

SPRING Workgroup
Internet-Draft
Intended status: Standards Track
Expires: April 28, 2022

S. Boutros, Ed.
S. Sivabalan, Ed.
H. Shah
Ciena Corporation
J. Uttaro
ATT
D. Voyer
Bell Canada
B. Wen
Comcast
L. Jalil
Verizon
October 25, 2021

A Simplified Scalable ELAN Service Model with Segment Routing Underlay
draft-boutros-spring-elan-services-over-sr-00

Abstract

This document proposes a new approach for realizing Ethernet LAN (ELAN) services with an objective of leveraging Segment Routing Control plane to achieve high scalability, faster network convergence, and reduced operational complexity. Furthermore, it naturally brings the benefits of All-Active multihoming as well as MAC learning in data-plane.

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 28, 2022.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Terminology	4
3. Abbreviations	5
4. Control Plane Behavior	5
4.1. Service discovery	5
4.2. All-Active Service Redundancy	6
4.3. Mass service withdrawal	6
4.4. E-Tree Support	6
5. Data Plane Behavior	6
5.1. Unicast Traffic	7
5.2. BUM Traffic	8
5.3. Data Plane MAC learning	8
5.3.1. Single Home CE	9
5.3.2. Multi-Home CE	9
5.4. ARP suppression	10
5.5. Distributed Anycast Gateway	10
5.6. Multi-pathing	10
5.7. E-Tree Support	11
6. Benefits of ELAN over SR	11
7. Security Considerations	11
8. IANA Considerations	11
9. Acknowledgements	11
10. References	11
10.1. Normative References	11
10.2. Informative References	12
Authors' Addresses	12

1. Introduction

Virtual Private LAN Service (VPLS) is based on Pseudo-Wire (PW) construct which identifies both the service type and the service termination node in both control and data planes. RFCs 4761 and 4762 specify mechanisms to signal PW for VPLS services using BGP and LDP respectively. An ingress Provider Edge (PE) node needs to maintain a PW per VPLS instance for each egress PE node. So, if we assume 10K ELAN instances over a network of 100 PE nodes, each PE node needs to setup and maintain approximately 1M PWs which can easily become a scalability bottleneck in large scale deployment.

As described in RFC7432, Ethernet Virtual Private Network (EVPN) technology builds ELAN services similar to BGP-based IP-VPN services with additional features such as MAC address learning in control plane, All-Active multihoming, etc. It eliminates the need for PWs, and hence the scale problem associated with PWs. However, an egress PE node cannot unambiguously identify ingress PE node in data-plane. As such, EVPN requires control plane mechanisms for MAC advertisement and learning which increases control plane complexity and overhead.

The goal of the proposed approach is to greatly simplify control plane functions and minimize the amount of control plane messages PE nodes have to process. In this version of the document, we assume Segment Routing (SR) underlay network. A future version of this document will generalize the underlay network to both classical MPLS and SR technologies.

The proposed approach does not require PW, and hence the control plane complexity and message overhead associated with signaling and maintaining PWs are eliminated.

An ELAN instance is uniquely identified by Segment ID (SID) regardless of the number of service termination points. Such a SID will be referred to as "Service SID" in the rest of the document. The number of states maintained at a PE node is equal to the number of ELAN instances in the corresponding broadcast domain. Referring to the above example, each PE node now needs to maintain states for 10K ELAN service instances as opposed to 1 M PWs in the case of classical VPLS model in data and control planes. A node can advertise service SID(s) of the ELAN instance(s) that it hosts via BGP for auto-discovery purpose. A Service SID can be:

- o MPLS label for SR-MPLS.
- o uSID (micro SID) for SRv6 representing network function associated with an ELAN service instance.

MAC address is learned in data-plane. Source node of a MAC address is identified by its node SID (assigned for regular SR operation) during MAC learning phase. In the data packets, the node SID of the source is inserted directly below the service SID so that a destination node can uniquely identify the source of the packets in an SR domain.

ELAN service instances are advertised such that a service message packs as many ELAN instances hosted by the advertising PE node as possible at the time of advertisement. A possible approach is to use a bit-map in which each bit position represents an ELAN instance, as well as the starting value of Service SID. Using these parameters, an ingress PE receiving advertisements node can learn ELAN instance(s) hosted by an egress PE node.

All-Active multihoming redundancy is supported at the underlay level by making use of SR anycast SID. No overlay mechanism is required for this purpose.

Each node is also associated with another SID unique within the broadcast domain that is used to identify incoming Broadcast Unknown-unicast, and Multicast (BUM) traffic. We call such SID BUM SID. If node A wants to send BUM traffic to node B, it needs to use BUM SID assigned to node B as a destination SID. BUM SIDs can also be advertised via BGP for auto-discovery purpose. In order to send BUM traffic within a broadcast domain, P2MP SR policies can be used. Such policies may or may not be shared by ELAN instances.

The proposed solution can also be applicable to the EVPN control plane without compromising its benefits such as All-Active multihoming on access, multipathing in the core, auto-provisioning and auto-discovery, etc. With this approach, the need for advertising EVPN route types 1 through 4 as well Split-Horizon (SH) label is eliminated.

In the following sections, we will describe the functionalities of the proposed approach in detail.

2. Terminology

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

3. Abbreviations

BUM: Broadcast, unicast and multicast.

CE: Customer Edge node e.g., host or router or switch.

ELAN: Ethernet LAN.

EVPN: Ethernet VPN.

MAC: Media Access Control.

MAC-VRF: A Virtual Routing and Forwarding table for Media Access Control (MAC) addresses on a PE.

MH: Multi-Home.

OAM: Operations, Administration and Maintenance.

PE: Provide Edge Node.

SID: Segment Identifier.

SR: Segment Routing.

VPLS: Virtual Private LAN Service.

4. Control Plane Behavior

4.1. Service discovery

A node can discover ELAN service instances as well as the associated service SIDs hosted on other nodes via configuration or auto-discovery. With the latter, the service SIDs can be advertised using BGP. As mentioned earlier such update message will pack information about as many ELAN instances hosted by the advertising PE node to reduce the amount of update messages exchanged by PE nodes.

Similar to the service SID, an ingress PE node can discover BUM SID associated with an egress PE node via configuration or auto-discovery.

The necessary BGP extensions will be specified in a separate document.

4.2. All-Active Service Redundancy

An anycast SID per Ethernet Segment (ES) can be associated with the PE nodes attached to a Multi-Home (MH) CE. The anycast SIDs will be advertised in BGP by the PE nodes. Based on ES anycast SIDs, ingress PEs receiving updates can discover the redundancy membership and perform DF election. Aliasing/Multipathing can be achieved using the same mechanisms exercised by SR underlay for forwarding traffic to destinations belonging to anycast group.

4.3. Mass service withdrawal

Node failure can be detected due via IGP convergence. For faster detection of node failure, mechanism like BFD can be deployed. The proposed approach does not require additional MAC withdrawal mechanism.

On PE-CE link failure, the corresponding PE node withdraws the route to the corresponding ES in BGP in order to stop receiving traffic to that ES. With MH case with anycast SID, upon detecting a failure on PE-CE link, a PE node may forward incoming traffic to the impacted ES(s) to other PE node(s) that is/are part of the anycast group until it withdraws routes to the impacted ES(s) for faster convergence. For example, in Figure 1, assuming PE5 and PE6 are part of an anycast group, upon link failure between PE5 and CE5, PE5 can forward the received packets from the core to PE6 until it withdraws the anycast SID associated with the ES(s).

4.4. E-Tree Support

To be covered in the next revision of this document.

5. Data Plane Behavior

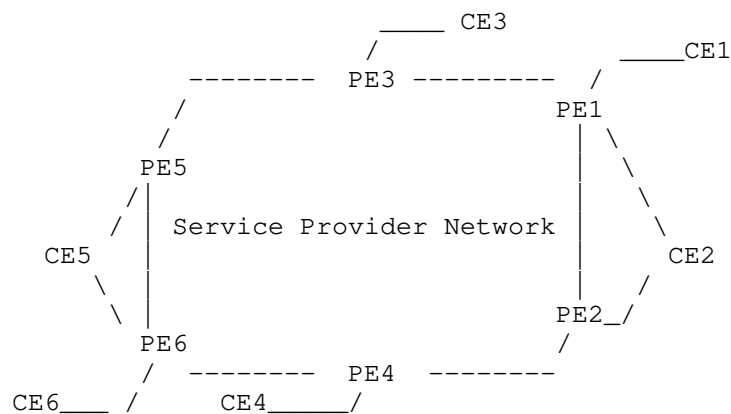


Figure 1: Reference network diagram used for examples below

5.1. Unicast Traffic

The proposed method requires unicast data packet be formed as shown in Figure 2.

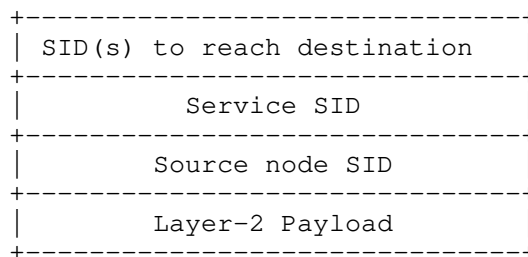


Figure 2: Data packet format for unicast traffic

- o SID(s) to reach destination: depends on the intent of the underlay transport:
 - * IGP shortest path: node SID of the destination. The destination can belong to an anycast group.
 - * IGP path with intent: Flex-Algo SID if the destination can be reached using the Flex-Algo SID for a specific intent (e.g., low latency). The destination can belong to an anycast group.
 - * SR policy (to support fine intent): a SID-list for the SR policy that can be used to reach the destination.

- o Service SID: The SID that uniquely identifies an ELAN instance in a broadcast domain.
- o Source node SID: The SID that uniquely identifies the source node. This can be a node SID which may be part of an anycast group. Note that such a SID is allocated as part of SR underlay operation, and the proposed approach does not impose any additional requirement.

5.2. BUM Traffic

In order to identify incoming BUM traffic a unique SID (which will be referred to as "BUM SID" in the rest of the document) per PE node is allocated. A BUM packet is formatted as shown in Figure 3:

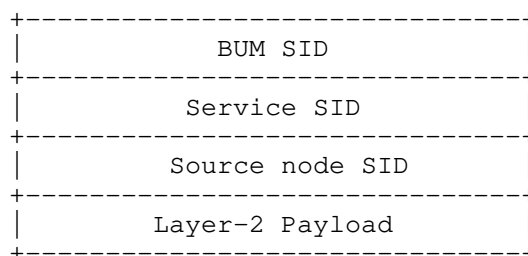


Figure 3: Data packet format for BUM traffic

In order to send BUM traffic, a P2MP SR policy may be established from a given node to rest of the nodes associated with an ELAN instance. If a dedicated P2MP SR policy is used per ELAN instance, a single SID may be used as both replication SID for the P2MP SR policy as well as to identify ELAN instance. With this approach, the number of SIDs imposed on data packet will be only two. It is possible to use a given P2MP SR policy for multiple ELAN instances in which case service SID needs to be inserted in the packet for egress PE to identify the ELAN instance for the BUM traffic.

5.3. Data Plane MAC learning

With the proposed approach, MAC address can be learned in data-plane using the packets formatted as shown in Figure 4.

Source MAC address on the received Layer 2 packet is learned against the source node SID placed directly under the service SID in the data-plane.

5.3.1. Single Home CE

In Figure 1, node 3 learns a MAC address from CE3 and floods it to all nodes configured with the same service SID. Nodes 1, 2, 4, 5 and 6 learn the MAC address as reachable via the source node SID of Node 3.

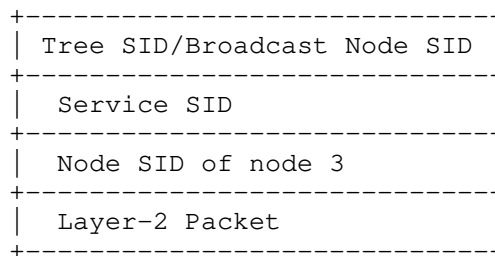


Figure 4: Packet format used for flooding

5.3.2. Multi-Home CE

Referring to Figure 1, let's assume that node 5 learns a MAC address from MH CE5, and floods it to all nodes in data-plane as per SID stack shown in Figure 5, including node 6. The receiving nodes learn the MAC address as reachable via the anycast SID belonging to node 5 and node 6. Node 6 applies SH and hence does not send the packet back to CE5, but treats the MAC address as reachable via CE5, as well floods the address to CE6.

The following diagram shows SID label stack for a Broadcast and Multicast MAC frame sent by Multi-Home PE. Note the presence of source SID after the service SID. This combination/order is necessary for the receiver to learn source MAC address (from L2 packet) associated with ingress PE (i.e. source node SID).

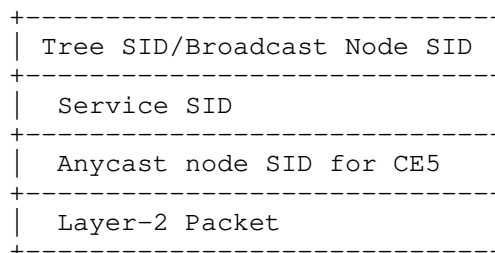


Figure 5: Data packet format for traffic sent by a MH PE

5.4. ARP suppression

Gleaning ARP packet requests and replies will be used to learn IP/MAC binding for ARP suppression. ARP replies are unicast, however flooding ARP replies can allow all nodes to learn the MAC/IP bindings for the destinations too.

5.5. Distributed Anycast Gateway

Distributed Anycast Gateway (GW) (aka inter-subnet IRB function) can be realized as follows:

- o All PEs connected to the tenant subnets share the same GW IP/MAC per subnet.
- o A PE MUST never learn its own GW IP/MAC via the tunnels connecting itself to other PE(s).
- o ARP requests/replies from the tenant subnet are flooded via the ingress PE(s) attached to the subnet to all egress PE(s) attached to the subnet so that egress PE(s) can learn the source MAC/IP address via the ingress PE(s).
- o ARP replies from tenants will be delivered to the local PE hosts the GW virtual MAC address. The local PE MUST flood the ARP replies over the tunnel to other PEs. Other PEs, including the PE which originated the ARP request, will learn the IP/MAC association of the tenant from the received ARP reply.

5.6. Multi-pathing

Packets destined to a MH CE is distributed to the PE nodes attached to the CE for load-balancing purpose. This is achieved implicitly due to the use of anycast SIDs for both ES as well as PE attached to

the ES. In our example, traffic destined to CE5 is distributed via PE5 and PE6.

5.7. E-Tree Support

To be covered in the next revision of this document.

6. Benefits of ELAN over SR

The proposed approach eliminates the need for establishing and maintaining PWs as with legacy VPLS technology. This yields significant reduction in control plane overhead. Also, due to MAC learning in data-plane (conversational MAC learning), the proposed approach provides the benefits as such fast convergence, fast MAC movement, etc. Finally, using anycast SID, the proposed approach provides All-Active multihoming as well as multipathing and ARP suppression.

7. Security Considerations

The mechanisms in this document use Segment Routing control plane as defined in Security considerations described in Segment Routing control plane are equally applicable.

8. IANA Considerations

TBD.

9. Acknowledgements

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>>.
- [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>>.

- [RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <<https://www.rfc-editor.org/info/rfc8660>>.
- [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>>.

10.2. Informative References

- [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-14 (work in progress), October 2021.
- [I-D.voyer-pim-sr-p2mp-policy] Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-voyer-pim-sr-p2mp-policy-02 (work in progress), July 2020.
- [RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <<https://www.rfc-editor.org/info/rfc4761>>.
- [RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <<https://www.rfc-editor.org/info/rfc4762>>.

Authors' Addresses

Sami Boutros (editor)
Ciena Corporation
USA

Email: sboutros@ciena.com

Siva Sivabalan (editor)
Ciena Corporation
Canada

Email: ssivabal@ciena.com

Himanshu Shah
Ciena Corporation
USA

Email: hshah@ciena.com

James Uttaro
ATT
USA

Email: jul738@att.com

Daniel Voyer
Bell Canada
Canada

Email: daniel.voyer@bell.ca

Bin Wen
Comcast
USA

Email: bin_wen@cable.comcast.com

Luay Jalil
Verizon
USA

Email: luay.jalil@verizon.com

INTAREA
Internet-Draft
Intended status: Informational
Expires: 28 April 2022

T. Eckert
Futurewei Technologies USA
N. Shenoy
Rochester Institute of Technology
25 October 2021

Functional Addressing (FA) for internets with Independent Network
Address Spaces (IINAS)
draft-eckert-intarea-functional-addr-internets-01

Abstract

Recent work has raised interest in exploring network layer addressing that is more flexible than fixed-length addressing as used in IPv4 (32 bit) and IPv6 (128 bit).

The reasons for the interest include both support for multiple and potentially novel address semantics, but also optimizations of addressing for existing semantics such as unicast tailored not for the global Internet but to better support private networks / limited domains.

This memo explores in the view of the author yet little explored reasons for more flexible addresses namely the problems and opportunities for Internetworking with Independent Network Address Spaces (IINAS).

To better enable such internetworks, this memo proposes a framework for a Functional Addressing model. This model also intends to support several other addressing goals including programmability and multiple semantics.

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 28 April 2022.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
1.1. Overview	3
1.2. Disclaimer	3
2. Challenges	4
2.1. High level observations	4
2.2. Internetworking limited domain networks with IP addressing	5
2.3. Shorter addresses	9
2.4. Additional semantics	9
2.5. Programmability	10
3. FA-IINAS: Functional Addressing (FA) for Internetworking with Independent Network Address Spaces (IINAS)	10
3.1. Addressing for unicast	11
3.2. Forwarding	12
3.2.1. Dispose Function	12
3.2.2. Steering Function	12
3.2.3. Multiple semantics	12
3.2.4. Internetworking Function	14
3.3. Control Plane	16
3.3.1. Unicast routing	16
3.3.2. Naming	17
3.3.3. Routing	18
3.3.4. Routing policies	19
3.4. Hardware considerations	20
3.4.1. Forwarding plane simplicity	20
3.4.2. Optimizing for smaller networks	21
3.4.3. Maximum address sizes	21
3.5. Example packet header encoding	21
4. Inspirations	22
4.1. E.164	23

4.2. MPLS	25
4.3. Segment Routing SR-MPLS / SRv6	25
4.4. Research	26
5. Summary and conclusions	26
6. Changelog	27
7. Informative References	27
Authors' Addresses	30

1. Introduction

1.1. Overview

Recent work has examined the value of more flexible than fixed-length addressing used in IPv4 (32 bit) and IPv6 (128 bit), see for example [I-D.jia-intarea-scenarios-problems-addressing], and [I-D.jia-flex-ip-address-structure].

The reasons for this interest include both support for multiple and potentially novel address semantics, see for example [I-D.king-irtf-semantic-routing-survey] and [I-D.king-irtf-challenges-in-routing], but also optimizations of addressing for existing semantics, such as unicast, that are tailored not for the global Internet but to better support private networks and limited domains ([RFC8799]).

This memo describes one, in the view of the author yet little explored reason, for more flexible addresses namely the problems and opportunities for Internetworking with Independent Network Address Spaces (IINAS).

To better enable such internetworks, this memo proposes a framework for a Functional Addressing model. This model also intends to support several other addressing model goals including programmability and multiple semantics.

This memo calls the addressing model functional, because addresses are constructed as a structure of
`func1{parameter(s),func2{parameter(s),...i.funcN{parameter(s)}}}`.

1.2. Disclaimer

Any proposals made by this document are explicitly for the purpose of presenting example options of realizing concepts introduced in the memo. There is no intent for any proposals in this document to directly become anything more than just experimental implementations for proof of concept purposes. Equally so or even more so, readers are welcome to pick up any subset of ideas from this memo that they are interested in and reuse it in other designs.

2. Challenges

This section discusses challenges that gave rise to the proposal in this document. It explores in more detail the core challenge not well explored elsewhere and already detailed elsewhere.

2.1. High level observations

There are three core challenges we can observe that limit the ability to build more varied internetworking solutions for non-solely Internet use-cases with especially IPv6:

- * Fixed size address space: IPv4/IPv6 address space is fixed length, not allowing to adopt address length to shorter or longer demands. While it is possible to add more addressing via extension headers, there is no option to not send, or shorten the IPv4/IPv6 base header addresses, when they are not required. While the reasons for fixed size addressing in IPv4/IPv6 can be understood for the feasible high-speed, low-cost forwarders of the 1900th, when IPv6 was conceived, these reasons are today (in the opinion of the author) as obsolete as ATM cells where by the end of the 1990th when both hardware forwarding and mathematical models allowed to provide all ATM type QoS with variable sized packets.
- * The Internet as the primary, if not only use-case driving the design: The address space semantics provided especially by IPv6 is very much focused on the one use-case that drove the development of IPv6: The Internet. While it was and will continue to be the core and sufficient reason for maintaining IPv6, it is not sufficient in the opinion of the author for the much broader use of IPv6. As of today, a likely overwhelming number of hosts using TCP/IP(v6) protocol stacks are not "on the Internet" and the majority likely is not even "connected to the Internet", but instead, they are part of limited domains. This even includes many routers in large service providers that are used to service Internet traffic. Routers in these networks are only in networks that may be called an "underlay" limited domain networks using MPLS, SR-MPLS or SRv6 and Internet traffic is tunneled across them. When the network design is secure, those routers are neither "on" the internet nor "connect to" the Internet.
- * Transparent end-to-end addressing at the core of the IP/IPv6 protocol design, but an ever more diverse reality breaking that design for good reasons: The current core principle of IPv4 and IPv6 is that forwarders have to be passing network layer (IPv4/IPv6) addresses transparently and are not allowed to touch/modifying them. This is the core behavior to support primarily the Internet use case. Yet, the IPv4 Internet today would not

work without NAT, and arguably, the same may also happen to the IPv6 Internet, especially when networks attaching to inexpensive Internet offerings want to avoid complex src/dst forwarding for IPv6 multihoming, and/or avoid renumbering upon change of provider addresses. Even more so, interconnecting IPv4 and IPv6 networks has resulted in no fewer than 24 IPv4/IPv6 NAT solutions (see https://en.wikipedia.org/wiki/IPv6_transition_mechanism), giving rise to the question if and how on-path processing of addressing can be proactively become part of future addressing designs to support more flexible internetworking - translating the best of past NAT experience into better future designs. This is a core option of what FA-IINAS can do.

