6man
Internet-Draft
Updates: 2460 (if approved)
Intended status: Best Current Practice
Expires: February 22, 2016

F. Gont
UTN-FRH / SI6 Networks
W. Liu
Huawei Technologies
R. Bonica
Juniper Networks

August 21, 2015

# Transmission and Processing of IPv6 Options
draft-gont-6man-ipv6-opt-transmit-02.txt

## Abstract

Various IPv6 options have been standardized since the core IPv6 standard was first published. This document updates RFC 2460 to clarify how nodes should deal with such IPv6 options and with any options that are defined in the future. It complements [RFC7045], which offers a similar clarification regarding IPv6 Extension Headers.

## 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 February 22, 2016.

Table of Contents



## 1. Introduction and Problem Statement

Various IPv6 options have been standardized since the core IPv6 standard [RFC2460] was first published. Except for the padding options (Pad1 and PadN), all the options that have so far been specified are meant to be employed with specific IPv6 Extension Header (EH) types. Additionally, some options have specific requirements such as, for example, only allowing a single instance of the option in the corresponding IPv6 extension header. This establishes some criteria for validating packets that employ IPv6 options.

[RFC2460] specifies that IPv6 extension headers (with the exception of the Hop-by-Hop Options extension header) are not examined or processed by any node along a packet's delivery path, until the packet reaches the node (or each of the set of nodes, in the case of multicast) identified in the Destination Address field of the IPv6 header. However, in practice this is not really the case: some routers, and a variety of middleboxes such as firewalls, load balancers, or packet classifiers, might inspect other parts of each packet [RFC7045]. Hence both end-nodes an intermediate nodes may end up inspecting the contents of extension headers and discard packets based on the presence of specific IPv6 options.

This document clarifies the default processing of IPv6 options. In those cases in which the specifications add additional constraints/ requirements regarding IPv6 options, such additional constraints/ requirements are also taken into account.

## 2. Terminology and Conventions Used in This Document

### 2.1. Terminology

In the remainder of this document, the term "forwarding node" refers to any router, firewall, load balancer, prefix translator, or any other device or middlebox that forwards IPv6 packets with or without examining the packet in any way.

In this document, "standard" IPv6 options are those specified in detail by IETF Standards Actions [RFC5226]. "Experimental" options include those defined by any Experimental RFC and the option types 0x1E, 0x3E, 0x5E, 0x7E, 0x9E, 0xBE, 0xDE, and 0xFE, defined by [RFC3692] and [RFC4727] when used as experimental options. "Defined" options are the "standard" options plus the "experimental" ones.

The terms "permit" (allow the traffic), "drop" (drop with no notification to sender), and "reject" (drop with appropriate notification to sender) are employed as defined in [RFC3871]. Throughout this document we also employ the term "discard" as a generic term to indicate the act of discarding a packet, irrespective of whether the sender is notified of such packet drops.

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].

### 2.2. Conventions

This document clarifies some basic validation of IPv6 options, and specifies the default processing of them. We recommend that a configuration option is made available to govern the processing of each IPv6 option type, on a per-EH-type granularity. Such configuration options may include the following possible settings:

- Permit this IPv6 Option type
- Drop (and log) packets containing this IPv6 option type
- Reject (and log) packets containing this IPv6 option type (where the packet drop is signaled with an ICMPv6 error message)

- o Rate-limit the processing of packets containing this IPv6 option type

- o Ignore this IPv6 option type (forwarding packets that contain them)

We note that special care needs to be taken when devices log packet drops/rejects. Devices should count the number of packets dropped/ rejected, but the logging of drop/reject events should be limited so as to not overburden device resources.

Finally, we note that when discarding packets, it is generally desirable that the sender be signaled of the packet drop, since this is of use for trouble-shooting purposes. However, throughout this document (when recommending that packets be discarded) we generically refer to the action as "discard" without specifying whether the sender is signaled of the packet drop.

## 3. Considerations for All IPv6 Options

Forwarding nodes that discard packets (by default) based on the presence of IPv6 options are known to cause connectivity failures and deployment problems. Any forwarding node along an IPv6 packet's path, which forwards the packet for any reason, SHOULD do so regardless of any IPv6 Destination Options that are present, as required by [RFC2460]. Exceptionally, if a forwarding node is designed to examine IPv6 Destination Options for any reason, such as firewalling, it MUST recognise and deal appropriately with all standard IPv6 options types and SHOULD recognise and deal appropriately with all experimental IPv6 options. The list of standard and experimental option types is maintained by IANA (see [IANA-IPV6-PARAM]), and implementors are advised to check this list regularly for updates.

In the case of some options meant to be included in IPv6 extension headers other than Hop-by-Hop Options, [RFC2460] requires destination hosts to discard the corresponding packet if the option is unrecognised. However, intermediate forwarding nodes SHOULD NOT do this, since doing so might cause them to inadvertently discard traffic using a recently standardised IPv6 option not yet recognised by the intermediate node. The exceptions to this rule are discussed next.

If a forwarding node discards a packet containing a standard IPv6 option, it MUST be the result of a configurable policy and not just the result of a failure to recognise such an option. This means that the discard policy for each standard type of IPv6 option MUST be

individually configurable. The default configuration SHOULD allow all standard IPv6 options.

Experimental IPv6 options SHOULD be treated in the same way as standard IPv6 options, including an individually configurable discard policy.

A node that processes the contents of an extension header MUST discard the corresponding packet if it contains any defined options that are not meant for the extension header being processed. This document requests IANA to add a new column to [IANA-IPV6-PARAM] to clearly mark the IPv6 Extension Header type(s) for which each option (defined by IETF Standards Action or IESG Approval) is valid.

A node that processes the contents of an IPv6 extension header MAY discard the corresponding packet if it contains any options that have become deprecated. Whether or not such packets are dropped SHOULD be configurable, and the default setting MUST be to not drop such packets.

A node that processes the contents of an extension header and encounters an undefined (unrecognised) IPv6 option MUST react to such option according to the highest-order two bits of the option type, as specified by Section 4.2 of [RFC2460].

A node that processes an IPv6 extension header MAY discard a packet containing any experimental IPv6 options.

## 4. Processing of currently-defined IPv6 Options

The following subsections provide advice on how to process the IPv6 options that have been defined at the time of this writing, according to the rules specified in the previous sections.

### 4.1. Hop-by-Hop Options Header

A node that processes the Hop-by-Hop Options extension header MUST discard the corresponding packet if it contains any options that are not valid for the Hop-by-Hop Options extension header [IANA-IPV6-PARAM].

A node that processes the Hop-by-Hop Options extension header MUST discard a packet containing multiple instances (i.e., more than one) of this option in the Hop-by-Hop Options extension header:

- Type 0x05: Router Alert [RFC2711]

NOTE: The rationale for discarding the packet is that [RFC2711] forbids multiple instances of this option.

A node that processes the Hop-by-Hop Options extension header MUST discard a packet that carries a Fragment Header and also contains this option in the Hop-by-Hop Options extension header:

- Type 0xC2: Jumbo Payload [RFC2675]

  NOTE: The rationale for discarding the packet is that [RFC2675] forbids the use of the Jumbo Payload Option in packets that carry a Fragment Header.

A node that processes the Hop-by-Hop Options extension header MAY discard a packet containing any of the following options in that header:

- Type=0x4D: Deprecated

  NOTE: The rationale for discarding the packet is that the aforementioned option has been deprecated.

A node that processes the Hop-by-Hop Options extension header MAY discard a packet containing any of the following options in that header:

- Type 0x1E: RFC3692-style Experiment [RFC4727]
- Type 0x3E: RFC3692-style Experiment [RFC4727]
- Type 0x5E: RFC3692-style Experiment [RFC4727]
- Type 0x7E: RFC3692-style Experiment [RFC4727]
- Type 0x9E: RFC3692-style Experiment [RFC4727]
- Type 0xBE: RFC3692-style Experiment [RFC4727]
- Type 0xDE: RFC3692-style Experiment [RFC4727]
- Type 0xFE: RFC3692-style Experiment [RFC4727]

  NOTE: This is in line with the corresponding specification in [RFC7045] for experimental extension headers.

### 4.2. Destination Options Header

A node that processes the Destination Options header MUST discard a packet containing any options that are not valid for the Destination Options header [IANA-IPV6-PARAM].

A node that processes the Destination Options extension header MAY discard a packet containing any of the following options in that header:

- Type 0x8A: Endpoint Identification [nimrod-eid] [NIMROD-DOC]
- Type 0x4D: Deprecated

  NOTE: The rationale for discarding the packet is that the aforementioned options have been deprecated.

A node that processes the Destination Options extension header MAY discard a packet containing any of the following options in that header:

- Type 0x1E: RFC3692-style Experiment [RFC4727]
- Type 0x3E: RFC3692-style Experiment [RFC4727]
- Type 0x5E: RFC3692-style Experiment [RFC4727]
- Type 0x7E: RFC3692-style Experiment [RFC4727]
- Type 0x9E: RFC3692-style Experiment [RFC4727]
- Type 0xBE: RFC3692-style Experiment [RFC4727]
- Type 0xDE: RFC3692-style Experiment [RFC4727]
- Type 0xFE: RFC3692-style Experiment [RFC4727]

  NOTE: This is in line with the corresponding specification in [RFC7045] for experimental extension headers.

## 5. IANA Considerations

IANA is requested to add an extra column entitled "Extension Header Types" to the "Destination Options and Hop-by-Hop Options" registry [IANA-IPV6-PARAM], to clearly mark the IPv6 Extension Header types for which each option (defined by IETF Standards Action or IESG Approval) is valid (see the list below). This also applies to Destination Options and Hop-by-Hop Options defined in the future.

What follows is the initial list of IPv6 options and the corresponding marks that indicate which Extension Header type(s) these IPv6 options are valid for:

| Hex Value | Description | Reference | EH Types |
|---|---|---|---|
| 0x00 | Pad1 | [RFC2460] | DH |
| 0x01 | PadN | [RFC2460] | DH |
| 0xC2 | Jumbo Payload | [RFC2675] | H |
| 0x63 | RPL Option | [RFC6553] | H |
| 0x04 | Tunnel Encapsulation Limit | [RFC2473] | D |
| 0x05 | Router Alert | [RFC2711] | H |
| 0x26 | Quick-Start | [RFC4782] | H |
| 0x07 | CALIPSO | [RFC5570] | H |
| 0x08 | SMF_DPD | [RFC6621] | H |
| 0xC9 | Home Address | [RFC6275] | D |
| 0x8A | Endpoint Identification | [nimrod-eid][NIMROD-DOC] | D |
| 0x8B | ILNP Nonce | [RFC6744] | D |
| 0x8C | Line-Identification Option | [RFC6788] | D |
| 0x4D | Deprecated | | U |
| 0x6D | MPL Option | [I-D.ietf-roll-trickle-mcast] | H |
| 0xEE | IPv6 DFF Header | [RFC6971] | H |
| 0x1E | RFC3692-style Experiment | [RFC4727] | DH |

| | | | |
|---|---|---|---|
| 0x3E | RFC3692-style Experiment | [RFC4727] | DH |
| 0x5E | RFC3692-style Experiment | [RFC4727] | DH |
| 0x7E | RFC3692-style Experiment | [RFC4727] | DH |
| 0x9E | RFC3692-style Experiment | [RFC4727] | DH |
| 0xBE | RFC3692-style Experiment | [RFC4727] | DH |
| 0xDE | RFC3692-style Experiment | [RFC4727] | DH |
| 0xFE | RFC3692-style Experiment | [RFC4727] | DH |

Additionally, the following legend should be added to the registry:

D: Destination Options Header
H: Hop-by-Hop Options Header
U: Unknown

## 6. Security Considerations

Forwarding nodes that operate as firewalls MUST conform to the requirements in this document. In particular, packets containing standard IPv6 options are only to be discarded as a result of an intentionally configured policy.

These requirements do not affect a firewall's ability to filter out traffic containing unwanted or suspect IPv6 options, if configured to do so. However, the changes do require firewalls to be capable of permitting any or all IPv6 options, if configured to do so. The default configurations are intended to allow normal use of any standard IPv6 option, avoiding the interoperability issues described in Section 1 and Section 3.

As noted above, the default configuration might discard packets containing experimental IPv6 options.

## 7. Acknowledgements

This document is heavily based on [RFC7045], authored by Brian Carpenter and Sheng Jiang.

The authors of this document would like to thank (in alphabetical order) Brian Carpenter, Mike Heard, and Jen Linkova, for providing valuable comments on earlier versions of this document.

## 8. References

### 8.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <http://www.rfc-editor.org/info/rfc1034>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <http://www.rfc-editor.org/info/rfc2205>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <http://www.rfc-editor.org/info/rfc2473>.

[RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms", RFC 2675, DOI 10.17487/RFC2675, August 1999, <http://www.rfc-editor.org/info/rfc2675>.

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, DOI 10.17487/RFC2710, October 1999, <http://www.rfc-editor.org/info/rfc2710>.

[RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, DOI 10.17487/RFC2711, October 1999, <http://www.rfc-editor.org/info/rfc2711>.

[RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, DOI 10.17487/RFC3692, January 2004, <http://www.rfc-editor.org/info/rfc3692>.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <http://www.rfc-editor.org/info/rfc4302>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <http://www.rfc-editor.org/info/rfc4303>.

[RFC4304] Kent, S., "Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP)", RFC 4304, DOI 10.17487/RFC4304, December 2005, <http://www.rfc-editor.org/info/rfc4304>.

[RFC4727] Fenner, B., "Experimental Values In IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers", RFC 4727, DOI 10.17487/RFC4727, November 2006, <http://www.rfc-editor.org/info/rfc4727>.

[RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-Start for TCP and IP", RFC 4782, DOI 10.17487/RFC4782, January 2007, <http://www.rfc-editor.org/info/rfc4782>.

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <http://www.rfc-editor.org/info/rfc5095>.

[RFC5201] Moskowitz, R., Nikander, P., Jokela, P., Ed., and T. Henderson, "Host Identity Protocol", RFC 5201, DOI 10.17487/RFC5201, April 2008, <http://www.rfc-editor.org/info/rfc5201>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, DOI 10.17487/RFC5533, June 2009, <http://www.rfc-editor.org/info/rfc5533>.

[RFC5570] StJohns, M., Atkinson, R., and G. Thomas, "Common Architecture Label IPv6 Security Option (CALIPSO)", RFC 5570, DOI 10.17487/RFC5570, July 2009, <http://www.rfc-editor.org/info/rfc5570>.

[RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <http://www.rfc-editor.org/info/rfc6275>.

[RFC6398] Le Faucheur, F., Ed., "IP Router Alert Considerations and Usage", BCP 168, RFC 6398, DOI 10.17487/RFC6398, October 2011, <http://www.rfc-editor.org/info/rfc6398>.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <http://www.rfc-editor.org/info/rfc6550>.

[RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, DOI 10.17487/RFC6553, March 2012, <http://www.rfc-editor.org/info/rfc6553>.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, DOI 10.17487/RFC6554, March 2012, <http://www.rfc-editor.org/info/rfc6554>.

[RFC6621] Macker, J., Ed., "Simplified Multicast Forwarding", RFC 6621, DOI 10.17487/RFC6621, May 2012, <http://www.rfc-editor.org/info/rfc6621>.

[RFC6740] Atkinson, RJ. and SN. Bhatti, "Identifier-Locator Network Protocol (ILNP) Architectural Description", RFC 6740, DOI 10.17487/RFC6740, November 2012, <http://www.rfc-editor.org/info/rfc6740>.

[RFC6744] Atkinson, RJ. and SN. Bhatti, "IPv6 Nonce Destination Option for the Identifier-Locator Network Protocol for IPv6 (ILNPv6)", RFC 6744, DOI 10.17487/RFC6744, November 2012, <http://www.rfc-editor.org/info/rfc6744>.

[RFC6788] Krishnan, S., Kavanagh, A., Varga, B., Ooghe, S., and E. Nordmark, "The Line-Identification Option", RFC 6788, DOI 10.17487/RFC6788, November 2012, <http://www.rfc-editor.org/info/rfc6788>.

[RFC6971] Herberg, U., Ed., Cardenas, A., Iwao, T., Dow, M., and S. Cespedes, "Depth-First Forwarding (DFF) in Unreliable Networks", RFC 6971, DOI 10.17487/RFC6971, June 2013, <http://www.rfc-editor.org/info/rfc6971>.

[RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <http://www.rfc-editor.org/info/rfc7045>.

[RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of Oversized IPv6 Header Chains", RFC 7112, DOI 10.17487/RFC7112, January 2014, <http://www.rfc-editor.org/info/rfc7112>.

## 8.2. Informative References

[Biondi2007]
Biondi, P. and A. Ebalard, "IPv6 Routing Header Security", CanSecWest 2007 Security Conference, 2007, <http://www.secdev.org/conf/IPv6_RH_security-csw07.pdf>.

[I-D.ietf-roll-trickle-mcast]
Hui, J. and R. Kelsey, "Multicast Protocol for Low power and Lossy Networks (MPL)", draft-ietf-roll-trickle-mcast-12 (work in progress), June 2015.

[I-D.ietf-v6ops-ipv6-ehs-in-real-world]
Gont, F., Linkova, J., Chown, T., and S. LIU, "Observations on IPv6 EH Filtering in the Real World", draft-ietf-v6ops-ipv6-ehs-in-real-world-00 (work in progress), April 2015.

[IANA-IPV6-PARAM]
Internet Assigned Numbers Authority, "Internet Protocol Version 6 (IPv6) Parameters", December 2013, <http://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml>.

[NIMROD-DOC]
Nimrod Documentation Page, , "http://ana-3.lcs.mit.edu/~jnc/nimrod/".

[nimrod-eid]
Lynn, C., "Endpoint Identifier Destination Option", IETF Internet Draft, draft-ietf-nimrod-eid-00.txt, November 1995.

[RFC3871] Jones, G., Ed., "Operational Security Requirements for Large Internet Service Provider (ISP) IP Network Infrastructure", RFC 3871, DOI 10.17487/RFC3871, September 2004, <http://www.rfc-editor.org/info/rfc3871>.

[RFC7126] Gont, F., Atkinson, R., and C. Pignataro, "Recommendations on Filtering of IPv4 Packets Containing IPv4 Options", BCP 186, RFC 7126, DOI 10.17487/RFC7126, February 2014, <http://www.rfc-editor.org/info/rfc7126>.

## Authors' Addresses

Fernando Gont
UTN-FRH / SI6 Networks
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fgont@si6networks.com
URI: http://www.si6networks.com

Will(Shucheng) Liu
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Email: liushucheng@huawei.com

Ronald P. Bonica
Juniper Networks
2251 Corporate Park Drive
Herndon, VA 20171
US

Phone: 571 250 5819
Email: rbonica@juniper.net

IPv6 maintenance Working Group (6man)
Internet-Draft
Updates: 6106 (if approved)
Intended status: Standards Track
Expires: August 30, 2015

F. Gont
SI6 Networks / UTN-FRH
P. Simerda

W. Liu
Huawei Technologies

February 26, 2015

# Current issues with DNS Configuration Options for SLAAC
# draft-gont-6man-slaac-dns-config-issues-01

## Abstract

RFC 6106 specifies two Neighbor Discovery options that can be included in Router Advertisement messages to convey information about DNS recursive servers and DNS Search Lists. Small lifetime values for the aforementioned options have been found to cause interoperability problems in those network scenarios in which these options are used to convey DNS-related information. This document analyzes the aforementioned problem, and formally updates RFC 6106 such that these issues are mitigated.

## 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 August 30, 2015.

Table of Contents



## 1. Introduction

RFC 6106 [RFC6106] specifies two Neighbor Discovery (ND) [RFC4861] options that can be included in Router Advertisement messages to convey information about DNS recursive servers and DNS Search Lists. Namely, the Recursive DNS Server (RDNSS) Option specifies the IPv6 addresses of recursive DNS servers, while the DNS Search List (DNSSL) Option specifies a "search list" to be used when trying to resolve a name by means of the DNS.

Each of this options include a "Lifetime" field which specifies the maximum time, in seconds, during which the information included in the option can be used by the receiving system. The aforementioned "Lifetime" value is set as a function of the Neighbor Discovery parameter 'MaxRtrAdvInterval', which specifies the maximum time allowed between sending unsolicited multicast Router Advertisements from an interface. The recommended bounds (MaxRtrAdvInterval <= Lifetime <= 2*MaxRtrAdvInterval) have been found to be too short for scenarios in which some Router Advertisement messages may be lost. In such scenarios, hosts may fail to receive unsolicited Router Advertisements and therefore fail to refresh the expiration time of the DNS-related information previously learned through the RDNSS and DNSSL options), thus eventually discarding the aforementioned DNS-related information prematurely.

Some implementations consider the lack of DNS-related information as a hard failure, thus causing configuration restart. This situation

is exacerbated in those implementations in which IPv6 connectivity and IPv4 connectivity are bound together, and hence failure in the configuration of one of them causes the whole link to be restarted.

This document formally updates RFC 6106 such that this issue is mitigated.

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. Changing the Semantics of the 'Lifetime' field of RDNSS and DNSSL options

The semantics of the 'Lifetime' field of the RDNSS and DNSSL options is updated as follows:

- The 'Lifetime' field indicates the amount of time during which the aforementioned DNS-related information is expected to be stable. A node is NOT required to discard the DNS-related information once the Lifetime expires.

- If the information received in a RDNSS or DNSSL option is already present in the corresponding local data structures, the corresponding 'Expiration' time should be updated according to the value in the 'Lifetime' field of the received option. A 'Lifetime' of '0' causes the corresponding information to be discarded, as already specified in [RFC6106].

- If a host has already gathered a sufficient number of RDNSS addresses (or DNS search domain names), and additional data is received while the existing entries have not yet expired, the received RDNSS addresses (or DNS search domain names) SHOULD be ignored.

- If a host receives new RDNSS addresses (or DNS search domain names), and some of the existing entries have expired, the newly-learned information SHOULD be used to replace the expired entries.

- A host SHOULD flush configured DNS-related information when it has any reason to believe that its network connectivity has changed in some relevant way (e.g., there has been a "link change event"). When that happens, the host MAY send a Router Solicitation message to re-learn the corresponding DNS-related information.

- The most-recently-updated information SHOULD have higher priority over the other DNS-related information already present on the local host.

We note that the original motivation for enforcing a short expiration timeout value was to allow mobile nodes to prefer local RDNSSes to remote RDNSSes. However, the above rules already allow for a timely update of the corresponding DNS-related information.

## 3. Changing the Default Values of the 'Lifetime' field of RDNSS and DNSSL options

The default RDNSS/DNSSL "Lifetime" value in current the current router solutions vary between MaxRtrAdvInterval and 2*MaxRtrAdvInterval. This means that common packet loss rates can lead to the problem described in this document.

One possible approach to mitigate this issue would be to avoid 'Lifetime' values that are on the same order as MaxRtrAdvInterval. This solution would require, of course, changes in router software.

When specifying a better default value, the following aspects should be considered:

- IPv6 will be used on many links (including IEEE 802.11) that experience packet loss. Therefore losing a few packets in a short period of time should not invalidate DNS configuration information.

- Unsolicited Router Advertisements sent on Ethernet networks result in packets that employ multicast Ethernet Destination Addresses. A number of network elements (including those that perform bridging between wireless networks and wired networks) have problems with multicasted Ethernet frames, thus typically leading to packet loss of some of those frames. Therefore, SLAAC implementations should be able to cope with devices that can lose several multicast packets in a row.

[RFC6106] is hereby updated as follows:

> The default value of AdvRDNSSLifetime and AdvDNSSLLifetime MUST be at least 10*MaxRtrAdvInterval so that the probability of hosts receiving unsolicited Router Advertisements is increased.

## 4. Use of Router Solicitations for active Probing

According to RFC 6106, hosts MAY send Router Solicitations to avoid expiry of RDNSS and DNSSL lifetimes. This technique could be employed as a "last resort" when expiration of the RDNSS and DNSSL information is imminent.

## 5. Sanitize the received RDNSS/DNSSL 'Lifetime' Values

A host that receives a RDNSS or DNSSL option that has a non-zero Lifetime smaller than 10*MaxRtrAdvInterval should employ 10*MaxRtrAdvInterval as the Lifetime value of the corresponding RDNSS or DNSSL option.

## 6. Security Considerations

This document does not introduce any additional security considerations to those documented in the "Security Considerations" section of [RFC6106].

## 7. Acknowledgements

The authors would like to thank Erik Nordmark and Mark Smith for their valuable input on the topic covered by this document.

## 8. 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.

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

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

## Authors' Addresses

Fernando Gont
SI6 Networks / UTN-FRH
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fgont@si6networks.com
URI: http://www.si6networks.com

Pavel Simerda

Phone: +420 775 996 256
Email: pavlix@pavlix.net

Will Liu
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Email: liushucheng@huawei.com

IPv6 maintenance Working Group (6man)
Internet-Draft
Updates: 4861 (if approved)
Intended status: Standards Track
Expires: September 9, 2015

F. Gont
SI6 Networks / UTN-FRH
R. Bonica
Juniper Networks
W. Liu
Huawei Technologies

March 8, 2015

# Validation of IPv6 Neighbor Discovery Options
## draft-ietf-6man-nd-opt-validation-00

Abstract

This memo specifies validation rules for IPv6 Neighbor Discovery (ND) Options. In order to avoid pathological outcomes, IPv6 implementations validate incoming ND options using these rules.

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 September 9, 2015.

Table of Contents



## 1. Introduction

IPv6 [RFC2460] nodes use Neighbor Discovery (ND) [RFC4861] to discover their neighbors and to learn their neighbors' link-layer addresses. IPv6 hosts also use ND to find neighboring routers that can forward packets on their behalf. Finally, IPv6 nodes use ND to verify neighbor reachability, and to detect link-layer address changes.

ND defines the following ICMPv6 [RFC4443] messages:

- Router Solicitation (RS)
- Router Advertisement (RA)
- Neighbor Solicitation (NS)
- Neighbor Advertisement (NA)
- Redirect

ND messages can include options that convey additional information. Currently, the following ND options are specified:

- Source link-layer address(SLLA) [RFC4861]
- Target link-layer address (TLLA) [RFC4861]
- Prefix information [RFC4861]
- Redirected header [RFC4861]
- MTU [RFC4861]
- Route Information [RFC4191]
- Recursive DNS Server (RDNSS) [RFC6106]
- DNS Search List (DNSSL) [RFC6106]

This memo specifies validation rules for the ND options mentioned above. In order to avoid pathological outcomes (such as [FreeBSD-rtsold]), IPv6 implementations validate incoming ND options using these rules.

## 2. Terminology

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].

## 3. Methodology

Section 4 through Section 11 of this document define validation rules for ND options. These sections also specify actions that are to be taken when an implementation encounters an invalid option. Possible actions are:

- The entire option MUST be ignored. However, the rest of the ND message MAY be processed.
- The entire ND message MUST be ignored

In the spirit of "being liberal in what you receive", the first action is always preferred. However, when an option length attribute is invalid, it is not possible to parse the rest of the ND message, and therefore subsequent ND options should be ignored.

We note that an implementation SHOULD NOT assume a particular length of an option (based on the option type) when it moves to the next option (whether it handles or ignores the current option) and SHOULD always use the length field of the option.

## 4. The Source Link-Layer Address (SLLA) Option

The SLLA Option is employed with NS, RS, and RA messages. If any other ND message contains an SLLA Option, the SLLA Option MUST be ignored. However, the rest of the ND message MAY be processed. (As per [RFC4861]).

Figure 1 illustrates the SLLA Option:

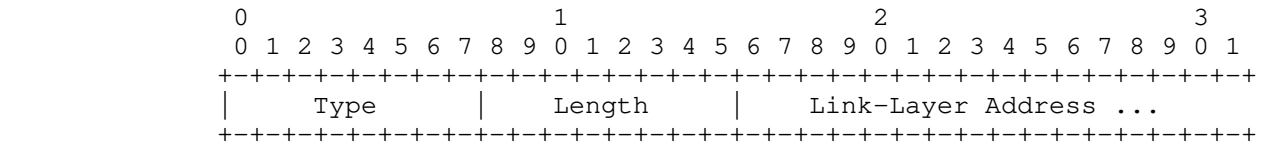



Figure 1: Source Link-Layer Address Option

The Type field is set to 1.

The Length field specifies the length of the option (including the Type and Length fields) in units of 8 octets. The Length field MUST be valid for the underlying link layer. For example, for IEEE 802 addresses the Length field MUST be 1 [RFC2464]. If an incoming ND message does not pass this validation check, the entire ND message MUST be discarded.

The Link-Layer Address field specifies the link-layer address of the packet's originator. It MUST NOT be any of the following:

- a broadcast address (see Appendix B for rationale)
- a multicast address (see Appendix B for rationale)
- an address belonging to the receiving node (see Appendix A for rationale)

If an incoming ND message does not pass this validation check, the SLLA Option MUST be ignored. However, the rest of the ND message MAY be processed.

An ND message that carries the SLLA Option MUST have a source address other than the unspecified address (0:0:0:0:0:0:0:0). If an incoming ND message does not pass this validation check, the SLLA Option MUST

be ignored. However, the rest of the ND message MAY be processed. (As per [RFC4861]).

## 5. The Target Link-Layer Address (TLLA) Option

NA and Redirect messages MAY contain a TLLA Option. If any other ND message contains an TLLA Option, the TLLA Option MUST be ignored. However, the rest of the ND message MAY be processed. (As per [RFC4861]).

Figure 2 illustrates the Target link-layer address:

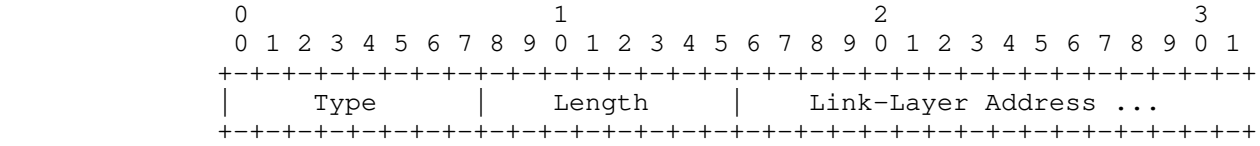



Figure 2: Target link-layer address option format

The Type field is set to 2.

The Length field specifies the length of the option (including the Type and Length fields) in units of 8 octets. The Length field MUST be valid for the underlying link layer. For example, for IEEE 802 addresses the Length field MUST be 1 [RFC2464]. If an incoming ND message does not pass this validation check, the entire ND message MUST be discarded.

An ND message that carries the TLLA option also includes a Target Address. The TLLA Option Link-Layer Address maps to the Target Address. The TLLA Option Link-Layer Address MUST NOT be any of the following:

- a broadcast address (see Appendix B for rationale)
- a multicast address (see Appendix B for rationale)
- an address belonging to the receiving node (see Appendix A for rationale)

If an incoming ND message does not pass this validation check, the TLLA Option MUST be ignored. However, the rest of the ND message MAY be processed.

