Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 6, 2021

J. Dong
Z. Li
Huawei Technologies
C. Xie
C. Ma
China Telecom

November 2, 2020

Carrying Virtual Transport Network Identifier in IPv6 Extension Header
draft-dong-6man-enhanced-vpn-vtn-id-02

Abstract

A Virtual Transport Network (VTN) is a virtual network which has a customized network topology and a set of dedicated or shared network resources allocated from the network infrastructure. A VTN can be used as the underlay for one or a group of VPNs to provide enhanced VPN (VPN+) services. In packet forwarding, some fields in data packet needs to be used to identify the VTN the packet belongs to, so that the VTN-specific processing can be performed.

This document proposes a new option type to carry VTN ID in an IPv6 extension headers to identify the Virtual Transport Network (VTN) the packet belongs to. The procedure for processing of the VTN option is also specified.

Table of Contents

1. Introduction

Virtual Private Networks (VPNs) provide different groups of users with logically isolated connectivity over a common shared network infrastructure. With the introduction of 5G, new service types may require connectivity services with advanced characteristics comparing to traditional VPNs, such as strict isolation from other services or guaranteed performance. These services are refered to as "enhanced VPNs" (VPN+). [I-D.ietf-teas-enhanced-vpn] describes a framework and candidate component technologies for providing VPN+ services.

The enhanced properties of VPN+ require tighter coordination and integration between the underlay network resources and the overlay network. VPN+ service can be built on a Virtual Transport Network (VTN) which has a customized network topology and a set of dedicated

or shared network resources allocated from the underlay network. The overlay VPN together with the corresponding VTN in the underlay provide the VPN+ service. In the network, traffic of different VPN+ services need to be processed separately based on the topology and the network resources associated with the corresponding VTN.

[I-D.dong-teas-enhanced-vpn-vtn-scalability] describes the scalability considerations for VPN+, one of which is to improve the data plane scalability through the introduction of a dedicated identifier in data packets that is used to identify the VTN the packets belong to, so that VTN-specific packet processing can be performed. This is called Resource Independent (RI) VTN.

This document proposes a mechanism to carry the VTN ID in an IPv6 extension header [RFC8200] of a packet, so that the packet will be processed by network nodes using the network resources allocated to the corresponding VTN. The procedure for processing the VTN ID is also specified. This provides a scalable solution for enhanced VPN data plane, so that it may be used to support a large number of VTNs in an IPv6 network.

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

3. New IPv6 Extension Header Option for VTN

A new option type "VTN" is defined to carry the Virtual Transport Network Identifier (VTN ID) in an IPv6 packet header. Its format is shown as below:

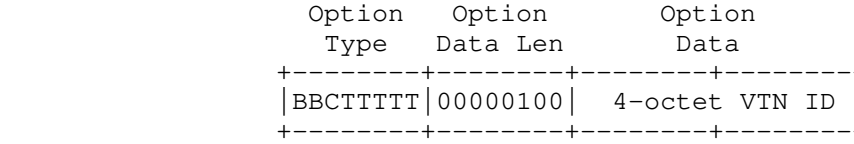


Figure 1. The format of VTN Option

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.

Option Data: 4-octet identifier which uniquely identifies a VTN.

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

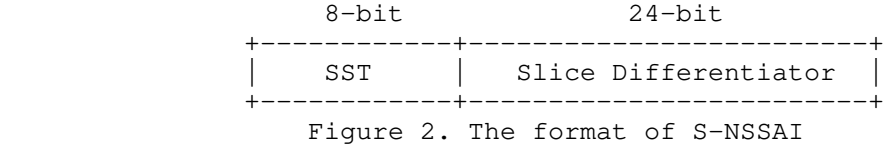


Figure 2. The format of S-NSSAI

4. 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 processed in forwarding plane by all the nodes along the path.

4.1. VTN Option Insertion

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

4.2. VTN based Packet Forwarding

On receipt of a packet with the VTN option, each network node which can parse the VTN option SHOULD use the VTN ID to identify the VTN the packet belongs to. This means the forwarding behavior is based on both the destination IP address and the VTN option. The destination IP address is used for the lookup of the next-hop node, and VTN-ID can be used to determine the set of network resources reserved for processing and sending the packet to the next-hop node.

The egress node of the IPv6 domain SHOULD decapsulate the outer IPv6 header.

There can be different implementations for reserving local network resources to the VTNs. For example, on one interface, a subset of forwarding plane resource allocated to a particular VTN can be considered as a virtual sub-interface with dedicated bandwidth and other associated resources. In packet forwarding, the IPv6 destination address of the received packet is used to identify the next-hop and the outgoing interface, and the VTN ID is used to further identify the virtual sub-interface which is associated with the VTN on the outgoing interface.

Routers which do not support Hop-by-Hop options header SHOULD ignore the Hop-by-Hop options header and forward the packet only based on the destination IP address. Routers which support Hop-by-Hop Options header, but do not support the VTN option SHOULD ignore the Hop-by-Hop option and continue to forward the packet only based on the destination IP address.

5. Operational Considerations

As described in [RFC8200], 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. This needs to be taken into consideration when VTN option is introduced to a network. The operator needs to make sure that all the network nodes in a VTN can either process Hop-by-Hop Options header in packet forwarding, or ignore the Hop-by-Hop Option header. 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 nodes in a VTN to process the Hop-by-Hop Options header if there is a nob for this.

6. 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

7. Security Considerations

TBD

8. Contributors

Zhibo Hu
Email: huzhibo@huawei.com

Lei Bao
Email: baolei7@huawei.com

9. Acknowledgements

The authors would like to thank Juhua Xu and James Guichard for their review and valuable comments.

10. References

10.1. 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>.

[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>.

10.2. Informative References

[I-D.dong-teas-enhanced-vpn-vtn-scalability] Dong, J., Li, Z., and F. Qin, "Virtual Transport Network (VTN) Scalability Considerations for Enhanced VPN", draft-dong-teas-enhanced-vpn-vtn-scalability-00 (work in progress), February 2020.

[I-D.ietf-teas-enhanced-vpn] Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Networks (VPN+) Service", draft-ietf-teas-enhanced-vpn-06 (work in progress), July 2020.

[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>.

[TS23501] "3GPP TS23.501", 2016, <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

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

Email: 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

Network Working Group
Internet Draft
Intended status: Standard
Expires: April 27, 2021

L. Dunbar
J. Kaippallimalil
Futurewei

October 27, 2020

IPv6 Solution for 5G Edge Computing Sticky Service
draft-dunbar-6man-5g-edge-compute-sticky-service-00

Abstract

This draft describes an IPv6 solution that enables packets from an application on a UE (User Equipment) sticking to the same application server location when the UE moves from one 5G cell site to another.

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

Table of Contents

1. Introduction

1.1. 5G Edge Computing Background

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

When a UE (User Equipment) initiates application packets using the destination address from a DNS reply or from its own cache, the packets from the UE are carried in a PDU session through 5G Core [5GC] to the 5G UPF-PSA (User Plan Function - PDU Session Anchor). The UPF-PSA decapsulate the 5G GTP outer header and forwards the packets from the UEs to the Ingress router of the Edge Computing (EC) Local Data Network (LDN). The LDN for 5G EC, which is the IP Networks from 5GC perspective, is responsible for forwarding the packets to the intended destinations.

