

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 27, 2022

J. Dong  
Z. Li  
Huawei Technologies  
C. Xie  
C. Ma  
China Telecom  
G. Mishra  
Verizon Inc.  
October 24, 2021

Carrying Virtual Transport Network (VTN) Identifier in IPv6 Extension  
Header  
draft-dong-6man-enhanced-vpn-vtn-id-06

Abstract

Virtual Private Networks (VPNs) provide different customers with logically separated connectivity over a common network infrastructure. With the introduction and evolvement of 5G and other network scenarios, some existing or new customers may require connectivity services with advanced characteristics comparing to traditional VPNs. Such kind of network service is called enhanced VPNs (VPN+).

A Virtual Transport Network (VTN) is a virtual underlay network which consists of a set of dedicated or shared network resources allocated from the physical underlay network, and is associated with a customized logical network topology. VPN+ services can be delivered by mapping one or a group of overlay VPNs to the appropriate VTNs as the virtual underlay. In packet forwarding, some fields in the data packet needs to be used to identify the VTN the packet belongs to, so that the VTN-specific processing can be performed on each node the packet traverses.

This document proposes a new Hop-by-Hop option of IPv6 extension header to carry the VTN Resource ID, which is used to identify the set of network resources allocated to a VTN for packet processing. The procedure for processing the VTN option is also specified.

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 <https://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 27, 2022.

#### Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

|   |   |
|---|---|
| 1. Introduction . . . . .                             | 2 |
| 1.1. Requirements Language . . . . .                  | 4 |
| 2. New IPv6 Extension Header Option for VTN . . . . . | 4 |
| 3. Procedures . . . . .                               | 5 |
| 3.1. VTN Option Insertion . . . . .                   | 5 |
| 3.2. VTN based Packet Forwarding . . . . .            | 5 |
| 4. Operational Considerations . . . . .               | 6 |
| 5. IANA Considerations . . . . .                      | 6 |
| 6. Security Considerations . . . . .                  | 6 |
| 7. Contributors . . . . .                             | 6 |
| 8. Acknowledgements . . . . .                         | 7 |
| 9. References . . . . .                               | 7 |
| 9.1. Normative References . . . . .                   | 7 |
| 9.2. Informative References . . . . .                 | 7 |
| Authors' Addresses . . . . .                          | 8 |

#### 1. Introduction

Virtual Private Networks (VPNs) provide different customers with logically isolated connectivity over a common network infrastructure. With the introduction and evolvement of 5G and other network

scenarios, some existing or new customers may require connectivity services with advanced characteristics comparing to traditional VPNs, such as resource isolation from other services or guaranteed performance. Such kind of network service is called enhanced VPN (VPN+). VPN+ service requires the coordination and integration between the overlay VPNs and the network characteristics of the underlay.

[I-D.ietf-teas-enhanced-vpn] describes a framework and the candidate component technologies for providing VPN+ services. It also introduces the concept of Virtual Transport Network (VTN). A Virtual Transport Network (VTN) is a virtual underlay network which consists of a set of dedicated or shared network resources allocated from the physical underlay network, and is associated with a customized logical network topology. VPN+ services can be delivered by mapping one or a group of overlay VPNs to the appropriate VTNs as the underlay, so as to provide the network characteristics required by the customers. In packet forwarding, traffic of different VPN+ services need to be processed separately based on the network resources and the logical topology associated with the corresponding VTN.

[I-D.dong-teas-enhanced-vpn-vtn-scalability] describes the scalability considerations and the possible optimizations for providing a relatively large number of VTNs for VPN+ services. One approach to improve the data plane scalability of VTN is to introduce a dedicated VTN Resource Identifier (VTN Resource ID) in the data packet to identify the set of network resources allocated to a VTN, so that VTN-specific packet processing can be performed using that set of resources, which avoids the possible resource competition with services in other VTNs. This is called Resource Independent (RI) VTN. A VTN Resource ID represents a subset of the resources (e.g. bandwidth, buffer and queuing resources) allocated on a given set of links and nodes which constitute a logical network topology. The logical topology associated with a VTN could be defined using mechanisms such as Multi-Topology [RFC4915], [RFC5120] or Flex-Algo [I-D.ietf-lsr-flex-algo], etc.

This document proposes a mechanism to carry the VTN resource ID in a new Hop-by-Hop option of IPv6 extension header [RFC8200] of IPv6 packet, so that on each network node along the packet forwarding path, the VTN option in the packet is parsed, and the obtained VTN Resource ID is used to instruct the network node to use the set of network resources allocated to the corresponding VTN to process and forward the packet. The procedure for processing the VTN Resource ID is also specified. This provides a scalable solution to support a relatively large number of VTNs in an IPv6 network.

### 1.1. 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 BCP14 RFC 2119 [RFC2119] RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 2. New IPv6 Extension Header Option for VTN

A new Hop-by-Hop option type "VTN" is defined to carry the VTN related Identifier in an IPv6 packet. Its format is shown as below:

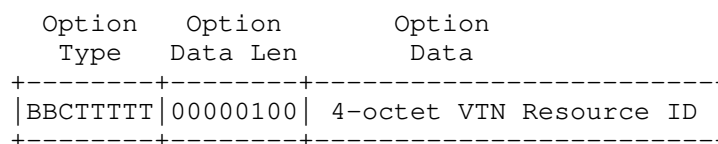


Figure 1. The format of VTN Option

**Option Type:** 8-bit identifier of the type of option. The type of VTN option is to be assigned by IANA. The highest-order bits of the type field are defined as below:

- o BB 00 The highest-order 2 bits are set to 00 to indicate that a node which does not recognize this type will skip over it and continue processing the header.
- o C 0 The third highest-order bit are set to 0 to indicate this option does not change en route.

**Opt Data Len:** 8-bit unsigned integer indicates the length of the option Data field of this option, in octets. The value of Opt Data Len of VTN option SHOULD be set to 4.

**VTN Resource ID:** 4-octet identifier which uniquely identifies the set of network resources allocated to a VTN.

**Editor's note:** The length of the VTN Resource ID is defined as 4-octet in correspondence to the 4-octet Single Network Slice Selection Assistance Information (S-NSSAI) defined in 3GPP [TS23501].

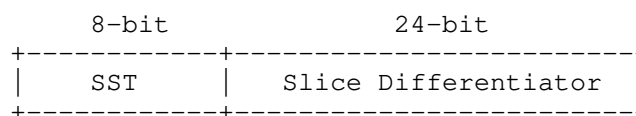


Figure 2. The format of S-NSSAI

### 3. Procedures

As the VTN option needs to be processed by each node along the path for VTN-specific forwarding, it SHOULD be carried in IPv6 Hop-by-Hop options header when the Hop-by-Hop options header can be either processed or ignored in forwarding plane by all the nodes along the path.

#### 3.1. VTN Option Insertion

When an ingress node of an IPv6 domain receives a packet, according to the traffic classification or mapping policy, the packet is steered into one of the VTNs in the network, then the packet SHOULD be encapsulated in an outer IPv6 header, and the Resource ID of the VTN which the packet is mapped to SHOULD be carried in the VTN option of the Hop-by-Hop options header associated with the outer IPv6 header.

#### 3.2. VTN based Packet Forwarding

On receipt of a packet with the VTN option, each network node which can process the VTN option in fast path SHOULD use the VTN Resource ID to determine the set of local network resources allocated to the VTN for packet processing. The packet forwarding behavior is based on both the destination IP address and the VTN Resource ID. More specifically, the destination IP address is used to determine the next-hop and the outgoing interface, and VTN Resource ID is used to determine the set of network resources on the outgoing interface which are reserved to the VTN for processing and sending the packet. The Traffic Class field of the outer IPv6 header MAY be used to provide Diffserv treatment for packets which belong to the same VTN. The egress node of the IPv6 domain SHOULD decapsulate the outer IPv6 header which includes the VTN option.

In the forwarding plane, there can be different approaches of partitioning the local network resources and allocating them to different VTNs. For example, on one physical interface, a subset of the forwarding plane resources (e.g. the bandwidth and the associated buffer and queuing resources) can be allocated to a particular VTN and represented as a virtual sub-interface with reserved bandwidth resource. In packet forwarding, the IPv6 destination address of the received packet is used to identify the next-hop and the outgoing layer-3 interface, and the VTN Resource ID is used to further identify the virtual sub-interface which is associated with the VTN on the outgoing interface.

Network nodes which do not support the processing of Hop-by-Hop options header SHOULD ignore the Hop-by-Hop options header and

forward the packet only based on the destination IP address. Network nodes which support Hop-by-Hop Options header, but do not support the VTN option SHOULD ignore the VTN option and continue to forward the packet based on the destination IP address and MAY also based on the rest of the Hop-by-Hop Options.

#### 4. Operational Considerations

As described in [RFC8200], network nodes may be configured to ignore the Hop-by-Hop Options header, and in some implementations a packet containing a Hop-by-Hop Options header may be dropped or assigned to a slow processing path. The proposed modification to the processing of IPv6 Hop-by-Hop options header is specified in [I-D.hinden-6man-hbh-processing]. Operator needs to make sure that all the network nodes involved in a VTN can either process Hop-by-Hop Options header in the fast path, or ignore the Hop-by-Hop Option header. Since a VTN is associated with a logical network topology, it is practical to ensure that all the network nodes involved in that logical topology support the processing of the HBH options header and the VTN option. In other word, packets steered into a VTN MUST NOT be dropped due to the existence of the Hop-by-Hop Options header. It is RECOMMENDED to configure all the network nodes involved in a VTN to process the Hop-by-Hop Options header and the VTN option if there is a nob for this.

#### 5. IANA Considerations

This document requests IANA to assign a new option type from "Destination Options and Hop-by-Hop Options" registry.

| Value | Description | Reference     |
|-------|-------------|---------------|
| TBD   | VTN Option  | this document |

#### 6. Security Considerations

The security considerations with IPv6 Hop-by-Hop options header are described in [RFC8200], [RFC7045] and [I-D.hinden-6man-hbh-processing]. This document introduces a new IPv6 Hop-by-Hop option which is either processed in the fast path or ignored by network nodes, thus it does not introduce additional security issues.

#### 7. Contributors

Zhibo Hu  
Email: huzhibo@huawei.com

Lei Bao  
Email: baolei7@huawei.com

## 8. Acknowledgements

The authors would like to thank Juhua Xu, James Guichard, Joel Halpern and Tom Petch for their review and valuable comments.

## 9. References

### 9.1. Normative References

- [I-D.ietf-teas-enhanced-vpn]  
Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Network (VPN+) Services", draft-ietf-teas-enhanced-vpn-08 (work in progress), July 2021.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

### 9.2. Informative References

- [I-D.dong-teas-enhanced-vpn-vtn-scalability]  
Dong, J., Li, Z., Gong, L., Yang, G., Guichard, J. N., Mishra, G., and F. Qin, "Scalability Considerations for Enhanced VPN (VPN+)", draft-dong-teas-enhanced-vpn-vtn-scalability-03 (work in progress), July 2021.
- [I-D.hinden-6man-hbh-processing]  
Hinden, R. M. and G. Fairhurst, "IPv6 Hop-by-Hop Options Processing Procedures", draft-hinden-6man-hbh-processing-01 (work in progress), June 2021.

- [I-D.ietf-lsr-flex-algo]  
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and  
A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-  
algo-17 (work in progress), July 2021.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P.  
Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF",  
RFC 4915, DOI 10.17487/RFC4915, June 2007,  
<<https://www.rfc-editor.org/info/rfc4915>>.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi  
Topology (MT) Routing in Intermediate System to  
Intermediate Systems (IS-IS)", RFC 5120,  
DOI 10.17487/RFC5120, February 2008,  
<<https://www.rfc-editor.org/info/rfc5120>>.
- [RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing  
of IPv6 Extension Headers", RFC 7045,  
DOI 10.17487/RFC7045, December 2013,  
<<https://www.rfc-editor.org/info/rfc7045>>.
- [TS23501] "3GPP TS23.501", 2016,  
<[https://portal.3gpp.org/desktopmodules/Specifications/  
SpecificationDetails.aspx?specificationId=3144](https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144)>.

#### Authors' Addresses

Jie Dong  
Huawei Technologies  
Huawei Campus, No. 156 Beiqing Road  
Beijing 100095  
China

Email: [jie.dong@huawei.com](mailto:jie.dong@huawei.com)

Zhenbin Li  
Huawei Technologies  
Huawei Campus, No. 156 Beiqing Road  
Beijing 100095  
China

Email: [lizhenbin@huawei.com](mailto:lizhenbin@huawei.com)



Chongfeng Xie  
China Telecom  
China Telecom Beijing Information Science & Technology, Beiqijia  
Beijing 102209  
China

Email: xiechf@chinatelecom.cn

Chenhao Ma  
China Telecom  
China Telecom Beijing Information Science & Technology, Beiqijia  
Beijing 102209  
China

Email: machh@chinatelecom.cn

Gyan Mishra  
Verizon Inc.

Email: gyan.s.mishra@verizon.com

Network Working Group  
Internet Draft  
Intended status: Standard  
Expires: September 7, 2022

L. Dunbar  
J. Kaippallimalil  
Futurewei

March 7, 2022

IPv6 Solution for 5G Edge Computing Sticky Service  
draft-dunbar-6man-5g-edge-compute-sticky-service-06

## Abstract

This draft describes the IPv6-based solutions that can stick an application flow originated from a mobile device to the same ANYCAST server location when the mobile device moves from one 5G cell site to another.

## Status of this Memo

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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

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

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

This Internet-Draft will expire on April 7, 2021.

#### Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the  
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal  
Provisions Relating to IETF Documents  
(<http://trustee.ietf.org/license-info>) in effect on the date of  
publication of this document. Please review these documents  
carefully, as they describe your rights and restrictions with  
respect to this document. Code Components extracted from this  
document must include Simplified BSD License text as described  
in Section 4.e of the Trust Legal Provisions and are provided  
without warranty as described in the Simplified BSD License.

#### Table of Contents

|  |    |
|--|----|
| 1. Introduction.....   | 3  |
| 1.1. 5G Edge Computing Background.....                                   | 3  |
| 1.2. 5G Edge Computing Network Properties.....                           | 4  |
| 1.3. Problem #1: Discovery of Edge Application Server.....               | 5  |
| 1.4. Problem #2: sticking to original App Server.....                    | 6  |
| 2. Conventions used in this document.....                                | 7  |
| 3. Stick a Flow to an ANYCAST Server.....                                | 9  |
| 4. Sticky flow for QUIC based Applications.....                          | 9  |
| 5. Other Solutions within a Limited Domain.....                          | 10 |
| 5.1. Use Case of 5G Edge Computing in a limited domain....               | 10 |
| 5.2. End Node Based Sticky Service Solution.....                         | 10 |
| 5.2.1. Edge Controller Based Solution.....                               | 11 |
| 5.3. Sticky Egress Address Discovery.....                                | 12 |
| 5.4. Sticky-Dst-SubTLV in Destination Extension Header....               | 12 |
| 5.5. Processing at the Ingress router.....                               | 13 |
| 6. Tunnel based Sticky Service Solution.....                             | 13 |
| 6.1. Desired functions by the Network Controller.....                    | 14 |
| 6.2. Ingress and Egress Routers Processing Behavior.....                 | 14 |
| 6.3. A Solution without the Communication with 5G system..               | 16 |
| 6.4. A Solution that depends on the communication with 5G<br>system..... | 16 |
| 7. Expanding APN6 for Sticky Service information.....                    | 17 |
| 7.1. Sticky Service ID encoded in the Application-aware ID               | 17 |

|  |    |
|--|----|
| 7.2. Sticky Service Sub-TLV encoded in APN6 Service-para option..... | 18 |
| 8. Manageability Considerations.....                                 | 18 |
| 9. Security Considerations.....                                      | 18 |
| 10. IANA Considerations.....   | 18 |
| 11. References.....  | 18 |
| 11.1. Normative References.....                                      | 18 |
| 11.2. Informative References.....                                    | 19 |
| 12. Acknowledgments.....   | 20 |

## 1. Introduction

### 1.1. 5G Edge Computing Background

As described in [5G-EC-Metrics], one application in 5G Edge Computing environment can have multiple application servers hosted in different Edge Computing data centers close in proximity. Those Edge Computing (mini) data centers are usually very close to, or co-located with, 5G base stations, to minimize latency and optimize the performances.

When a mobile device sends packets using the destination address from a DNS reply or its own cache, the packets are carried by a GTP tunnel from the 5G eNB to the 5G UPF-PSA (User Plan Function - PDU Session Anchor). The UPF-PSA decapsulates the 5G GTP outer header and forwards the packets from the mobile devices to the Ingress router of the Edge Computing (EC) Local Data Network (LDN). The LDN for 5G EC, the IP Networks, is responsible for forwarding the packets to the intended destinations.

When the mobile device moves out of coverage of its current gNB (next-generation Node B) (gNB1), handover procedures are initiated, and the 5G SMF (Session Management Function) selects a new UPF-PSA. The standard handover procedures are described in 3GPP TS 23.501 and TS 23.502. When the handover process is complete, the mobile device might be anchored to a new UPF-PSA. 5G Session Management function (SMF) may maintain a path from the old UPF to the new UPF for a short period of time for SSC [Session and Service Continuity] mode 3 to make the handover process more seamless.

## 1.2. 5G Edge Computing Network Properties

In this document, 5G Edge Computing Network refers to multiple Local IP Data Networks (LDN) in one region that interconnect the Edge Computing mini-data centers. Those IP LDN networks are the N6 interfaces from 3GPP 5G perspective.

The ingress routers to the 5G Edge Computing Network are directly connected to 5G UPFs. The egress routers to the 5G Edge Computing Network are the routers that have a direct link to the Edge Computing servers. The servers and the egress routers are co-located. Some of those mini Edge Computing Data centers may have Virtual switches or Top of Rack switches between the egress routers and the servers. But transmission delay between the egress routers and the Edge Computing servers is very small, which is considered negligible in this document.

When multiple Edge Computing Servers attached to one App Layer Load Balancer, only the App Layer Load Balancer address is visible to the 5G Edge Computing Network. How the App Layer Load balancer manages the individual servers is out of the scope of the document.

The Edge Computer Services are registered services that need to utilize the network topology and balance among multiple mini Edge Computing Data Centers with the same ANYCAST address. Majority services are not registered 5G Edge Computing Services.

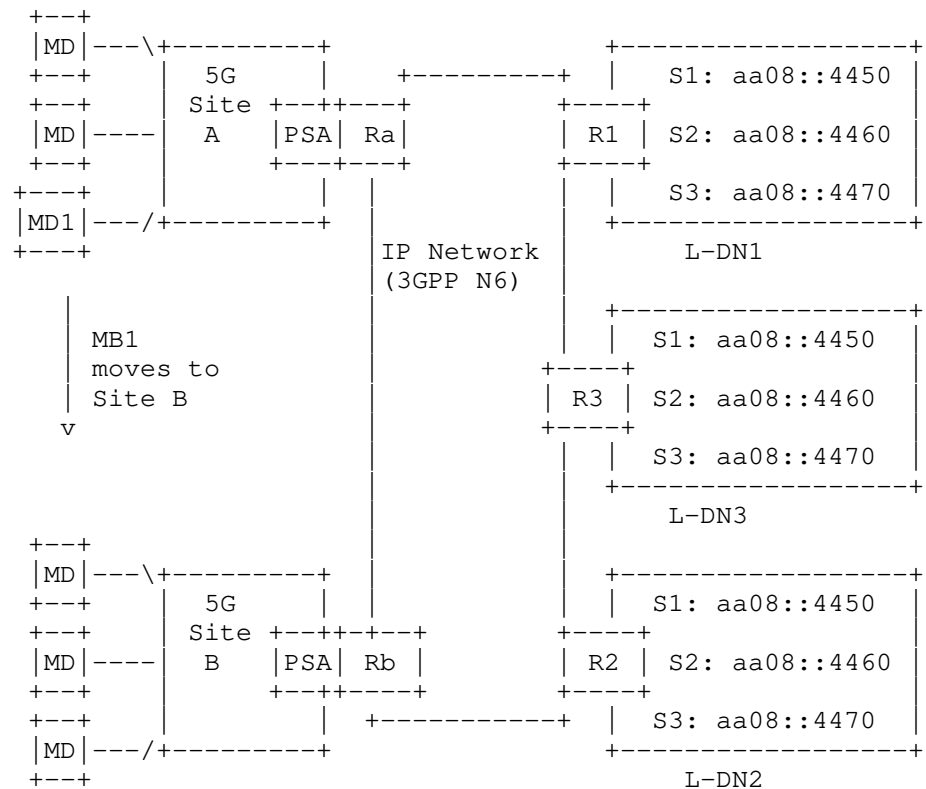


Figure 1: App Servers in different edge DCs

1.3. Problem #1: Discovery of Edge Application Server

Key Issue #1 identified by 3GPP Edge Computing Study [TR 23.748] is that one application service might be served by multiple Edge Application Servers typically deployed in different sites. These multiple Edge Application Server instances that host same content or service may use a single IP address (anycast address) or different IP addresses.

Key Issue #2 identified by 3GPP Edge Computing Study [TR 23.748] is Edge server relocation.

Application Server discovery and relocation can be achieved by running IGP/BGP routing protocols among the routers in LDN.

Increasingly, ANYCAST is used extensively by various application providers because it is possible to dynamically load balance across multiple locations of the same address based on network conditions. When multiple servers in different locations have the same IP address (ANYCAST), the routers see multiple paths to the IP address. The IGP/BGP routing protocols can inform all the nodes where the servers are and when servers move to new locations.

Application Server location selection using Anycast address leverages the proximity information present in the network routing layer and eliminates the single point of failure and bottleneck at the DNS resolvers and application layer load balancers. Another benefit of using ANYCAST address is removing the dependency on mobile devices that use their cached IP addresses instead of querying DNS when they move to a new location.

However, having multiple locations for the same ANYCAST address in the 5G Edge Computing environment can be problematic because all those edge computing Data Centers can be close in proximity. There might not be any difference in the routing cost to reach the Application Servers in different Edge DCs. The same routing cost to multiple locations can cause packets from one flow to be forwarded to different locations, which can cause service glitches.

#### 1.4. Problem #2: sticking to original App Server

When a mobile device moves to a new location but continues the same application flow, the router connected to the new UPF might choose the App Server closer to the new location. As shown in the figure below, when the MD1 in 5G-site-A moves to the 5G-Site-B, the router directly connected to 5G PSA2 might forward the packets destined towards the S1: aa08::4450 to the server located in L-DN2 because L-DN2 has the lowest cost based on routing. This is not the desired behavior for some services, which are called Sticky Services in this document.

Even for some advanced applications with built-in mechanisms to re-sync the communications at the application layer after switching to a new location, service glitches are often experienced.

It worth noting that not all services need to be sticky. We assume only a subset of services are, and the Network is informed of the services that need to be sticky, usually by requests from application developers or controllers.

This document describes an IPv6-based network layer solution to stick the packets belonging to the same flow of a mobile device to its original App Server location after the mobile device is anchored to a new nearby UPF-PSA.

Note: for ease of description, the Edge Computing Server, Application Server, or App Server are used interchangeably throughout this document.

## 2. Conventions used in this document

- APN6            Application aware network using IPv6. The term "Application" has very broad meanings. In this document the term "Application" refers to any applications that use ANYCAST servers in the 5G Edge Computing Environment.
- A-ER:           Egress Router to an Application Server, [A-ER] is used to describe the last router that the Application Server is attached. For 5G EC environment, the A-ER can be the gateway router to a (mini) Edge Computing Data Center.
- Application Server: An application server is a physical or virtual server that host the software system for the application.
- Application Server Location: Represent a cluster of servers at one location serving the same Application. One application may have a Layer 7 Load balancer, whose address(es) are reachable from external IP network, in front of a set of application servers. From IP network perspective, this whole group of servers are considered as the Application server at the location.



