Internet Engineering Task Force (IETF)
Request for Comments: 9524
Category: Standards Track
ISSN: 2070-1721

D. Voyer, Ed.
Bell Canada
C. Filsfils
R. Parekh
Cisco Systems, Inc.
H. Bidgoli
Nokia
Z. Zhang
Juniper Networks

February 2024

Segment Routing Replication for Multipoint Service Delivery

Abstract

This document describes the Segment Routing Replication segment for multipoint service delivery. A Replication segment allows a packet to be replicated from a replication node to downstream nodes.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9524.

Table of Contents

1. Introduction

The Replication segment is a new type of segment for Segment Routing (SR) [RFC8402], which allows a node (henceforth called a "replication node") to replicate packets to a set of other nodes (called "downstream nodes") in an SR domain. A Replication segment can replicate packets to directly connected nodes or to downstream nodes (without the need for state on the transit routers). This document focuses on specifying the behavior of a Replication segment for both Segment Routing with Multiprotocol Label Switching (SR-MPLS) [RFC8660] and Segment Routing with IPv6 (SRv6) [RFC8986]. The examples in Appendix A illustrate the behavior of a Replication Segment in an SR domain. The use of two or more Replication segments stitched together to form a tree using a control plane is left to be specified in other documents. The management of IP multicast groups, building IP multicast trees, and performing multicast congestion control are out of scope of this document.

1.1. Terminology

This section defines terms introduced and used frequently in this document. Refer to the Terminology sections of [RFC8402], [RFC8754], and [RFC8986] for other terms used in SR.

Replication segment: A segment in an SR domain that replicates packets. See Section 2 for details.

Replication node: A node in an SR domain that replicates packets based on a Replication segment.

Downstream nodes: A Replication segment replicates packets to a set of nodes. These nodes are downstream nodes.

Replication state: State held for a Replication segment at a replication node. It is conceptually a list of Replication branches to downstream nodes. The list can be empty.

Replication-SID: Data plane identifier of a Replication segment. This is an SR-MPLS label or SRv6 Segment Identifier (SID).

SRH: IPv6 Segment Routing Header [RFC8754].

Point-to-Multipoint (P2MP) Service: A service that has one ingress node and one or more egress nodes. A packet is delivered to all the egress nodes.

Root node: An ingress node of a P2MP service.

Leaf node: An egress node of a P2MP service.

Bud node: A node that is both a replication node and a leaf node.

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.

1.2. Use Cases

In the simplest use case, a single Replication segment includes the ingress node of a multipoint service and the egress nodes of the service as all the downstream nodes. This achieves Ingress Replication [RFC7988] that has been widely used for Multicast VPN (MVPN) [RFC6513] and Ethernet VPN (EVPN) [RFC7432] bridging of Broadcast, Unknown Unicast, and Multicast (BUM) traffic. This Replication segment on ingress and egress nodes can either be provisioned locally or using dynamic autodiscovery procedures for MVPN and EVPN. Note SRv6 [RFC8986] has End.DT2M replication behavior for EVPN BUM traffic.

Replication segments can also be used to form trees by stitching Replication segments on a root node, intermediate replication nodes, and leaf nodes for efficient delivery of MVPN and EVPN BUM traffic.

2. Replication Segment

In an SR domain, a Replication segment is a logical construct that connects a replication node to a set of downstream nodes. A Replication segment is a local segment instantiated at a Replication node. It can be either provisioned locally on a node or programmed by a control plane.

Replication segments can be stitched together to form a tree by either local provisioning on nodes or using a control plane. The procedures for doing this are out of scope of this document. One such control plane using a PCE with the SR P2MP policy is specified in [P2MP-POLICY]. However, if local provisioning is used to stitch Replication segments, then a chain of Replication segments SHOULD NOT form a loop. If a control plane is used to stitch Replication segments, the control plane specification MUST prevent loops or detect and mitigate loops in steady state.

A Replication segment is identified by the tuple <Replication-ID, Node-ID>, where:

Replication-ID: An identifier for a Replication segment that is unique in context of the replication node.

