architecture are defined in 'Cryptographic Algorithm Implementation Requirements For ESP and AH' [RFC4835]. IPv6 nodes implementing the IP Security Architecture MUST conform to the requirements in [RFC4835]. Preferred cryptographic algorithms often change more frequently than security protocols. Therefore implementations MUST allow for migration to new algorithms, as RFC4835 is replaced or updated in the future.

The current set of mandatory-to-implement algorithms for IKEv2 are defined in 'Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)' [RFC4307]. IPv6 nodes implementing IKEv2 MUST conform to the requirements in [RFC4307] and/or any future updates or replacements to [RFC4307].

12. Router-Specific Functionality

This section defines general host considerations for IPv6 nodes that act as routers. Currently, this section does not discuss routing-specific requirements.

12.1. IPv6 Router Alert Option - RFC 2711

The IPv6 Router Alert Option [RFC2711] is an optional IPv6 Hop-by-Hop Header that is used in conjunction with some protocols (e.g., RSVP [RFC2205] or MLD [RFC2710]). The Router Alert option will need to be implemented whenever protocols that mandate its usage (e.g., MLD) are implemented. See Section 5.9.

12.2. Neighbor Discovery for IPv6 - RFC 4861

Sending Router Advertisements and processing Router Solicitation MUST be supported.

Section 7 of RFC 3775 includes some mobility-specific extensions to Neighbor Discovery. Routers SHOULD implement Sections 7.3 and 7.5, even if they do not implement Home Agent functionality.

12.3. Stateful Address Autoconfiguration (DHCPv6) - RFC 3315

A single DHCP server ([RFC3315] or [RFC4862]) can provide configuration information to devices directly attached to a shared link, as well as to devices located elsewhere within a site. Communication between a client and a DHCP server located on different links requires the use of DHCP relay agents on routers.

In simple deployments, consisting of a single router and either a single LAN, or multiple LANs attached to the single router, together with a WAN connection, a DHCP server embedded within the router is one common deployment scenario (e.g., [RFC6204]). However, there is no need for relay agents in such scenarios.

In more complex deployment scenarios, such as within enterprise or service provider networks, the use of DHCP requires some level of configuration, in order to configure relay agents, DHCP servers, etc. In such environments, the DHCP server might even be run on a traditional server, rather than as part of a router.

Because of the wide range of deployment scenarios, support for DHCP server functionality on routers is optional. However, routers targeted for deployment within more complex scenarios (as described above) SHOULD support relay agent functionality. Note that "Basic Requirements for IPv6 Customer Edge Routers" [RFC6204] requires implementation of a DHCPv6 server function in IPv6 CE routers.

13. Network Management

Network Management MAY be supported by IPv6 nodes. However, for IPv6 nodes that are embedded devices, network management may be the only possible way of controlling these nodes.

13.1. Management Information Base Modules (MIBs)

The following two MIB modules SHOULD be supported by nodes that support an SNMP agent.

13.1.1. IP Forwarding Table MIB

IP Forwarding Table MIB [RFC4292] SHOULD be supported by nodes that support an SNMP agent.

13.1.2. Management Information Base for the Internet Protocol (IP)

IP MIB [RFC4293] SHOULD be supported by nodes that support an SNMP agent.

14. Security Considerations

This document does not directly affect the security of the Internet, beyond the security considerations associated with the individual protocols.

Security is also discussed in Section 10 above.

15. IANA Considerations

This document has no requests for IANA.

16. Authors and Acknowledgments

16.1. Authors and Acknowledgments (Current Document)

For this version of the IPv6 Node Requirements document, the authors would like to thank Hitoshi Asaeda, Brian Carpenter, Tim Chown, Ralph Droms, Sheila Frankel, Sam Hartman, Bob Hinden, Paul Hoffman, Pekka Savola, Yaron Sheffer and Dave Thaler for their comments.