2.2. Internetworking limited domain networks with IP addressing

One of the core challenges of the existing IP(v4) and IPv6 addressing model are the addressing they provide for private networks with or without connectivity to the Internet, which are also called limited domain networks [RFC8799].

One reference example is that of networking inside a particular product/solution/installation, and then compositing this product with other products, probably even multiple times, hierarchical, as show in picture Figure 1. These type of designs are traditional in industrial networks. Similar issues and solutions can be found in networks with multiple layers of NAT such as Home Networks that are dorm rooms connected via NAT to a dorm network, connected via another NAT to a campus network, connected via yet another NAT to maybe finally, the Internet. Similarly designs can happen with more complex topologies in federated private networks.

In pre-IP industrial networks, individual products were hiding their interior elements by some (combination) of elements that controlled the interior behavior completely and provided only an abstracted view of the machinery to the outside.

With the introduction of IP networking into these type of solutions, the ability for gateways to become IP routers and providing connectivity into the machinery throughout the larger internetwork opened up many important improvements, but of course also challenges, especially for security.

Benefits of network layer internetwork connectivity includes options such as control loops that can more easily be built across multiple components/levels of the hierarchy and controllers that can be pulled out of machinery and positioned elsewhere in the network, enabling virtualization and resource multiplexing. Multiple independently running control systems can be implemented in parallel, including

solutions like device vendor preventive maintenance telemetry, operator managed firmware update or third-party orchestrated security audits or intrusion detection/prevention, just to name a few.

With IP connectivity, all this can be built without the need of understanding how to get through various layers of fixed-functionality higher-than-network layer gateways that can not be extended by third parties. Instead, new designs are based on end-to-end IP connectivity - plus appropriate set of security measures at gateway routers, of course an appropriate set of security/filtering measures, for example MUD, [RFC8520].

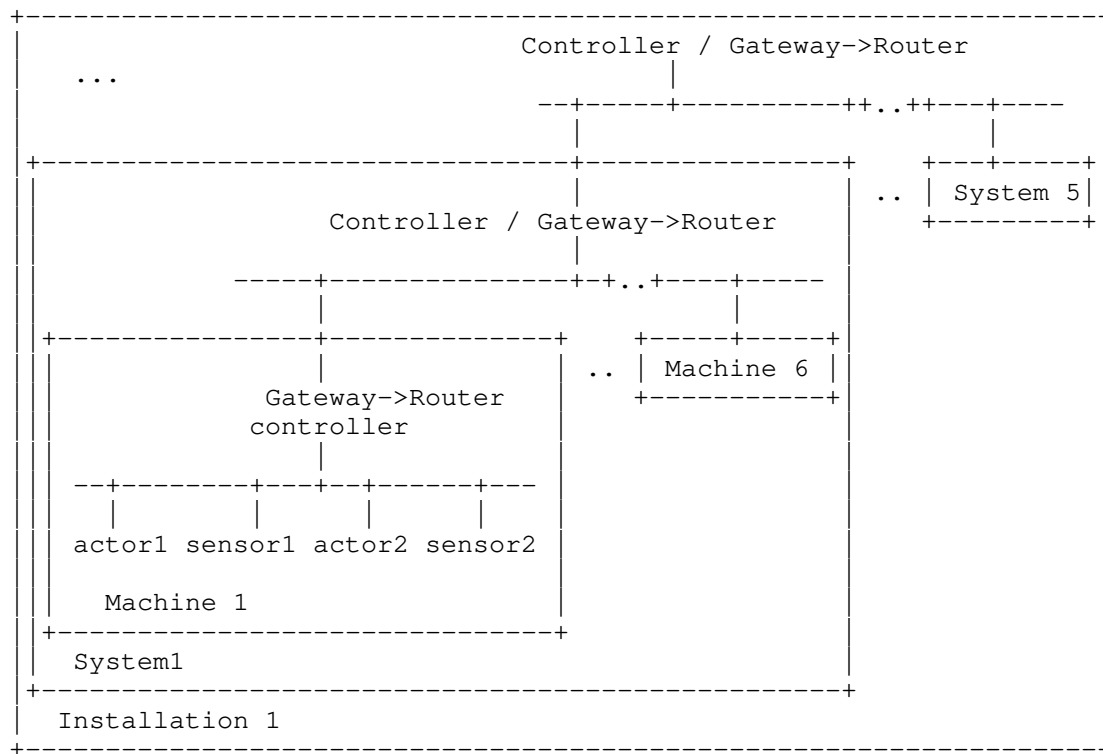


Figure 1: Example hierarchical composed internetwork

In the opinion of the author, the most easily adopted addressing architecture in these type of solutions today is also the one widely used: IPv4 with [RFC1918] addresses. These addresses are actually owned permanently for each deployment case - as long as the scope of addressing is well defined.

In result, a common scheme of addressing in machinery such as the one shown in Figure 1 is to reuse the same 10.0.0.0/8 or 192.168.0.0/16 addresses for every instance of a product/machinery manufactured. In the example, actor1 could use 10.0.0.1, sensor1 10.0.0.2 and so on. But equally, if Machine 3 was the same or similar, its internal components would share the same machinery. And when hundreds of these products are produced, they would all have the same addresses.

To allow deployment and composing those type of machineries, the router/switch connecting to the outside/next-level in a hierarchy will need simple NATing function for example statically mapping the 10.0.0.x on the inside to 10.0.1.x on the outside for Machine 1, where the same router/switch for Machine 3 would be configured to NAT from 10.0.0.x to 10.0.3.x. And likewise at the next layer of hierarchy, 10.0.y.x could be mapped to 10.z.y.x with a different y for every instance.

In support of solutions like this, many if not most industrial ethernet switches deployable as machinery gateways do therefore support this type of static NAT mappings. Likewise, common practices in industries rely on this addressing with composition via NAT approaches, including machineries as large as production lines or in transportation networks train cars and all their included machineries/equipment.

The desire to avoid NAT in IPv6 and availability of sufficient addressing space lead to replacing the concept of [RFC1918] in IPv4 with the concept of Unique Local Addresses (ULA) in IPv6, standardized in [RFC4193]. Instead of the few scoped prefixes of [RFC1918], ULA provide for 2^{40} different prefixes, and the design guidelines are theoretically simple: pick a random prefix and then you can interconnect your networks later on with a very low probability of address prefix collision/reuse.

Unfortunately, low probabilities of address collision is not a good design principle for most of these type of environments because there is really no good operational solution what do if such collision occurs, and rare errors are also very hard to build resilient solutions for. Also the probabilities begin to become much higher when not looking at a connection of just two or few of such ULA networks, but when there can be thousands of such networks, such as in the transportation networks use case.

In result, ULA is not very persuasive for many such deployments, especially when the alternative with IPv4 is address prefix mapping as required for NAT, when NAT an an almost free provisioning side effect of setting up the required connectivity via permit lists via network/transport filters. The need to automate such in-network filtering to secure such deployments can also be seen in the advent of MUD, [RFC8520].

If one considers that most of these subnet networks will have fewer than 253 hosts connected to it, then the IPv6 ULA solution does also not provide for any more bits for subnets than the 16 bits of z.y in the above example using IPv4 10.z.y.x with x being the host part: The lower 64 bits of the IPv6 address is hard to use for anything than the host parts with non-router hosts. The whole ULA prefix is 48 bits, leaving just 16 bit (128 - 64 - 48). Add to that the non insignificant IPv6 packet header overhead plus fewer availability of NAT in IPv6 products because it is assumed to be less required, plus the insufficiency of "low likelihood of collisions" when attempting to utilize only ULA.

Vendors of equipment that have assigned Provider Independent IPv6 address space could of course allocate addressing from that space for equipment they manufacture or integrate, whether it is globally unique or "generic", e.g.: reused across every instance of a product and hence requiring NAT. Unfortunately, and unlike ethernet, where one actually does own addresses after buying an OUI, assigned IPv6 addressing is not permanent, and even though revocation of address allocation is not standard practice, standardized solutions for global IPv6 address space (like IPv4 global address space) really need to allow the ability for those addresses to be returnable instead of being handed off in products to customers.

Even though in hindsight, the hierarchical address allocation from the available 16 bits in 10.x.y.z for two layers of interconnections in the above example looks obvious and simple, in many cases the creation of multiple hierarchies is only an afterthought and the fixed address length and prior suboptimal assignment of addressing in a deployment will cause the need for a lot of re-addressing. This is a recurring problem in larger enterprise/commercial networks under unplanned growth or mergers & acquisitions, especially of course in IPv4. Likewise, once the 16 available bits in the above described NAT approach are used up, whether it is IPv4 or IPv6 with ULA, no further extensions of the design are possible.

2.3. Shorter addresses

As has been noted in prior memos, shorter addresses than IPv6 128 bit are highly desirable in private networks / limited domains whenever it is clear that the total required addressing space is much smaller and connectivity to e.g.: the Internet is not required. Evidence of such requirements can be found for example in header compression for IoT networks such as [RFC6282]. Such compression introduces yet another layer of complexity - the whole ecosystem of devices and diagnostic options has to support it to be equally acceptable as uncompressed packets.

2.4. Additional semantics

New semantics can only be introduced into existing IPv4/IPv6 when their required address size fits nicely into the 32 or 128 bit address space.

This section does not aim to be complete, see [I-D.king-irtf-semantic-routing-survey] for a broader survey. Instead it will provide additional levels of details for the benefits of fittingly sized addresses for few examples, that the author is familiar with.

When ignoring Anycast, IP Multicast is likely the most widely adopted additional semantic added to IPv4. With IPv6, IP Multicast became even more flexible and easy to deploy, because the additional bits of IPv6 addresses allowed to encode additional IP multicast parameters through additional fields in IPv6 addresses: Scope address field [RFC4291], SSM addresses [RFC4607], Unicast prefix multicast addresses [RFC3306] and embedded-RP [RFC3956]. Nevertheless, especially embedded-RP could have benefitted from even longer addresses because with the 128 bits available the solution had to take a hit in the complexity of deployment. It requires to engineer that RP address such that its non-0 host port is very short (4 bits).

In contrast, Bit Indexed Explicit Replication (BIER) which started in the IETF in 2014 and resulted in the architecture [RFC8279], did not choose the option to integrate into IP/IPv6 because it desired addresses sizes of at least per-network configurable from 64 to 4096 bit plus additional qualifiers of at least 16 bits (so-called SD, SI address qualifiers). This made it necessary for BIER to (re-)invent its own network layer packet header, [RFC8296] which duplicates pretty much all packet header fields of MPLS plus IP packets plus additional BIER header fields, so that it can be used in both MPLS and non-MPLS networks.

Similar arguments about the limited size of IPv6 address could likely be made for ICN/CCN networks because the semantic of their addresses is that of data items such as time slices of specific spatial and temporal resolutions of some media such as an audio/video recording - and those name spaces would ideally have addresses as long as URLs.

2.5. Programmability

Segment Routing via IPv6 (SRv6) introduced with [RFC8986] and [RFC8754] (SRH) and architecture in which source routing with an IPv6 extension header is combined with encoding of additional processing semantics into the destination and source routing hops IPv6 addresses. SRv6 calls this programmability.

SRv6 is a very flexible and theoretically extensible concept but challenged by the fixed address length design of IPv6. For most steering hop addresses, the bits reserved for this additional packet processing are not required, but when they are required there may even be too few bits available. Variable length addresses allowing for variable long programming field in the address would in the opinion of the author be highly beneficial.

One evidence for the programmability bits seen as wasteful in many cases is a variety of currently proposed drafts to provide more compressed source routing options for SRv6 (as of mid 2021).

3. FA-IINAS: Functional Addressing (FA) for Internetworking with Independent Network Address Spaces (IINAS)

This section outlines an addressing design that attempts to solve the above described challenges and calls it tentatively FA-IINAS. Functional Addressing refers to the design aspect that addresses in this design can be interpreted as functions with parameters.

Notwithstanding other granularities or options, this document assumes that addresses are textually represented in hexadecimal and that the minimum structure element of an address is 4 bit so that the different structural elements of an address can simply be shown as concatenation of hex digits. The "." character is inserted optionally to show where in an address one semantic part ends and another starts.

Like in IPv6 IoT networks, such as those using RPL ([RFC6550]) as their routing protocol, this memo starts by assuming all nodes are routers and that addresses are predominantly node addresses as opposed to IP/IPv6, which defines unicast addresses to be interface addresses. This is but an academic differentiation, because node addresses can also be represented as interface addresses of so-called "loopback" interfaces.

A network in this design is an independent address space, not shared with other networks. A network has theoretically unlimited long addresses whose prefixes are mapped onto the nodes of the network, which are expected to form a graph of transitively connected nodes. Practical limits to address length are subject to acceptable packetization.

3.1. Addressing for unicast

Each node is assigned one or more node prefixes from the networks address space and none of these node prefixes can be overlapping. In other words, no assigned nodeprefix can be a prefix of another assigned nodeprefix. This rule ensures that every node "owns" any address equal or longer to its assigned nodeprefix. Allocation of node prefixes is currently out of scope for this memo but could rely on any well-known methods including manual operator assigned, SDN controll managed, or as initially described in this document assigned by manufacturer/vendor.

Routing in a network is assumed to enable forwarding across the graph of the network to the node owning the nodeprefix of the address.

Given variable long addresses, the first observation of this addressing scheme is that it allows to combine short addresses with extensibility.

In a simple example the first 200 nodes are assigned addresses 01 ... c8, at which point in time the network operator gets worried about growth exceeding the 256 mark and starts to assign longer addresses: c90 ... f000, at which point in time ever increasing success might cause assignment of even longer prefixes.

Addresses longer than the assigned "nodeprefix" are used to instantiate a specific function on the node itself. A generic representation of an address could be
nodeprefix.function.{parameter}.

3.2. Forwarding

3.2.1. Dispose Function

When using a single digit function field, function = 0 could for example be "dispose" to decapsulate the packets payload and deliver it to the host stack. Parameter could for example be the next-protocol value, eliminating the need to have a separate packet header field for this parameter.

While not being the same crucial issue as for the node prefixes themselves, putting the next-protocol into the address makes it extensible too, so one would not run out of a 256 space as IPv4/IPv6 might do at some point.

3.2.2. Steering Function

Command = 1 could be a "steer" command and the parameter is another address. To act on the command, the node would strip the nodeprefix and command part of the address and forward it based on the address parameter. For example node 73 (e.g.: node with nodeprefix 73) receives a packet with destination address 73.1.55.1.33.0. It forwards the same packet with the stripped destination address 55.1.33.0 to node 55, which likewise forwards the packet with stripped destination address 33.0 to node 33, which ultimately receives it.

3.2.3. Multiple semantics

To introduce additional semantics into a network, such as for example multicasting, we need to generalize how to interpret the first part of the address, which so far was only interpreted to be a nodeprefix for unicast forwarding.

```
address = prefix{.nodefunction{.nodefunction-parameters}}
prefix = semantic{.semantic-parameters}
```

```
semantic / = unicast-forward
```

```
unicast-forward = <set of prefixes>
unicast-forward-parameters = node-prefix
```

```
semantic /= multicast-forward
multicast-forward = <set of prefixes>
multicast-forward-parameters = multicast-group
```

Figure 2

In other words, the prefix at the start of the address is composed of a semantic and its parameter, and the case discussed so far is simply the unicast-forward semantic followed by a node-prefix parameter.

Again, semantic can be an arbitrarily long or short prefix, but no semantic can be a prefix of another semantic.

In a practical example, this scheme is easily applied to existing IPv4 / IPv6 address spaces. For IPv4:

```
unicast-forward = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D
multicast-forward = E
```

Figure 3

In other words, because IP multicast uses addresses 224.0.0.0/4, its non-overlapping semantic prefix is E, and IPv4 unicast addresses use the non-overlapping prefixes 0...D. Assume further that a node in the network had assigned prefix 10.0.0.0/24, then this would translate in our scheme into:

0.A0000.XX

Figure 4

When a node processes this address, the 4-bit prefix 0 indicates that the following prefix has to be looked up in unicast forwarding. This prefix is A0000. Once the packet is delivered to the node, the remaining 8 bit XX can accordingly be interpreted by the node as a nodefunction with parameters.

Likewise, an address 239.1.2.3 would translate into E.F010203, so the first 4-bit E value would indicate that multicast forwarding needs to be applied to the rest of the address, and with IP Multicast forwarding not having further structure (ignoring willfully for simplicity of the example that it does, for example with SSM), all the remainder of the IPv4 address is the multicast-group

In summary, the logic does really only generalize what routers today already do when they do prefix lookups, except for the following core differences:

- * In IPv4/IPv6, the address semantic is hard-coded by IETF standards. In FA-IINAS they are definable by every network.
- * In IPv4/IPv6, there is no notion of nodefunction{.nodefunction-parameters}, only SRv6 has this concept.

In actual IPv4/IPv6 hardware forwarding lookups, one would not do one lookup for the semantic, followed by another lookup for the semantic-parameters for the case of unicast-forward, instead this would be flattened. The same type of flattening would of course be useable in FA-IINAS. Whether or how flattening or other optimizations are feasible for other semantics such as multicast is of course highly semantic and node implementation specific.

3.2.4. Internetworking Function

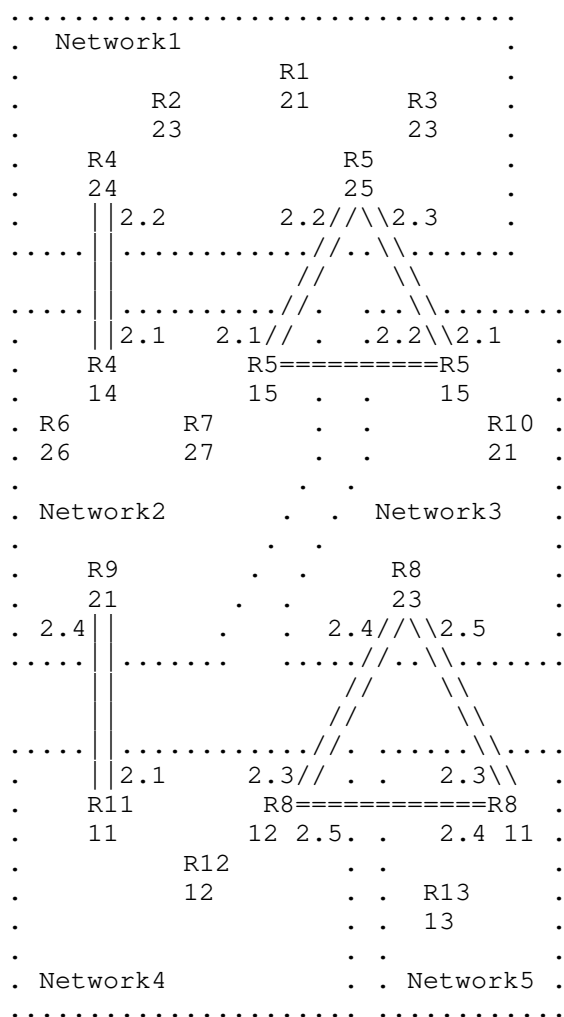


Figure 5: Internetworking example

Figure 5 shows an example internetworking topology of 5 networks, each with its own independent address space. Globally unique Rxx numbers are used to refer to routers.

An edge node is a router that has prefixes from two or more networks into which it connects. In the example, R4 connects into Network1 with prefix 24 and into Network2 with Prefix 14. Likewise, R8 connects into Network3 with prefix 23, into Network4 with prefix 12 and into Network5 with prefix 11. An edge node can be a router simply with different interfaces into different networks, or it can be decomposed into multiple devices, each in a separate network. In this section we describe behavior as if it was a single device.

For an edge node to pass a network into a separate network, the internetworking function on the node has to be called. In the example, this function is codepoint 2 on all edge nodes, and the first parameter is an identifier of local relevance for the network into which to pass the packet. In actual deployment, this function number can of course be locally significant to the Network and/or even each edge router, assuming appropriate control plane to assign the number to this function.

Assume R12 (12) in Network4 wants to send a packet to R1 (21) in Network1. To send it R12->R8->R5->R1, R12 would have to use a destination address of 12.2.3.15.2.1.21.0, or numerically without separators 0x12231521210.

12 will route the packet in Network4 towards R8 because of the destination address 12/8 prefix. .2 indicates to R8 that it should invoke the interworking function and pass the packet into Network 3. As part of the interworking function, R8 then strips all the address prefix it has processed so far from the destination address, leaving 15.2.1.21.0. R8 then forwards the packet with this destination address into Network 3, where it will be received by R5, which again invokes the interworking function due to .2, forwarding the packet into Network1, stripping 15.2.1.0 from the destination address and forwarding the packet with destination address 21.0 into Network1, where it will finally be received by R1 which passes the packet to its host stack because of dispose function 0.

To (optionally) allow for a return path, each edge node could equally but inversely process the source address: When R12 sends the packet, it would indicate a source address of 12.0. When R8 passes the packet via its interworking function into Network3, it would prepend its return path interworking function address, making the source address 23.2.4.12.0, where 23 is R8 address prefix in Network3 and 2.4 interworking function to return the packet into Network4. Likewise, when R5 processes the packet by its interworking function,

it would prepend its return path address element to the source address, before sending the packet into Network1, making the source address 25.2.3.23.2.4.12.0. This is then the address to which R1 could send return packets, and likewise, on its way towards R1, the address, for example when travelling via Network3 always has a returnable source address.