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

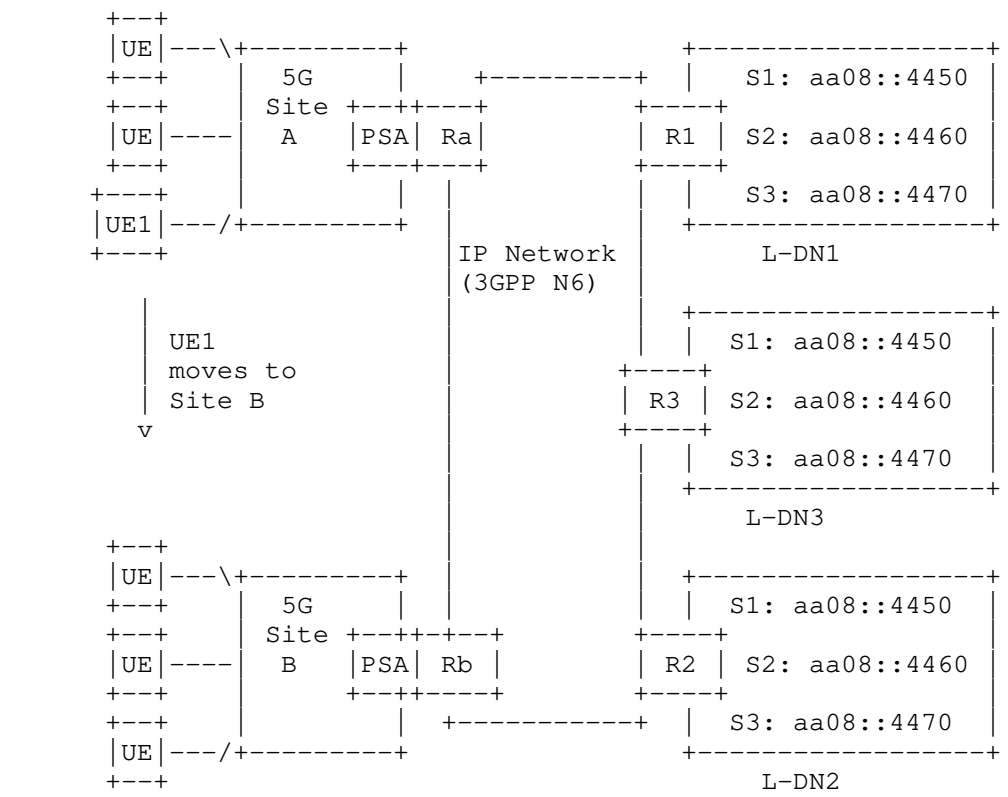


Figure 1: App Servers in different edge DCs

1.2. Problem #1: ANYCAST in 5G EC Environment

Increasingly, Anycast is used extensively by various application providers and CDNs because it is possible to dynamically load balance across multiple locations of the same address based on network conditions. BGP is an integral part in the way IP anycast usually functions. Within BGP routing there are multiple routes for the same IP address which are pointing to different 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 UEs that use their cached IP addresses instead of querying DNS when they move to a new location.

But, having multiple locations for the same ANYCAST address in 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. Same routing cost to multiple ANYCAST locations can cause packets from one flow to be forwarded to different locations, which can cause service glitches.

1.3. Problem #2: sticking to original App Server

When a UE moves to a new location but continues the same application flow, routers at the new location might choose the App Server closer to the new location. As shown in the figure below, when the UE1 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 instance 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 very often experienced by users.

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 UE to its original App Server location after the UE is anchored to a new UPF-PSA.

1.4. Problem #3: Application Server Relocation

When an Application Server is added to, moved, or deleted from a 5G Edge Computing Data Center, the routing protocol has to

propagate the changes to 5G PSA or the PSA adjacent routers. After the change, the cost associated with the site [5G-EC-Metrics] might change as well.

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

2. Conventions used in this document

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 5G Base Station and not only host 5G core functions.

gNB next generation Node B

L-DN: Local Data Network

PSA: PDU Session Anchor (UPF)

SSC: Session and Service Continuity

UE: User Equipment

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. Proposed IPv6 Based Solution

Many Mobile operators have adopted IPv6, at least on the UE side. With IPv6, routers within one Local Data Network can utilize the Flow Label in IPv6 Header to avoid sending packets from the same flow to the App Server at different locations with the same cost, (which is a great benefit for migrating to IPv6 network).

The solution described in this section is for the ingress router in the new LDN to steer an application flow from a UE to the App Server at the location that was used by the UE before the move.

3.1. Solution Overview

Here are some assumptions for the solution:

- Network is aware of the Sticky Services, by the Sticky Service Identifiers, which can be ANYCAST addresses or regular IP addresses. If an application service needs to be sticky in the 5G Edge Computing environment, the service ID is registered with the 5G Edge Computing controller.

From the network perspective, a sticky service is no longer sticky if there are no packets from the UE towards the service ID for a specified time. The Timer should be larger than a typical TCP session Timeout value.

Here is the overview of the proposed solution:

- Each Sticky service is assigned with a service ID, which can be one or a group of ANYCAST addresses. It is out of the scope of this document on how the Sticky Service ID is generated.
- Routers that have the Sticky service servers (physical or virtual) directly attached, a.k.a. egress routers in this document, are configured with the ACLs that can filter out the packets towards and from the attached App Servers. Those egress routers usually are the Gateway routers to the Edge computing DC,
  When the Egress router filters out packets that match the Sticky Service ID ACLs from the App Servers directly attached, it inserts a Sticky-Dst-SubTLV [Section 3.4]in the IPv6 extension header before forwarding the packets to its destination (i.e. the UE).

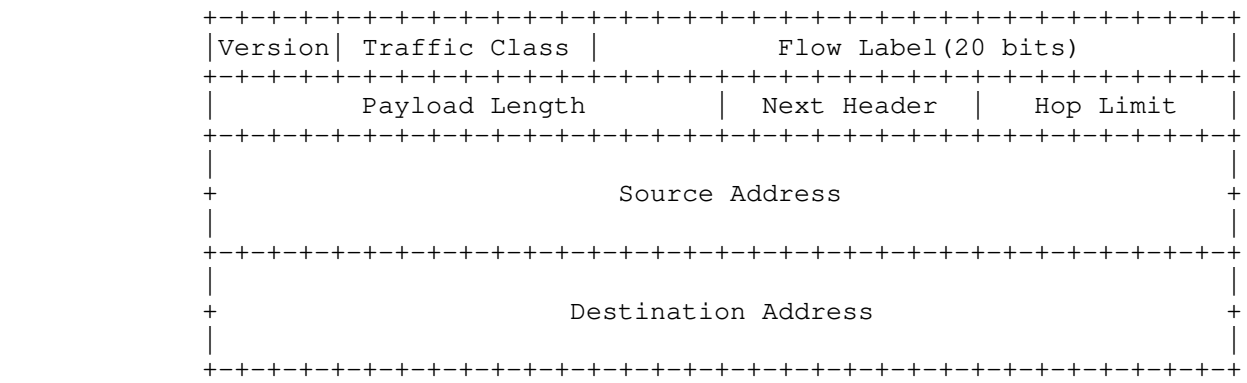
- It is expected that application client on a UE copy the extension header from its received IPv6 header to the subsequent packets belonging to the same application if the application prefers to same server instance when UE moves.

For UEs that cannot perform this procedure, section 4 and 5 describes the detailed steps.

- Ingress routers in the 5G LDN are also configured with the ACLs that can filter out packets whose destination addresses match with the Sticky Service Identifiers. When an ingress router filters out packets based on the Sticky Service ACL, it extracts the Sticky-Dst-SubTLV from the packet header if the packet header has the Sticky-Dst-SubTLV. The ingress router forwards the packets to the egress router extracted from the Sticky-Dst-SubTLV. If the Packet Header doesn't have the Sticky-Dst-SubTLV or fail to match the Sticky Service ACL, the packets are forwarded based on the least cost [5G-EC-Metrics].

3.2. Ensure one IPv6 Flow Go to the Same Location

RFC8200 specifies the IPv6 Header with the following required fields plus multiple extension headers.



For most cases, it is preferable for the routers not to split packets from one flow to the App Server located at different Edge DCs to minimize the service glitches. The TCP based session can be interrupted when packets from one flow are sent to different instances.

The site-specific ingress routers can use the IPv6 header Flow Label field to ensure the packets from one flow are forwarded to the same Egress router to which the App Server with the ANYCAST address is attached.

3.3. Discover the Egress Routers for App Servers

As shown in the Figure 1 above, the App Servers at each Edge DC are attached to one or two routers, which are usually the Gateway routers to the Edge DC. The DC Gateway routers can discover if there are any App Servers locally attached by sending ARP/ND scan of the ANYCAST address. They can propagate the addresses of the attached App Servers via BGP or IGP to other routers. When BGP is used, the Gateway router can use the Tunnel-Encap NLRI to inform other routers of the attached App Servers together with the supported encapsulation tunnels [Tunnel-Encap]. For the sake of easy description, those Gateway routers are called Egress routers to the App Servers, or A-ER, throughout this document.

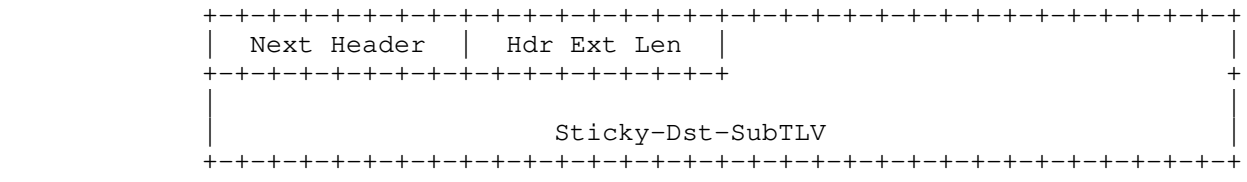
The routers directly connected to the 5G PSA (Packet Session Anchor) learn the addresses of the egress routers that the Edge Servers are attach via BGP UPDATE messages advertised from those egress routers. For the sake of easy description, the routers that are directly connected to PSA in each 5G site are called 5G site-specific Ingress Routers to the Local Data Network.

The 5G site-specific ingress routers learn the egress routers to the App Servers by the BGP UPDATE messages or the IGP advertisement originated from those routers.