Edge Application Server: used interchangeably with Application Server throughout this document.

EC:              Edge Computing

Edge Hosting Environment: An environment providing support required for Edge Application Server's execution.

NOTE: The above terminologies are the same as those used in 3GPP TR 23.758

Edge DC:        Edge Data Center, which provides the Edge Computing Hosting Environment. It might be co-located with or very close to a 5G Base Station.

gNB             next generation Node B

L-DN:           Local Data Network

MD:             Mobile Device, which is the same as the UE (User Equipment) used in 3GPP. The term "mobile device" is used instead of UE to emphasize on sticking services originated from the devices that are mobile to same server.

PSA:            PDU Session Anchor (UPF)

SSC:            Session and Service Continuity

UE:             User Equipment. UE is same as a mobile device in this document.

UPF:            User Plane Function

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

### 3. Stick a Flow to an ANYCAST Server

When servers attached to different egress routers are assigned with the same IP address, the routers in the LDN see multiple paths to the IP address. The Egress nodes' unicast addresses are the Next Hops (i.e., R1, R2, and R3) to reach the Edge Computing server ANYCAST address.

The routers choose the lowest cost path. [5G-EC-OSPF-EXT] and [5G-EC-BGP-EXT] describe the OSPF and BGP extension to propagate additional costs about the site where the servers are located so that the site costs can be incorporated into the path computation.

Flow sticking to one server is not the same as flow nailing down to the same server. When the network cost is significantly increased, such as the mobile device moving to a very far away location or the extreme case of link failure to the original server, another server with the same IP address is selected.

The Flow Affinity feature, which most commercial routers support today, can ensure packets belonging to one flow be forwarded along the same path to the same egress router, which then delivers the packets to the attached server.

Editor's note: for IPv6 traffic, Flow Affinity can be supported by the Local Data Network (LDN) routers forwarding the packets with the same Flow Label in the packets' IPv6 Header along the same path towards the same egress router. For IPv4 traffic, 5 tuples in the IPv4 header can be used to achieve the Flow Affinity.

When a UE moves to a different cell site, the packets from the UE might enter the 5G LDN from a different UPF. Suppose the handover to the new cell site is in the middle of a flow from the UE. In that case, the new ingress router directly connected to the new UPF needs to have the original egress router information to stick the flow from the UE to the original egress router. The original egress router is called Sticky Egress throughout this document.

### 4. Sticky flow for QUIC based Applications

For applications using QUIC transport protocol, ANYCAST stickiness are supported natively. During the initial handshake, QUIC servers can provide a "preferred address" (IP or IPv6 and port number), and the client can immediately migrate the connection to use that address. This was

specifically designed to support servers listening on anycast addresses, so the connection can be pinned to a unicast address specific to the server.

## 5. Other Solutions within a Limited Domain

This section describes some sticky flow solutions within a limited domain [RFC8799] for applications not based on QUIK.

Within a limited domain [RFC8799], mobile devices, edge servers, and network functions are under one administrative domain. Therefore, it is feasible for mobile devices to perform specific actions.

### 5.1. Use Case of 5G Edge Computing in a limited domain.

Some 5G Connected devices, such as drones for fighting natural disasters or robots in Industry 4.0 environments, need ultra-low latency responses from their analytic servers. To reach ultra-low latency, those analytic functions can be hosted on servers very close to radio towers.

All the functions (including networking and analytics) and devices are administrated by one operator. Network devices within the 5G LDN limited domain might be provided by different vendors, therefore needing interoperable solutions.

### 5.2. End Node Based Sticky Service Solution

The End-Node-based Sticky Service solution needs IPv6 mobile devices to insert the Destination Option header extracted from the packet received from the network side to the IPv6 Header of the next packet if the next packet belongs to the same flow. This action dramatically simplifies the processing at the LDN's Ingress routers.

Here are some assumptions for the End-Node based Sticky Service solution:

- The mobile devices are under the same administrative control as the Edge computing servers.
- If an Edge Computing service needs to be sticky in the 5G Edge Computing environment, the corresponding service ID is registered with the 5G Edge Computing controller. The Sticky Service ID can be the IP address (unicast or ANYCAST) of the server.

Here is the overview of the End-Node based Sticky Service solution:

- Each ANYCAST Edge Computing server either learns or is informed of the unicast Sticky Egress address (Section 3). The goal is to deliver packets belonging to one flow to the same Sticky Egress address for the ANYCAST address.
- When an Edge Computing server sends data packets back to a client (or the mobile device), it inserts the Sticky-Dst-SubTLV (described in Section 4.4) into the packets' Destination Option Header.
- The client (or the mobile device) needs to copy the Destination Option Header from the received packet to the next packet's Destination Header if the next packet belongs to the same flow as the previous packet.
- If the following conditions are true, the ingress router encapsulates the packet from the client in a tunnel whose outer destination address is set to the Sticky Egress Address extracted from the packet's Sticky-Dst-SubTLV:
  - o The destination of the packet from the client-side matches with one of the Sticky Service ACLs configured on the ingress router of the LDN,
  - o the packet header has the Destination Option present with Sticky-Dst-SubTLV.
- Else (i.e., one of the conditions above is not true), the ingress node uses its algorithm, such as the least cost as described in [5G-EC-Metrics], to select the optimal Sticky Egress address for forwarding the packet.

#### 5.2.1. Edge Controller Based Solution.

To be added.

[Editor's note: can consider adding something along the line of the following, which is suggested by the email: say 5G/MEC control plane can tell the UE what address to use, it does NOT mean a UE will query whenever it is anchored to a new UPF. The initial query when it needs a service will return the unicast address of a server based

on all kinds of information/constraints, including the server load information talked about in draft-dunbar-idr-5g-edge-compute-app-meta-data. After that, the server won't change until new server is indeed needed (this is what "sticky service" is about, right). When a server change is indeed needed, the 5G/MEC control plane will tell the UE the new unicast address to use and tell the servers to move the corresponding application data when necessary.

]

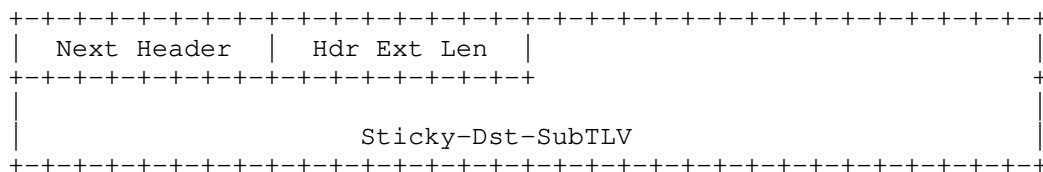
### 5.3. Sticky Egress Address Discovery

To an App server with ANYCAST address, the Sticky Egress address is the same as its default Gateway address.

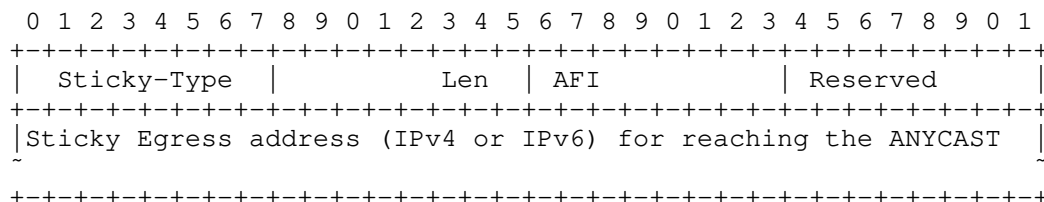
To prevent malicious entities sending DDOS attacks to routers within 5G EC LDN, e.g., the Sticky Egress address that is encoded in the Destination option header in the packets sent back to the clients, a proxy Sticky Egress address can be encoded in the Destination option header. The proxy Sticky Egress address is only recognizable by the 5G EC LDN ingress nodes, i.e., the Ra and Rb in Figure 1, but not routable in other networks. The LDN ingress routers can translate the proxy Sticky Egress to a routable address for the Sticky Egress node after the source addresses of the packets are authenticated.

### 5.4. Sticky-Dst-SubTLV in Destination Extension Header

A new Sticky-Dst-SubTLV is specified as below, which can be inserted into the IPv6 Destination Options header. The IPv6 Destination Option Header is specified by [RFC8200] as having a Next Header value of 60:



Sticky-Dst-SubTLV is specified as:



Sticky-Type = 1: indicate the Sticky Egress unicast address at encoded in the Sticky-Dst-SubTLV.

### 5.5. Processing at the Ingress router

- An Ingress router is configured with an ACL for filtering out the applications that need sticky service.

Note, not all applications need sticky service. Using ACL can significantly reduce the processing on the routers.

- When an Ingress router receives a packet from the 5G side that matches the ACL, the Ingress router extracts the Sticky-Dst-SubTLV from the packet IPv6 header if the field exists in the packet header.
- Encapsulate the packet with the tunnel type that are supported by the original Sticky Egress node, using the extracted Sticky Egress address in the destination field of the outer Header, and forward the packet.

Note: if the proxy Sticky Egress address is encoded in the Sticky-Dst-SubTLV, the ingress router needs to translate the proxy Sticky Egress address to a routable address.

If none of the above conditions are met, the ingress router uses its algorithm to select the optimal Sticky Egress node to forward the packet.

### 6. Tunnel based Sticky Service Solution

For environments that mobile devices cannot change their processing behavior as described in Section 4, a Tunnel based

Sticky Service solution can be used. This solution does not depend on mobile device's behavior. However, this solution does require ingress routers to filter out the registered sticky services and might need some level of assistance from the LDN network controller.

#### 6.1. Desired functions by the Network Controller

#### 6.2. Ingress and Egress Routers Processing Behavior

The solution assumes that both ingress routers and egress routers support at least one type of tunnel and are configured with ACLs to filter out packets whose destination or source addresses match with the Sticky Service Identifier. The solution also assumes there are only limited number of Sticky Services to be supported.

An ingress router needs to build a Sticky-Service-Table, with the following minimum attributes. The Sticky-Service-Table is initialized to be empty.

- Sticky Service ID
- Flow Label
- Sticky Egress address
- Timer

#### Editor's Note:

When a mobile device moves from one 5G Site to another, the same mobile device will have a new IP address. "Flow Label + Sticky Service ID" stays the same when a mobile device is anchored to a new PSA. Therefore, this solution uses "Flow Label + Sticky Service ID" to identify a sticky flow. Since the chance of different mobile devices sending packets to the same ANYCAST address using the same Flow Label is very low, it is with high probability that "Flow Label + Sticky Service ID" can uniquely identify a flow. When multiple mobile devices using the same Flow Label sending packets to the same ANYCAST address, the solution described in this section will stick the flows to the same ANYCAST server attached to the Sticky Egress router. This behavior doesn't cause any harm.

Each entry in the Sticky-Service-Table has a Timer because a sticky service is no longer sticky if there are no packets of the same flow destined towards the service ID for a period of time. The Timer should be larger than a typical TCP session Timeout value. An entry is automatically removed from the Sticky-Service-Table when its timer expires.

Note: since there are only small number of Sticky services, the Sticky-Service-Table is not very large.

When an ingress router receives a packet from a mobile device matching with one of the Sticky Service ACLs and there is no entry in the Sticky-Service-Table matching the Flow Label and the Sticky Service ID, the ingress router considers the packet to be the first packet of the flow. There is no need to sticking the packet to any location. The ingress router uses its own algorithm to select the optimal egress node as the Sticky Egress address for the ANYCAST address, encapsulates the packet with a tunnel that is supported by the egress node. The tunnel's destination address is set to the egress node address.

When an egress router receives a packet from an attached host with the packet's source address matching with one of the Sticky Service IDs, the egress router encapsulates the packet with a tunnel that is supported by the ingress router and the tunnel's destination address is set to the ingress router address. An Egress router learns the ingress router address for a mobile device IP address via BGP UPDATE messages.

When an ingress router receives a packet in a tunnel from any egress router and the packet's source address matches with a Sticky Service ID, the egress router address is set as the Sticky Egress address for the Sticky Service ID. The ingress router adds the entry of "Sticky-Service-ID + Flow Label + the associated Sticky Egress address + Timer" to the Sticky-Service-Table if the entry doesn't exist yet in the table. If the entry exists, the ingress router refreshes the Timer of the entry in the table.

When the ingress router receives the subsequent packets of a flow from the 5G side matching with an Sticky Service ID and the Sticky-Service ID exists in the Sticky-Service-Table, the ingress router uses the Sticky Egress address found in the Sticky-Service-Table to encapsulate the packet and refresh the Timer of the entry. If the Sticky-Service ID doesn't exist in the table, the ingress router considers the packet as the first packet of a flow.



The subsequent sections describe how ingress nodes prorogate their Sticky-Service-Table to their neighboring ingress nodes. The propagation is for neighboring ingress nodes to be informed of the Sticky Egress address to a sticky service if a mobile device moves to a new neighboring 5G site resulting in anchoring to a new ingress node.

### 6.3. A Solution without the Communication with 5G system.

When a mobile device moves to a very far away 5G site, say a different geographic region, the benefit of sticking to the original ANYCAST server is out weighted by network delay. Then, there is no point sending packets to the Sticky Egress node if the ingress router very far away. Therefore, it is necessary for each ingress router to have a group of neighboring ingress routers that are not too far away from the potential Sticky Egress nodes selected by the ingress router. This group of ingress routers is called the Neighboring Ingress Group. Each ingress router can either automatically discover its Neighboring Ingress Group by routing protocols or is configured by its controller. It is out of the scope of this document on how ingress nodes discover its Neighboring Ingress Group.

Each ingress node needs to periodically advertise its Sticky-Service-Table to the routers within its Neighboring Ingress Group.

Upon receiving the Sticky-Service-Table from routers in its Neighboring Ingress Group, each ingress router merges the entries from the received Sticky-Service-Table to its own.

The ingress and the egress nodes perform the same actions as described in Section 5.1.

### 6.4. A Solution that depends on the communication with 5G system

In this scenario, there is communication with 5G System and network get notified by a mobile device is anchored to a new PSA.

When a mobile device is re-anchoring from PSA1 to PSA2, 5GC EC management system sends a notification to the router that is directly connected to PSA1. The notification includes the address of the new PSA that the mobile device is to be anchored, i.e. the PSA2, and the mobile device's new IP address.

In this scenario, the Sticky Service can be uniquely identified by "Sticky Service ID" + "mobile device address". the Sticky-Service-Table should include the following attributes:

- Sticky Service ID
- mobile device address
- Sticky Egress address
- Timer

Upon receiving the notification from the 5G EC management system, the ingress router (i.e. the one directly connected to the old PSA) sends the specific entry of the Sticky-Service Table, i.e. "Sticky Service ID" + mobile device address + Sticky Egress + Timer to the router directly connected to the new PSA.

Upon receiving the entry, the ingress router merges the entry into its own Sticky-Service-Table.

The ingress and egress router processing are the same as described in Section 5.1 except a flow is now uniquely identified by the "Sticky Service ID" + "mobile device address" instead of "Sticky Service ID" + "Flow Label".

## 7. Expanding APN6 for Sticky Service information

The Application-aware ID and Service-Para Option described [APN6] can be expanded to include the sticky service information.

### 7.1. Sticky Service ID encoded in the Application-aware ID

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Sticky Level | StickyServiceID | Reserved      | Flow ID      |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Sticky Level: represent how important for an application to stick to its ANYCAST servers. Some applications may prefer one flow sticking to the original ANYCAST server, but not required. Some applications may require the stickiness.

StickyServiceID: the ANYCAST address of the application servers.

The Reserved field can be used for future to identifier the 5G access domain for the flow.

Flow ID: the identifier for the flow that needs to stick to a specific ANYCAST server.

#### 7.2. Sticky Service Sub-TLV encoded in APN6 Service-para option

The Sticky-Dst-SubTLV described in the Section 4.2 of this document can be included in the Service-Para Sub-TLVs field.

### 8. Manageability Considerations

To be added.

### 9. Security Considerations

To be added.

### 10. IANA Considerations

To be added.

### 11. References

#### 11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4364] E. rosen, Y. Rekhter, "BGP/MPLS IP Virtual Private networks (VPNs)", Feb 2006.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

[RFC8200] s. Deering R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", July 2017

## 11.2. Informative References

- [3GPP-EdgeComputing] 3GPP TR 23.748, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on enhancement of support for Edge Computing in 5G Core network (5GC)", Release 17 work in progress, Aug 2020.
- [5G-EC-Metrics] L. Dunbar, H. Song, J. Kaippallimalil, "IP Layer Metrics for 5G Edge Computing Service", draft-dunbar-ippm-5g-edge-compute-ip-layer-metrics-00, work-in-progress, Oct 2020.
- [5G-EC-OSPF-EXT] L. Dunbar, H.Chen, A. Wang, "OSPF extension for 5G Edge Computing Service", draft-dunbar-lsr-5g-edge-compute-ospf-ext-05, work-in-progress, March 2021.
- [5G-EC-BGP-EXT] L. Dunbar, K. Majumdar, H. Wang, "BGP NLRI App Meta Data for 5G Edge Computing Service", draft-dunbar-idr-5g-edge-compute-app-meta-data-02, work-in-progress, March 2021.
- [APN6] Z. Li, et al, "Application-aware IPv6 Networking (APN6) Encapsulation", draft-li-6man-app-aware-ipv6-network-03, work-in-progress, Feb 2021.
- [RFC5521] P. Mohapatra, E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", April 2009.
- [BGP-SDWAN-Port] L. Dunbar, H. Wang, W. Hao, "BGP Extension for SDWAN Overlay Networks", draft-dunbar-idr-bgp-sdwan-overlay-ext-03, work-in-progress, Nov 2018.

[SDWAN-EDGE-Discovery] L. Dunbar, S. Hares, R. Raszuk, K. Majumdar, "BGP UPDATE for SDWAN Edge Discovery", draft-dunbar-idr-sdwan-edge-discovery-00, work-in-progress, July 2020.

[Tunnel-Encap] E. Rosen, et al "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-10, Aug 2018.

## 12. Acknowledgments

Acknowledgements to Gyan Mishra, Jeffrey Zhang, Joel Halpern, Ron Bonica, Donald Eastlake, and Eduard Vasilenko for their review and contributions.

This document was prepared using 2-Word-v2.0.template.dot.

## Authors' Addresses

Linda Dunbar  
Futurewei  
Email: ldunbar@futurewei.com

John Kaippallimalil  
Futurewei  
Email: john.kaippallimalil@futurewei.com



6MAN Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: October 30, 2022

G. Fioccola  
T. Zhou  
Huawei  
M. Cociglio  
Telecom Italia  
F. Qin  
China Mobile  
R. Pang  
China Unicom  
April 28, 2022

IPv6 Application of the Alternate Marking Method  
draft-ietf-6man-ipv6-alt-mark-14

## Abstract

This document describes how the Alternate Marking Method can be used as a passive performance measurement tool in an IPv6 domain. It defines a new Extension Header Option to encode Alternate Marking information in both the Hop-by-Hop Options Header and Destination Options Header.

## 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 <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 30, 2022.

## Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .                                 | 2  |
| 1.1. Terminology . . . . .                                | 3  |
| 1.2. Requirements Language . . . . .                      | 3  |
| 2. Alternate Marking application to IPv6 . . . . .        | 3  |
| 2.1. Controlled Domain . . . . .                          | 5  |
| 2.1.1. Alternate Marking Measurement Domain . . . . .     | 6  |
| 3. Definition of the AltMark Option . . . . .             | 7  |
| 3.1. Data Fields Format . . . . .                         | 7  |
| 4. Use of the AltMark Option . . . . .                    | 8  |
| 5. Alternate Marking Method Operation . . . . .           | 10 |
| 5.1. Packet Loss Measurement . . . . .                    | 10 |
| 5.2. Packet Delay Measurement . . . . .                   | 12 |
| 5.3. Flow Monitoring Identification . . . . .             | 13 |
| 5.4. Multipoint and Clustered Alternate Marking . . . . . | 15 |
| 5.5. Data Collection and Calculation . . . . .            | 16 |
| 6. Security Considerations . . . . .                      | 16 |
| 7. IANA Considerations . . . . .                          | 20 |
| 8. Acknowledgements . . . . .                             | 20 |
| 9. References . . . . .                                   | 20 |
| 9.1. Normative References . . . . .                       | 20 |
| 9.2. Informative References . . . . .                     | 21 |
| Authors' Addresses . . . . .                              | 22 |

## 1. Introduction

[I-D.ietf-ippm-rfc8321bis] and [I-D.ietf-ippm-rfc8889bis] describe a passive performance measurement method, which can be used to measure packet loss, latency and jitter on live traffic. Since this method is based on marking consecutive batches of packets, the method is often referred to as the Alternate Marking Method.

This document defines how the Alternate Marking Method can be used to measure performance metrics in IPv6. The rationale is to apply the Alternate Marking methodology to IPv6 and therefore allow detailed packet loss, delay and delay variation measurements both hop-by-hop and end-to-end to exactly locate the issues in an IPv6 network.

The Alternate Marking is an on-path telemetry technique and consists of synchronizing the measurements in different points of a network by



switching the value of a marking bit and therefore dividing the packet flow into batches. Each batch represents a measurable entity recognizable by all network nodes along the path. By counting the number of packets in each batch and comparing the values measured by different nodes, it is possible to precisely measure the packet loss. Similarly, the alternation of the values of the marking bits can be used as a time reference to calculate the delay and delay variation. The Alternate Marking operation is further described in Section 5.

The format of IPv6 addresses is defined in [RFC4291] while [RFC8200] defines the IPv6 Header, including a 20-bit Flow Label and the IPv6 Extension Headers.

This document introduces a new TLV (type-length-value) that can be encoded in the Options Headers (Hop-by-Hop or Destination) for the purpose of the Alternate Marking Method application in an IPv6 domain.

The threat model for the application of the Alternate Marking Method in an IPv6 domain is reported in Section 6. As with all on-path telemetry techniques, the only definitive solution is that this methodology MUST be applied in a controlled domain.

### 1.1. Terminology

This document uses the terms related to the Alternate Marking Method as defined in [I-D.ietf-ippm-rfc8321bis] and [I-D.ietf-ippm-rfc8889bis].

### 1.2. 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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 2. Alternate Marking application to IPv6

The Alternate Marking Method requires a marking field. Several alternatives could be considered such as IPv6 Extension Headers, IPv6 Address and Flow Label. But, it is necessary to analyze the drawbacks for all the available possibilities, more specifically:

Reusing existing Extension Header for Alternate Marking leads to a non-optimized implementation;

Using the IPv6 destination address to encode the Alternate Marking processing is very expensive;

Using the IPv6 Flow Label for Alternate Marking conflicts with the utilization of the Flow Label for load distribution purpose ([RFC6438]).

In the end, a new Hop-by-Hop or a new Destination Option is the best choice.

The approach for the Alternate Marking application to IPv6 specified in this memo is compliant with [RFC8200]. It involves the following operations:

- o The source node is the only one that writes the Option Header to mark alternately the flow (for both Hop-by-Hop and Destination Option). The intermediate nodes and destination node MUST only read the marking values of the option without modifying the Option Header.
- o In case of Hop-by-Hop Option Header carrying Alternate Marking bits, it is not inserted or deleted, but can be read by any node along the path. The intermediate nodes may be configured to support this Option or not and the measurement can be done only for the nodes configured to read the Option. As further discussed in Section 4, the presence of the hop-by-hop option should not affect the traffic throughput both on nodes that do not recognize this option and on the nodes that support it. However, it is worth mentioning that there is a difference between theory and practice. Indeed, in a real implementation it can happen that packets with hop-by-hop option could also be skipped or processed in the slow path. While some proposals are trying to address this problem and make Hop-by-Hop Options more practical ([I-D.peng-v6ops-hbh], [I-D.hinden-6man-hbh-processing]), these aspects are out of the scope for this document.
- o In case of Destination Option Header carrying Alternate Marking bits, it is not processed, inserted, or deleted by any node along the path until the packet reaches the destination node. Note that, if there is also a Routing Header (RH), any visited destination in the route list can process the Option Header.