Node-ID: The address of the replication node for the Replication segment. Note that the root of a multipoint service is also a Replication node.

Replication-ID is a variable-length field. In the simplest case, it can be a 32-bit number, but it can be extended or modified as required based on the specific use of a Replication segment. This is out of scope for this document. The length of the Replication-ID is specified in the signaling mechanism used for the Replication segment. Examples of such signaling and extensions are described in [P2MP-POLICY]. When the PCE signals a Replication segment to its node, the <Replication-ID, Node-ID> tuple identifies the segment.

A Replication segment includes the following elements:

Replication-SID: The Segment Identifier of a Replication segment. This is an SR-MPLS label or an SRv6 SID [RFC8402].

Downstream nodes: Set of nodes in an SR domain to which a packet is replicated by the Replication segment.

Replication state: See below.

The downstream nodes and Replication state (RS) of a Replication segment can change over time, depending on the network state and leaf nodes of a multipoint service that the segment is part of.

The Replication-SID identifies the Replication segment in the forwarding plane. At a replication node, the Replication-SID operates on the RS of the Replication segment.

RS is a list of Replication branches to the downstream nodes. In this document, each branch is abstracted to a <downstream node, downstream Replication-SID> tuple. <downstream node> represents the reachability from the replication node to the downstream node. In its simplest form, this MAY be specified as an interface or next-hop if the downstream node is adjacent to the replication node. The reachability may be specified in terms of a Flexible Algorithm path (including the default algorithm) [RFC9350] or specified by an SR-explicit path represented either by a SID list (of one or more SIDs) or by a Segment Routing Policy [RFC9256]. The downstream Replication-SID is the Replication-SID of the Replication segment at the downstream node.

A packet is steered into a Replication segment at a replication node in two ways:

* When the active segment [RFC8402] is a locally instantiated Replication-SID.

* By the root of a multipoint service based on local configuration that is outside the scope of this document.

In either case, the packet is replicated to each downstream node in the associated RS.

If a downstream node is an egress (leaf) of the multipoint service, no further replication is needed. The leaf node's Replication segment has an indicator for the leaf role, and it does not have any RS (i.e., the list of Replication branches is empty). The Replication-SID at a leaf node MAY be used to identify the multipoint service. Notice that the segment on the leaf node is still referred to as a "Replication segment" for the purpose of generalization.

A node can be a bud node (i.e., it is a replication node and a leaf node of a multipoint service [P2MP-POLICY]). The Replication segment of a bud node has a list of Replication branches as well as a leaf role indicator.

In principle, it is possible for different Replication segments to replicate packets to the same Replication segment on a downstream node. However, such usage is intentionally left out of scope of this document.

2.1. SR-MPLS Data Plane

When the active segment is a Replication-SID, the processing results in a POP [RFC8402] operation and the lookup of the associated RS. For each replication in the RS, the operation is a PUSH [RFC8402] of the downstream Replication-SID and an optional segment list onto the packet to steer the packet to the downstream node.

The operation performed on the incoming Replication-SID is NEXT [RFC8402] at a leaf or bud node where delivery of payload off the tree is per local configuration. For some usages, this may involve looking at the next SID, for example, to get the necessary context.

When the root of a multipoint service steers a packet to a

Replication segment, it results in a replication to each downstream node in the associated RS. The operation is a PUSH of the Replication-SID and an optional segment list onto the packet, which is forwarded to the downstream node.

The following applies to a Replication-SID in MPLS encapsulation:

* SIDs MAY be inserted before the downstream SR-MPLS Replication-SID in order to guide a packet from a non-adjacent SR node to a replication node.

* A replication node MAY replicate a packet to a non-adjacent downstream node using SIDs it inserts in the copy preceding the downstream Replication-SID. The downstream node may be a leaf node of the Replication segment, another replication node, or both in the case of a bud node.

* A replication node MAY use an Anycast-SID or a Border Gateway Protocol (BGP) PeerSet-SID in the segment list to send a replicated packet to one downstream replication node in a set of Anycast nodes. This occurs if and only if all nodes in the set have an identical Replication-SID and reach the same set of receivers.