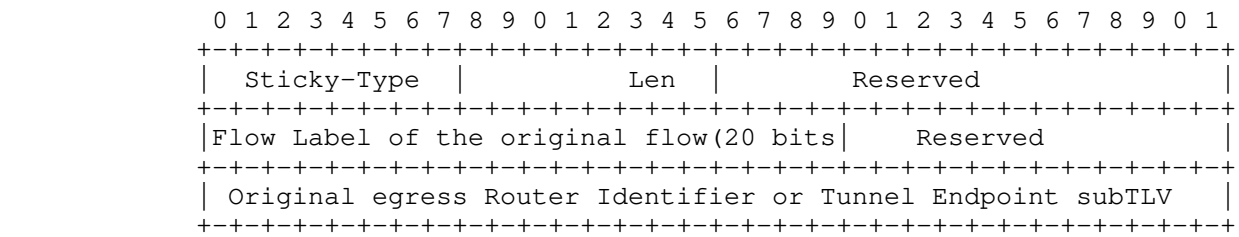
The following subsections describe how the RFC8200 specified IPv6 extension headers, Destination Header or routing Header, are used to achieve the goal for sticky to original App Server location after the UE moves.

3.4. Using Destination Extension Header

A new Sticky-Dst-SubTLV is added to the IPv6 Destination Options header. The IPv6 Destination Optino Header is specified by RFC8200 as having a Next Header value of 60:



Sticky-Dst-SubTLV is specified as:



Sticky-Type = 1: indicate the 32 bits identifier is used to represent the Egress Router to an App Server.

Original egress Router Identifier: assume that all the egress routers in the 5G Edge computing environment have a 32-bit identifier even though they might use an IPv6 address.

Sticky-Type = 2: indicate Tunnel Endpoint SubTLV [Tunnel-Encap] used to represent the Egress router from the original site before the UE moves to the new site.

Here is the processing at the Egress router:

- An Egress router is configured with an ACL for filtering out addresses of the App Servers that need sticky service.

  Let's assume that each Sticky Service can be identified by a Sticky Service ID, which can be one or a set of ANYCAST addresses.

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

- When an egress router receives a packet from an attached App Server that matches the ACL, the egress router inserts the Destination Extension Header with the Sticky-Dst-SubTLV to the data packet Header before forwarding the packet back to the UE.

Here are the steps to build the Sticky-Dst-subTLV:

- Copy the flow label from the received packet into the Flow Label Field of the Sticky-DST-subTLV
- If the egress router has its unique 32 bits identifier, such as the router's IPv4 loopback address, then:
  - o Set Sticky-Type = 1;
  - o Copy its own 32 bits identifier to the Original Egress Router Identifier field;
- Else
  - o Set Sticky-Type = 2;
  - o build the Tunnel-Endpoint SubTLV per Tunnel-Encap and insert into the field;

Here is the Expected behavior at the UE:

For edge computing services that need sticky service while UEs roaming among multiple 5G sites, the UEs need to extract the Destination Extension Header field from packets received from the App Server and inserts the extracted Destination Extension Header into the subsequent packets belonging to the same flow.

Granted, it might take some time for Edge Computing clients to adopt the practice of copying the IPv6 Destination Extension Header field from the received packets to the subsequent packets belonging to the same flow. However, once the egress routers and ingress routers for 5G local data network support the feature, more and more Edge Computing services would want to utilize this special feature by adding this step.

Section 4 describes the network layer processing if UEs do not perform the steps described here.

Here is the processing at the Ingress router:

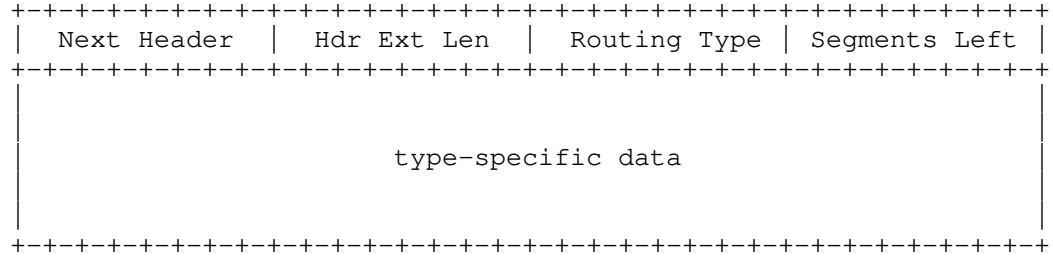
- An Ingress router is configured with an ACL for filtering out the applications that need sticky service.

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

- When an Ingress router receives a packet from the 5G PSA 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.
- The Ingress router extracts the Flow Label and the Egress Router Identifier from the Sticky-Dst-SubTLV.
- If the extracted Flow label matches the Flow Label in the IPv6 header, encapsulate the packet with the tunnel type that supported by the original egress router, using the extracted egress router address in the destination field of the outer header, and forward the packet. If Tunnel is not supported, refer to Section 5 for detailed processing.

3.5. Using IPv6 Routing Extension Header

Under this option, the Sticky-Dst-SubTLV specified in Section 3.4 is inserted into Type-Specific Data field of the Routing Extension Header of the IPv6 Header. RFC8200 describes the Routing header as used by an IPv6 source to list one or more intermediate nodes to be "visited" on the way to a packet's destination. The Routing header is identified by a Next Header value of 43 in the immediately preceding header and has the following format:



All the process for Egress router, Ingress router and UE are the same as using Destination Extension Header.

4. Achieve Sticky Service without dependence on UE

If UEs cannot perform the steps described in the section 3.4, LDN ingress routers have to do more.

This procedure assumes that the ingress routers are configured with an ACL to filter out packets whose destination addresses match with the Sticky Service Identifier.

Here are the processing steps:

- When an ingress router filters out packets that match the Sticky Service ID ACL, ingress router registers its own address, the source address of the filtered packets (i.e. the UE address) and the directly connected PSA address to the 5G EC Management System, or the 5G Network Exposure Function (NEF).
  The registration is associated with a Timer, which should be set to larger than a typical TCP session timeout value. When the Timer expires, the registration record is considered no-longer active or self-removed.

  The Sticky Service Registration Record should include:

  - o UE address
  - o The currently anchored PSA address
  - o The LDN router address that is directly connected to the PSA
  - o Timer
- When a UE is re-anchoring from PSA1 to PSA2 and there is an active sticky service record with the UE, 5GC EC management system sends a notification to the router that is in the sticky service record, which should be the router directly connected to the original PSA before the re-anchoring. The Notification includes the following information:
  - o the address of the new PSA that the UE is to be anchored, i.e. the PSA2 in the example above,
  - o the UE's new IP address
- Upon receiving the Notification from the 5G EC management system, the router (i.e. the one directly connected to the

old PSA) sends the "Sticky Service ID" (e.g. ANYCAST Address) + Sticky-Dst-SubTLV to the router directly connected to the new PSA.
- Upon receiving the Sticky-Dst-SubTLV, the router directly connected to the new PSA performs the same step as described in the Section3.4.

5. Forwarding to the desired A-ER without using Tunnel

Tunneling, i.e. encapsulating packets with an outer header that has the desired A-ER router address in the destination field of the outer header can guarantee to the packets to be delivered to the egress router where the Application Server is attached.

But the tunneling has the problem of single point of failure at the egress router. Or tunneling is not supported by the routers.

This section describes a solution that can prioritize egress routers for receiving packets without using tunneling.

Let's use this scenario to explain the solution:

One Application has its Application Servers attached to four different egress routers R1, R2, R3, and R4 respectively. For packets of a flow "A", the priority is sending to the App Server attached to R1. When R1 or the App Server attached to R1 fails, the packets of the flow "A" need to be forwarded to another server of the same application attached to R2, R3, or R4, depending on the network & utilization cost [5G-EC-Metrics]. This desired feature is called Location Preferred Forwarding.

Here is the procedure to achieve this type of forwarding for ANYCAST traffic:

- Each App server is configured with multiple ANYCAST addresses. All of them can be used as the Sticky Service ID for the App.
- Location preferred ANYCAST addresses:
  To make a specific location preferred. When failure occurs at the preferred location, the packets are forwarded to other locations.

For example, for the "App.net", four different ANYCAST addresses are allocated: L1, L2, L3, and L4:

- o ANYCAST address L1 has cost lowest, say 10, when attached to R1. L1 has higher cost, say 30, when attached to R2, R3, and R4.
- o ANYCAST L2 has the lowest cost when attached to R2. L2 has higher cost when attached to R1, R3, R4 respectively.
- o ANYCAST L3 has the lowest cost when attached to R3. L3 has higher cost when attached to R1, R2, R4 respectively, and
- o ANYCAST L4 has the lowest cost when attached to R4. L4 has higher cost for the instance attached to R1, R2, R3 respectively

For sticky service that needs to be sent to the App Server attached to R1, the ingress router to which 5G PSA is directly connected, can replace the destination address for packets matching with the Sticky Service ACL with the L1 for steering the packets towards R1. If R1 fails, the packets of the Flow "App.net" can be automatically sent to R2, R3, or R4 depending the network cost without any manual intervention.