Hop-by-Hop Option Header is also useful to signal to routers on the path to process the Alternate Marking. However, as said, routers will only examine this option if properly configured.

The optimization of both implementation and scaling of the Alternate Marking Method is also considered and a way to identify flows is

required. The Flow Monitoring Identification field (FlowMonID), as introduced in Section 5.3, goes in this direction and it is used to identify a monitored flow.

The FlowMonID is different from the Flow Label field of the IPv6 Header ([RFC6437]). The Flow Label field in the IPv6 header is used by a source to label sequences of packets to be treated in the network as a single flow and, as reported in [RFC6438], it can be used for load-balancing/equal cost multi-path (LB/ECMP). The reuse of Flow Label field for identifying monitored flows is not considered because it may change the application intent and forwarding behavior. Also, the Flow Label may be changed en route and this may also invalidate the integrity of the measurement. Furthermore, since the Flow Label is pseudo-random, there is always a finite probability of collision. Those reasons make the definition of the FlowMonID necessary for IPv6. Indeed, the FlowMonID is designed and only used to identify the monitored flow. Flow Label and FlowMonID within the same packet are totally disjoint, have different scope, are used to identify flows based on different criteria, and are intended for different use cases.

The rationale for the FlowMonID is further discussed in Section 5.3. This 20 bit field allows easy and flexible identification of the monitored flow and enables improved measurement correlation and finer granularity since it can be used in combination with the traditional TCP/IP 5-tuple to identify a flow. An important point that will be discussed in Section 5.3 is the uniqueness of the FlowMonID and how to allow disambiguation of the FlowMonID in case of collision.

The following section highlights an important requirement for the application of the Alternate Marking to IPv6. The concept of the controlled domain is explained and it is considered an essential precondition, as also highlighted in Section 6.

## 2.1. Controlled Domain

[RFC8799] introduces the concept of specific limited domain solutions and, in this regard, it is reported the IPv6 Application of the Alternate Marking Method as an example.

IPv6 has much more flexibility than IPv4 and innovative applications have been proposed, but for a number of reasons, such as the policies, options supported, the style of network management and security requirements, it is suggested to limit some of these applications to a controlled domain. This is also the case of the Alternate Marking application to IPv6 as assumed hereinafter.

Therefore, the IPv6 application of the Alternate Marking Method MUST be deployed in a controlled domain. It is RECOMMENDED that an implementation filters packets that carry Alternate Marking data and are entering or leaving the controlled domains.

A controlled domain is a managed network where it is required to select, monitor and control the access to the network by enforcing policies at the domain boundaries in order to discard undesired external packets entering the domain and check the internal packets leaving the domain. It does not necessarily mean that a controlled domain is a single administrative domain or a single organization. A controlled domain can correspond to a single administrative domain or can be composed by multiple administrative domains under a defined network management. Indeed, some scenarios may imply that the Alternate Marking Method involves more than one domain, but in these cases, it is RECOMMENDED that the multiple domains create a whole controlled domain while traversing the external domain by employing IPsec [RFC4301] authentication and encryption or other VPN technology that provides full packet confidentiality and integrity protection. In a few words, it must be possible to control the domain boundaries and eventually use specific precautions if the traffic traverse the Internet.

The security considerations reported in Section 6 also highlight this requirement.

#### 2.1.1. Alternate Marking Measurement Domain

The Alternate Marking measurement domain can overlap with the controlled domain or may be a subset of the controlled domain. The typical scenarios for the application of the Alternate Marking Method depend on the controlled domain boundaries, in particular:

the user equipment can be the starting or ending node, only in case it is fully managed and if it belongs to the controlled domain. In this case the user generated IPv6 packets contain the Alternate Marking data. But, in practice, this is not common due to the fact that the user equipment cannot be totally secured in the majority of cases.

the CPE (Customer Premises Equipment) is most likely to be the starting or ending node since it connects the user's premises with the service provider's network and therefore belongs to the operator's controlled domain. Typically the CPE encapsulates a received packet in an outer IPv6 header which contains the Alternate Marking data. The CPE can also be able to filter and drop packets from outside of the domain with inconsistent fields to make effective the relevant security rules at the domain

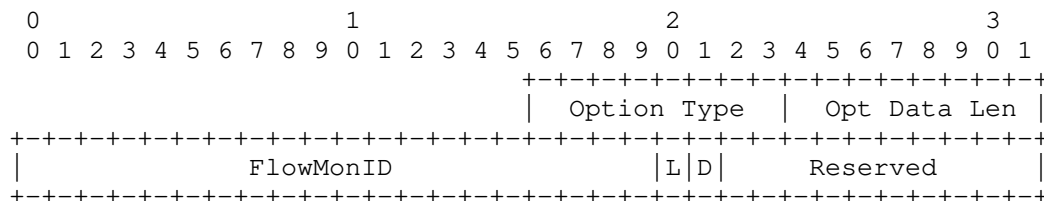
boundaries, for example a simple security check can be to insert the Alternate Marking data if and only if the destination is within the controlled domain.

### 3. Definition of the AltMark Option

The definition of a new TLV for the Options Extension Headers, carrying the data fields dedicated to the Alternate Marking method, is reported below.

#### 3.1. Data Fields Format

The following figure shows the data fields format for enhanced Alternate Marking TLV (AltMark). This AltMark data can be encapsulated in the IPv6 Options Headers (Hop-by-Hop or Destination Option).



where:

- o Option Type: 8-bit identifier of the type of Option that needs to be allocated. Unrecognized Types MUST be ignored on processing. For Hop-by-Hop Options Header or Destination Options Header, [RFC8200] defines how to encode the three high-order bits of the Option Type field. The two high-order bits specify the action that must be taken if the processing IPv6 node does not recognize the Option Type; for AltMark these two bits MUST be set to 00 (skip over this Option and continue processing the header). The third-highest-order bit specifies whether the Option Data can change en route to the packet's final destination; for AltMark the value of this bit MUST be set to 0 (Option Data does not change en route). In this way, since the three high-order bits of the AltMark Option are set to 000, it means that nodes can simply skip this Option if they do not recognize and that the data of this Option do not change en route, indeed the source is the only one that can write it.
- o Opt Data Len: 4. It is the length of the Option Data Fields of this Option in bytes.

- o FlowMonID: 20-bit unsigned integer. The FlowMon identifier is described in Section 5.3. As further discussed below, it has been picked as 20 bits since it is a reasonable value and a good compromise in relation to the chance of collision. It MUST be set pseudo randomly by the source node or by a centralized controller.
- o L: Loss flag for Packet Loss Measurement as described in Section 5.1;
- o D: Delay flag for Single Packet Delay Measurement as described in Section 5.2;
- o Reserved: is reserved for future use. These bits MUST be set to zero on transmission and ignored on receipt.

#### 4. Use of the AltMark Option

The AltMark Option is the best way to implement the Alternate Marking method and it is carried by the Hop-by-Hop Options header and the Destination Options header. In case of Destination Option, it is processed only by the source and destination nodes: the source node inserts and the destination node processes it. While, in case of Hop-by-Hop Option, it may be examined by any node along the path, if explicitly configured to do so.

It is important to highlight that the Option Layout can be used both as Destination Option and as Hop-by-Hop Option depending on the Use Cases and it is based on the chosen type of performance measurement. In general, it is needed to perform both end to end and hop by hop measurements, and the Alternate Marking methodology allows, by definition, both performance measurements. In many cases the end-to-end measurement is not enough and it is required the hop-by-hop measurement, so the most complete choice can be the Hop-by-Hop Options Header.

IPv6, as specified in [RFC8200], allows nodes to optionally process Hop-by-Hop headers. Specifically the Hop-by-Hop Options header is not inserted or deleted, but may be 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. Also, it is expected that nodes along a packet's delivery path only examine and process the Hop-by-Hop Options header if explicitly configured to do so.

Another scenario that can be mentioned is the presence of a Routing Header, in particular it is possible to consider SRv6. A new type of Routing Header, referred as Segment Routing Header (SRH), has been defined in [RFC8754] for SRv6. Like any other use case of IPv6, Hop-

by-Hop and Destination Options are usable when SRv6 header is present. Because SRv6 is implemented through a Segment Routing Header (SRH), Destination Options before the Routing Header are processed by each destination in the route list, that means, in case of SRH, by every SR node that is identified by the SR path. More details about the SRv6 application are described in [I-D.fz-spring-srv6-alt-mark].

In summary, it is possible to list the alternative possibilities:

- o Destination Option not preceding a Routing Header => measurement only by node in Destination Address.
- o Hop-by-Hop Option => every router on the path with feature enabled.
- o Destination Option preceding a Routing Header => every destination node in the route list.

In general, Hop-by-Hop and Destination Options are the most suitable ways to implement Alternate Marking.

It is worth mentioning that new Hop-by-Hop Options are not strongly recommended in [RFC7045] and [RFC8200], unless there is a clear justification to standardize it, because nodes may be configured to ignore the Options Header, drop or assign packets containing an Options Header to a slow processing path. In case of the AltMark data fields described in this document, the motivation to standardize a new Hop-by-Hop Option is that it is needed for OAM (Operations, Administration, and Maintenance). An intermediate node can read it or not, but this does not affect the packet behavior. The source node is the only one that writes the Hop-by-Hop Option to mark alternately the flow, so, the performance measurement can be done for those nodes configured to read this Option, while the others are simply not considered for the metrics.

The Hop-by-Hop Option defined in this document is designed to take advantage of the property of how Hop-by-Hop options are processed. Nodes that do not support this Option SHOULD ignore them. This can mean that, in this case, the performance measurement does not account for all links and nodes along a path. The definition of the Hop-by-Hop Options in this document is also designed to minimize throughput impact both on nodes that do not recognize the Option and on node that support it. Indeed, the three high-order bits of the Options Header defined in this draft are 000 and, in theory, as per [RFC8200] and [I-D.hinden-6man-hbh-processing], this means "skip if do not recognize and data do not change en route". [RFC8200] also mentions that the nodes only examine and process the Hop-by-Hop Options header

if explicitly configured to do so. For these reasons, this Hop-by-Hop Option should not affect the throughput. However, in practice, it is important to be aware that the things may be different in the implementation and it can happen that packets with Hop-by-Hop are forced onto the slow path, but this is a general issue, as also explained in [I-D.hinden-6man-hbh-processing]. It is also worth mentioning that the application to a controlled domain should avoid the risk of arbitrary nodes dropping packets with Hop-by-Hop Options.

## 5. Alternate Marking Method Operation

This section describes how the method operates.

[I-D.ietf-ippm-rfc8321bis] introduces several applicable methods which are reported below, and a new field is introduced to facilitate the deployment and improve the scalability.

### 5.1. Packet Loss Measurement

The measurement of the packet loss is really straightforward in comparison to the existing mechanisms, as detailed in [I-D.ietf-ippm-rfc8321bis]. The packets of the flow are grouped into batches, and all the packets within a batch are marked by setting the L bit (Loss flag) to a same value. The source node can switch the value of the L bit between 0 and 1 after a fixed number of packets or according to a fixed timer, and this depends on the implementation. The source node is the only one that marks the packets to create the batches, while the intermediate nodes only read the marking values and identify the packet batches. By counting the number of packets in each batch and comparing the values measured by different network nodes along the path, it is possible to measure the packet loss occurred in any single batch between any two nodes. Each batch represents a measurable entity recognizable by all network nodes along the path.

Both fixed number of packets and fixed timer can be used by the source node to create packet batches. But, as also explained in [I-D.ietf-ippm-rfc8321bis], the timer-based batches are preferable because they are more deterministic than the counter-based batches. There is no definitive rule for counter-based batches, differently from timer-based batches. Using a fixed timer for the switching offers better control over the method, indeed the length of the batches can be chosen large enough to simplify the collection and the comparison of the measures taken by different network nodes. In the implementation the counters can be sent out by each node to the controller that is responsible for the calculation. It is also possible to exchange this information by using other on-path techniques. But this is out of scope for this document.



Packets with different L values may get swapped at batch boundaries, and in this case, it is required that each marked packet can be assigned to the right batch by each router. It is important to mention that for the application of this method there are two elements to consider: the clock error between network nodes and the network delay. These can create offsets between the batches and out-of-order of the packets. The mathematical formula on timing aspects, explained in section 5 of [I-D.ietf-ippm-rfc8321bis], must be satisfied and it takes into considerations the different causes of reordering such as clock error and network delay. The assumption is to define the available counting interval where to get stable counters and to avoid these issues. Specifically, if the effects of network delay are ignored, the condition to implement the methodology is that the clocks in different nodes MUST be synchronized to the same clock reference with an accuracy of  $\pm B/2$  time units, where B is the fixed time duration of the batch, which refers to the original marking interval at the source node considering that this interval could fluctuate along the path. In this way each marked packet can be assigned to the right batch by each node. Usually the counters can be taken in the middle of the batch period to be sure to take still counters. In a few words this implies that the length of the batches MUST be chosen large enough so that the method is not affected by those factors. The length of the batches can be determined based on the specific deployment scenario.

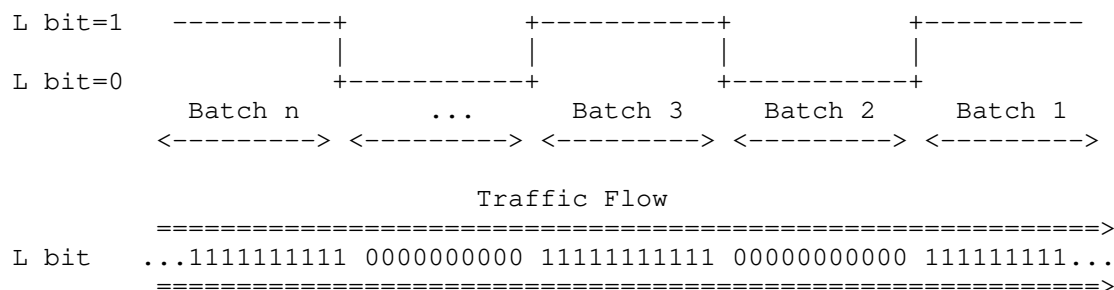


Figure 1: Packet Loss Measurement and Single-Marking Methodology using L bit

It is worth mentioning that the duration of the batches is considered stable over time in the previous figure. In theory, it is possible to change the length of batches over time and among different flows for more flexibility. But, in practice, it could complicate the correlation of the information.

## 5.2. Packet Delay Measurement

The same principle used to measure packet loss can be applied also to one-way delay measurement. Delay metrics MAY be calculated using the two possibilities:

1. **Single-Marking Methodology:** This approach uses only the L bit to calculate both packet loss and delay. In this case, the D flag MUST be set to zero on transmit and ignored by the monitoring points. The alternation of the values of the L bit can be used as a time reference to calculate the delay. Whenever the L bit changes and a new batch starts, a network node can store the timestamp of the first packet of the new batch, that timestamp can be compared with the timestamp of the first packet of the same batch on a second node to compute packet delay. But this measurement is accurate only if no packet loss occurs and if there is no packet reordering at the edges of the batches. A different approach can also be considered and it is based on the concept of the mean delay. The mean delay for each batch is calculated by considering the average arrival time of the packets for the relative batch. There are limitations also in this case indeed, each node needs to collect all the timestamps and calculate the average timestamp for each batch. In addition, the information is limited to a mean value.
2. **Double-Marking Methodology:** This approach is more complete and uses the L bit only to calculate packet loss and the D bit (Delay flag) is fully dedicated to delay measurements. The idea is to use the first marking with the L bit to create the alternate flow and, within the batches identified by the L bit, a second marking is used to select the packets for measuring delay. The D bit creates a new set of marked packets that are fully identified over the network, so that a network node can store the timestamps of these packets; these timestamps can be compared with the timestamps of the same packets on a second node to compute packet delay values for each packet. The most efficient and robust mode is to select a single double-marked packet for each batch, in this way there is no time gap to consider between the double-marked packets to avoid their reorder. Regarding the rule for the selection of the packet to be double-marked, the same considerations in Section 5.1 apply also here and the double-marked packet can be chosen within the available counting interval that is not affected by factors such as clock errors. If a double-marked packet is lost, the delay measurement for the considered batch is simply discarded, but this is not a big problem because it is easy to recognize the problematic batch and skip the measurement just for that one. So in order to have more

information about the delay and to overcome out-of-order issues this method is preferred.

In summary the approach with double marking is better than the approach with single marking. Moreover, the two approaches provide slightly different pieces of information and the data consumer can combine them to have a more robust data set.

Similar to what said in Section 5.1 for the packet counters, in the implementation the timestamps can be sent out to the controller that is responsible for the calculation or could also be exchanged using other on-path techniques. But this is out of scope for this document.

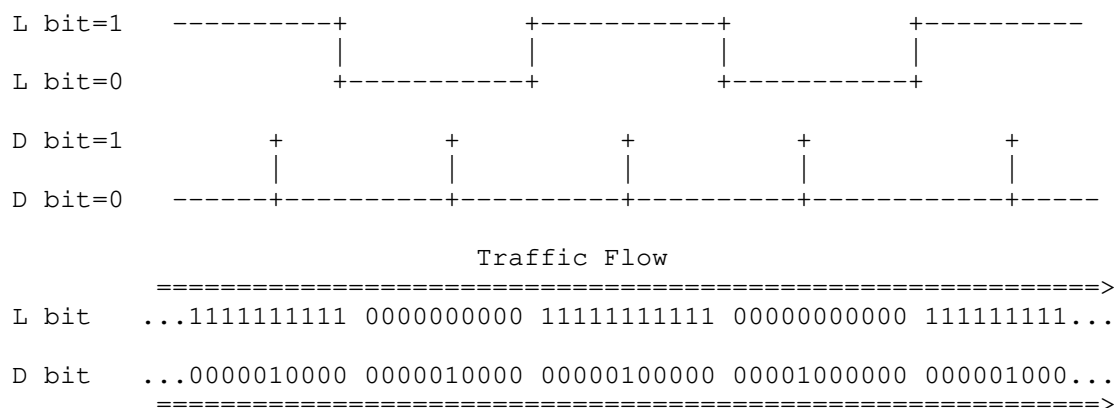


Figure 2: Double-Marking Methodology using L bit and D bit

Likewise to packet delay measurement (both for Single Marking and Double Marking), the method can also be used to measure the inter-arrival jitter.

### 5.3. Flow Monitoring Identification

The Flow Monitoring Identification (FlowMonID) identifies the flow to be measured and is required for some general reasons:

First, it helps to reduce the per node configuration. Otherwise, each node needs to configure an access-control list (ACL) for each of the monitored flows. Moreover, using a flow identifier allows a flexible granularity for the flow definition, indeed, it can be used together with other identifiers (e.g. 5-tuple).

Second, it simplifies the counters handling. Hardware processing of flow tuples (and ACL matching) is challenging and often incurs into performance issues, especially in tunnel interfaces.

Third, it eases the data export encapsulation and correlation for the collectors.

The FlowMonID MUST only be used as a monitored flow identifier in order to determine a monitored flow within the measurement domain. This entails not only an easy identification but improved correlation as well.

The value of 20 bits has been selected for the FlowMonID since it is a good compromise and implies a low rate of ambiguous FlowMonIDs that can be considered acceptable in most of the applications. The disambiguation issue can be solved by tagging the pseudo randomly generated FlowMonID with additional flow information. In particular, it is RECOMMENDED to consider the 3-tuple FlowMonID, source and destination addresses:

- o If the 20 bit FlowMonID is set independently and pseudo randomly in a distributed way there is a chance of collision. Indeed, by using the well-known birthday problem in probability theory, if the 20 bit FlowMonID is set independently and pseudo randomly without any additional input entropy, there is a 50% chance of collision for 1206 flows. So, for more entropy, FlowMonID is combined with source and destination addresses. Since there is a 1% chance of collision for 145 flows, it is possible to monitor 145 concurrent flows per host pairs with a 1% chance of collision.
- o If the 20 bits FlowMonID is set pseudo randomly but in a centralized way, the controller can instruct the nodes properly in order to guarantee the uniqueness of the FlowMonID. With 20 bits, the number of combinations is 1048576, and the controller should ensure that all the FlowMonID values are used without any collision. Therefore, by considering source and destination addresses together with the FlowMonID, it can be possible to monitor 1048576 concurrent flows per host pairs.

A consistent approach MUST be used in the Alternate Marking deployment to avoid the mixture of different ways of identifying. All the nodes along the path and involved into the measurement SHOULD use the same mode for identification. As mentioned, it is RECOMMENDED to use the FlowMonID for identification purpose in combination with source and destination addresses to identify a flow. By considering source and destination addresses together with the FlowMonID it can be possible to monitor 145 concurrent flows per host pairs with a 1% chance of collision in case of pseudo randomly

generated FlowMonID, or 1048576 concurrent flows per host pairs in case of centralized controller. It is worth mentioning that the solution with the centralized control allows finer granularity and therefore adds even more flexibility to the flow identification.

The FlowMonID field is set at the source node, which is the ingress point of the measurement domain, and can be set in two ways:

- a. It can be algorithmically generated by the source node, that can set it pseudo-randomly with some chance of collision. This approach cannot guarantee the uniqueness of FlowMonID since conflicts and collisions are possible. But, considering the recommendation to use FlowMonID with source and destination addresses the conflict probability is reduced due to the FlowMonID space available for each endpoint pair (i.e. 145 flows with 1% chance of collision).
- b. It can be assigned by the central controller. Since the controller knows the network topology, it can allocate the value properly to avoid or minimize ambiguity and guarantee the uniqueness. In this regard, the controller can verify that there is no ambiguity between different pseudo-randomly generated FlowMonIDs on the same path. The conflict probability is really small given that the FlowMonID is coupled with source and destination addresses and up to 1048576 flows can be monitored for each endpoint pair. When all values in the FlowMonID space are consumed, the centralized controller can keep track and reassign the values that are not used any more by old flows.

If the FlowMonID is set by the source node, the intermediate nodes can read the FlowMonIDs from the packets in flight and act accordingly. While, if the FlowMonID is set by the controller, both possibilities are feasible for the intermediate nodes which can learn by reading the packets or can be instructed by the controller.

#### 5.4. Multipoint and Clustered Alternate Marking

The Alternate Marking method can also be extended to any kind of multipoint to multipoint paths, and the network clustering approach allows a flexible and optimized performance measurement, as described in [I-D.ietf-ippm-rfc8889bis].

The Cluster is the smallest identifiable subnetwork of the entire Network graph that still satisfies the condition that the number of packets that goes in is the same that goes out. With network clustering, it is possible to use the partition of the network into clusters at different levels in order to perform the needed degree of detail. So, for Multipoint Alternate Marking, FlowMonID can identify

in general a multipoint-to-multipoint flow and not only a point-to-point flow.

#### 5.5. Data Collection and Calculation

The nodes enabled to perform performance monitoring collect the value of the packet counters and timestamps. There are several alternatives to implement Data Collection and Calculation, but this is not specified in this document.

There are documents on the control plane mechanisms of Alternate Marking, e.g. [I-D.ietf-idr-sr-policy-ifit], [I-D.chen-pce-pcep-ifit].

#### 6. Security Considerations

This document aims to apply a method to perform measurements that does not directly affect Internet security nor applications that run on the Internet. However, implementation of this method must be mindful of security and privacy concerns.