* For some use cases, there MAY be SIDs after the Replication-SID in the segment list of a packet. These SIDs are used only by the leaf and bud nodes to forward a packet off the tree independent of the Replication-SID. Coordination regarding the absence or presence and value of context information for leaf and bud nodes is outside the scope of this document.

2.2. SRv6 Data Plane

For SRv6 [RFC8986], this document specifies "Endpoint with replication and/or decapsulate" behavior (End.Replicate for short) to replicate a packet and forward the replicas according to an RS.

When processing a packet destined to a local Replication-SID, the packet is replicated according to the associated RS to downstream nodes and/or locally delivered off the tree when this is a leaf or bud node. For replication, the outer header is reused, and the downstream Replication-SID, from RS, is written into the outer IPv6 header Destination Address (DA). If required, an optional segment list may be used on some branches using H.Encaps.Red [RFC8986] (while some other branches may not need that). Note that this H.Encaps.Red is independent of the Replication segment: it is just used to steer the replicated packet on a traffic-engineered path to a downstream node. The penultimate segment in the encapsulating IPv6 header will execute the Ultimate Segment Decapsulation (USD) flavor [RFC8986] of End/End.X behavior and forward the inner (replicated) packet to the downstream node. If H.Encaps.Red is used to steer a replicated packet to a downstream node, the operator must ensure the MTU on path to the downstream node is sufficient to account for additional SRv6 encapsulation. This also applies when the Replication segment is for the root node, whose upstream node has placed the Replication-SID in the header.

A local application on root (e.g., MVPN [RFC6513] or EVPN [RFC7432]) may also apply H.Encaps.Red and then steer the resulting traffic into the Replication segment. Again, note that H.Encaps.Red is independent of the Replication segment: it is the action of the application (e.g. MVPN or EVPN service). If the service is on a root node, then the two H.Encaps mentioned, one for the service and the other in the previous paragraph for replication to the downstream node, SHOULD be combined for optimization (to avoid extra IPv6 encapsulation).

When processing a packet destined to a local Replication-SID, the IPv6 Hop Limit MUST be decremented and MUST be non-zero to replicate the packet. A root node that encapsulates a payload can set the IPv6 Hop Limit based on a local policy. This local policy SHOULD set the IPv6 Hop Limit so that a replicated packet can reach the furthest leaf node. A root node can also have a local policy to set the IPv6 Hop Limit from the payload. In this case, the IPv6 Hop Limit may not be sufficient to get the replicated packet to all the leaf nodes. Non-replication nodes (i.e., nodes that forward replicated packets based on the IPv6 locator unicast prefix) can decrement the IPv6 Hop Limit to zero and originate ICMPv6 error packets to the root node. This can result in a storm of ICMPv6 packets (see Section 2.2.3) to the root node. To avoid this, a Replication segment has an optional IPv6 Hop Limit Threshold. If this threshold is set, a replication node MUST discard an incoming packet with a local Replication-SID if the IPv6 Hop Limit in the packet is less than the threshold and log this in a rate-limited manner. The IPv6 Hop Limit Threshold SHOULD be set so that an incoming packet can be replicated to the furthest leaf node.

For leaf and bud nodes, local delivery off the tree is per Replication-SID or the next SID (if present in the SRH). For some usages, this may involve getting the necessary context either from the next SID (e.g., MVPN with a shared tree) or from the Replication-SID itself (e.g., MVPN with a non-shared tree). In both cases, the context association is achieved with signaling and is out of scope of this document.

The following applies to a Replication-SID in SRv6 encapsulation:

* There MAY be SIDs preceding the SRv6 Replication-SID in order to guide a packet from a non-adjacent SR node to a replication node via an explicit path.

* A replication node MAY steer a replicated packet on an explicit path to a non-adjacent downstream node using SIDs it inserts in the copy preceding the downstream Replication-SID. The downstream node may be a leaf node of the Replication segment, another replication node, or both in the case of a bud node.

* For SRv6, as described in above paragraphs, the insertion of SIDs prior to the Replication-SID entails a new IPv6 encapsulation with the SRH. However, this can be optimized on the root node or for compressed SRv6 SIDs.