16.2. Authors and Acknowledgments From RFC 4279

The original version of this document (RFC 4279) was written by the IPv6 Node Requirements design team:

Jari Arkko
jari.arkko@ericsson.com
Marc Blanchet
marc.blanchet@viagenie.qc.ca
Samita Chakrabarti
samita.chakrabarti@eng.sun.com
Alain Durand
alain.durand@sun.com
Gerard Gastaud
gerard.gastaud@alcatel.fr
Jun-ichiro itojun Hagino
itojun@iijlab.net
Atsushi Inoue
inoue@isl.rdc.toshiba.co.jp
Masahiro Ishiyama
masahiro@isl.rdc.toshiba.co.jp
John Loughney
john.loughney@nokia.com
Rajiv Raghunarayan
raraghun@cisco.com
Shoichi Sakane
shouichi.sakane@jp.yokogawa.com

Dave Thaler
dthaler@windows.microsoft.com
Juha Wiljakka
juha.wiljakka@Nokia.com

The authors would like to thank Ran Atkinson, Jim Bound, Brian Carpenter, Ralph Droms, Christian Huitema, Adam Machalek, Thomas Narten, Juha Ollila, and Pekka Savola for their comments. Thanks to Mark Andrews for comments and corrections on DNS text. Thanks to Alfred Hoenes for tracking the updates to various RFCs.

17. Appendix: Changes from One ID version to Another

RFC Editor: Please remove this section upon publication.

17.1. Appendix: Changes from -10to -11

1. Editorial cleanups.
2. Added section on DHCPv6 for servers. SHOULD implement relay agent functionality, MAY implement servers.

17.2. Appendix: Changes from -09 to -10

1. With changes in requirements for IPsec and Routing Headers, clarified language regarding processing of unknown options, and removed paragraph lising which extension headers were required to be implemented.
2. Removed "RFC4292-bis" from title.
3. Expanded the text on Jumbograms.
4. Changed recommendation of DHCPv6 from MAY to SHOULD.
5. Expanded the text on RFC4191, and changed recommendation from MAY to SHOULD.

17.3. Appendix: Changes from -08 to -09

1. Updated MLD section to include reference to Lightweight MLD [RFC5790]

17.4. Appendix: Changes from -07 to -08

1. Dropped reference to "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels" [RFC2429] in favor of a reference to tunneling via Basic IPv6 Transition Mechanisms (RFC4313).
2. Added reference to "Default Router Preferences and More-Specific Routes" [RFC4191] as a MAY.

3. Added reference to "Optimistic Duplicate Address Detection (DAD) for IPv6" (RFC4429).
4. Added reference to RFC4941 "Reserved IPv6 Interface Identifiers"
5. Added Section on APIs. References are FYI, and none are required.
6. Added text that "IPv6 Host-to-Router Load Sharing" [RFC4311] SHOULD be implemented
7. Added reference to RFC5722 (Overlapping Fragments), made it a MUST to implement.
8. Made "A Recommendation for IPv6 Address Text Representation" [RFC5952] a SHOULD.

17.5. Appendix: Changes from -06 to -07

1. Added recommendation that routers implement Section 7.3 and 7.5 of RFC 3775.
2. "IPv6 Router Advertisement Options for DNS Configuration" (RFC 6106) has been published.
3. Further clarifications to the MLD recommendation.
4. "Extended ICMP to Support Multi- Part Messages" [RFC4884] added as a MAY.
5. Added pointer to subnet clarification document (RFC 5942).
6. Added text that "IPv6 Host-to-Router Load Sharing" [RFC4311] SHOULD be implemented
7. Added reference to RFC5722 (Overlapping Fragments), made it a MUST to implement.
8. Made "A Recommendation for IPv6 Address Text Representation" [RFC5952] a SHOULD.

17.6. Appendix: Changes from -05 to -06

1. Completely revised IPsec/IKEv2 section. Text has been discussed by 6man and saag.
2. Added text to introduction clarifying that this document applies to general nodes and that other profiles may be more specific in their requirements
3. Editorial cleanups in Neighbor Discovery section in particular. Text made more crisp.
4. Moved some of the DHCP text around. Moved stateful address discussion to Section 5.8.5.
5. Added additional nuance to the redirect requirements w.r.t. default configuration setting.

17.7. Appendix: Changes from -04 to -05

1. Cleaned up IPsec section, but key questions (MUST vs. SHOULD) still open.

2. Added background section on DHCP vs. RA options.
3. Added SHOULD recommendation for DNS configuration vi RAs (RFC5006bis).
4. Cleaned up DHCP section, as it was referring to the M&O bits.
5. Cleaned up the Security Considerations Section.