## 6. The Prefix Information Option

The RA message MAY contain a Prefix Information Option. If any other ND message contains a Prefix Information Option, the Prefix Information Option MUST be ignored. However, the rest of the ND message MAY be processed. (As per [RFC4861]).

Figure 3 illustrates the Prefix Information Option:

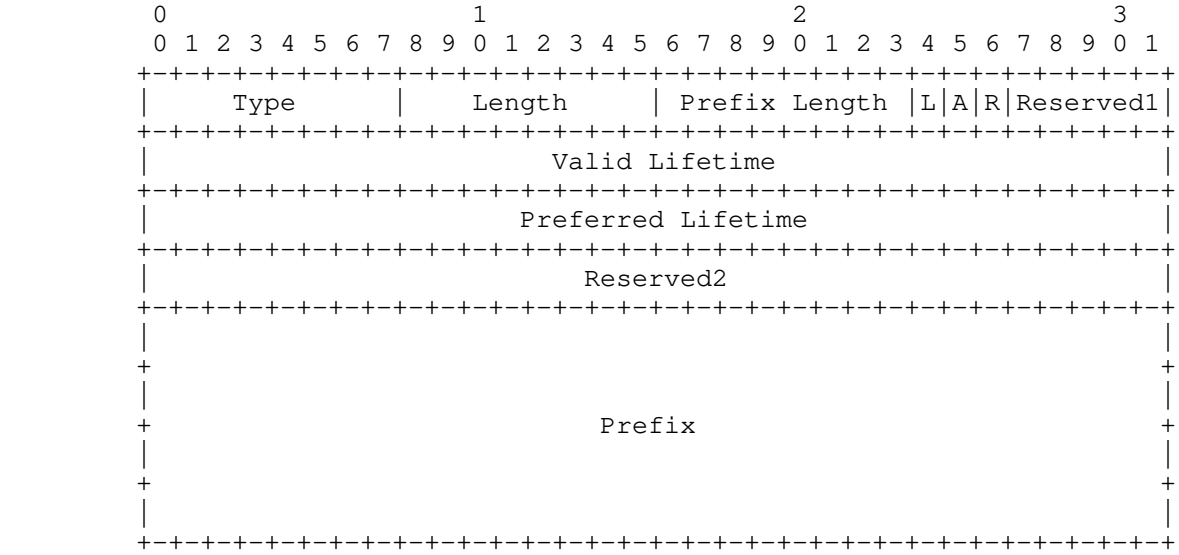



Figure 3: Prefix Information option format

The Type field is set to 3.

The Length field MUST be set to 4. If an incoming ND message does not pass this validation check, the entire ND message MUST be discarded.

As stated in [RFC4861] the Preferred Lifetime MUST be less than or equal to the Valid Lifetime. If an incoming ND message does not pass this validation check, the Prefix Information Option MUST be ignored. However, the rest of the ND message MAY be processed.

The Prefix Length contains the number of leading bits in the prefix that are to be considered valid. It MUST be greater than or equal to 0, and smaller than or equal to 128. If the field does not pass this check, the Prefix Information Option MUST be ignored. However, the rest of the ND message MAY be processed.

The Prefix field MUST NOT contain a link-local or multicast prefix. If an incoming ND message does not pass this validation check, the Prefix Information Option MUST be ignored. However, the rest of the ND message MAY be processed.

## 7. The Redirected Header Option

The Redirect message MAY contain a Redirect Header Option. If any other ND message contains an Redirect Header Option, the Redirect Header Option MUST be ignored. However, the rest of the ND message MAY be processed. (As per [RFC4861]).

Figure 4 illustrates the Redirected Header option:

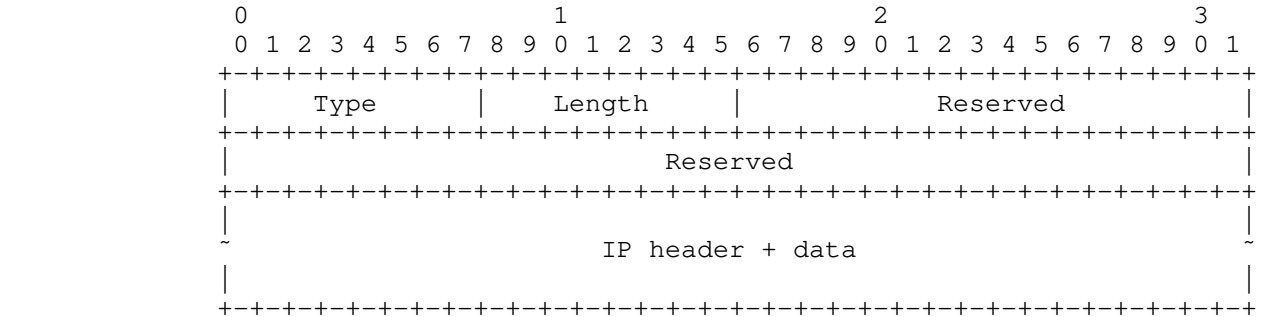



Figure 4: Redirected Header Option format

The Type field is 4.

The Length field specifies the option size (including the Type and Length fields) in units of 8 octets. Its value MUST be greater than or equal to 6. If an incoming ND message does not pass this validation check, the entire ND message MUST be discarded.

The value 6 was chosen to accommodate mandatory fields (8 octets) plus the base IPv6 header (40 octets).

## 8. The MTU Option

The RA message MAY contain an MTU Option. If any other ND message contains an MTU Option, the MTU Option MUST be ignored. However, the rest of the ND message MAY be processed. (As per [RFC4861]).

Figure 5 illustrates the MTU option:

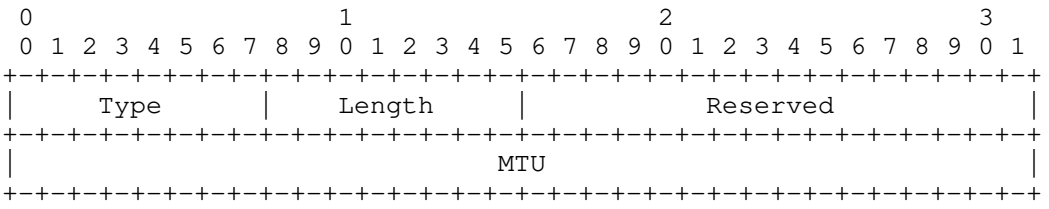



Figure 5: MTU Option Format

The Type field identifies the kind of option and is set to 5.

The Length field MUST BE set to 1 by the sender. If an incoming ND message does not pass this validation check, the entire ND message MUST be discarded.

The MTU field is a 32-bit unsigned integer that specifies the MTU value that should be used for this link. [RFC2460] specifies that the minimum IPv6 MTU is 1280 octets. Therefore, the MTU MUST be greater than or equal to 1280. If an incoming ND message does not pass this validation check, the MTU Option MUST be ignored. However, the rest of the ND message MAY be processed.

Additionally, the advertised MTU MUST NOT exceed the maximum MTU specified for the link-type (e.g., [RFC2464] for Ethernet networks). If an incoming ND message does not pass this validation check, the MTU Option MUST be ignored. However, the rest of the ND message MAY be processed.

## 9. The Route Information Option

The RA message MAY contain a Route Information Option. If any other ND message contains a Route Information Option, the Route Information Option MUST be ignored. However, the rest of the ND message MAY be processed.

Figure 6 illustrates Route Information option:

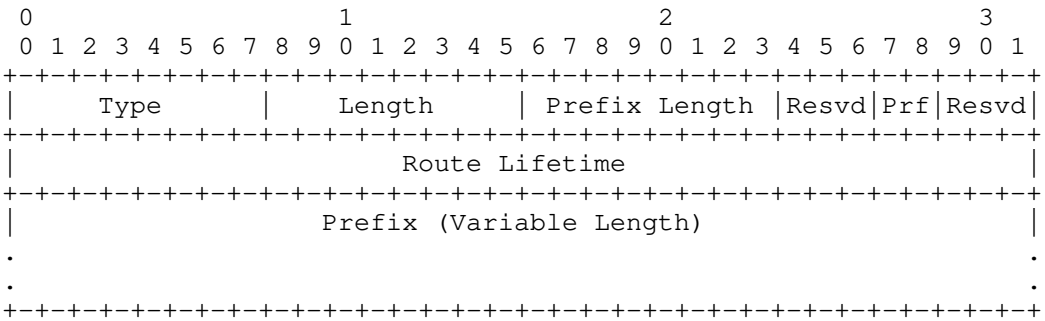



Figure 6: Route Information Option Format

The Type field is 24.

The Length field contains the length of the option (including the Type and Length fields) in units of 8 octets. Its value MUST be at least 1 and at most 3. If an incoming ND message does not pass this validation check, the entire ND message MUST be discarded.

The Prefix Length field indicates the number of significant bits in the Prefix field that are significant. Its value MUST be less than or equal to 128. If the field does not pass this check, the Route Information Option MUST be ignored.

The Length field and the Prefix Length field are closely related, as the Length field constrains the possible values of the Prefix Length field. If the Prefix Length is equal to 0, the Length MUST be equal to 1. If the Prefix Length is greater than 0 and less than 65, the Length MUST be equal to 2. If the Prefix Length is greater than 65 and less than 129, the Length MUST be equal to 3. If an incoming ND message does not pass this validation check, the entire ND message MUST be discarded.

The Prefix field MUST NOT contain a link-local unicast prefix (fe80::/10) or a link-local multicast prefix (e.g., ff02::/64). If an incoming ND message does not pass this validation check, the Route Information Option MUST be ignored. However, the rest of the ND message MAY be processed.

## 10. The Recursive DNS Server (RDNSS) Option

The RA message MAY contain a Recursive DNS Server (RDNSS) Option. If any other ND message contains an RDNSS Option, the RDNSS Option MUST be ignored. However, the rest of the ND message MAY be processed.

Figure 7 illustrates the syntax of the RDNSS option:

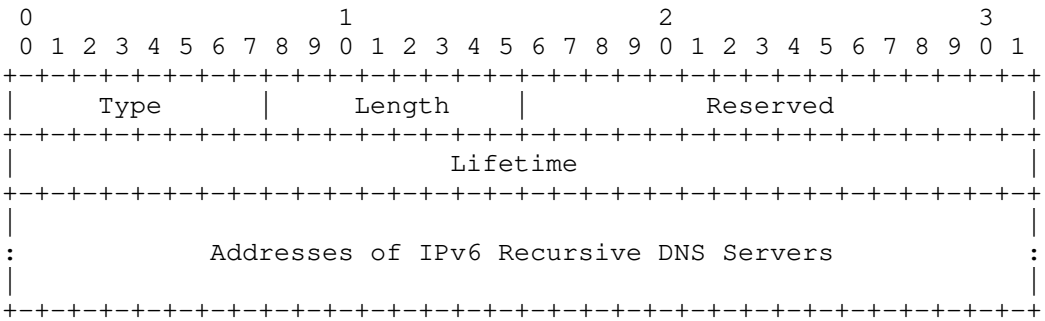



Figure 7: Recursive DNS Server Option Format

The Type field is 25.

The Length field specifies the length of the option (including the Type and Length fields) in units of 8 octets. Its value MUST be greater than or equal to 3. Additionally the Length field MUST pass the following check:

(Length -1) % 2 == 0

Figure 8

If the option does not pass these validation checks, the entire ND message MUST be discarded.

The RDNSS address list MUST NOT contain multicast addresses or the unspecified address. If an incoming ND message does not pass this validation check, the RDNSS Option MUST be ignored. However, the rest of the ND message MAY be processed.

## 11. The DNS Search List (DNSSL) Option

The RA message MAY contain a DNS Search List (DNSSL) Option. If any other ND message contains a DNSSL Option, the DNSSL Option MUST be ignored. However, the rest of the ND message MAY be processed.

Figure 9 illustrates the syntax of the DNSSL option:

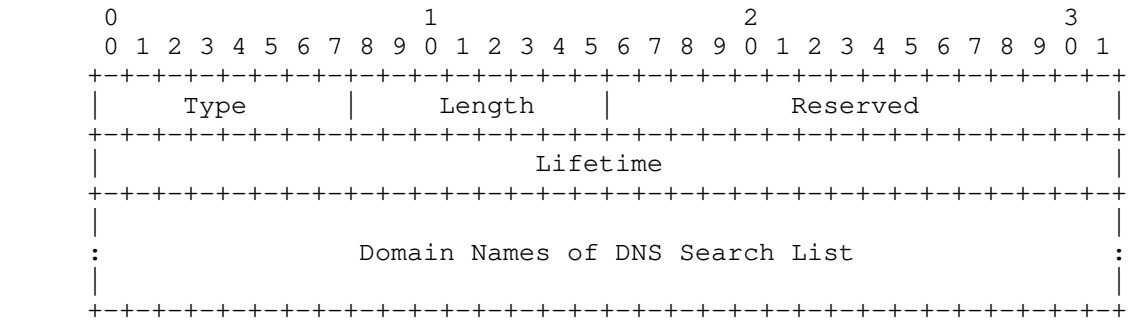



Figure 9: DNS Search List Option Format

The Type field is 31.

The Length field specifies the length of the option (including the Type and Length fields) in units of 8 octets. Its value MUST be greater than or equal to 2. If an incoming ND message does not pass these validation checks, the entire ND message MUST be discarded.

[RFC6106] specifies the valid format of domain suffixes. If a suffix is not validly encoded as specified, the corresponding DNSSL option MUST be ignored.

## 12. IANA Considerations

There are no IANA registries within this document. The RFC-Editor can remove this section before publication of this document as an RFC.

## 13. Security Considerations

This document specifies sanity checks to be performed on Neighbor Discovery options. By enforcing the checks specified in this document, a number of pathological behaviors (including some leading to Denial of Service scenarios) are eliminated.

## 14. Acknowledgements

Thanks to Tomoyuki Sahara and Jinmei Tatuya for their careful review and comments.

## 15. References

### 15.1. Normative References

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

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

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

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

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

[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.

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

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

### 15.2. Informative References

[FreeBSD-rtsold]
FreeBSD, , "rtsold(8) remote buffer overflow vulnerability", 2014, <https://www.freebsd.org/security/advisories/FreeBSD-SA-14:20.rtsold.asc>.

## Appendix A. Mapping an IPv6 Address to a Local Router's Own Link-layer Address

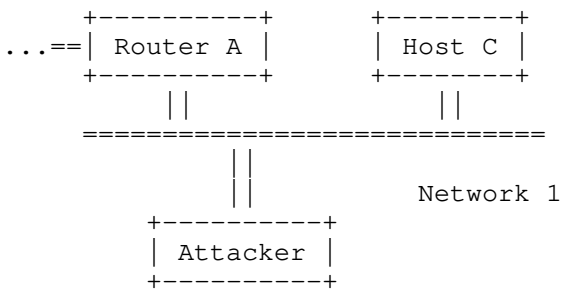



Figure 10: Unicast Forwarding Loop

In Figure 10, an on-link attacker sends Router A a crafted ND message that maps Host C's IPv6 address to the link-layer address of Router A's interface to Network 1. The crafted ND message causes Router A to map Host C's IPv6 address to the link layer address of its own interface to Network 1 and sets up the scenario for a subsequent attack.

A packet is sent to Router A with the IPv6 Destination Address of Host C. Router A forwards the packet on Network 1, specifying its own Network 1 interface as the link-layer destination. Because Router A specified itself as the link layer destination, Router A receives the packet and forwards it again. This process repeats until the IPv6 Hop Limit is decremented to 0 (and hence the packet is discarded). In this scenario, the amplification factor is equal to the Hop Limit minus one.

An attacker can realize this attack by sending either of the following:

- An ND message whose SLLA maps an IPv6 address to the link layer address of the victim router's (Router A's in our case) interface to the local network (Network 1 in our case)

- An ND message whose TLLA maps an IPv6 address to the link layer address of the victim router's (Router A's in our case) interface to the local network (Network 1 in our case)

## Appendix B. Mapping a Unicast IPv6 Address to A Broadcast Link-Layer Address

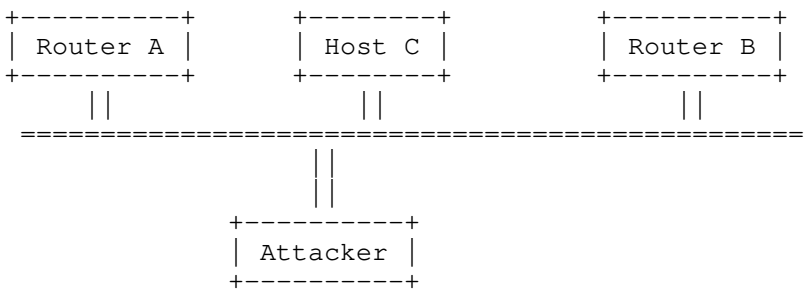



Figure 11: Broadcast Forwarding Loop

In Figure 11, the Attacker sends one crafted ND message to Router A, and one crafted ND message to Router B. Each crafted ND message contains the Target Address set to Host C's IPv6 address, and a TLLA option set to the Ethernet broadcast address (ff:ff:ff:ff:ff:ff). These ND messages causes each router to map Host C's IPv6 address to the Ethernet broadcast address. This sets up the scenario for a subsequent attack.

The Attacker sends a packet to the Ethernet broadcast address (ff:ff:ff:ff:ff:ff), with an IPv6 Destination Address equal to the IPv6 address of Host C. Upon receipt, both Router A and Router C decrement the Hop Limit of the packet, and resend it to the Ethernet broadcast address. As a result, both Router A and Router B receive two copies of the same packet (one sent by Router A, and another sent by Router B). This would result in a "chain reaction" that would only disappear once the Hop Limit of each of the packets is decremented to 0. The equation in Figure 12 describes the amplification factor for this scenario :

$$\text{Packets} = \sum_{x=0}^{\text{HopLimit}-1} \text{Routers}^{x}$$

Figure 12: Maximum amplification factor

This equation does not take into account ICMPv6 Redirect messages that each of the Routers could send, nor the possible ICMPv6 "time exceeded in transit" error messages that each of the routers could send to the Source Address of the packet when each of the "copies" of the original packet is discarded as a result of their Hop Limit being decremented to 0.

An attacker can realize this attack by sending either of the following:

- An ND message whose SLLA maps an IPv6 address not belonging to the victim routers to the broadcast link-layer address
- An ND message whose TLLA maps an IPv6 address not belonging to the victim routers to the broadcast link-layer address

An additional mitigation would be for routers to not forward IPv6 packets on the same interface if the link-layer destination address of the received packet was a broadcast or multicast address.

## Authors' Addresses

Fernando Gont
SI6 Networks / UTN-FRH
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fgont@si6networks.com
URI: http://www.si6networks.com

Ronald P. Bonica
Juniper Networks
2251 Corporate Park Drive
Herndon, VA 20171
US

Phone: 571 250 5819
Email: rbonica@juniper.net

Will (Shucheng) Liu
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Email: liushucheng@huawei.com

MIF
Internet-Draft
Intended status: Standards Track
Expires: August 28, 2016

J. Korhonen
Broadcom Limited
S. Krishnan
Ericsson
S. Gundavelli
Cisco Systems

February 25, 2016

# Support for multiple provisioning domains in IPv6 Neighbor Discovery Protocol
draft-ietf-mif-mpvd-ndp-support-03

## Abstract

The MIF working group is producing a solution to solve the issues that are associated with nodes that can be attached to multiple networks. One part of the solution requires associating configuration information with provisioning domains. This document details how configuration information provided through IPv6 Neighbor Discovery Protocol can be associated with provisioning domains.

## 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 August 28, 2016.

Table of Contents



## 1. Introduction

The MIF working group is producing a solution to solve the issues that are associated with nodes that can be attached to multiple networks based on the Multiple Provisioning Domains (MPVD) architecture work [RFC7556]. One part of the solution requires associating configuration information with Provisioning Domains (PVD). This document describes an IPv6 Neighbor Discovery Protocol (NDP) [RFC4861] mechanism for explicitly indicating provisioning domain information along with any configuration which is associated with that provisioning domain. The proposed mechanism uses an NDP option that indicates the identity of the provisioning domain and encapsulates the options that contain the configuration information as well as optional authentication/authorization information. The solution defined in this document aligns as much as possible with the existing IPv6 Neighbor Discovery security, namely with Secure Neighbor Discovery (SeND) [RFC3971].

## 2. Terminology

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. PVD Container option

The PVD container option (PVD_CO) is used to encapsulate the configuration options that belong to the explicitly identified provisioning domain. The PVD container option always encapsulates exactly one PVD identity. The PVD container option MAY occur multiple times in a Router Advertisement (RA) message. In this case each PVD container MUST belong to a different provisioning domain. The PVD container options MUST NOT be nested. The PVD Container option is defined only for the RA NDP message.

Since implementations are required to ignore any unrecognized options [RFC4861], the backward compatibility and the reuse of existing NDP options is implicitly enabled. Implementations that do not recognize the PVD container option will ignore it, and any PVD container option "encapsulated" NDP options without associating them into any provisioning domain (since the implementation has no notion of provisioning domains). For example, the PVD container could "encapsulate" a Prefix Information Option (PIO), which would mark that this certain advertised IPv6 prefix belongs and originates from a specific provisioning domain. However, if the implementation does not understand provisioning domains, then this specific PIO is also skipped and not configured on the interface.

The optional security for the PVD container is based on X.509 certificates [RFC6487] and reuses mechanisms already defined for SeND [RFC3971] [RFC6495]. However, the use of PVD containers does not assume or depend on SeND being deployed or even implemented. The PVD containers SHOULD be signed per PVD certificates, which provides both integrity protection and proves that the configuration information source is authorized for advertising the given information. See [RFC6494] for discussion how to enable deployments where the certificates needed to sign PVD containers belong to different administrative domains i.e., to different provisioning domains.

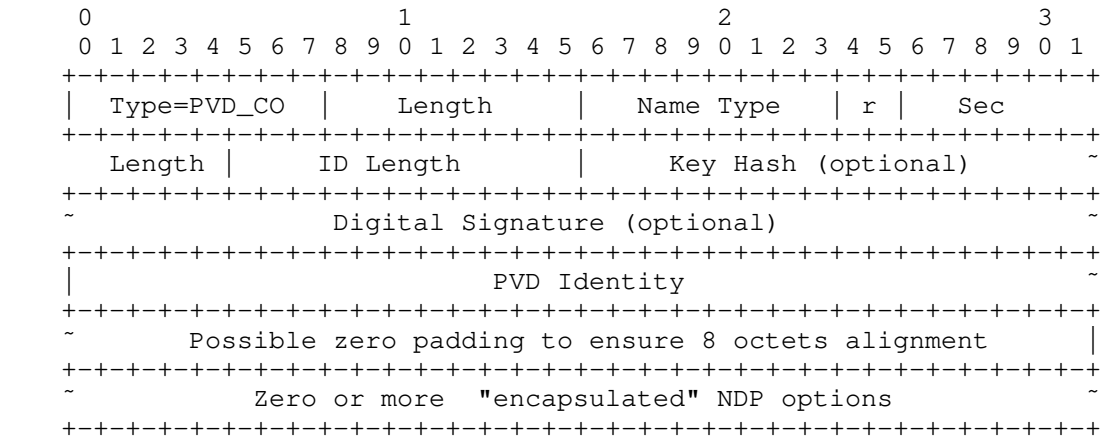



Figure 1: PVD Container Option

Type

PVD Container; Set to TBD1.

Length

Length of the PVD_CO. The actual length depends on the number of "encapsulated" NDP options, length of the PVD Identity, and the optional Key Hash/Digital Signature/Padding.

Name Type

Names the algorithm used to identify a specific X.509 certificate using the method defined for the Subject Key Identifier (SKI) extension for the X.509 certificates. The usage and the Name Type registry aligns with the mechanism defined for SeND [RFC6495]. Name Type values starting from 3 are supported and an implementation MUST at least support SHA-1 (value 3). Note that if Sec Length=0 the Name field serves no use and MUST be set to 0.

r

Reserved. MUST be set to 0 and ignored when received.

Sec Length

11-bit length of the Key Hash and Digital Signature in a units of 1 octet. When no security is enabled the Sec Length MUST be set to value of 0.

ID Length

11-bit length of the PVD Identity in a units of 1 octet. The ID Length MUST be greater than 0.

Key Hash

This field is only present when Sec Length>0. A hash of the public key using the algorithm identified by the Name Type. The procedure how the Key Hash is calculated is defined in [RFC3971] and [RFC6495].

Digital Signature

This field is only present when Sec Length>0. A signature calculated over the PVD_CO option including all option data from the beginning of the option until to the end of the container. The procedure of calculating the signature is identical to the one defined for SeND [RFC3971]. During the signature calculation the contents of the Digital Signature option MUST be treated as all zero.

PVD Identity

The provisioning domain identity. The contents of this field is defined in a separate document [I-D.ietf-mif-mpvd-id].

Implementations MUST ensure that the PVD container option meets the 8 octets NDP option alignment requirement as described in [RFC4861].

If the PVD_CO does not contain a digital signature, then other means to secure the integrity of the NDP message SHOULD be provided, such as utilizing SeND. However, the security provided by SeND is for the entire NDP message and does not allow verifying whether the sender of the NDP message is actually authorized for the information for the provisioning domain.

If the PVD_CO contains a signature and the verification fails, then the whole PVD_CO option MUST be silently ignored and the event SHOULD be logged.

## 4. Set of allowable options

The PVD container option MAY be used to encapsulate any allocated IPv6 NDP options, which may appear more than once in a NDP message. The PVD container option MUST NOT be used to encapsulate other PVD_CO option(s).

## 5. Security Considerations

An attacker may attempt to modify the information provided inside the PVD container option. These attacks can easily be prevented by using SeND [RFC3971] or per PVD container signature that would detect any form of tampering with the IPv6 NDP message contents.

A compromised router may advertise configuration information related to provisioning domains it is not authorized to advertise. e.g. A coffee shop router may provide configuration information purporting to be from an enterprise and may try to attract enterprise related traffic. The only real way to avoid this is that the provisioning domain container contains embedded authentication and authorization information from the owner of the provisioning domain. Then, this attack can be detected by the client by verifying the authentication and authorization information provided inside the PVD container option after verifying its trust towards the provisioning domain owner (e.g. a certificate with a well-known/common trust anchor).

A compromised configuration source or an on-link attacker may try to capture advertised configuration information and replay it on a different link or at a future point in time. This can be avoided by including some replay protection mechanism such as a timestamp or a nonce inside the PVD container to ensure freshness of the provided information. This specification does not define a replay protection solution. Rather it is assumed that if replay protection is required, the access network and hosts also deploy existing security solutions such as SeND [RFC3971].

## 6. IANA Considerations

This document defines two new IPv6 NDP options into the "IPv6 Neighbor Discovery Option Formats" registry. Option TBD1 is described in Section 3.

## 7. Acknowledgements

The authors would like to thank the members of the MIF architecture design team for their comments that led to the creation of this draft.

## 8. References

### 8.1. Normative References

[I-D.ietf-mif-mpvd-id]
Krishnan, S., Korhonen, J., Bhandari, S., and S. Gundavelli, "Identification of provisioning domains", draft-ietf-mif-mpvd-id-02 (work in progress), October 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <http://www.rfc-editor.org/info/rfc3971>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <http://www.rfc-editor.org/info/rfc4861>.

[RFC6494] Gagliano, R., Krishnan, S., and A. Kukec, "Certificate Profile and Certificate Management for SEcure Neighbor Discovery (SEND)", RFC 6494, DOI 10.17487/RFC6494, February 2012, <http://www.rfc-editor.org/info/rfc6494>.

[RFC6495] Gagliano, R., Krishnan, S., and A. Kukec, "Subject Key Identifier (SKI) SEcure Neighbor Discovery (SEND) Name Type Fields", RFC 6495, DOI 10.17487/RFC6495, February 2012, <http://www.rfc-editor.org/info/rfc6495>.

## 8.2. Informative References

[RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, DOI 10.17487/RFC6487, February 2012, <http://www.rfc-editor.org/info/rfc6487>.

[RFC7556] Anipko, D., Ed., "Multiple Provisioning Domain Architecture", RFC 7556, DOI 10.17487/RFC7556, June 2015, <http://www.rfc-editor.org/info/rfc7556>.

# Appendix A. Examples

### A.1. One implicit PVD and one explicit PVD

Figure 2 shows how the NDP options are laid out in an RA for one implicit provisioning domain and one explicit provisioning domain. The example does not include security (and signing of the PVD container). The assumption is the PVD identity consumes total 18 octets (for example encoding a NAI Realm string "dana.example.com").

The explicit provisioning domain contains a specific PIO for 2001:db8:abad:cafe::/64 and the MTU of 1337 octets. The implicit provisioning domain configures a prefix 2001:db8:cafe:babe::/64 and the link MTU of 1500 octets. There are two cases: 1) the host receiving the RA implements provisioning domains and 2) the host does not understand provisioning domains.

1. The host recognizes the PVD_CO and "starts" a provisioning domain specific configuration. Security is disabled, thus there are no Key Hash or Digital Signature fields to process. The prefix 2001:db8:abad:cafe::/64 is found and configured on the interface. Once the PVD_ID option is located the interface prefix configuration for 2001:db8:abad:cafe::/64 and the MTU of 1337 octets can be associated to the provisioning domain found in the PVD_CO option.

   The rest of the options are parsed and configured into the implicit provisioning domain since there is no encapsulating provisioning domain. The interface is configured with prefix 2001:db8:cafe:babe::/64. The implicit provisioning domain uses the link MTU of 1500 octets, whereas the "dana.example.com" provisioning domain uses the MTU of 1337 octets (this means when packets are sourced using 2001:db8:abad:cafe::/64 prefix the link MTU is different than when sourcing packets using 2001:db8:cafe:babe::/64 prefix).

2. The host ignores the PVD_CO and ends up configuring one prefix on its interface ( 2001:db8:cafe:babe::/64) with a link MTU of 1500 octets.