* The locator of the Replication-SID is sufficient to guide a packet on the shortest path between non-adjacent nodes for default or Flexible Algorithms.

* A replication node MAY use an Anycast-SID or a BGP PeerSet-SID in the segment list to send a replicated packet to one downstream replication node in an Anycast set. This occurs if and only if all nodes in the set have an identical Replication-SID and reach the same set of receivers.

* There MAY be SIDs after the Replication-SID in the SRH of a packet. These SIDs are used to provide additional context for processing a packet locally at the node where the Replication-SID is the active segment. Coordination regarding the absence or presence and value of context information for leaf and bud nodes is outside the scope of this document.

2.2.1. End.Replicate: Replicate and/or Decapsulate

The "Endpoint with replication and/or decapsulate" (End.Replicate for

short) is a variant of End behavior. The pseudocode in this section follows the convention introduced in [RFC8986].

An RS conceptually contains the following elements:

```
Replication state:
{
  Node-Role: {Head, Transit, Leaf, Bud};
  IPv6 Hop Limit Threshold; # default is zero
  # On Leaf, replication list is zero length
  Replication-List:
  {
    downstream node: <Node-Identifier>;
    downstream Replication-SID: R-SID;
    # Segment-List may be empty
    Segment-List: [SID-1, .... SID-N];
  }
}
```

Below is the Replicate function on a packet for Replication state (RS).

```
S01. Replicate(RS, packet)
S02. {
S03.    For each Replication R in RS.Replication-List {
S04.       Make a copy of the packet
S05.       Set IPv6 DA = RS.R-SID
S06.       If RS.Segment-List is not empty {
S07.         # Head node may optimize below encapsulation and
S08.         # the encapsulation of packet in a single encapsulation
S09.         Execute H.Encaps or H.Encaps.Red with RS.Segment-List
             on packet copy #RFC 8986, Sections 5.1 and 5.2
S10.       }
S11.       Submit the packet to the egress IPv6 FIB lookup and
           transmission to the new destination
S12.   }
S13. }
```

Notes:

* The IPv6 DA in the copy of a packet is set from the local state and not from the SRH.

When N receives a packet whose IPv6 DA is S and S is a local End.Replicate SID, N does:

```
S01.   Lookup FUNCT portion of S to get Replication state (RS)
S02.   If (IPv6 Hop Limit <= 1) {
S03.     Discard the packet
S04.     # ICMPv6 Time Exceeded is not permitted
           (see Section 2.2.3)
S05.   }
S06.   If RS is not found {
S07.     Discard the packet
S08.   }
S09.   If (IPv6 Hop Limit < RS.IPv6 Hop Limit Threshold) {
S10.     Discard the packet
S11.     # Rate-limited logging
S12.   }
S13.   Decrement IPv6 Hop Limit by 1
S14.   If (IPv6 NH == SRH and SRH TLVs present) {
S15.     Process SRH TLVs if allowed by local configuration
S16.   }
S17.   Call Replicate(RS, packet)
S18.   If (RS.Node-Role == Leaf OR RS.Node-Role == bud) {
S19.     If (IPv6 NH == SRH and Segments Left > 0) {
```

```
S20.       Derive packet processing context (PPC) from Segment List
S21.       If (Segments Left != 0) {
S22.         Discard the packet
S23.         # ICMPv6 Parameter Problem message with Code 0
S24.         # (Erroneous header field encountered)
S25.         # is not permitted (Section 2.2.3)
S26.       }
S27.     } Else {
S28.       Derive packet processing context (PPC)
           from FUNCT of Replicatio-SID
S29.     }
S30.     Process the next header
S31.   }
```