17.8. Appendix: Changes from -03 to -04

1. Updated the Introduction to indicate document is an applicability statement
2. Updated the section on Mobility protocols
3. Changed Sub-IP Layer Section to just list relevant RFCs, and added some more RFCs.
4. Added Section on SEND (make it a MAY)
5. Redid Section on Privacy Extensions (RFC4941) to add more nuance to recommendation
6. Redid section on Mobility, and added additional RFCs.

18. Appendix: Changes from RFC 4294

1. There have been many editorial clarifications as well as significant additions and updates. While this section highlights some of the changes, readers should not rely on this section for a comprehensive list of all changes.
2. Updated the Introduction to indicate document is an applicability statement and that this document is aimed at general nodes.
3. Significantly updated the section on Mobility protocols, adding references and downgrading previous SHOULDs to MAY.
4. Changed Sub-IP Layer Section to just list relevant RFCs, and added some more RFCs.
5. Added Section on SEND (it is a MAY)
6. Revised Section on Privacy Extensions (RFC4941) to add more nuance to recommendation.
7. Completely revised IPsec/IKEv2 Section, downgrading overall recommendation to a SHOULD.
8. Upgraded recommendation of DHCPv6 to SHOULD.
9. Added background section on DHCP vs RA options, added SHOULD recommendation sfor DNS configuration via RAs (RFC 6106), cleaned up DHCP recommendations
10. Added recommendation that routers implement Section 7.3 and 7.5 of RFC 3775.
11. Added pointer to subnet clarification document (RFC 5942).
12. Added text that "IPv6 Host-to-Router Load Sharing" [RFC4311] SHOULD be implemented

13. Added reference to RFC5722 (Overlapping Fragments), made it a MUST to implement.
14. Made "A Recommendation for IPv6 Address Text Representation" [RFC5952] a SHOULD.
15. Removed mention of "DNAME" from the discussion about RFC-3363.
16. Numerous updates to reflect newer versions of IPv6 documents, including 4443, 4291, 3596, 4213.
17. Removed discussion of "Managed" and "Other" flags in RAs. There is no consensus at present on how to process these flags and discussion of their semantics was removed in the most recent update of Stateless Address Autoconfiguration (RFC 4862).
18. Added many more references to optional IPv6 documents.
19. Made "A Recommendation for IPv6 Address Text Representation" [RFC5952] a SHOULD.
20. Added reference to RFC5722 (Overlapping Fragments), made it a MUST to implement.
21. Updated MLD section to include reference to Lightweight MLD [RFC5790]
22. Added SHOULD recommendation for "Default Router Preferences and More-Specific Routes" [RFC4191].

19. References

19.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2671] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671, August 1999.

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.

[RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, October 1999.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

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

[RFC3590] Haberman, B., "Source Address Selection for the Multicast Listener Discovery (MLD) Protocol", RFC 3590, September 2003.

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, October 2003.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC4292] Haberman, B., "IP Forwarding Table MIB", RFC 4292, April 2006.

[RFC4293] Routhier, S., "Management Information Base for the Internet Protocol (IP)", RFC 4293, April 2006.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC4307] Schiller, J., "Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)", RFC 4307, December 2005.

[RFC4311] Hinden, R. and D. Thaler, "IPv6 Host-to-Router Load Sharing", RFC 4311, November 2005.

[RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

[RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast", RFC 4604, August 2006.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, August 2006.

[RFC4835] Manral, V., "Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 4835, April 2007.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, December 2007.

[RFC5453] Krishnan, S., "Reserved IPv6 Interface Identifiers", RFC 5453, February 2009.

[RFC5722] Krishnan, S., "Handling of Overlapping IPv6 Fragments",

RFC 5722, December 2009.

[RFC5790] Liu, H., Cao, W., and H. Asaeda, "Lightweight Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Version 2 (MLDv2) Protocols", RFC 5790, February 2010.

[RFC5942] Singh, H., Beebee, W., and E. Nordmark, "IPv6 Subnet Model: The Relationship between Links and Subnet Prefixes", RFC 5942, July 2010.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

[RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.

[RFC6204] Singh, H., Beebee, W., Donley, C., Stark, B., and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", RFC 6204, April 2011.

19.2. Informative References

[DODv6] DISR IPv6 Standards Technical Working Group, "DoD IPv6 Standard Profiles For IPv6 Capable Products Version 5.0", July 2010, <http://jitc.fhu.disa.mil/apl/ipv6/pdf/disr_ipv6_50.pdf>.