With this behavior of the interworking function, it is obvious, that address management of networks would want to keep a sufficiently large number of very short prefixes, such as those in this example or even shorter to address the interworking function in a sufficiently larger number of edge routers so that a complete internetwork path address will not become too long to exceed the maximum address lengths.

3.3. Control Plane

This section reviews a range of control plane considerations necessary to build a working solution out of the functional addressing. In short, what is required for functions to be flexibly configurable and extensible in the network, it requires a control plane that in its principles is very much based on what was learned in MPLS.

3.3.1. Unicast routing

FA-IINAS expects a control plane that supports routing for unicast-forward parameters (address prefixes) in the same way as it is done today for IPv4/IPv6. Except that it would be for address prefixes (multicast-forward-parameter) of different length and not limited to just 32/128 bits as in IPv4/IPv6.

In addition, FA-IINAS needs control-plane functions that allow defining the semantics and their prefixes, like the above example of 0...D for IPv4 style unicast-forwarding semantic and D for IPv4 style multicast-forwarding semantic.

One of the core challenges for this control plane function is that inconsistency between nodes can have significant different negative impacts than the today accepted "eventual consistency" in IPv4/IPv6 unicast routing that is achieved by the most widely deployed unicast forwarding control planes: distributed routing protocols (IGP/BGP).

The degree of concerns will highly depend on the actual new issues that could happen in the face of inconsistencies, and this can only be vetted with a given set of semantics.

In a most simple example, semantics may simple be configurable via a management plane, and such an approach can be pre-staged, pre-configured, validated network devices, such as in industrial or embedded environments.

In the case of a most flexible, agile type of network, control plane mechanisms would have to be extended to support strong consistency models, for example through node-to-node security associations coupled with a strong consistency network-wide-core-config mechanism. Such mechanisms could in the opinion of the author easily be built on the framework provided by [RFC8994] which provides these hop-by-hop security associations and inband control plane infrastructure, coupled with [RFC8990] as the protocol to negotiate the configuration with strong consistency.

3.3.2. Naming

3.3.2.1. Intra network naming

In FA-IINAS, nodes are acting as routers, and the addresses described are assigned to them persistently. This eliminates in many cases, especially when the network is primarily for m2m communications the need for DNS names, because effectively the address of a node is its persistent name.

In networks small enough, e.g.: maybe $\leq 20,000$ nodes, the very same argument can also apply to nodes that are hosts, e.g. without the need to support full routing/FA-IINAS operations, but still having a persistent address assigned that is routed in the networks routing protocol.

If indeed there is a need to use DNS or other naming schemes, then this is no different than applying naming with DNS to today's [RFC1918] addresses.

3.3.2.2. Simple inter network naming

The need to support (DNS) names is equally lower in interconnected FA-IINAS networks assuming the intra network naming arguments outlined before apply to the interconnected networks.

Because an address in a different FA-IINAS network is dependent on the path from/to its corresponding peer, it is of course not sufficient to simply have a global internetwork name to address mapping.

One of the likely oldest solutions is to align name resolution with packet forwarding so that the very same edge nodes between two networks that do translate addresses can accordingly also translate their name resolution. This was productized and fairly widely deployed as early as the late 1990th for IPv4 with rfc1918 addresses, see for example [CiscoNAT].

This type of solutions relies on well-known routing policies such as simple hierarchical routing though and are not generic for arbitrary topologies.

3.3.3. Routing

3.3.3.1. With internetwork topology knowledge

When FA-IINAS networks are connected in an arbitrary topology instead of a simple hierarchy, the fundamental problem is that of constructing the address of a target peer as a path through a set of appropriate network edge nodes in the address, followed by the nodes address within its network.

In many interconnected FA-IINAS networks, one can assume to have systems that can do this, such as in an industrial setting where a global view of the topology of networks exists and a PCE/SDN-controller will choose the path and can accordingly calculate also the addresses from the path.

3.3.3.2. With internetwork naming knowledge

A decentralized solution can be built by relying on a combination of naming and internetwork routing.

Every network (name space) is assigned a globally unique identifier. This identifier is only used in the control-plane, so it should be reasonably easy to have a set of construction mechanisms allowing everyone to easily create its own namespace, such as for example from some owned location (street address) and/or other owned names/identifier.

When a global naming system like DNS then exists, an FA-IINAS address is the combination of FA-IINAS network identifier and address within that network.

Across the interconnected FA-IINAS networks, the edge-routers would operate extended versions of a protocol like BGP through which any party can calculate desired paths. The extensions would include the FA-IINAS network identifiers and address prefix mapping rules of the edge-nodes, thereby allowing to also calculate addresses from FA-IINAS network identifiers and address.

When large number of small networks (such as users homes) connect to larger networks (such as an ISP), those ISP would be concerned of having to propagate millions of small FA-IINAS network mappings into BGP. This is not done today with IPv4/IPv6, and it would not scale any better with FA-IINAS. Instead, the fact that the home network would be reachable with one or more ISP could be done by also creating naming system mappings from the home networks identifier to the identifier and address prefix mappings of the ISP to which the home network is connected.

When a peer looks up a name and retrieves an FA-IINAS address but cannot find the FA-IINAS network identifier in its internetwork routing information, it can instead resolve it to the "next higher up" ISP FA-IINAS network-identifier/prefix - and recurse this until it has routing information.

Likewise, when a peer does not have any routing information (because it does not participate in internet routing information), it has to forward the appropriate resolution request hierarchically upward.

In summary, it would be architecturally "easy" to extend DNS and BGP with the necessary extensions to resolve names to FA-IINAS addresses and construct relative FA-IINAS addresses from this information.

3.3.4. Routing policies

Note that this "easy" part does not include the possible desire to be more or less flexible in path selection. Whereas today, packets, once they enter "the Internet" are not under steering control of the sender but under "hop by hop hot-potato steering" control of the ISP, with FA-IINAS this may be different - or the same. If a sender then constructed an FA-IINAS address implying an internetwork path that was not desirable for this traffic by the indicated transit networks, this would cause an error. Therefore, the above outlined procedures hinted at relying on the internetwork routing information whenever available and only resort to using naming system to fill in the additional (edge) information.

Today it is becoming more common to use alternative than "native Internet" paths by steering traffic across virtual/container routers in cloud DC, many of which have ample and underutilized international

connectivity. However, additional charges for compute and forwarding will apply. These type of high-overhead solutions could be replaced by FA-IINAS to steer traffic across such additional networks and without the need to instantiate VM/containers. It would require appropriate and lightweight identity and accounting forwarding plane packet header information so that those additional charges could be applied.

3.4. Hardware considerations

3.4.1. Forwarding plane simplicity

Forwarding of FA-IINAS packets based on destination address is the same type of prefix lookup on the destination address as it is today in IPv4/IPv6, except that the maximum lookup prefix can be shorter or longer, this is detailed in the next section.

The steering function should have a lookup complexity whose complexity is in the order of SR-MPLS or even simpler. It can constitute of a prefix lookup in the same forwarding table as non-steered forwarding, but the adjacency would then have to strip the looked up prefix from the destination address (comparable to MPLS label pop) and forward the packet again based on the remainder of the destination address - unless additional on-node service functions have to be invoked.

The interworking function is very much like the steering function, but it also prepends a return prefix to the source address field, making it the most expensive forwarding plane operation.

In general, the author assumes that packet processing that strips a prefix from the destination address and optionally adds a prefix to the source address is well feasible in next generation, highest-speed, lowest-cost forwarding engines.

Optimizations beyond this are possible but would break the independent address allocation across networks. For example, if it is possible for an edge node to have the same prefix length across the networks it connects to, and source address follows destination address in the packet encoding, then stripping the destination address could be achieved by shifting the destination address in a contiguous packet buffer, making head for the source address prefix to be prepended to the following source address field.

3.4.2. Optimizing for smaller networks

One of the benefits of FI-IINAS is that it allows to adopt the address space size based on the requirements of networks and therefore also allows to optimize hardware known to be built/sold only into limited size networks, such as many industrial and almost all embedded networks.

For example, low-cost, high-speed hardware forwarding might be possible to design less expensive with just 16 bit lookups instead of for up to 128 bit lookups, as may be required for IPv6. Equipment could be sold with that profile parameters "for networks with up to 2^{16} nodes".

Because of the way FA-IINAS is designed, a limit to 2^{16} nodes does not mean that FA-IINAS addresses are only 16 bits. Instead they can still be "arbitrary" long (where arbitrary is subject to a discussion point further below in this section). Just the length of the unicast-forward part of the address is limited to 16 bits.

3.4.3. Maximum address sizes

The permissible maximum size of source and destination address are primarily subject to the header size that inexpensive hardware forwarding can examine and modify. For future generations, this might likely be as much as 512 bytes, so to optimize hardware lookup it might be interesting to consider the option of carrying the addresses not consecutively, but carry them as

3.5. Example packet header encoding

The following encodings propose a couple of ideas that could be interesting in addressing.

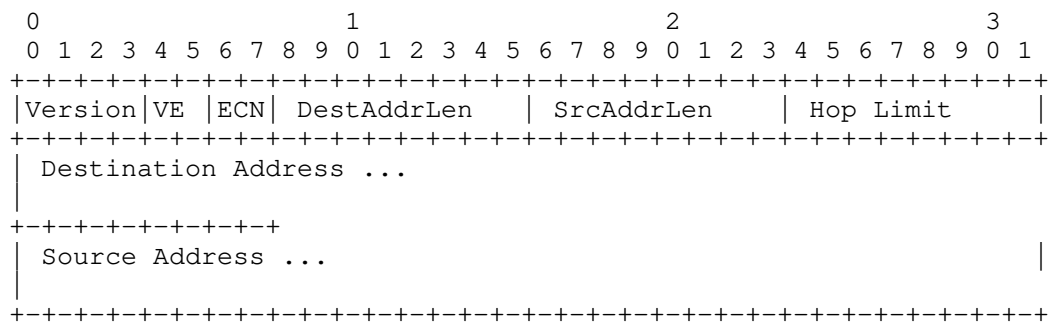


Figure 6: Example packet header encoding

Version: A version number for this packet header from the same registry as the IPv4/IPv6 version number field.

VE: Version Extension. 00. Reserved for future variations of the header, such as new extension header formats if desired, so as to not use up any more than one Version code point.

DestAddrLen: The length of the Destination Address field in bytes. Valid values are 1...255 bytes. One byte minimum length is mandatory because of the need to indicate some semantic for processing the packet.

SrcAddrLen: The length of the Source Address field in bytes. Valid values are 0...255 bytes. The Source Address field is therefore optional.

ECN: See [RFC3168] and the documents updating it.

Rsv: Reserved.

Hop Limit: As in IPv6

Beside the variable length of the Source and Destination address fields and hence their length indications, the difference to the IPv6 header are as follows:

Only the two ECN bits are maintained from the IPv4/IPv6 Traffic Class field. This is because in the majority of networks, the other 6 bits of Traffic Class, DSCP are not being used, and where QoS differentiation would be used, often additional or different QoS parameters may be required that are not supported by IPv4/IPv6. Such a new network header would thus be a great opportunity to improve on QoS header parameters through a better QoS extension header, where it is needed (outside scope of this document), and not proliferate not ubiquitously used elements in the base header. The same reason applies to removing the Flow Label field.

ECN on the other hand is very fundamental for the majority of all traffic in Internet and limited domain networks.

4. Inspirations

This section reviews prior addressing and networking technologies that did inspire this memo and compares it with them.

4.1. E.164

E.164 telephone numbers traditionally worked (and may still work) similar to this mechanisms handling of addresses by adding and removing prefixes and allowing to grow networks hierarchically.

In Germany for example a town/city might have had a subscriber numbering plan starting with 3 digit numbers and expanding over time into 5 digits. 0 was excluded as the first digit of any assigned number. Let our example subscriber have number 1234

When the phone systems of towns/cities where connected, dialing a different town/city would use a concatenation of the inter-city traffic discrimination code "0" followed by the dial code for the town/city, followed by the subscriber number. Let our example town dial code be 4111, the subscriber number dialed from a different city would be 04111 1234. Again, "0" was excluded as the first digit of a trunk prefix.

When finally the phone systems of countries where connected, dialing a different country would use a concatenation of the international traffic discrimination code "00" followed by the country dial code, which in our example is 49 for Germany followed by the dial code for the city, followed by the subscriber number - 0049 4111 1234 for our example subscriber. Note that this number would of course only work when calling from countries that also do use "00" as the international traffic discrimination code. When calling the number from the USA, one would have to dial 011 4111 1234, because the USA uses 011 as the internal traffic discrimination code.

Of course, understanding foreign countries traffic discrimination code rules to reverse engineer a foreign telephone number so as to translate it to the according rules of the calling-from country is one of the problems that is leading more and more subscribers to prefer the absolute E.164 telephone numbers like +49 4111 1234.

On the other hand, when the interplanetary telephone network will "soon" [I-D.draft-farrel-soon] arrive and there are not enough country codes available in Earth's existing numbering plan, one would have to find a way to attach prefixes in front of existing E.164 numbers, something that E.164 likely cannot afford, but which would be possible with UPVLA.

In our example the UPVLA address could be 0003 49 4111 1234 and a new solar system "absolute" address could be ++3 49 4111 1234.

Obviously, Mercury has to get 1, Venus 2 and Earth 3 and so on, so that it would be easier to remember how to dial other planets than it is now to remember how to dial other countries.

If one was to use the solution proposed in this memo to build the phone network addressing system with the example numbering plan, one could set up a multi-tiered internetwork as shown in Figure 7.

Soon:

```

.
. Solar System network .
.
. prefix "3" . |
. .... v strip 3 from dst, prepend 0 to dst
...| Planet Edge Node .... forward into global network
. .... ^ strip 0 from dst, prepend 3 to src
. prefix "0" . | forward into solar system network
.

```

Today:

```

.
. "global" network .
.
. prefix "49" . |
. +-----+ v strip 49 from dst, prepend 0 to dst
...| Country Edge Node |... forward into country network
. +-----+ ^ strip 0 from dst, prepend 49 to src
. prefix "0" . | forward into global network
.
. "country" network .
.
. prefix "4111" . |
. +-----+ v strip 4111 from dst, prepend 0 to dst
...| City Edge Node |... forward into city network
. +-----+ ^ strip 0 from dst, prepend 4111 to src
. prefix "0" . forward into country network
.
. city network .
.
. subscriber 1234 .
.....

```

Figure 7: Example internetwork for E.164 style address structure with FA-INAAS

Packets destined to an address starting with "0" would be routed to an edge node of the city network, which "owns" the "0" prefix, there that "0" prefix is stripped, but the cities own prefix of in the example "4111" is prepended to the source address, and then the packet is forwarded into the country network.

When a packet is received from the country network on a city edge node, the opposite happens, the cities own prefix, e.g.: 4111 is stripped from the destination address and 0 is prepended to the source address, then the packet is forwarded into the city network and routed to the destination. Which can generate return packets by just swapping source and destination addresses.

The same process will happen across 2 network tiers when a 00 prefix is used or even 3 network tiers, once we have (soon ;-) a Solar System Network.

Of course, each tier and each instance of each tier can choose its own addressing scheme and prefixes for the edge routers. It is left as an exercise to the reader for example to amend the example with its own home countries traffic discrimination codes.

4.2. MPLS

Adding/Removing or swapping prefixes is the core forwarding process step in Multiprotocol Label Switching [RFC3031]. Due to the time MPLS was designed, it had to have a very fixed size and functionality stack architecture, but as claimed in before, the author thinks that today an MPLS stack could easily be built just with the proposed addressing scheme address.

Compared to MPLS, the proposed scheme does not mandate that that every steering address needs to contain QoS (EXP) and TTL fields as are present in MPLS Label Stack entries, but of course they would be equally possible as parameters of appropriate functions.

Likewise the proposal does not think it is appropriate to require complicated scanning ahead into the address in search of Special Label Stack entries. Therefore, FA-IINAS would require that any per-hop accessible information that is not included in the hops function/parameters would have to be carried would have to be carried in a separated extensions header.

4.3. Segment Routing SR-MPLS / SRv6

FA-IINAS can support more compact packet steering than SR-MPLS when the prefixes are accordingly chosen to be shorter than the 32 bits for an LSE.

While it would be possible to emulate MPLS LSE by using prefixes of 20 bit and following them with 12 bit of functional parameters indicating EXP and TTL, the proposal in this memo does not assume that transit routers would be able to act on those EXP or TTL bits. While it would be easily possible to define such additional transit hop semantic through extensions to the control plane, the author believes that the per-path parameters of TTL in a base header and more flexible QoS in an extension header is the more likely most useful option for these two functions.

In comparison to SRv6, FA-IINAS allows of course more compact representation of steering hops and also more easily few or many per-hop bits for programmability, as desired.

What FA-IINAS does not provide for is to keep the sequence of steering addresses in the header up to the final receiver. This might be useful for diagnostics, but it is seemingly not so important that it did stop the adoption of SR-MPLS, where the steering hops are likewise removed from the packet header when steering happens.

Other functions than steering and per-steering hop programmability provided by SRv6 via SRH (such as its TLV for the receiver) are unaffected by this proposal and could equally be provided for by an SRH style extension header without the source routing part.

4.4. Research

[Haoyu] proposes a hierarchical addressing scheme and provides reviews in a lot more detail a set of other reasons for such addressing scheme. That paper does not allow for arbitrary composition of networks without a clear hierarchy or root thereof, as FA-IINAS does.

5. Summary and conclusions

This memo introduces a simple but hopefully very attractive addressing scheme that leverages variable length address sizes with the potential for simple address prefix processing (push/pop/swap) on steering hops, service-function hops and especially network edge nodes.

Push/pop/swap of network prefixes on network edge nodes allows to introduce a non-global internetworking address architecture that should make it a lot easier to build and manage many embedded, private or otherwise limited domain internetworks and optimize forwarding engines for a variety of different of these type of networks such as through known maximum network prefix lengths.

When network addresses as in FA-IINAS become effectively internetwork path addresses, they also allow for a much wider range of possible routing policies. In a time where the classical Internet with its "sender just gets one path", this can be a highly beneficial enhancement to explore (not that this was already proposed, maybe way ahead of its time and with vastly different mechanisms in solutions as early as [RFC1621], [RFC1622]).

In this version of the memo, these are only limited considerations about how to refine details of the proposal to find incremental, near-term deployment options, for example by using existing IPv6 headers and using an unassigned prefix to define FA-IINAS addressing semantic into it (limited of course to 128 bit then). These type of considerations can be subject for future revisions of this memo.

6. Changelog

00: Initial version

01: Refresh, new co-author

7. Informative References

[CiscoNAT] Akkiraju, P., Delgadillo, K., and Y. Rekhter, "Enabling Enterprise Multihoming with Cisco IOS Network Address Translation (NAT)", 2000, <http://staff.ustc.edu.cn/~james/cisco/nat/emios_wp.htm>.

[Haoyu] Song, H., Zhang, Z., Qu, Y., and J. Guichard, "Adaptive Addresses for Next Generation IP Protocol in Hierarchical Networks", IEEE 2020 IEEE 28th International Conference on Network Protocols (ICNP), n.d..

[I-D.draft-farrel-soon] Farrel, A., "A Definition of the Term "Soon" for Use in Discussions with Working Group Chairs and Area Directors", Work in Progress, Internet-Draft, draft-farrel-soon-07, 8 March 2021, <<https://www.ietf.org/archive/id/draft-farrel-soon-07.txt>>.

[I-D.jia-flex-ip-address-structure] Jia, Y., Chen, Z., and S. Jiang, "Flexible IP: An Adaptable IP Address Structure", Work in Progress, Internet-Draft, draft-jia-flex-ip-address-structure-00, 31 October 2020, <<https://www.ietf.org/archive/id/draft-jia-flex-ip-address-structure-00.txt>>.

- [I-D.jia-intarea-scenarios-problems-addressing]
Jia, Y., Trossen, D., Iannone, L., Shenoy, N., Mendes, P., 3rd, D. E. E., and P. Liu, "Challenging Scenarios and Problems in Internet Addressing", Work in Progress, Internet-Draft, draft-jia-intarea-scenarios-problems-addressing-02, 23 October 2021, <<https://www.ietf.org/archive/id/draft-jia-intarea-scenarios-problems-addressing-02.txt>>.
- [I-D.king-irtf-challenges-in-routing]
King, D. and A. Farrel, "Challenges for the Internet Routing Infrastructure Introduced by Changes in Address Semantics", Work in Progress, Internet-Draft, draft-king-irtf-challenges-in-routing-03, 14 June 2021, <<https://www.ietf.org/archive/id/draft-king-irtf-challenges-in-routing-03.txt>>.
- [I-D.king-irtf-semantic-routing-survey]
King, D. and A. Farrel, "A Survey of Semantic Internet Routing Techniques", Work in Progress, Internet-Draft, draft-king-irtf-semantic-routing-survey-02, 28 June 2021, <<https://www.ietf.org/archive/id/draft-king-irtf-semantic-routing-survey-02.txt>>.
- [RFC1621] Francis, P., "Pip Near-term Architecture", RFC 1621, DOI 10.17487/RFC1621, May 1994, <<https://www.rfc-editor.org/info/rfc1621>>.
- [RFC1622] Francis, P., "Pip Header Processing", RFC 1622, DOI 10.17487/RFC1622, May 1994, <<https://www.rfc-editor.org/info/rfc1622>>.
- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <<https://www.rfc-editor.org/info/rfc1918>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<https://www.rfc-editor.org/info/rfc3168>>.