There are two types of security concerns: potential harm caused by the measurements and potential harm to the measurements.

Harm caused by the measurement: Alternate Marking implies modifications on the fly to an Option Header of IPv6 packets by the source node, but this must be performed in a way that does not alter the quality of service experienced by the packets and that preserves stability and performance of routers doing the measurements. As already discussed in Section 4, it is RECOMMENDED that the AltMark Option does not affect the throughput and therefore the user experience.

Harm to the measurement: Alternate Marking measurements could be harmed by routers altering the fields of the AltMark Option (e.g. marking of the packets, FlowMonID) or by a malicious attacker adding AltMark Option to the packets in order to consume the resources of network devices and entities involved. As described above, the source node is the only one that writes the Option Header while the intermediate nodes and destination node only read it without modifying the Option Header. But, for example, an on-path attacker can modify the flags, whether intentionally or accidentally, or deliberately insert a new option to the packet flow or delete the option from the packet flow. The consequent effect could be to give the appearance of loss or delay or invalidate the measurement by modifying option identifiers, such as FlowMonID. The malicious implication can be to cause actions from the network administrator where an intervention is not necessary or to hide real issues in the

network. Since the measurement itself may be affected by network nodes intentionally altering the bits of the AltMark Option or injecting Options headers as a means for Denial of Service (DoS), the Alternate Marking MUST be applied in the context of a controlled domain, where the network nodes are locally administered and this type of attack can be avoided. For this reason, the implementation of the method is not done on the end node if it is not fully managed and does not belong to the controlled domain. Packets generated outside the controlled domain may consume router resources by maliciously using the HbH Option, but this can be mitigated by filtering these packets at the controlled domain boundary. This can be done because, if the end node does not belong to the controlled domain, it is not supposed to add the AltMark HbH Option, and it can be easily recognized.

An attacker that does not belong to the controlled domain can maliciously send packets with AltMark Option. But if Alternate Marking is not supported in the controlled domain, no problem happens because the AltMark Option is treated as any other unrecognized option and will not be considered by the nodes since they are not configured to deal with it, so the only effect is the increased MTU (by 48 bits). While if Alternate Marking is supported in the controlled domain, it is also necessary to avoid that the measurements are affected and external packets with AltMark Option MUST be filtered. As any other Hop-by-Hop Options or Destination Options, it is possible to filter AltMark Options entering or leaving the domain e.g. by using ACL extensions for filtering.

The flow identifier (FlowMonID) composes the AltMark Option together with the two marking bits (L and D). As explained in Section 5.3, there is a chance of collision if the FlowMonID is set pseudo randomly and a solution exists. In general this may not be a problem and a low rate of ambiguous FlowMonIDs can be acceptable, since this does not cause significant harm to the operators or their clients and this harm may not justify the complications of avoiding it. But, for large scale measurements, a big number of flows could be monitored and the probability of a collision is higher, thus the disambiguation of the FlowMonID field can be considered.

The privacy concerns also need to be analyzed even if the method only relies on information contained in the Option Header without any release of user data. Indeed, from a confidentiality perspective, although AltMark Option does not contain user data, the metadata can be used for network reconnaissance to compromise the privacy of users by allowing attackers to collect information about network performance and network paths. AltMark Option contains two kinds of metadata: the marking bits (L and D bits) and the flow identifier (FlowMonID).

The marking bits are the small information that is exchanged between the network nodes. Therefore, due to this intrinsic characteristic, network reconnaissance through passive eavesdropping on data-plane traffic is difficult. Indeed, an attacker cannot gain information about network performance from a single monitoring point. The only way for an attacker can be to eavesdrop on multiple monitoring points at the same time, because they have to do the same kind of calculation and aggregation as Alternate Marking requires.

The FlowMonID field is used in the AltMark Option as the identifier of the monitored flow. It represents a more sensitive information for network reconnaissance and may allow a flow tracking type of attack because an attacker could collect information about network paths.

Furthermore, in a pervasive surveillance attack, the information that can be derived over time is more. But, as further described hereinafter, the application of the Alternate Marking to a controlled domain helps to mitigate all the above aspects of privacy concerns.

At the management plane, attacks can be set up by misconfiguring or by maliciously configuring AltMark Option. Thus, AltMark Option configuration MUST be secured in a way that authenticates authorized users and verifies the integrity of configuration procedures. Solutions to ensure the integrity of AltMark Option are outside the scope of this document. Also, attacks on the reporting of the statistics between the monitoring points and the network management system (e.g. centralized controller) can interfere with the proper functioning of the system. Hence, the channels used to report back flow statistics MUST be secured.

As stated above, the precondition for the application of the Alternate Marking is that it MUST be applied in specific controlled domains, thus confining the potential attack vectors within the network domain. [RFC8799] analyzes and discusses the trend towards network behaviors that can be applied only within a limited domain. This is due to the specific set of requirements especially related to security, network management, policies and options supported which may vary between such limited domains. A limited administrative domain provides the network administrator with the means to select, monitor and control the access to the network, making it a trusted domain. In this regard it is expected to enforce policies at the domain boundaries to filter both external packets with AltMark Option entering the domain and internal packets with AltMark Option leaving the domain. Therefore, the trusted domain is unlikely subject to hijacking of packets since packets with AltMark Option are processed and used only within the controlled domain.



As stated, the application to a controlled domain ensures the control over the packets entering and leaving the domain, but despite that, leakages may happen for different reasons, such as a failure or a fault. In this case, nodes outside the domain MUST simply ignore packets with AltMark Option since they are not configured to handle it and should not process it.

Additionally, it is to be noted that the AltMark Option is carried by the Options Header and it may have some impact on the packet sizes for the monitored flow and on the path MTU, since some packets might exceed the MTU. However, the relative small size (48 bit in total) of these Option Headers and its application to a controlled domain help to mitigate the problem.

It is worth mentioning that the security concerns may change based on the specific deployment scenario and related threat analysis, which can lead to specific security solutions that are beyond the scope of this document. As an example, the AltMark Option can be used as Hop-by-Hop or Destination Option and, in case of Destination Option, multiple administrative domains may be traversed by the AltMark Option that is not confined to a single administrative domain. In this case, the user, aware of the kind of risks, may still want to use Alternate Marking for telemetry and test purposes but the controlled domain must be composed by more than one administrative domains. To this end, the inter-domain links need to be secured (e.g., by IPsec, VPNs) in order to avoid external threats and realize the whole controlled domain.

It might be theoretically possible to modulate the marking or the other fields of the AltMark Option to serve as a covert channel to be used by an on-path observer. This may affect both the data and management plane, but, here too, the application to a controlled domain helps to reduce the effects.

The Alternate Marking application described in this document relies on a time synchronization protocol. Thus, by attacking the time protocol, an attacker can potentially compromise the integrity of the measurement. A detailed discussion about the threats against time protocols and how to mitigate them is presented in [RFC7384]. Network Time Security (NTS), described in [RFC8915], is a mechanism that can be employed. Also, the time, which is distributed to the network nodes through the time protocol, is centrally taken from an external accurate time source, such as an atomic clock or a GPS clock. By attacking the time source it can be possible to compromise the integrity of the measurement as well. There are security measures that can be taken to mitigate the GPS spoofing attacks and a network administrator should certainly employ solutions to secure the network domain.

## 7. IANA Considerations

The Option Type should be assigned in IANA's "Destination Options and Hop-by-Hop Options" registry.

This draft requests the following IPv6 Option Type assignment from the Destination Options and Hop-by-Hop Options sub-registry of Internet Protocol Version 6 (IPv6) Parameters (<https://www.iana.org/assignments/ipv6-parameters/>).

| Hex Value | Binary Value<br>act chg rest |   |     | Description | Reference    |
|-----------|------------------------------|---|-----|-------------|--------------|
| TBD       | 00                           | 0 | tbd | AltMark     | [This draft] |

## 8. Acknowledgements

The authors would like to thank Bob Hinden, Ole Troan, Martin Duke, Lars Eggert, Roman Danyliw, Alvaro Retana, Eric Vyncke, Warren Kumari, Benjamin Kaduk, Stewart Bryant, Christopher Wood, Yoshifumi Nishida, Tom Herbert, Stefano Previdi, Brian Carpenter, Greg Mirsky, Ron Bonica for the precious comments and suggestions.

## 9. References

### 9.1. Normative References

- [I-D.ietf-ippm-rfc8321bis]  
Fioccola, G., Cociglio, M., Mirsky, G., Mizrahi, T., and T. Zhou, "Alternate-Marking Method", draft-ietf-ippm-rfc8321bis-01 (work in progress), April 2022.
- [I-D.ietf-ippm-rfc8889bis]  
Fioccola, G., Cociglio, M., Sapio, A., Sisto, R., and T. Zhou, "Multipoint Alternate-Marking Clustered Method", draft-ietf-ippm-rfc8889bis-01 (work in progress), April 2022.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

## 9.2. Informative References

- [I-D.chen-pce-pcep-ifit]  
Yuan, H., Zhou, T., Li, W., Fioccola, G., and Y. Wang, "Path Computation Element Communication Protocol (PCEP) Extensions to Enable IFIT", draft-chen-pce-pcep-ifit-06 (work in progress), February 2022.
- [I-D.fz-spring-srv6-alt-mark]  
Fioccola, G., Zhou, T., and M. Cociglio, "Segment Routing Header encapsulation for Alternate Marking Method", draft-fz-spring-srv6-alt-mark-02 (work in progress), February 2022.
- [I-D.hinden-6man-hbh-processing]  
Hinden, R. M. and G. Fairhurst, "IPv6 Hop-by-Hop Options Processing Procedures", draft-hinden-6man-hbh-processing-01 (work in progress), June 2021.
- [I-D.ietf-idr-sr-policy-ifit]  
Qin, F., Yuan, H., Zhou, T., Fioccola, G., and Y. Wang, "BGP SR Policy Extensions to Enable IFIT", draft-ietf-idr-sr-policy-ifit-03 (work in progress), January 2022.
- [I-D.peng-v6ops-hbh]  
Peng, S., Li, Z., Xie, C., Qin, Z., and G. Mishra, "Processing of the Hop-by-Hop Options Header", draft-peng-v6ops-hbh-06 (work in progress), August 2021.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <<https://www.rfc-editor.org/info/rfc6437>>.

- [RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <<https://www.rfc-editor.org/info/rfc6438>>.
- [RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <<https://www.rfc-editor.org/info/rfc7045>>.
- [RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <<https://www.rfc-editor.org/info/rfc7384>>.
- [RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.
- [RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <<https://www.rfc-editor.org/info/rfc8799>>.
- [RFC8915] Franke, D., Sibold, D., Teichel, K., Dansarie, M., and R. Sundblad, "Network Time Security for the Network Time Protocol", RFC 8915, DOI 10.17487/RFC8915, September 2020, <<https://www.rfc-editor.org/info/rfc8915>>.

## Authors' Addresses

Giuseppe Fioccola  
Huawei  
Riesstrasse, 25  
Munich 80992  
Germany

Email: [giuseppe.fioccola@huawei.com](mailto:giuseppe.fioccola@huawei.com)

Tianran Zhou  
Huawei  
156 Beiqing Rd.  
Beijing 100095  
China

Email: [zhoutianran@huawei.com](mailto:zhoutianran@huawei.com)

Mauro Cociglio  
Telecom Italia  
Via Reiss Romoli, 274  
Torino 10148  
Italy

Email: [mauro.cociglio@telecomitalia.it](mailto:mauro.cociglio@telecomitalia.it)

Fengwei Qin  
China Mobile  
32 Xuanwumenxi Ave.  
Beijing 100032  
China

Email: [qinfengwei@chinamobile.com](mailto:qinfengwei@chinamobile.com)

Ran Pang  
China Unicom  
9 Shouti South Rd.  
Beijing 100089  
China

Email: [pangran@chinaunicom.cn](mailto:pangran@chinaunicom.cn)

Network Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: 11 November 2022

R. Hinden  
Check Point Software  
G. Fairhurst  
University of Aberdeen  
10 May 2022

IPv6 Minimum Path MTU Hop-by-Hop Option  
draft-ietf-6man-mtu-option-15

Abstract

This document specifies a new IPv6 Hop-by-Hop option that is used to record the minimum Path MTU along the forward path between a source host to a destination host. The recorded value can then be communicated back to the source using the return Path MTU field in the option.

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 <https://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 11 November 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

|   |    |
|---|----|
| 1. Introduction . . . . .   | 2  |
| 1.1. Example Operation . . . . .  | 3  |
| 1.2. Use of the IPv6 Hop-by-Hop Options Header . . . . .                  | 4  |
| 2. Motivation and Problem Solved . . . . .                                | 5  |
| 3. Requirements Language . . . . .  | 6  |
| 4. Applicability Statements . . . . .                                     | 6  |
| 5. IPv6 Minimum Path MTU Hop-by-Hop Option . . . . .                      | 6  |
| 6. Router, Host, and Transport Layer Behaviors . . . . .                  | 8  |
| 6.1. Router Behavior . . . . .  | 8  |
| 6.2. Host Operating System Behavior . . . . .                             | 8  |
| 6.3. Transport Layer Behavior . . . . .                                   | 9  |
| 6.3.1. Including the Option in an Outgoing Packet . . . . .               | 10 |
| 6.3.2. Validation of the Packet that includes the Option . . . . .        | 12 |
| 6.3.3. Receiving the Option . . . . .                                     | 12 |
| 6.3.4. Using the Rtn-PMTU Field . . . . .                                 | 13 |
| 6.3.5. Detecting Path Changes . . . . .                                   | 14 |
| 6.3.6. Detection of Dropping Packets that include the<br>Option . . . . . | 14 |
| 7. IANA Considerations . . . . .  | 14 |
| 8. Security Considerations . . . . .                                      | 14 |
| 8.1. Router Option Processing . . . . .                                   | 15 |
| 8.2. Network Layer Host Processing . . . . .                              | 15 |
| 8.3. Validating use of the Option Data . . . . .                          | 16 |
| 8.4. Direct use of the Rtn-PMTU Value . . . . .                           | 16 |
| 8.5. Using the Rtn-PMTU Value as a Hint for Probing . . . . .             | 17 |
| 8.6. Impact of Middleboxes . . . . .                                      | 17 |
| 9. Experiment Goals . . . . .   | 17 |
| 10. Implementation Status . . . . .                                       | 18 |
| 11. Acknowledgments . . . . .   | 18 |
| 12. Change log [RFC Editor: Please remove] . . . . .                      | 18 |
| 13. References . . . . .  | 21 |
| 13.1. Normative References . . . . .                                      | 21 |
| 13.2. Informative References . . . . .                                    | 22 |
| Appendix A. Examples of Usage . . . . .                                   | 24 |
| Authors' Addresses . . . . .  | 26 |

## 1. Introduction

This document specifies a new IPv6 Hop-by-Hop (HBH) Option to record the minimum Maximum Transmission Unit (MTU) along the forward path between a source and a destination host. The source host creates a packet with this option and initializes the Min-PMTU field with the value of the MTU for the outbound link that will be used to forward the packet towards the destination host.

At each subsequent hop where the option is processed, the router compares the value of the Min-PMTU Field in the option and the MTU of its outgoing link. If the MTU of the link is less than the Min-PMTU, it rewrites the value in the option data with the smaller value. When the packet arrives at the destination host, the host can send the value of the minimum reported MTU for the path back to the source host using the Rtn-PMTU field in the option. The source host can then use this value as input to the method that sets the Path MTU (PMTU) used by upper layer protocols.

The IPv6 Minimum Path MTU Hop-by-Hop (MinPMTU HBH) Option is designed to work with packet sizes that can be specified in the IPv6 header. The maximum packet size that can be specified in an IPv6 header is 65,535 octets ( $2^{16}$ ).

This method has the potential to complete Path MTU discovery in a single round trip time, even over paths that have successive links each with a lower MTU.

The mechanism defined in this document is focused on Unicast, it does not describe Multicast. That is left for future work.

### 1.1. Example Operation

The figure below illustrates the operation of the method. In this case, the path between the source host and the destination host comprises three links, the source has a link MTU of size MTU-S, the link between routers R1 and R2 has an MTU of size 9000 bytes, and the final link to the destination has an MTU of size MTU-D.

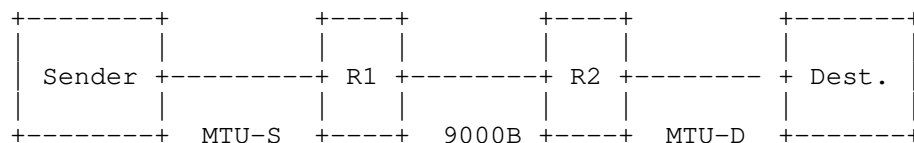


Figure 1

Three scenarios are described:

- \* Scenario 1, considers all links to have an 9000 byte MTU and the method is supported by both routers. The initial Min-PMTU is not modified along the path, and therefore the PMTU is 9000 bytes.
- \* Scenario 2, considers the link between R2 and destination host (MTU-D) to have an MTU of 1500 bytes. This is the smallest MTU, router R2 updates the Min-PMTU to 1500 bytes and the method



correctly updates the PMTU to 1500 bytes. Had there been another smaller MTU at a link further along the path that also supports the method, the lower MTU would also have been detected.

- \* Scenario 3, considers the case where the router preceding the smallest link (R2) does not support the method, and the link to the destination host (MTU-D) has an MTU of 1500 bytes. Therefore, router R2 does not update the Min-PMTU to 1500 bytes. The method then fails to detect the actual PMTU.

In Scenarios 2 and 3, a lower PMTU would also fail to be detected in the case where PMTUD had been used and an ICMPv6 Packet Too Big (PTB) message had not been delivered to the sender [RFC8201].

These scenarios are summarized in the table below. "H" in R1 and/or R2 columns means the router understands the MinPMTU HBH option.

|   | MTU-S | MTU-D | R1 | R2 | Rec PMTU | Note   |
|---|-------|-------|----|----|----------|--|
| 1 | 9000B | 9000B | H  | H  | 9000 B   | Endpoints attempt to use a 9000 B PMTU.  |
| 2 | 9000B | 1500B | H  | H  | 1500 B   | Endpoints attempt to use a 1500 B PMTU.  |
| 3 | 9000B | 1500B | H  | -  | 9000 B   | Endpoints attempt to use a 9000 B PMTU, but need to implement a method to fall back to discover and use a 1500 B PMTU. |

Figure 2

## 1.2. Use of the IPv6 Hop-by-Hop Options Header

IPv6 as specified in [RFC8200] allows nodes to optionally process the Hop-by-Hop header. Specifically, from Section 4:

- \* The Hop-by-Hop Options header is not inserted or deleted, but may be 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. The Hop-by-Hop Options header, when present, must immediately follow the IPv6 header. Its presence is indicated by the value zero in the Next Header field of the IPv6 header.
- \* NOTE: While [RFC2460] required that all nodes must examine and process the Hop-by-Hop Options header, it is now expected that nodes along a packet's delivery path only examine and process the Hop-by-Hop Options header if explicitly configured to do so.

The Hop-by-Hop Option defined in this document is designed to take advantage of this property of how Hop-by-Hop options are processed. Nodes that do not support this Option SHOULD ignore them. This can mean that the Min-PMTU value does not account for all links along a path.

## 2. Motivation and Problem Solved

The current state of Path MTU Discovery on the Internet is problematic. The mechanisms defined in [RFC8201] are known to not work well in all environments. It fails to work in various cases, including when nodes in the middle of the network do not send ICMPv6 PTB messages, or rate-limited ICMPv6 messages, or do not have a return path to the source host.

This results in many transport layer connections being configured to use smaller packets (e.g., 1280 bytes) by default and makes it difficult to take advantage of paths with a larger PMTU where they do exist. Applications that send large packets are forced to use IPv6 Fragmentation [RFC8200], which can reduce the reliability of Internet communication [RFC8900].

Encapsulations and network-layer tunnels further reduce the payload size available for a transport protocol to use. Also, some use-cases increase packet overhead, for example, Network Virtualization Using Generic Routing Encapsulation (NVGRE) [RFC7637] encapsulates L2 packets in an outer IP header and does not allow IP Fragmentation.

Sending larger packets can improve host performance, e.g., avoiding limits to packet processing by the packet rate. For example, the packet per second rate required to reach wire speed on a 10G link with 1280 byte packets is about 977K packets per second (pps), vs. 139K pps for 9000 byte packets.

The purpose of this document is to improve the situation by defining a mechanism that does not rely on reception of ICMPv6 Packet Too Big messages from nodes in the middle of the network. Instead, this provides information to the destination host about the minimum Path MTU, and sends this information back to the source host. This is expected to work better than the current RFC8201-based mechanisms.

A similar mechanism was proposed in 1988 for IPv4 in [RFC1063] by Jeff Mogul, C. Kent, Craig Partridge, and Keith McCloghrie. It was later obsoleted in 1990 by [RFC1191], the current deployed approach to Path MTU Discovery. In contrast, the method described in this document uses the Hop-by-Hop option of IPv6. It does not replace PMTUD [RFC8201], PLPPMTUD [RFC4821] or Datagram PLPMTUD [RFC8899], but rather is designed to compliment these methods.

### 3. 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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

### 4. Applicability Statements

The Path MTU option is designed for environments where there is control over the hosts and nodes that connect them, and where there is more than one MTU size in use. For example, in Data Centers and on paths between Data Centers, to allow hosts to better take advantage of a path that is able to support a large PMTU.

The design of the option is sufficiently simple that it can be executed on a router's fast path. A successful experiment depends on both implementation by host and router vendors and deployment by operators. The contained use-case of connections within and between Data Centers could be a driver for deployment.

The method could also be useful in other environments, including the general Internet, and offers advantage when this Hop-by-Hop Option is supported on all paths. The method is more robust when used to probe the path using packets that do not carry application data and when also paired with a method such as Packetization Layer PMTUD [RFC4821] or Datagram PLPMTUD [RFC8899].

### 5. IPv6 Minimum Path MTU Hop-by-Hop Option

The Minimum Path MTU Hop-by-Hop Option has the following format:

| Option<br>Type | Option<br>Data Len | Option<br>Data |          |   |
|----------------|--------------------|----------------|----------|---|
| BBCTTTTT       | 00000100           | Min-PMTU       | Rtn-PMTU | R |

Option Type (see Section 4.2 of [RFC8200]):

BB 00 Skip over this option and continue processing.

C 1 Option data can change en route to the packet's final destination.

TTTTT 10000 Option Type assigned from IANA [IANA-HBH].

Length: 4 The size of the value field in Option Data field supports PMTU values from 0 to 65,534 octets, the maximum size represented by the Path MTU option.

Min-PMTU: n 16-bits. The minimum MTU recorded along the path in octets, reflecting the smallest link MTU that the packet experienced along the path. A value less than the IPv6 minimum link MTU [RFC8200] MUST be ignored.

Rtn-PMTU: n 15-bits. The returned Path MTU field, carrying the 15 most significant bits of the latest received Min-PMTU field for the forward path. The value zero means that no Reported MTU is being returned.

R n 1-bit. R-Flag. Set by the source to signal that the destination host should include the received Rtn-PMTU field updated by the reported Min-PMTU value when the destination host is to send a PMTU Option back to the source host.

Figure 3

NOTE: The encoding of the final two octets (Rtn-PMTU and R-Flag) could be implemented by a mask of the latest received Min-PMTU value with 0xFFFE, discarding the right-most bit and then performing a logical 'OR' with the R-Flag value of the sender. This encoding fits in the minimum-sized Hop-by-Hop Option header.

## 6. Router, Host, and Transport Layer Behaviors

### 6.1. Router Behavior

Routers that are not configured to support Hop-by-Hop Options are not expected to examine or process the contents of this option [RFC8200].