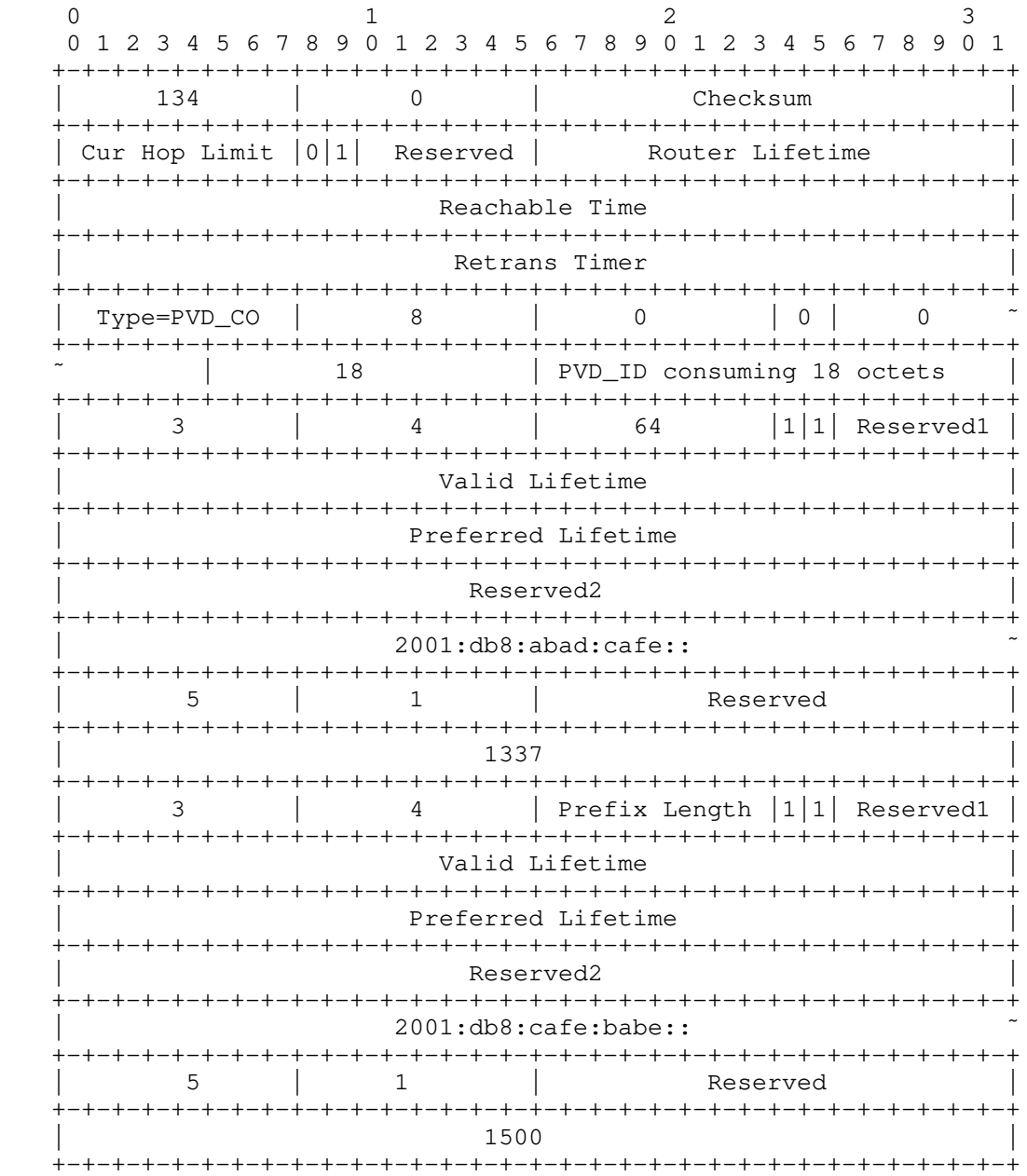



Figure 2: An RA with one implicit PVD and one explicit PVD

Authors' Addresses

Jouni Korhonen
Broadcom Limited
3151 Zanker Road
San Jose, CA 95134
USA

Email: jouni.nospam@gmail.com

Suresh Krishnan
Ericsson
8400 Decarie Blvd.
Town of Mount Royal, QC
Canada

Phone: +1 514 345 7900 x42871
Email: suresh.krishnan@ericsson.com

Sri Gundavelli
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
USA

Email: sgundave@cisco.com

6MAN
Internet-Draft
Intended status: Informational
Expires: April 3, 2016

E. Nordmark
Arista Networks

Oct 2015

# Possible approaches to make DAD more robust and/or efficient
## draft-nordmark-6man-dad-approaches-02

Abstract

This outlines possible approaches to solve the issues around IPv6 Duplicate Address Detection robustness and/or efficiency which are specified in the "DAD issues" dradft.

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 3, 2016.

Table of Contents

## 1. Introduction

Duplicate Address Detection (DAD) is a procedure in IPv6 performed on an address before it can be assigned to an interface [RFC4862]. By default it consists of sending a single multicast Neighbor Soliciation message and waiting for a response for one second. If no response is received, the address is declared to not be a duplicate. Once the address has been tested once, there is no further attempts to check for duplicates (unless the interface is re-initialized).

The companion document [I-D.yourtchenko-6man-dad-issues] outlines a set of issues around Duplicate Address Detection (DAD) which either result in reduced robustness, or result in lower efficiency for either the hosts wanting to sleep or the network handling more multicast packets.

The reader is encourage to review the issues in that document. In summary, the lack of robustness is due to only sending one or a few DAD probe initially, and not having any positive acknowledgement that "there are no duplicates". This implies that partioned links that later heal can result in persistent undetected duplicate IPv6 addresses, including cases of "local partitions" such as the case of a modem not having connected when the DAD probes are sent. The inefficiences appears when there are low-powered devices on the link that wish to sleep a significant amount of time. Such devices must either be woken up by multicast Neighbor Soliciations sent to one of their solicited-node multicast addresses, or they need to redo DAD each time they wake up from sleep. Both drain the battery; the second one results in sending a DAD probe and then waiting for a second with the radion receiver enabled to see if a DAD message indicates a duplicate.

## 2. Robustness Solution Approaches

IPv4 ARP robustness against partitions and joins is greately improved by Address Conflict Detection (ACD) [RFC5227]. That approach is leverages the fact that ARP requests are broadcast on the link and also makes the ARP replies be broadcast on the link. That combination means that a host can immediately detect when some other host provides a different MAC address for what the host thinks is its own IPv4 address. That is coupled with state machines and logic for determining whether to try to reclaim the address or give up and let the other host have it. When giving up the host will form a new IPv4 address. The ACD approach results in more broadcast traffic than normal ARP [RFC0826] since the ARP replies are broadcast.

Applying the same approach to IPv6 would require sending the Neighbor

Solicitations and Neighbor Advertisements to the all-nodes multicast address so that a host can see when a different host is claiming/ using the same source IPv6 address. That would remove the efficiency that Neighbor Discovery gets from "spreading" the resolution traffic over 4 million multicast addresses.

One can envision variants on the theme of ACD that fit better with the use of solicited-node multicast addresses. Suppose we have Host1 with IP1 that hashes to solicited-node multicast address SN1. And we also have Host2 with IP2 and SN2.The link-layer addresses are MAC1 and MAC2, respectively. In [RFC4861] when Host1 wants to communicate with Host2 we will see

1. Host1 multicasts a NS from IP1 to SN2. That include a claim for IP1->MAC1 using the Source Link-layer Address option.
2. Host2 receives the NA and unicasts a NA from IP2 to IP1. That includes a claim for IP2->MAC using the Taget Link-layer Address option.

If we want other hosts which might think they own either IP1 or IP2 to see the NA or NS (and we don't want to send the NS and NA to all-nodes), then we can add additional multicast packets which explicitly send the claim and send it to the Solicited-node multicast address of the address that is being claimed. Thus

1. Host1 multicasts a NS from IP1 to SN2. That include a claim for IP1->MAC1 using the Source Link-layer Address option.
2. Host1 multicasts a NA from IP1 to SN1 explicitly claiming IP1->MAC1 using the TLLAO.
3. Host2 receives the NA and unicasts a NA from IP2 to IP1. That includes a claim for IP2->MAC using the Taget Link-layer Address option.
4. Host2 multicasts a NA from IP2 to SN2 explicitly claiming IP2->MAC2 using the TLLAO.

The above explicit claims can then trigger the state machine described in ACD. The claims can probably be rate limited for any given source address since there is no need to repeat the claim just because a NS needs to be sent for a new IP3 etc. The impact of such rate limitig on the ability to detect duplicates.

In the worst case the above approach turns one multicast and one unicast into three multicasts and one unicast, but all the multicasts are sent to solicited-node multicast addresss. Thus a host would not need to process the additional multicast packets.

This ACD-multicast approach assumes that the multicast packets are delivered with reasonable reliability, but does not assume perfect delivery. If multicast reliability is lower than unicast it will result in retransmitted multicast NS in [RFC4861]. However, the

above rate limiting idea might need care to ensure that claims are re-transmitted when the NS is re-transmitted.

A slightly different approach to on-going DAD is what is implemented in Solaris where the node sends a periodic NA annoucement for the address it is using, plus the ACD behavior of detecting such an NA with a conflicting address. Presumably the NA annoucement can be sent to the solicited-node multicast address. It might make sense to use the Nonce option used by [I-D.ietf-6man-enhanced-dad] to avoid issues where a host would hear its own announcement.

## 3. Approaches to efficiency

There exists some form of sleep proxies [ECMA-393][http://en.wikipedia.org/wiki/Bonjour_Sleep_Proxy] which perform handover of Neighbor Discovery protocol processing. [ECMA-393] does not specify the handover mechanism, and there is no know dcumentation for the handover mechanism. Even though the details are not specified, the approach seems to allow a host to sleep without worrying about DAD; the sleep proxy will respond to DAD probes. This seems to entail sending multicast NAs to all-nodes to hand-over the IP address to the proxy's MAC address before going to sleep and then again to hand it back to the host's MAC address when it wakes up.

It is not clear whether such sleep proxies provide protection against Single Points of Failure i.e., whether the host can hand over things to a pair of sleep proxies.

FCFS SAVI [RFC6620] builds up state in devices to be able to detect and prevent when some host is trying to use an IPv6 address already used by another host on the link. This binding is built and checked for DAD packets, but also for data packets to ensure that an attacker can not inject a data packet with somebody elses source address. When FCFS SAVI detects a potential problem it checks whether the IPv6 address has merely moved to a different binding anchor (e.g., port on the switch) by sending a probe to its old anchor. Thus it assumes the host is always awake or can be awoken to answer that probe. Futhermore, implementation of the data triggered aspects can run into hardware limitations since it requires something like an ACL for every IPv6 address which has been validated.

DAD proxies as specified in [RFC6957] was designed to handle split-horizon forwarding which means that a host would never receive a multicast DAD probe sent by another host. This approach maintains a binding cache built up by DAD probes and checked when handling DAD probes. However, just like SAVI in order to handle host mobility and legitimate host MAC address change, it the case of a potential

conflict the proxy ends up verifying whether the IP address is still present at its old port and MAC address. Hence the host can not sleep.

One could explore something along the SAVI and DAD proxy approach that uses timestamps to allow better sleep. In principle would could start some fixed timer each time an IPv6 address is added or updated in the binding cache, and during that time the proxy would respond to DAD probes on behalf of the (potentially sleeping) host. To enable movement between ports/anchors such an approach would have to compare MAC address and assume that if the MAC address is the same it is the same host. (Unclear whether that is a good idea if we end up with random MAC addresses for better privacy.) And if a host would like to change its MAC address it would need to wait for the timeout before it can succeed in doing the change. Thus on one hand one would want a long time (24 hours?) to facilitate for sleeping hosts, and on the other hand a short time to allow for MAC address change and movement.

In essence the above forms an implicit request for the proxy to handle DAD on behalf of the host, with a fixed time limit. If the host can instead make that time explicit, then the host can also remove the proxy behavior (by passing a time of zero). Such a "proxy for me" request can leverage the ARO option defined for 6LoWPan in [RFC6775] but use it only for the purposes of DAD offload to the proxy. That option can also carry an additional identifier which can be used to distinguish between the same host aka same identifier changing the MAC address. In the RFC that is an EUI-64 and in [I-D.chakrabarti-nordmark-energy-aware-nd] in is a more generalized identifier field. For redundancy the ARO can be sent to more than one proxy.

## 4. Security Considerations

If the working group decides to pursue one of the outlined approaches to improve the robustness and/or efficiency of DAD, then the security issues for that partiticular approach will need to be studied.

In general DAD is subject to a Denial of Service attack since a malicious host can claim all the IPv6 addresses [RFC4218].

## 5. Acknowledgements

Sowmini Varadhan pointed out the Solaris approach to use periodic NA annoucements to increase robustness.

# 6. References

## 6.1. Normative References

[I-D.yourtchenko-6man-dad-issues]
Yourtchenko, A. and E. Nordmark, "A survey of issues related to IPv6 Duplicate Address Detection", draft-yourtchenko-6man-dad-issues-01 (work in progress), March 2015.

[RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826, November 1982, <http://www.rfc-editor.org/info/rfc826>.

[RFC4218] Nordmark, E. and T. Li, "Threats Relating to IPv6 Multihoming Solutions", RFC 4218, DOI 10.17487/RFC4218, October 2005, <http://www.rfc-editor.org/info/rfc4218>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <http://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <http://www.rfc-editor.org/info/rfc4862>.

[RFC5227] Cheshire, S., "IPv4 Address Conflict Detection", RFC 5227, DOI 10.17487/RFC5227, July 2008, <http://www.rfc-editor.org/info/rfc5227>.

[RFC6620] Nordmark, E., Bagnulo, M., and E. Levy-Abegnoli, "FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses", RFC 6620, DOI 10.17487/RFC6620, May 2012, <http://www.rfc-editor.org/info/rfc6620>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <http://www.rfc-editor.org/info/rfc6775>.

[RFC6957] Costa, F., Combes, J-M., Ed., Pougnard, X., and H. Li, "Duplicate Address Detection Proxy", RFC 6957,

DOI 10.17487/RFC6957, June 2013,
<http://www.rfc-editor.org/info/rfc6957>.

## 6.2. Informative References

[I-D.chakrabarti-nordmark-energy-aware-nd]
Chakrabarti, S., Nordmark, E., and M. Wasserman, "Energy Aware IPv6 Neighbor Discovery Optimizations", draft-chakrabarti-nordmark-energy-aware-nd-02 (work in progress), March 2012.

[I-D.ietf-6man-enhanced-dad]
Asati, R., Singh, H., Beebee, W., Pignataro, C., Dart, E., and W. George, "Enhanced Duplicate Address Detection", draft-ietf-6man-enhanced-dad-15 (work in progress), March 2015.

## Author's Address

Erik Nordmark
Arista Networks
Santa Clara, CA
USA

Email: nordmark@arista.com

6man WG
Internet-Draft
Updates: 4861 (if approved)
Intended status: Standards Track
Expires: April 30, 2015

E. Nordmark
Arista Networks
A. Yourtchenko
Cisco
S. Krishnan
Ericsson

October 27, 2014

# IPv6 Neighbor Discovery Optional Unicast RS/RA Refresh
# draft-nordmark-6man-rs-refresh-01

## Abstract

IPv6 Neighbor Discovery relies on periodic multicast Router Advertisement messages to update timer values and to distribute new information (such as new prefixes) to hosts. On some links the use of periodic multicast messages to all host becomes expensive, and in some cases it results in hosts waking up frequently. Many implementations of RFC 4861 also use multicast for solicited Router Advertisement messages, even though that behavior is optional.

This specification provides an optional mechanism for hosts and routers where instead of periodic multicast Router Advertisements the hosts are instructed (by the routers) to use unicast Router Solicitations to request refreshed Router Advertisements. This mechanism is enabled by configuring the router to include a new option in the Router Advertisement in order to allow the network administrator to choose host behavior based on whether periodic multicast are more efficient on their link or not. The routers can also tell whether the hosts are capable of the new behavior through a new flag in the Router Solicitations.

## 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 30, 2015.

Table of Contents

## 1. Introduction

IPv6 Neighbor Discovery [RFC4861] was defined at a time when local area networks had different properties than today. A common link was the yellow-coax shared wire Ethernet, where a link-layer multicast and unicast worked the same - send the packet on the wire and the interested receivers will pick it up. Thus the network cost (ignoring any processing cost on the receivers that might not completely filter out Ethernet multicast addresses that they did not want) and the reliability of sending a link-layer unicast and multicast was the same. Furthermore, the hosts at the time was always on and connected. Powering on and off the workstation/PC hosts at the time was slow and disruptive process.

Under the above assumptions it was quite efficient to maintain the shared state of the link such as the prefixes and their lifetimes using periodic multicast Router Advertisement messages. It was also efficient to use multicast Neighbor Solicitations for address resolution as a slight improvement over the broadcast use in ARP. And finally, checking for a potential duplicate IPv6 address using multicast was efficient and natural.

There are still links, such a satellite links, where periodic multicast advertisements is the most efficient and reliable approach to keep the hosts up to date. However other links have different performance and reliability for multicast than for unicast (see for instance [I-D.vyncke-6man-mcast-not-efficient] which discusses WiFi links). Cellular networks which employ paging and support sleeping hosts have different issues (see e.g., [I-D.garneij-6man-nd-m2m-issues] that would benefit from having the hosts wake up and request information from the routers instead of the routers periodically multicasting the information.

Since different links types and deployments have different needs, this specification provides mechanism by which the routers can determine whether all the hosts support the RS refresh, and the hosts only employ the RS refresh when instructed by the routers using an option in the Router Advertisement.

The operator retains the option to use unsolicited multicast Router Advertisement to announce new or removed information. That can be useful for uncommon cases while allowing using a higher refresh time for normal network operations.

The specification does not assume that all hosts on the link implement the new capability. As soon as there are router(s) on a link which supports these optimizations, then the updated hosts on the link can sleep better, while co-existing on the same link with

unmodified hosts.

## 2. Goals and Requirements

The key goal is to allow the operator to choose whether unicast RS refresh is more efficient than periodic multicast RAs, while preserving the timely and scalable reconfiguration capabilities that a periodic RA model provides.

In addition, an operator might want to be notified whether the link includes hosts that do not support the new mechanism. Potential router implementations can react dynamically to that information, or can log events to system management when hosts appear which do not implement this new capability.

The assumption is that host which implement this specification also implement [I-D.ietf-6man-resilient-rs] as that ensures resiliency to packet loss at host initialization.

## 3. Definition Of Terms

The keywords "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].

## 4. Protocol Overview

The hosts include a new flag in the Router Solicitation message, which allows the routers to report to system management whether there are hosts that do not support the RS refresh on the link.

If the network administrator has configured the routers to send the new Refresh Timer option, then the option will be included in all the Router Advertisements. This option includes the time interval when the hosts should unicast Router Solicitations.

The host maintains the value of the Refresh Timer option (RTO) by recording it in the default router list. A value of zero can be used to indicate that a router did not include a Refresh Timer option.

The host calculates a timeout after it has received a RTO - either per router or per link. If it is maintained per link then the host SHOULD use the minimum Refresh Timer it has received from the routers on the link. The timeout is a random value uniformly distributed between 0.5 and 1.5 times the Refresh Timer value (in order to avoid

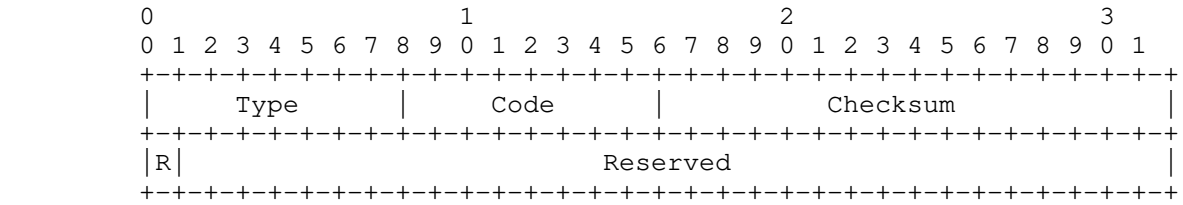
synchronization of the timers across hosts. [TBD: Add SYNC reference from RFC 4861.] When this timer fires the host sends one unicast Router Solicitation to the router (if maintained per router) or to all the routers in the default router list for link (if maintained per link.)

## 5. New Neighbor Discovery Flags and Options

This specification introduces a option used in the RAs which both indicates that the router can handle RS refresh using unicast RA, and a flag for the RS that indicates to the router that the host will do RS refresh if the router so wishes.

### 5.1. Introducing a Router Solicitation Flag

A node which implements this specification sets the R flag in all the Router Solicitation messages it sends. That allows the router to determine whether there are legacy hosts on the link.



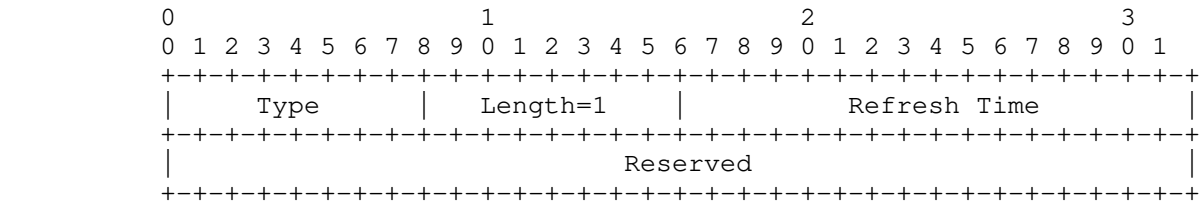

New fields:

R-flag: When set indicates that the sending node is capable of doing unicast RS refresh.

Reserved: Field is reduced from 32 bits to 31 bits. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.

### 5.2. Refresh Time option

A router which implements this specification can be configured to operate without periodic multicast Router Advertisements. When the operator configures this mode of operation, then the router MUST include this new option in the RA.




Fields:

Type: TBD ND option code value (IANA)

Length: 8-bit unsigned integer. The length of the option (including the type and length fields) in units of 8 bytes. The value 0 is invalid. Value is 1 for this option.

Refresh Time: 16-bit unsigned integer. Units is seconds. The all-ones value (65535) means infinite.

Reserved: 32 bits. This field is unused. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.

## 6. Conceptual Data Structures

In addition to the Conceptual Data structures in [RFC4861] a host records the received RTO value in the default router list. It also maintains a timeout - either per link or per default router.

## 7. Host Behavior

See Protocol Overview section above.

A host implementing this specification SHOULD also implement [I-D.ietf-6man-resilient-rs]. That ensures that a router that has been configured to not send periodic RA messages will always receive an RS from the host as the host initializes.

If there is no RTO in the received Router Advertisements, then the host behavior does not change. However, if RTOs start appearing in RAs after the initial RAs, the host SHOULD start performing RS refresh. As the last router that included RTO options time out from the default router list, the host SHOULD stop sending RS refresh messages.

The host MUST join the all-nodes multicast address as in [RFC4861] since the routers MAY send multicast RAs for important changes.

Some links might have routers with different configuration where some router includes RTO in the RA and others do not. Hosts MAY make the simplifying assumption that if any router on the link includes RTO then the host can use RS refresh to all the routers in the default router list. Also, the routers might advertise different refresh time, and hosts MAY use the minimum of the time received from any router that remains in the default router list. Note that section Section 9 says that routers SHOULD report such inconsistencies to system management.

### 7.1. Sleep and Wakeup

The protocol allows the sleepy nodes to complete its sleep schedule without waking up due to multicast Router Advertisement messages and the host is not required to wake up solely for the purposes of performing RS refresh. This assumes that sleepy nodes perform a RS refresh when they wake up. If hosts do wake up due to multicast RAs, then the host only needs to perform a refresh on wakeup if the Refresh timeout has expired while the host was sleeping.

### 7.2. Movement

When a host wakes up it can combine movement detecting (DNA), NUD, and refreshing its prefixes etc by sending a unicast RS to each of its existing default router(s). If it receives unicast RA from a router, then it can mark the router as REACHABLE.

Note that DNA [RFC6059] specifies using NS messages since many IPv6 routers delay (and multicast) solicited RAs and DNA wants to avoid that delay. Routers which implement this specification SHOULD unicast solicited RAs, hence if a router included the RTO then the host can use RS for DNA. For non-RTO routers the host MAY choose to use NS for DNA as in [RFC6059].

## 8. Router Behavior

See Protocol Overview section.

A router implementing this specification (and including RTO in the RAs) SHOULD also respond to unicast RS messages (that do not have an unspecified source address) with unicast RAs. If a RS message has an unspecified source address then the host MAY respond with a RA unicast at layer 2 (sent to the link-layer source address of the RS), or it MAY follow the rate-limited multicast RA procedure in

[RFC4861].

The RECOMMENDED default configuration for routers is to have RTO disabled.

### 8.1. Router and/or Interface Initialization

This specification does not change the initialization procedure. Thus a router multicasts some initial Router Advertisements (MAX_INITIAL_RTR_ADVERTISEMENTS) at system startup or interface initialization as specified in [RFC4861] and its updates.

### 8.2. Periodic Multicast RA for unmodified hosts

By default a router MUST send periodic multicast RAs as specified in [RFC4861]. A router can be configured to omit those, which can be used in particular deployments. If they are omitted, then there MUST be a mechanism to prevent or detect the existence of unmodified hosts on the link. That be be performed at deployment time (e.g., only hosts which are known to support RTO are configured with the layer 2 security keys), or the routers detect any RSs which do not include the R-flag and report this to system management, or dynamically enable periodic multicast RAs when observing at least one RS without the R-flag.

Note that such dynamic detection is not bullet proof. If a host does not implement RS refresh nor implements resilient RS [I-D.ietf-6man-resilient-rs], then the host might receive a multicast RA (from router initialization or the periodic multicast RAs) without the router ever receiving a RS from the host. Such a host would function as long as the routers are sending periodic multicast RAs.

### 8.3. Unsolicited RAs to share new information

When a router has new information to share (new prefixes, prefixes that should be immediately deprecated, etc) it MAY multicast up to MAX_INITIAL_RTR_ADVERTISEMENTS number of Router Advertisements.

On links where multicast is expensive the router MAY instead unicast up to MAX_INITIAL_RTR_ADVERTISEMENTS number of Router Advertisements to the hosts in its neighbor cache.

. Note that such new information is not likely to reach sleeping hosts until those hosts refresh by sending a RS.

## 9. Router Advertisement Consistency

The routers follows section 6.2.7 in [RFC4861] by receiving RAs from other routers on the link. In addition to the checks in that section, the routers SHOULD verify that the RTO have the same Refresh Time, and report to system management if they differ. While the host will pick the lowest time and operate correctly, it is not useful to use different Refresh Times for different routers.

## 10. Security Considerations

These optimizations are not known to introduce any new threats against Neighbor Discovery beyond what is already documented for IPv6 [RFC3756].

Section 11.2 of [RFC4861] applies to this document as well.

The mechanisms in this document work with SeND [RFC3971].

## 11. IANA Considerations

A new flag (R-flag) in the Router Solicitation message has been introduced by carving out a bit from the Reserved field. There is currently no IANA registry for RS flags. Perhaps one should be created?

This document needs a new Neighbor Discovery option type for the RTO.

## 12. Acknowledgements

The original idea came up in a discussion with Suresh Krishnan. Comments from Erik Kline and Samita Chakrabarti have helped improve the document.

This document has been discussed in the efficient-nd design team.

## 13. Open Issues

Should we make the Refresh Time 32 bits instead of 16? 16 bits implies maximum of 18 hours and in some deployments a refresh time measured in days might be desirable.

Should we update the DNA procedures [RFC6059]? We can use a unicast RS with this approach since that will result in an

immediate unicast RA which would include any updated prefixes. Note that a RS can not have an unspecified source and a SLLAO, hence some care would be needed in the interaction with DAD.

Would it be worth-while to try to remove unchanged information from the refreshed RAs? If so it could be done by including some epoch number in the RS and RA, and if the RS contains the current epoch then the RA would not need to include any options except the epoch number indicating that none of the options are the same as before.

## 14. References

### 14.1. Normative References

[I-D.ietf-6man-resilient-rs]
Krishnan, S., Anipko, D., and D. Thaler, "Packet loss resiliency for Router Solicitations", draft-ietf-6man-resilient-rs-04 (work in progress), October 2014.

[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.

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

### 14.2. Informative References

[I-D.garneij-6man-nd-m2m-issues]
Garneij, F., Chakrabarti, S., and S. Krishnan, "Impact of IPv6 Neighbor Discovery on Cellular M2M Networks", draft-garneij-6man-nd-m2m-issues-00 (work in progress), July 2014.

[I-D.vyncke-6man-mcast-not-efficient]
Vyncke, E., Thubert, P., Levy-Abegnoli, E., and A. Yourtchenko, "Why Network-Layer Multicast is Not Always Efficient At Datalink Layer", draft-vyncke-6man-mcast-not-efficient-01 (work in progress), February 2014.

[RFC3756] Nikander, P., Kempf, J., and E. Nordmark, "IPv6 Neighbor

Discovery (ND) Trust Models and Threats", RFC 3756, May 2004.

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

[RFC6059] Krishnan, S. and G. Daley, "Simple Procedures for Detecting Network Attachment in IPv6", RFC 6059, November 2010.

## Authors' Addresses

Erik Nordmark
Arista Networks
Santa Clara, CA
USA

Email: nordmark@acm.org

Andrew Yourtchenko
Cisco
7a de Kleetlaan
Diegem, 1831
Belgium

Phone: +32 2 704 5494
Email: ayourtch@cisco.com

Suresh Krishnan
Ericsson
8400 Decarie Blvd.
Town of Mount Royal, QC
Canada

Phone: +1 514 345 7900 x42871
Email: suresh.krishnan@ericsson.com

Network Working Group
Internet-Draft
Updates: 4191 (if approved)
Intended status: Standards Track
Expires: December 24, 2015

P. Pfister
Cisco Systems

June 22, 2015

# Source Address Dependent Route Information Option for Router Advertisements
## draft-pfister-6man-sadr-ra-01

Abstract

This document defines the Source Address Dependent Route Information option for Router Advertisements, enabling source address dependent routes to be installed in hosts by neighboring routers. It also adds a new flag to the existing Route Information option for backward compatibility purposes.

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 24, 2015.

Table of Contents



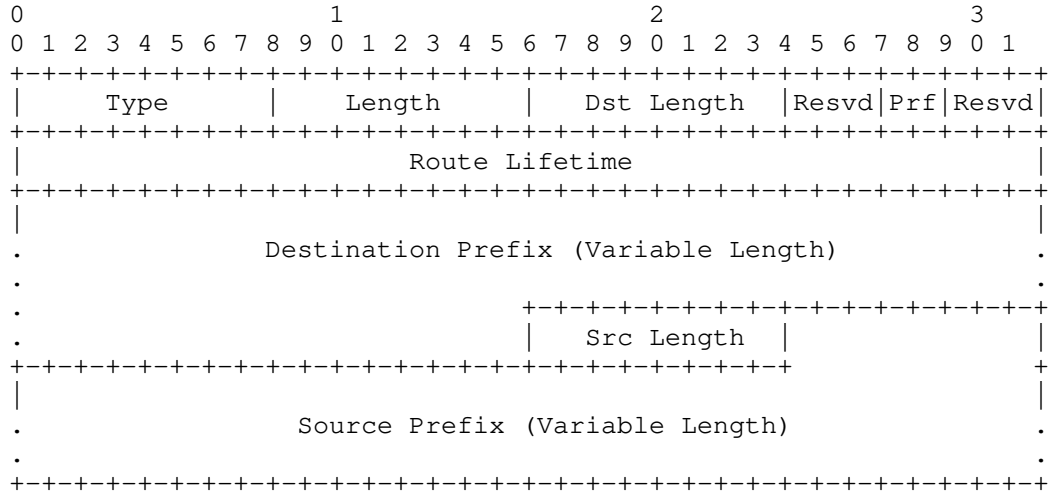
## 1. Introduction

Hosts may have multiple non-link-local addresses, possibly provided by different routers located on one or multiple links. In such situations, hosts must make sure packets with a given source address are sent to the right next-hop router. Failing in selecting the right next-hop router may, at best, induce sub-optimal routing and, at worst, cause the packet to be dropped ([RFC2827]). Rules 5 and 5.5 from the default address selection algorithm [RFC6724] make sure that, once the next-hop is chosen, care is taken to pick the right source address. Nevertheless, these rules may fail in some situations, e.g., when the same prefix is advertised on the same link by different routers. Additionally, they don't handle situations where the application picks the source-address before sending the packet.