For the procedure described in Section 3.4, the Application egress router needs to insert L1 into the Egress-Router-ID of the Sticky-Dst-SubTLV, so that ingress router at new 5G site can send packets to L1, which goes to R1 if there is no failure at R1, otherwise, packets are automatically forwarded to other egress routers for the Application.

For initial packet, local DNS resolver can reply L1, L2, L3 or L4 depending on where the DNS request come from to steer packets towards the preferred site for the Edge Application server.

6. Manageability Considerations

To be added.

7. Security Considerations

To be added.

8. IANA Considerations

To be added.

9. References

9.1. 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

9.2. 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.

[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.

10. Acknowledgments

Acknowledgements to Donald Eastlake 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: April 16, 2021

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

October 13, 2020

IPv6 Application of the Alternate Marking Method
draft-ietf-6man-ipv6-alt-mark-02

Abstract

This document describes how the Alternate Marking Method can be used as the passive performance measurement tool in an IPv6 domain and reports implementation considerations. It proposes how to define a new Extension Header Option to encode alternate marking technique and both Hop-by-Hop Options Header and Destination Options Header are considered.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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 16, 2021.

Table of Contents

1. Introduction

[RFC8321] and [RFC8889] 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 as Alternate Marking Method.

The Alternate Marking Method has become mature to be implemented and encoded in the IPv6 protocol and this document defines how it can be used to measure packet loss and delay metrics in IPv6.

The format of the IPv6 addresses is defined in [RFC4291] while [RFC8200] defines the IPv6 Header, including a 20-bit Flow Label and the IPv6 Extension Headers. The Segment Routing Header (SRH) is defined in [RFC8754] to apply Segment Routing over IPv6 dataplane (SRv6).

[I-D.fioccola-v6ops-ipv6-alt-mark] reported a summary on the possible implementation options for the application of the Alternate Marking Method in an IPv6 domain. This document, starting from the outcome of [I-D.fioccola-v6ops-ipv6-alt-mark], introduces a new TLV that can be encoded in the Options Headers (both Hop-by-Hop or Destination) for the purpose of the Alternate Marking Method application in an IPv6 domain. The case of SRH ([RFC8754]) is also discussed, anyway this is valid for all the types of Routing Header (RH).

2. Alternate Marking application to IPv6

The Alternate Marking Method requires a marking field. As mentioned, several alternatives have been analysed in [I-D.fioccola-v6ops-ipv6-alt-mark] such as IPv6 Extension Headers, IPv6 Address and Flow Label.

In consequence to the previous document and to the discussion within the community, it is possible to state that the only correct and robust choice that can actually be standardized would be the use of a new TLV to be encoded in the Options Header (Hop-by-Hop or Destination Option).

This approach is compliant with [RFC8200] indeed the Alternate Marking application to IPv6 involves the following operations:

- 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).

- 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. Anyway this does not impact the traffic since the measurement can be done only for the nodes configured to read the Option.

- 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, anyway it is to be expected that some routers cannot process it unless explicitly 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 the next sections, goes in this direction and it is used to identify a monitored flow.

Note that the FlowMonID is different from the Flow Label field of the IPv6 Header ([RFC8200]). Flow Label is used for application service, like load-balancing/equal cost multi-path (LB/ECMP) and QoS. Instead, FlowMonID is only used to identify the monitored flow. The reuse of flow label field for identifying monitored flows is not considered since it may change the application intent and forwarding behaviour. Furthermore the flow label may be changed en route and this may also violate the measurement task. Those reasons make the definition of the FlowMonID necessary for IPv6. Flow Label and FlowMonID within the same packet have different scope, identify different flows, and associate different uses.

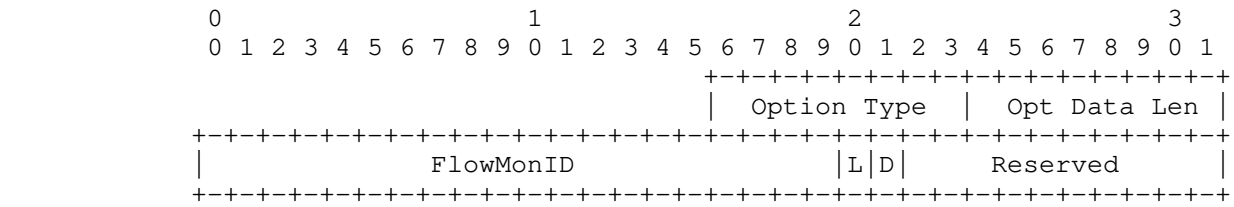
An important point that will also be discussed in this document is the the uniqueness of the FlowMonID and how to allow disambiguation of the FlowMonID in case of collision. [RFC6437] states that the Flow Label cannot be considered alone to avoid ambiguity since it could be accidentally or intentionally changed en route for compelling operational security reasons and this could also happen to the IP addresses that can change due to NAT. But the Alternate Marking is usually applied in a controlled domain, which would not have NAT and there is no security issue that would necessitate rewriting Flow Labels. So, for the purposes of this document, both IP addresses and Flow Label should not change in flight and, in some cases, they could be considered together with the FlowMonID for disambiguation.

3. Definition of the AltMark Option

The desired choice is to define a new TLV for the Options Extension Headers, carrying the data fields dedicated to the alternate marking method.

3.1. Data Fields Format

The following figure shows the data fields format for enhanced alternate marking TLV. This AltMark data is expected to be encapsulated in the IPv6 Options Headers (Hop-by-Hop or Destination Option).



where:

- Option Type: 8 bit identifier of the type of Option that needs to be allocated. Unrecognised Types MUST be ignored on receipt. 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 or not 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).

- Opt Data Len: The length of the Option Data Fields of this Option in bytes.

- FlowMonID: 20 bits unsigned integer. The FlowMon identifier is described hereinafter.

- L: Loss flag for Packet Loss Measurement as described hereinafter;

- D: Delay flag for Single Packet Delay Measurement as described hereinafter;

- 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 can be 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 removes 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. In this way an unrecognized Hop-by-Hop Option may be just ignored without impacting the traffic.

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. Anyway, in many cases the end-to-end measurement is not enough and it is required also the hop-by-hop measurement, so the most complete choice is 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.

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.

Another application that can be mentioned is the presence of a Routing Header, in particular it is possible to consider SRv6. SRv6 leverages the Segment Routing header which consists of a new type of routing header. Like any other use case of IPv6, Hop-by-Hop and Destination Options are useable 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 node that is an identity in the SR path.

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

- Destination Option => measurement only by node in Destination Address.

- Hop-by-Hop Option => every router on the path with feature enabled.

- Destination Option + any 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. 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.

In addition to the previous alternatives, for legacy network it is possible to mention a non-conventional application of the Destination Option for the hop by hop usage. [RFC8200] defines that the nodes along a path examine and process the Hop-by-Hop Options header only if Hop-by-Hop processing is explicitly configured. On the other hand, using the Destination Option for hop by hop action would cause worse performance than Hop-by-Hop. The only motivation for the hop by hop usage of Destination Options can be for compatibility reasons but in general it is not recommended.

5. Alternate Marking Method Operation

This section describes how the method operates. [RFC8321] introduces several alternatives but in this section the most applicable methods are reported and a new fied is introduced to facilitate the deployment and improve the scalability.

5.1. Packet Loss Measurement

The measurement of the packet loss is really straightforward. 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. 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 unambiguously recognizable by all network nodes along the path.

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. There is the condition on timing aspects explained in [RFC8321] that must be satisfied and it takes into considerations the different causes of reordering such as clock error, network delay. The consequence is that it is necessary to define a waiting interval where to get stable counters and to avoid these issues. 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.

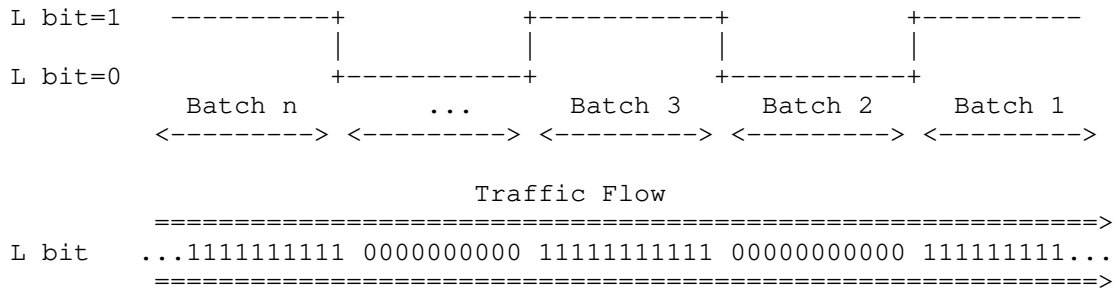


Figure 1: Packet Loss Measurement and Single-Marking Methodology using L bit

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. Anyway 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. 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.