Routers that support Hop-by-Hop Options, but are not configured to support this option SHOULD skip over this option and continue to processing the header [RFC8200].

Routers that support this option MUST compare the value of the Min-PMTU field with the MTU configured for the outgoing link. If the MTU of the outgoing link is less than the Min-PMTU, the router rewrites the Min-PMTU in the Option to use the smaller value. (The router processing is performed without checking the valid range of the Min-PMTU or the Rtn-PMTU fields.)

A router MUST ignore and MUST NOT change the Rtn-PMTU field or the R-Flag in the option.

### 6.2. Host Operating System Behavior

The PMTU entry associated with the destination in the host's destination cache [RFC4861] SHOULD be updated after detecting a change using the IPv6 Minimum Path MTU Hop-by-Hop Option. This cached value can be used by other flows that share the host's destination cache.

The value in the host destination cache SHOULD be used by PLPMTUD to select an initial PMTU for a flow. The cached PMTU is only increased by PLPMTUD when the Packetization Layer determines the path actually supports a larger PMTU [RFC4821] [RFC8899].

When requested to send an IPv6 packet with the MinPMTU HBH option, the source host includes the option in an outgoing packet. The source host MUST fill the Min-PMTU field with the MTU configured for the link over which it will send the packet on the next hop towards the destination host.

When a host includes the option in a packet it sends, the host SHOULD set the Rtn-PMTU field to the previously cached value of the received Minimum Path MTU for the flow in the Rtn-PMTU field (see Section 6.3.3). If this value is not set (for example, because there is no cached reported Min-PMTU value), the Rtn-PMTU field value MUST be set to zero.

The source host MAY request the destination host to return the reported Min-PMTU value by setting the R-Flag in the option of an outgoing packet. The R-Flag SHOULD NOT be set when the MinPMTU HBH Option was sent solely to provide requested feedback on the return Path MTU to avoid each response generating another response.

The destination host controls when to send a packet with this option in response to an R-flag, as well as which packets to include it in. The destination host MAY limit the rate at which it sends these packets.

A destination host only sets the R Flag if it wishes the source host to also return the discovered PMTU value for the path from the destination to the source.

The normal sequence of operation of the R-Flag using the terminology from the diagram in Figure 1 is:

1. The source sends a probe to the destination. The sender sets the R-Flag.
2. The destination responds by sending a probe including the received Min-PMTU as the Rtn-PMTU. A destination that does not wish to probe the return path sets the R-Flag to 0.

### 6.3. Transport Layer Behavior

This Hop-by-Hop option is intended to be used with a path MTU discovery method.

PLPMTUD [RFC9000] uses probe packets for two distinct functions:

- \* Probe packets are used to confirm connectivity. Such probes can be of any size up to the PLPMTU. These probe packets are sent to solicit a response use the path to the remote node. These probe packets can carry the Hop-by-Hop PMTU option, providing the final size of the packet does not exceed the current PLPMTU. After validating that the packet originates from the path (section 4.6.1), the PLPMTUD method can use the reported size from the Hop-by-Hop option as the next search point when it resumes the search algorithm. (This use resembles the use of the PTB\_SIZE information in section 4.6.2 of [RFC8899])
- \* A second use of probe packets is to explore if a path supports a packet size greater than the current PLPMTU. If this probe packet is successfully delivered (as determined by the source host), then the PLPMTU is raised to the size of the successful probe. These probe packets do not usually set the Path MTU Hop-by-Hop option.

See section 1.2 of [RFC8899]. Section 4.1 of [RFC8899] also describes ways that a Probe Packet can be constructed, depending on whether the probe packets carry application data.

- \* The PMTU Hop-by-Hop Option Probe can be sent on packets that include application data, but needs to be robust to potential loss of the packet (i.e., with the possibility that retransmission might be needed if the packet is lost).
- \* Using a PMTU Probe on packets that do not carry application data will avoid the need for loss recovery if a router on the path drops packets that set this option. (This avoids the transport needing to retransmit a lost packet that includes this option.) This is the normal default format for both uses of probes.

#### 6.3.1. Including the Option in an Outgoing Packet

The upper layer protocol can request the MinPMTU HBH option to be included in an outgoing IPv6 packet. A transport protocol (or upper layer protocol) can include this option only on specific packets used to test the path. This option does not need to be included in all packets belonging to a flow.

NOTE: Including this option in a large packet (e.g., one larger than the present PMTU) is not likely to be useful, since the large packet would itself be dropped by any link along the path with a smaller MTU, preventing the Min-PMTU information from reaching the destination host.

Discussion:

- \* In the case of TCP, the option could be included in a packet that carries a TCP segment sent after the connection is established. A segment without data could be used, to avoid the need to retransmit this data if the probe packet is lost. The discovered value can be used to inform PLPMTUD [RFC4821].

NOTE: A TCP SYN can also negotiate the Maximum Segment Size (MSS), which acts as an upper limit to the packet size that can be sent by a TCP sender. If this option were to be included in a TCP SYN, it could increase the probability that the SYN segment is lost when routers on the path drop packets with this option (see Section 6.3.6), which could have an unwanted impact on the result of racing options [I-D.ietf-taps-arch] or feature negotiation.

- \* The use with datagram transport protocols (e.g., UDP) is harder to characterize because applications using datagram transports range from very short-lived (low data-volume applications) exchanges, to longer (bulk) exchanges of packets between the source and destination hosts [RFC8085].
- \* Simple-exchange protocols (i.e., low data-volume applications [RFC8085] that only send one or a few packets per transaction), might assume that the PMTU is symmetrical. That is, the PMTU is the same in both directions, or at least not smaller for the return path. This optimization does not hold when the paths are not symmetric.
- \* The MinPMTU HBH option can be used with ICMPv6 [RFC4443]. This requires a response from the remote node and therefore is restricted to use with ICMPv6 echo messages. The MinPMTU HBH option could provide additional information about the PMTU that might be supported by a path. This could be use as a diagnostic tool to measure the PMTU of a path. As with other uses, the actual supported PMTU is only confirmed after receiving a response to a subsequent probe of the PMTU size.
- \* A datagram transport can utilise DPLPMTUD [RFC8899]. For example, QUIC (see section 14.3 of [RFC9000]), can use DPLPMTUD to determine whether the path to a destination will support a desired maximum datagram size. When using the IPv6 MinPMTU HBH option, the option could be added to an additional QUIC PMTU Probe that is of minimal size (or one no larger than the currently supported PMTU size). Once the return Path MTU value in the MinPMTU HBH option has been learned, DPLPMTUD can be triggered to test for a larger PLPMTU using an appropriately sized PLPMTU Probe Packet (see section 5.3.1 of [RFC8899]).
- \* The use of this option with DNS and DNSSEC over UDP is expected to work for paths where the PMTU is symmetric. The DNS server will learn the PMTU from the DNS query messages. If the Rtn-PMTU value is smaller, then a large DNSSEC response might be dropped and the known problems with PMTUD will then occur. DNS and DNSSEC over transport protocols that can carry the PMTU ought to work.
- \* This method also can be used with Anycast to discover the PMTU of the path, but the use needs to be aware that the Anycast binding might change.



### 6.3.2. Validation of the Packet that includes the Option

An upper layer protocol (e.g., transport endpoint) using this option needs to provide protection from data injection attacks by off-path devices [RFC8085]. This requires a method to assure that the information in the Option Data is provided by a node on the path. This validates that the packet forms a part of an existing flow, using context available at the upper layer. For example, a TCP connection or UDP application that maintains the related state and uses a randomized ephemeral port would provide this basic validation to protect from off-path data injection, see Section 5.1 of [RFC8085]. IPsec [RFC4301] and TLS [RFC8446] provide greater assurance.

The upper layer discards any received packet when the packet validation fails. When packet validation fails, the upper layer **MUST** also discard the associated Option Data from the MinPMTU HBH option without further processing.

### 6.3.3. Receiving the Option

For a connection-oriented upper layer protocol, caching of the received Min-PMTU could be implemented by saving the value in the connection context at the transport layer. A connection-less upper layer (e.g., one using UDP), requires the upper layer protocol to cache the value for each flow it uses.

A destination host that receives a MinPMTU HBH Option with the R-Flag **SHOULD** include the MinPMTU HBH option in the next outgoing IPv6 packet for the corresponding flow.

A simple mechanism could only include this option (with the Rtn-PMTU field set) the first time this option is received or when it notifies a change in the Minimum Path MTU. This limits the number of packets including the option packets that are sent. However, this does not provide robustness to packet loss or recovery after a sender loses state.

#### Discussion:

- \* Some upper layer protocols send packets less frequently than the rate at which the host receives packets. This provides less frequent feedback of the received Rtn-PMTU value. However, a host always sends the most recent Rtn-PMTU value.

#### 6.3.4. Using the Rtn-PMTU Field

The Rtn-PMTU field provides an indication of the PMTU from on-path routers. It does not necessarily reflect the actual PMTU between the source and destination hosts. Care therefore needs to be exercised in using the Rtn-PMTU value. Specifically:

- \* The actual PMTU can be lower than the Rtn-PMTU value because the Min-PMTU field was not updated by a router on the path that did not process the option.
- \* The actual PMTU may be lower than the Rtn-PMTU value because there is a layer-2 device with a lower MTU.
- \* The actual PMTU may be larger than the Rtn-PMTU value because of a corrupted, delayed or mis-ordered response. A source host **MUST** ignore a Rtn-PMTU value larger than the MTU configured for the outgoing link.
- \* The path might have changed between the time when the probe was sent and when the Rtn-PMTU value received.

IPv6 requires that every link in the Internet have an MTU of 1280 octets or greater. A node **MUST** ignore a Rtn-PMTU value less than 1280 octets [RFC8200].

To avoid unintentional dropping of packets that exceed the actual PMTU (e.g., Scenario 3 in Section 1.1), the source host can delay increasing the PMTU until a probe packet with the size of the Rtn-PMTU value has been successfully acknowledged by the upper layer, confirming that the path supports the larger PMTU. This probing increases robustness, but adds one additional path round trip time before the PMTU is updated. This use resembles that of PTB messages in section 4.6 of DPLPMTUD [RFC8899] (with the important difference that a PTB message can only seek to lower the PMTU, whereas this option could trigger a probe packet to seek to increase the PMTU.)

Section 5.2 of [RFC8201] provides guidance on the caching of PMTU information and also the relation to IPv6 flow labels. Implementations should consider the impact of Equal Cost Multipath (ECMP) [RFC6438]. Specifically, whether a PMTU ought to be maintained for each transport endpoint, or for each network address.

### 6.3.5. Detecting Path Changes

Path characteristics can change and the actual PMTU could increase or decrease over time. For instance, following a path change when packets are forwarded over a link with a different MTU than that previously used. To bound the delay in discovering an increase in the actual PMTU, a host with a link MTU larger than the current PMTU SHOULD periodically send the MinPMTU HBH Option with the R-bit set. DPLPMTUD provides recommendations concerning how this could be implemented (see Section 5.3 of [RFC8899]). Since the option consumes less capacity than a full-sized probe packet, there can be advantage in using this to detect a change in the path characteristics.

### 6.3.6. Detection of Dropping Packets that include the Option

There is evidence that some middleboxes drop packets that include Hop-by-Hop options. For example, a firewall might drop a packet that carries an unknown extension header or option. This practice is expected to decrease as an option becomes more widely used. It could result in generation of an ICMPv6 message indicating the problem. This could be used to (temporarily) suspend use of this option.

A middlebox that silently discards a packet with this option results in dropping of any packet using the option. This dropping can be avoided by appropriate configuration in a controlled environment, such as within a data centre, but needs to be considered for Internet usage. Section 6.2 recommends that this option is not used on packets where loss might adversely impact performance.

## 7. IANA Considerations

IANA has assigned and registered an IPv6 Hop-by-Hop Option type with Temporary status from the "Destination Options and Hop-by-Hop Options" registry [IANA-HBH]. This assignment is shown in Section 5.

IANA is requested to update this registry to point to this document and remove the Temporary status.

## 8. Security Considerations

This section discusses the security considerations. It first reviews router option processing. It then reviews host processing when receiving this option at the network layer. It then considers two ways in which the Option Data can be processed, followed by two approaches for using the Option Data. Finally, it discusses middlebox implications related to use in the general Internet.

### 8.1. Router Option Processing

This option shares the characteristics of all other IPv6 Hop-by-Hop Options, in that if not supported at line rate it could be used to degrade the performance of a router. This option, while simple, is no different to other uses of IPv6 Hop-by-Hop options.

It is common for routers to ignore the Hop-by-Hop Option header or drop packets containing a Hop-by-Hop Option header. Routers implementing IPv6 according to [RFC8200] only examine and process the Hop-by-Hop Options header if explicitly configured to do so.

### 8.2. Network Layer Host Processing

A malicious attacker can forge a packet directed at a host that carries the MinPMTU HBH option. By design, the fields of this IP option can be modified by the network.

For comparison, the ICMPv6 Packet Too Big message used in [RFC8201] Path MTU Discovery, the source host has an inherent trust relationship with the destination host including this option. This trust relationship can be used to help verify the option. ICMPv6 Packet Too Big messages are sent from any router on the path to the destination host, the source host has no prior knowledge of these routers (except for the first hop router).

Reception of this packet will require processing as the network stack parses the packet before the packet is delivered to the upper layer protocol. This network layer option processing is normally completed before any upper layer protocol delivery checks are performed.

The network layer does not normally have sufficient information to validate that the packet carrying an option originated from the destination (or an on-path node). It also does not typically have sufficient context to demultiplex the packet to identify the related transport flow. This can mean that any changes resulting from reception of the option applies to all flows between a pair of endpoints.

These considerations are no different to other uses of Hop-by-Hop options, and this is the use case for PMTUD. The following section describes a mitigation for this attack.

### 8.3. Validating use of the Option Data

Transport protocols should be designed to provide protection from data injection attacks by off-path devices and mechanisms should be described in the Security Considerations for each transport specification (see Section 5.1 of the UDP Guidelines [RFC8085]). For example, a TCP or UDP application that maintains the related state and uses a randomized ephemeral port would provide basic protection. TLS [RFC8446] or IPsec [RFC4301] provide cryptographic authentication. An upper layer protocol that validates each received packet discards any packet when this validation fails. In this case, the host **MUST** also discard the associated Option Data from the MinPMTU HBH option without further processing (Section 6.3).

A network node on the path has visibility of all packets it forwards. By observing the network packet payload, the node might be able to construct a packet that might be validated by the destination host. Such a node would also be able to drop or limit the flow in other ways that could be potentially more disruptive. Authenticating the packet, for example, using IPsec [RFC4301] or TLS [RFC8446] mitigates this attack. Note that AH style authentication [RFC4302] while authenticating the payload and outer IPv6 header, does not check Hop-by-Hop options that change on route.

### 8.4. Direct use of the Rtn-PMTU Value

The simplest way to utilize the Rtn-PMTU value is to directly use this to update the PMTU. This approach results in a set of security issues when the option carries malicious data:

- \* A direct update of the PMTU using the Rtn-PMTU value could result in an attacker inflating or reducing the size of the host PMTU for the destination. Forcing a reduction in the PMTU can decrease the efficiency of network use, might increase the number of packets/fragments required to send the same volume of payload data, and prevents sending an unfragmented datagram larger than the PMTU. Increasing the PMTU can result in black-holing (see Section 1.1 of [RFC8899]) when the source host sends packets larger than the actual PMTU. This persists until the PMTU is next updated.
- \* The method can be used to solicit a response from the destination host. A malicious attacker could forge a packet that causes the destination to add the option to a packet sent to the source host. A forged value of Rtn-PMTU in the Option Data might also impact the remote endpoint, as described in the previous bullet. This persists until a valid MinPMTU HBH option is received. This attack could be mitigated by limiting the sending of the MinPMTU HBH option in reply to incoming packets that carry the option.

### 8.5. Using the Rtn-PMTU Value as a Hint for Probing

Another way to utilize the Rtn-PMTU value is to indirectly trigger a probe to determine if the path supports a PMTU of size Rtn-PMTU. This approach needs context for the flow, and hence assumes an upper layer protocol that validates the packet that carries the option (see Section 8.3). This is the case when used in combination with DPLPMTUD [RFC8899]. A set of security considerations result when an option carries malicious data:

- \* If the forged packet carries a validated option with a non-zero Rtn-PMTU field, the upper layer protocol could utilize the information in the Rtn-PMTU field. A Rtn-PMTU larger than the current PMTU can trigger a probe for a new size.
- \* If the forged packet carries a non-zero Min-PMTU field, the upper layer protocol would change the cached information about the path from the source. The cached information at the destination host will be overwritten when the host receives another packet that includes a MinPMTU HBH option corresponding to the flow.
- \* Processing of the option could cause a destination host to add the MinPMTU HBH option to a packet sent to the source host. This option will carry a Rtn-PMTU value that could have been updated by the forged packet. The impact of the source host receiving this resembles that discussed previously.

### 8.6. Impact of Middleboxes

There is evidence that some middleboxes drop packets that include Hop-by-Hop options. For example, a firewall might drop a packet that carries an unknown extension header or option. This practice is expected to decrease as the option becomes more widely used. Methods to address this are discussed in Section 6.3.6.

When a forged packet causes a packet to be sent including the MinPMTU HBH option, and the return path does not forward packets with this option, the packet will be dropped Section 6.3.6. This attack is mitigated by validating the option data before use and by limiting the rate of responses generated. An upper layer could further mitigate the impact by responding to an R-Flag by including the option in a packet that does not carry application data.

## 9. Experiment Goals

This section describes the experimental goals of this specification.

A successful deployment of the method depends upon several components being implemented and deployed:

- \* Support in the sending node (see Section 6.2). This also requires corresponding support in upper layer protocols (see Section 6.3).
- \* Router support in nodes (see Section 6.1). The IETF continues to provide recommendations on the use of IPv6 Hop-by-Hop options, for example Section 2.2.2 of [RFC9099]. This document does not update the way router implementations configure support for Hop-by-Hop options.
- \* Support in the receiving node (see Section 6.3.3).

Experience from deployment is an expected input to any decision to progress this specification from Experimental to IETF Standards Track. Appropriate inputs might include:

- \* Reports of implementation experience;
- \* Measurements of the number paths where the method can be used;
- \* Measurements showing the benefit realized or the implications of using specific methods over specific paths.

## 10. Implementation Status

At the time this document was published there are two known implementations of the Path MTU Hop-by-Hop option. These are:

- \* Wireshark dissector. This is shipping in production in Wireshark version 3.2 [WIRESHARK].
- \* A prototype in the open source version of the FD.io Vector Packet Processing (VPP) technology [VPP]. At the time this document was published, the source code can be found [VPP\_SRC].

## 11. Acknowledgments

Helpful comments were received from Tom Herbert, Tom Jones, Fred Templin, Ole Troan, Tianran Zhou, Jen Linkova, Brian Carpenter, Peng Shuping, Mark Smith, Fernando Gont, Michael Dougherty, Erik Kline, and other members of the 6MAN working group.

## 12. Change log [RFC Editor: Please remove]

draft-ietf-6man-mtu-option-15, 2022-May-10

- \* Correcting an editing mistake in Appendix A.
- \* Editorial Change.

draft-ietf-6man-mtu-option-14, 2022-April-15

- \* Area Director Reviews:
  - Lars Eggert's Review: Fixed "nits".
  - Eric Vyncke's Review: Added that this work is focused on Unicast, removed Discussion from Section 6.1, revised text on PLPMTUD probing, changed SHOULD to MUST in Section 6.3.4, and fixed several NITs.
  - Alvaro Retana's Review: Changed SHOULD language to more general text in Section 6.1
  - ARTART Review: Added new Appendix "Examples of Usage" with diagrams showing examples of use.
  - Zaheduzzaman Sarker's Review: Fixed some editorial issues, and updated SHOULD language.
- \* Editorial Changes.

draft-ietf-6man-mtu-option-13, 2022-February-28

- \* Area Directorate Reviews:
  - SECDIR Review: Fixed "nit".
  - TSVART Review: Restructured Section 6 including making Transport Behavior more prominent, added text about ICMPv6 to Section 6.3.1, moved the text about prior work in RFC1063 to Section 2.
  - GENART Review: Added text to Section 1 that this option was designed to work with packet sizes that can be specified in the IPv6 Header.
- \* Editorial Changes.

draft-ietf-6man-mtu-option-12, 2022-January-26

- \* Clarified a few issues raised by AD review by Erik Kline AD review.

draft-ietf-6man-mtu-option-11, 2021-September-30

- \* Clarifications and editorial changes to the Security Considerations section based on early AD review by Erik Kline.

draft-ietf-6man-mtu-option-10, 2021-September-27

- \* Clarifications and editorial changes based on second chair review by Ole Troan.
- \* Editorial changes.



draft-ietf-6man-mtu-option-09, 2021-September-23

- \* Clarifications and editorial changes based on review by Michael Dougherty.

draft-ietf-6man-mtu-option-08, 2021-September-7

- \* Clarifications and editorial changes based on chair review by Ole Troan.
- \* Correction and clarifications based on review by Fernando Gont.

draft-ietf-6man-mtu-option-07, 2021-August-31

- \* Added Experiment Goals section.
- \* Added Implementation Status section.
- \* Updated the IANA Considerations section to point to this document and remove Temporary status.
- \* Clarifications and editorial changes based on review by Mark Smith.

draft-ietf-6man-mtu-option-06, 2021-August-7

- \* Transport usage of the mechanism clarified in response to feedback and suggestions from Jen Linkova.
- \* Restructured Section 6 to improve readability.
- \* Editorial changes.

draft-ietf-6man-mtu-option-05, 2021-April-28

- \* Editorial changes.

draft-ietf-6man-mtu-option-04, 2020-Oct-23

- \* Fixes for typos.

draft-ietf-6man-mtu-option-03, 2020-Sept-14

- \* Rewrite to make text and terminology more consistent.
- \* Added the notion of validating the packet before use of the HBH option data.
- \* Method aligned with the way common APIs send/receive HBH option data.
- \* Added reference to DPLPMTUD and clarified upper layer usage.
- \* Completed security considerations section.

draft-ietf-6man-mtu-option-02, 2020-March-9

- \* Editorial changes to make text and terminology more consistent.

- \* Added reference to DPLPMTUD.

draft-ietf-6man-mtu-option-01, 2019-September-13

- \* Changes to show IANA assigned code point.
- \* Editorial changes to make text and terminology more consistent.
- \* Added a reference to RFC8200 in Section 2 and a reference to RFC6438 in Section 6.3.

draft-ietf-6man-mtu-option-00, 2019-August-9

- \* First 6man w.g. draft version.
- \* Changes to request IANA allocation of code point.
- \* Editorial changes.

draft-hinden-6man-mtu-option-02, 2019-July-5

- \* Changed option format to also include the Returned PMTU value and Return flag and made related text changes in Section 6.2 to describe this behavior.
- \* ICMPv6 Packet Too Big messages are no longer used for feedback to the source host.
- \* Added to Acknowledgements Section that a similar mechanism was proposed for IPv4 in 1988 in [RFC1063].
- \* Editorial changes.

draft-hinden-6man-mtu-option-01, 2019-March-05

- \* Changed requested status from Standards Track to Experimental to allow use of experimental option type (11110) to allow for experimentation. Removed request for IANA Option assignment.
- \* Added Section 2 "Motivation and Problem Solved" section to better describe what the purpose of this document is.
- \* Added appendix describing planned experiments and how the results will be measured.
- \* Editorial changes.

draft-hinden-6man-mtu-option-00, 2018-Oct-16

- \* Initial draft.