The processing of the Upper-Layer header of a packet matching the End.Replicate SID at a leaf or bud node is as follows:

```
S01.   If (Upper-Layer header type == 4(IPv4) OR
           Upper-Layer header type == 41(IPv6) ) {
S02.     Remove the outer IPv6 header with all its extension headers
S03.     Process the packet in context of PPC
S04.   } Else If (Upper-Layer header type == 143(Ethernet) ) {
S05.     Remove the outer IPv6 header with all its extension headers
S06.     Process the Ethernet Frame in context of PPC
S07.   } Else If (Upper-Layer header type is allowed
                  by local configuration) {
S08.     Proceed to process the Upper-Layer header
S09.   } Else {
S10.     Discard the packet
S11.     # ICMPv6 Parameter Problem message with Code 4
S12.     # (SR Upper-Layer header Error)
S13.     # is not permitted (Section 2.2.3)
S14.   }
```

Notes:

* The behavior above MAY result in a packet with a partially processed segment list in the SRH under some circumstances. For example, a head node may encode a context-SID in an SRH. As per the pseudocode above, a replication node that receives a packet with a local Replication-SID will not process the SRH segment list and will just forward a copy with an unmodified SRH to downstream nodes.

* The packet processing context is usually a FIB table "T".

If configured to process TLVs, processing the Replication-SID may modify the "variable-length data" of TLV types that change en route. Therefore, TLVs that change en route are mutable. The remainder of the SRH (Segments Left, Flags, Tag, Segment List, and TLVs that do not change en route) are immutable while processing this SID.

2.2.1.1. Hashed Message Authentication Code (HMAC) SRH TLV

If a root node encodes a context-SID in an SRH with an optional HMAC SRH TLV [RFC8754], it MUST set the ’D’ bit as defined in Section 2.1.2 of [RFC8754] because the Replication-SID is not part of the segment list in the SRH.

HMAC generation and verification is as specified in [RFC8754]. Verification of an HMAC TLV is determined by local configuration. If verification fails, an implementation of a Replication-SID MUST NOT originate an ICMPv6 Parameter Problem message with code 0. The failure SHOULD be logged (rate-limited) and the packet SHOULD be discarded.

2.2.2. OAM Operations

[RFC9259] specifies procedures for Operations, Administration, and Maintenance (OAM) like ping and traceroute on SRv6 SIDs.

Assuming the source node knows the Replication-SID a priori, it is possible to ping a Replication-SID of a leaf or bud node directly by putting it in the IPv6 DA without an SRH or in an SRH as the last segment. While it is not possible to ping a Replication-SID of a transit node because transit nodes do not process Upper-Layer headers, it is still possible to ping a Replication-SID of a leaf or bud node of a tree via the Replication-SID of intermediate transit nodes. The source of the ping MUST compute the ICMPv6 Echo Request checksum using the Replication-SID of the leaf or bud node as the DA. The source can then send the Echo Request packet to a transit node's Replication-SID. The transit node replicates the packet by replacing the IPv6 DA until the packet reaches the leaf or bud node, which responds with an ICMPv6 Echo Reply. Note that a transit replication node may replicate Echo Request packets to other leaf or bud nodes. These nodes will drop the Echo Request due to an incorrect checksum. Procedures to prevent the misdelivery of an Echo Request may be addressed in a future document. Appendix A.2.1 illustrates examples of a ping to a Replication-SID.

Traceroute to a leaf or bud node Replication-SID is not possible due to restrictions prohibiting the origination of the ICMPv6 Time Exceeded error message for a Replication-SID as described in Section 2.2.3.

2.2.3. ICMPv6 Error Messages

Section 2.4 of [RFC4443] states an ICMPv6 error message MUST NOT be originated as a result of receiving a packet destined to an IPv6 multicast address. This is to prevent a source node from being overwhelmed by a storm of ICMPv6 error messages resulting from replicated IPv6 packets. There are two exceptions:

1. The Packet Too Big message for Path MTU discovery, and

2. The ICMPv6 Parameter Problem message with Code 2 reporting an unrecognized IPv6 option.

An implementation of a Replication segment for SRv6 MUST enforce these same restrictions and exceptions.

3. IANA Considerations

IANA has assigned the following codepoint for End.Replicate behavior in the "SRv6 Endpoint Behaviors" registry in the "Segment Routing" registry group.