This document defines the Source Address Dependent Route Information Option for Router Advertisements [RFC4861], enabling source address dependent routes to be installed in hosts by neighboring routers. It also adds a new flag to the Route Information Option meaning that the option may be ignored by hosts implementing this specification.

## 2. Source Address Dependent Route Information Option

This section defines a new Router Advertisement option called the Source Address Dependent Route Information option. Its use is similar to the Route Information option defined in [RFC4191] but also includes additional source prefix fields, allowing source address dependent routes to be installed on hosts receiving the Router Advertisement.




Source Address Dependent Route Information Option

Type: To be defined by IANA.

Length: The length of the option (including the Type and Length fields) in units of 8 octets. It ranges from 2 to 6.

Dst Length: The number of significant bits in the Destination Prefix field.

Resvd (Reserved): Bits reserved for futur use. They MUST be set to zero by the sender and ignored by the receiver.

Prf (Route Preference): The route preference as specified in [RFC4191]. When the Reserved value (10) is received, the option MUST be ignored.

Route Lifetime: Time in seconds (relative to the time the packet is sent) that the prefix is valid for route determination. A value of all one bits (0xffffffff) represents infinity.

Destination Prefix: The destination prefix significant bits padded to the next 8-bits boundary.

Src Length: The number of significant bits in the Source Prefix field.

Source Prefix: The source prefix significant bits padded to the next 64-bits boundary.

The following C code is given as an help for implementation:

```
#define ALIGN(bitlength, alignment) \
    (((bitlength != 0)?(((bitlength - 1) / alignment) + 1):0) * \
          (alignment / 8))

unsigned char *option;
size_t src_len_index = 8 + ALIGN(option[2], 8);
size_t total_byte_length = ALIGN((src_len_index + 1) * 8
                           + option[src_len_index], 64);
```

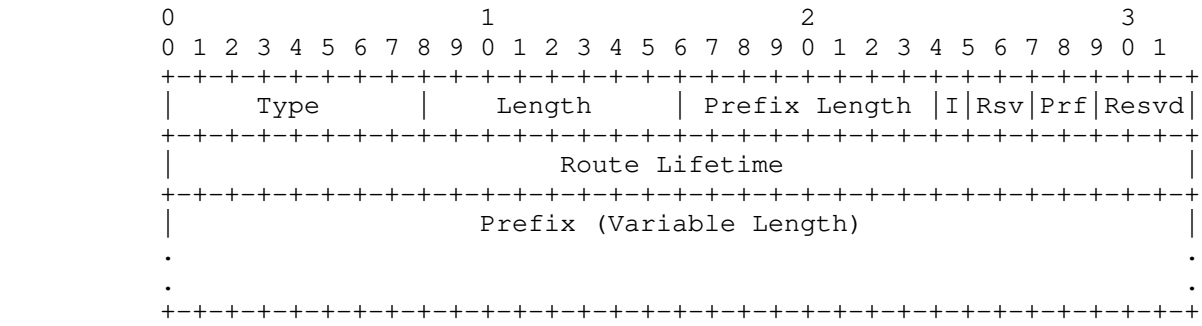
Note: Comments have been made regarding address alignment. There is no format providing at the same time good alignment and optimal TLV size, while aligning both source and destination prefixes would waste from 7 to 21 bytes per option. This TLV format is proposed based on implementation experience and provides both TLV size efficiency, and relative compatibility with the Route Information option (Linux implementation of this option support is less than 100 lines of code).

Comments and propositions are welcome regarding which format to adopt.

## 3. Route Information Option ignore flag

This document adds the Ignore flag to the Route Information option specified in [RFC4191]. It is used in order to configure type C hosts with more specific routes which will be ignored by hosts implementing this specification. Most of the time, such options with the I bit set will be used in conjunction with Source Address Dependent Route Information options including the same or a similar destination prefix.

The option is re-defined with an additional flag.




Route Information Option

I flag: Ignore flag. When this flag is set, the option MUST be ignored.

Other fields: No changes (see [RFC4191]).

## 4. Host Behavior

Hosts implementing this specification are referred to as type D hosts, in reference to host types A, B and C defined in [RFC4191]. As a reminder, type A hosts are hosts behaving as specified in [RFC4191]. Type B hosts behave similarly to type A hosts with the addition that they act upon the Default Router Preference values present in Router Advertisement headers. Finally, type C hosts behave as type B hosts with the addition that they act upon received Route Information Options.

This section specifies type D hosts behavior. Type D hosts MUST behave as type C hosts unless stated otherwise in this section. For the sake of clarity, in this whole section, 'host' refers to 'type D host'.

Hosts MUST use a Routing Table with source address dependent entries. Such entries have a:

- Source prefix
- Destination prefix
- Preference value
- Interface

- Next-hop router address
- Lifetime and associated timer

### 4.1. Selecting the next-hop router

When sending a packet, hosts MUST select the next-hop router based on the usual source address dependent routing algorithm, i.e., by picking the matching entry with, by order of precedence:

The longest destination address match.

The longest source address match.

The greatest route preference value.

In case of a tie, hosts MAY either pick one entry or use load-sharing techniques.

### 4.2. Receiving Source Address Dependent Route Information option

When receiving a Source Address Dependent Route Information option, a host MUST look for an existing routing entry with:

1. The same source prefix.
2. The same destination prefix.
3. The next-hop router address equal to the source address of the received Router Advertisement.
4. The outgoing interface equal to the interface the Router Advertisement is received on.

If no routing entry is found and the Route Lifetime is not null, insert a routing entry with the given source prefix, destination prefix, route preference, having as next-hop the source address of the received Router Advertisement, on the interface receiving the packet. If the Route Lifetime is not infinity, set the routing entry timer to the Route Lifetime value.

If a routing entry is found and the Route Lifetime is not null, cancel the associated timer. If the Route Lifetime is not infinity, set the timer to the Route Lifetime value. Finally, update the entry preference with the Route Preference value.

If a routing entry is found and the Route Lifetime is null, remove the routing entry.

If both destination and source prefixes specified by the option are ::/0, the router preference and route lifetime present in the option overrides the default router lifetime and default router preference present in the header of the Router Advertisement.

### 4.3. Receiving Route Information options

When receiving a Route Information option, a host MUST behave as follows:

If the I bit is set, ignore the option.

Otherwise, act as when receiving a Source Address Dependent Route Information option with source prefix length set to zero.

## 5. Router Behavior

Routers MAY send one or multiple Source Address Dependent Route Information options in their Router Advertisements.

Routers MUST NOT send multiple Route Information options with the same Prefix (no matter what the Ignore flag value is) or multiple Source Address Dependent Route Information options with the same Source and Destination Prefixes. Additionally, routers MUST NOT send a Route Information option with the Ignore bit not set and a Source Address Dependent Route Information with the source length equal to zero if the Prefix from the Route Information option is equal to the Destination Prefix from the Source Address Dependent Route Information option.

The Ignore bit is used to configure type D hosts differently from hosts of types A, B or C. Different combinations will result in different behaviors. For instance:

When injecting a source address dependent route is desired, a Source Address Dependent Route Information option is sent in every RA. Depending on the context, a Route Information with the same prefix and the Ignore bit set MAY be sent as well in order to inject a non source address dependent route into type C hosts. Obviously, Source Address Dependent Route Information options can be used to inject non-source dependent routes as well. This technique and the use of the Ignore bit allow type C hosts and type D hosts to be configured with possibly independent routes.

When injecting a non source address dependent route is desired, the router MAY either use a Route Information option with the Ignore flag not set, in which case both type C and D hosts will be configured, or use a Source Address Dependent Route Information

option with a source prefix ::/0, in which case type C hosts will not be configured.

When a Source Address Dependent Route Information option is removed from the set of advertised options, or when the interface ceases to be an advertising interface, the router SHOULD send up to MAX_INITIAL_RTR_ADVERTISEMENTS unsolicited Router Advertisements, using the same rule as in [RFC2461], with the Route Lifetime set to zero in all Source Address Dependent Route Information options that have become invalid.

## 6. Security Considerations

This document allows routers to configure neighboring hosts with source address dependent routing entries. Based on [RFC4191], attackers can inject default routes to type A and B hosts as well as destination address dependent routes to type C hosts. The Source Address Dependent Route Information option adds the ability for attackers to inject even more specific routes, making attacks slightly harder to detect.

## 7. IANA Considerations

IANA is kindly asked to reserve a Router Advertisement option type to be used by the Source Address Dependent Route Information option.

## 8. Acknowledgments

The author would appreciate reviews and comments.

## 9. References

### 9.1. Normative References

[RFC2461] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

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

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

### 9.2. Informative References

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

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

## Author's Address

Pierre Pfister
Cisco Systems
Paris
France

Email: pierre.pfister@darou.fr

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 4, 2016

S. Previdi, Ed.
C. Filsfils
Cisco Systems, Inc.
B. Field
Comcast
I. Leung
Rogers Communications
J. Linkova
Google
E. Aries
Facebook
T. Kosugi
NTT
E. Vyncke
Cisco Systems, Inc.
D. Lebrun
Universite Catholique de Louvain

October 2, 2015

# IPv6 Segment Routing Header (SRH)
# draft-previdi-6man-segment-routing-header-08

## Abstract

Segment Routing (SR) allows a node to steer a packet through a controlled set of instructions, called segments, by prepending a SR header to the packet. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any path (topological, or application/service based) while maintaining per-flow state only at the ingress node to the SR domain.

Segment Routing can be applied to the IPv6 data plane with the addition of a new type of Routing Extension Header. This draft describes the Segment Routing Extension Header Type and how it is used by SR capable nodes.

## 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.

This Internet-Draft will expire on April 4, 2016.



Table of Contents

## 1. Segment Routing Documents

Segment Routing terminology is defined in [I-D.ietf-spring-segment-routing].

Segment Routing use cases are described in [I-D.ietf-spring-problem-statement] and [I-D.ietf-spring-ipv6-use-cases].

Segment Routing protocol extensions are defined in [I-D.ietf-isis-segment-routing-extensions], and [I-D.ietf-ospf-ospfv3-segment-routing-extensions].

## 2. Introduction

Segment Routing (SR), defined in [I-D.ietf-spring-segment-routing], allows a node to steer a packet through a controlled set of instructions, called segments, by prepending a SR header to the packet. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any path (topological or service/application based) while maintaining per-flow state only at the ingress node to the SR domain. Segments can be derived from different components: IGP, BGP, Services, Contexts,

Locators, etc. The list of segment forming the path is called the Segment List and is encoded in the packet header.

SR allows the use of strict and loose source based routing paradigms without requiring any additional signaling protocols in the infrastructure hence delivering an excellent scalability property.

The source based routing model described in [I-D.ietf-spring-segment-routing] is inherited from the ones proposed by [RFC1940] and [RFC2460]. The source based routing model offers the support for explicit routing capability.

## 2.1. Data Planes supporting Segment Routing

Segment Routing (SR), can be instantiated over MPLS ([I-D.ietf-spring-segment-routing-mpls]) and IPv6. This document defines its instantiation over the IPv6 data-plane based on the use-cases defined in [I-D.ietf-spring-ipv6-use-cases].

This document defines a new type of Routing Header (originally defined in [RFC2460]) called the Segment Routing Header (SRH) in order to convey the Segment List in the packet header as defined in [I-D.ietf-spring-segment-routing]. Mechanisms through which segment are known and advertised are outside the scope of this document.

A segment is materialized by an IPv6 address. A segment identifies a topological instruction or a service instruction. A segment can be either:

- global: a global segment represents an instruction supported by all nodes in the SR domain and it is instantiated through an IPv6 address globally known in the SR domain.

- local: a local segment represents an instruction supported only by the node who originates it and it is instantiated through an IPv6 address that is known only by the local node.

## 2.2. Segment Routing (SR) Domain

We define the concept of the Segment Routing Domain (SR Domain) as the set of nodes participating into the source based routing model. These nodes may be connected to the same physical infrastructure (e.g.: a Service Provider's network) as well as nodes remotely connected to each other (e.g.: an enterprise VPN or an overlay).

A non-exhaustive list of examples of SR Domains is:

- The network of an operator, service provider, content provider, enterprise including nodes, links and Autonomous Systems.

- A set of nodes connected as an overlay over one or more transit providers. The overlay nodes exchange SR-enabled traffic with segments belonging solely to the overlay routers (the SR domain). None of the segments in the SR-enabled packets exchanged by the overlay belong to the transit networks

The source based routing model through its instantiation of the Segment Routing Header (SRH) defined in this document equally applies to all the above examples.

While the source routing model defined in [RFC2460] doesn't mandate which node is allowed to insert (or modify) the SRH, it is assumed in this document that the SRH is inserted in the packet by its source. For example:

- At the node originating the packet (host, server).

- At the ingress node of a SR domain where the ingress node receives an IPv6 packet and encapsulates it into an outer IPv6 header followed by a Segment Routing header.

### 2.2.1. SR Domain in a Service Provider Network

The following figure illustrates an SR domain consisting of an operator's network infrastructure.

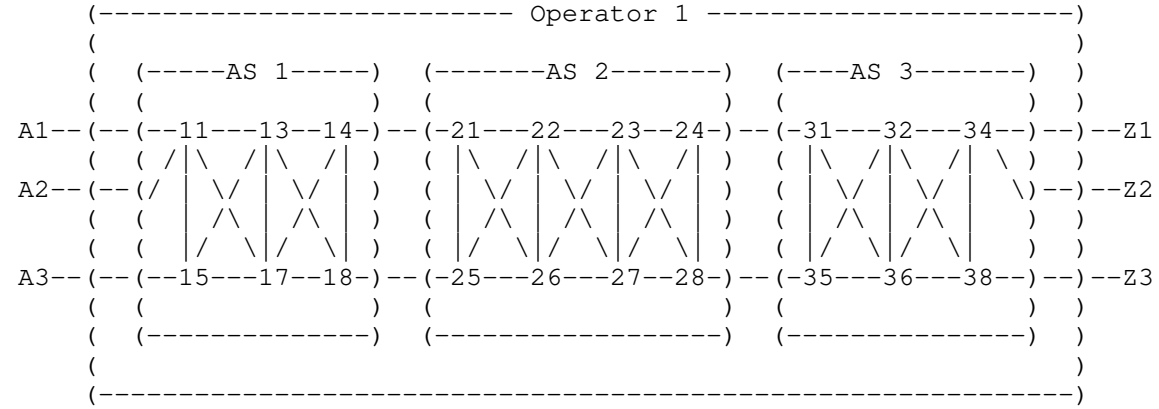



Figure 1: Service Provider SR Domain

Figure 1 describes an operator network including several ASes and delivering connectivity between endpoints. In this scenario, Segment

Routing is used within the operator networks and across the ASes boundaries (all being under the control of the same operator). In this case segment routing can be used in order to address use cases such as end-to-end traffic engineering, fast re-route, egress peer engineering, data-center traffic engineering as described in [I-D.ietf-spring-problem-statement], [I-D.ietf-spring-ipv6-use-cases] and [I-D.ietf-spring-resiliency-use-cases].

Typically, an IPv6 packet received at ingress (i.e.: from outside the SR domain), is classified according to network operator policies and such classification results into an outer header with an SRH applied to the incoming packet. The SRH contains the list of segment representing the path the packet must take inside the SR domain. Thus, the SA of the packet is the ingress node, the DA (due to SRH procedures described in Section 4) is set as the first segment of the path and the last segment of the path is the egress node of the SR domain.

The path may include intra-AS as well as inter-AS segments. It has to be noted that all nodes within the SR domain are under control of the same administration. When the packet reaches the egress point of the SR domain, the outer header and its SRH are removed so that the destination of the packet is unaware of the SR domain the packet has traversed.

The outer header with the SRH is no different from any other tunneling encapsulation mechanism and allows a network operator to implement traffic engineering mechanisms so to efficiently steer traffic across his infrastructure.

#### 2.2.2. SR Domain in a Overlay Network

The following figure illustrates an SR domain consisting of an overlay network over multiple operator's networks.

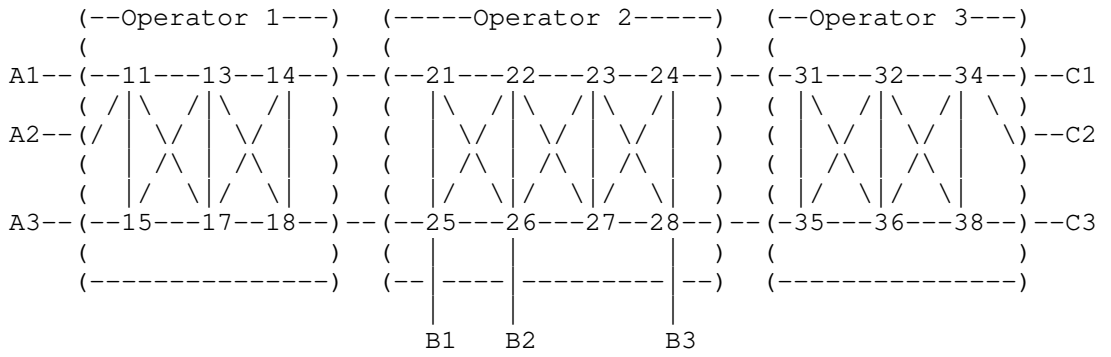



Figure 2: Overlay SR Domain

Figure 2 describes an overlay consisting of nodes connected to three different network operators and forming a single overlay network where Segment routing packets are exchanged.

The overlay consists of nodes A1, A2, A3, B1, B2, B3, C1, C2 and C3. These nodes are connected to their respective network operator and form an overlay network.

Each node may originate packets with an SRH which contains, in the segment list of the SRH or in the DA, segments identifying other overlay nodes. This implies that packets with an SRH may traverse operator's networks but, obviously, these SRHs cannot contain an address/segment of the transit operators 1, 2 and 3. The SRH originated by the overlay can only contain address/segment under the administration of the overlay (e.g. address/segments supported by A1, A2, A3, B1, B2, B3, C1,C2 or C3).

In this model, the operator network nodes are transit nodes and, according to [RFC2460], MUST NOT inspect the routing extension header since there are not the DA of the packet.

It is a common practice in operators networks to filter out, at ingress, any packet whose DA is the address of an internal node and it is also possible that an operator would filter out any packet destined to an internal address and having an extension header in it.

This common practice does not impact the SR-enabled traffic between the overlay nodes as the intermediate transit networks do never see a destination address belonging to their infrastructure. These SR-enabled overlay packets will thus never be filtered by the transit operators.

In all cases, transit packets (i.e.: packets whose DA is outside the domain of the operator's network) will be forwarded accordingly without introducing any security concern in the operator's network. This is similar to tunneled packets.

## 2.3. Illustration

In the context of Figure 3 we illustrate an example of how segment routing an be used within a SR domain in order to engineer traffic. Let's assume that the SR domain is configured as a single AS and the IGP (OSPF or IS-IS) is configured using the same cost on every link. Let's also assume that a packet P enters the SR domain at an ingress edge router I and that the operator requests the following requirements for packet P:

- The local service S offered by node B must be applied to packet P.
- The links AB and CE cannot be used to transport the packet P.
- Any node N along the journey of the packet should be able to determine where the packet P entered the SR domain and where it will exit. The intermediate node should be able to determine the paths from the ingress edge router to itself, and from itself to the egress edge router.
- Per-flow State for packet P should only be created at the ingress edge router.
- The operator can forbid, for security reasons, anyone outside the operator domain to exploit its intra-domain SR capabilities.

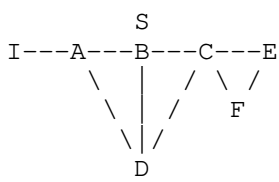



Figure 3: An illustration of SR properties

All these properties may be realized by instructing the ingress SR edge router I to create a SRH with the list of segments the packet must traverse: D, B, S, F, E. Therefore, the ingress router I creates an outer header where:

- the SA is the IPv6 address of I

- the final destination of the packet is the SR egress node E however, D being the first segment of the path, the DA is set to D IPv6 address.

- the SRH is inserted with the segment list consisting of following IPv6 addresses: D, B, S, F, E

The SRH contains a source route encoded as a list of segments (D, B, S, F, E). The ingress and egress nodes are identified in the packet respectively by the SA and the last segment of the segment list.

The packet P reaches the ingress SR node I. Node I pushes the newly created outer header and SRH with the Segment List as illustrated above (D, B, S, F, E)

D is the IPv6 address of node D and it is recognized by all nodes in the SR domain as the forwarding instruction "forward to D according to D route in the IPv6 routing table". The routing table being built through IGPs (OSPF or IS-IS) it is equivalent to say "forward according to shortest path to D".

Once at D, the next segment is inspected and executed (segment B).

B is an instruction recognized by all the nodes in the SR domain which causes the packet to be forwarded along the shortest path to B.

Once at B, the next segment is executed (segment S).

S is an instruction only recognized by node B which causes the packet to receive service S.

Once the service S is applied, the next segment is executed (segment F) which causes the packet to be forwarded along the shortest path to F.

Once at F, the next segment is executed (segment E).

E is an instruction recognized by all the nodes in the SR domain which causes the packet to be forwarded along the shortest path to E.

E being the destination of the packet, removes the outer header and the SRH. Then, it inspects the inner packet header and forwards the packet accordingly.

All of the requirements are met:

- First, the packet P has not used links AB and CE: the shortest-path from I to D is I-A-D, the shortest-path from D to B is D-B,

the shortest-path from B to F is B-C-F and the shortest-path from F to E is F-E, hence the packet path through the SR domain is I-A-D-B-C-F-E and the links AB and CE have been avoided.

- Second, the service S supported by B has been applied on packet P.

- Third, any node along the packet path is able to identify the service and topological journey of the packet within the SR domain by inspecting the SRH and SA/DA fields of the packet header.

- Fourth, only node I maintains per-flow state for packet P. The entire program of topological and service instructions to be executed by the SR domain on packet P is encoded by the ingress edge router I in the SR header in the form of a list of segments where each segment identifies a specific instruction. No further per-flow state is required along the packet path. Intermediate nodes only hold states related to the global node segments and their local segments. These segments are not per-flow specific and hence scale very well. Typically, an intermediate node would maintain in the order of 100's to 1000's global node segments and in the order of 10's to 100 of local segments.

- Fifth, the SR header (and its outer header) is inserted at the entrance to the domain and removed at the exit of the operator domain. For security reasons, the operator can forbid anyone outside its domain to use its intra-domain SR capability (e.g. configuring ACL that deny any packet with a DA towards its infrastructure segment).

# 3. IPv6 Instantiation of Segment Routing

## 3.1. Segment Identifiers (SIDs)

Segment Routing, as described in [I-D.ietf-spring-segment-routing], defines Node-SID and Adjacency-SID. When SR is used over IPv6 data-plane the following applies.

### 3.1.1. Node-SID

The Node-SID identifies a node. With SR-IPv6 the Node-SID is an IPv6 address that the operator configured on the node and that is used as the node identifier. Typically, in case of a router, this is the IPv6 address of the node loopback interface. Therefore, SR-IPv6 does not require any additional SID advertisement for the Node Segment. The Node-SID is in fact the IPv6 address of the node.

### 3.1.2. Adjacency-SID

Adjacency-SIDs can be either globally scoped IPv6 addresses or IPv6 addresses known locally by the node but not advertised in any control plane (in other words an Adjacency-SID may well be any 128-bit identifier). Obviously, in the latter case, the scope of the Adjacency-SID is local to the router and any packet with the a such Adjacency-SID would need first to reach the node through the node's Segment Identifier (i.e.: Node-SID) prior for the node to process the Adjacency-SID. In other words, two segments (SIDs) would then be required: the first is the node's Node-SID that brings the packet to the node and the second is the Adjacency-SID that will make the node to forward the packet through the interface the Adjacency-SID is allocated to.

In the SR architecture defined in [I-D.ietf-spring-segment-routing] a node may advertise one (or more) Adj-SIDs allocated to the same interface as well as a node can advertise the same Adj-SID for multiple interfaces. Use cases of Adj-SID advertisements are described in [I-D.ietf-spring-segment-routing]The semantic of the Adj-SID is:

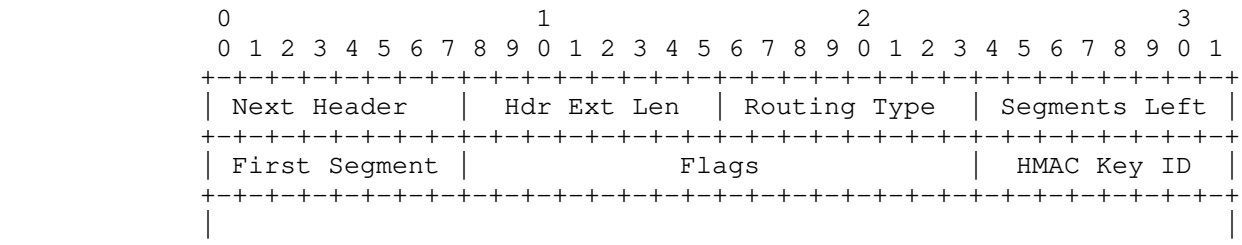
> Send out the packet to the interface this Adj-SID is allocated to.

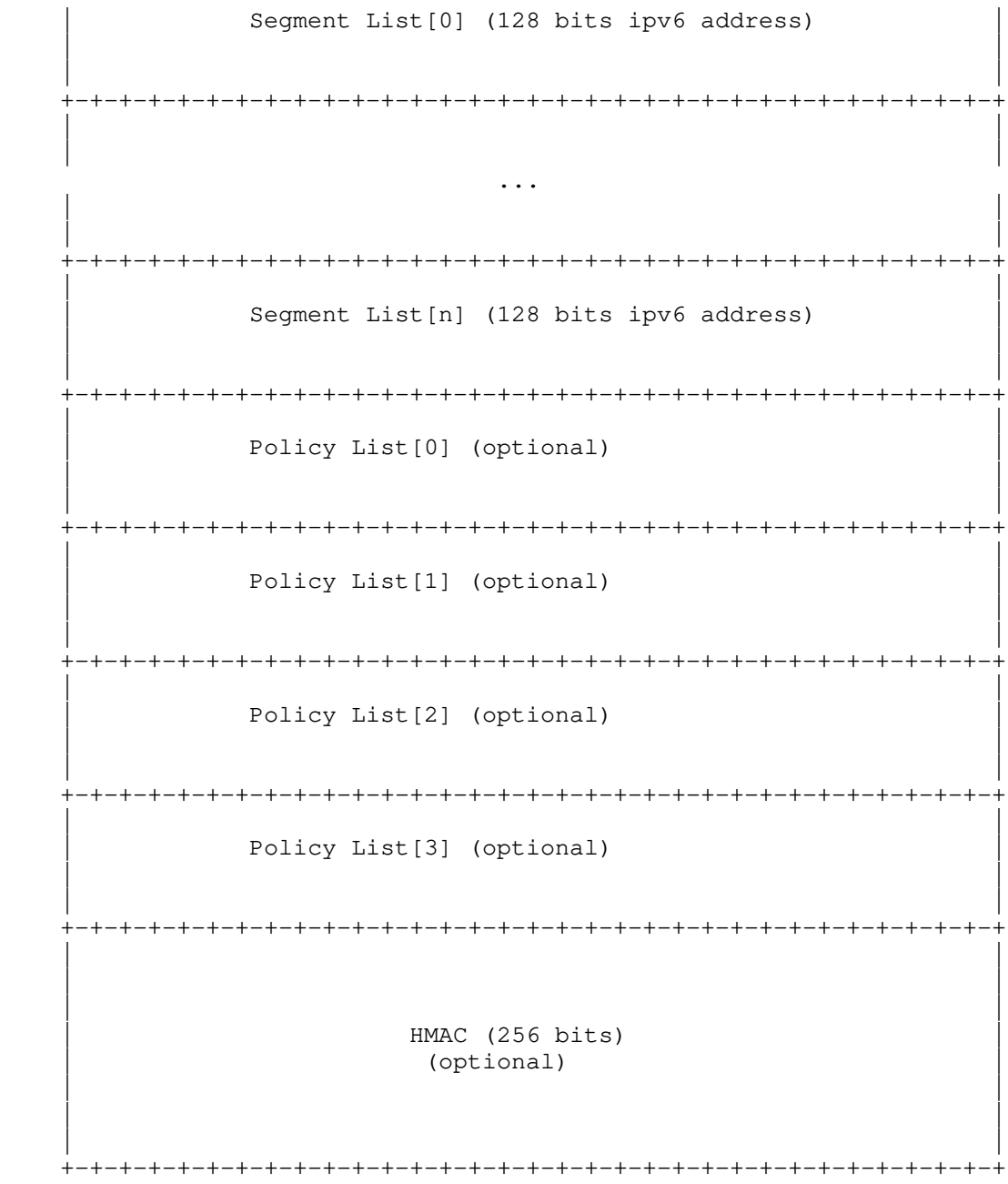
Advertisement of Adj-SID may be done using multiple mechanisms among which the ones described in ISIS and OSPF protocol extensions: [I-D.ietf-isis-segment-routing-extensions] and [I-D.ietf-ospf-ospfv3-segment-routing-extensions]. The distinction between local and global significance of the Adj-SID is given in the encoding of the Adj-SID advertisement.

## 3.2. Segment Routing Extension Header (SRH)

A new type of the Routing Header (originally defined in [RFC2460]) is defined: the Segment Routing Header (SRH) which has a new Routing Type, (suggested value 4) to be assigned by IANA.

The Segment Routing Header (SRH) is defined as follows:




Segment List[0] (128 bits ipv6 address)
...
Segment List[n] (128 bits ipv6 address)
Policy List[0] (optional)
Policy List[1] (optional)
Policy List[2] (optional)
Policy List[3] (optional)
HMAC (256 bits)
(optional)

where:

- Next Header: 8-bit selector. Identifies the type of header immediately following the SRH.