- [RFC3306] Haberman, B. and D. Thaler, "Unicast-Prefix-based IPv6 Multicast Addresses", RFC 3306, DOI 10.17487/RFC3306, August 2002, <<https://www.rfc-editor.org/info/rfc3306>>.
- [RFC3956] Savola, P. and B. Haberman, "Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address", RFC 3956, DOI 10.17487/RFC3956, November 2004, <<https://www.rfc-editor.org/info/rfc3956>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<https://www.rfc-editor.org/info/rfc4193>>.
- [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>>.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, DOI 10.17487/RFC4607, August 2006, <<https://www.rfc-editor.org/info/rfc4607>>.
- [RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <<https://www.rfc-editor.org/info/rfc6282>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.

- [RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <<https://www.rfc-editor.org/info/rfc8520>>.
- [RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.
- [RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <<https://www.rfc-editor.org/info/rfc8799>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.
- [RFC8990] Bormann, C., Carpenter, B., Ed., and B. Liu, Ed., "GeneRic Autonomic Signaling Protocol (GRASP)", RFC 8990, DOI 10.17487/RFC8990, May 2021, <<https://www.rfc-editor.org/info/rfc8990>>.
- [RFC8994] Eckert, T., Ed., Behringer, M., Ed., and S. Bjarnason, "An Autonomic Control Plane (ACP)", RFC 8994, DOI 10.17487/RFC8994, May 2021, <<https://www.rfc-editor.org/info/rfc8994>>.

Authors' Addresses

Toerless Eckert
Futurewei Technologies USA
Santa Clara, CA 95050
United States of America

Email: tte@cs.fau.de

Nirmala Shenoy
Rochester Institute of Technology
New York, NY 14623
United States of America

Email: nxsvks@rit.edu

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 7, 2022

G. Fioccola
T. Zhou
Huawei
M. Cociglio
Telecom Italia
February 3, 2022

Segment Routing Header encapsulation for Alternate Marking Method
draft-fz-spring-srv6-alt-mark-02

Abstract

This document describes how the Alternate Marking Method can be used as the passive performance measurement tool in an SRv6 network. It defines how Alternate Marking data fields are transported as part of the Segment Routing with IPv6 data plane (SRv6) header.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on August 7, 2022.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

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

Table of Contents

1. Introduction	2
2. Application of the Alternate Marking to SRv6	3
2.1. Controlled Domain	4
3. Definition of the SRH AltMark TLV	4
3.1. Data Fields Format	4
4. Use of the SRH AltMark TLV	6
5. Alternate Marking Method Operation	7
6. Security Considerations	7
7. IANA Considerations	7
8. Acknowledgements	8
9. References	8
9.1. Normative References	8
9.2. Informative References	8
Authors' Addresses	9

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.

This document defines how the Alternate Marking Method ([RFC8321]) can be used to measure packet loss and delay metrics for Segment Routing with IPv6 data plane (SRv6).

[RFC8754] defines the Segment Routing Header (SRH) and how it is used by nodes that are Segment Routing (SR) capable.

[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. [I-D.ietf-6man-ipv6-alt-mark] defines a new TLV that can be encoded in the Option Headers (both Hop-by-hop or Destination) for the purpose of the Alternate Marking Method application in an IPv6 domain.

This document defines how Alternate Marking data is carried as SRH TLV, that can be piggybacked in the packet and transported as part of the SRH. The usage of SRH TLV is introduced in [RFC8754].

2. Application of the Alternate Marking to SRv6

The Alternate Marking Method requires a marking field. A possibility is already offered by [I-D.ietf-6man-ipv6-alt-mark] while the use of a new TLV to be encoded in the SRH is defined in this document.

Since [I-D.ietf-6man-ipv6-alt-mark] defines the IPv6 Application of the Alternate Marking Method through both Hop-by-Hop and Destination Options Header, it is applicable also to SRv6 network. Indeed the use of Destination Option Header carrying Alternate Marking bits coupled with SRH allows to monitor every node along the SR path.

This document introduces the SRH TLV carrying Alternate Marking bits and this can be a preferred approach in case of SRv6 network since it does not rely on the use of Destination Option Header.

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 uniqueness of the FlowMonID and how to allow disambiguation of the FlowMonID in case of collision.

The following section highlights an important requirement for the application of the Alternate Marking to IPv6 and SRv6. The concept of the controlled domain is explained and it is considered an essential precondition.

2.1. Controlled Domain

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

IPv6 has much more flexibility than IPv4 and innovative applications have been proposed, but for a number of reasons, such as the policies, options supported, the style of network management and security requirements, it is suggested to limit some of these applications to a controlled domain. This is also the case of the Alternate Marking application to SRv6 as assumed hereinafter.

Therefore, the application of the Alternate Marking Method to SRv6 MUST NOT be deployed outside a controlled domain. It is RECOMMENDED that an implementation can be able to reject packets that carry Alternate Marking data and are entering or leaving the controlled domains.

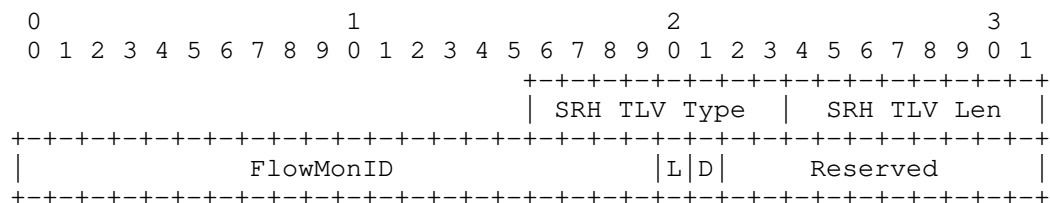
3. Definition of the SRH AltMark TLV

A new TLV carrying the data fields dedicated to the alternate marking method can be defined for the SRH extension headers.

This enables the Alternate Marking Method to take advantage of the network programmability capability of SRv6 ([I-D.ietf-spring-srv6-network-programming]). Specifically, the ability for an SRv6 endpoint to determine whether to process or ignore some specific SRH TLVs is based on the SID function. The nodes that are not capable of supporting the Alternate Marking functionality do not have to look or process the SRH AltMark TLV and can simply ignore it. This also enables collection of Alternate Marking data only from the supporting segment endpoints.

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 as SRH TLV.



where:

- o SRH TLV Type: 8 bit identifier of the type of Option/TLV that needs to be allocated. Unrecognised Types MUST be ignored on receipt.
- o SRH TLV Len: The length of the Data Fields of this TLV in bytes.
- o FlowMonID: 20 bits unsigned integer. The FlowMon identifier is described hereinafter.
- o L: Loss flag as defined in [RFC8321] and [I-D.ietf-6man-ipv6-alt-mark];
- o D: Delay flag as defined in [RFC8321] and [I-D.ietf-6man-ipv6-alt-mark];
- o Reserved: is reserved for future use. These bits MUST be set to zero on transmission and ignored on receipt.

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.

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.

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.

4. Use of the SRH AltMark TLV

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 a routing header, destination options before the routing header are processed by each destination in the route list.

SRH TLV can also be used to encode the AltMark Data Fields for SRv6 and to monitor every node along the SR path. For SRv6, it may be preferred to use the SRH TLV, while for all the other cases with IPv6 data plane the use of the Hop-by-Hop and Destination Option to carry AltMark data fields (as described in [I-D.ietf-6man-ipv6-alt-mark]) is the best choice.

It is to be noted that the SR nodes implementing the Alternate Marking functionality follows the MTU and other considerations outlined in [I-D.voyer-6man-extension-header-insertion]. Furthermore, in a SRv6 network, the intermediated nodes that are not in the SID list do not consider the SRH, therefore they cannot support and dig into the SRH TLV.

It is possible to summarize the procedure for AltMark data encapsulation in SRv6 SRH:

- * Ingress Node: As part of the SRH encapsulation, the ingress node of an SR domain or an SR Policy [I-D.ietf-spring-segment-routing-policy] MAY add the AltMark TLV in the SRH of the data packet, if it supports AltMark functionality and based on local configuration.

- * Intermediate SR Node: The intermediate SR node is any node receiving an IPv6 packet where the destination address of that packet is a local SID. If an intermediate SR node is not capable of processing AltMark TLV, it simply ignores it. While, if an intermediate SR node is capable of processing AltMark TLV, it checks if SRH AltMark TLV is present in the packet using procedures defined in [RFC8754] and process it.

- * Egress Node: The Egress node is the last node in the segment-list of the SRH. The processing of AltMark TLV at the Egress node

is similar to the processing of AltMark TLV at the Intermediate SR Nodes.

5. Alternate Marking Method Operation

[RFC8321], [RFC8889] describe the Alternate Marking Method in general. While [I-D.ietf-6man-ipv6-alt-mark] describe in detail the application and the Operation of the methodology for IPv6.

6. Security Considerations

The security considerations of SRv6 are discussed in [RFC8754] and [I-D.ietf-spring-srv6-network-programming], and the security considerations of Alternate Marking in general and its application to IPv6 are discussed in [RFC8321] and [I-D.ietf-6man-ipv6-alt-mark].

The Alternate Marking application to IPv6, defined in [I-D.ietf-6man-ipv6-alt-mark], analyzes different security concerns and related solutions. These aspects are valid and applicable also to this document. In particular the fundamental security requirement is that Alternate Marking MUST be applied in a specific limited domain, as also mentioned in [RFC8799].

Alternate Marking is a feature applied to a trusted domain, where one or several operators decide on leveraging and configuring Alternate Marking according to their needs. Additionally, operators need to properly secure the Alternate Marking domain to avoid malicious configuration and attacks, which could include injecting malicious packets into a domain. So the implementation of Alternate Marking is applied within a controlled domain where the network nodes are locally administered. A limited administrative domain provides the network administrator with the means to select, monitor and control the access to the network.

7. IANA Considerations

The SRH TLV Type should be assigned in IANA's Segment Routing Header TLVs Registry.

This draft requests to allocate a SRH TLV Type for Alternate Marking TLV data fields under registry name "Segment Routing Header TLVs" requested by [RFC8754].

SRH TLV Type	Description	Reference
TBD	AltMark Data Fields TLV	This document

8. Acknowledgements

TBD

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>>.

9.2. Informative References

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

[I-D.ietf-6man-ipv6-alt-mark]
Fioccola, G., Zhou, T., Cociglio, M., Qin, F., and R. Pang, "IPv6 Application of the Alternate Marking Method", draft-ietf-6man-ipv6-alt-mark-12 (work in progress), October 2021.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-16 (work in progress), January 2022.

[I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Garvia, P. C., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", draft-ietf-spring-srv6-network-programming-28 (work in progress), December 2020.

[I-D.voyer-6man-extension-header-insertion]
Voyer, D., Filsfils, C., Dukes, D., Matsushima, S., Leddy, J., Li, Z., and J. Guichard, "Deployments With Insertion of IPv6 Segment Routing Headers", draft-voyer-6man-extension-header-insertion-10 (work in progress), November 2020.

- [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>>.
- [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>>.
- [RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <<https://www.rfc-editor.org/info/rfc8799>>.
- [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

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: 19 August 2022

R. Gandhi, Ed.
C. Filsfils
Cisco Systems, Inc.
N. Vaghamshi
Reliance
M. Nagarajah
Telstra
R. Foote
Nokia
M. Chen
Huawei
A. Dhamija
Rakuten
15 February 2022

Enhanced Performance Measurement Using Simple TWAMP in Segment Routing
Networks
draft-gandhi-spring-enhanced-srpm-01

Abstract

Segment Routing (SR) leverages the source routing paradigm. SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes. This document defines procedure for Enhanced Performance Measurement of end-to-end SR paths including SR Policies for both SR-MPLS and SRv6 data planes using Simple Two-Way Active Measurement Protocol (STAMP) defined in RFC 8762. The procedure reduces the deployment and operational complexities in a network.

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 19 August 2022.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Conventions Used in This Document	3
2.1. Requirements Language	4
2.2. Abbreviations	4
2.3. Reference Topology	5
3. Overview	5
3.1. Enhanced Loopback Mode Enabled with Network Programming Function	5
3.2. Example Provisioning Model	6
4. Enhanced Performance Measurement Procedure	7
4.1. Enhanced Performance Measurement Procedure for SR-MPLS Policies	7
4.1.1. Timestamp Label Allocation	9
4.1.2. Node Capability for Timestamp Label	9
4.2. Enhanced Performance Measurement Procedure for SRv6 Policies	9
4.2.1. Timestamp Endpoint Function Assignment	11
4.2.2. Node Capability for Timestamp Endpoint Function	12
5. Example Failure Notifications	12
6. Security Considerations	13
7. IANA Considerations	13
8. References	14
8.1. Normative References	14
8.2. Informative References	15
Acknowledgments	16
Authors' Addresses	16

1. Introduction

Segment Routing (SR) leverages the source routing paradigm and greatly simplifies network operations for Software Defined Networks (SDNs). SR is applicable to both Multiprotocol Label Switching (SR-MPLS) and IPv6 (SRv6) data planes [RFC8402]. SR Policies as defined in [I-D.ietf-spring-segment-routing-policy] are used to steer traffic through a specific, user-defined paths using a stack of Segments. A comprehensive SR Performance Measurement (PM) for delay and packet loss as well as Connectivity Verification (CV) is one of the essential requirements to measure network performance to provide Service Level Agreements (SLAs).

The Simple Two-Way Active Measurement Protocol (STAMP) provides capabilities for the measurement of various performance metrics in IP networks [RFC8762] without the use of a control channel to pre-signal session parameters. As described in [I-D.ietf-spring-stamp-srpm], STAMP can be used for performance measurement for delay and packet loss of end-to-end SR paths.

Seamless Bidirectional Forwarding Detection (S-BFD) [RFC7880] provides a simplified mechanism for using BFD for path monitoring with a large proportion of negotiation aspects eliminated. The S-BFD can be used for connectivity verification of end-to-end SR paths.

Both STAMP and S-BFD require protocol support on the far-end Reflector node to process the received packets, and hence the received packets need to be punted from the forwarding fast path and return packets need to be generated. This limits the scale for number sessions and the ability to provide faster detection interval.

Enabling multiple protocols, S-BFD for connectivity verification and STAMP for performance measurement increases the deployment and operational complexities in a network. Also, implementing multiple protocols in a hardware significantly increases the development cost.

This document defines procedure for Enhanced Performance Measurement of end-to-end SR paths including SR Policies for both SR-MPLS and SRv6 data planes, using Simple Two-Way Active Measurement Protocol (STAMP) defined in [RFC8762]. The procedure reduces the deployment and operational complexities in a network.

2. Conventions Used in This Document

2.1. Requirements Language

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

2.2. Abbreviations

S-BFD: Seamless Bidirectional Forwarding Detection.

BSID: Binding Segment ID.

ECMP: Equal Cost Multi-Path.

EB: Endpoint Behaviour.

HMAC: Hashed Message Authentication Code.

MBZ: Must be Zero.

MPLS: Multiprotocol Label Switching.

PM: Performance Measurement.

PTP: Precision Time Protocol.

SID: Segment ID.

SL: Segment List.

SR: Segment Routing.

SRH: Segment Routing Header.

SR-MPLS: Segment Routing with MPLS data plane.

SRv6: Segment Routing with IPv6 data plane.

STAMP: Simple Two-way Active Measurement Protocol.

TC: Traffic Class.

TTL: Time To Live.

2.3. Reference Topology

In the reference topology shown in Figure 1, the STAMP Session-Sender [RFC8762] S1 initiates a Session-Sender test packet and the Session-Reflector R1 returns the test packet. The return test packet may be transmitted back to the Session-Sender S1 on the same path (same set of links and nodes) or a different path in the reverse direction from the path taken towards the Session-Reflector R1.

The Session-Sender S1 and Session-Reflector R1 are connected via an SR path [RFC8402]. The SR path can be an SR Policy [I-D.ietf-spring-segment-routing-policy] on node S1 (called head-end) with destination to node R1 (called tail-end).

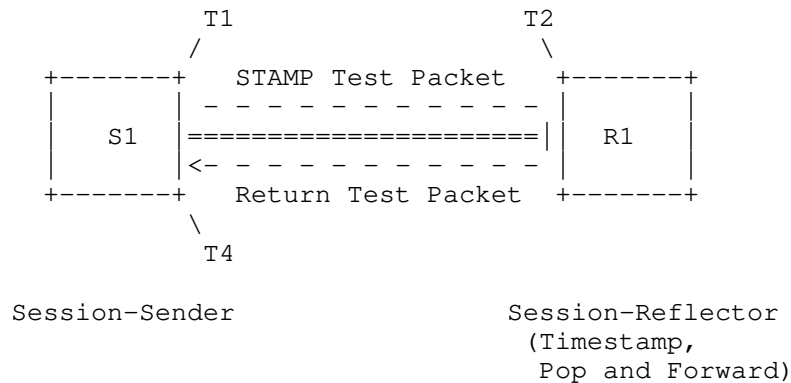


Figure 1: Loopback Mode Enabled with Network Programming Function

3. Overview

3.1. Enhanced Loopback Mode Enabled with Network Programming Function

As described in [I-D.ietf-spring-stamp-srpm], in loopback mode, the STAMP Session-Sender S1 initiates Session-Sender test packets and the Session-Reflector R1 forwards them back to the Session-Sender S1. The received STAMP test packets are not punted out of the fast path in forwarding at the Session-Reflector. At the Session-Reflector, the loopback function simply makes the necessary changes to the encapsulation including IP and UDP headers to return the STAMP test packet to the Session-Sender S1. No STAMP test session is created on the Session-Reflector R1. As described in [I-D.ietf-spring-stamp-srpm], only round-trip delay can be measured in the loopback mode. In SR networks, there is also a need to measure one-way delay to provide low latency services.

This document defines a new STAMP measurement mode, enhanced loopback mode, that is loopback mode enabled with network programming function. In this mode, both transmit (T1) and receive (T2) timestamps in data plane are collected by the Session-Sender test packets as shown in Figure 1. The network programming function optimizes the "operations of punt test packet and generate return test packet" on the Session-Reflector as timestamping is implemented in forwarding fast path in hardware. This helps to achieve higher STAMP test session scale and faster detection interval.

The Session-Sender adds transmit timestamp (T1) in the payload of the Session-Sender test packet. The Session-Reflector adds the receive timestamp (T2) in the payload of the received test packet in forwarding fast path in hardware without punting the test packet (e.g. to slow path or control-plane). The network programming function enables Session-Reflector to add the receive timestamp (T2) at a specific offset in the payload which is locally provisioned, consistently in the network.

3.2. Example Provisioning Model

An example provisioning model and typical measurement parameters are shown in Figure 2:

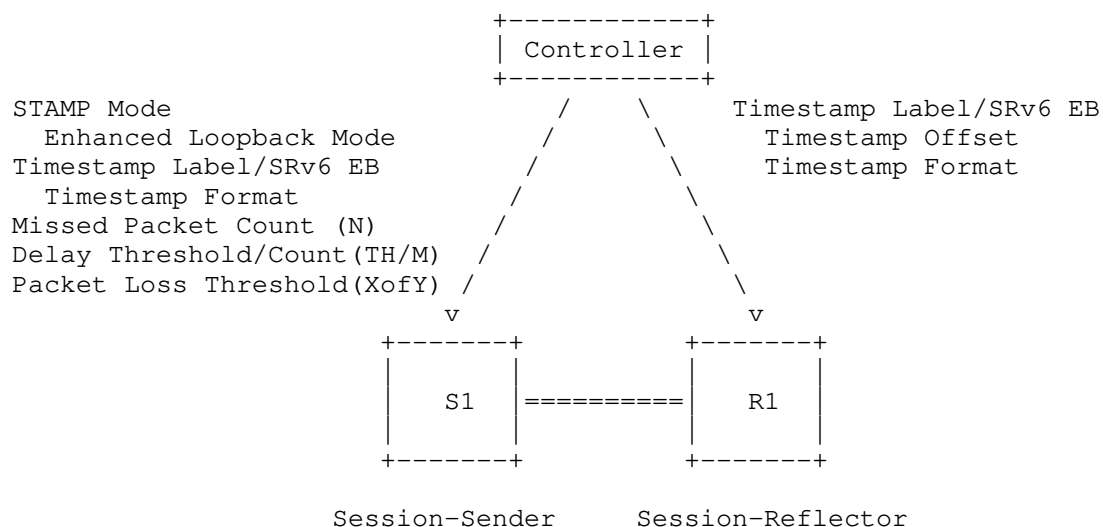


Figure 2: Example Provisioning Model

Example of a STAMP mode is enhanced loopback mode defined in this document. The values for Timestamp Label and SRv6 Endpoint Behaviour may be provisioned as described in this document. Example of

Timestamp Format is 64-bit PTPv2 [IEEE1588]. Example of Timestamp Offset is 16 and 32 bytes for the unauthenticated and authenticated STAMP Session-Sender test packets, respectively. Example of threshold values configured for generating notifications are: Missed Packet Count (N), Delay Exceeded Threshold and Packet Count (TH/M) and Packet Loss Threshold (XofY), as described in this document.

The mechanisms to provision the Session-Sender and Session-Reflector are outside the scope of this document.

4. Enhanced Performance Measurement Procedure

For enhanced performance monitoring of an end-to-end SR path including SR Policy, STAMP Session-Sender test packets are transmitted in loopback mode enabled with network programming function to timestamp and forward the packet.

For SR Policy, the Session-Sender test packets are transmitted using the Segment List (SL) of the Candidate-Path [I-D.ietf-spring-segment-routing-policy]. When a Candidate-Path has more than one Segment Lists, multiple Session-Sender test packets MUST be transmitted, one using each Segment List.

4.1. Enhanced Performance Measurement Procedure for SR-MPLS Policies

An SR-MPLS Policy may contain a number of Segment Lists (SLs). A Session-Sender test packet MUST be transmitted for each Segment List of the SR-MPLS Policy. The content of an example Session-Sender test packet for an end-to-end SR-MPLS Policy is shown in Figure 3.