Value	Hex	Endpoint Behavior	Reference	Change Controller
75	0x004B	End.Replicate	RFC 9524	IETF

Table 1: SRv6 Endpoint Behavior

4. Security Considerations

The SID behaviors defined in this document are deployed within an SR domain [RFC8402]. An SR domain needs protection from outside attackers (as described in [RFC8754]). The following is a brief reminder of the same:

* For SR-MPLS deployments:

  - Disable MPLS on external interfaces of each edge node or any other technique to filter labeled traffic ingress on these interfaces.

* For SRv6 deployments:

  - Allocate all the SIDs from an IPv6 prefix block S/s and configure each external interface of each edge node of the domain with an inbound Infrastructure Access Control List (IACL) that drops any incoming packet with a DA in S/s.

  - Additionally, an IACL may be applied to all nodes (k) provisioning SIDs as defined in this specification:

    o Assign all interface addresses from within IPv6 prefix A/a. At node k, all SIDs local to k are assigned from prefix Sk/sk. Configure each internal interface of each SR node k in the SR domain with an inbound IACL that drops any incoming packet with a DA in Sk/sk if the source address is not in A/a.

  - Deny traffic with spoofed source addresses by implementing recommendations in BCP 84 [RFC3704].

  - Additionally, the block S/s from which SIDs are allocated may be an address that is not globally routable such as a Unique Local Address (ULA) or the prefix defined in [SIDS-SRv6].

Failure to protect the SR-MPLS domain by correctly provisioning MPLS support per interface permits attackers from outside the domain to send packets that use the replication services provisioned within the domain.

Failure to protect the SRv6 domain with IACLs on external interfaces combined with failure to implement the recommendations of BCP 38 [RFC2827] or apply IACLs on nodes provisioning SIDs permits attackers from outside the SR domain to send packets that use the replication services provisioned within the domain.

Given the definition of the Replication segment in this document, an attacker subverting the ingress filters above cannot take advantage of a stack of Replication segments to perform amplification attacks nor link exhaustion attacks. Replication segment trees always terminate at a leaf or bud node resulting in a decapsulation. However, this does allow an attacker to inject traffic to the receivers within a P2MP service.

This document introduces an SR segment endpoint behavior that replicates and decapsulates an inner payload for both the MPLS and IPv6 data planes. Similar to any MPLS end-of-stack label, or SRv6 END.D* behavior, if the protections described above are not implemented, an attacker can perform an attack via the decapsulating segment (including the one described in this document).

Incorrect provisioning of Replication segments can result in a chain of Replication segments forming a loop. This can happen if Replication segments are provisioned on SR nodes without using a control plane. In this case, replicated packets can create a storm until MPLS TTL (for SR-MPLS) or IPv6 Hop Limit (for SRv6) decrements to zero. A control plane such as PCE can be used to prevent loops. The control plane protocols (like Path Computation Element Communication Protocol (PCEP), BGP, etc.) used to instantiate Replication segments can leverage their own security mechanisms such

as encryption, authentication filtering, etc.

For SRv6, Section 2.2.3 describes an exception for the ICMPv6 Parameter Problem message with Code 2. If an attacker sends a packet destined to a Replication-SID with the source address of a node and with an extension header using the unknown option type marked as mandatory, then a large number of ICMPv6 Parameter Problem messages can cause a denial-of-service attack on the source node. Although this document does not specify any extension headers, any future extension of this document that does so is susceptible to this security concern.

If an attacker can forge an IPv6 packet with:

* the source address of a node,

* a Replication-SID as the DA, and

* an IPv6 Hop Limit such that nodes that forward replicated packets on an IPv6 locator unicast prefix, decrement the Hop Limit to zero,

then these nodes can cause a storm of ICMPv6 error packets to overwhelm the source node under attack. The IPv6 Hop Limit Threshold check described in Section 2.2 can help mitigate such attacks.

5. References

5.1. 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>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

[RFC9259] Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing over IPv6 (SRv6)", RFC 9259, DOI 10.17487/RFC9259, June 2022, <https://www.rfc-editor.org/info/rfc9259>.