## 13. References

### 13.1. Normative References

[IANA-HBH] "Destination Options and Hop-by-Hop Options",  
<<https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#ipv6-parameters-2>>.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <<https://www.rfc-editor.org/info/rfc8201>>.

### 13.2. Informative References

- [I-D.ietf-taps-arch] Pauly, T., Trammell, B., Brunstrom, A., Fairhurst, G., and C. Perkins, "An Architecture for Transport Services", Work in Progress, Internet-Draft, draft-ietf-taps-arch-12, 3 January 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-taps-arch-12>>.
- [RFC1063] Mogul, J., Kent, C., Partridge, C., and K. McCloghrie, "IP MTU discovery options", RFC 1063, DOI 10.17487/RFC1063, July 1988, <<https://www.rfc-editor.org/info/rfc1063>>.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <<https://www.rfc-editor.org/info/rfc1191>>.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<https://www.rfc-editor.org/info/rfc2460>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.

- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <<https://www.rfc-editor.org/info/rfc4821>>.
- [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, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <<https://www.rfc-editor.org/info/rfc6438>>.
- [RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", RFC 7637, DOI 10.17487/RFC7637, September 2015, <<https://www.rfc-editor.org/info/rfc7637>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8899] Fairhurst, G., Jones, T., Tüxen, M., Rüngeler, I., and T. Völker, "Packetization Layer Path MTU Discovery for Datagram Transports", RFC 8899, DOI 10.17487/RFC8899, September 2020, <<https://www.rfc-editor.org/info/rfc8899>>.
- [RFC8900] Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", BCP 230, RFC 8900, DOI 10.17487/RFC8900, September 2020, <<https://www.rfc-editor.org/info/rfc8900>>.
- [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <<https://www.rfc-editor.org/info/rfc9000>>.

- [RFC9099] Vyncke, É., Chittimaneni, K., Kaeo, M., and E. Rey,  
"Operational Security Considerations for IPv6 Networks",  
RFC 9099, DOI 10.17487/RFC9099, August 2021,  
<<https://www.rfc-editor.org/info/rfc9099>>.
- [VPP] "VPP/What is VPP?",  
<[https://wiki.fd.io/view/VPP/What\\_is\\_VPP%3F](https://wiki.fd.io/view/VPP/What_is_VPP%3F)>.
- [VPP\_SRC] "VPP Source", <<https://gerriet.fd.io/r/c/vpp/+21948>>.
- [WIRESHARK] "Wireshark Network Protocol Analyzer",  
<<https://www.wireshark.org>>.

#### Appendix A. Examples of Usage

This section provides examples that illustrate a use of the MinPMTU HBH option by a source using DPLPMTUD to discover the PLPMTU supported by a path. They consider a path where the on-path router has been configured with an outgoing MTU of  $d'$ . The source starts by transmission of packets of size  $a$ , and then uses DPLPMTUD to seek to increase the size in steps resulting in sizes of  $b, c, d, e$ , etc., (chosen by the search algorithm used by DPLPMTUD). The search algorithm terminates with a PLPMTU that is at least  $d$  and is less than or equal to  $d'$ .

The first example considers DPLPMTUD without using the MinPMTU HBH option. In this case, DPLPMTUD searches using an increasing size of probe packet. Probe packets of size  $(e)$  are sent, which are larger than the actual PMTU. In this example, PTB messages are not received from the routers and repeated unsuccessful probes result in the search phase completing. Packets of data are never sent with a size larger than the size of the last confirmed probe packet. ACKs of data packets are not shown.

```

----Packets of data size (a) ----->
----Probe size (b) ----->
<----- ACK of probe -----
----Packets of data size (b) ----->
----Probe size (c) ----->
<----- ACK of probe -----
----Packets of data size (c) ----->
----Probe size (d) ----->
<----- ACK of probe -----
----Packets of data size (d) ----->
<----- ACK of probe -----
...
----Probe size (e) -----X
      X----ICMPv6 PTB (d') --|
----Packets of data size (d) ----->
----Probe size (e) -----X (again)
      X----ICMPv6 PTB (d') --|
----Packets of data size (d) -----
...
etc, until MaxProbes are unsuccessful and search phase completes.
----Packets of data size (d) ----->

```

Figure 4

The second example considers DPLPMTUD with the MinPMTU HBH option set on a connectivity probe packet.

The IPv6 option is sent end-to-end, and the Min-PMTU is updated by a router on the path to d', which is returned in a response that also sets the MinPMTU HBH option. Upon receiving Rtn-PMTU value is received, DPLPMTUD immediately sends a probe packet of the target size (d'). If the probe packet is confirmed for the path, the PLPMTU is updated, allowing the source to use data packets up to size d'. (The search algorithm is allowed to continue to probe to see if the path supports a larger size.) Packets of data are never sent with a size larger than the last confirmed probe size, d'.

```

----Packets of data size (a) ----->
----Connectivity probe with MinPMTU-
      +--updated to minPMTU=d'----->
<-----ACK with Rtn-PMTU=d'-----
----Packets of data size (a) ----->
----Probe size (d') ----->
<----- ACK of probe -----
----Packets of data size (d') ----->
Search phase completes.
----Packets of data size (d') ----->

```

Figure 5

The final example considers DPLPMTUD with the MinPMTU HBH option set on a connectivity probe packet, but shows the effect when this connectivity probe packet is dropped.

In this case, the packet with the MinPMTU HBH option is not received. DPLPMTUD searches using probe packets of increasing size, increasing the PLPMTU when the probes are confirmed. An ICMPv6 PTB message is received when the probed size exceeds the actual PMTU, indicating a PTB\_SIZE of d'. DPLPMTUD immediately sends a probe packet of the target size (d'). If the probe packet is confirmed for the path, the PLPMTU is updated, allowing the source to use data packets up to size d'. If the ICMPv6 PTB message is not received, the DPLPMTU will be the last confirmed probe size, d.

```

----Packets of data size (a) ----->
----Connectivity probe with MinPMTU -----X
----Packets of data size (a) ----->
----Probe size (b) ----->
<----- ACK of probe -----
----Packets of data size (b) ----->
----Probe size (c) ----->
<----- ACK of probe -----
----Packets of data size (c) ----->
----Probe size (d) ----->
<----- ACK of probe -----
----Packets of data size (d) ----->
----Probe size (e) -----X
<--ICMPv6 PTB PTB_SIZE(d') -|
----Packets of data size (d) ----->
----Probe size (d') using target set by PTB_SIZE ----->
<----- ACK of probe -----
Search phase completes.
----Packets of data size (d') ----->

```

Figure 6

The number of probe rounds depends on the number of steps needed by the search algorithm, and is typically larger for a larger PMTU.

#### Authors' Addresses

Robert M. Hinden  
 Check Point Software  
 959 Skyway Road  
 San Carlos, CA 94070  
 United States of America

Email: bob.hinden@gmail.com

Godred Fairhurst  
University of Aberdeen  
School of Engineering  
Fraser Noble Building  
Aberdeen  
AB24 3UE  
United Kingdom  
Email: gorrry@erg.abdn.ac.uk



6MAN Working Group  
Internet-Draft  
Updates: RFC5014, RFC6724 (if approved)  
Intended status: Standards Track  
Expires: May 18, 2021

D. Mudric  
Ciena  
A. Petrescu  
CEA, LIST  
November 14, 2020

Least-Common Scope Communications  
draft-mudric-6man-lcs-02

Abstract

This draft formulates a security problem statement. The problem arises when a Host uses its Global Unicast Address (GUA) to communicate with another Host situated on the same link.

To address this problem, we suggest to select and use addresses of a least scope that are common.

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 <https://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 May 18, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|  |   |
|--|---|
| 1. Terminology . . . . .                             | 2 |
| 2. Problem Statement . . . . .                       | 2 |
| 3. Least Common Scope Communications . . . . .       | 3 |
| 4. LL Address Resolution . . . . .                   | 3 |
| 5. Sending algorithm with LL Address . . . . .       | 7 |
| 6. Other Issues with LL Address Resolution . . . . . | 8 |
| 7. Security Considerations . . . . .                 | 8 |
| 8. IANA Considerations . . . . .                     | 8 |
| 9. Contributors . . . . .                            | 8 |
| 10. Acknowledgements . . . . .                       | 9 |
| 11. Normative References . . . . .                   | 9 |
| Appendix A. ChangeLog . . . . .                      | 9 |
| Authors' Addresses . . . . .                         | 9 |

## 1. 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].

## 2. Problem Statement

Sockets listening on a global addresses are exposed to attacks. RFC6724 Rule 8 selects a candidate address with the smallest scope. Applications don't always have LL candidate address. They usually have a GUA address. If GUA is on a local link, an application will open a socket using GUA. To avoid using GUA on the local link, a sender needs to find a destination LL address. Currently SASA algorithm (RFC 6724 "Default Address Selection for Internet Protocol Version 6 (IPv6)") cannot use the smallest common scope, given destination GUA.

For security reasons, hosts should use an address with the smallest scope. To avoid these attacks, the host should use LL or ULA addresses.

These security reasons, in more detail, are described next. There is a security problem when a Host uses (one of) its Global Unicast Address(es) (GUA) to communicate to another Host situated on the same link. The problem appears even if that second Host uses its link-local address (LL) for this communication.

The problem is that the Host that uses the GUA to actively communicate with another Host situated on the same link opens a globally reachable entry point in its operating system kernel. This entry point appears when the GUA is assigned to a socket structure. Were that address an LL, and not a GUA, that entry would not be globally reachable.

To realize communications between Hosts on the same link, it is sufficient to rather use LL addresses on both Hosts.

When a Host uses a GUA to communicate to another Host situated on the same link, it unnecessarily becomes an easy attack target. The attacker might be situated anywhere in the Internet (globally).

### 3. Least Common Scope Communications

It is recommended that a Host that needs to communicate with another Host that is situated in a particular scope, to use addresses of same scope, or of the least common scope.

For example, two Hosts situated on the same link should ideally use LL addresses to communicate to each other. An interpretation suggests that, given GUA and ULA, a least common 'scope' is the ULA scope (even though, formally, both ULA and GUA are of same global scope). But the global unicast addresses (GUAs) should not be used for two Hosts on the same link: the global scope is unnecessarily large; it unnecessarily opens doors to attacks.

### 4. LL Address Resolution

The operation of resolving an LL address (LL address resolution) is to find the link-local address that is assigned to the same interface as a GUA (or an ULA). This operation can be realized in several manners.

In one manner, the pair [GUA or ULA address; LL address] is stored in a distributed file such as the Active Directory or the DNS. The resolution operation is to query that file to find the LL address that corresponds to a GUA or ULA address. There are some issues to be considered. For example, typically the LL address is not assigned neither by DHCPv6 nor by RA (it is self formed by a Host when the interface is put up by using a universally known prefix "fe80::/10") then how would DNS get that LL address? Another example is: how to query DNS to request the LL address corresponding to an AAAA entry? (it is known how to query DNS to obtain the AAAA of an FQDN, but not the LL of an AAAA).

In another manner, the operation of resolving a link-local address (LL address resolution) is performed within the context of selecting source and destination addresses within a Host. In that context, the following steps occur:

1. Given multiple destination addresses, the DASA selects GUA and ULA destination. The term 'DASA' designates the Destination Address Selection Algorithm.
2. The LL address resolution operation is performed for these GUA and ULA.
3. The GUA and the LL addresses are given as input to the SASA. The term 'SASA' stands for Source Address Selection Algorithm. The SASA selects LL.

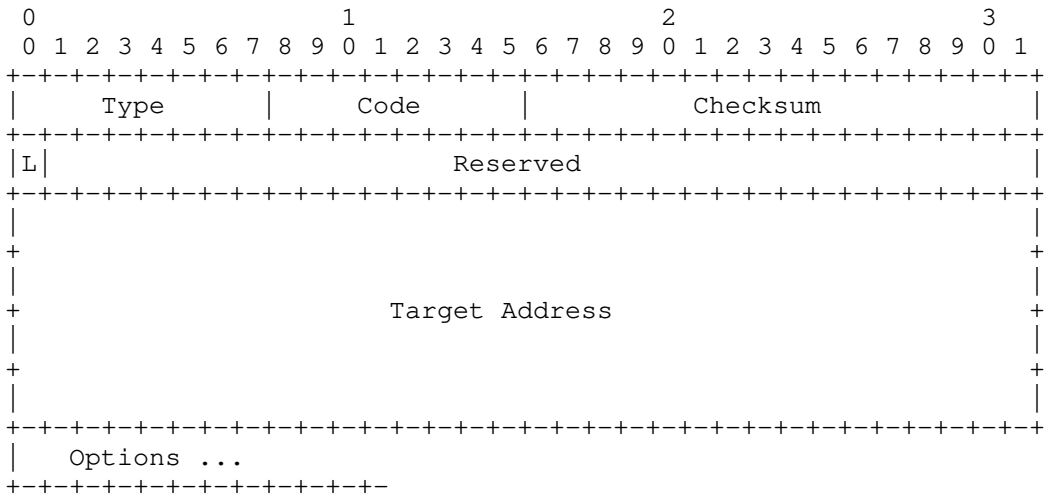
To facilitate LL communication on the local link, given a destination GUA or ULA:

- o Prior to SASA, a host needs to check if a destination is ON-LINK
- o for ON-LINK destination, a host needs to resolve the GUA or ULA destination address into a destination host LL address,
- o a socket needs to open a port for the source LL address, and
- o send packets to the destination LL address.

If both GUA and ULA destinations are known, and ULA destination is not on the link, SASA SHOULD use ULA address.

For the purposes of this document, Link Local (LL) address resolution is the process through which a host determines the Link Local address of a neighbor which is on the local subnet, given only neighbor's GUA or ULA IPv6 address (this 'address resolution' term is different than typical 'ND' term, or than the RFC4861 'address resolution' term which resolves an IP address into a MAC address). LL address resolution is performed only on addresses that are determined to be on-link and for which the sender does not know the corresponding Link Local address. Once the target LL address is learned, the communication sockets use LL addresses and are not exposed to security attacks.

For LL address resolution, 'L' flag is added to NS message. The Target-Address, TA, field in the NS message contains the address of the target of the solicitation (e.g., a host GUA or ULA address). The 'L' flag is added to Neighbor Solicitation Message, for LL address request



IP Fields:

Source Address  
If L bit is set, either LL address assigned to the interface from which this message is sent or (if Duplicate Address Detection is in progress [ADDRCONF rfc4861]) the unspecified address.

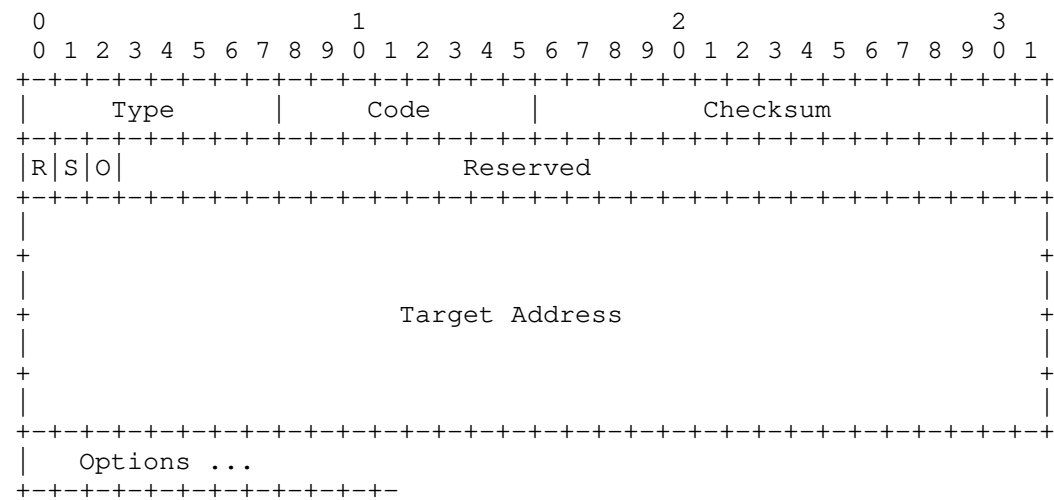
Destination Address  
Either the solicited-node multicast address corresponding to the target GUA or ULA address, or the target GUA or ULA address.

ICMP Fields:

L Link Local flag. When set, the L-bit indicates that the sender is requesting Link Local address from the target.

Figure 1: NS with 'L' bit

After receiving the Neighbor Solicitation message, the target returns its Link Local address in the Target Link-Local Address Option in a unicast Neighbor Advertisement, NA, message.



IP Fields:

Source Address  
If NS L bit is set, LL address of the same GUA target interface is provided

Possible options:

Target Link-Local address  
The Link Local address of the same GUA target, the sender of NA.  
This option MUST be included if NS L bit is set and LL is available.

Type                    4 (Target Link Local address)  
Length                   16 bytes  
Link Local Address: e.g. fe80:0:0:0:aa:bb:cc:dd

Receivers MUST silently ignore this option if they do not recognize it and continue processing the message.

Figure 2: NA for LL address resolution

The request for comments number 5014 [RFC5014], which treats about socket APIs, needs to be updated to use the given destination GUA or ULA addresses for ON-LINK determination, prior to SASA address selection; it also needs to be updated to specify to send packets using LL address while talking to ON-LINK destinations.

## 5. Sending algorithm with LL Address

A sender application can choose to use LL for on-link communication. That request can be passed via a socket API to ND. ND should set NS 'L' bit to indicate the LL address resolution is required and use of LL for the on-link communication, if a destination host returns it.

If a destination host is listening on GUA only for a particular application, and this algorithm is supported, the host should disable LL address resolution by not returning LL address in NA. By default, the LL address resolution should be disabled. Otherwise, a sender would send a packet to destination LL address and there is no socket listening on that address. LL address resolution should be enabled when all socket APIs are ready to support LL sockets (open one socket for GUA and one for LL and after LL address resolution, NA with LL is returned, close GUA socket) or all sockets are bound to ANY address.

The process starts with an application requesting a socket to send a packet to GUA destination. First step is a destination address selection and the sequence goes to the LL address resolution, step 4:

1st: A sending application should have an option to request LL vs. GUA communication, when opening a socket to GUA destination, that might be on a local link. Socket API should have this option and use it to initiate LL address resolution.

2nd: Host should choose destination address, if multiple GUA and ULA are provided

3rd: Host should choose a source address, for the selected destination address

4th: Host should choose a next hop, and outgoing interface, based on the source address prefix

5th: If a destination is on-link, the host should resolve destination GUA into destination LL. Step 5 is further broken down into:

5.1st: Sender creates a neighbour cache entry for GUA.

5.2nd: Sender sends NS, with L bit set, to GUA.

5.3rd: Sender receives NA with link-layer and LL addresses

5.4th: Sender updates GUA cache entry with the link-layer address

5.5th: Sender creates a neighbour cache entry for destination LL address and sets the destination link-layer address of the destination host

6th: Sender transmits a packet to link-layer address of the destination host, using destination host LL address as IPv6 packet destination address

7th: Application sending to GUA should obtain the SASA address (which is now LL address) for the further negotiations (e.g. SIP needs to negotiate media addresses by sending re-INVITE).

8th: Sender closes the socket listening on GUA and opens a socket listening on LL.

## 6. Other Issues with LL Address Resolution

If the Host 'switches' the destination address of an ongoing flow, between the GUA and the LL, there might be interruptions in communications. The 'switching' behaviour depends on the application. Some applications (e.g. a particular application using the SIP protocol) the destination address is selected prior to opening the socket dedicated to streaming the media data. In such an application, a hard outage (e.g. interface down), might involve the creation of a new socket, and thus interruptions in media streaming. The question of maintaining an ongoing communication upon 'switching' between a GUA and an LL destination address is valid, for certain applications.

Multiple DNS aspects, for the resolution operation. Which LL address corresponds to a GUA?. How would DNS get that LL address?

## 7. Security Considerations

Security

## 8. IANA Considerations

IANA

## 9. Contributors

Contributors.



## 10. Acknowledgements

Mark Smith, Eduard Vasilenko.

## 11. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5014] Nordmark, E., Chakrabarti, S., and J. Laganier, "IPv6 Socket API for Source Address Selection", RFC 5014, DOI 10.17487/RFC5014, September 2007, <<https://www.rfc-editor.org/info/rfc5014>>.
- [RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <<https://www.rfc-editor.org/info/rfc6724>>.

## Appendix A. ChangeLog

The changes are listed in reverse chronological order, most recent changes appearing at the top of the list.

-00: initial version, with Dusan's comments.

## Authors' Addresses

Dusan Mudric  
Ciena

,

Canada

Phone:  
+1-613-670-2425

Email:  
[dmudric@ciena.com](mailto:dmudric@ciena.com)

Alexandre Petrescu  
CEA, LIST

CEA Saclay

Gif-sur-Yvette

,

Ile-de-France

91190

France

Phone:

+33169089223

Email:

Alexandre.Petrescu@cea.fr

SPRING  
Internet-Draft  
Intended status: Informational  
Expires: January 10, 2022

W. Cheng  
China Mobile  
C. Xie  
China Telecom  
R. Bonica  
Juniper  
D. Dukes  
Cisco Systems  
C. Li  
Huawei  
P. Shaofu  
ZTE  
W. Henderickx  
Nokia  
July 09, 2021

Compressed SRv6 SID List Requirements  
draft-srcompdt-spring-compression-requirement-07

Abstract

This document specifies requirements for solutions to compress SRv6 SID lists.

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 <https://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 January 10, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .  | 3  |
| 2. Conventions used in this document . . . . .                   | 4  |
| 2.1. Requirements Language . . . . .                             | 4  |
| 2.2. Terminology . . . . .                                       | 4  |
| 3. SRv6 SID List Compression Requirements . . . . .              | 4  |
| 3.1. Dataplane Efficiency and Performance Requirements . . . . . | 4  |
| 3.1.1. Encapsulation Header Size . . . . .                       | 5  |
| 3.1.2. Forwarding Efficiency . . . . .                           | 5  |
| 3.1.3. State Efficiency . . . . .                                | 6  |
| 4. SRv6 Specific Requirements . . . . .                          | 6  |
| 4.1. SRv6 Based . . . . .  | 6  |
| 4.2. Functional Requirements . . . . .                           | 7  |
| 4.2.1. SRv6 Functionality . . . . .                              | 7  |
| 4.2.2. Heterogeneous SID lists . . . . .                         | 8  |
| 4.2.3. SID list length . . . . .                                 | 8  |
| 4.2.4. SID summarization . . . . .                               | 8  |
| 4.3. Operational Requirements . . . . .                          | 9  |
| 4.3.1. Lossless Compression . . . . .                            | 9  |
| 4.3.2. Preservation of non-routing information . . . . .         | 9  |
| 4.3.3. Address Planning . . . . .                                | 10 |
| 4.4. Scalability Requirements . . . . .                          | 10 |
| 4.4.1. Adjacency segment scale . . . . .                         | 10 |
| 4.4.2. Prefix segment scale . . . . .                            | 11 |
| 4.4.3. Service Scale . . . . .                                   | 11 |
| 4.4.4. Compression Levels . . . . .                              | 11 |
| 5. Protocol Design Requirements . . . . .                        | 11 |
| 5.1. SRv6 Base Coexistence . . . . .                             | 11 |
| 6. Security Requirements . . . . .                               | 12 |
| 6.1. Security Mechanisms . . . . .                               | 12 |
| 6.2. SR Domain Protection . . . . .                              | 12 |
| 7. IANA Considerations . . . . .                                 | 12 |
| 8. Security Considerations . . . . .                             | 13 |
| 9. Contributors . . . . .  | 13 |
| 10. Normative References . . . . .                               | 13 |
| Appendix A. Proposed Requirements . . . . .                      | 14 |
| A.1. IPv6 Based . . . . .  | 14 |

|                                    |    |
|------------------------------------|----|
| A.2. Point to Multipoint . . . . . | 15 |
| A.3. Parsability . . . . .         | 15 |
| Authors' Addresses . . . . .       | 15 |

## 1. Introduction

The SPRING working group defined SRv6, with [RFC8402] describing how the Segment Routing (SR) architecture is instantiated on two data-planes: SR over MPLS (SR-MPLS) and SR over IPv6 (SRv6). SRv6 uses a routing header called the SR Header (SRH) [RFC8754] and defines SRv6 SID behaviors and a registry for identifying them in [RFC8986]. SRv6 is a proposed standard and is deployed today.