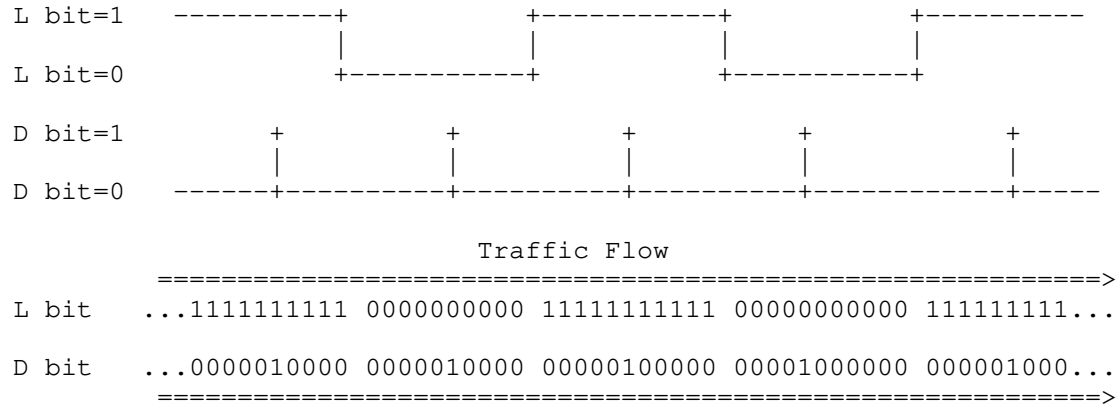


Figure 2: Double-Marking Methodology using L bit and D bit

Similar 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) 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.

- 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 FlowMon identifier field is to uniquely identify a monitored flow within the measurement domain. The field is set at the source node. The FlowMonID can be uniformly assigned by the central controller or algorithmically generated by the source node. The latter approach cannot guarantee the uniqueness of FlowMonID but it may be preferred for local or private network, where the conflict probability is small due to the large FlowMonID space.

5.3.1. Uniqueness of FlowMonID

It is important to note that if the 20 bit FlowMonID is set independently and pseudo randomly there is a chance of collision. So, in some cases, FlowMonID could not be sufficient for uniqueness.

In general the probability of a flow identifier uniqueness correlates to the amount of entropy of the inputs. For instance, 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 just 1206 flows. For a 32 bit identifier the 50% threshold jumps to 77,163 flows and so on. So, for more entropy, FlowMonID can either be combined with other identifying flow information in a packet (e.g. it is possible to consider the hashed 3-tuple Flow Label, Source and Destination addresses) or the FlowMonID size could be increased.

This issue is more visible when the FlowMonID is pseudo randomly generated by the source node and there needs to tag it with additional flow information to allow disambiguation. While, in case of a centralized controller, the controller should set FlowMonID by considering these aspects and instruct the nodes properly in order to guarantee its uniqueness.

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 [RFC8889].

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.

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. The advantage of the Alternate Marking method is that the marking bits are the only information that is exchanged between the network nodes. Therefore, network reconnaissance through passive eavesdropping on data-plane traffic does not allow attackers to gain information about the network performance. Moreover, Alternate Marking should usually be applied in a controlled domain and this also helps to limit the problem.

Harm to the Measurement: Alternate Marking measurements could be harmed by routers altering the marking of the packets or by an

attacker injecting artificial traffic. Since the measurement itself may be affected by network nodes along the path intentionally altering the value of the marking bits of IPv6 packets, the Alternate Marking should be applied in the context of a controlled domain, where the network nodes are locally administered and this type of attack can be avoided. Indeed the source and destination addresses are within the controlled domain and therefore it is unlikely subject to hijacking of packets, because it is possible to filter external packets at the domain boundaries. In addition, an attacker cannot gain information about network performance from a single monitoring point; it must use synchronized monitoring points at multiple points on the path, because they have to do the same kind of measurement and aggregation as Alternate Marking requires.

The privacy concerns of network measurement are limited because the method only relies on information contained in the Option Header without any release of user data. Although information in the Option Header is metadata that can be used to compromise the privacy of users, the limited marking technique seems unlikely to substantially increase the existing privacy risks from header or encapsulation metadata.

The Alternate Marking application described in this document relies on an 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].

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 assignments 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 act	chg	rest	Description	Reference
TBD	00	0	tbd	AltMark	[This draft]

8. Acknowledgements

The authors would like to thank Bob Hinden, Ole Troan, Tom Herbert, Stefano Previdi, Brian Carpenter, Eric Vyncke, Ron Bonica for the precious comments and suggestions.

9. References

9.1. 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>.

[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.fioccola-v6ops-ipv6-alt-mark] Fioccola, G., Velde, G., Cociglio, M., and P. Muley, "IPv6 Performance Measurement with Alternate Marking Method", draft-fioccola-v6ops-ipv6-alt-mark-01 (work in progress), June 2018.

[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>.

[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>.

[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>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

[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>.

[RFC8889] Fioccola, G., Ed., Cociglio, M., Sapio, A., and R. Sisto, "Multipoint Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8889, DOI 10.17487/RFC8889, August 2020, <https://www.rfc-editor.org/info/rfc8889>.

Authors' Addresses

Giuseppe Fioccola
Huawei
Riesstrasse, 25
Munich 80992
Germany

Email: giuseppe.fioccola@huawei.com

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing 100095
China

Email: zhoutianran@huawei.com

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

Email: mauro.cociglio@telecomitalia.it

Fengwei Qin
China Mobile
32 Xuanwumenxi Ave.
Beijing 100032
China

Email: qinfengwei@chinamobile.com

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

Email: pangran@chinaunicom.cn

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: 26 April 2021

R. Hinden
Check Point Software
G. Fairhurst
University of Aberdeen

23 October 2020

IPv6 Minimum Path MTU Hop-by-Hop Option
draft-ietf-6man-mtu-option-04

Abstract

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

This Hop-by-Hop option is intended to be used in environments like Data Centers and on paths between Data Centers, to allow them to better take advantage of paths able to support a large Path MTU. The method could also be useful in other environments, including the general Internet.

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 26 April 2021.

Table of Contents

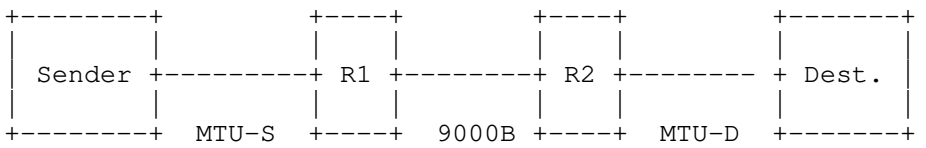
1. Introduction

This draft proposes a new IPv6 Hop-by-Hop Option to be used to record the minimum of the Maximum Transmission Unit (MTU) along the forward path between the source and destination hosts. The source host creates a packet with this option and fills 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 an input to the method that sets the Path MTU (PMTU) used by upper layer protocols.

1.1. Example Operation

The figure below illustrates the operation of the method. In this case, the path between the source and destination hosts comprises three links, the sender 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.



Three scenarios are described:

* Scenario 1, considers all links to have an 9000 byte MTU and the method is supported by both routers. The PMTU is therefore 9000 bytes.

* Scenario 2, considers the link to the 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 to Big (PTB) message had not been delivered to the sender [RFC8201].

These scenarios are summarized in the table below.

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

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

IPv6 as specified in [RFC8200] allows nodes to optionally process Hop-by-Hop headers. 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. This fails to work in various cases, including when nodes in the middle of the network do not send ICMP PTB messages, or rate-limited messages to the point of not making them a useful mechanism, or do not have a return path to the source host.

This results in many transport 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 can gain benefit from sending large packets are forced to use IPv6 Fragmentation [RFC8200], which can reduce the reliability of Internet communication [RFC8900].

Transport encapsulations and network-layer tunnels further reduce the the payload size available for a transport 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 small packets can limit performance, e.g., when packet processing is limited by the packet rate. The potential of multi-gigabit Ethernet will not be realized if the packet size is limited to 1280 bytes, because this exceeds the packet per second rate that most nodes can process. For example, the packet per second rate required to reach wire speed on a 10G Ethernet link with 1280 byte packets is about 977K packets per second (pps), vs. 139K pps for 9000 byte packets. A significant difference.

The purpose of the this draft 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.

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

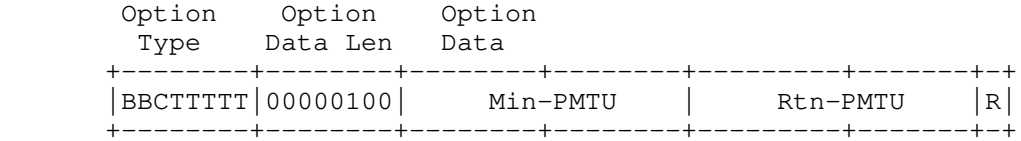
This Hop-by-Hop Option header is intended to be used in environments such as Data Centers and on paths between Data Centers, to allow a host to better take advantage of a path that is able to support a large PMTU.