5.2. Informative References

[P2MP-POLICY]
Voyer, D., Ed., Filsfils, C., Parekh, R., Bidgoli, H., and Z. J. Zhang, "Segment Routing Point-to-Multipoint Policy", Work in Progress, Internet-Draft, draft-ietf-pim-sr-p2mp-policy-07, 11 October 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-pim-sr-p2mp-policy-07>.

[PGM-ILLUSTRATION]
Filsfils, C., Camarillo, P., Ed., Li, Z., Matsushima, S., Decraene, B., Steinberg, D., Lebrun, D., Raszuk, R., and J. Leddy, "Illustrations for SRv6 Network Programming", Work in Progress, Internet-Draft, draft-filsfils-spring-srv6-net-pgm-illustration-04, 30 March 2021, <https://datatracker.ietf.org/doc/html/draft-filsfils-spring-srv6-net-pgm-illustration-04>.

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

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <https://www.rfc-editor.org/info/rfc3704>.

[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>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <https://www.rfc-editor.org/info/rfc7432>.

[RFC7988] Rosen, E., Ed., Subramanian, K., and Z. Zhang, "Ingress Replication Tunnels in Multicast VPN", RFC 7988, DOI 10.17487/RFC7988, October 2016, <https://www.rfc-editor.org/info/rfc7988>.

[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>.

[RFC9256] Filsfils, C., Talaulikar, K., Ed., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", RFC 9256, DOI 10.17487/RFC9256, July 2022, <https://www.rfc-editor.org/info/rfc9256>.

[RFC9350] Psenak, P., Ed., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", RFC 9350, DOI 10.17487/RFC9350, February 2023, <https://www.rfc-editor.org/info/rfc9350>.

[SIDS-SRv6]
Krishnan, S., "SRv6 Segment Identifiers in the IPv6 Addressing Architecture", Work in Progress, Internet-Draft, draft-ietf-6man-sids-06, 15 February 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-6man-sids-06>.

Appendix A. Illustration of a Replication Segment

This section illustrates an example of a single Replication segment. Examples showing Replication segments stitched together to form a P2MP tree (based on SR P2MP policy) are in [P2MP-POLICY].

Consider the following topology:

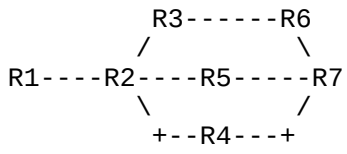

Figure 1: Topology for Illustration of a Replication Segment

A.1. SR-MPLS

In this example, the Node-SID of a node Rn is N-SIDn and the Adj-SID from node Rm to node Rn is A-SIDmn. The interface between Rm and Rn is Lmn. The state representation uses "R-SID->Lmn" to represent a packet replication with outgoing Replication-SID R-SID sent on interface Lmn.

Assume a Replication segment identified with R-ID at Replication node R1 and downstream nodes R2, R6, and R7. The Replication-SID at node n is R-SIDn. A packet replicated from R1 to R7 has to traverse R4.

The Replication segments at nodes R1, R2, R6, and R7 are shown below. Note nodes R3, R4, and R5 do not have a Replication segment.

Replication segment at R1:

Replication segment
<R-ID,R1>: Replication-SID: R-SID1 Replication state: R2:
<R-SID2->L12> R6: <N-SID6, R-SID6> R7: <N-SID4,
A-SID47, R-SID7>

Replication to R2 steers the packet directly to R2 on interface L12. Replication to R6, using N-SID6, steers the packet via the shortest path to that node. Replication to R7 is steered via R4, using N-SID4 and then adjacency SID A-SID47 to R7.

Replication segment at R2:

Replication segment
<R-ID,R2>: Replication-SID: R-SID2 Replication state: R2:
<Leaf>

Replication segment at R6:

Replication segment
<R-ID,R6>: Replication-SID: R-SID6 Replication state: R6:
<Leaf>

Replication segment at R7:

Replication segment
<R-ID,R7>: Replication-SID: R-SID7 Replication state: R7:
<Leaf>

When a packet is steered into the Replication segment at R1:

* R1 performs the PUSH operation with just the <R-SID2> label for the replicated copy and sends it to R2 on interface L12, since R1 is directly connected to R2. R2, as leaf, performs the NEXT

operation, pops the R-SID2 label, and delivers the payload.