[POSIX] IEEE, "IEEE Std. 1003.1-2001 Standard for Information Technology -- Portable Operating System Interface (POSIX), ISO/IEC 9945:2002", December 2001, <http://www.opengroup.org/austin>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[RFC2429] Bormann, C., Cline, L., Deisher, G., Gardos, T., Maciocco, C., Newell, D., Ott, J., Sullivan, G., Wenger, S., and C. Zhu, "RTP Payload Format for the 1998 Version of ITU-T

Rec. H.263 Video (H.263+)", RFC 2429, October 1998.

[RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.

[RFC2492] Armitage, G., Schulter, P., and M. Jork, "IPv6 over ATM Networks", RFC 2492, January 1999.

[RFC2590] Conta, A., Malis, A., and M. Mueller, "Transmission of IPv6 Packets over Frame Relay Networks Specification", RFC 2590, May 1999.

[RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms", RFC 2675, August 1999.

[RFC3146] Fujisawa, K. and A. Onoe, "Transmission of IPv6 Packets over IEEE 1394 Networks", RFC 3146, October 2001.

[RFC3363] Bush, R., Durand, A., Fink, B., Gudmundsson, O., and T. Hain, "Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System (DNS)", RFC 3363, August 2002.

[RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 3493, February 2003.

[RFC3542] Stevens, W., Thomas, M., Nordmark, E., and T. Jinmei, "Advanced Sockets Application Program Interface (API) for IPv6", RFC 3542, May 2003.

[RFC3678] Thaler, D., Fenner, B., and B. Quinn, "Socket Interface Extensions for Multicast Source Filters", RFC 3678, January 2004.

[RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[RFC3776] Arkko, J., Devarapalli, V., and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents", RFC 3776, June 2004.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

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

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

[RFC4338] DeSanti, C., Carlson, C., and R. Nixon, "Transmission of IPv6, IPv4, and Address Resolution Protocol (ARP) Packets over Fibre Channel", RFC 4338, January 2006.

[RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.

[RFC4429] Moore, N., "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, April 2006.

[RFC4584] Chakrabarti, S. and E. Nordmark, "Extension to Sockets API for Mobile IPv6", RFC 4584, July 2006.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.

[RFC4877] Devarapalli, V. and F. Dupont, "Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture", RFC 4877, April 2007.

[RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, April 2007.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.

[RFC5006] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Option for DNS Configuration", RFC 5006, September 2007.

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

[RFC5072] S.Varada, Haskins, D., and E. Allen, "IP Version 6 over PPP", RFC 5072, September 2007.

[RFC5121] Patil, B., Xia, F., Sarikaya, B., Choi, JH., and S. Madanapalli, "Transmission of IPv6 via the IPv6

Convergence Sublayer over IEEE 802.16 Networks", RFC 5121, February 2008.

[RFC5555] Soliman, H., "Mobile IPv6 Support for Dual Stack Hosts and Routers", RFC 5555, June 2009.

[USGv6] National Institute of Standards and Technology, "A Profile for IPv6 in the U.S. Government - Version 1.0", July 2008, <http://www.antd.nist.gov/usgv6/usgv6-v1.pdf>.

Authors' Addresses

Ed Jankiewicz
SRI International, Inc.
1161 Broad Street - Suite 212
Shrewsbury, NJ 07702
USA

Phone: 443-502-5815
Email: edward.jankiewicz@sri.com

John Loughney
Nokia
955 Page Mill Road
Palo Alto 94303
USA

Phone: +1 650 283 8068
Email: john.loughney@nokia.com

Thomas Narten
IBM Corporation
3039 Cornwallis Ave.
PO Box 12195
Research Triangle Park, NC 27709-2195
USA

Phone: +1 919 254 7798
Email: narten@us.ibm.com

6man
Internet-Draft
Intended status: Standards Track
Expires: April 11, 2011

M. Kohno
Juniper Networks, Keio University
B. Nitzan
Juniper Networks
R. Bush
Y. Matsuzaki
Internet Initiative Japan
L. Colitti
Google
T. Narten
IBM Corporation

October 8, 2010

Using 127-bit IPv6 Prefixes on Inter-Router Links
draft-kohno-ipv6-prefixlen-p2p-03.txt

Abstract

On inter-router point-to-point links, it is useful for security and other reasons, to use 127-bit IPv6 prefixes. Such a practice parallels the use of 31-bit prefixes in IPv4 [RFC3021]. This document specifies motivation and usages of 127-bit IPv6 prefix lengths on inter-router point-to-point links.