The design of the option is sufficiently simple that it could be executed on a router's fast path. A strong pull from router vendors customers will be required to create critical mass for this to happen. This could initially be the case for connections within and between Data Centers.

The method could also be useful in other environments, including the general Internet, if and when this Hop-by-Hop Option is supported on these paths.

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

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



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 each value field in Option Data field supports PMTU values from 0 to 65,535 octets.

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] should 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.

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.

6. Router, Host, and Transport Behaviors

6.1. Router Behavior

Routers that are not configured to support Hop-by-Hop Options SHOULD ignore this option and SHOULD forward the packet.

Routers that support Hop-by-Hop Options, but that are not configured to support this option SHOULD ignore the option and SHOULD forward the packet.

Routers that recognize this option SHOULD 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.

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

Discussion:

* The design of this option makes it feasible to be implemented within the fast path of a router, because the processing requirements are minimal.

6.2. Host Behavior

When requested to send an IPv6 packet with the Minimum Path MTU option, the source host includes the option in an outgoing packet. The source host SHOULD 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. If this value is not updated, the field MUST be set to zero.

The source host SHOULD set the Rtn-PMTU field to the cached value of the reported Min-PMTU value for the flow (see Section 6.3.3). If this value is not set, for example, because there is no cached reported Min-PMTU value, the field 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.

6.3. Transport Behavior

6.3.1. Including the Option in an Outgoing Packet

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

When it includes the option, the host supplies the previously cached value of the received Minimum Path MTU for the flow to set the Rtn-PMTU field (see Section 6.3.3). If a valid cached received Minimum Path MTU is not available, the Rtn-PMTU field value MUST be set to zero.

The source host MAY request the destination host to send a packet carrying the option by setting the R-Flag. The R-Flag SHOULD NOT be set when the Minimum Path MTU Option was sent solely to feedback the return Path MTU.

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 packets carrying a SYN segment as part of the connection set up, or can periodically be sent in packets carrying other segments. Including this packet in a SYN could increase the probability that the SYN segment is lost when routers on the path drop packets with this option (see Section 6.3.5). NOTE: A TCP connection 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.

* 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 use of this option with DNS and DNSSEC over UDP ought 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.

* Applications that use Anycast should include this option in all packets, because the actual destination host will vary due to the nature of Anycast.

6.3.2. Validation by the Upper Layer Protocol

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. 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. 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 minimum Path MTU option without further processing.

6.3.3. Receiving the Option

An upper layer protocol that receives a Minimum Path MTU Option included with a valid packet caches the value of the last received Min-PMTU. This value is specific to the instance of the upper layer protocol (i.e., matching the IPv6 flow ID, port-fields in UDP or the SPI in IPsec [RFC4301], etc), not to the pair of source and destination addresses, because network devices can make forwarding decisions that impact the PMTU of a flow based on the presence and value of the packet's upper layer fields.

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 Minimum Path MTU Option with the R-Flag SHOULD include the Minimum Path MTU 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 looses state.

Path characteristics can change and the actual PMTU could increase or decrease over time. For instance, following a path change when packets are then forwarded over a link with a different MTU than that previously used. To bound the delay in discovering a change in the

actual PMTU, a sender with a link MTU larger than the current PMTU SHOULD periodically send the Minimum Path MTU 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.

Discussion:

* Some upper layer protocols send packets less frequently than packets that 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 sender and destination. 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 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 the there is a layer 2 device with a lower MTU that does not perform IPv6 forwarding.

* The actual PMTU may be larger than the Rtn-PMTU value because of a corrupted, delayed or mis-ordered response. A source host SHOULD ignore a Rtn-PMTU value larger than the MTU configured for the outgoing link.

Using the method has the potential to complete discovery of the correct value in a single round trip time, even over paths that have successive links each configured with a lower MTU.

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 be maintained for each transport endpoint, or for each network address.

6.3.5. 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 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

No IANA actions are requested in this document.

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

8. Security Considerations

This section discusses the security considerations. It first 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. Network Layer Host Processing

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

Reception of this packet will incur receive 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 apply 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.

8.2. 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 minimum Path MTU 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.

8.3. 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 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 cause the sender to add the option to a packet sent to the source. 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 minimum Path MTU option is received. This attack could be mitigated by limiting the sending of the minimum Path MTU option in reply to incoming packets that carry the option.

8.4. 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 Section 8.2. 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 minimum Path MTU option corresponding to the flow.

* Processing of the option could cause a destination host to add the minimum Path MTU 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.

8.5. 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 6.3.5.

When a forged packet cause a packet to be sent including the minimum Path MTU option, and the return path does not forward packets with this option, the packet will be dropped Section 6.3.5. 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 a R-Flag by including the option in a packet that does not carry application data.

9. Acknowledgments

A somewhat similar mechanism was proposed for IPv4 in 1988 in [RFC1063] by Jeff Mogul, C. Kent, Craig Partridge, and Keith McCloghire. It was later obsoleted in 1990 by [RFC1191] the current deployed approach to Path MTU Discovery.

Helpful comments were received from Tom Herbert, Tom Jones, Fred Templin, Ole Troan, [Your name here], and other members of the 6MAN working group.

10. Change log [RFC Editor: Please remove]

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.
* ICMP 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.

11. References

11.1. 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>.

11.2. Informative References

[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>.

[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>.

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: gorry@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.

Table of Contents

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.

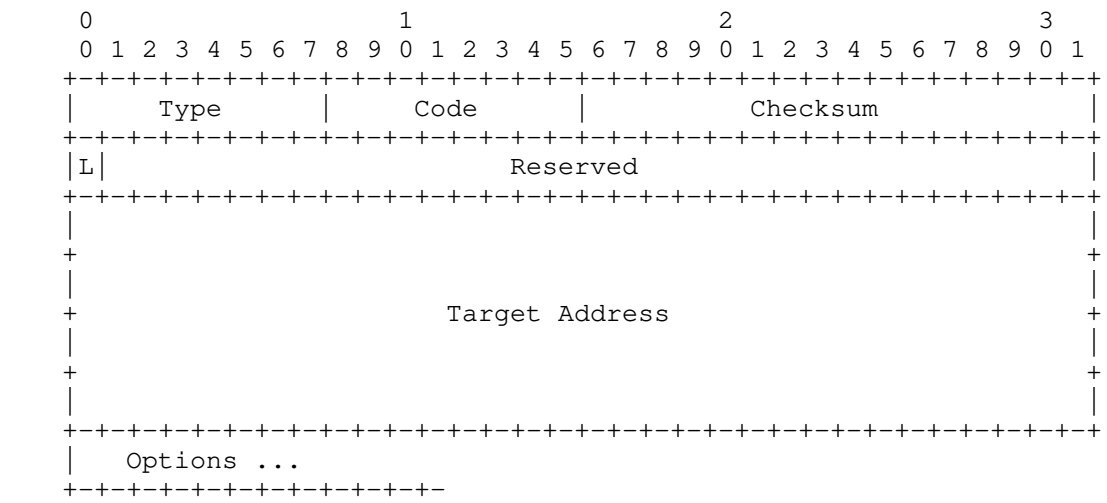
To facilitate LL communication on the local link, given a destination GUA or ULA:

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

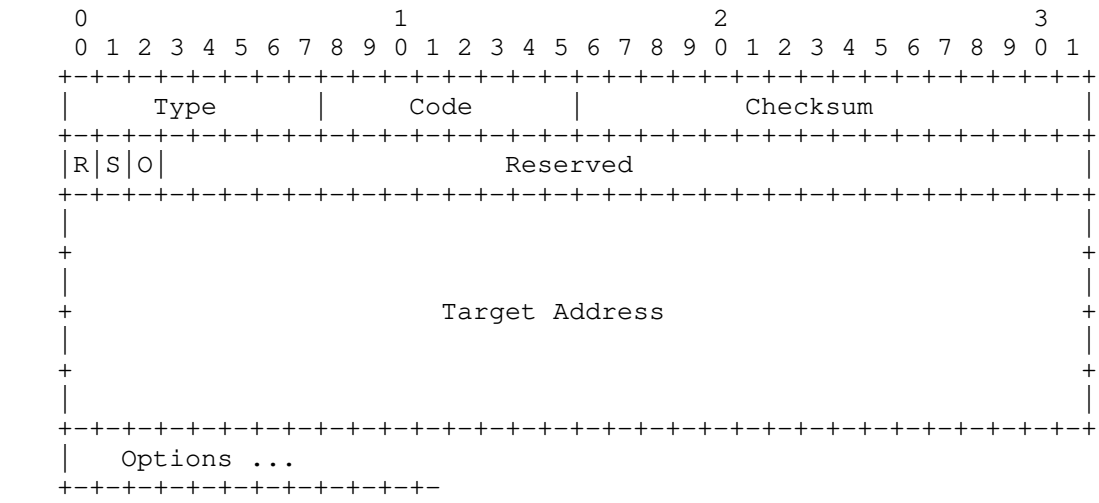
Destination Address
Either the solicited-node multicast address corresponding to the target GUA or ULA address, or the target GUA or ULA address.