The SR-MPLS header can contain the MPLS label stack of the forward path or both forward and the reverse direction paths. In the former case, the return test packets are received by the Session-Sender via IP/UDP [RFC0768] return path and the MPLS header is removed by the Session-Reflector.

In the latter case, the Segment List of the reverse direction SR path is added in the Session-Sender test packet header to receive the return test packet on a specific path, either using the Binding SID [I-D.ietf-pce-binding-label-sid] or Segment List of the Reverse SR Policy [I-D.ietf-pce-sr-bidir-path]. In this case, the MPLS header is not removed by the Session-Reflector.

In both cases, the Session-Sender MUST set the Destination Address equal to the Session-Sender address in the IP header of the test packets.

In this document, two new Timestamp Labels are defined for SR-MPLS data plane to enable network programming function for "timestamp, pop and forward" the received test packet, one for unauthenticated mode and one for authenticated mode.

In the Session-Sender test packets for SR-MPLS Policies, a Timestamp Label is added in the MPLS header as shown in Figure 3, to collect "Receive Timestamp" field in the payload of the test packet. The Label Stack for the reverse direction SR-MPLS path can be added after the Timestamp Label (not shown in the Figure) to receive the return test packet on a specific path. When a Session-Reflector receives a packet with Timestamp Label, after timestamping the packet at a specific offset, the Session-Reflector pops the Timestamp Label and forwards the packet using the next label or IP header in the packet (just like the data packets for the normal traffic).

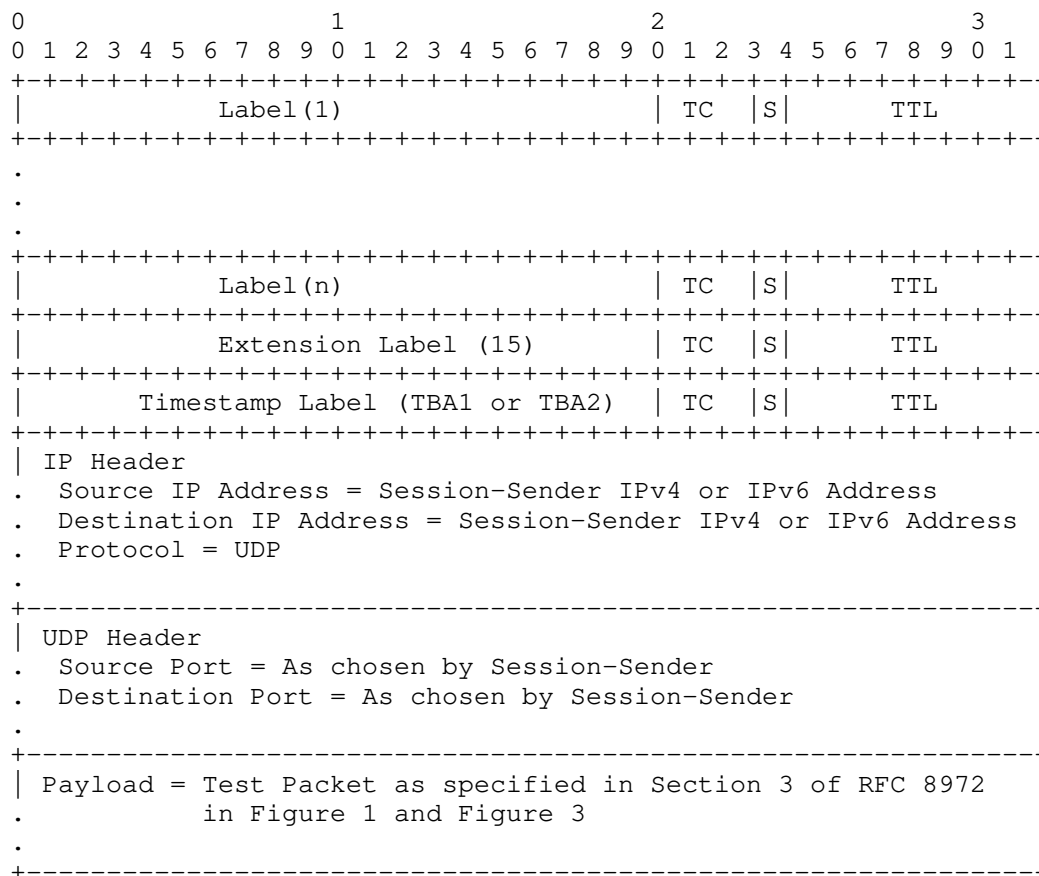


Figure 3: Example STAMP Test Packet with Timestamp Label for SR-MPLS

4.1.1. Timestamp Label Allocation

The timestamp Labels for STAMP test packets in unauthenticated and authenticated modes can be allocated using one of the following methods:

- * Labels (values TBA1 and TBA2) assigned by IANA from the "Extended Special-Purpose MPLS Values" [RFC9017]. For Label (value TBA1), the timestamp offset is fixed at byte-offset 16 from the start of the payload for the STAMP test packets in unauthenticated mode, and Label (value TBA2) at byte-offset 32 from the start of the payload for the STAMP test packets in authenticated mode, both using the timestamp format 64-bit PTPv2.
- * Labels allocated by a Controller from the global table of the Session-Reflector. The Controller provisions the labels on both Session-Sender and Session-Reflector, as well as timestamp offsets and timestamp formats.
- * Labels allocated by the Session-Reflector. The signaling and IGP flooding extension for the labels (including their timestamp offsets and timestamp formats) are outside the scope of this document.

4.1.2. Node Capability for Timestamp Label

The STAMP Session-Sender needs to know if the Session-Reflector can process the Timestamp Label to avoid dropping test packets. The signaling extension or local configuration for this capability exchange is outside the scope of this document.

4.2. Enhanced Performance Measurement Procedure for SRv6 Policies

An SRv6 Policy may contain a number of Segment Lists. Each Segment List may contain a number of SRv6 SIDs as defined in [RFC8986], [I-D.filsfils-spring-net-pgm-extension-srv6-usid] and [I-D.ietf-spring-srv6-srh-compression]. A Session-Sender test packet MUST be transmitted for each Segment List of the SRv6 Policy. An SRv6 Policy may contain an SRv6 Segment Routing Header (SRH) carrying a Segment List as described in [RFC8754]. The content of an example Session-Sender test packet for an end-to-end SRv6 Policy using an SRH is shown in Figure 4.

The SRH can contain the Segment List of the forward path only or both forward and the reverse direction paths. In the former case, an inner IPv6 header (after the SRH and before the UDP header) MUST be added that contains the Destination Address equal to the Session-Sender address as shown in Figure 4. In this case, the SRH is removed by the Session-Reflector and IP/UDP return path is used.

In the latter case, the Segment List of the reverse direction SR path is added in the SRH to receive the return test packet on a specific path, either using the Binding SID [I-D.ietf-pce-binding-label-sid] or Segment List of the Reverse SR Policy [I-D.ietf-pce-sr-bidir-path]. In this case, the SRH is not removed by the Session-Reflector and an inner IPv6 header is not required. When the return test packet contains an SRH at the Session-Sender, the procedure defined for upper-layer header processing for SRv6 SIDs in [RFC8986] MUST be used to process the UDP header after the SRH in the received test packets.

The [RFC8986] defines SRv6 Endpoint Behaviours (EB) for SRv6 nodes. In this document, two new Timestamp Endpoint Behaviours are defined for Segment Routing Header (SRH) [RFC8754] to enable "Timestamp and Forward (TSF)" function for the received test packets, one for unauthenticated mode and one for authenticated mode.

In the Session-Sender test packets for SRv6 Policies, Timestamp Endpoint Function (End.TSF) is carried with the target Segment Identifier (SID) in SRH [RFC8754] as shown in Figure 4, to collect "Receive Timestamp" field in the payload of the test packet. The Segment List for the reverse direction path can be added after the target SID to receive the return test packet on a specific path. When a Session-Reflector receives a packet with Timestamp Endpoint (End.TSF) for the target SID which is local, after timestamping the packet at a specific offset, the Session-Reflector forwards the packet using the next SID in the SRH or inner IPv6 header in the packet (just like the data packets for the normal traffic).

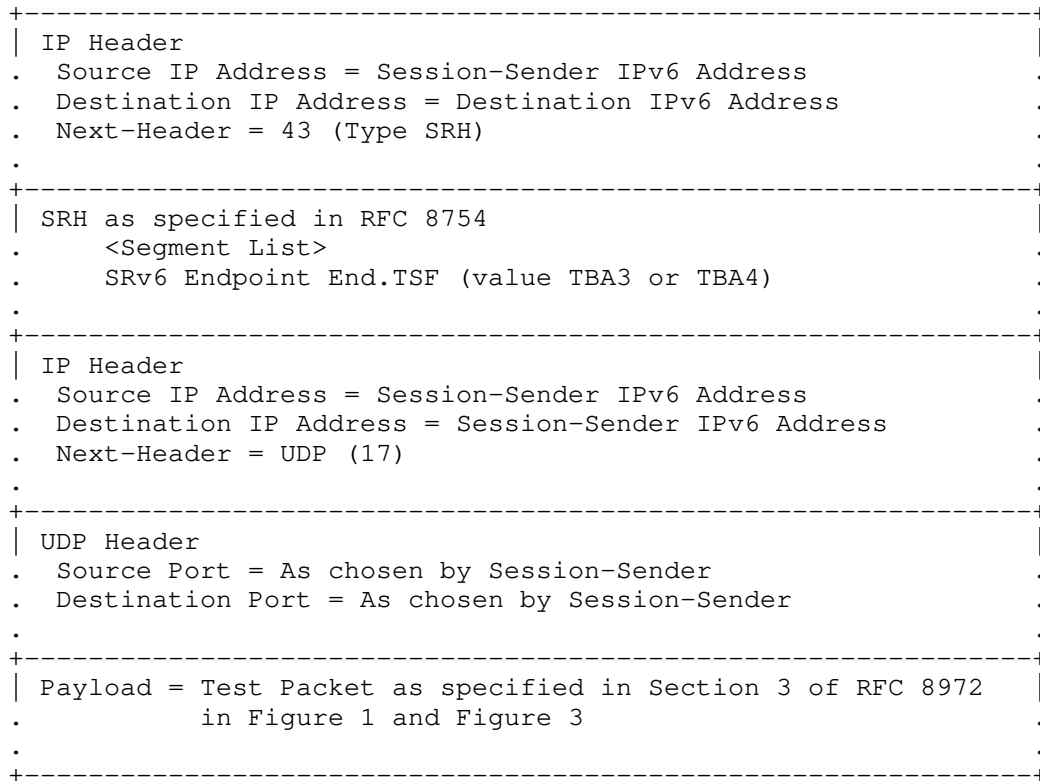


Figure 4: Example STAMP Test Packet with Endpoint Function for SRv6

4.2.1. Timestamp Endpoint Function Assignment

The Timestamp Endpoint Functions for "Timestamp and Forward" can be signaled using one of the following methods:

- * Timestamp Endpoint Functions (values TBA3 and TBA4) assigned by IANA from the "SRv6 Endpoint Behaviors Registry". For endpoint behaviour (value TBA3), the timestamp offset is fixed at byte-offset 16 from the start of the payload for the STAMP test packets in unauthenticated mode, and endpoint behaviour (value TBA4) at byte-offset 32 from the start of the payload for the STAMP test packets in authenticated mode, both using the timestamp format 64-bit PTPv2.
- * Timestamp Endpoint Functions assigned by a Controller. The Controller provisions the values on both Session-Sender and Session-Reflector, as well as timestamp offsets and timestamp formats.

- * Timestamp Endpoint Functions assigned by the Session-Reflector. The signaling and IGP flooding extension for the endpoint functions (including timestamp offsets and timestamp formats) are outside the scope of this document.

4.2.2. Node Capability for Timestamp Endpoint Function

The STAMP Session-Sender needs to know if the Session-Reflector can process the Timestamp Endpoint Function to avoid dropping test packets. The signaling extension for this capability exchange is outside the scope of this document.

5. Example Failure Notifications

The timestamps T1 and T2 are used to measure the one-way delay. The delay metrics for an end-to-end SR path are notified, for example, when consecutive M number of test packets have measured delay values exceed the user-configured threshold TH, where M (Delay Exceeded Packet Count) and TH (Absolute and Percentage Delay Exceeded Thresholds) are also locally provisioned values.

The round-trip packet loss for an end-to-end SR path is calculated using the Sequence Number in the Session-Sender test packets. The packet loss metric is notified when X number of Session-Sender test packets were lost out of last Y number of test packets transmitted by the Session-Sender, where Threshold XofY is locally provisioned value.

STAMP session state as UP (i.e. Connectivity verification success) for an end-to-end SR path is initially notified as soon as one or more return test packets are received at the Session-Sender.

STAMP session state as DOWN (i.e. Connectivity verification failure) for an end-to-end SR path is notified when consecutive N number of return test packets are not received at the Session-Sender, where N (Missed Packet Count) is a locally provisioned value.

In the loopback mode, a connectivity verification failure on the reverse direction path can cause the return test packets to not reach the Session-Sender. This is also true in the case where the return test packets are generated by the stateless Session-Reflector in two-way measurement. The stateful Session-Reflector can solve this issue by maintaining the forwarding direction state and notifying a connectivity verification success and failure to the Session-Sender.

6. Security Considerations

The STAMP protocol is intended for deployment in limited domains [RFC8799]. As such, it assumes that a node involved in the STAMP protocol operation has previously verified the integrity of the path and the identity of the far-end Session-Reflector.

The security considerations specified in [RFC8762] and [RFC8972] also apply to the procedures defined in this document. Specifically, the message integrity protection using HMAC, as defined in Section 4.4 of [RFC8762] also apply to the procedure described in this document.

7. IANA Considerations

IANA maintains the "Special-Purpose Multiprotocol Label Switching (MPLS) Label Values" registry (see <<https://www.iana.org/assignments/mpls-label-values/mpls-label-values.xml>>). IANA is requested to allocate Timestamp Label value from the "Extended Special-Purpose MPLS Label Values" registry:

Value	Description	Reference
TBA1	Timestamp Label for offset 16 for STAMP in Unauthenticated Mode	This document
TBA2	Timestamp Label for offset 32 for STAMP in Authenticated Mode	This document

IANA is requested to allocate, within the "SRv6 Endpoint Behaviors Registry" sub-registry belonging to the top-level "Segment Routing Parameters" registry [RFC8986], the following allocation:

Value	Endpoint Behavior	Reference
TBA3	End.TSF (Timestamp and Forward) for offset 16 for STAMP in Unauthenticated Mode	This document
TBA4	End.TSF (Timestamp and Forward) for offset 32 for STAMP in Authenticated Mode	This document

8. References

8.1. Normative References

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <<https://www.rfc-editor.org/info/rfc768>>.
- [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>>.
- [RFC8762] Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple Two-Way Active Measurement Protocol", RFC 8762, DOI 10.17487/RFC8762, March 2020, <<https://www.rfc-editor.org/info/rfc8762>>.
- [RFC8972] Mirsky, G., Min, X., Nydell, H., Foote, R., Masputra, A., and E. Ruffini, "Simple Two-Way Active Measurement Protocol Optional Extensions", RFC 8972, DOI 10.17487/RFC8972, January 2021, <<https://www.rfc-editor.org/info/rfc8972>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.

[I-D.ietf-spring-stamp-srpm]
Gandhi, R., Filsfils, C., Voyer, D., Chen, M., Janssens, B., and R. Foote, "Performance Measurement Using Simple TWAMP (STAMP) for Segment Routing Networks", Work in Progress, Internet-Draft, draft-ietf-spring-stamp-srpm-03, 1 February 2022, <<https://www.ietf.org/archive/id/draft-ietf-spring-stamp-srpm-03.txt>>.

8.2. Informative References

- [IEEE1588] IEEE, "1588-2008 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", March 2008.
- [RFC7880] Pignataro, C., Ward, D., Akiya, N., Bhatia, M., and S. Pallagatti, "Seamless Bidirectional Forwarding Detection (S-BFD)", RFC 7880, DOI 10.17487/RFC7880, July 2016, <<https://www.rfc-editor.org/info/rfc7880>>.
- [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>>.
- [RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <<https://www.rfc-editor.org/info/rfc8799>>.
- [RFC9017] Andersson, L., Kompella, K., and A. Farrel, "Special-Purpose Label Terminology", RFC 9017, DOI 10.17487/RFC9017, April 2021, <<https://www.rfc-editor.org/info/rfc9017>>.
- [I-D.ietf-spring-srv6-srh-compression]
Cheng, W., Filsfils, C., Li, Z., Decraene, B., Cai, D., Voyer, D., Clad, F., Zadok, S., Guichard, J. N., Aihua, L., Raszuk, R., and C. Li, "Compressed SRv6 Segment List Encoding in SRH", Work in Progress, Internet-Draft, draft-ietf-spring-srv6-srh-compression-00, 11 February 2022, <<https://www.ietf.org/archive/id/draft-ietf-spring-srv6-srh-compression-00.txt>>.

[I-D.filsfils-spring-net-pgm-extension-srv6-usid]
Filsfils, C., Garvia, P. C., Cai, D., Voyer, D., Meilik, I., Patel, K., Henderickx, W., Jonnalagadda, P., Melman, D., Liu, Y., and J. Guichard, "Network Programming extension: SRv6 uSID instruction", Work in Progress, Internet-Draft, draft-filsfils-spring-net-pgm-extension-srv6-usid-12, 13 December 2021, <<https://www.ietf.org/archive/id/draft-filsfils-spring-net-pgm-extension-srv6-usid-12.txt>>.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", Work in Progress, Internet-Draft, draft-ietf-spring-segment-routing-policy-16, 28 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-spring-segment-routing-policy-16.txt>>.

[I-D.ietf-pce-binding-label-sid]
Sivabalan, S., Filsfils, C., Tantsura, J., Previdi, S., and C. L. (editor), "Carrying Binding Label/Segment Identifier in PCE-based Networks.", Work in Progress, Internet-Draft, draft-ietf-pce-binding-label-sid-12, 24 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-pce-binding-label-sid-12.txt>>.

[I-D.ietf-pce-sr-bidir-path]
Li, C., Chen, M., Cheng, W., Gandhi, R., and Q. Xiong, "Path Computation Element Communication Protocol (PCEP) Extensions for Associated Bidirectional Segment Routing (SR) Paths", Work in Progress, Internet-Draft, draft-ietf-pce-sr-bidir-path-08, 9 September 2021, <<https://www.ietf.org/archive/id/draft-ietf-pce-sr-bidir-path-08.txt>>.

Acknowledgments

The authors would like to thank Greg Mirsky, Kireeti Kompella, and Adrian Farrel for providing useful comments.

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada

Email: rgandhi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.

Email: cfilsfil@cisco.com

Navin Vaghamshi
Reliance

Email: Navin.Vaghamshi@ril.com

Moses Nagarajah
Telstra

Email: Moses.Nagarajah@team.telstra.com

Richard Foote
Nokia

Email: footer.foote@nokia.com

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Amit Dhamija
Rakuten

Email: amit.dhamija@rakuten.com

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

Z. Li
J. Dong
Huawei Technologies
R. Pang
China Unicom
Y. Zhu
China Telecom
24 March 2022

Segment Routing for End-to-End IETF Network Slicing
draft-li-spring-sr-e2e-ietf-network-slicing-03

Abstract

Network slicing can be used to meet the connectivity and performance requirement of different services or customers in a shared network. An IETF network slice can be realized as enhanced VPNs (VPN+), which is delivered by integrating the overlay VPN service with a Virtual Transport Network (VTN) as the underlay. An end-to-end IETF network slice may span multiple network domains. Within each domain, traffic of the end-to-end network slice service is mapped to a domain VTN. In the context of IETF network slicing, a VTN can be instantiated as a Network Resource Partition (NRP).

When segment routing (SR) is used to build a multi-domain IETF network slices, information of the local network slices in each domain can be specified using special SR binding segments called NRP binding segments (NRP BSID). The multi-domain IETF network slice can be specified using a list of NRP BSIDs in the packet, each of which can be used by the corresponding domain edge nodes to steer the traffic of end-to-end IETF network slice into the specific NRP in the local domain.

This document describes the functionality of NRP binding segment and its instantiation in SR-MPLS and SRv6.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on 25 September 2022.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Segment Routing for IETF E2E Network Slicing	4
3. SRv6 NRP Binding Functions	5
3.1. End.B6NRP.Encaps	5
3.2. End.NRP.Encaps	6
3.3. End.BNRP.Encaps	7
4. SR-MPLS NRP BSIDs	8
5. IANA Considerations	9
6. Security Considerations	9
7. Acknowledgements	9
8. References	10
8.1. Normative References	10
8.2. Informative References	10
Authors' Addresses	11

1. Introduction

[I-D.ietf-teas-ietf-network-slices] introduces the concept and the characteristics of IETF network slice, and describes a general framework for IETF network slice management and operation. It also introduces the concept Network Resource Partition (NRP), which is a collection of resources identified in the underlay network.

[I-D.ietf-teas-enhanced-vpn] describes the framework and the candidate component technologies for providing enhanced VPN (VPN+) services based on existing VPN and Traffic Engineering (TE) technologies with enhanced characteristics that specific services require above traditional VPNs. It also introduces the concept of Virtual Transport Network (VTN). A Virtual Transport Network (VTN) is a virtual underlay network which consists of a set of dedicated or shared network resources allocated from the physical underlay network, and is associated with a customized logical network topology. VPN+ services can be delivered by mapping one or a group of overlay VPNs to the appropriate VTNs as the underlay, so as to provide the network characteristics required by the customers. Enhanced VPN (VPN+) and VTN can be used for the realization of IETF network slices. In the context of IETF network slicing, a VTN can be instantiated as an NRP. VTN and NRP are considered interchangeable terms in this document.