Table of Contents

1. Conventions Used In This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Introduction

[RFC4291] specifies that interface IDs for all unicast address, except those that start with the binary value 000, are required to be 64 bits long and to be constructed in Modified EUI-64 format. In addition, it defines the Subnet-Router anycast address, which is intended to be used for applications where a node needs to communicate with any one of the set of routers on a link.

Some operators have been using 127-bit prefixes, but this has been discouraged due to conflicts with Subnet-Router anycast [RFC3627]. However, using 64-bit prefixes creates security issues which are particularly problematic on inter-router links, and there are other valid reasons to use prefixes longer than 64 bits, in particular /127 (see Section 5).

This document provides rationale for using 127-bit prefix lengths, reevaluates the reasons why doing so was considered harmful, and specifies how /127 prefixes can be used on inter-router links configured for use as point-to-point links.

3. Scope Of This Memo

This document is applicable to cases where operators assign specific addresses on inter-router point-to-point links and do not rely on link-local addresses. Many operators assign specific addresses for purposes of network monitoring, reverse DNS resolution for traceroute and other management tools, EBGP peering sessions, and so on.

For the purposes of this document, an inter-router point-to-point link is a link to which only two routers and no hosts are attached. This may include Ethernet links which are configured to be point-to-point. In such cases, there is no need to support Neighbor Discovery for address resolution, and other general scenarios like the use of stateless address autoconfiguration are not relevant.

Links between a router and a host, or links to which both routers and hosts are attached, are out of scope of this document.

4. Problems identified with 127-bit prefix lengths in the past

[RFC3627] discourages the use of 127-bit prefix lengths due to conflicts with the Subnet-Router anycast addresses, while stating that the utility of Subnet-Router Anycast for point-to-point links is questionable.

[RFC5375] also says the usage of 127-bit prefix lengths is not valid and should be strongly discouraged, but the stated reason for doing this is to be in compliance with [RFC3627].

Though the analyses in the RFCs are correct, operational experience with IPv6 has shown that /127 prefixes can be used successfully.

5. Reasons for using longer prefixes

There are reasons network operators use IPv6 prefix lengths greater than 64, particularly 127, for inter-router point-to-point links.

5.1. Ping-pong issue

A forwarding loop may occur on a point-to-point link with a prefix length shorter than 127. This does not affect interfaces that perform Neighbor Discovery, but some point-to-point links, which uses medium such as SONET, do not use Neighbor Discovery. As a consequence, configuring any prefix length shorter than 127 bits on these links can create an attack vector in the network.

The pingpong issue happens in case of IPv4 as well. But due to the scarcity of IPv4 address space, the current practice is to assign long prefix lengths such as /30 or /31 [RFC3021] on point-to-point links, thus the problem did not come to the fore.

The latest ICMPv6 specification [RFC4443] mitigates this problem by specifying that a router receiving a packet on a point-to-point link, which is destined to an address within a subnet assigned to that same link (other than one of the receiving router's own addresses), MUST NOT forward the packet back on that link. Instead, it SHOULD generate an ICMPv6 Destination Unreachable message code 3 in response. This check is on the forwarding processing path, so it may have performance impact.

5.2. Neighbor Cache Exhaustion issue

As described in Section 4.3.2 of [RFC3756], the use of a 64-bit prefix length on an inter-router link that uses Neighbor Discovery (e.g., Ethernet) potentially allows for denial-of-service attacks on

the routers on the link.

Consider an Ethernet link between two routers A and B to which a /64 subnet has been assigned. A packet sent to any address on the /64 (except the addresses of A and B) will cause the router attempting to forward it to create an new cache entry in state INCOMPLETE, send a Neighbor Solicitation message to be sent on the link, start a retransmit timer, and so on [RFC4861].

By sending a continuous stream of packets to a large number of the 2^64 - 3 unassigned addresses on the link (one for each router and one for Subnet-Router Anycast), an attacker can create a large number of neighbor cache entries and send a large number of Neighbor Solicitation packets which will never receive replies, thereby consuming large amounts of memory and processing resources. Sending the packets to one of the 2^24 addresses on the link which has the same Solicited-Node multicast address as one of the routers also causes the victim to spend large amounts of processing time discarding useless Neighbor Solicitation messages.