The SPRING working group has observed that some use cases, such as strict path TE, may require long SRv6 SID lists. There are several proposed methods to reduce the resulting SRv6 encapsulation size by compressing the SID list.

The SPRING working group formed a design team to define requirements for, and analyze proposals to, compress SRv6 SID lists.

It is a goal of the design team to identify solutions to SRv6 SID list compression that are based on the SRv6 standards. As such, this document provides requirements for SRv6 SID list compression solutions that utilize the existing SRv6 data plane and control plane.

It is also a goal of the design team to consider proposals that are not based on the SRv6 data plane and control plane. As such, this document includes requirements to evaluate whether a compression proposal provides all the functionality of SRv6 (section "SRv6 Functionality") in addition to satisfying compression specific requirements.

For each requirement, a description, rationale and metrics are described.

The design team will produce a separate document to analyze the proposals.

This document is a draft; additional requirements are under review, additional requirements will be added, and current requirements may change. Appendix A contains a subset of requirements without unanimous consensus. Additional requirements without unanimous consensus are not in the appendix.

## 2. Conventions used in this document

### 2.1. 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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

### 2.2. Terminology

SR: Segment Routing

SRH: Segment Routing Header

MPLS: Multiprotocol Label Switching

SR-MPLS: Segment Routing over MPLS data plane

SID: Segment Identifier

SRv6: Segment Routing over IPv6

SRv6 SID List: A list of SRv6 SIDs

Compression proposal: A proposal to compress SRv6 SID lists

SRv6 base: SRv6 as defined in [RFC8402], [RFC8754], [RFC8986]

SID numbering space: may be implemented as

- o a single IGP instance
- o a single IGP level or area
- o two or more autonomous systems that coordinate SID numbering space
- o two or more IGP instances that coordinate SID numbering space

SRv6 Encapsulation Header: The IPv6 header, and any extension headers preceding a payload, used to implement a SRv6 base or compression proposal.

## 3. SRv6 SID List Compression Requirements

### 3.1. Dataplane Efficiency and Performance Requirements

### 3.1.1. Encapsulation Header Size

Description: The compression proposal MUST reduce the size of the SRv6 encapsulation header.

Rationale: A smaller SRv6 encapsulation results in better MTU efficiency.

Metric: Compression is the ratio of the IPv6 encapsulation size of SRv6 as defined in [RFC8402], [RFC8754], [RFC8986] vs the IPv6 encapsulation size of a given proposal. The encapsulation savings of a compression proposal vs the SRv6 base is a useful measurement to compare proposals.

The encapsulation metric (E) records the number of bytes required for a proposal to encapsulate a packet given a specific segment list.

o  $E(\text{proposal}, \text{segment list})$ .

The encapsulation savings (ES) records the encapsulation savings for a proposal to encapsulate a packet given a specific segment list.

o  $ES(\text{proposal}, \text{segment list}) = 1 - E(\text{proposal}, \text{segment list})/E(\text{SRv6 base}, \text{segment list})$ .

### 3.1.2. Forwarding Efficiency

Description: The compression proposal SHOULD minimize the number of required hardware resources accessed to process a segment.

Rationale: Efficiency in bits on the wire and processing efficiency are both important. Optimizing one at the expense of the other may lead to significant performance impact.

Metric: The data plane efficiency metric (D) records the data plane forwarding efficiency of the proposed solution. Two metrics are used and recorded at each segment endpoint:

- o  $D.PRS(\text{segment list})$ : number of headers parsed during processing of the segment list, starting from and including the IPv6 header.
- o  $D.LKU(\text{segment list})$ : number of FIB lookups during processing of the segment list. The type of lookup is also recorded as longest prefix match (LPM) or exact match (EM)

### 3.1.3. State Efficiency

Description: The compression proposal SHOULD minimize the amount of additional forwarding state stored at a node.

Rationale: Additional state increases the complexity of the control plane and data plane. It can also result in an increase in memory usage.

Metric: The state efficiency metric (S) records the amount of additional forwarding state required by the proposed solution.

- o S(node parameters): the number of additional forwarding states that need to be stored at a node, given a set of node parameters consisting of the number of nodes in the network, number of local interfaces, number of adjacencies. The forwarding state is counted as entries required in a Forwarding Information Base (FIB) at a node.

## 4. SRv6 Specific Requirements

### 4.1. SRv6 Based

Description: A solution to compress SRv6 SID Lists SHOULD be based on the SRv6 architecture, control plane and data plane. The compression solution MAY be based on a different data plane and control plane, provided that it derives sufficient benefit.

Rationale: A compression proposal built on existing IETF standards is preferable to creating new standards with equivalent functionality and performance.

Metric: The utilization metric (U) records whether a proposal utilizes the SRv6 specifications.

Utilization is recorded in a table, with a column per proposal and rows consisting of the following metrics:

- o U.RFC8402: utilizes [RFC8402].
- o U.RFC8754: utilizes [RFC8754].
- o U.PGM: utilizes [RFC8986].
- o U.IGP: utilizes [I-D.ietf-lsr-isis-srv6-extensions].
- o U.BGP: utilizes [I-D.ietf-bess-srv6-services].
- o U.POL: utilizes [I-D.ietf-spring-segment-routing-policy].
- o U.BLS: utilizes [I-D.ietf-idr-bgppls-srv6-ext].
- o U.SVC: utilizes [I-D.ietf-spring-sr-service-programming].
- o U.OAM: utilizes [I-D.ietf-6man-spring-srv6-oam].
- o U.ALG: utilizes [I-D.ietf-lsr-flex-algo].



Each cell contains "yes" for utilizes, or "no" for does not utilize.

## 4.2. Functional Requirements

### 4.2.1. SRv6 Functionality

Description: A solution to compress an SRv6 SID list MUST support the functionality of SRv6. This requirement ensures no SRv6 functionality is lost. It is particularly important to understand how a proposal, as evaluated in section "SRv6 Based", provides this functionality.

Rationale: Operators require SRv6 functionality. Evaluating the extent to which a proposal supports SRv6 functionality is important for operators and implementors to understand the impact on network operations.

Metric: The Functionality metric (F) records whether a proposal supports SRv6 functionality and how the functionality is provided.

Functionality is recorded in a table with columns for each proposal and rows consisting of the following metrics:

- o F.SID: Supports SRv6 SID functionality as described in [RFC8402]
- o F.SCOPE: Supports globally and locally scoped SID functionality as described in [RFC8402]
- o F.PFX: Supports prefix SID functionality as described in [RFC8402] and [RFC8986]
- o F.ADJ: Supports adjacency SID functionality as described in [RFC8402] and [RFC8986]
- o F.BIND: Supports binding SID functionality as described in [RFC8402] and [RFC8986]
- o F.PEER: Supports BGP peering SID functionality as described in [RFC8402] and [RFC8986]
- o F.SVC: Supports L3 and L2 VPN service SID functionality as described in [RFC8986]
- o F.ALG: Supports flexible algorithms functionality as described in [I-D.ietf-lsr-flex-algo]
- o F.TILFA: Supports TI-LFA functionality as described in [I-D.ietf-rtgwg-segment-routing-ti-lfa]
- o F.SEC: Supports securing an SR domain with ingress filtering as functionally defined in [RFC8754]
- o F.IGP: Supports distributing topological SIDs and behaviors via ISIS as functionally described in [I-D.ietf-lsr-isis-srv6-extensions]
- o F.BGP: Supports BGP VPNs as functionally described in [I-D.ietf-bess-srv6-services]

- o F.POL: Supports SR policies and steering traffic over those policies as functionally described in [I-D.ietf-spring-segment-routing-policy]
- o F.BLS: Supports Link State distribution via BGP as functionally described in [I-D.ietf-idr-bgpls-srv6-ext]
- o F.SFC: Supports stateless service programming as functionally described in [I-D.ietf-spring-sr-service-programming]
- o F.PING: Supports pinging a SID to verify the SID is implemented as functionally described in [I-D.ietf-6man-spring-srv6-oam]

Each cell contains the specification name documenting the functionality.

#### 4.2.2. Heterogeneous SID lists

Description: The compression proposal SHOULD support a combination of compressed and non-compressed segments in a single path. As an example, a solution may satisfy this requirement without being SRv6 based by using a binding SID to impose an additional SRv6 header (IPv6 header plus optional SRH) with non-compressed SID.

Rationale: Support of SID lists with compressed and non-compressed SIDs reduces encapsulation size when not all SRv6 nodes deploy the compression proposal or 128-bit SIDs are required.

Metric: A compliant compression proposal supports both:

- o classic 128-bit SRv6 SIDs in the IPv6 Destination Address field
- o segment lists (i.e., paths) with both compressed and 128-bit SRv6 SIDs.

#### 4.2.3. SID list length

Description: The compression proposal MUST be able to represent SR paths that contain up to 16 segments.

Rationale: Strict TE paths require SID list lengths proportional to the diameter of the SR domain.

Metric: The compression proposal must be able to steer a packet through an SR path that contains up to sixteen segments.

#### 4.2.4. SID summarization

Description: The solution MUST be compatible with segment summarization.

**Rationale:** Summarization of segments is a key benefit of SRv6 vs SR MPLS. In interdomain deployments, any node can reach any other node via a single prefix segment. Without summarization, border router SIDs must be leaked, and an additional global prefix segment is required for each domain border to be traversed.

**Metric:** A solution supports summarization when segments can be summarized for advertisement into other IGP domains or levels.

#### 4.3. Operational Requirements

##### 4.3.1. Lossless Compression

**Description:** A path traversed using a compressed SID list MUST always be the same as the path traversed using the uncompressed SID list if no compression was applied.

**Rationale:** In SRv6, we can represent a path to meet certain objectives. A compression proposal needs to support the objectives with the same path.

**Metric:** Information present in the pre-compression segment list MUST also be present in the post-compression SID list.

##### 4.3.2. Preservation of non-routing information

**Description:** The compression mechanism MUST NOT cause the loss of non-routing information when delivering a packet from the SR ingress node to the egress/penultimate SR node

**Rationale:** SRv6 ingress nodes encode non-routing information in the IPv6 header chain. This information can be encoded in the following fields:

- o Hop Count
- o DSCP bits
- o ECN bits
- o Flow label
- o HBH Options Extension header
- o Fragment Extension header
- o Authentication Extension header
- o Encrypted Security Payload Extension header
- o Destination Options Extension header

Some of these fields are mutable (e.g., Hop Count) while others are immutable (e.g., Fragment Extension Header).

Some of these fields contain information that is required by every node along a packet's delivery path (e.g., Hop Count). Others contain information that is required only by the packet's ultimate destination (e.g., Fragment Extension Header).

Therefore, the compression mechanism MUST NOT prevent this information from being delivered, in an IPv6 header chain, to any node that needs it.

**Metric:** The SR source node encapsulates its payload (e.g., Ethernet, IP, TCP) in an IPv6 header. The SRv6 header contains both routing and non-routing information. The compression mechanism MUST NOT cause the loss of non-routing information when delivering a packet from the SR ingress node to the egress/penultimate SR node.

#### 4.3.3. Address Planning

**Description:** Network operators require addressing plan flexibility, The compression mechanism MUST support flexible IPv6 address planning, it MUST support deployment by using GUA from different address blocks.

**Rationale:** The address planning of the network may vary based on the addressing scheme of the operator, so the solution MUST support a flexible addressing scheme. Operators need to deploy the solution based on their own address planning.

**Metric:** The compression proposal supports locators drawn from different prefixes with the solutions analysis indicating efficiency.

#### 4.4. Scalability Requirements

##### 4.4.1. Adjacency segment scale

**Description:** The compression proposal MUST be capable of representing 65000 adjacency segments per node

**Rationale:** Typically, network operators deploy networks with tens or hundreds of adjacency segments per node, but some network operators may deploy networks that use more adjacency segments per node.

**Metric:** A proposal that allows 65000 adjacency segments per node satisfies this requirement.

#### 4.4.2. Prefix segment scale

Description: The compression proposal MUST be capable of representing 1 million prefix segments per SID numbering space.

Rationale: Typically, network operators deploy networks with thousands of prefix segments per SID numbering space, but some network operators may deploy networks that use more prefix segments per SID numbering space.

Metric: A proposal that allows 1 million prefix segments per SID numbering space satisfies this requirement.

#### 4.4.3. Service Scale

Description: The compression proposal MUST be capable of representing 1 million services per node.

Rationale: Typically, network operators deploy networks with tens to hundreds of thousands of services per node, but some network operators may deploy networks that use more services per node.

Metric: A proposal that allows 1 million services per node satisfies this requirement.

#### 4.4.4. Compression Levels

Description: The compression proposal SHOULD be able to support multiple levels of compression.

Rationale: The compression proposal will be deployed in networks of varying size with SID numbering spaces of varying size. Network and service scale can directly impact SID length and the ability of a proposal to compress the SID list.

Metric: A compression proposal that supports relatively better compression for smaller SID numbering spaces and service scale satisfies this requirement.

### 5. Protocol Design Requirements

#### 5.1. SRv6 Base Coexistence

Description: The compression proposal MUST support simultaneous deployment with SRv6 networks.

Rationale: SRv6 is deployed today. A compression proposal that interoperates well with SRv6, as deployed, will reduce the overhead

and simplify operations. For Network operators who would migrate to compressed SRv6 SID lists, the migration is expected to gradually occur over a period of time as they upgrade networks, domains, device families and software instances.

Metric: A compliant compression proposal provides the following

- o Supports simultaneous deployment at a node with current SRv6 SIDs.
- o Supports simultaneous deployment at a node with current SRv6 control plane.
- o Supports simultaneous operation of current SRv6 paths with compressed paths.
- o Supports the behaviors in [RFC8986].
- o Does not require removal of existing IPv6 address planning.

## 6. Security Requirements

### 6.1. Security Mechanisms

Description: The compression solution SHOULD be able to address security issues that it introduces, using existing security mechanisms.

Rationale: It is important to identify security issues and how to address them in any specification.

Metric: A compression solution that does not introduce unresolved security issues meets this requirement.

### 6.2. SR Domain Protection

Description: A compression solution must not require nodes outside the SR domain to know SID values within the SR domain, and it must provide the ability to block nodes outside an SR domain from accessing SIDS.

Rationale: The unauthorized use of SIDs within the SR domain by nodes outside the domain can disrupt an operators' network.

Metric: A compliant solution describes how access to SIDs within the SR domain is denied to nodes outside the SR domain.

## 7. IANA Considerations

This document has no requests to IANA.

## 8. Security Considerations

TBD

## 9. Contributors

The following individuals contributed to this draft

Sanders Steffann, SJM Steffann Consultancy, sander@steffann.nl

## 10. Normative References

[I-D.ietf-6man-spring-srv6-oam]

Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)", draft-ietf-6man-spring-srv6-oam-10 (work in progress), April 2021.

[I-D.ietf-bess-srv6-services]

Dawra, G., Filsfils, C., Talaulikar, K., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "SRv6 BGP based Overlay Services", draft-ietf-bess-srv6-services-07 (work in progress), April 2021.

[I-D.ietf-idr-bgppls-srv6-ext]

Dawra, G., Filsfils, C., Talaulikar, K., Chen, M., Bernier, D., and B. Decraene, "BGP Link State Extensions for SRv6", draft-ietf-idr-bgppls-srv6-ext-07 (work in progress), March 2021.

[I-D.ietf-lsr-flex-algo]

Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-15 (work in progress), April 2021.

[I-D.ietf-lsr-isis-srv6-extensions]

Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extension to Support Segment Routing over IPv6 Dataplane", draft-ietf-lsr-isis-srv6-extensions-14 (work in progress), April 2021.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]

Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-06 (work in progress), February 2021.

- [I-D.ietf-spring-segment-routing-policy]  
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-11 (work in progress), April 2021.
- [I-D.ietf-spring-sr-service-programming]  
Clad, F., Xu, X., Filsfils, C., Bernier, D., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", draft-ietf-spring-sr-service-programming-04 (work in progress), March 2021.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.

## Appendix A. Proposed Requirements

This appendix contains requirements that the design team discussed but could not be agreed upon.

### A.1. IPv6 Based

Description: The compression mechanism requires every node along the packet's delivery path to be IPv6-capable. It MUST not require any



node along the packet's forwarding path to support any other forwarding plane (e.g., IPv4, MPLS)

Rational: According to RFC 8402, SRv6 is an instantiation of the SR Architecture over the IPv6 data plane.

Metric: A compliant solution requires every node along the packet's delivery path to be IPv6-capable. It does not require any node along the packet's forwarding path to support any other forwarding plane (e.g., IPv4, MPLS)

#### A.2. Point to Multipoint

Description: The compression mechanism SHOULD support point-to-multipoint SR paths.

Rationale: Many VPN services require point-to-multipoint SR paths.

Metric: A compliant proposal can encode a multicast address in the ultimate segment of the segment list.

#### A.3. Parsability

Description: The compression mechanism MUST be parsable. That is, the node that consumes the compressed SID list must be able to decode the active and next segment. Parsing information MAY be conveyed in either the forwarding or control plane.

Rationale: Failure to parse the compressed SID list leads to undesired behaviors.

Metric: In the nominal case the producer and consumer of the SID list agree on the active segment and next segment. In foreseeable failure modes it is possible to determine why the producer and consumer don't agree.

#### Authors' Addresses

Weiqiang Cheng  
China Mobile

Email: chengweiqiang@chinamobile.com

Chongfeng Xie  
China Telecom

Email: xiechf@chinatelecom.cn

Ron Bonica  
Juniper

Email: [rbonica@juniper.net](mailto:rbonica@juniper.net)

Darren Dukes  
Cisco Systems

Email: [ddukes@cisco.com](mailto:ddukes@cisco.com)

Cheng Li  
Huawei

Email: [c.l@huawei.com](mailto:c.l@huawei.com)

Peng Shaofu  
ZTE

Email: [peng.shaofu@zte.com.cn](mailto:peng.shaofu@zte.com.cn)

Wim Henderickx  
Nokia

Email: [wim.henderickx@nokia.com](mailto:wim.henderickx@nokia.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: August 16, 2021

F. Templin, Ed.  
The Boeing Company  
A. Whyman  
MWA Ltd c/o Inmarsat Global Ltd  
February 12, 2021

IPv6 Neighbor Discovery Overlay Multilink Network Interface (OMNI)  
Option  
draft-templin-6man-omni-option-09

## Abstract

This document defines a new IPv6 Neighbor Discovery (ND) option termed the "Overlay Multilink Network Interface (OMNI) Option". The OMNI option may appear in any IPv6 ND message type; it is processed by interface types that recognize the option and ignored by all other interface types. The option supports functions such as prefix registration and multilink coordination, and is extensible to support additional functions in the future.

## 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 <https://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 16, 2021.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .  | 2  |
| 2. Terminology . . . . .   | 2  |
| 3. The Overlay Multilink Network Interface (OMNI) IPv6 ND Option | 3  |
| 3.1. Sub-Options . . . . .                                       | 4  |
| 3.1.1. Pad1 . . . . .  | 5  |
| 3.1.2. PadN . . . . .  | 6  |
| 3.1.3. Interface Attributes (Type 1) . . . . .                   | 6  |
| 3.1.4. Sub-Type Extension . . . . .                              | 8  |
| 4. IANA Considerations . . . . .                                 | 9  |
| 5. Security Considerations . . . . .                             | 9  |
| 6. Acknowledgements . . . . .                                    | 9  |
| 7. References . . . . .  | 9  |
| 7.1. Normative References . . . . .                              | 9  |
| 7.2. Informative References . . . . .                            | 10 |
| Authors' Addresses . . . . .                                     | 11 |

## 1. Introduction

This document defines a new IPv6 Neighbor Discovery (ND) option termed the "Overlay Multilink Network Interface (OMNI) Option". The OMNI option may appear in any IPv6 ND message type; it is processed by interface types that recognize the option and ignored by all other interface types. The option supports functions such as prefix registration and multilink coordination for interface types such as the OMNI interface [I-D.templin-6man-omni-interface], and is extensible to support additional functions in the future.

The following sections discuss the OMNI option format and contents. Use cases appear in IPv6 over specific link layer documents such as [I-D.templin-6man-omni-interface], where the International Civil Aviation Organization (ICAO) has expressed interest in the option in support of their Document 9896 [ATN][ATN-IPS]. An IPv6 ND option Type number assignment is requested in the IANA Considerations section.

## 2. Terminology

The terminology in the normative references applies. The term "underlying interface" refers to one of potentially multiple Layer-2 interfaces over which a Layer-3 (virtual) interface is configured.

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 BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

### 3. The Overlay Multilink Network Interface (OMNI) IPv6 ND Option

An Overlay Multilink Network Interface (OMNI) IPv6 ND option is defined. The option (known as the "OMNI option") is formatted as shown in Figure 1:

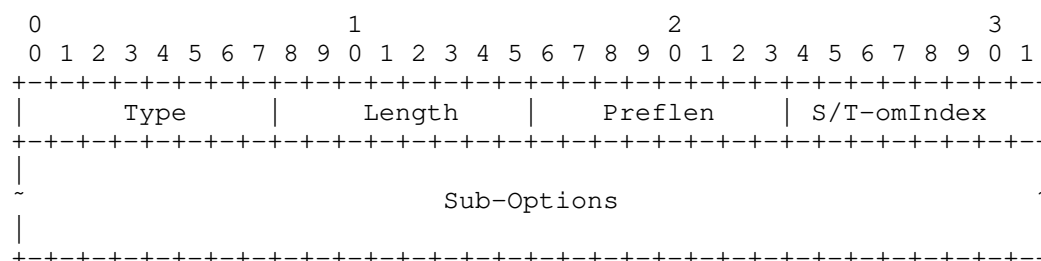


Figure 1: OMNI Option Format

In this format:

- o Type is set to TBD1.
- o Length is set to the number of 8 octet blocks in the option.
- o Preflen is an 8 bit field that determines the length of prefix associated with an IPv6 address of the IPv6 ND message. Values 0 through 128 specify a valid prefix length (all other values are invalid). For IPv6 ND messages sent from the source to a target node, Preflen applies to the IPv6 source address and provides the length that the source is requesting or asserting. For IPv6 ND messages replies from the target to the original source, Preflen applies to the IPv6 destination address and indicates the length that the target is granting.
- o S/T-omIndex is a 1-octet field that encodes a value between 0 and 255 identifying the source or target underlying interface for the IPv6 ND message. For RS and NS messages S/T-omIndex refers to the "Source" underlying interface over which the message is sent, while for RA and NA messages S/T-omIndex refers to the "Target" underlying interface that will receive the message.

- o Sub-Options is a Variable-length field, of length such that the complete OMNI Option is an integer multiple of 8 octets long. Contains one or more Sub-Options, as described in Section 3.1.

The OMNI option may appear in any IPv6 ND message type; it is processed by interfaces that recognize the option and ignored by all other interfaces. If multiple OMNI option instances appear in the same IPv6 ND message, the interface processes the Preflen and S/T-omIndex fields in the first instance and ignores those fields in all other instances. The interface processes the Sub-Options of all OMNI option instances in the same IPv6 ND message in the consecutive order in which they occur.