[I-D.dong-teas-nrp-scalability] describes the scalability considerations in the control plane and data plane to enable NRPs and provide the suggestions to improve the scalability of NRP. In the control plane, It proposes the approach of decoupling the topology and resource attributes of NRP, so that multiple NRPs may share the same topology and the result of topology based path computation. In the data plane, it proposes to carry a dedicated NRP-ID of a network domain in the data packet to determine the set of resources reserved for the corresponding NRP.

An IETF network slice may span multiple network domains. Within each domain, traffic of the end-to-end network slice is mapped to a local network slice. The NRP ID which identifies the NRP in the local domain for the end-to-end network slice needs to be determined on the domain edge node.

When segment routing (SR) is used to build a multi-domain IETF network slice, information of the local network slices in each domain can be specified using special SR binding segments called NRP binding segments (NRP BSID). The multi-domain IETF network slice can be specified using a list of NRP BSIDs in the packet, each of which can be used by the corresponding domain edge nodes to steer the traffic of end-to-end IETF network slice using the specific resource-aware segments or NRP-ID of the local domain.

This document describes the functionality of the network slice binding segment and its instantiation in SR-MPLS and SRv6.

2. Segment Routing for IETF E2E Network Slicing

[I-D.dong-teas-nrp-scalability] describes the scalability considerations in the control plane and data plane to create NRPs. In data plane, it proposes to carry a dedicated NRP-ID in data packet to determine the set of resources reserved for the corresponding NRP in a network domain.

[I-D.li-teas-e2e-ietf-network-slicing] describes the framework of carrying network slice related identifiers in the data plane, each of the network slice IDs may have a different network scope. It provides an approach of mapping the global NRP-ID to domain NRP-IDs at the network domain border nodes.

With Segment Routing, there are several optional approaches to realize the mapping between the end-to-end network slice and the network slice constructs in the local domain.

The first type of approaches are to use one type of NRP BSID to steer traffic to an SR Policy associated with a local NRP. This is called the NRP-TE BSID. There are some variants in terms of the detailed behavior:

- * The first variant is to use one type of NRP BSID to specify the mapping of traffic to a SR policy which consists of list of resource-aware segments [I-D.ietf-spring-resource-aware-segments] associated with a local NRP.
- * The second variant is to use one type of NRP BSID to specify the mapping of traffic to a SR policy which is bound to a local NRP-ID.

The second type of approaches is to use one type of NRP BSID to steer traffic to follow the shortest path within a local domain NRP. This is called the NRP-BE BSID. There are some variants in terms of the detailed behavior:

- * The first variant is to use one type of NRP BSID to determine a local NRP-ID, and instruct the encapsulation of the local NRP-ID into the packet at the domain edge node.
- * The second variant is to use one type of NRP BSID to specify the mapping of traffic to a local NRP, the local NRP-ID is specified in the associated fields by the ingress node, and is encapsulated into the packet at the domain edge node.

The behavior of the first type of NRP BSID is similar to the function of the existing SR BSID, the difference is it is associated with a particular NRP. The second type of the NRP BSID is different from the existing binding segment. The instantiation of the NRP BSIDs in SR-MPLS and SRv6 are described in the following sections.

3. SRv6 NRP Binding Functions

[RFC8986] defines the SRv6 Network Programming concept and specifies the base set of SRv6 behaviors. The SRv6 End.B6.Encaps function is defined to instantiate the Binding SID in SRv6, which can be reused as one type of NRP-TE BSID to specify the mapping of traffic to a list of resource-aware SRv6 segments of a domain NRP.

[I-D.ietf-6man-enhanced-vpn-vtn-id] describes the mechanism of carrying the VTN-ID of a network domain in the IPv6 Hop-by-Hop (HBH) extension header. For the type 2, 3, 4 of NRP binding segments described in section 2, three new SRv6 Binding functions are defined in the following sections.

3.1. End.B6NRP.Encaps

A new SRv6 function called End.B6NRP.Encaps: Endpoint bound to a SRv6 Policy in a NRP with IPv6 encapsulation is defined in this section. This is a variation of the End behavior. It instructs the endpoint node to determine an SRv6 Policy in a specific NRP of the local domain, and encapsulate the SID list of the SR Policy and the NRP-ID in a new IPv6 header.

Any SID instance of this behavior is associated with an SR Policy B, a NRP-ID V and a source address A.

When node N receives a packet whose IPv6 DA is S, and S is a local End.B6NRP.Encaps SID, N does the following:

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.     Stop processing the SRH, and proceed to process the next
        header in the packet, whose type is identified by
        the Next Header field in the routing header.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address
        with Code 0 (Hop limit exceeded in transit),
        interrupt packet processing, and discard the packet.
S07.   }
S08.   max_LE = (Hdr Ext Len / 2) - 1
S09.   If ((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
S10.     Send an ICMP Parameter Problem to the Source Address
        with Code 0 (Erroneous header field encountered)
        and Pointer set to the Segments Left field,
        interrupt packet processing, and discard the packet.
S11.   }
S12.   Decrement IPv6 Hop Limit by 1
S13.   Decrement Segments Left by 1
S14.   Update IPv6 DA with Segment List [Segments Left]
S15.   Push a new IPv6 header with its own SRH containing B, and
        the VTN-ID in VTN option set to V in the HBH Ext header
S16.   Set the outer IPv6 SA to A
S17.   Set the outer IPv6 DA to the first SID of B
S18.   Set the outer Payload Length, Traffic Class, Flow Label,
        Hop Limit, and Next Header fields
S19.   Submit the packet to the egress IPv6 FIB lookup for
        transmission to the new destination
S20. }
```

3.2. End.NRP.Encaps

A new SRv6 function called End.NRP.Encaps is defined. This is a variation of the End behavior. It instructs the endpoint node to determine the corresponding NRP-ID of the local domain based on the mapping relationship between the End.NRP.Encaps SID and the NRPs maintained on the endpoint. The NRP-ID is encapsulated in the VTN option in the IPv6 HBH extension header.

Any SID instance of this behavior is associated with one NRP-ID V and a source address A.

When node N receives a packet whose IPv6 DA is S, and S is a local End.NRP.Encaps SID, N does the following:

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.     Stop processing the SRH, and proceed to process the next
           header in the packet, whose type is identified by
           the Next Header field in the routing header.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address
           with Code 0 (Hop limit exceeded in transit),
           interrupt packet processing, and discard the packet.
S07.   }
S08.   max_LE = (Hdr Ext Len / 2) - 1
S09.   If ((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
S10.     Send an ICMP Parameter Problem to the Source Address
           with Code 0 (Erroneous header field encountered)
           and Pointer set to the Segments Left field,
           interrupt packet processing, and discard the packet.
S11.   }
S12.   Decrement IPv6 Hop Limit by 1
S13.   Decrement Segments Left by 1
S14.   Update IPv6 DA with Segment List [Segments Left]
S15.   Set the VTN-ID in VTN option to V in the HBH Ext header
S16.   Submit the packet to the egress IPv6 FIB lookup for
           transmission to the new destination
S17. }
```

3.3. End.BNRP.Encaps

A new SRv6 function called End.BNRP.Encaps: Endpoint bound to a NRP with IPv6 encapsulation is defined. This is a variation of the End behavior. For the End.BNRP SID, its corresponding NRP-ID should be specified and encapsulated by the ingress node of SRv6 Path. It instructs the endpoint node to obtain the corresponding NRP-ID from the SRH, and encapsulate it in the VTN option in the IPv6 HBH extension header. Through the End.BNRP.Encaps, the ingress node can flexibly specify the local NRP the packet traverses in the network.

Any SID instance of this behavior is associated with one NRP-ID V and a source address A.

There can be several options to carry the local NRP-ID corresponding to the End.BNRP.Encaps function:

1. The NRP-ID is carried in the argument field of the End.BNRP.Encaps SID.
2. The NRP-ID is carried in the SRH TLV field.

3. The NRP-ID is carried in the next SID following the End.BNRP.Encaps SID in the SID list.

Editor's note: In the current version of this document, option 1 is preferred, in which the local NRP-ID is carried in the argument field of the SRv6 SID.

When an ingress node of an SR path encapsulates the End.BNRP.Encaps SID into the packet, it SHOULD put the NRP-ID which the packet is expected to be mapped to into the argument part of the SID.

When node N receives a packet whose IPv6 DA is S, and S is a local End.BNRP.Encaps SID, N does the following:

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.     Stop processing the SRH, and proceed to process the next
        header in the packet, whose type is identified by
        the Next Header field in the routing header.
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address
        with Code 0 (Hop limit exceeded in transit),
        interrupt packet processing, and discard the packet.
S07.   }
S08.   max_LE = (Hdr Ext Len / 2) - 1
S09.   If ((Last Entry > max_LE) or (Segments Left > Last Entry+1)) {
S10.     Send an ICMP Parameter Problem to the Source Address
        with Code 0 (Erroneous header field encountered)
        and Pointer set to the Segments Left field,
        interrupt packet processing, and discard the packet.
S11.   }
S12.   Obtain the NRP-ID V from the argument part of the IPv6 DA
S13.   Decrement IPv6 Hop Limit by 1
S14.   Decrement Segments Left by 1
S15.   Update IPv6 DA with Segment List [Segments Left]
S16.   Set the VTN-ID in VTN option to V in the HBH Ext header
S17.   Submit the packet to the egress IPv6 FIB lookup for
        transmission to the new destination
S18. }
```

4. SR-MPLS NRP BSIDs

[I-D.li-mpls-enhanced-vpn-vtn-id] describes the mechanism of carrying the VTN-ID of a network domain in the MPLS extension header.

With SR-MPLS data plane, NRP BSIDs can be allocated by a domain edge node for the three types of NRP binding behaviors described in section 2.

For the first type of NRP BSID, a BSID can be bound to a list of resource-aware segments of a local NRP. When a node receives a packet with a locally assigned NRP BSID, it determines the corresponding SID list which consists of the resource-aware segments of a local NRP, and encapsulates the SID list to the MPLS label stack.

For another variant of the first type NRP BSID, a NRP BSID is bound to a SR Policy and a local NRP-ID. When a node receives a packet with a locally assigned NRP BSID, it determines the corresponding SID list and the local NRP-ID, and encaps the packet with the SID list and an MPLS VTN extension header which carries the local NRP-ID. Note this requires to assign a NRP BSID for each SR policy in each NRP the node participates in.

For the second type of NRP BSID, a NRP BSID is bound to the shortest path in an NRP of the local network domain. When a node receives a packet with a locally assigned NRP BSID, it determines the corresponding local NRP-ID based on the mapping relationship between the NRP BSID and the NRP-ID, and encapsulates the packet with an MPLS VTN extension header which carries the local NRP-ID. Note this requires to assign a NRP BSID for each local NRP.

For a variant of the second type NRP BSID, a NRP BSID is bound to the shortest path in an NRP of the local network domain, the NRP-ID is specified and encapsulated by the ingress node in the MPLS VTN extension header. When a node receives a packet with a locally assigned NRP BSID, it obtains the corresponding local NRP-ID from the NRP-ID list in the VTN extension header, and update the local NRP-ID in the VTN extension header with the obtained NRP-ID.

5. IANA Considerations

TBD

6. Security Considerations

TBD

7. Acknowledgements

The authors would like to thank Zhibo Hu for his review and valuable comments.

8. References

8.1. Normative References

- [I-D.ietf-teas-enhanced-vpn]
Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Network (VPN+) Services", Work in Progress, Internet-Draft, draft-ietf-teas-enhanced-vpn-10, 6 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-teas-enhanced-vpn-10.txt>>.
- [I-D.ietf-teas-ietf-network-slices]
Farrel, A., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slices-08, 6 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-teas-ietf-network-slices-08.txt>>.
- [I-D.li-teas-e2e-ietf-network-slicing]
Li, Z., Dong, J., Pang, R., and Y. Zhu, "Framework for End-to-End IETF Network Slicing", Work in Progress, Internet-Draft, draft-li-teas-e2e-ietf-network-slicing-02, 7 March 2022, <<https://www.ietf.org/archive/id/draft-li-teas-e2e-ietf-network-slicing-02.txt>>.
- [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>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.

8.2. Informative References

- [I-D.dong-teas-nrp-scalability]
Dong, J., Li, Z., Gong, L., Yang, G., Guichard, J. N., Mishra, G., Qin, F., Saad, T., and V. P. Beeram, "Scalability Considerations for Network Resource Partition", Work in Progress, Internet-Draft, draft-dong-teas-nrp-scalability-01, 7 February 2022, <<https://www.ietf.org/archive/id/draft-dong-teas-nrp-scalability-01.txt>>.

[I-D.ietf-6man-enhanced-vpn-vtn-id]

Dong, J., Li, Z., Xie, C., Ma, C., and G. Mishra,
"Carrying Virtual Transport Network (VTN) Identifier in
IPv6 Extension Header", Work in Progress, Internet-Draft,
draft-ietf-6man-enhanced-vpn-vtn-id-00, 5 March 2022,
<<https://www.ietf.org/archive/id/draft-ietf-6man-enhanced-vpn-vtn-id-00.txt>>.

[I-D.ietf-spring-resource-aware-segments]

Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li,
Z., and F. Clad, "Introducing Resource Awareness to SR
Segments", Work in Progress, Internet-Draft, draft-ietf-
spring-resource-aware-segments-04, 5 March 2022,
<<https://www.ietf.org/archive/id/draft-ietf-spring-resource-aware-segments-04.txt>>.

[I-D.ietf-spring-sr-for-enhanced-vpn]

Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li,
Z., and F. Clad, "Segment Routing based Virtual Transport
Network (VTN) for Enhanced VPN", Work in Progress,
Internet-Draft, draft-ietf-spring-sr-for-enhanced-vpn-02,
5 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-spring-sr-for-enhanced-vpn-02.txt>>.

[I-D.li-mpls-enhanced-vpn-vtn-id]

Li, Z. and J. Dong, "Carrying Virtual Transport Network
Identifier in MPLS Packet", Work in Progress, Internet-
Draft, draft-li-mpls-enhanced-vpn-vtn-id-02, 7 March 2022,
<<https://www.ietf.org/archive/id/draft-li-mpls-enhanced-vpn-vtn-id-02.txt>>.

Authors' Addresses

Zhenbin Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Road
Beijing
100095
China
Email: lizhenbin@huawei.com

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

Email: jie.dong@huawei.com

Ran Pang
China Unicom
Email: pangran@chinaunicom.cn

Yongqing Zhu
China Telecom
Email: zhuyq8@chinatelecom.cn

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 28, 2022

Z. Li
Z. Hu
J. Dong
Huawei Technologies
October 25, 2021

Intent-based Routing
draft-li-teas-intent-based-routing-00

Abstract

This document defines the intent-based routing mechanism through which the packet can carry the intent information and the network node can enforce the policy according to the intent information (typically steering the packet into the SR policy or the underlay slice which can meet the intent). The intent-based routing mechanism provides a simple and scalable solution to meet the different service requirements for the inter-domain routing.

Requirements Language

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

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <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 28, 2022.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Terminologies	3
3. Intent-based Routing	3
4. Illustration	5
5. IPv6 Encapsulation	8
6. Security Considerations	9
7. IANA Considerations	9
8. References	9
8.1. Normative References	9
8.2. Informative References	10
Authors' Addresses	11

1. Introduction

[I-D.hegde-spring-mpls-seamless-sr] describes the requirements for end-to-end intent-based paths spanning multi-domain networks. [I-D.kaliraj-idr-bgp-classful-transport-planes] specifies the BGP based mechanisms to signal the packet paths which span multiple domains and provide different SLA characteristics. Since these SR paths need to setup according to the pair <color, endpoint>, it means more SR paths are introduced and this will cause more challenges on scalability.

In order to reduce the challenge of scalability introduced by the inter-domain routing with different service requirements, this document proposes the intent-based routing mechanism through which the packet can carry the intent information and the network node can steer the packet into the SR policy to satisfy the service requirement (that is, meet the specific intent). With the intent-based routing mechanism, network nodes do not need to maintain the fine-granularity connection state for each destination in the control plane, which can improve the scalability of the end- to-end routing significantly.

Besides steering the packet into the SR policy, the intent-based routing mechanism can also be used to steer the traffic into the

underlay network slice to meet the specific intent or enforce policy for other intents such as network measurement, security, etc. Since the same intent can be satisfied by different solutions in the different network domain, the intent-based routing also improve the flexibility to satisfying the service requirement through the combined solutions for the same intent.

2. Terminologies

The following terminologies are used in this document.

SR: Segment Routing

SRv6: Segment Routing over IPv6

3. Intent-based Routing

The Intent-based routing mechanism introduces the concept of intent as the information carried in the data plane to represent the specific service requirement for the destination on the network. The intent can be associated with a series of service attributes, such as low latency and high bandwidth. The value can be allocated by the administrator. The allocation of values of the intent in the multiple domain must be consistent.

[I-D.ietf-spring-segment-routing-policy] defines the color used for the SR policy. The color is a 32-bit numerical value that associates the SR Policy with an intent (e.g. low-latency). There can be the mapping as follows between the color and the intent. If the intent and the color can be designed and allocated consistently, the value of the color can be the same as that of the intent and the mapping between the color and the intent can be saved in the data plane.

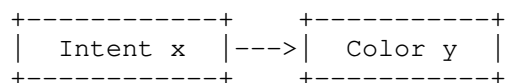


Figure 1 Mapping between Intent and Color

Figure 1: Figure 1: Reference Topology

In the scenario of the inter-domain routing, the SR policy group for a specific Endpoint shown in the Figure 2 can be set up in the data plane in the local network domain. That is, it is not necessary to advertise the pair <color, endpoint> to set up the end-to-end SR path. When the packet carrying the intent information arrives at the

edge node of the network domain, the edge node can search the SR policy group according to the destination, then steer the packet into the corresponding SR policy according to the mapping between the color and the intent and the mapping between the color and the SR policy.

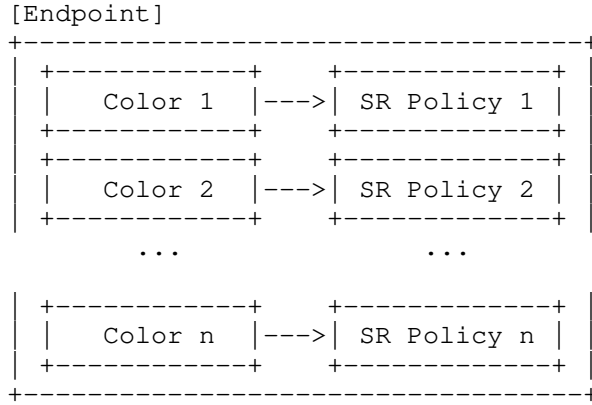


Figure 2: Figure 2: SR Policy Group

In the scenario of the inter-domain network slicing, the following mapping between the color and the local underlay network slice can be set up in the data plane in the local network domain. When the packet carrying the intent information arrives at the edge node of the network domain, the edge node can steer the packet into the local underlay network slice according to the mapping between the color and the intent and the mapping between the color and the local underlay network slice.

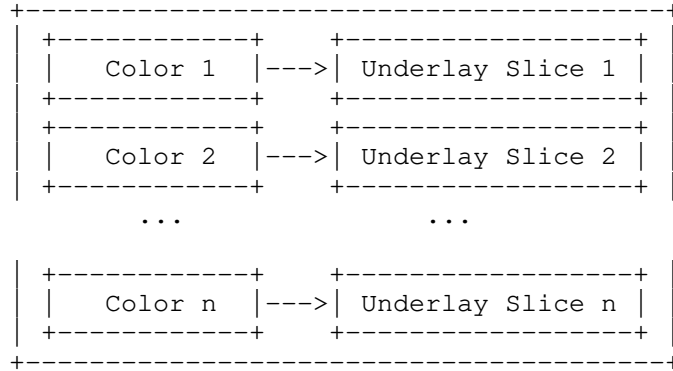


Figure 3: Figure 3: Mapping between Color and Underlay Network Slice

Since the same Intent may be satisfied by the SR policy or the underlay network slice, the local network domain can choose the different solutions flexibly without the need of coordination with other network domains. This can also improve the flexibility of the inter-domain routing.

Besides steering the packet into the SR policy or the underlay network slice, the network node can also enforce the policy for other possible intents such as network measurement, security, etc. This will be defined in the future version of the draft.

4. Illustration

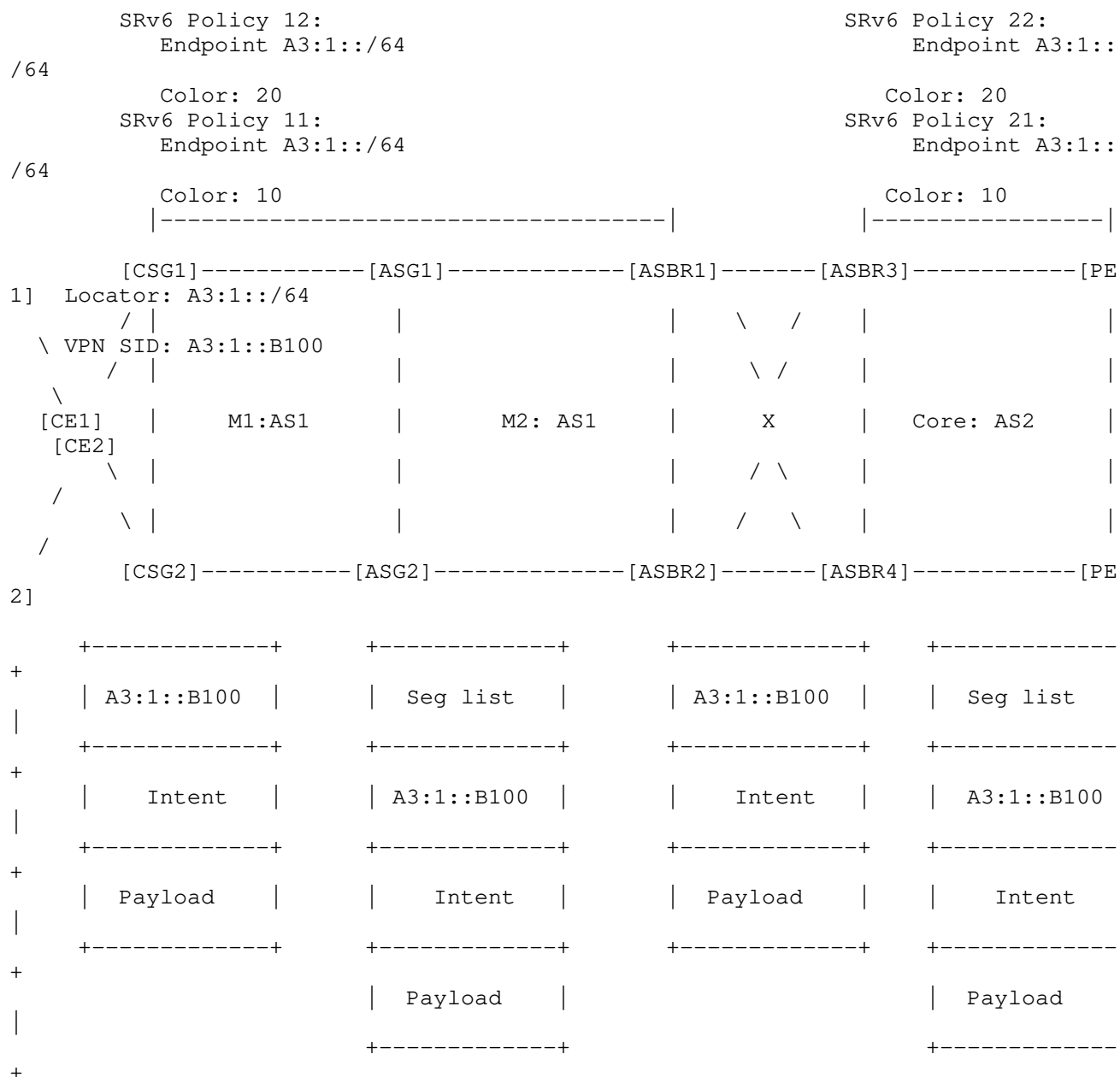


Figure 4: Figure 4: Illustration of Intent-based Inter-domain Routing

Figure 4 shows an example of a service provider network that comprises of two Autonomous systems, AS1 and AS2. The customer requests a leased line that requires bandwidth guarantee from CSG1 to PE1. Assume that the following is applied in the network shown in the Figure 4:

- o Independent ISIS instance in core (C) region.
- o Independent ISIS instance in Metro1 (M1) region.
- o Independent ISIS instance in Metro2 (M2) region.

- o BGP between ASBRs
- o PE1's locator is A3:1::/64, and VPN SID is A3:1::B100.
- o Core's aggregated routes are redistributed from Core to M (M1 and M2).

- o SRv6 policy group is set up in the AS1 between the CSG1 and ASBR1. It includes two SRv6 policies with the same Endpoint A3:1::/64 and color 10 and 20 respectively.
- o SRv6 policy group is set up in the AS2 between the ASBR3 and PE1. It includes two SRv6 policies with the same Endpoint A3:1::/64 and color 10 and 20 respectively.

PE1 advertises the VPN route with color 10 to CSG1. After CSG1 receive the VPN route, it maps color to the Intent and installs the VPN route with VPN SID A3:1::B100 and the corresponding intent. When CSG1 receives a packet from CE1, assume that CE1 finds the VPN route and the forwarding process is as follows:

1. CE1 encapsulates a new IPv6 header to the packet with the destination IPv6 address set as VPN SID A3:1::B100 and the Intent in the packet.
2. CE1 can search the forwarding entry according to the destination IPv6 address A3:1::B100 and the Intent.
3. After CE1 finds the SRv6 Policy 11 with the color 10, it encapsulates the new IPv6 header with the corresponding segment list to the packet.
4. The packet is forwarded to ASBR1 and the segment list is decapsulated at ASBR1.
5. ASBR1 can send the packet to ASBR3 according to the destination address A3:1::B100 by IPv6 forwarding process.
6. ASBR3 searches the forwarding entry according to the destination IP address A3:1::B100 and the Intent.
7. ASBR3 finds the SRv6 policy 21 with the color 10 and encapsulates the new IPv6 header with the corresponding segment list to the packet.
8. The packet is forwarded to PE1 and the segment list is decapsulated at PE1.
9. The packet is forwarding in the corresponding VPN instance identified by the destination IPv6 address A3:1::B100.

5. IPv6 Encapsulation

The intent can be encapsulated in the different data plane. This document firstly define the IPv6 encapsulation for the intent.

In order to support the intent-based routing, one new option, the Intent option, is defined.

The Intent option has the following format:

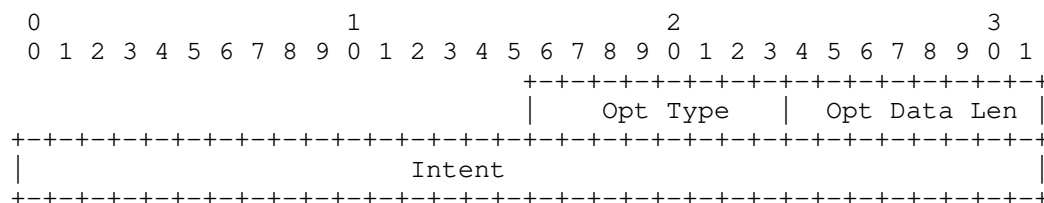


Figure 5. Intent Option

where:

- o Opt Type: Type value is TBD. 8-bit unsigned integer. Identifier of the type of this Intent Option.
- o Opt Data Len: 8-bit unsigned integer. Length of the Option Data field of this option, that is, length of the Intent.
- o Option Data: Option-Type-specific data. It carries the Intent. A 32-bit identifier.

The Intent option can be placed in several locations in the IPv6 packet header depending upon the scenarios and implementation requirements.

1. Hop-by-Hop Options Header (HBH)

The Intent option can be carried in the Hop-by-Hop Options Header as the new option. By using the HBH Options Header, the intent information carried can be read by every node along the path.

2. Destination Options Header (DOH)

The Intent option can be carried in the Destination Options Header as the new option. By using the DOH Options Header, the intent

information carried can be read by the destination node along the path.

Besides the Intent option, the intent can also be carried combining with Application-aware Networking ([I-D.li-apn-framework]). [I-D.li-apn-header] and [I-D.li-apn-ipv6-encap] defines that the intent can be carried in the APN header which is encapsulated in the APN option in the IPv6 data plane.

6. Security Considerations

TBD

7. IANA Considerations

TBD

8. References

8.1. 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>>.
- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <<https://www.rfc-editor.org/info/rfc7356>>.
- [RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <<https://www.rfc-editor.org/info/rfc7490>>.
- [RFC8400] Chen, H., Liu, A., Saad, T., Xu, F., and L. Huang, "Extensions to RSVP-TE for Label Switched Path (LSP) Egress Protection", RFC 8400, DOI 10.17487/RFC8400, June 2018, <<https://www.rfc-editor.org/info/rfc8400>>.
- [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>>.

- [RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <<https://www.rfc-editor.org/info/rfc8665>>.
- [RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <<https://www.rfc-editor.org/info/rfc8667>>.
- [RFC8679] Shen, Y., Jeganathan, M., Decraene, B., Gredler, H., Michel, C., and H. Chen, "MPLS Egress Protection Framework", RFC 8679, DOI 10.17487/RFC8679, December 2019, <<https://www.rfc-editor.org/info/rfc8679>>.

8.2. Informative References

- [I-D.hegde-spring-mpls-seamless-sr]
Hegde, S., Bowers, C., Xu, X., Gulko, A., Bogdanov, A., Uttaro, J., Jalil, L., Khaddam, M., Alston, A., and L. M. Contreras, "Seamless SR Problem Statement", draft-hegde-spring-mpls-seamless-sr-06 (work in progress), September 2021.
- [I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-14 (work in progress), October 2021.
- [I-D.kaliraj-idr-bgp-classful-transport-planes]
Vairavakkalai, K., Venkataraman, N., Rajagopalan, B., Mishra, G., Khaddam, M., Xu, X., Szarecki, R. J., and D. J. Gowda, "BGP Classful Transport Planes", draft-kaliraj-idr-bgp-classful-transport-planes-12 (work in progress), August 2021.
- [I-D.li-apn-framework]
Li, Z., Peng, S., Voyer, D., Li, C., Liu, P., Cao, C., Mishra, G., Ebisawa, K., Previdi, S., and J. N. Guichard, "Application-aware Networking (APN) Framework", draft-li-apn-framework-03 (work in progress), May 2021.

[I-D.li-apn-header]

Li, Z. and S. Peng, "Application-aware Networking (APN) Header", draft-li-apn-header-00 (work in progress), October 2021.

[I-D.li-apn-ipv6-encap]

Li, Z. and S. Peng, "Application-aware IPv6 Networking (APN6) Encapsulation", draft-li-apn-ipv6-encap-00 (work in progress), October 2021.

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

[RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, DOI 10.17487/RFC5462, February 2009, <<https://www.rfc-editor.org/info/rfc5462>>.

Authors' Addresses

Zhenbin Li
Huawei Technologies
Beijing 100095
China

Email: lizhenbin@huawei.com

Zhibo Hu
Huawei Technologies
Beijing 100095
China

Email: huzhibo@huawei.com

Jie Dong
Huawei Technologies
Beijing 100095
China

Email: jie.dong@huawei.com

SPRING Working Group
Internet-Draft
Intended status: Standards Track
Expires: 3 September 2022

K. Salih
S. Hegde
M. Rajesh
R. Bonica
Juniper Networks
H. wang
Huawei Technologies
Shaofu. Peng
ZTE Corporation
2 March 2022

SRv6 inter-domain mapping SIDs
draft-salih-spring-srv6-inter-domain-sids-02

Abstract

This document describes three new SRv6 end-point behaviors, called END.REPLACE, END.REPLACEB6 and END.DB6. These behaviors are used in distributed inter-domain solutions and are normally executed on border routers. They also can be used to provide multiple intent-based paths across these domains.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <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 3 September 2022.

Copyright Notice

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

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

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

Table of Contents

1. Overview	2
2. Requirements Language	2
3. Usecases	3
3.1. usecase 1	3
3.2. usecase 2	3
4. SRv6 SID Behaviors	4
4.1. END.REPLACE	4
4.2. END.REPLACEB6	4
4.3. END.DB6	5
5. Interworking Procedures	6
5.1. Option C Transport Interworking	6
5.2. Option B service interworking	9
6. IANA Considerations	10
7. Security Considerations	10
8. Contributors	10
9. Acknowledgements	11
10. References	11
10.1. Normative References	11
10.2. Informative References	11
Authors' Addresses	12

1. Overview

Segment Routing (SR) [RFC8402] allows source nodes to steer packets through SR paths. It can be implemented over IPv6 [RFC8200] or MPLS [RFC3031]. When SR is implemented over IPv6, it is called SRv6 [RFC8986].

This document describes three new SRv6 end-point behaviors, called END.REPLACE, END.REPLACEB6 and END.DB6. These behaviors are used to build paths across SRv6 domains. They also facilitate end-to-end SRv6 intent-based path stitching.

2. Requirements Language

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

3. Usecases

3.1. usecase 1

This use-case is mentioned in Section 4.1.1 of [I-D.hegde-spring-mpls-seamless-sr].

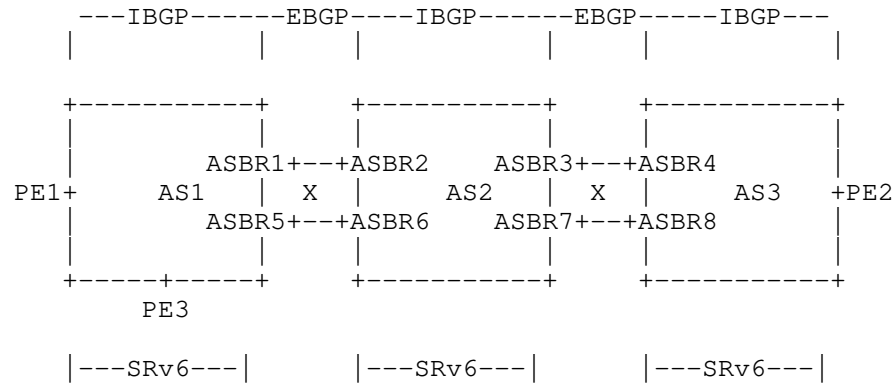


Figure 1: Multiple ASes connected with E-BGP

Figure 1 depicts three ASes (AS1, AS2 and AS3). All the three domains deploy SRv6. Inter-provider Option C[RFC4364] connectivity is maintained from PE1 to PE2.

3.2. usecase 2

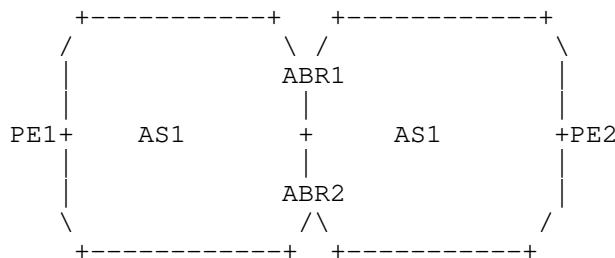


Figure 2: Single AS with different IGP domains

The above diagram Figure 2 shows two different SRv6 IGP domains. Services are running between PE1 and PE2 in option B [RFC4364] style. The requirement here is to avoid service route lookup on ABR1 and ABR2 to provide option B style end to end connectivity

4. SRv6 SID Behaviors

4.1. END.REPLACE

The END.REPLACE behavior is applicable in the Multiple ASes Connected With E-BGP (Section 3.1) use-case.

The End.REPLACE SID cannot be the last segment in SRH or SR Policy.

Any SID instance of this behavior is associated with a set, J, of one or more L3 adjacencies of immediate BGP neighbors

When Node N receives a packet destined to S and S is a locally instantiated End.REPLACE SID, Node N executes the following procedure:

```
S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.     Stop processing the SRH, and proceed to process the next
        header in the packet, whose type is identified by
        the Next Header field in the routing header. Procedure is as
        per Section 4.1.1 of [RFC8986].
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address with Code 0
        (Hop limit exceeded in transit), interrupt packet processing, and di
scard packet
S07.   }
S08.   Decrement IPv6 Hop Limit by 1
S09.   Update IPv6 DA with new destination address(SID) mapped with END.REPLAC
E SID.
S10.   Submit the packet to the IPv6 module for transmission
        to the new destination via a member of J.
S11. }
```

4.2. END.REPLACEB6

The END.REPLACEB6 behavior is applicable in the Multiple ASes Connected With E-BGP (Section 3.1) use-case.

The End.REPLACEB6 SID cannot be the last segment in a SRH or SR Policy.

Node N is configured with an IPv6 address T (e.g., assigned to its loopback).

When Node N receives a packet destined to S and S is a locally instantiated End.REPLACEB6 SID, Node N executes the following procedure:

```

S01. When an SRH is processed {
S02.   If (Segments Left == 0) {
S03.     Stop processing the SRH, and proceed to process the next
        header in the packet, whose type is identified by
        the Next Header field in the routing header. Procedure is as
        per Section 4.1.1 of [RFC8986].
S04.   }
S05.   If (IPv6 Hop Limit <= 1) {
S06.     Send an ICMP Time Exceeded message to the Source Address with Code 0
        (Hop limit exceeded in transit), interrupt packet processing, and di
scard packet
S07.   }
S08.   Decrement IPv6 Hop Limit by 1
S09.   Update IPv6 DA with new destination address(SID) mapped with END.REPLAC
EB6.
S10.   Push an IPv6 header with an SRH.
S11.   Set outer IPv6 SA = T and outer IPv6 DA to the first SID in the segment
list
S12.   Set outer Payload Length, Traffic Class, Hop Limit, and Flow Label fiel
ds
S13.   Set the outer Next Header value
S14.   Submit the packet to the IPv6 module for transmission to the First SID.
S15. }
```

Note :

S10 - S13. Implementation may choose to avoid outer encapsulation for flex-algo and best effort based SRv6 transport tunnels.

S12. The Payload Length, Traffic Class, Hop Limit, and Next Header fields are set as per [RFC2473]. The Flow Label is computed as per [RFC6437].

4.3. END.DB6

For the use-case mentioned under Section 3.2 END.DB6 SID is applicable.

The End.DB6 SID MUST be the last segment in SRH or SR Policy.

Node N is configured with an IPv6 address T (e.g., assigned to its loopback).

When Node N receives a packet destined to S and S is a locally instantiated End.DB6 SID, Node N executes the following procedure:


```

S01. When an SRH is processed {
S02.   If (Segments Left != 0) {
S03.     Send an ICMP Parameter Problem to the Source Address,
        Code 0 (Erroneous header field encountered),
        Pointer set to the Segments Left field,
        interrupt packet processing and discard the packet.
S04.   }
S05.   If (Upper-Layer header type == 4(IPv4) OR Upper-Layer header type == 4
1(IPv6) OR
        Upper-Layer header type == 143(Ethernet)) {
S06.     Remove the outer IPv6 header with all its extension headers.
S07.     Push the new IPv6 header with the SRv6 SIDs associated with the END.D
B6 sid in an SRH.
S08.     Set outer IPv6 SA = T and outer IPv6 DA to the first SID in the segme
nt list.
S09.     Set outer Payload Length, Traffic Class, Hop Limit, and Flow Label fi
elds
S10.     Set the outer Next Header value
S11.     Submit the packet to the IPv6 module for transmission to First SID.
S12.   } else {
S13.     Process as per Section 4.1.1 of [RFC8986].
S14.   }
S15. }

```

Note :

S09. The Payload Length, Traffic Class, Hop Limit, and Next Header fields are set as per [RFC2473]. The Flow Label is computed as per [RFC6437].

5. Interworking Procedures

Here we will describe the control plane and data plane procedures by taking examples.

Node n has a classic IPv6 loopback address An::<1/128. One of the SID at node n with locator block B and function F is represented by B:n:F::sid_num.

A SID list is represented as

<S1, S2, S3>

where S1 is the first SID to visit, S2 is the second SID to visit and S3 is the last SID to visit along the SR path.

5.1. Option C Transport Interworking

Here we will discuss the use-case mentioned under Section 3.1

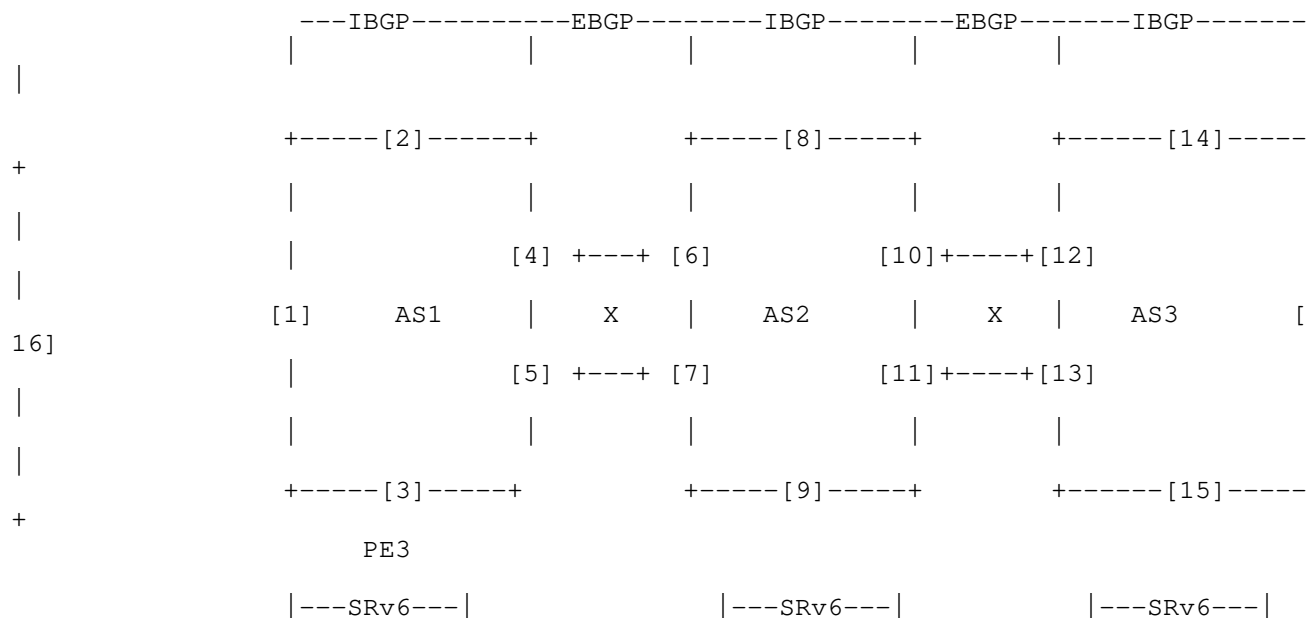


Figure 3: Option C Style Interworking

Node [1] acts as ingress PE and Node [16] acts as egress PE.

Nodes [2], [3], [8], [9], [14] and [15] are P routers.

Nodes [4], [5], [6], [7], [10], [11], [12] and [13] are ASBR routers.

A VPN route is advertised via service RRs between an egress PE (node 16) and an ingress PE (node 1). The example below shows IBGP-CT connection between border routers in each domain and single hop EBGP-CT for inter-domain connections. However the forwarding procedure for the sids remains the same irrespective of the various inter-domain protocol extensions used to advertise the sids. AS1, AS2 and AS3 has SRTE policy for the required intent paths.

Control plane example:

For simplicity only one path is tracked.

For a route if the next hop is one hop away then while advertising use END.REPLACE SID. For a route if the next hop is multi hop away then while advertising use END.REPLACEB6 SID. For single hop neighbor case, no encapsulation required as it is just replace and forward on specific link while in multihop case one encapsulation will be required.

Routing Protocol(RP) @16:
 * In ISIS advertise locator B:16::/48 and an END SID B:16::END::1.
 * BGP AFI=1,SAFI=128 originates a VPN route RD:V/v via A:16::1 and Prefix-SID attribute B:16:DT4::1.
 This route is advertised to service RR with color extended community red.
 * BGP originates prefix A:16::1 with color red to ASBR [12] with SRv6 SID B:16:END::1 since its the egress node.
RP @12:
 * BGP receives the route A:16::1 over the ibgp session and readvertises with nexthop self to ASBR [10].
 it advertises the SRv6 SID B:12:REPLACEB6::1 in the protocol extensions. As the advertisement was received on a multihop i-bgp session this node allocates a REPLACEB6 sid.
RP @10:
 * BGP receives the route A:16::1 over the ebgp session and readvertises with nexthop self to ASBR [6].
 it advertises the SRv6 SID B:10:REPLACE::1 in the protocol extensions. As the advertisement was received on a single hop e-bgp session this node allocates a REPLACE sid.
RP @6:
 * BGP receives the route A:16::1 over the ibgp session and readvertises with nexthop self to ASBR [4].
 it advertises the SRv6 SID B:6:REPLACEB6::1 in the protocol extensions. As the advertisement was received on a multihop i-bgp session this node allocates a REPLACEB6 sid.
RP @4:
 * BGP receives the route A:16::1 over the ebgp session and readvertises with nexthop self to PE [1].
 it advertises the SRv6 SID B:4:REPLACE::1 in the protocol extensions. As the advertisement was received on a single hop e-bgp session this node allocates a REPLACE sid.
RP @1:
 * BGP receives the route A:16::1 with color red over the ibgp session.
 * BGP AFI=1, SAFI=128 learn service prefix RD:V/v, next hop A:16::1 and PrefixSID attribute TLV type 5
 with SRv6 SID B:16:DT4

FIB State:

```

    @1: IPv4 VRF V/v => H.Encaps.red <B:2:END::1, B:4:REPLACE::1, B:16:DT4::1
> with SRH, SRH NextHeader=IPv4 where the first
    sid B:2:END::1 belongs to the SR-policy in AS1.
    @2: IPv6 Table: B:2:END::1 => Update DA with B:4:REPLACE::1, decrement SL
and forward towards the ASBR [4].
    @4: IPv6 Table: B:4:REPLACE::1 => Update DA with B:6:REPLACEB6::1 and for
ward on the interface/interfaces identified by the
    ebgp neighbor; the SL remains at 1.
    @6: IPv6 Table: B:6:REPLACEB6::1 => Update DA with B:10:REPLACE::1 AND do
a fresh H.Encaps.red <B:8:END::1, B:10:END::1>
    with SRH where the new SRH SIDs belong to SR policy in AS2.
    @8: IPv6 Table: B:8:END::1 => Update outer IPv6 packet DA with B:10:END::
1 and forward towards ASBR [10]
    @10: IPv6 table: B:10:END::1 => Decap Outer IPv6 header and lookup next I
Pv6 DA B:10:REPLACE::1 => Update DA to B:12:REPLACEB6::1
    and forward on the interface/interfaces identified by the ebgp neighb
our. SL remains at 1.
    @12: IPv6 Table B:12:REPLACEB6::1 => Update DA with B:16:END::1 and do a
fresh H.Encaps.red <B:15:END::1, B:16:END::1> with SRH
    where the new SIDs belong to the SR policy in AS3.
    @15: IPv6 Table B:15:END::1 => Update outer IPv6 packet DA with B:16:END:
:1 and forward towards [16].
    @16: IPv6 Table B:16:END::1 => Decap the outer header and lookup the inne
r DA which results in B:16:DT4::1 lookup. DT4 lookup
    results in Decap and inner IPv4 packet DA lookup in the correspondin
g VRF.

```

Note: At [16] its possible to optimize the lookups required with minor control plane extensions.

5.2. Option B service interworking

Here we will discuss the use-case mentioned under Section 3.2

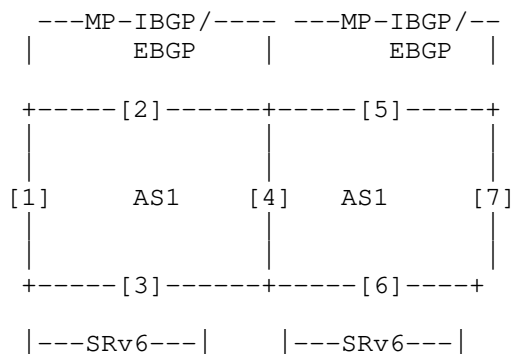


Figure 4: Option B style Service Interworking

Nodes [1] and [7] are PE routers. Node [4] is an option B style configured ABR/RR.

Control Plane example:

Routing Protocol (RP) @7:
* BGP AFI=1,SAFI=128 originates a VPN route RD:V/v via A:7::1 and Prefix-SID attribute B:7:DT4::1. This route is advertised to service RR [4].

RP @4:
* BGP receives the route over MP-IBGP/MP-EBGP session and readvertises with next hop self to PE [1].
it advertises the SRv6 SID B:4:DB6::1 in the Prefix-SID attribute TLV along with it. For all prefixes having SRv6 service SID B:7:DT4::1; the same DB6 SID B:4:DB6::1 will be reused. if a different service sid B:7:DT4::2 comes then a different DB6 SID B:4:DB6::2 will be allocated. This ensures that if the egress allocates per CE sid; the translation at border also ensure per CE sid.

RP @1:
* BGP AFI=1, SAFI=128 learn service prefix RD:V/v, next hop A:4::1 and PrefixSID attribute TLV type 5 with SRv6 SID B:4:DB6::1

FIB State:

@1: IPv4 VRF V/v => H.Encaps.red <B:4:DB6::1> with SRH, SRH NextHeader=IPv4 where the first sid belongs to the SR-policy in AS1
@4: IPv6 Table: B:4:DB6::1 => Decapsulate the incoming IPv6 header and H.Encaps <B:7:DT4::1>
@7: IPv6 Table: B:7:DT4::1 => Decapsulate the header and lookup the inner IPv4 packet DA in the VRF

6. IANA Considerations

This document requires no IANA action.

The authors will request an early allocation from the "SRv6 Endpoint Behaviors" sub-registry of the "Segment Routing Parameters" registry.

7. Security Considerations

Because SR inter-working requires co-operation between inter-working domains, this document introduces no security consideration beyond those addressed in [RFC8402], [RFC8754] and [RFC8986].

8. Contributors

Jie Dong
Huawei Technologies
Email: jie.dong@huawei.com

Swamy SRK
Juniper Networks
Email: swamys@juniper.net

G. Sri Karthik Goud
Juniper Networks
Email: gkarthik@juniper.net

9. Acknowledgements

Thanks to Ram Santhanakrishnan, Srihari Sangli, Rajendra Prasad Bollam and Kiran Kushalad for their 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>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [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>>.
- [RFC8402] Filts, 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] Filts, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.
- [RFC8986] Filts, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.

10.2. Informative References

- [I-D.hegde-spring-mpls-seamless-sr] Hegde, S., Bowers, C., Xu, X., Gulko, A., Bogdanov, A., Uttaro, J., Jalil, L., Khaddam, M., Alston, A., and L. M.

Contreras, "Seamless SR Problem Statement", Work in Progress, Internet-Draft, draft-hegde-spring-mpls-seamless-sr-06, 24 September 2021, <<https://www.ietf.org/archive/id/draft-hegde-spring-mpls-seamless-sr-06.txt>>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.

Authors' Addresses

Salih K A
Juniper Networks
Embassy Business Park
Bangalore 560093
KA
India
Email: salih@juniper.net

Shraddha Hegde
Juniper Networks
Embassy Business Park
Bangalore 560093
KA
India
Email: shraddha@juniper.net

Rajesh
Juniper Networks
Embassy Business Park
Bangalore 560093
KA
India
Email: mrjesh@juniper.net

Ron Bonica
Juniper Networks
Herndon, Virginia 20171
United States of America
Email: rbonica@juniper.net

Haibo Wang
Huawei Technologies
Huawei Campus, No. 156 Beiqing Road
Beijing
100095
China
Email: rainsword.wang@huawei.com

Peng Shaofu
ZTE Corporation
China
Email: peng.shaofu@zte.com.cn

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 28 April 2022

J. Xie
X. Geng
Huawei Technologies
Y. Liu
China Mobile
25 October 2021

Source Segment for Multicast Source Routing over IPv6
draft-xl-msr6-source-segment-00

Abstract

This document defines the general concept of source segment which is used as the IPv6 source address in an IPv6 packet. Source segment for multicast service is introduced in this document.

Requirements Language

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

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <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 28 April 2022.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	2
2. Terminologies	3
3. Source Segment Definition	3
4. SID Format	4
5. Source Segment for MVPN	4
5.1. Behaviors	4
5.2. SRC.DT4	5
5.3. SRC.DT6	5
5.4. SRC.DT46	5
5.5. Src.DT2	6
6. Use Case	6
7. IANA Considerations	7
8. Security Considerations	7
9. References	7
9.1. Normative References	7
9.2. Informative References	8
Authors' Addresses	10

1. Introduction

Segment Routing ([RFC8402]) leverages the mechanism of source routing. An ingress node steers a packet through an ordered list of instructions, called "segments". Each one of these instructions represents a function to be implemented at a specific location in the network. A function is locally defined on the node where it is executed. Network Programming combines Segment Routing functions to achieve a networking objective that goes beyond mere packet routing. [RFC8986] defines the SRv6 Network Programming concept and specifies the main Segment Routing behaviors and network programming functions.

Previous segments defined in SRv6 can be used as the destination address of an IPv6 packet. This document introduces the new segments, source segments, which can be used as the IPv6 source address of an IPv6 packet. This document defines the general concept of source segment and the source segment used for multicast service. Protocol extensions on the control plane are not in the scope of this document.

This document defines the general concept of source segment and the source segment used for multicast service. Protocol extensions on the control plane are not in the scope of this document.

2. Terminologies

The following new terms are used throughout this document:

MSR6: Multicast Source Routing over IPv6;

MSR6 Domain: a set of nodes participating in the multicast source routing;

3. Source Segment Definition

Source segment is different from the existing SID defined in RFC8402 from the following aspects:

- * Source segment is unchanged along the SRv6 path
- * Source segment is distributed by the ingress node but indicates functions in other nodes along the path, e.g., egress node. Forwarding table should be maintained in the nodes where the instruction takes place.
- * When the source segment is encapsulated in an SRv6 packet, it is activated by other instructions in the data plane because source address is not parsed in existing forwarding process of a unicast packet

Using source segment for SRv6 Network Programming have several benefits including:

- * Enhance network programming capability for more SRv6 functions and extend the programming space in IPv6 header;
- * Provide semantic for source address with similar IPv6 address allocation and management method as SRv6;
- * Facilitates security management inside the limited domain;

Source segment should be avoided to process hop by hop. Per-hop process of source segment which will degrade forwarding performance and bring compatibility issues.

4. SID Format

Source segment leverages the format of SID defined in SRv6 network programming.

Source segment consists of LOC:FUNCT:ARG, where a locator (LOC) is encoded in the L most significant bits of the SID, followed by F bits of function (FUNCT) and A bits of arguments (ARG).

A locator may be represented as B:N where B is the SRv6 SID block (IPv6 prefix allocated for SRv6 SIDs by the operator) and N is the identifier of the ingress node .

The FUNCT is an opaque identification of the behavior bound to the SID. The behavior could be executed in other nodes except ingress node.

The behavior indicated by FUNCT may require additional information for its processing. This information may be encoded in the ARG bits of the SID.

5. Source Segment for MVPN

In the multicast service, packet is replicated along the tree towards a set of leaf nodes. MVPN routing and the corresponding information could be encapsulated in the source segment carried in the IPv6 source address. Source Segment for MVPN is distributed by the multicast source node and the function is executed by the multicast leaf nodes. As described in section 3, Source Segment for MVPN is not changed when the packet is replicated and forwarded along the P2MP path.

This section defines the source segment for MVPN.

5.1. Behaviors

The following is a set of behaviors that can be associated with a source segment for MVPN.

Src.DT4	Source address for decapsulation and IPv4 table lookup
Src.DT6	Source address for decapsulation and IPv6 table lookup
Src.DT46	Source address for decapsulation and IP table lookup
Src.DT2	Source address for decapsulation and L2 table lookup

5.2. SRC.DT4

The "Source address for decapsulation and IPv4 table lookup" behavior ("Src.DT4" for short) is used in MVPNv4 use case where an MFIB lookup in a specific VRF table T at the egress node is required. The Src.DT4 SID is an SID associated with an IPv4 MFIB table T on the egress PE, either through a control-plane message advertised by the ingress PE, or through a local configuration on the egress PE. When an IPv6 encapsulated packet with IPv6 source address being S is received on an egress PE, and S is associated with an Src.DT4 SID on the egress PE, the egress PE does the following behavior:

```
S01. If (Upper-Layer header type == 4(IPv4) ) {
S02.   Remove the outer IPv4 header with all its extension headers
S03.   Set the packet's associated MFIB table to T
S04.   Submit the packet to the egress IPv4 MFIB lookup for
       transmission to the new multicast downstreams
S05. } Else {
S06.   Drop the packet;
S07. }
```

5.3. SRC.DT6

SRC.DT6 behavior could be used in MVPNv6 use case where a MFIB lookup in a specific VRF table at the egress node is required.

```
S01. If (Upper-Layer header type == 41(IPv6) ) {
S02.   Remove the outer IPv6 header with all its extension headers
S03.   Set the packet's associated IPv6 MFIB table to T
S04.   Submit the packet to the egress IPv6 MFIB lookup for
       transmission to the new multicast downstreams
S05. } Else {
S06.   Drop the packet;
S07. }
```

5.4. SRC.DT46

SRC.DT46 behavior could be used in MVPN use case where a MFIB lookup in a specific VRF table at the egress node is required.

```
S01. If (Upper-Layer header type == 4(IPv4) ) {
S02.   Remove the outer IPv4 header with all its extension headers
S03.   Set the packet's associated MFIB table to T
S04.   Submit the packet to the egress IPv4 MFIB lookup for
        transmission to the new destination
S05. } Else if (Upper-Layer header type == 41(IPv6) ) {
S06.   Remove the outer IPv6 header with all its extension headers
S07.   Set the packet's associated MFIB table to T
S08.   Submit the packet to the egress IPv6 MFIB lookup for
        transmission to the new destination
S09. } Else {
S10.   Drop the packet;
S11. }
```

5.5. Src.DT2

SRC.DT2 behavior could be used in MVPN use case where a L2 table lookup in a specific Layer-2 Multicast forwarding table at the egress node is required.

```
S01. If (Upper-Layer header type == 143(Ethernet) ) {
S02.   Remove the outer IPv6 header with all its extension headers
S03.   Set the packet's associated Layer-2 Multicast forwarding table to T
S04.   Submit the packet to the egress Layer-2 Multicast forwarding table
        lookup for transmission to the new multicast downstreams
S05. } Else {
S06.   Send an ICMP Parameter Problem to the Source Address
        with Code 4 (SR Upper-layer Header Error)
        and Pointer set to the offset of the Upper-Layer header,
        interrupt packet processing, and discard the packet
S07. }
```

6. Use Case

The source segment could be applied in the following case:

1. MSR6: The MSR6 MVPN uses the source segment in the IPv6 source address for identifying a VRF in IPv6 multicast source routing.
2. Tree SID over SRv6: MVPN service can use Tree SID over SRv6 [I-D.ietf-bess-mvpn-evpn-sr-p2mp] for point-to-multipoint transport of a packet. When a Tree SID over SRv6 P-tunnel is shared across different MVPNs, an IPv6 address in IPv6 source address for identifying a VRF is possible.

3. MVPN service can use Ingress Replication(IR) [RFC6513] to simulate a point-to-multipoint P-tunnel. In an IPv6 environment, Ingress Replication can use IPv6 encapsulation for each branch. When the egress PE of an Ingress Replication P-tunnel branch receives a packet, it gets to know the VRF of the packet through the Destination address in the IPv6 header. This means that, every egress PE of the IR P-tunnel branch need to allocate an IPv6 address to identify a VRF. If the source segment is used for the IPv6 source address, only one IPv6 address of the Ingress PE is needed for identifying a VRF, and thus save the IPv6 addresses and their operation costs.

7. IANA Considerations

TBD

8. Security Considerations

TBD

9. References

9.1. Normative References

- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <<https://www.rfc-editor.org/info/rfc6513>>.
- [RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <<https://www.rfc-editor.org/info/rfc6514>>.
- [RFC6515] Aggarwal, R. and E. Rosen, "IPv4 and IPv6 Infrastructure Addresses in BGP Updates for Multicast VPN", RFC 6515, DOI 10.17487/RFC6515, February 2012, <<https://www.rfc-editor.org/info/rfc6515>>.
- [RFC6625] Rosen, E., Ed., Rekhter, Y., Ed., Hendrickx, W., and R. Qiu, "Wildcard in Multicast VPN Auto-Discovery Routes", RFC 6625, DOI 10.17487/RFC6625, May 2012, <<https://www.rfc-editor.org/info/rfc6625>>.

- [RFC7716] Zhang, J., Giuliano, L., Rosen, E., Ed., Subramanian, K., and D. Pacella, "Global Table Multicast with BGP Multicast VPN (BGP-MVPN) Procedures", RFC 7716, DOI 10.17487/RFC7716, December 2015, <<https://www.rfc-editor.org/info/rfc7716>>.
- [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>>.
- [RFC8556] Rosen, E., Ed., Sivakumar, M., Przygienda, T., Aldrin, S., and A. Dolganow, "Multicast VPN Using Bit Index Explicit Replication (BIER)", RFC 8556, DOI 10.17487/RFC8556, April 2019, <<https://www.rfc-editor.org/info/rfc8556>>.
- [RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.

9.2. Informative References

- [I-D.cheng-spring-ipv6-msr-design-consideration]
Cheng, W., Mishra, G., Li, Z., Wang, A., Qin, Z., and C. Fan, "Design Consideration of IPv6 Multicast Source Routing (MSR6)", Work in Progress, Internet-Draft, draft-cheng-spring-ipv6-msr-design-consideration-00, 12 July 2021, <<https://www.ietf.org/archive/id/draft-cheng-spring-ipv6-msr-design-consideration-00.txt>>.
- [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)", Work in Progress, Internet-Draft, draft-ietf-6man-spring-srv6-oam-11, 2 June 2021, <<https://www.ietf.org/archive/id/draft-ietf-6man-spring-srv6-oam-11.txt>>.

[I-D.ietf-bess-mvpn-evpn-sr-p2mp]

Parekh, R., Filsfils, C., Venkateswaran, A., Bidgoli, H., Voyer, D., and Z. Zhang, "Multicast and Ethernet VPN with Segment Routing P2MP", Work in Progress, Internet-Draft, draft-ietf-bess-mvpn-evpn-sr-p2mp-04, 19 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-bess-mvpn-evpn-sr-p2mp-04.txt>>.

[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", Work in Progress, Internet-Draft, draft-ietf-bess-srv6-services-07, 11 April 2021, <<https://www.ietf.org/archive/id/draft-ietf-bess-srv6-services-07.txt>>.

[I-D.ietf-rtgwg-dst-src-routing]

Lamparter, D. and A. Smirnov, "Destination/Source Routing", Work in Progress, Internet-Draft, draft-ietf-rtgwg-dst-src-routing-07, 10 March 2019, <<https://www.ietf.org/archive/id/draft-ietf-rtgwg-dst-src-routing-07.txt>>.

[I-D.ietf-spring-sr-replication-segment]

(editor), D. V., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "SR Replication Segment for Multi-point Service Delivery", Work in Progress, Internet-Draft, draft-ietf-spring-sr-replication-segment-05, 20 August 2021, <<https://www.ietf.org/archive/id/draft-ietf-spring-sr-replication-segment-05.txt>>.

[I-D.raszuk-teas-ip-te-np]

Raszuk, R., "IP Traffic Engineering Architecture with Network Programming", Work in Progress, Internet-Draft, draft-raszuk-teas-ip-te-np-00, 2 October 2019, <<https://www.ietf.org/archive/id/draft-raszuk-teas-ip-te-np-00.txt>>.

[I-D.xie-bier-ipv6-encapsulation]

Xie, J., Geng, L., McBride, M., Asati, R., Dhanaraj, S., Zhu, Y., Qin, Z., Shin, M., Mishra, G., and X. Geng, "Encapsulation for BIER in Non-MPLS IPv6 Networks", Work in Progress, Internet-Draft, draft-xie-bier-ipv6-encapsulation-10, 22 February 2021, <<https://www.ietf.org/archive/id/draft-xie-bier-ipv6-encapsulation-10.txt>>.

[I-D.xie-bier-ipv6-mvpn]

Xie, J., McBride, M., Dhanaraj, S., Geng, L., and G. Mishra, "Use of BIER IPv6 Encapsulation (BIERv6) for Multicast VPN in IPv6 networks", Work in Progress, Internet-Draft, draft-xie-bier-ipv6-mvpn-03, 10 October 2020, <<https://www.ietf.org/archive/id/draft-xie-bier-ipv6-mvpn-03.txt>>.

[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>>.

Authors' Addresses

Jingrong Xie
Huawei Technologies

Email: xiejingrong@huawei.com

Xuesong Geng
Huawei Technologies

Email: gengxuesong@huawei.com

Yisong Liu
China Mobile

Email: liuyisong@chinamobile.com