Careful implementation and rate-limiting can limit the impact of such an attack, but are unlikely to neutralize it completely. Rate-limiting neighbor solicitation messages will reduce CPU usage, and following the garbage-collection recommendations in [RFC4861] will maintain reachability, but if the link is down and neighbor cache entries have expired while the attack is ongoing, legitimate traffic (for example, BGP sessions) over the link might never be re-established because the routers cannot resolve each others' IPv6 addresses to MAC addresses.

This attack is not specific to point-to-point links, but is particularly harmful in the case of point-to-point backbone links, which may carry large amounts of traffic to many destinations over long distances.

While there are a number of ways to mitigate this kind of issue, assigning /127 subnets eliminates it completely.

5.3. Other reasons

Though address space conservation considerations are less important for IPv6 than they are in IPv4, some operators prefer not to assign /64s to individual point-to-point links. Instead, they may be able to number all of their point-to-point links out of a single (or small number of) /64s.

6. Recommendations

Routers MUST support the assignment of /127 prefixes on point-to-point inter-router links.

When assigning and using any /127 prefixes, the following considerations apply.Some addresses have special meanings, in particular addresses corresponding to reserved anycast addresses. When assigning prefixes (and addresses) to links, care should be taken to ensure that addresses reserved for such purposes aren't inadvertently assigned and used as unicast addresses. Otherwise, nodes may receive packets that they are not intended to receive. Specifically, assuming that a number of point-to-point links will be numbered out of a single /64 prefix:

a) Addresses with all zeros in the rightmost 64 bits SHOULD NOT be assigned as unicast addresses, to avoid colliding with the Subnet-Router anycast address. [RFC4291]

b) Addresses in which the rightmost 64 bits are assigned the highest 128 values SHOULD NOT be used as unicast addresses, to avoid colliding with Reserved Subnet Anycast Addresses. [RFC2526]

7. Security Considerations

Section 5.1 and 5.2 discuss about seurity related issues.

8. IANA Considerations

None.

9. Contributors

Chris Morrow, morrowc@google.com

Pekka Savola, pekkas@netcore.fi

Remi Despres, remi.despres@free.fr

Seiichi Kawamura, karamucho@mesh.ad.jp

10. Acknowledgments

We'd like to thank Ron Bonica, Pramod Srinivasan, Olivier Vautrin,

Tomoya Yoshida, Warren Kumari and Tatsuya Jinmei for their helpful inputs.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

11.2. Informative References

[RFC2526] Johnson, J. and S. Deering, "Reserved IPv6 Subnet Anycast Addresses", RFC 2526, March 1999.

[RFC3021] Retana, A., White, R., and V. Fuller, "Using 31-Bit Prefixes on IPv4 Point-to-Point Links", December 2000.

[RFC3627] Savola, P., "Use of /127 Prefix Length Between Routers Considered Harmful", RFC 3627, September 2003.

[RFC3756] Nikander, P., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, May 2004.

[RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

[RFC5375] Van de Velde, G., Popoviciu, C., Chown, T., Bonness, O., and C. Hahn, "IPv6 Unicast Address Assignment Considerations", RFC 5375, December 2008.

Authors' Addresses

Miya Kohno
Juniper Networks, Keio University
Shinjuku Park Tower, 3-7-1 Nishishinjuku
Shinjuku-ku, Tokyo 163-1035
Japan

Email: mkohno@juniper.net

Becca Nitzan
Juniper Networks
1194 North Marhilda Avenue
Sunnyvale, CA 94089
USA

Email: nitzan@juniper.net

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, WA 98110
USA

Email: randy@psg.com

Yoshinobu Matsuzaki
Internet Initiative Japan
Jinbocho Mitsui Building,
1-105 Kanda Jinbo-cho, Tokyo 101-0051
Japan

Email: maz@iij.ad.jp

Lorenzo Colitti
Google
1600 Amphitheatre Parkway,
Mountain View, CA 94043
USA

Email: lorenzo@google.com

Thomas Narten
IBM Corporation
3039 Cornwallis Ave.
PO Box 12195 - BRQA/502 Research Triangle Park, NC 27709-2195
USA

Email: narten@us.ibm.com