The OMNI option(s) in each IPv6 ND message may include full or partial information for the neighbor. The union of the information in the most recently received OMNI options is therefore retained, and the information is aged/removed in conjunction with the corresponding neighbor cache entry.

### 3.1. Sub-Options

The OMNI option includes zero or more Sub-Options. Each consecutive Sub-Option is concatenated immediately after its predecessor. All Sub-Options except Pad1 (see below) are in type-length-value (TLV) encoded in the following format:

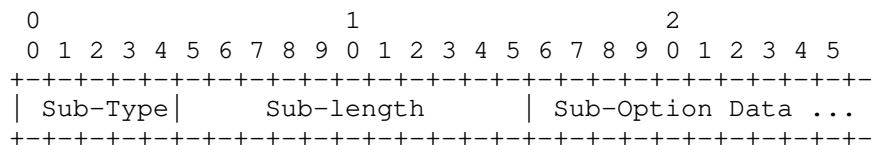


Figure 2: Sub-Option Format

- o Sub-Type is a 5-bit field that encodes the Sub-Option type. Sub-Options defined in this document are:

| Option Name                   | Sub-Type |
|-------------------------------|----------|
| Pad1                          | 0        |
| PadN                          | 1        |
| Interface Attributes (Type 1) | 2        |
| Sub-Type Extension            | 30       |

Figure 3

Sub-Types 3-29 are available for future assignment for major protocol functions. Sub-Type 31 is reserved by IANA.

- o Sub-Length is an 11-bit field that encodes the length of the Sub-Option Data (i.e., ranging from 0 to 2034 octets).
- o Sub-Option Data is a block of data with format determined by Sub-Type and length determined by Sub-Length.

During transmission, the OMNI interface codes Sub-Type and Sub-Length together in network byte order in 2 consecutive octets, where Sub-Option Data may be up to 2034 octets in length. This allows ample space for coding large objects (e.g., ascii character strings, protocol messages, security codes, etc.), while a single OMNI option is limited to 2040 octets the same as for any IPv6 ND option. If the Sub-Options to be coded would cause an OMNI option to exceed 2040 octets, the OMNI interface codes any remaining Sub-Options in additional OMNI option instances in the intended order of processing in the same IPv6 ND message. Implementations must therefore observe size limitations, and must refrain from sending IPv6 ND messages larger than the OMNI interface MTU.

During reception, the OMNI interface processes each OMNI option Sub-Option while skipping over and ignoring any unrecognized Sub-Options. The OMNI interface processes the Sub-Options of all OMNI option instances in the consecutive order in which they appear in the IPv6 ND message, beginning with the first instance and continuing through any additional instances to the end of the message. If a Sub-Option length would cause the running total for that OMNI option to exceed the length coded in the option header, the interface accepts any Sub-Options already processed and ignores the remainder of that OMNI option. The interface then processes any remaining OMNI options in the same fashion to the end of the IPv6 ND message.

Note: large objects that exceed the Sub-Option Data limit of 2034 octets are not supported under the current specification; if this proves to be limiting in practice, future specifications may define support for fragmenting large objects across multiple OMNI options within the same IPv6 ND message.

The following Sub-Option types and formats are defined in this document (note that other documents that are active at the time of this writing will define additional Sub-Option types in the near future):

#### 3.1.1. Pad1

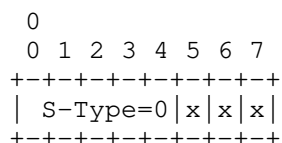


Figure 4: Pad1

- o Sub-Type is set to 0. If multiple instances appear in OMNI options of the same message all are processed.
- o Sub-Type is followed by three 'x' bits, set randomly on transmission and ignored on receipt. Pad1 therefore consists of a whole single octet with the most significant 5 bits set to 0, and with no Sub-Length or Sub-Option Data fields following.

### 3.1.2. PadN

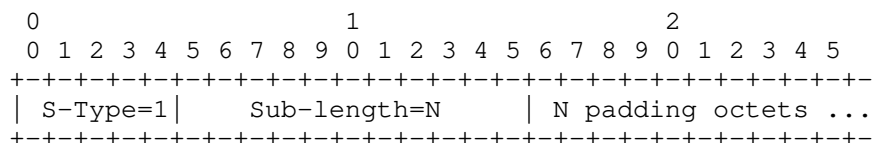


Figure 5: PadN

- o Sub-Type is set to 1. If multiple instances appear in OMNI options of the same message all are processed.
- o Sub-Length is set to N (from 0 to 2047) being the number of padding octets that follow.
- o Sub-Option Data consists of N padding octets that are typically zero-valued (any non-zero values that may appear in the padding octets are not to be interpreted in any way other than as simple padding).

### 3.1.3. Interface Attributes (Type 1)



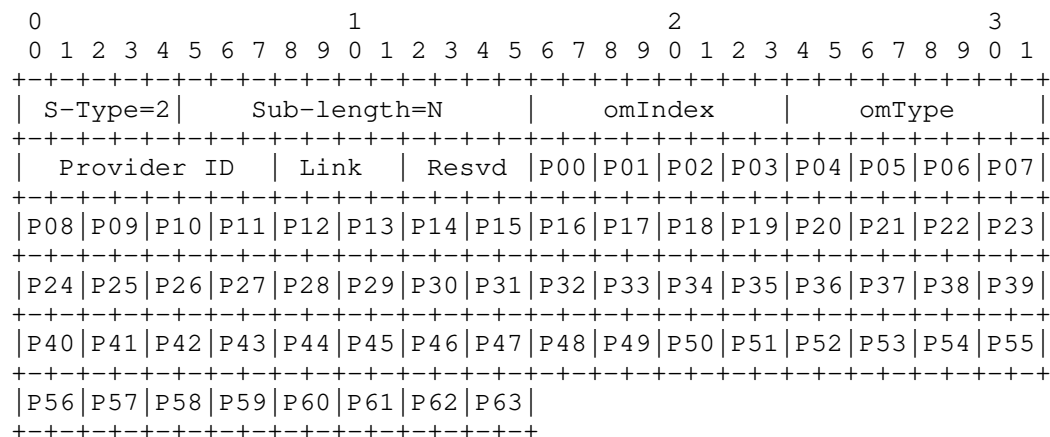


Figure 6: Interface Attributes (Type 1)

- o Sub-Type is set to 2. If multiple instances with different omIndex values appear in OMNI options of the same message all are processed; if multiple instances with the same omIndex value appear, the first is processed and all others are ignored.
- o Sub-Length is set to N (from 1 to 2047) that encodes the number of Sub-Option Data octets that follow.
- o omIndex is a 1-octet field containing a value from 0 to 255 identifying the underlying interface for which the interface attributes apply.
- o omType is a 1-octet field containing a value from 0 to 255 corresponding to the underlying interface identified by omIndex.
- o Provider ID is a 1-octet field containing a value from 0 to 255 corresponding to the underlying interface identified by omIndex.
- o Link encodes a 4-bit link metric. The value '0' means the link is DOWN, and the remaining values mean the link is UP with metric ranging from '1' ("lowest") to '15' ("highest").
- o Resvd is reserved for future use.
- o A 16-octet "Preferences" field immediately follows 'Resvd', with values P[00] through P[63] corresponding to the 64 Differentiated Service Code Point (DSCP) values [RFC2474]. Each 2-bit P[\*] field is set to the value '0' ("disabled"), '1' ("low"), '2' ("medium") or '3' ("high") to indicate a QoS preference for underlying interface selection purposes.

### 3.1.4. Sub-Type Extension

Since the Sub-Type field is only 5 bits in length, future specifications of major protocol functions may exhaust the remaining Sub-Type values available for assignment. This document therefore defines Sub-Type 30 as an "extension", meaning that the actual sub-option type is determined by examining a 1 octet "Extension-Type" field immediately following the Sub-Length field. The Sub-Type Extension is formatted as shown in Figure 7:

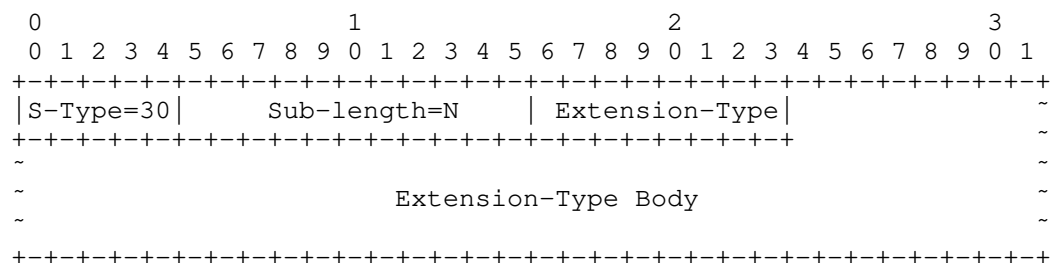


Figure 7: Sub-Type Extension

- o Sub-Type is set to 30. If multiple instances appear in OMNI options of the same message all are processed, where each individual extension defines its own policy for processing multiple of that type.
- o Sub-Length is set to N (from 1 to 2034) that encodes the number of Sub-Option Data octets that follow. The Extension-Type field is always present; hence, the maximum Extension-Type Body length is 2033 octets.
- o Extension-Type contains a 1 octet Sub-Type Extension value between 0 and 255.
- o Extension-Type Body contains an N-1 octet block with format defined by the given extension specification.

Extension-Type values 0 through 252 are available for assignment by future specifications, which must also define the format of the Extension-Type Body and its processing rules. Extension-Type values 253 and 254 are reserved for experimentation, as recommended in [RFC3692], and value 255 is reserved by IANA.

#### 4. IANA Considerations

The IANA is instructed to allocate a Type number TBD1 from the registry "IPv6 Neighbor Discovery Option Formats" for the OMNI option (see: Section 13 of [RFC4861]) as a provisional registration in accordance with Section 4.13 of [RFC8126].

The OMNI option defines a 5-bit Sub-Type field, for which IANA is instructed to create and maintain a new registry entitled "OMNI option Sub-Type values". Initial values for the OMNI option Sub-Type values registry are given below; future assignments are to be made through Expert Review [RFC8126].

| Value | Sub-Type name                 | Reference |
|-------|-------------------------------|-----------|
| ----- | -----                         | -----     |
| 0     | Pad1                          | [RFCXXXX] |
| 1     | PadN                          | [RFCXXXX] |
| 2     | Interface Attributes (Type 1) | [RFCXXXX] |
| 3-29  | Unassigned                    |           |
| 30    | Sub-Type Extension            | [RFCXXXX] |
| 31    | Reserved                      | [RFCXXXX] |

Figure 8: OMNI Option Sub-Type Values

#### 5. Security Considerations

Security considerations for IPv6 [RFC8200] and IPv6 Neighbor Discovery [RFC4861] apply.

#### 6. Acknowledgements

This document is aligned with the International Civil Aviation Organization (ICAO) Aeronautical Telecommunications Network (ATN) with Internet Protocol Services (ATN/IPS) development program.

This document is aligned with the IETF 6man (IPv6) working group.

#### 7. References

##### 7.1. Normative References

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

- [RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, DOI 10.17487/RFC3692, January 2004, <<https://www.rfc-editor.org/info/rfc3692>>.
- [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, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

## 7.2. Informative References

- [ATN] Maiolla, V., "The OMNI Interface - An IPv6 Air/Ground Interface for Civil Aviation, IETF Liaison Statement #1676, <https://datatracker.ietf.org/liaison/1676/>", March 2020.
- [ATN-IPS] WG-I, ICAO., "ICAO Document 9896 (Manual on the Aeronautical Telecommunication Network (ATN) using Internet Protocol Suite (IPS) Standards and Protocol), Draft Edition 3 (work-in-progress)", December 2020.
- [I-D.templin-6man-omni-interface] Templin, F. and T. Whyman, "Transmission of IP Packets over Overlay Multilink Network (OMNI) Interfaces", draft-templin-6man-omni-interface-69 (work in progress), January 2021.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.

Authors' Addresses

Fred L. Templin (editor)  
The Boeing Company  
P.O. Box 3707  
Seattle, WA 98124  
USA

Email: fltemplin@acm.org

Tony Whyman  
MWA Ltd c/o Inmarsat Global Ltd  
99 City Road  
London EC1Y 1AX  
England

Email: tony.whyman@mccallumwhyman.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 25 April 2022

T. Winters  
QA Cafe  
O. Troan  
cisco  
22 October 2021

The Universal IPv6 Configuration Option  
draft-troan-6man-universal-ra-option-06

## Abstract

One of the original intentions for the IPv6 host configuration, was to configure the network-layer parameters only with IPv6 ND, and use service discovery for other configuration information. Unfortunately that hasn't panned out quite as planned, and we are in a situation where all kinds of configuration options are added to RAs. This document proposes a new universal option for RA in a self-describing data format, with the list of elements maintained in an IANA registry, with greatly relaxed rules for registration.

## 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 <https://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 25 April 2022.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|   |   |
|---|---|
| 1. Introduction . . . . .                             | 2 |
| 2. Conventions . . . . .                              | 3 |
| 3. Introduction . . . . .                             | 3 |
| 4. The Universal IPv6 Configuration option . . . . .  | 3 |
| 5. CBOR encoding . . . . .                            | 5 |
| 6. Implementation Guidance . . . . .                  | 5 |
| 7. Implementation Status . . . . .                    | 5 |
| 8. Security Considerations . . . . .                  | 5 |
| 9. IANA Considerations . . . . .                      | 6 |
| 9.1. Universal configuration option . . . . .         | 6 |
| 9.2. Initial objects in the registry . . . . .        | 6 |
| 9.3. Initial objects in the registry . . . . .        | 6 |
| 9.3.1. CDDL/JSON Mapping Parameters to CBOR . . . . . | 6 |
| 9.3.2. Key Registry . . . . .                         | 7 |
| 10. Normative References . . . . .                    | 8 |
| 11. Informative References . . . . .                  | 9 |
| Appendix A. Acknowledgements . . . . .                | 9 |
| Authors' Addresses . . . . .                          | 9 |

## 1. Introduction

This document proposes a new universal option for the Router Advertisement IPv6 ND message [RFC4861]. Its purpose is to use the RA messages as opaque carriers for configuration information between an agent on a router and a host.

DHCP is suited to give per-client configuration information, while the RA mechanism advertises configuration information to all hosts on the link. There is a long running history of "conflict" between the two. The arguments go; there is less fate-sharing in DHCP, DHCP doesn't deal with multiple sources of information, or make it more difficult to change information independent of the lifetimes, RA cannot be used to configure different information to different clients and so on. And of course some options are only available in RAs and some options are only available in DHCP.

While this proposal does not resolve the DHCP vs RA debate, it proposes a solution to the problem of a very slow process of standardizing new Router Advertisement options, and the IETF spending an inordinate amount of time arguing over new configuration options in Router Advertisements. It is possible in the future to use the new universal option in DHCP, since this would lead to additional conflict resolution an additional document will need to be considered for that.

## 2. Conventions

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].

Additionally, the key words "\*MIGHT\*", "\*COULD\*", "\*MAY WISH TO\*", "\*WOULD PROBABLY\*", "\*SHOULD CONSIDER\*", and "\*MUST (BUT WE KNOW YOU WON'T)\*" in this document are to be interpreted as described in RFC 6919 [RFC6919].

## 3. Introduction

This document specifies a new "self-describing" universal configuration option. Currently new configuration option requires "standards action". The proposal is that no future IETF document will be required. The configuration option is described directly in the universal configuration IANA registry.

## 4. The Universal IPv6 Configuration option

The option data is described using the schema language CDDL [RFC8610], encoded in CBOR [RFC7049].

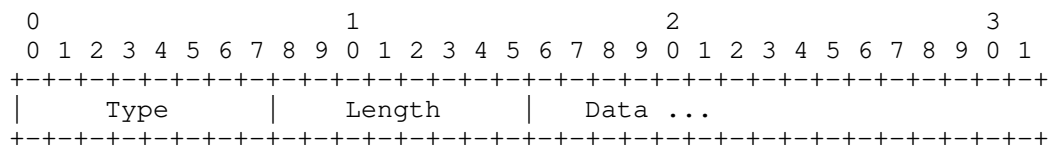


Figure 1: IPv6 Configuration Option Format

Fields:

Type: 42 for Universal IPv6 Configuration Option

Length: The length of the option (including the type and length fields) in units of 8 octets.



Data: CBOR encoded data.

The Option is zero-padded to nearest 8-octet boundary.

Example of an JSON instance of the option:

```
{
  "ietf": {
    "dns": {
      "dnssl": [
        "example.com"
      ],
      "rdnss": [
        "2001:db8::1",
        "2001:db8::2"
      ]
    },
    "nat64": {
      "prefix": "64:ff9b::/96"
    },
    "rio": [
      {
        "prefix": "::/0",
        "next-hop": "fe80::1"
      },
      {
        "prefix": "2001:db8::/32",
        "next-hop": "fe80::2"
      }
    ]
  }
}
```

The universal IPv6 Configuration option MUST be small enough to fit within a single IPv6 ND packet. It then follows that a single element in the dictionary cannot be larger than what fits within a single option. Different elements can be split across multiple universal configuration options (in separate packets). All IANA registered elements are under the "ietf" key in the dictionary. Private configuration information can be included in the option using different keys.

If information learnt via this option conflicts with other configuration information learnt via Router Advertisement messages, that is considered a configuration error. How those conflicts should be resolved is left up to the implementation.

## 5. CBOR encoding

It is recommended that the user can configure the option using JSON. Likewise an application registering interest in an option SHOULD be able to use string keys. The CBOR encoding to save space, uses integers for map keys. The mapping table between integer and string map keys are part of the IANA registry for the option.

Values -23-23 encodes to a single byte in CBOR, and these values are reserved for IETF used map keys.

## 6. Implementation Guidance

The purpose of this option is to allow users to use the RA as an opaque carrier for configuration information without requiring code changes in the option carrying infrastructure.

On the router there should be an API allowing a user to add an element, e.g. a JSON object [RFC8259] or a pre-encoded CBOR string to RAs sent on a given interface.

On the host side, an API SHOULD be available allowing applications to subscribe to received configuration elements. It SHOULD be possible to subscribe to configuration object by dictionary key.

The contents of any elements that are not recognized, either in whole or in part, by the receiving host MUST be ignored and the remainder of option's contents MUST be processed as normal.

An implementation SHOULD provide a "JSON interface" for configuring the option.

## 7. Implementation Status

The Universal IPv6 configuration option sending side is implemented in VPP (<https://wiki.fd.io/view/VPP> (<https://wiki.fd.io/view/VPP>)).

The implementation is a prototype released under Apache license and available at: <https://github.com/vpp-dev/vpp/commit/156db316565e77de30890f6e9b2630bd97b0d61d> (<https://github.com/vpp-dev/vpp/commit/156db316565e77de30890f6e9b2630bd97b0d61d>).

## 8. Security Considerations

Unless there is a security relationship between the host and the router (e.g. SEND), and even then, the consumer of configuration information can put no trust in the information received.

## 9. IANA Considerations

IANA is requested to add a new registry for the Universal IPv6 Configuration option. The registry should be named "IPv6 Universal Configuration Information Option".

The schema field follows the CDDL schema definition in [RFC8610].

Changes and additions to the registry follow the policies below [RFC8126]:

| Range                      | Registration Procedure |
|----------------------------|------------------------|
| -23-23                     | Standards Action       |
| 24-32767                   | Specification Required |
| 32768-18446744073709551615 | Expert Review          |

Table 1

A new registration requires a new CBOR key to parameter name assignment and a CDDL definition.

### 9.1. Universal configuration option

The IANA is requested to add the universal option to the "IPv6 Neighbor Discovery Option Formats" registry with the value of 42.

### 9.2. Initial objects in the registry

The PVD [RFC8801] elements and DNS [RFC8106]) are included to provide an alternative representation for the proposed new options in that draft.

### 9.3. Initial objects in the registry

#### 9.3.1. CDDL/JSON Mapping Parameters to CBOR

| Parameter Name / JSON key | CBOR Key |
|---------------------------|----------|
| ietf                      | -23      |
| pio                       | -22      |

|                    |         |         |
|--------------------|---------|---------|
| mtu                | -21     |         |
| +-----+            | +-----+ | +-----+ |
| rio                | -20     |         |
| +-----+            | +-----+ | +-----+ |
| dns                | -19     |         |
| +-----+            | +-----+ | +-----+ |
| nat64              | -18     |         |
| +-----+            | +-----+ | +-----+ |
| ipv6-only          | -17     |         |
| +-----+            | +-----+ | +-----+ |
| pvd                | -16     |         |
| +-----+            | +-----+ | +-----+ |
| prefix             | -15     |         |
| +-----+            | +-----+ | +-----+ |
| preferred-lifetime | -14     |         |
| +-----+            | +-----+ | +-----+ |
| valid-lifetime     | -13     |         |
| +-----+            | +-----+ | +-----+ |
| lifetime           | -12     |         |
| +-----+            | +-----+ | +-----+ |
| a-flag             | -11     |         |
| +-----+            | +-----+ | +-----+ |
| l-flag             | -10     |         |
| +-----+            | +-----+ | +-----+ |
| preference         | -9      |         |
| +-----+            | +-----+ | +-----+ |
| nexthop            | -8      |         |
| +-----+            | +-----+ | +-----+ |
| nssl               | -7      |         |
| +-----+            | +-----+ | +-----+ |
| dnss               | -6      |         |
| +-----+            | +-----+ | +-----+ |
| fqdn               | -5      |         |
| +-----+            | +-----+ | +-----+ |
| uri                | -4      |         |
| +-----+            | +-----+ | +-----+ |

Table 2

### 9.3.2. Key Registry

| CDDL  | Reference                                     |
|---|---|
| <pre> ietf = {   ? pio : [+ pio]   ? rio : [+ rio]   ? dns : dns   ? nat64: nat64   ? ipv6-only: bool   ? pvd : pvd }  dns = {   nssl : [* tstr]   dnss : [+ ipv6-address]   lifetime : uint .size 4 }  nat64 = {   prefix : ipv6-prefix } ipv6-only : bool  pvd = {   fqdn : tstr   uri : tstr   ? dns : dns   ? nat64: nat64   ? pio : [+ pio]   ? rio : [+ rio] } </pre> | <p>RFC8106</p> <p>RFC7050</p> <p>[v6only]</p> |

## 10. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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, <<https://www.rfc-editor.org/info/rfc4861>>.

- [RFC6919] Barnes, R., Kent, S., and E. Rescorla, "Further Key Words for Use in RFCs to Indicate Requirement Levels", RFC 6919, DOI 10.17487/RFC6919, April 2013, <<https://www.rfc-editor.org/info/rfc6919>>.
- [RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <<https://www.rfc-editor.org/info/rfc7049>>.
- [RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <<https://www.rfc-editor.org/info/rfc8610>>.

## 11. Informative References

- [RFC8106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 8106, DOI 10.17487/RFC8106, March 2017, <<https://www.rfc-editor.org/info/rfc8106>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.
- [RFC8801] Pfister, P., Vyncke, É., Pauly, T., Schinazi, D., and W. Shao, "Discovering Provisioning Domain Names and Data", RFC 8801, DOI 10.17487/RFC8801, July 2020, <<https://www.rfc-editor.org/info/rfc8801>>.

## Appendix A. Acknowledgements

Many thanks to Dave Thaler for feedback and suggestions of a more effective CBOR encoding. Thank you very much to Carsten Bormann for CBOR and CDDL help.

## Authors' Addresses

T. Winters  
QA Cafe

Email: [tim@qacafe.com](mailto:tim@qacafe.com)

O. Troan  
cisco

Email: [ot@cisco.com](mailto:ot@cisco.com)