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 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 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

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: May 20, 2021

W. Cheng
China Mobile

November 16, 2020

Compressed SRv6 SID List Requirements
draft-srcompdt-spring-compression-requirement-02

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 May 20, 2021.

Table of Contents

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 [I-D.ietf-spring-srv6-network-programming]. 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 the requirements for proposals to SR over IPv6 SID list compression.

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], [I-D.ietf-spring-srv6-network-programming]

SID numbering space: may be implemented as

- a single IGP instance
- a single IGP level or area
- two or more autonomous systems that coordinate SID numbering space
- 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], [I-D.ietf-spring-srv6-network-programming] 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.

- E(proposal, segment list).

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

- ES(proposal, segment list) = 1 - E(proposal, segment list)/E(SRv6 base, 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:

- D.PRS(segment list): number of headers parsed during processing of the segment list, starting from and including the IPv6 header.

- D.LKU(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.

- 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. Functional Requirements

4.1.1. 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.

4.1.2. 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.

4.2. Operational Requirements

4.2.1. Lossless Compression

Description: The segments of the compressed SID list MUST be equivalent to the original SID List. For example, a strict path TE SID List is not compressed to a loose path TE SID list.

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. Scalability Requirements

4.3.1. 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.

4.3.2. 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.

4.3.3. 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.

5. Protocol Design Requirements

5.1. SRv6 Base Coexistence

Description: The compression proposal MUST support deployment in existing 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

- Supports simultaneous deployment at a node with current SRv6 SIDs.
- Supports simultaneous deployment at a node with current SRv6 control plane.
- Supports simultaneous operation of current SRv6 paths with compressed paths.

- Supports the behaviors in [I-D.ietf-spring-srv6-network-programming].
- Does not require removal of existing IPv6 address planning.

6. IANA Considerations

This document has no requests to IANA.

7. Security Considerations

TBD

8. Contributors

The following individuals contributed to this draft

Chongfeng Xie, China Telecom, xiechf@chinatelecom.cn

Ron Bonica, Juniper Networks, rbonica@juniper.net

Darren Dukes, Cisco Systems, ddukes@cisco.com

Cheng Li, Huawei, c.l@huawei.com

Peng Shaofu, ZTE, peng.shaofu@zte.com.cn

Wim Henderickx, Nokia, wim.henderickx@nokia.com

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

9. 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-08 (work in progress), October 2020.

[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-05 (work in progress), November 2020.

[I-D.ietf-idr-bgpls-srv6-ext]
Dawra, G., Filsfils, C., Talaulikar, K., Chen, M., daniel.bernier@bell.ca, d., and B. Decraene, "BGP Link State Extensions for SRv6", draft-ietf-idr-bgpls-srv6-ext-04 (work in progress), November 2020.

[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-13 (work in progress), October 2020.

[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-11 (work in progress), October 2020.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Decraene, B., Francois, P., Voyer, D., Clad, F., and P. Camarillo, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-04 (work in progress), August 2020.

[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-09 (work in progress), November 2020.

[I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., daniel.bernier@bell.ca, 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-03 (work in progress), September 2020.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-24 (work in progress), October 2020.

[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>.

Appendix A. Proposed Requirements

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

A.1. Introduction

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.

A.2. Requirements

A.2.1. SRv6 Based

Description: A solution to compress SRv6 SID Lists SHOULD be based on the SRv6 architecture, control plane and data plane.

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:

- U.RFC8402: utilizes [RFC8402].
- U.RFC8754: utilizes [RFC8754].
- U.PGM: utilizes [I-D.ietf-spring-srv6-network-programming].
- U.IGP: utilizes [I-D.ietf-lsr-isis-srv6-extensions].
- U.BGP: utilizes [I-D.ietf-bess-srv6-services].
- U.POL: utilizes [I-D.ietf-spring-segment-routing-policy].
- U.BLS: utilizes [I-D.ietf-idr-bgpls-srv6-ext].
- U.SVC: utilizes [I-D.ietf-spring-sr-service-programming].
- U.OAM: utilizes [I-D.ietf-6man-spring-srv6-oam].
- U.ALG: utilizes [I-D.ietf-lsr-flex-algo].
- U.TOT: the total number of specifications utilized.

Each cell contains "yes" for utilizes, or "no" for does not utilize. U.TOT counts the number of "yes" in each column.

A.2.2. SRv6 Functionality

Description: A solution to compress an SRv6 SID list MUST support the functionality of SRv6. This requirement and set of metrics is meant to assess whether a proposal that is not fully SRv6 based, as evaluated in section "SRv6 Based", provides equivalent functionality to SRv6. Such a proposal may utilize different control planes and or data planes.

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:

- F.SID: Supports SRv6 SIDs described in [RFC8402]
- F.SCOPE: Supports globally and locally scoped SIDs described in [RFC8402]
- F.PFX: Supports prefix SIDs described in [RFC8402] and [I-D.ietf-spring-srv6-network-programming]
- F.ADJ: Supports adjacency SIDs described in [RFC8402] and [I-D.ietf-spring-srv6-network-programming]
- F.BIND: Supports binding SIDs described in [RFC8402] and [I-D.ietf-spring-srv6-network-programming]
- F.PEER: Supports BGP peering SIDs described in [RFC8402] and [I-D.ietf-spring-srv6-network-programming]
- F.SVC: Supports L3 and L2 VPN service SIDs described in [I-D.ietf-spring-srv6-network-programming]
- F.ALG: Supports flexible algorithms described in [I-D.ietf-lsr-flex-algo]
- F.TILFA: Supports TI-LFA as described in [I-D.ietf-rtgwg-segment-routing-ti-lfa]
- F.SEC: Supports securing an SR domain with ingress filtering as defined in [RFC8754]
- F.IGP: Supports distributing topological SIDs and behaviors via ISIS as described in [I-D.ietf-lsr-isis-srv6-extensions]
- F.BGP: Supports BGP VPNs as described in [I-D.ietf-bess-srv6-services]
- F.POL: Supports SR policies and steering traffic over those policies as described in [I-D.ietf-spring-segment-routing-policy]
- F.BLS: Supports Link State distribution via BGP as described in [I-D.ietf-idr-bgpls-srv6-ext]
- F.SFC: Supports stateless service programming as described in [I-D.ietf-spring-sr-service-programming]
- F.PING: Supports pinging a SID to verify the SID is implemented as described in [I-D.ietf-6man-spring-srv6-oam]

- o F.TOT: The total number of SRv6 functionality metrics supported by a proposal

Each cell contains the specification name documenting the functionality. F.TOT counts the number of specifications in each column.

A.2.3. Heterogeneous SID lists

Description: The compression proposal SHOULD support a combination of compressed and non-compressed segments in a single path.

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.

Author's Address

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com

Network Working Group F. Templin, Ed.
Internet-Draft The Boeing Company
Intended status: Standards Track A. Whyman
Expires: May 27, 2021 MWA Ltd c/o Inmarsat Global Ltd
November 23, 2020

IPv6 Neighbor Discovery Overlay Multilink Network Interface (OMNI) Option
draft-templin-6man-omni-option-02

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 May 27, 2021.

Table of Contents

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. Uses for the option are specified in IPv6 over specific link layer documents such as [I-D.templin-6man-omni-interface]. 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:

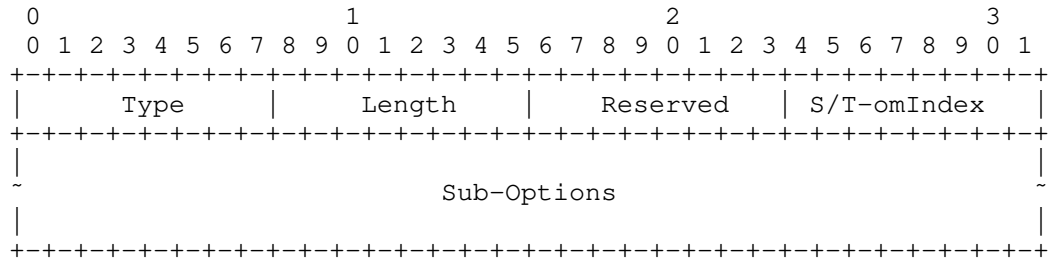


Figure 1: OMNI Option Format

In this format:

- Type is set to TBD.
- Length is set to the number of 8 octet blocks in the option.
- Reserved is a 1-octet field that encodes the value 0 (an unrelated Prefix Length field is found in the source or destination Link-Local Address (LLA) of the IPv6 ND message that incldues the OMNI option).
- 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.
- 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 a single IPv6 ND message contains multiple OMNI options, the first option is processed and any additional options are ignored.