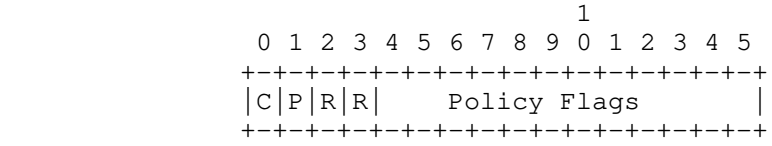
- Hdr Ext Len: 8-bit unsigned integer, is the length of the SRH header in 8-octet units, not including the first 8 octets.

- Routing Type: TBD, to be assigned by IANA (suggested value: 4).

- Segments Left. Defined in [RFC2460], it contains the index, in the Segment List, of the next segment to inspect. Segments Left is decremented at each segment.

- First Segment: contains the index, in the Segment List, of the first segment of the path which is in fact the last element of the Segment List.

- Flags: 16 bits of flags. Following flags are defined:




C-flag: Clean-up flag. Set when the SRH has to be removed from the packet when packet reaches the last segment.

P-flag: Protected flag. Set when the packet has been rerouted through FRR mechanism by a SR endpoint node.

R-flags. Reserved and for future use.

Policy Flags. Define the type of the IPv6 addresses encoded into the Policy List (see below). The following have been defined:

Bits 4-6: determine the type of the first element after the segment list.

Bits 7-9: determine the type of the second element.

Bits 10-12: determine the type of the third element.

Bits 13-15: determine the type of the fourth element.

The following values are used for the type:

0x0: Not present. If value is set to 0x0, it means the element represented by these bits is not present.

0x1: SR Ingress.

0x2: SR Egress.

0x3: Original Source Address.

0x4 to 0x7: currently unused and SHOULD be ignored on reception.

- HMAC Key ID and HMAC field, and their use are defined in Section 5.

- Segment List[n]: 128 bit IPv6 addresses representing the nth segment in the Segment List. The Segment List is encoded starting from the last segment of the path. I.e., the first element of the segment list (Segment List [0]) contains the last segment of the path while the last segment of the Segment List (Segment List[n]) contains the first segment of the path. The index contained in "Segments Left" identifies the current active segment.

- Policy List. Optional addresses representing specific nodes in the SR path such as:

  SR Ingress: a 128 bit generic identifier representing the ingress in the SR domain (i.e.: it needs not to be a valid IPv6 address).

  SR Egress: a 128 bit generic identifier representing the egress in the SR domain (i.e.: it needs not to be a valid IPv6 address).

  Original Source Address: IPv6 address originally present in the SA field of the packet.

  The segments in the Policy List are encoded after the segment list and they are optional. If none are in the SRH, all bits of the Policy List Flags MUST be set to 0x0.

#### 3.2.1. SRH and RFC2460 behavior

The SRH being a new type of the Routing Header, it also has the same properties:

SHOULD only appear once in the packet.

Only the router whose address is in the DA field of the packet header MUST inspect the SRH.

Therefore, Segment Routing in IPv6 networks implies that the segment identifier (i.e.: the IPv6 address of the segment) is moved into the DA of the packet.

The DA of the packet changes at each segment termination/completion and therefore the original DA of the packet MUST be encoded as the last segment of the path.

As illustrated in Section 2.3, nodes that are within the path of a segment will forward packets based on the DA of the packet without inspecting the SRH. This ensures full interoperability between SR-capable and non-SR-capable nodes.

# 4. SRH Procedures

In this section we describe the different procedures on the SRH.

## 4.1. Segment Routing Node Functions

SR packets are forwarded to segments endpoints (i.e.: the segment endpoint is the node representing the segment and whose address is in the segment list and in the DA of the packet when traveling in the segment). The segment endpoint, when receiving a SR packet destined to itself, does:

- Inspect the SRH.
- Determine the next active segment.
- Update the Segments Left field (or, if requested, remove the SRH from the packet).
- Update the DA.
- Forward the packet to the next segment.

The procedures applied to the SRH are related to the node function. Following nodes functions are defined:

Source SR Node.

SR Domain Ingress Node.

Transit Node.

SR Endpoint Node.

#### 4.1.1. Source SR Node

A Source SR Node can be any node originating an IPv6 packet with its IPv6 and Segment Routing Headers. This include either:

A host originating an IPv6 packet

A SR domain ingress router encapsulating a received IPv6 packet into an outer IPv6 header followed by a SRH

The mechanism through which a Segment List is derived is outside of the scope of this document. As an example, the Segment List may be obtained through:

Local path computation.

Local configuration.

Interaction with a centralized controller delivering the path.

Any other mechanism.

The following are the steps of the creation of the SRH:

Next Header and Hdr Ext Len fields are set according to [RFC2460].

Routing Type field is set as TBD (SRH).

The Segment List is built with the FIRST segment of the path encoded in the LAST element of the Segment List. Subsequent segments are encoded on top of the first segment. Finally, the LAST segment of the path is encoded in the FIRST element of the Segment List. In other words, the Segment List is encoded in the reverse order of the path.

The original DA of the packet is encoded as the last segment of the path (encoded in the first element of the Segment List).

The DA of the packet is set with the value of the first segment (found in the last element of the segment list).

The Segments Left field is set to n-1 where n is the number of elements in the Segment List.

The First Segment field is set to n-1 where n is the number of elements in the Segment List.

The packet is sent out towards the first segment (i.e.: represented in the packet DA).

HMAC and HMAC Key ID may be set according to Section 5.

#### 4.1.2. SR Domain Ingress Node

The SR Domain Ingress Node is the node where ingress policies are applied and where the packet path (and processing) is determined.

After policies are applied and packet classification is done, the result may be instantiated into a Segment List representing the path the packet should take. In such case, the SR Domain Ingress Node instantiate a new outer IPv6 header to which the SRH is appended (with the computed Segment List). The procedures for the creation and insertion of the new SRH are described in Section 4.1.1.

#### 4.1.3. Transit Node

According to [RFC2460], the only node who is allowed to inspect the Routing Extension Header (and therefore the SRH), is the node corresponding to the DA of the packet. Any other transit node MUST NOT inspect the underneath routing header and MUST forward the packet towards the DA and according to the IPv6 routing table.

In the example case described in Section 2.2.2, when SR capable nodes are connected through an overlay spanning multiple third-party infrastructure, it is safe to send SRH packets (i.e.: packet having a Segment Routing Header) between each other overlay/SR-capable nodes as long as the segment list does not include any of the transit provider nodes. In addition, as a generic security measure, any service provider will block any packet destined to one of its internal routers, especially if these packets have an extended header in it.

#### 4.1.4. SR Segment Endpoint Node

The SR segment endpoint node is the node whose address is in the DA. The segment endpoint node inspects the SRH and does:

```
1.   IF DA = myself (segment endpoint)
2.      IF Segments Left > 0 THEN
           decrement Segments Left
           update DA with Segment List[Segments Left]
3.         IF Segments Left == 0 THEN
              IF Clean-up bit is set THEN remove the SRH
4.      ELSE give the packet to next PID (application)
             End of processing.
5.   Forward the packet out
```

## 5. Security Considerations

This section analyzes the security threat model, the security issues and mitigation techniques of SRH.

SRH is simply another type of the routing header as described in RFC 2460 [RFC2460] and is:

- added to a new outer IP header by the ingress router when entering the SR domain or by the originating node itself. The source host can be outside the SR domain;

- inspected and acted upon when reaching the destination address of the IP header per RFC 2460 [RFC2460].

Per RFC2460 [RFC2460], routers on the path that simply forward an IPv6 packet (i.e. the IPv6 destination address is none of theirs) will never inspect and process the content of any routing header (including SRH). Routers whose one interface IPv6 address equals the destination address field of the IPv6 packet MUST to parse the SRH and, if supported and if the local configuration allows it, MUST act accordingly to the SRH content.

According to RFC2460 [RFC2460], non SR-capable (or non SR-configured) router upon receipt of an IPv6 packet with SRH destined to an address of its:

- must ignore the SRH completely if the Segment Left field is 0 and proceed to process the next header in the IPv6 packet;

- must discard the IPv6 packet if Segment Left field is greater than 0 and send a Parameter Problem ICMP message back to the Source Address.

### 5.1. Threat model

#### 5.1.1. Source routing threats

Using a SRH is a specific case of loose source routing, therefore it has some well-known security issues as described in RFC4942 [RFC4942] section 2.1.1 and RFC5095 [RFC5095]:

- amplification attacks: where a packet could be forged in such a way to cause looping among a set of SR-enabled routers causing unnecessary traffic, hence a Denial of Service (DoS) against bandwidth;

- reflection attack: where a hacker could force an intermediate node to appear as the immediate attacker, hence hiding the real attacker from naive forensic;

- bypass attack: where an intermediate node could be used as a stepping stone (for example in a De-Militarized Zone) to attack another host (for example in the datacenter or any back-end server).

#### 5.1.2. Applicability of RFC 5095 to SRH

First of all, the reader must remember this specific part of section 1 of RFC5095 [RFC5095], "A side effect is that this also eliminates benign RH0 use-cases; however, such applications may be facilitated by future Routing Header specifications.". In short, it is not forbidden to create new secure type of Routing Header; for example, RFC 6554 (RPL) [RFC6554] also creates a new Routing Header type for a specific application confined in a single network.

The main use case for SR consists of the single administrative domain (or cooperating administrative domains) where only trusted nodes with SR enabled and explicitely configured participate in SR: this is the same model as in RFC6554 [RFC6554]. All non-trusted nodes do not participate as either SR processing is not enabled by default or because they only process SRH from nodes within their domain.

Moreover, all SR routers SHOULD ignore SRH created by outsiders based on topology information (received on a peering or internal interface) or on presence and validity of the HMAC field. Therefore, if intermediate SR routers ONLY act on valid and authorized SRH (such as within a single administrative domain), then there is no security threat similar to RH-0. Hence, the RFC 5095 [RFC5095] attacks are not applicable.

#### 5.1.3. Service stealing threat

Segment routing is used for added value services, there is also a need to prevent non-participating nodes to use those services; this is called 'service stealing prevention'.

#### 5.1.4. Topology disclosure

The SRH may also contains IPv6 addresses of some intermediate SR routers in the path towards the destination, this obviously reveals those addresses to the potentially hostile attackers if those attackers are able to intercept packets containing SRH. On the other hand, if the attacker can do a traceroute whose probes will be forwarded along the SR path, then there is little learned by intercepting the SRH itself. The clean-bit of SRH can help by removing the SRH before forwarding the packet to potentially a non-trusted part of the network; if the attacker can force the generation of an ICMP message during the transit in the SR domain, then the ICMP will probably contain the SRH header (totally or partially) depending on the ICMP-generating router behavior.

#### 5.1.5. ICMP Generation

Per section 4.4 of RFC2460 [RFC2460], when destination nodes (i.e. where the destination address is one of theirs) receive a Routing Header with unsupported Routing Type, the required behavior is:

- If Segments Left is zero, the node must ignore the Routing header and proceed to process the next header in the packet.

- If Segments Left is non-zero, the node must discard the packet and SHOULD send an ICMP Parameter Problem, Code 0, message to the packet's Source Address, pointing to the unrecognized Routing Type.

This required behavior could be used by an attacker to force the generation of ICMP message by any node. The attacker could send packets with SRH (with Segment Left different than 0) destined to a node not supporting SRH. Per RFC2460 [RFC2460], the destination node must then generate an ICMP message per RFC 2460, causing a local CPU utilization and if the source of the offending packet with SRH was spoofed could lead to a reflection attack without any amplification.

It must be noted that this is a required behavior for any unsupported Routing Type and not limited to SRH packets. So, it is not specific to SRH and the usual rate limiting for ICMP generation is required anyway for any IPv6 implementation and has been implemented and deployed for many years.

### 5.2. Security fields in SRH

This section summarizes the use of specific fields in the SRH. They are based on a key-hashed message authentication code (HMAC).

The security-related fields in SRH are:

- HMAC Key-id, 8 bits wide;
- HMAC, 256 bits wide (optional, exists only if HMAC Key-id is not 0).

The HMAC field is the output of the HMAC computation (per RFC 2104 [RFC2104]) using a pre-shared key and hashing algorithm identified by HMAC Key-id and of the text which consists of the concatenation of:

- the source IPv6 address;
- First Segment field;
- an octet whose bit-0 is the clean-up bit flag and others are 0;
- HMAC Key-id;
- all addresses in the Segment List.

The purpose of the HMAC field is to verify the validity, the integrity and the authorization of the SRH itself. If an outsider of the SR domain does not have access to a current pre-shared secret, then it cannot compute the right HMAC field and the first SR router on the path processing the SRH and configured to check the validity of the HMAC will simply reject the packet.

The HMAC field is located at the end of the SRH simply because only the router on the ingress of the SR domain needs to process it, then all other SR nodes can ignore it (based on local policy) because they trust the upstream router. This is to speed up forwarding operations because SR routers which do not validate the SRH do not need to parse the SRH until the end.

The HMAC Key-id field allows for the simultaneous existence of several hash algorithms (SHA-256, SHA3-256 ... or future ones) as well as pre-shared keys. This allows for pre-shared key roll-over when two pre-shared keys are supported for a while when all SR nodes converged to a fresher pre-shared key. The HMAC Key-id field is opaque, i.e., it has neither syntax not semantic except as an index to the right combination of pre-shared key and hash algorithm and except that a value of 0 means that there is no HMAC field. It could

also allow for interoperation among different SR domains if allowed by local policy and assuming a collision-free Key Id allocation which is out of scope of this memo.

When a specific SRH is linked to a time-related service (such as turbo-QoS for a 1-hour period), then it is important to refresh the shared-secret frequently as the HMAC validity period expires only when the HMAC Key-id and its associated shared-secret expires.

#### 5.2.1. Selecting a hash algorithm

The HMAC field in the SRH is 256 bits wide. Therefore, the HMAC MUST be based on a hash function whose output is at least 256 bits. If the output of the hash function is 256, then this output is simply inserted in the HMAC field. If the output of the hash function is larger than 256 bits, then the output value is truncated to 256 by taking the least-significant 256 bits and inserting them in the HMAC field.

SRH implementations can support multiple hash functions but MUST implement SHA-2 [FIPS180-4] in its SHA-256 variant.

NOTE: SHA-1 is currently used by some early implementations used for quick interoperations testing, the 160-bit hash value must then be right-hand padded with 96 bits set to 0. The authors understand that this is not secure but is ok for limited tests.

#### 5.2.2. Performance impact of HMAC

While adding a HMAC to each and every SR packet increases the security, it has a performance impact. Nevertheless, it must be noted that:

- the HMAC field SHOULD be used only when SRH is inserted by a device (such as a home set-up box) which is outside of the segment routing domain. If the SRH is added by a router in the trusted segment routing domain, then, there is no need for a HMAC field, hence no performance impact.

- when present, the HMAC field MUST be checked and validated only by the first router of the segment routing domain, this router is named 'validating SR router'. Downstream routers may not inspect the HMAC field.

- this validating router can also have a cache of <IPv6 header + SRH, HMAC field value> to improve the performance. It is not the same use case as in IPsec where HMAC value was unique per packet, in SRH, the HMAC value is unique per flow.

- Last point, hash functions such as SHA-2 have been optmized for security and performance and there are multiple implementations with good performance.

With the above points in mind, the performance impact of using HMAC is minimized.

#### 5.2.3. Pre-shared key management

The field HMAC Key-id allows for:

- key roll-over: when there is a need to change the key (the hash pre-shared secret), then multiple pre-shared keys can be used simultaneously. The validating routing can have a table of <HMAC Key-id, pre-shared secret, hash algorithm> for the currently active and future keys.

- different algorithm: by extending the previous table to <HMAC Key-id, hash function, pre-shared secret>, the validating router can also support simultaneously several hash algorithms (see section Section 5.2.1)

The pre-shared secret distribution can be done:

- in the configuration of the validating routers, either by static configuration or any SDN oriented approach;

- dynamically using a trusted key distribution such as [RFC6407]

The intent of this document is NOT to define yet-another-key-distribution-protocol.

### 5.3. Deployment Models

#### 5.3.1. Nodes within the SR domain

The routers inside a SR domain can be trusted to generate the outer IP header and the SRH and to process SRH received on interfaces that are part of the SR domain. These nodes MUST drop all SRH packets received on any interface that is not part of the SR domain and containing a SRH whose HMAC field cannot be validated by local policies. This includes obviously packet with a SRH generated by a non-cooperative SR domain.

If the validation fails, then these packets MUST be dropped, ICMP error messages (parameter problem) SHOULD be generated (but rate limited) and SHOULD be logged.

#### 5.3.2. Nodes outside of the SR domain

Nodes outside of the SR domain cannot be trusted for physical security; hence, they need to obtain by some trusted means (outside of the scope of this document) a complete SRH for each new connection (i.e. new destination address). The received SRH MUST include a HMAC Key-id and HMAC field which has been computed correctly (see Section 5.2).

When a outside the SR domain sends a packet with a SRH and towards a SR domain ingress node, the packet MUST contain the HMAC Key-id and HMAC field and the the destination address MUST be an address of a SR domain ingress node .

The ingress SR router, i.e., the router with an interface address equals to the destination address, MUST verify the HMAC field with respect to the HMAC Key-id.

If the validation is successful, then the packet is simply forwarded as usual for a SR packet. As long as the packet travels within the SR domain, no further HMAC check needs to be done. Subsequent routers in the SR domain MAY verify the HMAC field when they process the SRH (i.e. when they are the destination).

If the validation fails, then this packet MUST be dropped, an ICMP error message (parameter problem) SHOULD be generated (but rate limited) and SHOULD be logged.

#### 5.3.3. SR path exposure

As the intermediate SR nodes addresses appears in the SRH, if this SRH is visible to an outsider then he/she could reuse this knowledge to launch an attack on the intermediate SR nodes or get some insider knowledge on the topology. This is especially applicable when the path between the source node and the first SR domain ingress router is on the public Internet.

The first remark is to state that 'security by obscurity' is never enough; in other words, the security policy of the SR domain SHOULD assume that the internal topology and addressing is known by the attacker.

IPsec Encapsulating Security Payload [RFC4303] cannot be use to protect the SRH as per RFC4303 the ESP header must appear after any routing header (including SRH).

When the SRH is not generated by the actual source node but by an SR domain ingress router, it is added after a new outer IP header, this

means that a normal traceroute will not reveal the routers in the SR domain (pretty much like in a MPLS network) and that if ICMP are generated by routers in the SR domain they will be sent to the ingress router of the SR domain without revealing anything to the outside of the SR domain.

To prevent a user to leverage the gained knowledge by intercepting SRH, it it recommended to apply an infrastructure Access Control List (iACL) at the edge of the SR domain. This iACL will drop all packets from outside the SR-domain whose destination is any address of any router inside the domain. This security policy should be tuned for local operations.

#### 5.3.4. Impact of BCP-38

BCP-38 [RFC2827], also known as "Network Ingress Filtering", checks whether the source address of packets received on an interface is valid for this interface. The use of loose source routing such as SRH forces packets to follow a path which differs from the expected routing. Therefore, if BCP-38 was implemented in all routers inside the SR domain, then SR packets could be received by an interface which is not expected one and the packets could be dropped.

As a SR domain is usually a subset of one administrative domain, and as BCP-38 is only deployed at the ingress routers of this administrative domain and as packets arriving at those ingress routers have been normally forwarded using the normal routing information, then there is no reason why this ingress router should drop the SRH packet based on BCP-38. Routers inside the domain commonly do not apply BCP-38; so, this is not a problem.

## 6. IANA Considerations

TBD but should at least require a new type for routing header

## 7. Manageability Considerations

TBD should we talk about traceroute? about SRH in ICMP replies?

## 8. Contributors

The authors would like to thank Dave Barach, John Leddy, John Brzozowski, Pierre Francois, Nagendra Kumar, Mark Townsley, Christian Martin, Roberta Maglione, James Connolly, Aloys Augustin and Fred Baker for their contribution to this document.

## 9. Acknowledgements

TBD

## 10. References

### 10.1. Normative References

[FIPS180-4] National Institute of Standards and Technology, "FIPS 180-4 Secure Hash Standard (SHS)", March 2012, <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <http://www.rfc-editor.org/info/rfc4303>.

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <http://www.rfc-editor.org/info/rfc5095>.

[RFC6407] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain of Interpretation", RFC 6407, DOI 10.17487/RFC6407, October 2011, <http://www.rfc-editor.org/info/rfc6407>.

### 10.2. Informative References

[I-D.ietf-isis-segment-routing-extensions] Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-05 (work in progress), June 2015.

[I-D.ietf-ospf-ospfv3-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPFv3 Extensions for Segment Routing", draft-ietf-ospf-ospfv3-segment-routing-extensions-03 (work in progress), June 2015.

[I-D.ietf-spring-ipv6-use-cases]
Brzozowski, J., Leddy, J., Leung, I., Previdi, S., Townsley, W., Martin, C., Filsfils, C., and R. Maglione, "IPv6 SPRING Use Cases", draft-ietf-spring-ipv6-use-cases-05 (work in progress), September 2015.

[I-D.ietf-spring-problem-statement]
Previdi, S., Filsfils, C., Decraene, B., Litkowski, S., Horneffer, M., and R. Shakir, "SPRING Problem Statement and Requirements", draft-ietf-spring-problem-statement-04 (work in progress), April 2015.

[I-D.ietf-spring-resiliency-use-cases]
Francois, P., Filsfils, C., Decraene, B., and R. Shakir, "Use-cases for Resiliency in SPRING", draft-ietf-spring-resiliency-use-cases-01 (work in progress), March 2015.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and r. rjs@rob.sh, "Segment Routing Architecture", draft-ietf-spring-segment-routing-05 (work in progress), September 2015.

[I-D.ietf-spring-segment-routing-mpls]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J., and E. Crabbe, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-01 (work in progress), May 2015.

[RFC1940] Estrin, D., Li, T., Rekhter, Y., Varadhan, K., and D. Zappala, "Source Demand Routing: Packet Format and Forwarding Specification (Version 1)", RFC 1940, DOI 10.17487/RFC1940, May 1996, <http://www.rfc-editor.org/info/rfc1940>.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <http://www.rfc-editor.org/info/rfc2104>.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <http://www.rfc-editor.org/info/rfc2827>.

[RFC4942] Davies, E., Krishnan, S., and P. Savola, "IPv6 Transition/ Co-existence Security Considerations", RFC 4942, DOI 10.17487/RFC4942, September 2007, <http://www.rfc-editor.org/info/rfc4942>.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, DOI 10.17487/RFC6554, March 2012, <http://www.rfc-editor.org/info/rfc6554>.

## Authors' Addresses

Stefano Previdi (editor)
Cisco Systems, Inc.
Via Del Serafico, 200
Rome 00142
Italy

Email: sprevidi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Brussels
BE

Email: cfilsfil@cisco.com

Brian Field
Comcast
4100 East Dry Creek Road
Centennial, CO 80122
US

Email: Brian_Field@cable.comcast.com

Ida Leung
Rogers Communications
8200 Dixie Road
Brampton, ON L6T 0C1
CA

Email: Ida.Leung@rci.rogers.com

Jen Linkova
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: furry@google.com

Ebben Aries
Facebook
US

Email: exa@fb.com

Tomoya Kosugi
NTT
3-9-11, Midori-Cho Musashino-Shi,
Tokyo 180-8585
JP

Email: kosugi.tomoya@lab.ntt.co.jp

Eric Vyncke
Cisco Systems, Inc.
De Kleetlaann 6A
Diegem 1831
Belgium

Email: evyncke@cisco.com

David Lebrun
Universite Catholique de Louvain
Place Ste Barbe, 2
Louvain-la-Neuve, 1348
Belgium

Email: david.lebrun@uclouvain.be

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 24, 2015

B. Sarikaya
Huawei USA

February 20, 2015

# Source Address Dependent Routing and Source Address Selection for IPv6 Hosts
## draft-sarikaya-6man-sadr-overview-05

Abstract

This document presents the source address dependent routing from the host perspective. Multihomed hosts and hosts with multiple interfaces are considered. Different architectures are introduced and with their help, why source address selection and next hop resolution in view of source address dependent routing is needed is explained. The document concludes with an informative guidelines on the different solution approaches.

Table of Contents



## 1. Introduction

BCP 38 recommends ingress traffic routing to prohibit Denial of Service (DoS) attacks, i.e. datagrams which have source addresses that do not match with the network where the host is attached are discarded [RFC2827]. Avoiding packets to be dropped because of ingress filtering is difficult especially in multihomed networks where the host receives more than one prefix from the connected Internet Service Providers (ISP) and may have more than one source addresses. Based on BCP 38, BCP 84 introduced recommendations on the routing system for multihomed networks [RFC3704].

Recommendations on the routing system for ingress filtering such as in BCP 84 inevitably involve source address checks. This leads us to the source address dependent routing. Source address dependent routing is an issue especially when the host is connected to a multihomed network and is communicating with another host in another multihomed network. In such a case, the communication can be broken in both directions if ISPs apply ingress filtering and the datagrams contain wrong source addresses [I-D.huitema-multi6-ingress-filtering].

Hosts with simultaneously active interfaces receive multiple prefixes and have multiple source addresses. Datagrams originating from such hosts carry greats risks to be dropped due to ingress filtering. Source address selection algorithm needs to be careful to try to avoid ingress filtering on the next-hop router [RFC6724].

Many use cases have been reported for source/destination routing in [I-D.baker-rtgwg-src-dst-routing-use-cases]. These use cases clearly indicate that the multihomed host or Customer Premises Equipment (CPE) router needs to be configured with correct source prefixes/ addresses so that it can route packets upstream correctly to avoid ingress filtering applied by an upstream ISP to drop the packets.

In multihomed networks there is a need to do source address based routing if some providers are performing the ingress filtering defined in BCP38 [RFC2827]. This requires the routers to consider the source addresses as well as the destination addresses in determining the next hop to send the packet to.

Based on the use cases defined in [I-D.baker-rtgwg-src-dst-routing-use-cases], the routers may be informed about the source addresses to use in routing using extensions to the routing protocols like IS-IS defined in [ISO.10589.1992] [I-D.baker-ipv6-isis-dst-src-routing] and OSPF defined in [RFC5340] [I-D.baker-ipv6-ospf-dst-src-routing]. In this document we describe the use cases for source address dependent routing from the host perspective.

There are two cases. A host may have a single interface with multiple addresses (from different prefixes or /64s). Each address or prefix is connected to or coming from different exit routers, and this case can be called multi-prefix multihoming (MPMH). A host may have simultaneously connected multiple interfaces where each interface is connected to a different exit router and this case can be called multi-prefix multiple interface (MPMI).

It should be noted that Network Address and Port Translation (NAPT) [RFC3022] in IPv4 and IPv6-to-IPv6 Network Prefix Translation (NPTv6) [RFC6296] in IPv6 implement the functions of source address selection and next-hop resolution and as such they address multihoming (and hosts with multiple interfaces) requirements arising from source address dependent routing [RFC7157]. In this case, the gateway router or CPE router does the source address and next hop selection for all the hosts connected to the router. However, for end-to-end connectivity, NAPT and NPTv6 should be avoided and because of this, NAPT and NPTv6 are left out of scope in this document.

## 2. Terminology

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. SADR Scenarios

Source address dependent routing can be facilitated at the host with proper next hop and source address selection. For this, each router connected to different interfaces of the host uses Router Advertisements to distribute default route, next hop as well as source address/prefix information to the host.

The use case shown in Figure 1 is multi-prefix multi interface use case where rtr1 and rtr2 represent customer premises equipment/ routers (CPE) and there are exit routers in both network 1 and network 2. The issue in this case is ingress filtering. If the packets from the host communicating with a remote destination are routed to the wrong exit router, i.e. carry wrong source address, they will get dropped.

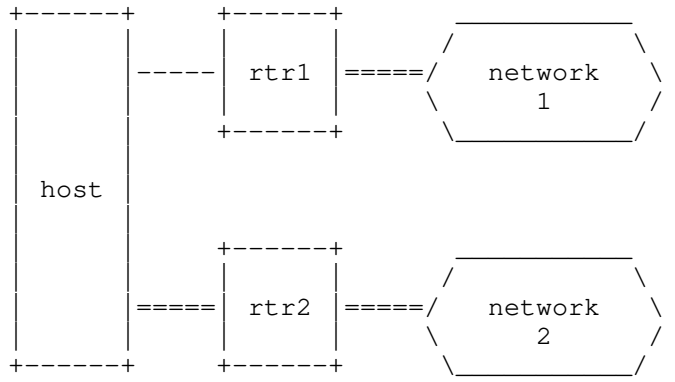



Figure 1: multiple Interfaced Host with Two CPE Routers

Our next use case is shown in Figure 2. This use case is a multi-prefix multihoming use case. rtr is CPE router which is connected to two ISPs each advertising their own prefixes. In this case, the host may have a single interface but it receives multiple prefixes from the connected ISPs. Assuming that ISPs apply ingress filtering policy the packets for any external communication from the host should follow source address dependent routing in order to avoid getting dropped.

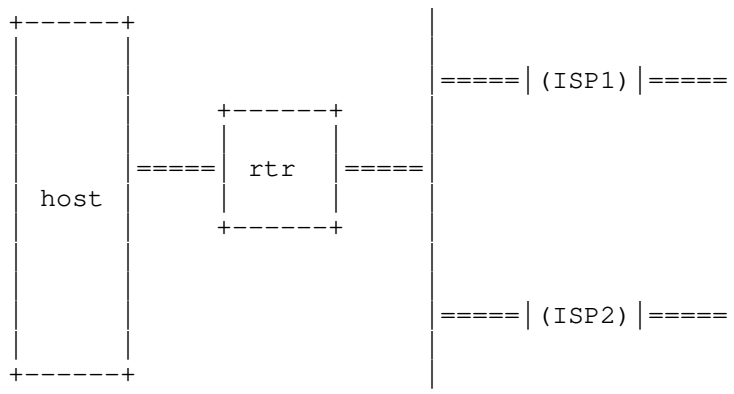



Figure 2: Multihomed Host with Multiple CPE Routers

A variation of this use case is specialized egress routing. Upstream networks offer different services with specific requirements, e.g. video service. The hosts using this service need to use the service's source and destination addresses. No other service will accept this source address, i.e. those packets will be dropped [I-D.baker-rtgwg-src-dst-routing-use-cases].

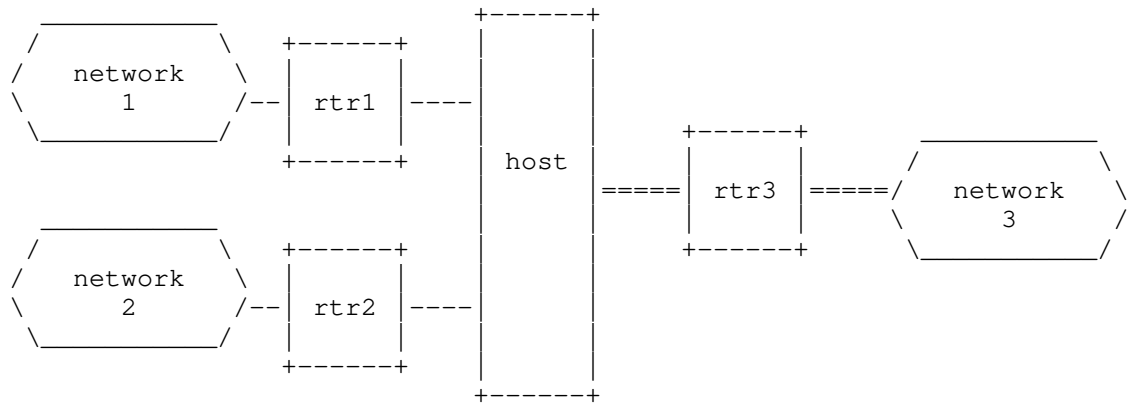



Figure 3: multiple Interfaced Host with Three CPE Routers

Next use case is shown in Figure 3. It is a variation of multi-prefix multi interface use case above. rtr1, rtr2 and rtr3 are CPE Routers. The networks apply ingress routing. Source address dependent routing should be used to avoid any external communications be dropped.

In the homenet scenario given in Figure 4, representing a simple home network, there is a host connected to two CPEs which are connected to ISP1 and ISP2, respectively. Each ISP provides a different prefix. Also each router provides a different prefix to the host. The issue in this scenario is also ingress filtering used by each ISP.

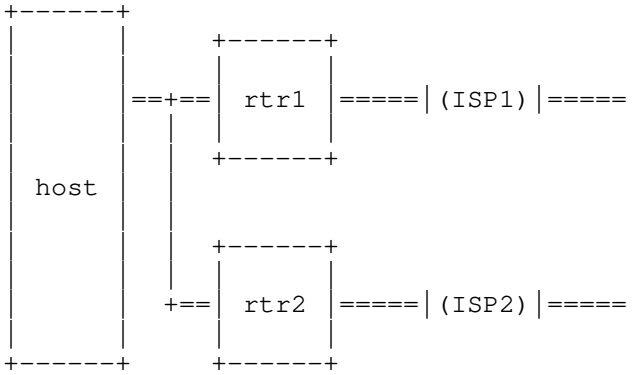



Figure 4: Simple Home Network with Two CPE Routers

The host has to select the source address from the prefixes of ISP1 or ISP2 when communicating with other hosts in ISP1 or ISP2. The next issue is to select the correct next hop router, rtr1 or rtr2 that can reach the right ISP, ISP1 or ISP2.

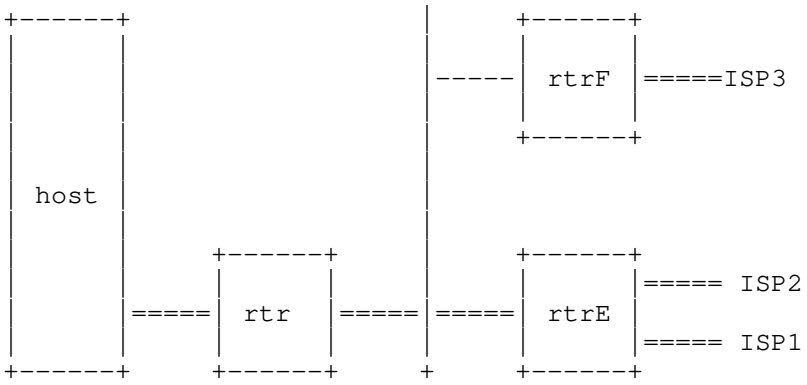



Figure 5: Shim6 Host with Two Routers

The last use case in Figure 5 is also a variation of multi-prefix multihoming use case above. In this case rtrE is connected to two ISPs. All ISPs are assumed to apply ingress routing. The host receives prefixes from each ISP and starts communicating with

external hosts, e.g. H1, H2, etc. H1 and H2 may be accessible both from ISP1 and ISP3.

The host receives multiple provider-allocated IPv6 address prefixes, e.g. P1, P2 and P3 for ISP1, ISP2 and ISP3 and supports shim6 protocol [RFC5533]. rtr is a CPE router and the default router for the host. rtr receives OSPF routes and has a default route for rtrE and rtrF.

## 4. Analysis of Source Address Dependent Routing

In this section we present an analysis of the scenarios of Section 3 and then discuss the relevance of SADR to the provisioning domains.

### 4.1. Scenarios Analysis

As in [RFC7157] we assume that the routers in Section 3 use Router Advertisements to distribute default route, next hop and source address prefixes supported in each next hop to the hosts or the gateway/CPE router relayes this information to the hosts.

Referring to the scenario in Figure 1, source address dependent routing can present a solution to the problem of the host wishes to reach a destination in network 2 and the host may choose rtr1 as the default router. The solution should start with the correct configuration of the host. The host should be configured with the next hop addresses and the prefixes supported in these next hops. This way the host having received many prefixes will have the correct knowledge in selecting the right source address and next hop when sending packets to remote destinations.

Note that similar considerations apply to the scenario in Figure 3.

In the configuration of the scenario in Figure 2 also it is useful to configure the host with the next hop addresses and the prefixes and source address prefixes they support. This will enable the host to select the right prefix when sending packets to the right next hop and avoid any ingress filtering.

Source address dependent routing in the use case of specialized egress routing may work as follows. The specialized service router advertizes one or more specific prefixes with appropriate source prefixes, e.g. to the CPE Router, rtr in Figure 2. The CPE router in turn advertizes the specific service's prefixes and source prefixes to the host. This will allow proper configuration at the host so that the host can use the service by sending the packets with the correct source and destination addresses.

Let us analyze the use case in Figure 4. If a source address dependent routing protocol is used, the two routers (rtr1 and rtr2) are both able to route traffic correctly, no matter which next-hop router and source address the host selects. In case the host chooses the wrong next hop router, e.g. for ISP2 rtr1 is selected, rtr1 will forward the traffic to rtr2 to be sent to ISP2 and no ingress filtering will happen.

Note that home networks are expected to comply with requirements for source address dependent routing and the routers will be configured accordingly, no matter which routing protocol, e.g. OSPF is used [I-D.ietf-homenet-hncp].

This would work but with issues. The host traffic to ISP2 will have to go over two links instead of one, i.e. the link bandwidth will be halved. Another possibility is rtr1 can send an ICMPv6 Redirect message to the host to direct the traffic to rtr2. Host would redirect ISP2 traffic to rtr2.

The problem with redirects is that ICMPv6 Redirect message can only convey two addresses, i.e. in this case the router address, or rtr2 address and the destination address, or the destination host in ISP2. That means the source address will not be communicated. As a result, the host would send packets to the same destination using both source addresses which causes rtr2 to send a redirect message to rtr1, resulting in ping-pong redirects sent by rtr1 and rtr2.

The best solution to these issues is to configure the host with both the next hop and the source address prefixes that the next hop supports. In homenets, each interface of the host can be configured by its next hop, so that all that is needed is to add the information on source address prefixes. This results in the hosts to select the right router no matter what.

Finally, the use case in Figure 5 shows that even though all the routers may have source address dependent routing support, the packets still may get dropped.

The host in Figure 5 starts external communication with H1 and sends the first packet with source address P3::iid. Since rtr has a default route to rtrE it will use this default route in sending the host's packet out towards rtrE. rtrE will route this packet to ISP1 and the packet will be dropped due to the ingress filtering.

A solution to this issue could be that rtrE having multiple routes to H1 could use the path through rtrF and could direct the packet to the other route, i.e. rtrF which would reach H1 in ISP3 without being

subject to ingress routing
[I-D.baker-6man-multiprefix-default-route].

### 4.2. Provisioning Domains and SADR

Consistent set of network configuration information is called provisioning domain (PvD). In case of multi-prefix multihoming (MPMH), more than one provisioning domain is present on a single link. In case of multi-prefix multiple interface (MPMI) environments, elements of the same domain may be present on multiple links. PvD aware nodes support association of configuration information into PvDs and use these PvDs to serve requests for network connections, e.g. chosing the right source address for the packets. PvDs can be constructed from one of more DHCP or Router Advertisement (RA) options carrying such information as PvD identity and PvD container [I-D.ietf-mif-mpvd-ndp-support], [I-D.ietf-mif-mpvd-dhcp-support]. PvDs constructed based on such information are called explicit PvDs [I-D.ietf-mif-mpvd-arch].

Apart from PvD identity, PvD content may be encapsulated in separate RA or DHCP options called PvD Container Option. Examples of such content are defined in [I-D.sarikaya-6man-next-hop-ra] and [I-D.sarikaya-dhc-6man-dhcpv6-sadr]. They constitute the content or parts of the content of an explicit PvD.

Explicit PvDs may be received from different interfaces. Single PvD may be accessible over one interface or simulatenously accessible over multiple interfaces. Explicit PvDs may be scoped to a configuration related to a particular interface, however in general this may not apply. What matters is PvD ID provided that PvD ID is authenticated by the node even in cases where the node has a single connected interface. The authentication of the PvD ID should meet the level required by the node policy. Single PvD information may be received over multiple interfaces as long as PvD ID is the same. This applies to the router advertisements (RAs) in which case a multi-homed host (that is, with multiple interfaces) should trust a message from a router on one interface to install a route to a different router on another interface.

## 5. Guidelines on Standardization Work

We presented many topologies in which a host with multiple interfaces or a multihomed host is connected to various networks or ISPs which in turn may apply ingress routing. Our scenario analysis showed that in order to avoid packets getting dropped due to ingress routing, source address dependent routing is needed. Also, source address dependent routing should be supported by routers throughout a site that has multiple exits.

In this section, we provide informative guidelines on different existing and future solutions vis a vis the scenarios presented in Section 3. We start with source address selection rule 5.5 and the scenarios it solves and continue with solutions that state exactly what information hosts need in terms of new router advertisement options for correct source address selection in those scenarios.

## 5.1. Source Address Selection Rule 5.5

One possible solution is the default source address selection Rule 5.5 in [RFC6724] which recommends to select source addresses advertized by the next hop. Considering the above scenarios, we can state that this rule can solve the problem in Figure 1, Figure 2 and Figure 3.

In using Rule 5.5 the following guidelines should be kept in mind. Source address selection rules can be distributed by DHCP server using DHCP Option OPTION_ADDRSEL_TABLE defined in [RFC7078].

In case of DHCP based host configuration, DHCP server can configure only the interface of the host to which it is directly connected. In order for Rule 5.5 to apply on other interfaces the option should be sent on those interfaces as well using [RFC7078].

The default source address selection Rule 5.5 solves that problem when an application sends a packet with an unspecified source address. In the presence of two default routes, one route will be chosen, and Rule 5.5 will make sure the right source address is used.

When the application selects a source address, i.e. the source address is chosen before next-hop selection, even though the source address is a way for the application to select the exit point, in this case that purpose will not be served. In the presence of multiple default routes, one will be picked, ignoring the source address which was selected by the application because it is known that IPv6 implementations are not required to remember which next-hops advertised which prefixes. Therefore, the next-hop router may not be the correct one, and the packets may be filtered.

This implies that the hosts should register which next-hop router announced each prefix.

## 5.2. Router Advertisement Option

There is a need to configure the host not only with the next hops and their prefixes but also with the source prefixes they support. Such a configuration may avoid the host getting ingress/egress policy error messages such as ICMP source address failure message.

If host configuration is done using router advertisement messages then there is a need to define new router advertisement options for source address dependent routing. These options include Route Prefix with Source Address/Prefix Option. Other options such as Next Hop Address with Route Prefix option and Next Hop Address with Source Address and Route Prefix option will be considered in Section 5.3.

As we observed in Section 4.1, the scenario in Figure 4 can be solved by defining a new router advertisement option, i.e. Route Prefix with Source Address/Prefix Option as defined in Section 13 in [I-D.sarikaya-6man-next-hop-ra].

If host configuration is done using DHCP then there is a need to define new DHCP options for Route Prefix with Source Address/Prefix. As mentioned above, DHCP server configuration is interface specific. New DHCP options for source address dependent routing such as route prefix and source prefix need to be configured for each interface separately.

The scenario in Figure 4 can be solved by defining a new DHCP option, i.e. Route Prefix with Source Address/Prefix Option, if DHCP configuration is a must.

### 5.3. Router Advertisement Option Set

The source address selection rule 5.5 may possibly be a solution for selecting the right source addresses for each next hop but there are cases where the next hop routers on each interface of the host are not known by the host initially. A typical use case is the Virtual Private Network (VPN) access. The host in VPN access is configured by the VPN router which should also give the information on the next hop routers and host needs to solicit the router advertisment using RS/RA exchange.

The solution then calls for configuring hosts with Next Hop Addresses and the Route Prefix, Source Address/Prefixes that they support. A set of new router advertisement options as in [I-D.sarikaya-6man-next-hop-ra] needs to be defined.

The guideline for this solution is that routers in the whole site should be configured to provide the correct configuration information to the hosts. This may result is fate sharing in which one router, e.g. VPN router failure may effect the whole system. In order to avoid such failures, the availability and reliability of routing paths need to be provided using Virtual Router Redundancy Protocol (VRRP) which is widely deployed in industry.

Additional guideline for this solution is that regular router operation calls for unsolicited router advertisements which are commonly available in shared links. Also this type of operation does not require inter router communication and thus avoids the fate sharing, i.e. each router can autonomously operate independent of other routers.

If host configuration is done using DHCP then there is a need to define new DHCP options for Next Hop Address, Route Prefix with Source Address/Prefix. Since DHCP server configuration is interface specific, new DHCP options for source address dependent routing such as next hop address, route prefix and source prefix need to be configured for each interface separately.

The scenarios in Figure 1, Figure 2, Figure 3 and Figure 4 as well as the ones involving the next hop addresses can be solved by defining new DHCP options as in [I-D.sarikaya-dhc-6man-dhcpv6-sadr].

### 5.4. Other Solutions

So far we have singled out the scenario in Figure 5. All the above solutions do not work in this case. This brings us the issue of IP path probing [I-D.naderi-ipv6-probing].

For a given destination, the host selects a source address and a next hop and sends its packet. When the selected path fails, in case of IP probing, the host can probe all available paths until finding one that works.

The guideline in probing is Source Address Dependent Routing (SADR) should be used, i.e. it is a necessary tool. Basically, SADR saves time in eliminating wrong paths, i.e. sending the packets to the wrong exit router. If SADR is not taken into account correctly the host will end up wasting resources trying to explore paths that are certain to fail.

## 6. Security Considerations

This document describes some use cases and thus brings no new security risks to the Internet.

## 7. IANA Considerations

None.

## 8. Acknowledgements

In writing this document, we benefited from the ideas expressed by the electronic mail discussion participants on 6man Working Group: Brian Carpenter, Ole Troan, Pierre Pfister, Alex Petrescu, Ray Hunter, Lorenzo Colitti and others. Pierre Pfister proposed the scenario in Figure 4 as well as some text for Rule 5.5.

## 9. References

### 9.1. Normative References

[I-D.ietf-homenet-hncp]
Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", draft-ietf-homenet-hncp-03 (work in progress), January 2015.

[ISO.10589.1992]
International Organization for Standardization, "Intermediate system to intermediate system intra-domain-routing routine information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473), ISO Standard 10589", ISO ISO.10589.1992, 1992.

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

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

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, March 2004.

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

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

[RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.

[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.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.

[RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, June 2009.

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

[RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.

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

[RFC7078] Matsumoto, A., Fujisaki, T., and T. Chown, "Distributing Address Selection Policy Using DHCPv6", RFC 7078, January 2014.

[RFC7157] Troan, O., Miles, D., Matsushima, S., Okimoto, T., and D. Wing, "IPv6 Multihoming without Network Address Translation", RFC 7157, March 2014.

## 9.2. Informative References

[I-D.baker-6man-multiprefix-default-route]
Baker, F., "Multiprefix IPv6 Routing for Ingress Filters", draft-baker-6man-multiprefix-default-route-00 (work in progress), November 2007.

[I-D.baker-ipv6-isis-dst-src-routing]
Baker, F. and D. Lamparter, "IPv6 Source/Destination Routing using IS-IS", draft-baker-ipv6-isis-dst-src-routing-02 (work in progress), October 2014.

[I-D.baker-ipv6-ospf-dst-src-routing]
Baker, F., "IPv6 Source/Destination Routing using OSPFv3", draft-baker-ipv6-ospf-dst-src-routing-03 (work in progress), August 2013.

[I-D.baker-rtgwg-src-dst-routing-use-cases]
Baker, F., "Requirements and Use Cases for Source/ Destination Routing", draft-baker-rtgwg-src-dst-routing-use-cases-01 (work in progress), October 2014.

[I-D.huitema-multi6-ingress-filtering]
Huitema, C., "Ingress filtering compatibility for IPv6 multihomed sites", draft-huitema-multi6-ingress-filtering-00 (work in progress), October 2004.

[I-D.ietf-mif-mpvd-arch]
Anipko, D., "Multiple Provisioning Domain Architecture", draft-ietf-mif-mpvd-arch-10 (work in progress), February 2015.

[I-D.ietf-mif-mpvd-dhcp-support]
Krishnan, S., Korhonen, J., and S. Bhandari, "Support for multiple provisioning domains in DHCPv6", draft-ietf-mif-mpvd-dhcp-support-00 (work in progress), August 2014.

[I-D.ietf-mif-mpvd-ndp-support]
Korhonen, J., Krishnan, S., and S. Gundavelli, "Support for multiple provisioning domains in IPv6 Neighbor Discovery Protocol", draft-ietf-mif-mpvd-ndp-support-00 (work in progress), August 2014.

[I-D.naderi-ipv6-probing]
Naderi, H. and B. Carpenter, "Experience with IPv6 path probing", draft-naderi-ipv6-probing-00 (work in progress), October 2014.

[I-D.sarikaya-6man-next-hop-ra]
Sarikaya, B., "IPv6 RA Options for Next Hop Routes", draft-sarikaya-6man-next-hop-ra-04 (work in progress), December 2014.

[I-D.sarikaya-dhc-6man-dhcpv6-sadr]
Sarikaya, B., "DHCPv6 Route Options for Source Address Dependent Routing", draft-sarikaya-dhc-6man-dhcpv6-sadr-00 (work in progress), December 2014.

Author's Address

Behcet Sarikaya
Huawei USA
5340 Legacy Dr. Building 175
Plano, TX 75024

Email: sarikaya@ieee.org

6man Group
Internet-Draft
Intended status: Standards Track
Expires: August 29, 2015

E. Vyncke, Ed.
S. Previdi
Cisco Systems, Inc.
D. Lebrun
Universite Catholique de Louvain

February 25, 2015

# IPv6 Segment Routing Security Considerations
# draft-vyncke-6man-segment-routing-security-02

## Abstract

Segment Routing (SR) allows a node to steer a packet through a controlled set of instructions, called segments, by prepending a SR header to the packet. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any path (topological, or application/service based) while maintaining per-flow state only at the ingress node to the SR domain.

Segment Routing can be applied to the IPv6 data plane with the addition of a new type of Routing Extension Header. This document analyzes the security aspects of the Segment Routing Extension Header (SRH) and how it is used by SR capable nodes to deliver a secure service.

## 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

Table of Contents



## 1. Introduction

This document analyzes the security threat model, the security issues and proposed solutions related to the new routing header for segment routing with an IPv6 data plane.

The Segment Routing Header (SRH) is simply another type of the routing header as described in RFC 2460 [RFC2460] and is:

- inserted by a SR edge router when entering the segment routing domain or by the originating host itself. The source host can even be outside the SR domain;

- inspected and acted upon when reaching the destination address of the IP header per RFC 2460 [RFC2460].

Per RFC2460 [RFC2460], routers on the path that simply forward an IPv6 packet (i.e. the IPv6 destination address is none of theirs) will never inspect and process the content of SRH. Routers whose one interface IPv6 address equals the destination address field of the IPv6 packet MUST to parse the SRH and, if supported and if the local configuration allows it, MUST act accordingly to the SRH content.

According to RFC2460 [RFC2460], the default behavior of a non SR-capable router upon receipt of an IPv6 packet with SRH destined to an address of its, is to:

- ignore the SRH completely if the Segment Left field is 0 and proceed to process the next header in the IPv6 packet;

- discard the IPv6 packet if Segment Left field is greater than 0, it MAY send a Parameter Problem ICMP message back to the Source Address.

### 1.1. Segment Routing Documents

Segment Routing terminology is defined in [I-D.ietf-spring-segment-routing] and in [I-D.ietf-spring-problem-statement]. Segment Routing use cases are described in [I-D.filsfils-spring-segment-routing-use-cases]. Segment Routing protocol extensions are defined in [I-D.ietf-isis-segment-routing-extensions], and [I-D.ietf-ospf-ospfv3-segment-routing-extensions].

Segment Routing IPv6 use cases are described in [I-D.ietf-spring-ipv6-use-cases]. And the IPv6 Segment Routing header is described in [I-D.previdi-6man-segment-routing-header].

## 2. Threat model

### 2.1. Source routing threats

Using a SRH is similar to source routing, therefore it has some well-known security issues as described in RFC4942 [RFC4942] section 2.1.1 and RFC5095 [RFC5095]:

- amplification attacks: where a packet could be forged in such a way to cause looping among a set of SR-enabled routers causing unnecessary traffic, hence a Denial of Service (DoS) against bandwidth;

- reflection attack: where a hacker could force an intermediate node to appear as the immediate attacker, hence hiding the real attacker from naive forensic;

- bypass attack: where an intermediate node could be used as a stepping stone (for example in a De-Militarized Zone) to attack another host (for example in the datacenter or any back-end server).

### 2.2. Applicability of RFC 5095 to SRH

First of all, the reader must remember this specific part of section 1 of RFC5095 [RFC5095], "A side effect is that this also eliminates benign RH0 use-cases; however, such applications may be facilitated by future Routing Header specifications.". In short, it is not forbidden to create new secure type of Routing Header; for example, RFC 6554 (RPL) [RFC6554] also creates a new Routing Header type for a specific application confined in a single network.

In the segment routing architecture described in [I-D.ietf-spring-segment-routing] there are basically two kinds of nodes (routers and hosts):

- nodes within the SR domain, which is within one single administrative domain, i.e., where all nodes are trusted anyway else the damage caused by those nodes could be worse than amplification attacks: traffic interception, man-in-the-middle attacks, more server DoS by dropping packets, and so on.

- nodes outside of the SR domain, which is outside of the administrative segment routing domain hence they cannot be trusted because there is no physical security for those nodes, i.e., they can be replaced by hostile nodes or can be coerced in wrong behaviors.

The main use case for SR consists of the single administrative domain where only trusted nodes with SR enabled and configured participate

in SR: this is the same model as in RFC6554 [RFC6554]. All non-trusted nodes do not participate as either SR processing is not enabled by default or because they only process SRH from nodes within their domain.

Moreover, all SR nodes ignore SRH created by outsiders based on topology information (received on a peering or internal interface) or on presence and validity of the HMAC field. Therefore, if intermediate nodes ONLY act on valid and authorized SRH (such as within a single administrative domain), then there is no security threat similar to RH-0. Hence, the RFC 5095 [RFC5095] attacks are not applicable.

### 2.3. Service stealing threat

Segment routing is used for added value services, there is also a need to prevent non-participating nodes to use those services; this is called 'service stealing prevention'.

### 2.4. Topology disclosure

The SRH may also contains IPv6 addresses of some intermediate SR-nodes in the path towards the destination, this obviously reveals those addresses to the potentially hostile attackers if those attackers are able to intercept packets containing SRH. On the other hand, if the attacker can do a traceroute whose probes will be forwarded along the SR path, then there is little learned by intercepting the SRH itself. Also the clean-bit of SRH can help by removing the SRH before forwarding the packet to potentially a non-trusted part of the network.

### 2.5. ICMP Generation

Per section 4.4 of RFC2460 [RFC2460], when destination nodes (i.e. where the destination address is one of theirs) receive a Routing Header with unsupported Routing Type, the required behavior is:

- If Segments Left is zero, the node must ignore the Routing header and proceed to process the next header in the packet.

- If Segments Left is non-zero, the node must discard the packet and send an ICMP Parameter Problem, Code 0, message to the packet's Source Address, pointing to the unrecognized Routing Type.

This required behavior could be used by an attacker to force the generation of ICMP message by any node. The attacker could send packets with SRH (with Segment Left set to 0) destined to a node not supporting SRH. Per RFC2460 [RFC2460], the destination node could

generate an ICMP message, causing a local CPU utilization and if the source of the offending packet with SRH was spoofed could lead to a reflection attack without any amplification.

It must be noted that this is a required behavior for any unsupported Routing Type and not limited to SRH packets. So, it is not specific to SRH and the usual rate limiting for ICMP generation is required anyway for any IPv6 implementation and has been implemented and deployed for many years.

## 3. Security fields in SRH

This section summarizes the use of specific fields in the SRH; they are integral part of [I-D.previdi-6man-segment-routing-header] and they are again described here for reader's sake. They are based on a key-hashed message authentication code (HMAC).

The security-related fields in SRH are:

- HMAC Key-id, 8 bits wide;
- HMAC, 256 bits wide (optional, exists only if HMAC Key-id is not 0).

The HMAC field is the output of the HMAC computation (per RFC 2104 [RFC2104]) using a pre-shared key identified by HMAC Key-id and of the text which consists of the concatenation of:

- the source IPv6 address;
- First Segment field;
- an octet whose bit-0 is the clean-up bit flag and others are 0;
- HMAC Key-id;
- all addresses in the Segment List.

The purpose of the HMAC field is to verify the validity, the integrity and the authorization of the SRH itself. If an outsider of the SR domain does not have access to a current pre-shared secret, then it cannot compute the right HMAC field and the first SR router on the path processing the SRH and configured to check the validity of the HMAC will simply reject the packet.

The HMAC field is located at the end of the SRH simply because only the router on the ingress of the SR domain needs to process it, then all other SR nodes can ignore it (based on local policy) because they

trust the upstream router. This is to speed up forwarding operations because SR routers which do not validate the SRH do not need to parse the SRH until the end.

The HMAC Key-id field allows for the simultaneous existence of several hash algorithms (SHA-256, SHA3-256 ... or future ones) as well as pre-shared keys. This allows for pre-shared key roll-over when two pre-shared keys are supported for a while when all SR nodes converged to a fresher pre-shared key. The HMAC Key-id field is opaque, i.e., it has neither syntax not semantic except as an index to the right combination of pre-shared key and hash algorithm and except that a value of 0 means that there is no HMAC field. It could also allow for interoperation among different SR domains if allowed by local policy and assuming a collision-free Key Id allocation.

When a specific SRH is linked to a time-related service (such as turbo-QoS for a 1-hour period) where the DA, Segment ID (SID) are identical, then it is important to refresh the shared-secret frequently as the HMAC validity period expires only when the HMAC Key-id and its associated shared-secret expires.

### 3.1. Selecting a hash algorithm

The HMAC field in the SRH is 256 bit wide. Therefore, the HMAC MUST be based on a hash function whose output is at least 256 bits. If the output of the hash function is 256, then this output is simply inserted in the HMAC field. If the output of the hash function is larger than 256 bits, then the output value is truncated to 256 by taking the least-significant 256 bits and inserting them in the HMAC field.

SRH implementations can support multiple hash functions but MUST implement SHA-2 [FIPS180-4] in its SHA-256 variant.

NOTE: SHA-1 is currently used by some early implementations used for quick interoperations testing, the 160-bit hash value must then be right-hand padded with 96 bits set to 0. The authors understand that this is not secure but is ok for limited tests.

### 3.2. Performance impact of HMAC

While adding a HMAC to each and every SR packet increases the security, it has a performance impact. Nevertheless, it must be noted that:

- the HMAC field is used only when SRH is inserted by a device (such as a home set-up box) which is outside of the segment routing domain. If the SRH is added by a router in the trusted segment

routing domain, then, there is no need for a HMAC field, hence no performance impact.

- when present, the HMAC field MUST only be checked and validated by the first router of the segment routing domain, this router is named 'validating SR router'. Downstream routers MAY NOT inspect the HMAC field.

- this validating router can also have a cache of <IPv6 header + SRH, HMAC field value> to improve the performance. It is not the same use case as in IPsec where HMAC value was unique per packet, in SRH, the HMAC value is unique per flow.

- Last point, hash functions such as SHA-2 have been optmized for security and performance and there are multiple implementations with good performance.

With the above points in mind, the performance impact of using HMAC is minimized.

### 3.3. Pre-shared key management

The field HMAC Key-id allows for:

- key roll-over: when there is a need to change the key (the hash pre-shared secret), then multiple pre-shared keys can be used simultaneously. The validating routing can have a table of <HMAC Key-id, pre-shared secret> for the currently active and future keys.

- different algorithm: by extending the previous table to <HMAC Key-id, hash function, pre-shared secret>, the validating router can also support simultaneously several hash algorithms (see section Section 3.1)

The pre-shared secret distribution can be done:

- in the configuration of the validating routers, either by static configuration or any SDN oriented approach;

- dynamically using a trusted key distribution such as [RFC6407]

The intent of this document is NOT to define yet-another-key-distribution-protocol.

# 4. Deployment Models

## 4.1. Nodes within the SR domain

A SR domain is defined as a set of interconnected routers where all routers at the perimeter are configured to insert and act on SRH. Some routers inside the SR domain can also act on SRH or simply forward IPv6 packets.

The routers inside a SR domain can be trusted to generate SRH and to process SRH received on interfaces that are part of the SR domain. These nodes MUST drop all SRH packets received on an interface that is not part of the SR domain and containing a SRH whose HMAC field cannot be validated by local policies. This includes obviously packet with a SRH generated by a non-cooperative SR domain.

If the validation fails, then these packets MUST be dropped, ICMP error messages (parameter problem) SHOULD be generated (but rate limited) and SHOULD be logged.

## 4.2. Nodes outside of the SR domain

Nodes outside of the SR domain cannot be trusted for physical security; hence, they need to request by some trusted means (outside of the scope of this document) a complete SRH for each new connection (i.e. new destination address). The received SRH MUST include a HMAC Key-id and HMAC field which is computed correctly (see Section 3).

When an outside node sends a packet with an SRH and towards a SR domain ingress node, the packet MUST contain the HMAC Key-id and HMAC field and the the destination address MUST be an address of a SR domain ingress node .

The ingress SR router, i.e., the router with an interface address equals to the destination address, MUST verify the HMAC field with respect to the HMAC Key-id.

If the validation is successful, then the packet is simply forwarded as usual for a SR packet. As long as the packet travels within the SR domain, no further HMAC check needs to be done. Subsequent routers in the SR domain MAY verify the HMAC field when they process the SRH (i.e. when they are the destination).