An OMNI option may include full or partial information for the neighbor. If an OMNI option with full information is received, its contents provide new information and/or update any previously cached information. If an OMNI option with partial information is received, it contents provide new information and/or update only the corresponding previously cached information. 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:

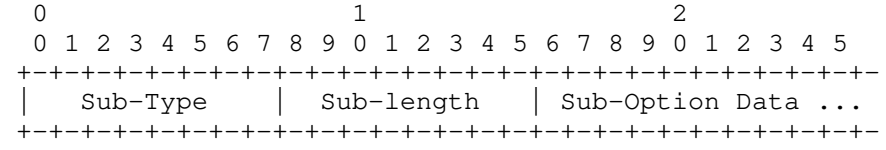


Figure 2: Sub-Option Format

- Sub-Type is a 1-octet 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

Figure 3

  Sub-Types 253 and 254 are reserved for experimentation; Sub-Type 255 is reserved by IANA.

- Sub-Length is a 1-octet field that encodes the length of the Sub-Option Data (i.e., ranging from 0 to 255 octets).

- Sub-Option Data is a block of data with format determined by Sub-Type.

During processing, unrecognized Sub-Options are ignored and the next Sub-Option processed until the end of the OMNI option is reached.

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

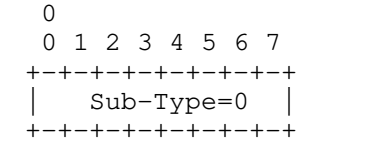


Figure 4: Pad1

- Sub-Type is set to 0. If multiple instances appear in the same OMNI option all are processed.
- No Sub-Length or Sub-Option Data follows (i.e., the "Sub-Option" consists of a single zero octet).

3.1.2. PadN

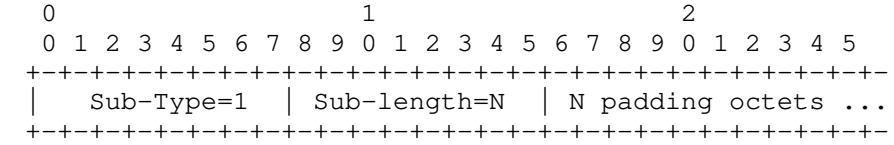


Figure 5: PadN

- Sub-Type is set to 1. If multiple instances appear in the same OMNI option all are processed.
- Sub-Length is set to N (from 0 to 255) being the number of padding octets that follow.
- 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)

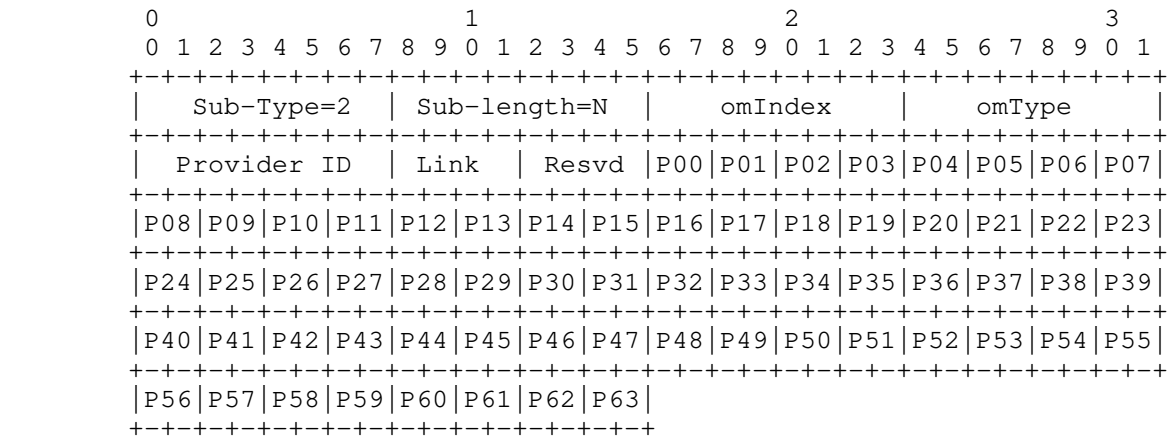


Figure 6: Interface Attributes (Type 1)

- Sub-Type is set to 2. If multiple instances with different omIndex values appear in the same OMNI option all are processed; if multiple instances with the same omIndex value appear, the first is processed and all others are ignored

- Sub-Length is set to N (from 1 to 255) that encodes the number of Sub-Option Data octets that follow.

- omIndex is a 1-octet field containing a value from 0 to 255 identifying the underlying interface for which the interface attributes apply.

- omType is a 1-octet field containing a value from 0 to 255 corresponding to the underlying interface identified by omIndex.

- Provider ID is a 1-octet field containing a value from 0 to 255 corresponding to the underlying interface identified by omIndex.

- 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").

- Resvd is reserved for future use.

- 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.

4. IANA Considerations

The IANA is instructed to allocate a Type number TBD 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 also defines an 8-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]
3-252	Unassigned	
253-254	Experimental	[RFCXXXX]
255	Reserved	[RFCXXXX]

Figure 7: 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. The document supports ICAO Document 9896 Draft Edition 3 (work-in-progress).

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>.

[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

[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-50 (work in progress), October 2020.

[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: 6 May 2021

T. Winters
QA Cafe
O. Troan
cisco

2 November 2020

The Universal IPv6 Configuration Option
draft-troan-6man-universal-ra-option-04

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 and DHCP. This document proposes a new universal option for RA and DHCP 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 6 May 2021.

Copyright Notice

Table of Contents

1. Introduction

This document proposes a new universal option for the Router Advertisement IPv6 ND message [RFC4861] and DHCPv6 [RFC8415]. Its purpose is to use the RA and DHCP messages as opaque carriers for configuration information between an agent on a router or DHCP server and host / host application.

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 options, and the IETF spending an inordinate amount of time arguing over new configuration options.

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 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].

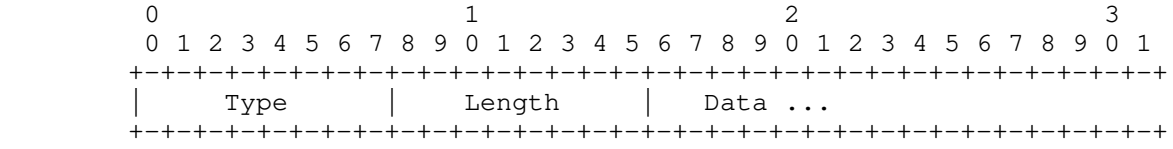


Figure 1: IPv6 Configuration Option Format

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:

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

The universal IPv6 Configuration option MUST be small enough to fit within a single IPv6 ND or DHCPv6 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 or via DHCPv6, that is considered a configuration error. How those conflicts should be resolved is left up to the implementation.

5. Implementation Guidance

The purpose of this option is to allow users to use the RA or DHCPv6 as an opaque carrier for configuration information without requiring code changes in the option carrying infrastructure.

On the router or DHCPv6 server side 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 or to DHCPv6 messages sent to a client.

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.

6. 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).

7. 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.

8. 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". Changes and additions to the registry require expert review [RFC8126].

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

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

The IANA is requested to add the universal option to the "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option Codes" registry.

8.1. Initial objects in the registry

The PVD [RFC8801] elements (and PIO, RIO [RFC4191]) are included to provide an alternative representation for the proposed new options in that draft.

CDDL Description	Reference

ietf = {
  ? dns : dns
  ? nat64: nat64
  ? ipv6-only: bool
  ? pvd : pvd
  ? mtu : uint .size 4
  ? rio : rio
}	

pio = {
  prefix : tstr
  ? preferred-lifetime : uint
  ? valid-lifetime : uint
  ? a-flag : bool
  ? l-flag : bool
}
	[RFC4861]

rio_route = {
  prefix : tstr
  ? preference : (0..3)
  ? lifetime : uint
  ? mtu : uint .size 4
  ? nexthop: tstr
}
	[RFC4191]
	
	[this]

rio = {
  routes : [+ rio_route]
}	

dns = {
 dnssl : [* tstr]
 rdnss : ipv6-addresses : [* tstr]
 ? lifetime : uint
}
	[RFC8106]

nat64 = {
  prefix : tstr
}
	[RFC7050]

ipv6-only : bool	[v6only]

pvd = {
  fqdn : tstr
  uri : tstr
  ? dns : dns
  ? nat64: nat64
  ? pio : pio
  ? rio : rio
}
		[pvd]

9. 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>.

[RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <https://www.rfc-editor.org/info/rfc8415>.

[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>.

10. Informative References

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <https://www.rfc-editor.org/info/rfc4191>.

[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>.

Authors' Addresses

T. Winters
QA Cafe

Email: tim@qacafe.com

O. Troan
cisco

Email: ot@cisco.com