If the validation fails, then this packet MUST be dropped, an ICMP error message (parameter problem) SHOULD be generated (but rate limited) and SHOULD be logged.

### 4.3. SR path exposure

As the intermediate SR nodes addresses appears in the SRH, if this SRH is visible to an outsider then he/she could reuse this knowledge to launch an attack on the intermediate SR nodes or get some insider knowledge on the topology. This is especially applicable when the path between the source node and the first SR domain ingress router is on the public Internet.

The first remark is to state that 'security by obscurity' is never enough; in other words, the security policy of the SR domain MUST assume that the internal topology and addressing is known by the attacker. A simple traceroute will also give the same information (with even more information as all intermediate nodes between SID will also be exposed). IPsec Encapsulating Security Payload [RFC4303] cannot be use to protect the SRH as per RFC4303 the ESP header must appear after any routing header (including SRH).

To prevent a user to leverage the gained knowledge by intercepting SRH, it it recommended to apply an infrastructure Access Control List (iACL) at the edge of the SR domain. This iACL will drop all packets from outside the SR-domain whose destination is any address of any router inside the domain. This security policy should be tuned for local operations.

### 4.4. Impact of BCP-38

BCP-38 [RFC2827], also known as "Network Ingress Filtering", checks whether the source address of packets received on an interface is valid for this interface. The use of loose source routing such as SRH forces packets to follow a path which differs from the expected routing. Therefore, if BCP-38 was implemented in all routers inside the SR domain, then SR packets could be received by an interface which is not expected one and the packets could be dropped.

As a SR domain is usually a subset of one administrative domain, and as BCP-38 is only deployed at the ingress routers of this administrative domain and as packets arriving at those ingress routers have been normally forwarded using the normal routing information, then there is no reason why this ingress router should drop the SRH packet based on BCP-38. Routers inside the domain commonly do not apply BCP-38; so, this is not a problem.

## 5. IANA Considerations

There are no IANA request or impact in this document.

## 6. Manageability Considerations

TBD

## 7. Security Considerations

Security mechanisms applied to Segment Routing over IPv6 networks are detailed in Section 3.

## 8. Acknowledgements

The authors would like to thank Dave Barach and Stewart Bryant for their contributions to this document.

## 9. References

### 9.1. Normative References

[FIPS180-4] National Institute of Standards and Technology, "FIPS 180-4 Secure Hash Standard (SHS)", March 2012, <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>.

[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.

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

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

[RFC6407] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain of Interpretation", RFC 6407, October 2011.

### 9.2. Informative References

[I-D.filsfils-spring-segment-routing-use-cases]
Filsfils, C., Francois, P., Previdi, S., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., Kini, S., and E. Crabbe, "Segment Routing Use Cases", draft-filsfils-spring-segment-routing-use-cases-01 (work in progress), October 2014.

[I-D.ietf-isis-segment-routing-extensions]
Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-03 (work in progress), October 2014.

[I-D.ietf-ospf-ospfv3-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPFv3 Extensions for Segment Routing", draft-ietf-ospf-ospfv3-segment-routing-extensions-02 (work in progress), February 2015.

[I-D.ietf-spring-ipv6-use-cases]
Brzozowski, J., Leddy, J., Leung, I., Previdi, S., Townsley, W., Martin, C., Filsfils, C., and R. Maglione, "IPv6 SPRING Use Cases", draft-ietf-spring-ipv6-use-cases-03 (work in progress), November 2014.

[I-D.ietf-spring-problem-statement]
Previdi, S., Filsfils, C., Decraene, B., Litkowski, S., Horneffer, M., and R. Shakir, "SPRING Problem Statement and Requirements", draft-ietf-spring-problem-statement-03 (work in progress), October 2014.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-ietf-spring-segment-routing-01 (work in progress), February 2015.

[I-D.previdi-6man-segment-routing-header]
Previdi, S., Filsfils, C., Field, B., and I. Leung, "IPv6 Segment Routing Header (SRH)", draft-previdi-6man-segment-routing-header-05 (work in progress), January 2015.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC4942] Davies, E., Krishnan, S., and P. Savola, "IPv6 Transition/ Co-existence Security Considerations", RFC 4942, September 2007.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, March 2012.

## Authors' Addresses

Eric Vyncke (editor)
Cisco Systems, Inc.
De Kleetlaann 6A
Diegem 1831
Belgium

Email: evyncke@cisco.com

Stefano Previdi
Cisco Systems, Inc.
Via Del Serafico, 200
Rome 00142
Italy

Email: sprevidi@cisco.com

David Lebrun
Universite Catholique de Louvain
Place Ste Barbe, 2
Louvain-la-Neuve, 1348
Belgium

Email: david.lebrun@uclouvain.be

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2015

A. Wang
China Telecom
S. Jiang
Huawei Technologies Co., Ltd

March 8, 2015

# IPv6 Flow Label Reflection
# draft-wang-6man-flow-label-reflection-01

## Abstract

The current definition of the IPv6 Flow Label focuses mainly on how the packet source forms the value of this field and how the forwarder in-path treats it. In network operations, there are needs to correlate an upstream session and the corresponding downstream session together. This document propose a flow label reflection mechanism that network devices copy the flow label value from received packets to the corresponding flow label field in return packets. This mechanism could simplify the network traffic recognition process in network operations and make the policy for both directions of traffic of one session consistent.

## 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 September 9, 2015.

Table of Contents



## 1. Introduction

The IPv6 flow label [RFC6437] in the fixed IPv6 header is designed to differentiate the various flow session of IPv6 traffic; it can accelerate the clarification and treatment of IPv6 traffic by the network devices in its forwarding path. In practice, many current implementations use the 5-tuple {dest addr, source addr, protocol, dest port, source port} as the identifier of network flows. However, transport-layer information, such as the port numbers, is not always in a fixed position, since it follows any IPv6 extension headers that may be present; in contrast, the flow label is at a fixed position in every IPv6 packet and easier to access. In fact, the logic of finding the transport header is always more complex for IPv6 than for IPv4, due to the absence of an Internet Header Length field in IPv6. Additionally, if packets are fragmented, the flow label will be present in all fragments, but the transport header will only be in one packet. Therefore, within the lifetime of a given transport-

layer connection, the flow label can be a more convenient "handle" than the port number for identifying that particular connection.

The usages of IPv6 flow label, so far as briefly summarized in Section 1.1, only exploit the characteristic of IPv6 flow label in one direction.

In current practice, an application session is often recognized as two separated IP traffics, in two opposite directions. However, from the point view of a service provider, the upstream and downstream of one session should be handled together, particularly, when application-aware operations are placed in the network. A ubiquitous example is that end user initiates a request, with small-scale data transmitted, towards a content server, then the server responds with a large set of follow-up packets. The bi-directional flows should be correlated together and handled with the same policy. Ideally, the request embeds a flow recognition identifier that is accessible and the follow-up response packets carry the same identifier. The flow label is a good choice for the flow recognition identifier.

This document proposes a flow label reflection mechanism so that network devices copy the flow label value from received packets to the corresponding flow label field in return packets. By having the same flow label value in the downstream and upstream of one IPv6 traffic session, the network traffic recognition process and the traffic policy deployment in network operations could be simplified. It may also increase the accuracy of network traffic recognition.

Several applicable scenarios of the IPv6 flow label reflection are also given, in Section 5. For now, this document only considers the scenario in a single administrative domain, although the IPv6 flow label reflection mechanism may also bring benefits into cross domain scenarios.

### 1.1. Summary of the current usage for IPv6 Flow Label

[RFC6438] describe the usage of IPv6 Flow Label for ECMP and link aggregation in Tunnels; it mainly utilizes the random distribution characteristic of IPv6 flow label. [RFC7098] also describes similar usage in server farms.

All these usage scenarios consider only the usage of IPv6 flow label in one direction, while many bi-directional network traffics need to be treated together.

## 2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. When these words are not in ALL CAPS (such as "should" or "Should"), they have their usual English meanings, and are not to be interpreted as [RFC2119] key words.

Flow Label Reflection A mechanism/behavior so that a network device copies the value of flow label from a IPv6 flow into a corresponding return IPv6 flow.

Flow Label Reflection Device A network device that applies the flow label reflection mechanism. It is the end of an IPv6 flow and the initiation node of the corresponding return IPv6 flow.

## 3. Potential Benefit of Flow Label Reflection

With flow label reflection mechanism, the IPv6 Flow Label could be used to correlate the upstream and downstream packets of bi-directional traffics:

- It makes the downstream and upstream of one session be easily recognized. It makes the correlation of traffic and then the recognition of various traffics easier.

- The network operator can easily apply the same policy to the bi-directional traffic of one interested session

- The traffic analyzer can also easily correlate the upstream and downstream of one session to find the symptoms of various internet protocols.

## 4. Flow Label Reflection Behaviors on Network Devices

To fulfill the flow label reflection mechanism, the below proposed behaviors on network devices:

- The generation method of IPv6 flow label in source IPv6 node SHOULD follow the guidelines in [RFC6437], that is the IPv6 flow label should be generated randomly and distributed enough.

- On the Flow Label Reflection Device, the value of IPv6 Flow Label from received packets SHOULD be copied into the corresponding flow label field in return packets by the flow label reflection devices.

- The forwarding nodes within the management domain SHOULD follow the specification in [RFC6437], that is the IPv6 flow label SHOULD NOT be modified in the path, unless flow label value in arriving packets is zero. The forwarding nodes MAY follow the specification in [RFC6438] when using the flow label for load balancing by equal cost multipath (ECMP) routing and for link aggregation, particularly for IPv6-in-IPv6 tunneled traffic.

- The network traffic recognition devices, or devices that may have differentiated operations per flow, SHOULD recognize and analyze network traffics based on 3-tuple of {dest addr, source addr, flowlabel}. It SHOULD consider the traffics that have same flow label value and reversed source/dest addr as upstream and downstream of the same flow, match them together to accomplish the traffic recognition process.

- Other network operations MAY also be based on 3-tuple of {dest addr, source addr, flowlabel}.

## 5. Applicable Scenarios

This section describes some applicable scenarios, which network operators can benefit from deploying the flow label reflection mechanism. It is not a complete enumeration. More scenarios may be introduced in the future.

### 5.1. Flow Label Reflection on CP servers

There is rapidly increasing requirement from service providers (SP) to cooperate with the content providers (CP) to provide more accurate services and charging policies based on accurate traffic recognition. The service providers need to recognize the CP/SP's bi-directional traffics at the access edge devices of the network, such as BRAS/PDSN/P-GW devices.

Normally, the burden for these edge devices to recognize the subscriber's upstream traffic is light, because request messages are typically small. But they often need more resource to recognize downstream traffics, which normally contain large data. With flow label reflection on CP servers, recognition based on the 3-tuple of {dest addr, source addr, flowlabel} would reduce the consumption of recognition and make the correlation process much easier.

In this scenario, the CP servers would be the Flow Label Reflection Devices. They copy the flow label value from received upstream user request packets to the corresponding flow label field in return downstream packets.

The access edge devices of service provider scrutinize the subscriber's upstream IPv6 traffic and record the binding of 3-tuple and traffic-specific policy. If the flow label is zero, the access edge devices must rewrite the flow label value according to local policy. With the recorded binding information, the access edge devices can easily recognize and match the downstream packet to the previous recognized upstream packet, by just accessing 3-tuple. The edge devices can then apply the corresponding traffic policy to the upstream/downstream of the session to the cooperated CP.

Note: this mechanism may not reliable when the CP servers are not directly connected to the service provider, because there is no guarantee the flow label would not be changed by intermediate devices in other administrative domains.

### 5.2. Flow Label Reflection for Bi-direction Tunnels

Tunnel is ubiquitous within service provider networks. It is very difficult (important if the tunnel is encrypted) for intermediate network devices to recognize the inner encapsulated packet, although such recognition could be very helpful in some scenarios, such as traffic statistics, network diagnoses, etc. Furthermore, such recognition normally requires to correlate bi-direction traffic together. The flow label reflection mechanism could provide help in such requirement scenarios.

In this scenario, the tunnel end devices would be the Flow Label Reflection Devices. They record the flow label value from received tunnel packets, and copy it to the corresponding flow label field in return packets, which can be recognized by 5-tuple or 3-tuple of the inner packet at the tunnel end devices.

The tunnel initiating devices should generate different flow label values for different inner flow traffics based on their 5-tuple or 3-tuple in accordance with [RFC6437]. Note: if the inner flow is encryption in ESP model [RFC4303], the transport-layer port numbers are inaccessiable. In such case, 5-tuple is not available.

Then the intermediate network device can easily distinguish the different flow within the same tunnel transport link and correlate bi-direction traffics of same flow together. This can also increase the service provider's traffic control capabilities.

This mechanism can also work when the encapsulated traffics are IPv4 traffics, such as DS-Lite scenario [RFC6333]. The IPv4 5-tuple may be used as the input for the flow label generation.

### 5.3. Flow Label Reflection on edge devices

If the flow label reflection mechanisms have been applied on peer host, the service provider could always use it for bi-directional traffic recognition. However, there is no guarantee the flow label would not be changed by intermediate devices in other administrative domains. Therefore, to make the flow label value trustful, the edge devices need to validate the flow label reflection.

In this scenario, the edge devices would be the (backup) Flow Label Reflection Devices. They record the flow label value from the packets that leave the domain. When the corresponding flow label field in return packets are recognized by 5-tuple or 3-tuple at the edge devices, the edge devices should check the flow label as below:

- if the flow label matches the record value, it remains;
- if the flow label is zero, the edge devices copy the record value into it;
- if the flow label is non-zero, but does not matches the record value, the edge devices can decide the flow label are modified by other intermediate devices (with the assumption the peer host has reflect the original flow label), then restore the flow label using the record value.

Then the network recognition devices located anywhere within the service provider network could easily correlate bi-directional traffics together, and apply traffic-specific policy accordingly.

### 5.4. Misc Possible Scenarios

In the below scenarios, the flow label reflection mechanism needs to be combined with other mechanisms in order to achieve the design goals.

#### 5.4.1. Aid to mitigate the ND cache DDoS Attack

Neighbor Discovery Protocol [RFC4861][RFC4861] is vulnerable for the possible DDoS attack to the device's ND cache, see section 11.1, [RFC4861]. There are many proposals are aiming to mitigate this problem, but none of them are prevalent now. It is mainly because that there is no obvious mechanism to assure the validation of the NS/RS packet on the first arrival, the receiving node by default will cache the link-layer address of the NA packet. Reverse detection mechanisms can be added to solve this issue. However, for reverse detection mechanisms, there would be another issue: how to pair the return NA/RA packet with the NS/RS packet on the sending node. It

can be solved by applying the flow label reflection mechanism in the return NA/RA packet. Then the sending node can pair the reverse detect NS/RS packet with original NA/RA packet and response to the reverse detect NS/RS packet correctly. Only the NS/RS packet that passed the reverse detection validation will be accept by the node and the link-layer address within it will be cached.

#### 5.4.2. Improve the efficiency of PTB problem solution in load-balance environment

[I-D.v6ops-pmtud-ecmp-problem] introduces the Packet Too Big [RFC4443] problem in load-balance environment. The downstream packet from a server, which responses to a client request message, may meet a forwarding node that rejects the packet for "too big" reason. The PTB error ICMPv6 message should be returned to the original server. However, it requests the load balancer to distribute the PTB error ICMPv6 message based on the information of the invoking packet within the ICMPv6 packet, not the ICMPv6 packet itself. The load balancer needs to obtain the source IP address and transport port information within the invoking packet.

However, if both the server and the forwarding node that generates the PTB message apply the flow label reflection mechanism, the PTB error ICMPv6 message would have the same flow label with the original client request message. Then, the load balancer, that follows [RFC7098], could easily forward the PTB packet to same server without parsing the transport port in the invoking packet, thus increases the efficiency.

## 6. Deployment Consideration

The IPv6 flow label reflection mechanism requires the "Flow Label Reflection Device" to be stateful, store the flow label value and copy it to the corresponding return packet. Such change cannot be accomplished within a short term, and therefore the deployment of this mechanism will be accomplished gradually. During the incremental deployment period, the traditional recognition mechanisms, which are more expensive, would coexist. The traffics that could not be recognized by 3-tuple of {dest addr, source addr, flowlabel} could fall back to the traditional process or be skipped over by advanced services. The more devices support the flow label reflection mechanism, the less consumption for traffic recognition from the network management perspective, or the better coverage of advanced services that are based on the traffic recognition.

## 7. Security Considerations

Security aspects of the flow label are discussed in [RFC6437]. A malicious source or man-in-the-middle could disturb the traffic recognition by manipulating flow labels. However, the worst case is that fall back to the current practice that an application session is often recognized as two separated IP traffics. The flow label does not significantly alter this situation.

Specifically, the IPv6 flow label specification [RFC6437] states that "stateless classifiers should not use the flow label alone to control load distribution." This is answered by also using the source and destination addresses with flow label.

## 8. IANA Considerations

This draft does not request any IANA action.

## 9. Acknowledgements

The authors would like to thanks Brian Carpenter, who gave many useful advices. The authors would also like to thanks the valuable comments made by Fred Baker, Lee Howard, Mark ZZZ Smith, Jeroen Massar, Florent Fourcot and other members of V6OPS WG. Also, special thanks for Florent Fourcot, who have implemented the flow label reflection mechanims in the Linux.

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

## 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.

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

[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.

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

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, November 2011.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, November 2011.

## 10.2. Informative References

[I-D.v6ops-pmtud-ecmp-problem] Byerly, M., Hite, M., and J. Jaeggli, "Close encounters of the ICMP type 2 kind (near misses with ICMPv6 PTB)", draft-v6ops-pmtud-ecmp-problem-00 (work in progress), August 2014.

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

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

[RFC7098] Carpenter, B., Jiang, S., and W. Tarreau, "Using the IPv6 Flow Label for Load Balancing in Server Farms", RFC 7098, January 2014.

## Authors' Addresses

Aijun Wang
China Telecom
Beijing Research Institute, China Telecom Cooperation Limited
No.118, Xizhimenneidajie, Xicheng District, Beijing 100035
China

Phone: 86-10-58552347
Email: wangaj@ctbri.com.cn

Sheng Jiang
Huawei Technologies Co., Ltd
Q14, Huawei Campus, No.156 Beiqing Road
Hai-Dian District, Beijing, 100095
P.R. China

Email: jiangsheng@huawei.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 1, 2015

A. Yourtchenko
cisco
E. Nordmark
Arista Networks
February 28, 2015

# A survey of issues related to IPv6 Duplicate Address Detection
draft-yourtchenko-6man-dad-issues-01

## Abstract

This document enumerates the practical issues observed with respect to Duplicate Address Detection.

## 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 September 1, 2015.

## Copyright Notice

Table of Contents

## 1. Introduction

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].

Duplicate Address Detection (DAD) is a procedure in IPv6 performed on an address before it can be assigned to an interface [RFC2462]. By default it consists of sending a single multicast Neighbor Soliciation message and waiting for a response for one second. If no response is received, the address is declared to not be a duplicate. Once the address has been tested once, there is no further attempts to check for duplicates (unless the interface is re-initialized).

On one hand, it is mandatory for all addresses. On the other hand, it is a "best effort" activity. These somewhat counter-intuitive properties result in some issues that arise related to DAD. They are listed below. The issues have been grouped to facilitate discussing them.

## 2. Open Issues

Whether it is due to the assumptions made in 1995, or changes in how networks are built or deployed, there are many reasons why DAD would fail to detect a duplicate even when one exists. From a historical perspective it is important to keep in mind that when DAD was designed we had two forms of IPv6 addresses; those derived from EUI-64 and statically assigned. Since the IETF has developed additional methods for address assignment like DHCPv6 and addresses that improve privacy by reducing linkability.

### 2.1. Robustness: Interaction with delay in forwarding

The DAD makes an assumption that if a link layer is up, the traffic can be immediately forwarded, which is frequently not the case in modern networks. Two prominent cases include the switches running Spanning Tree Protocol (STP), and bridging modems.

When a port on an STP-enabled switch comes up, it goes through three phases of Listening then Learning then Forwarding. The default is to keep it for 15 seconds in Listening and 15 seconds in Learning states. During this time no user traffic is forwarded by the switch from and to this port. Therefore, if a DAD process happens during this period it is guaranteed to not detect any duplicates. This results in DAD being ineffective for link-local and otherwise pre configured addresses.

Similarly, a modem-like device whose line status is invisible to IP stack either within the modem or to a host connected on the Ethernet side, also renders the DAD ineffective - the delay before the connectivity is established can be much longer than any DAD wait.

Some of the link types, notably cable modems, have link-specific standards to address this issue by requiring a new DAD each time the RF-side interface bounces, as well as bouncing the LAN interface triggered by the bounce of the RF interface.

Note that [I-D.ietf-6man-resilient-rs] makes the router solicitation resilent to the above cases, but there is no counterpart to make DAD robust.

## 2.2. Robustness: Behavior on links with unreliable multicast

DAD requires two multicast messages to pass through - the NS and NA. Thus it shows a noticeable failure rate on links that do not pass multicast reliably e.g. the 802.11a/b/g/n series of technologies. See [I-D.vyncke-6man-mcast-not-efficient] for more information.

The author's ad-hoc experimentation at IETF90 revealed the success rate of detecting the duplicate address on the IETF WiFi network being about 4 in 5. This may violate the assumptions that other protocols make.

## 2.3. Robustness: Partition-join tolerance

[RFC4862] explicitly mentions this problem: "Note that the method for detecting duplicates is not completely reliable, and it is possible that duplicate addresses will still exist (e.g., if the link was partitioned while Duplicate Address Detection was performed)."

In contrast, IPv4 stacks typically implement the Address Conflict Detection (ACD) from [RFC5227]. This disparity results in a less robust operation of IPv6 compared to IPv4 and is undesirable.

Note that solutions along the lines of ACD, while improving robustness, might result in more resource usage in on the links and nodes by multicasting more ND packets.

## 2.4. Robustness: Behavior on collision

[RFC4862] in its section "5.4.5. When Duplicate Address Detection Fails" is much more prescriptive than [RFC2462] that it superceeds. However, it has been observed that some implementations may simply reset the network interface and attempt the DAD process again. This behavior, while being more resilient in case the DAD failure is

happening erroneously, is different from what is recommended in the standard.

TBD: Do the other RFCs for address allocation require some retry behavior?

### 2.5. Energy Efficiency

The use of multicast messages for DAD results in some inefficiencies for both the network, in particular when multicast uses more layer 2 resources than unicast, and also has efficiency implications for hosts. Potential techniques for making DAD reliably detect and recover from duplicates might result in reduced efficiency. The impact for WiFi is shown in [I-D.desmouceaux-ipv6-mcast-wifi-power-usage].

If a node wants to "defend" its address using DAD, it has to be awake and listening on the solicited node multicast address in order to receive the DAD NS. In the low-power environments this may significantly impact the battery life of the devices.

### 2.6. Wake-up and L2 events

In mobile environments, node may roam in different parts of the network and also take "naps". The specification in [RFC4862] does not explicitly discuss this scenario, nor does DNA [RFC6059], so there is a room for ambiguity in implementation. This may either result in less robust DAD coverage (if the node does not perform the DAD again when an L2 event happens), or an excessive amount of multicast packets (when a node performs the dad every time L2 event happens and there is a lot of them moving within a segment).

Thus this item could be categorized as being either in the robustness or efficiency group of items.

## 3. Solved Issues

Some issues have been or are in the process of being solved.

### 3.1. Interaction with looped interfaces

[RFC4862] explicitly defines that the case of a physically looped back interface is not a failure: "If the solicitation is from the node itself (because the node loops back multicast packets), the solicitation does not indicate the presence of a duplicate address."

However, the practical experiences show that the measures described

in [RFC4862] are either incomplete or incorrectly implemented: a loopback on the interface causes DAD failure.

[I-D.ietf-6man-enhanced-dad] provides the solution to this issue.

### 3.2. Delays before an address can be used

Section "5.4. Duplicate Address Detection" of [RFC4862] specifies that until the DAD procedure completes, the address remains in Tentative state. In this state, any traffic to this address other than that related to DAD-related is dropped. This introduces delay between the interface getting connected to the network and an address on this interface becoming usable. For fast-moving nodes it may be a problem.

[RFC4429] introduces "Optimistic DAD" process, which addresses this. That document has some notes about potentially causing TCP RST when there is a duplicate, which can reset an existing TCP connection for the existing user of the IPv6 address. That has some overall impact on the robustness of the network and implicitly assumes that all application protocols will always retry in order to handle such an event.

## 4. Observations

Some issues we can't do much about in that they are more observations of what can be done.

### 4.1. Duplicate L2 address detection

DAD does not detect duplicate L2 addresses in all cases. Depending on the medium, it may be impossible to detect a duplicate L2 address - e.g. if this address itself is used as a determinant in order to establish the L2 connection.

### 4.2. Usage of DAD to create state

[RFC4862] in section "5.4. Duplicate Address Detection" states that DAD must be performed on all addresses. Given the potentially decentralized nature of address assignment in IPv6, this property is being used to prebuild the state in the network about the host's addresses - e.g. for "First Come First Served" security as described in section "3.2.3. Processing of Local Traffic" of [RFC6620].

If the delivery of the DAD_NS packets is unreliable or there are nodes on the segment which use the Optimistic DAD mechanism, state created purely on DAD_NS packets might be also unreliable. The

specific case of [RFC6620] solves the issue by triggering the recreation of state based on data packets as well, however it might not be possible in some scenarios.

### 4.3. No support of multi-link subnets

DAD doesn't support multi-link subnets: a multicast DAD_NS sent on one link will not be seen on the other.

[RFC6275] specifically provides one way to construct a multi-link subnet (consisting of a broadcast link and a collection of point to point tunnels). It explicitly defines the procedures for making DAD work in that topology.

[RFC4903] discusses the issues related to multi-link subnets - and given the multi-link subnets might be created in many ways, it might be prudent to keep enhancements to DAD whose sole purpose is related to multi-link subnets, to be out of scope.

One may also argue that since [RFC4861] defers the clarifications on IPv6 operation on NBMA networks to [RFC2491], it is unreasonable to expect [RFC4862] describe the operation of DAD on NBMA type links, and it is up to a link-specific document to describe such operation. (An example is cable industry, where the cable standards define it).

However, it is then unclear where to address the frequently used scenario of WiFi with blocked direct communication between the stations - whether it is supposed to be an IEEE document or IETF document ? And is there enough fundamental differences between the different NBMA models to warrant the link-specific approaches to DAD ?

### 4.4. Anycast Addresses and Duplicate Address Detection

Section 5.4 "Duplicate Address Detection" of [RFC4862] specifies that Duplicate Address Detection MUST NOT be performed on anycast addresses. This, stems from the fact that the anycast addresses are syntactically indistinguishable from unicast addresses. One can argue that this allows for misconfiguration if an address deemed to be anycast already exist on the network.

### 4.5. Implementations doing DAD once per IID

Section 5.4 of [RFC4862] mentions the implementations performing a single DAD per interface identifier, and discourages that "optimization". As the practice is emerging in the industry is to move away from the fixed interface identifiers anyhow, the necessity to perform a DAD on a per-address basis might be useful to elevate to

a requirement status.

### 4.6. Backwards compatibility and presence of the DAD proxies

While not being an issue as such, this is a reminder that the operation of DAD has to remain backwards compatible, both to remain cooperative with the existing hosts, and the potentially present DAD proxies as described in [RFC6957].

There are also various forms of sleep proxies [ECMA-393] [http://en.wikipedia.org/wiki/Bonjour_Sleep_Proxy] which perform handoffs of Neighbor Discovery protocol processing that need to be considered.

## 5. Acknowledgements

Thanks to Ole Troan for creating and curating the original list. Thanks a lot to Lorenzo Colitti, Suresh Krishnan, Hemant Singh, Hesham Soliman, Eric Vyncke, and James Woodyatt for the reviews and useful suggestions.

## 6. IANA Considerations

None.

## 7. Security Considerations

There are no additional security considerations as this document only outlines the issues observed with the current Duplicate Address Detection protocol.

## 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.

[RFC2462] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

[RFC2491] Armitage, G., Schulter, P., Jork, M., and G. Harter, "IPv6 over Non-Broadcast Multiple Access (NBMA) networks", RFC 2491, January 1999.

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

[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.

[RFC4903] Thaler, D., "Multi-Link Subnet Issues", RFC 4903, June 2007.

[RFC5227] Cheshire, S., "IPv4 Address Conflict Detection", RFC 5227, July 2008.

[RFC6059] Krishnan, S. and G. Daley, "Simple Procedures for Detecting Network Attachment in IPv6", RFC 6059, November 2010.

[RFC6275] Perkins, C., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, July 2011.

[RFC6620] Nordmark, E., Bagnulo, M., and E. Levy-Abegnoli, "FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses", RFC 6620, May 2012.

[RFC6957] Costa, F., Combes, J-M., Pougnard, X., and H. Li, "Duplicate Address Detection Proxy", RFC 6957, June 2013.

## 8.2. Informative References

[I-D.desmouceaux-ipv6-mcast-wifi-power-usage] Desmouceaux, Y., "Power consumption due to IPv6 multicast on WiFi devices", draft-desmouceaux-ipv6-mcast-wifi-power-usage-01 (work in progress), August 2014.

[I-D.ietf-6man-enhanced-dad] Asati, R., Singh, H., Beebee, W., Pignataro, C., Dart, E., and W. George, "Enhanced Duplicate Address Detection", draft-ietf-6man-enhanced-dad-13 (work in progress), February 2015.

[I-D.ietf-6man-resilient-rs] Krishnan, S., Anipko, D., and D. Thaler, "Packet loss resiliency for Router Solicitations",

draft-ietf-6man-resilient-rs-04 (work in progress), October 2014.

[I-D.vyncke-6man-mcast-not-efficient]
Vyncke, E., Thubert, P., Levy-Abegnoli, E., and A. Yourtchenko, "Why Network-Layer Multicast is Not Always Efficient At Datalink Layer", draft-vyncke-6man-mcast-not-efficient-01 (work in progress), February 2014.

## Authors' Addresses

Andrew Yourtchenko
cisco
6b de Kleetlaan
Diegem 1831
Belgium

Email: ayourtch@cisco.com

Erik Nordmark
Arista Networks
Santa Clara, CA
USA

Email: nordmark@arista.com

IPv6 maintenance Working Group (6man)
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2015

M. Zhang
S. Kapadia
L. Dong
Cisco Systems

March 9, 2015

# Improving Scalability of Switching Systems in Large Data Centers
## draft-zhang-6man-scale-large-datacenter-00

Abstract

Server virtualization has been overwhelmingly accepted especially in cloud-based data centers. Accompanied with expansion of services and technology advancements, the size of a data center has increased significantly. There could be hundreds or thousands of physical servers installed in a single large data center which implies that the number of Virtual Machines (VMs) could be in the order of millions. Effectively supporting millions of VMs with limited hardware resources, becomes a real challenge to networking vendors. This document describes a method to scale a switching system with limited hardware resources using IPv6 in large data center environments.

Table of Contents



## 1. Introduction

Server virtualization is extremely common in large data centers realized with a large number of Virtual Machines (VMs) or containers. Typically, multiple VMs share the resources of a physical server. Accompanied with expansion of services and technology advancements, the size of a data center has increased significantly. There could be hundreds or thousands of physical servers in a single large data center, which implies that the number of VMs could be in the order of millions. Such large number of VMs imposes challenges to network equipment providers on how to effectively support millions of VMs with limited hardware resources.

The CLOS based spine-leaf topology has become a defacto-standard of choice for data center deployments. A typical data center topology consists of two tiers of switches: Aggregation or spine tier and ccess/Edge or leaf tier.

Figure 1 describes a two tiers network topology in a data center cluster. S1 to Sn are spine switches. L1 to Lm are leaf switches. Every leaf switches has at least one direct connection to every spine switch. H1 to Hz are hosts/VMs attached to leaf switches directly or indirectly through L2 switches. E1 is an edge router/switch. Multiple data center clusters are interconnected by edge routers/switches.

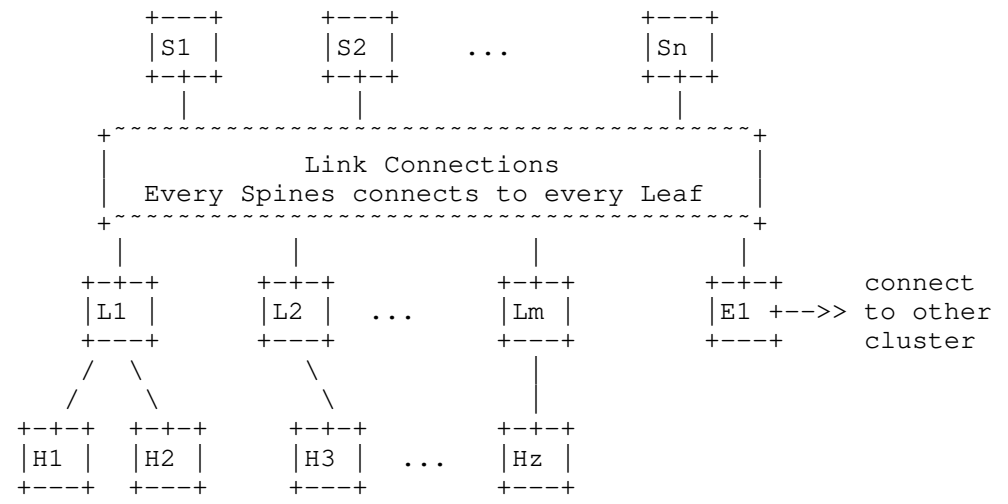



Figure 1: Typical two tier network topology in a DC cluster

Switches at the aggregation tier are large expensive entities with many ports to interconnect multiple access switches together and provide fast switching between access switches. Switches at access tier are low cost, low latency, smaller switches that are connected to physical servers for switching traffic among local servers and servers connected to other access switches through aggregation switches. For maximizing profit, low cost, and low latency ASICs are commonly selected when designing access switches, more commonly SoCs or system-on-chips. In these types of ASICs, the Layer 3 hardware Forwarding Information Base (FIB) table is typically split into two tables: 1) Host Route Table or HRT for host routes (/32 for IPv4 host routes and /128 for IPv6 host routes) that is typically implemented as a hash table; 2) Longest Prefix Match (LPM) Table for prefix routes. Due to high cost of implementing a large LPM table in ASIC either with traditional Ternary Content Addressable Memory [TCAM] or other alternatives, LPM table size in hardware is restricted to a few thousand entries (from 16k to 64k for IPv4) on access switches. Note that with the size of an IPv6 address being 4 times as long as an IPv4 address, the effective number of FIB LPM entries available for IPv6 is essentially one-fourth (or half depending on the width of the LPM entry). Note that the same tables need to be shared by all IPv4, IPv6, unicast and multicast traffic.

For years, people are looking for solutions for super scale data centers, but there has been no major break-throughs. Overlay-based [OVERLAYS] approaches using VXLAN, TRILL, FabricPath, LISP etc. have certainly allowed for separation of the tenant end-host address space from the topology address space thereby providing a level of indirection that aids scalability by reducing the requirements on the aggregation switches. However, the scale requirements on the access switches still remains high since they need to be aware of all the

tenant end-host addresses to support any-to-any communication requirement in large data centers (both East-West and North-South traffic).

Software-Defined-Network controllers gaining a lot of traction, there has been a direction to go toward a God-box like model where all the information about all the end hosts will be known. In this way, based on incoming packet, if an access switch does not know how to reach a destination, it queries the God-box and locally caches the information (the vanilla OpenFlow model). The inherent latency associated with this approach as well as the centralized model presents a single-point of failure due to which such systems will not scale beyond a point. Alternatively, the access switch can forward unknown traffic toward a set of Oracle boxes (typically one or more aggregation switches with huge tables that know all about end-hosts) which in turn takes care of forwarding traffic to the destination. As scale increases, throwing more silicon at the solution is never a good idea. The costs for building such large systems will be prohibitively high making it impractical to deploy these systems in the field.

This document describes an innovative approach to improve scalability of switching systems for large data centers with IPv6-based end-hosts or VMs. Major improvements include: 1) Reduced resource allocation from FIB tables in hardware both on access switches and almost no FIB resource allocation on aggregation switches. One single cluster can support multi-millions of hosts/VMs. 2) Eliminate L2 flooding and and L3 multicast for NDP packets between access switches 3) Reduction in the control plane processing on the access switches.

## 1.1 Specification of Requirements

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. Terminology

HRT:
Host Route Table in packet forwarding ASIC

LPM:
Longest Path Match Table in packet forwarding ASIC

Switch ID:
A unique ID for a switch in a DC cluster

Cluster ID:
A unique ID for a DC cluster in a data center

VRF:
Virtual Routing and Forwarding Instance

Switch Subnet (SS):
Subnet of a VLAN on an access switch in a data center cluster.

Switch Subnet Prefix (SSP):
An IPv6 prefix assigned to a switch subnet. It consists of Subnet Prefix, Cluster ID, and Switch ID. In a VRF, there could be one SSP per VLAN per access switch.

Aggregated Switch Subnet Prefix (ASSP):
It equals to SSP excluding Subnet ID. For better scalability, SSPs in a VRF on an access switch can be aggregated to a single ASSP. It is used for hardware programming and IPv6 forwarding.

Cluster Subnet Prefix (CSP):
Subnet prefixes for forwarding between DC clusters. It consists of Subnet Prefix and Cluster ID.

DC Cluster Prefix:
A common IPv6 prefix used by all hosts/VMs in a DC Cluster.

Subnet ID:
The ID for a subnet in a data center. It equals to Subnet Prefix excluding DC Cluster Prefix.

## 3. Large Data center Requirements

These are the major requirements for large data centers:

- Any subnet, any where, any time
- Multi-million hosts/VMs
- Any to Any communication
- VLANs (aka subnets) span across access switches
- VM Mobility
- Control plane scalability
- Easy management, trouble-shooting, debug-ability
- Scale-out model

## 4. Scaling Through Aggregation

The proposed architecture employs a distributed gateway approach at the access layer. Distributed gateway allows localization of the failure domains as well as distributed processing of ARP, DHCP etc. messages thereby allowing for a scale-out model without any restriction on host placement (any subnet, any where). Forwarding within the same subnet adheres to bridging semantics while forwarding across subnets is achieved via routing. For communication between end-hosts in different subnets below the same access switch, routing is performed locally at that access switch. For communication between end-hosts in different subnets on different access switches, routing lookups are performed both on the ingress access switch as well as the egress access switch. With distributed subnets and a distributed gateway deployment, host (/128)

addresses need to be propagated between the access switches using some IGP such as MP-BGP. As the number of hosts in the data center goes up, this would be a huge burden on the control plane in terms of advertisement of every single host address prefix. The problem is further exacerbated with the fact that a host can have multiple addresses. Our proposal indicates how this problem can be solved via flexible address assignment and intelligent control and data plane processing.

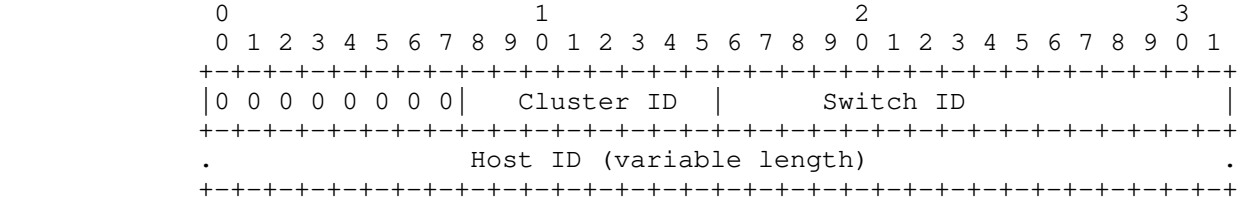
A Data Center Cluster (DCC) is a data center network that consists of a cluster of aggregation switches and access switches for switching traffic among all servers connected to the access switches in the cluster. A data center can include multiple DCCs. One unique DC Cluster Prefix (DCCP) MUST be assigned to a DCC. DC Cluster Prefix could be locally unique if the prefix is not exposed to the external internet or globally unique otherwise.

A public IPv6 address block can be procured from IANA. With the assigned address block, a service provider or enterprise can subdivide the block into multiple prefixes for their networks and Data Center Clusters (DCC). A DCCP length SHOULD be less than 64 bits. With the bits left between DCCP and IPv6 Network Prefix, many subnet prefixes can be allocated. All subnet prefixes in the DC cluster SHOULD share the common DCCP.

A new terminology is introduced in this document - Switch Subnet Prefix (SSP) which is defined as follow:

[RFC 4291] defines the 128-bit unicast IPv6 address format. It consists of two portions: Subnet Prefix and Interface ID. 64-bits Subnet Prefix is most common and highly recommended. For this scaling method, we subdivide the Interface ID in IPv6 address: N bits for Host ID, 16 bits for Switch ID, and 8 bits for Cluster ID.

Interface ID format



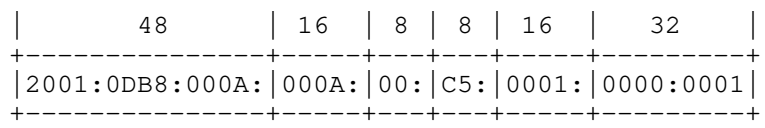

A SSP is assigned to a VLAN on an access switch. SSP includes the Subnet Prefix assigned to the VLAN, the Switch ID for the access switch, and Cluster ID for the Cluster.

Each access switch MUST has a unique Switch ID in a DC cluster. Switch ID is assigned by a user or from a management tool. Because the Switch

ID is a portion of IPv6 address for all host addresses assigned to hosts attached to the same access switch, it is recommended to assign the Switch IDs with certain characteristics, for example its location, so that it could be helpful when trouble-shooting traffic-loss issues in large data centers where millions of VMs are hosted.

Each cluster MUST have a unique Cluster ID in a data center at a campus. Cluster ID is used for routing traffic across DC clusters.

Switch Subnet Prefix Example




| | |
|---|---|
| Cluster ID: | C5 |
| Switch ID: | 1 |
| VLAN: | 100 |
| DC Cluster Prefix: | 2001:DB8:A::/48 |
| Subnet ID: | A |
| Subnet Prefix: | 2001:DB8:A:A::/64. |
| Cluster Subnet Prefix | 2001:DB8:A:A:C5::/80 |
| Switch Subnet Prefix: | 2001:DB8:A:A:C5:1::/96 |
| Host Address: | 2001:A:A:A:0:C5:1:1/128 |

In this example, the DC Cluster Prefix 2001:DB8:A::/48 is a common prefix for the cluster. From the Cluster Prefix block, there is plenty of address space (16 bits Subnet ID) available for subnet prefixes. 2001:DB8:A:A::/64 is a subnet prefix assigned to a subnet in this example that is assigned to VLAN 100. Note that for the purpose of exposition, we assume a 1:1 correspondence between a VLAN and a subnet. However, the proposal does not have any restriction if multiple subnets are assigned to the same VLAN or vice-versa. The subnet prefix is for a logical L3 interface/VLAN typically referred to as an Integrated Routing and Bridging (IRB) interface. The subnet or VLAN spans across multiple access switches thereby allowing placement of any host any where within the cluster. On each access switch, there is a Switch Subnet Prefix ( SSP) per subnet or VLAN. 2001:DB8:A:A:C5:1::/96 is the SSP for VLAN 100 on switch 1. It is combination of the Subnet Prefix, Cluster ID, and Switch ID. A Host/VM Address provisioned to a host/VM connected to this access switch MUST include the SSP associated to the VLAN on the switch. In this example, 2001:DB8:A:A:C5:1:0:1/128 is a host/VM address assigned to a host/VM connected to the access switch.

Host/VM addresses can be configured Using Stateful DHCPv6 or other network management tools. In this model, DHCPv6 is chosen for illustration. It illustrates how IPv6 host addresses are assigned from DHCPv6 server. Similar implementations can be done with other protocol/tools. Section 11 describes how address pools are configured on

DHCPv6 server and how information between switches and DHCP server is exchanged with DHCPv6 messages that allows seamless address assignment based on the proposed scheme. This makes it completely transparent to the end-user thereby alleviating the problems of address management from the network administrator.

## 5. SSP Aggregation

Typically, a routing domain is identified by a Virtual Routing and Forwarding (VRF) instance. Reachability within a VRF is achieved via regular layer-3 forwarding or routing. By default, reachability from within a VRF to outside as well as vice-versa is restricted. In that sense, a VRF provides isolation for a routing domain. A tenant can be associated with multiple VRFs and each VRF can be associated with multiple subnets/VLANs. There can be overlapping IP addressing across VRFs allowing address re-usage. To simplify implementation, reduce software processing, and improve scalability, all SSPs in a VRF on an access switch can be aggregated into a single Aggregated SSP (ASSP). Only one ASSP is needed per switch for a VRF in a DC cluster. ASSPs are employed to aid simplified processing both in the control plane as well as the data plane.

Typically, for every subnet instantiated on an access switch, a corresponding subnet prefix needs to be installed in the LPM that points to the glean adjacency. With ASSP, only a single entry needs to be installed in the LPM irrespective of the number of subnets that are instantiated on the access switch. In addition, the same benefit is leveraged at the remote access switches where there needs to be a single ASSP installed for every other access switch independent of what subnets are instantiated at the remote switches. More details of how this FIB programming is achieved are presented in the next section.

ASSP entries on an access switch MUST be distributed to all other access switches in a cluster through a routing protocol such as BGP. When an ASSP entry is learned through IGP/BGP protocol, a LPM entry SHOULD be installed. Because of better scalability in large data center environment (BGP Reflector Router can be used to reduce number of peers a BGP node communicates with), BGP is recommended for this forwarding model. In this document, we describe how BGP can be used for ASSP and CSP distribution. A new BGP Opaque Extended community is specified in section 10 for this solution.

As mentioned earlier, in modern data centers, overlay networks are typically used for forwarding data traffic between access switches. On aggregation switches, a very small number of FIB entries are needed for underlay reachability since the aggregation switches are oblivious to the tenant host addresses. So aggregation switching platforms can be designed to be simple, low latency, high port density, and low cost.

ASSP entries programmed in LPM table are for forwarding data traffic between access switches. The rewrite information in the corresponding next-hop (or Adjacency) entry SHOULD include information to forward

packets to the egress access switch corresponding to the Switch ID.

One ASSP local FIB CAM entry is also needed. The rewrite information in the corresponding next-hop (or Adjacency) entry SHOULD include information to punt packet to local processor. This local FIB CAM entry is used for triggering address resolution if a destined host/VM is not in the IPv6 Neighbor Cache (equivalent to a glean entry). Host/VM addresses (/128) discovered through IPv6 ND protocol are installed in Host Route table (HRT) on the access switch and only on that access switch. Host routes learned through routing protocol MUST NOT be programmed HRT table in hardware. Note exception can occur if a VM moves across access switch boundary. For VM moves across access switch boundary, special handlings are required that will be discussed in a different draft for VM Mobility.

A IPv6 unicast data packet from a host/VM connected to an ingress switch destined to another host on an egress switch is forwarded in the following steps: 1) It arrives at the ingress switch; 2) A L3 lookup in FIB (LPM) CAM table hits an entry because the packets destination address includes the Switch Subnet Prefix; 3) The packet is forwarded to the egress switch based on the FIB CAM entry and the corresponding Adjacency entry; 4) The packet is forwarded to its destined host by the egress switch.

For forwarding packets outside of the DC Cluster, a Default route ::/0 SHOULD be installed in FIB CAM that routes packets to one of DC edge routers/switches that provides reachability both to other data center sites as well as the external world (Internet).

To summarize in this forwarding model, only local Host/VM routes are installed in HRT table. That greatly reduces the number of HRT table entries required at an access switch. ASSP routes are installed in LPM table for forwarding traffic between access switches. Because of ASSPs are independent of subnet/VLANs, the total number of LPM entries required are greatly reduced. These reduced requirements on the HRT and LPBM on the access switches allow supporting very large number of VMs with much smaller hardware FIB tables.

Similar forwarding model SHOULD be implemented in software. For example, if special mark is used as discussed in section 6, when forwarding an IPv6 packet in an SSP enabled VRF, the SSP subnet bits can be masked with 0s when doing lookup in software FIB. If it results in a match with an ASSP entry, the packet will be forwarded to the egress access switch with the adjacency attached to the ASSP.

## 6. Programming in FIB CAM with Special Mask

Typically, FIB lookup requires a longest prefix match (LPM) for which a CAM is utilized. CAM in ASIC is implemented with Value bits and mask bits for each of its entries. Value bits are the value (0 or 1) of the bits in the memory for L3 forwarding lookup against a lookup key in the CAM table that includes typically the destination address of a data

packet to be forwarded (the lookup key is typically vpn-id (corresponding to the VRF, destination-IP). The mask bits are used to include or exclude each bit in the value field of a CAM entry when deciding if a match has occurred or not. Mask bit=1, or mask-in, means include the value bit and mask bit=0, or mask-out, means exclude the value bit or its a DONT-CARE (corresponding value bit can be 1 or 0).

When programming the FIB CAM for all Switch Subnet Prefixes from an ACCESS switch, only one entry is installed in FIB CAM per destination ACCESS switch by masking in all DC Cluster Prefix bits, masking out all bits after DC Cluster Prefix and before the Cluster ID bits, and then masking in both Cluster ID bits and ACCESS ID bits and masking out remaining bits.

For example,

DC Cluster Prefix:  2001:0DB8:000A::/48
Cluster ID: 0xC5
ACCESS ID in hex: 0x1234

FIB CAM programming
   Value:  2001:0DB8:000A:0000:00C5:1234:0000:0000
   Mask:  FFFF:FFFF:FFFF:0000:00FF:FFFF:0000:0000

With one such FIB CAM entry, it can match all Switch Subnet Prefixes that includes the DC Cluster Prefix 2001:0DB8:000A::/48, Cluster ID 0xC5 and Switch ID 0x1234 no matter what values on those bits between DC Cluster Prefix and the Cluster ID. That means only single FIB CAM entry is needed for all packets destined to hosts connected to a switch no matter what subnet prefixes are configured on VLANs on that switch. On a given switch, one FIB CAM entry is required for each of other access switches in the DC Cluster.

In case the LPM is not implemented as a true CAM but instead as an algorithmic CAM as is the case with some of the ASICs, an alternative approach can be employed. That is to set all subnet bits to 0s when programming an ASSP entry in LPM table. Subnet bits SHOULD be cleared when doing lookup in LPM table. This approach requires certain changes in lookup logic of the ASIC.

Note that the above explanation applies on a per VRF basis since the FIB lookup is always based on (VRF, Destination-IP). For example, in a data center with 100 access switches, if a VRF spans 10 access switches, then the number of LPM entries on those 10 access switches for this VRF is equal to 10 (1 local and 9, one for each of the remote switches). Section 11 provides additional details on scalability in terms of the number of entries required for supporting a large multi-tenant data center with millions of VMs.

## 7. VM Mobility

VM mobility will be discussed in a separate IETF draft.

## 8. Scaling Neighbor Discovery

Another major issue with the traditional forwarding model is the scalability of processing the Neighbor Discovery protocol (NDP) messages. In a data center cluster with large number of VLANs and as many of the VLANs span across multiple access switches, the volume of NDP messages handled by software on an access switch could be huge that can easily overwhelm the CPU. On the other hand, the large number of entries in neighbor cache on an access switch could causes HRT table overflow.

In our proposed forwarding model, Neighbor Discovery can be distributed to access switches as described below. Please note all following descriptions in this section only apply to ND operation for global unicast target. No ND operation change is required for Link-local target.

All NDP messages from host/VMs are restricted to the local access switch.

Multicast NDP messages are flooded to all local switch ports on a VLAN and also copied to local CPU. It SHOULD NOT be sent on link(s) connected to aggregation switches.

When a multicast NS message is received, if its target matches with the local ASSP, then it can be ignored because the host/VM SHOULD reply to the NS since the destination is also locally attached to the access switch; otherwise, a unicast NA message MUST be sent by the switch with link-layer address equals to the switch's MAC (aka Router MAC).

When an unicast data packet is received, if the destination address belongs to a remote switch, it will match the ASSP for the remote switch in FIB table and be forwarded to the remote switch. On the remote switch, if that destined host/VM is not discovered yet, the data packet will be punt to the CPU and a ND will be triggered for host discovery in software.

Distributed ND model can reduce software processing in CPU substantially. It also takes much less space in hardware HRT table. Most importantly there is no flooding both in L2 and L3. Flooding is a major concern in large data centers so it SHOULD be avoided as much as possible.

A subnet prefix and a unique address are configured on a L3 logical interface on access switch. When the L3 logical interface has member ports on multiple switches, the same subnet prefix and the address MUST be configured on the L3 logical interface on all those switches. ND operation on hosts/VMs remains the same without any change.

## 9. DHCPv6

This section describes the host address assignment model with DHCPv6 protocol. Similar implementations can be done with other protocols and management tools.

DHCPv6 Relay Agent [RFC 3315] SHOULD be supported on access switches for this address assignment proposal. [draft-ietf-dhc-topo-conf-04] specifies recommendations on real DHCPv6 Relay Agent deployments. For the forwarding model described in this document, the method of using link-address as described in section 3.2 of [draft-ietf-dhc-topo-conf-04] SHOULD be implemented as follows:

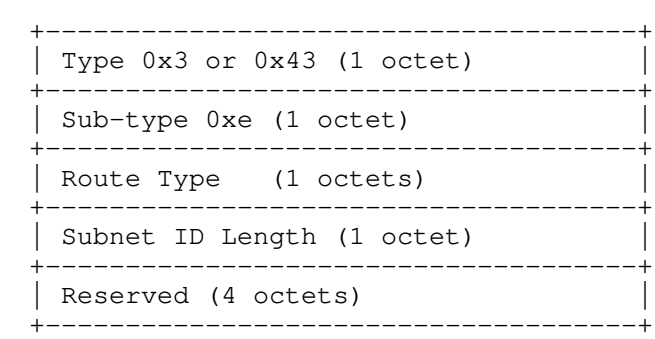
The Switch Subnet Prefix (SSP) for the subnet on the switch SHOULD be used as link-address in Relay-Forward message sent from switch. On DHCPv6 server, the link-address is used to identify the link. A prefix or address range should be configured on DHCPv6 server for the link. The prefix or address range MUST match with the SSP on the switch. By doing these, it is guaranteed that addresses assigned by DHCPv6 server always include the SSP for the interface on the switch.

The number of SSP address pools could be very large on the DHCP server. This can be alleviated by employing a cluster of DHCP servers to ensure effective load distribution of client DHCPv6 requests.

## 10. BGP

As mentioned earlier, ASSP entries are redistributed to all access switches through BGP. ASSP entries learned from BGP are inserted in RIB. They will be used for FIB CAM programming in hardware and IPv6 Forwarding in software.

In this document, we define a BGP opaque extended community that can be attached to BGP UPDATE messages to indicate the type of routes that are advertised in the BGP UPDATE messages. This is the IPv6 Route Type Community [RFC4360] with the following encoding:




Type Field:
The value of the high-order octet of this Opaque Extended Community is 0x03 or 0x43. The value of the low-order octet of the extended type

field for this community is 0x0E(or another value allocated by IANA).

Value Field:
The 6 octet Value field contains three distinct sub-fields, described below:

The route type sub-field defines the type of IPv6 routes carried in this BGP message. The following values are defined:

1: ASSP_Route indicates that the routes carried in this BGP Update message are ASSP route
2: CSP_Route indicates that the routes carried in this BGP Update message are CSP routes

The Subnet ID Length specifies the number of bits in an ASSP route. Those bits can be ignored in the FIB look up either with special mask when FIB lookup CAM is used or an alternative way as described in section 5. This field is only used when the route type is ASSP_Route.

The 4 octet reserved field is for future use.

The IPv6 Route Type Community does not need to be carried in the BGP Withdraw messages.

All operations SHOULD follow [RFC4360]. There is no exception for this forwarding model.

## 11. Scalability

With this innovative forwarding model, the scalability of data center switching system is improved significantly while still allowing any-to-any communications between all hosts, and no restriction on host placement or host mobility.

FIB TCAM utilization on an access switch becomes independent of number of VLANs/subnets instantiated on that switch.

It is important to note that the number of host prefix routes (/128) only depends on the number of VMs that are local to an access switch. Network administrator can add as many as access switches with the same network design and would never worry about running out of FIB HRT resources. This greatly simplifies network design for large data centers

The total number of VMs can be supported in a data center cluster can be calculated as the following (assuming single VRF):

Number of LPM entries:
Only one LPM entry per access switch is required for local ASSP. The total number of LPM entries on an access switch is equivalent to the total number of access switches in a DC cluster plus 1 (for the default route).

Number of HRT entries:
There will be one HRT entry for each directly connected host/VM.

Scalability Calculation

H: max number of HRT entries
V: Number of VMs/port
P: number of ports/access switch

H = V x P

For example,
48 ports/access switch, 128 VMs/port
H = 48 x 128 = 6k HRT entries/access switch

T: total number of hosts/VMs
L: number of access switches

T = H x L
xample: 200 access switches
1.2 Million (6k x 200) VMs can be supported in a large data center cluster.

## 12. DC edge router/switch

Multiple data center clusters can be interconnected with DC edge routers/switches. The same subnet can span across multiple data center clusters. While each subnet has a unique subnet prefix, each cluster in which that subnet extends has a unique cluster subnet prefix. This will be advertised over BGP to the edge routers, which in turn will attract traffic for hosts that are part of that subnet in a given cluster. Again, procedure to handle host mobility across clusters will be described separately in a different draft.

### 12.1 DC Cluster Interconnect

This section describes a way to support VLAN across DC clusters for this forwarding model.

Subnet Prefixes SHOULD be advertised by routing protocol within a DC Cluster, but subnet prefixes SHOULD NOT be installed in hardware FIB table. On a DC edge router/switch, Cluster Subnet Prefixes (CSP) can be configured or auto-generated if SSP is enabled. CSP is special prefix to be used at DC edge router/switch to forward traffic between directly

connected DC clusters. Please refer to section 4 for CSP definition and example. There SHOULD be one CSP per subnet.

A CSP SHOULD be advertised through a routing protocol between DC edge router/switch that connects the DC Clusters. In section 10, special BGP option is defined for advertising CSP routes. CSP routes SHOULD not be advertised into a DC cluster.

CSP route message SHOULD be handled as follow:

On CSP originating DC edge router/switch, CSP SHOULD NOT be installed in FIB table in hardware. On the receiving DC edge router/switch, CSP SHOULD be installed in FIB table in hardware. All bits between DCCP and Cluster ID MUST be masked out if the special mask scheme can be implemented, or set those bits to 0s if FIB key mask is not supported.

Because CSPs consume FIB CAM space, user SHOULD determine if there is enough FIB CAM resource on DC edge router/switch before enabling this feature.

## 13. Multiple VRFs and Multiple Tenancies

For flexibility to users, an implementation can let user to enable/disable this feature at VRF level on one or more access switches. When it is enabled in a VRF, all functionalities described in this document SHOULD be applied to that VRF on all those access switches. No behavior changes SHOULD happen in other VRFs without this feature enabled.

Multi-tenancy can be supported by employing multiple VRFs. A tenant can be allocated VRFs.

### 13.1 Resource Allocation and Scalability with VRFs

For supporting more VRFs in a DC cluster, a DC network administrator can enable this feature for a VRF only on a few access switches in the cluster. The max number of VRFs can be calculated with this formula:

Scalability Calculation
L: Number of LPM entries
V: number of VRFs
P: number of ACCESSs per VRF (average)

$$L = V \times (P + 1) \quad \text{or}$$
$$V = L/(P + 1)$$

Example
8k LPM entries available per access switch and on average 9 ACCESSs are allocated per VRF.
Number of VRFs that can be supported: $V = 8000/(9 + 1) = 800$

More VRFs can be supported if the number of access switches per VRF is decreased.

To support a large number of VRFs or tenants, larger LPM tables MAY be required. That SHOULD be considered at ASIC design phase.

## 14. Security

No new security threat is expected to be imposed by this proposal.

## 15. References

[TCAM] Soraya Kasnavi and Vincent C. Gaudet, Paul Berube and Jose Nelson Amaral, A Hardware-Based Longest Prefix Matching Scheme for TCAMs IEEE, 2005

[OVERLAYS] S. Hooda, S. Kapadia, P. Krishnan, Using TRILL, FabricPath, and VXLAN: Designing Massively Scalable Data Centers (MSDC) with Overlays, ISBN-978-1587143939, 2014

[RFC 4291] IP Version 6 Addressing Architecture

[RFC 4861] Neighbor Discovery for IP version 6 (IPv6)

[RFC 3315] Dynamic Host Configuration Protocol for IPv6 (DHCPv6)

[draft-ietf-dhc-topo-conf-04] Customizing DHCP Configuration on the Basis of Network Topology

[RFC 4271] A Border Gateway Protocol 4 (BGP-4)

[RFC4360] BGP Extended Community Attribute

## Authors' Addresses

Ming Zhang
Cisco Systems
170 West Tasman Dr
San Jose, CA 95134
USA

Phone: +1 408 853 2419
EMail: mzhang@cisco.com

Shyam Kapadia
Cisco Systems
170 West Tasman Dr

San Jose, CA 95134
USA

Phone: +1 408 527 8228
EMail: shkapadi@cisco.com

Liqin Dong
Cisco Systems
170 West Tasman Dr
San Jose, CA 95134
USA

Phone: +1 408 527 1532
EMail: liqin@cisco.com