* R1 performs the PUSH operation with the <N-SID6, R-SID6> label stack for the replicated copy to R6 and sends it to R2, which is the nexthop on the shortest path to R6. R2 performs the CONTINUE operation on N-SID6 and forwards it to R3. R3 is the penultimate hop for N-SID6; it performs penultimate hop popping, which corresponds to the NEXT operation. The packet is then sent to R6 with <R-SID6> in the label stack. R6, as leaf, performs the NEXT operation, pops the R-SID6 label, and delivers the payload.

* R1 performs the PUSH operation with the <N-SID4, A-SID47, R-SID7> label stack for the replicated copy to R7 and sends it to R2, which is the nexthop on the shortest path to R4. R2 is the penultimate hop for N-SID4; it performs penultimate hop popping, which corresponds to the NEXT operation. The packet is then sent to R4 with <A-SID47, R-SID1> in the label stack. R4 performs the NEXT operation, pops A-SID47, and delivers the packet to R7 with <R-SID7> in the label stack. R7, as leaf, performs the NEXT operation, pops the R-SID7 label, and delivers the payload.

A.2. SRv6

For SRv6, we use the SID allocation scheme, reproduced below, from "Illustrations for SRv6 Network Programming" [PGM-ILLUSTRATION]:

* 2001:db8::/32 is an IPv6 block allocated by a Regional Internet Registry (RIR) to the operator.

* 2001:db8:0::/48 is dedicated to the internal address space.

* 2001:db8:cccc::/48 is dedicated to the internal SRv6 SID space.

* We assume a location expressed in 64 bits and a function expressed in 16 bits.

* Node k has a classic IPv6 loopback address 2001:db8::k/128, which is advertised in the Interior Gateway Protocol (IGP).

* Node k has 2001:db8:cccc:k::/64 for its local SID space. Its SIDs will be explicitly assigned from that block.

* Node k advertises 2001:db8:cccc:k::/64 in its IGP.

* Function :1:: (function 1, for short) represents the End function with the Penultimate Segment Pop (PSP) of the SRH [RFC8986] and USD support.

* Function :Cn:: (function Cn, for short) represents the End.X function from to Node n with PSP and USD support.

Each node k has:

* An explicit SID instantiation 2001:db8:cccc:k:1::/128 bound to an End function with additional support for PSP and USD.

* An explicit SID instantiation 2001:db8:cccc:k:Cj::/128 bound to an End.X function to neighbor J with additional support for PSP and USD.

* An explicit SID instantiation 2001:db8:cccc:k:Fk::/128 bound to an End.Replicate function.

Assume a Replication segment identified with R-ID at Replication node R1 and downstream nodes R2, R6, and R7. The Replication-SID at node k, bound to an End.Replicate function, is 2001:db8:cccc:k:Fk::/128.

A packet replicated from R1 to R7 has to traverse R4.

The Replication segments at nodes R1, R2, R6, and R7 are shown below. Note nodes R3, R4, and R5 do not have a Replication segment. The state representation uses "R-SID->Lmn" to represent a packet replication with outgoing Replication-SID R-SID sent on interface Lmn. "SL" represents an optional segment list used to steer a replicated packet on a specific path to a downstream node.

Replication segment at R1:

Replication segment
<R-ID,R1>: Replication-SID: 2001:db8:cccc:1:F1::0 Replication state: R2: <2001:db8:cccc:2:F2::0->L12> R6: <2001:db8:cccc:6:F6::0> R7: <2001:db8:cccc:4:C7::0>, SL: <2001:db8:cccc:7:F7::0>

Replication to R2 steers the packet directly to R2 on interface L12. Replication to R6, using 2001:db8:cccc:6:F6::0, steers the packet via the shortest path to that node. Replication to R7 is steered via R4, using H.Encaps.Red with End.X SID 2001:db8:cccc:4:C7::0 at R4 to R7.

Replication segment at R2:

Replication segment
<R-ID,R2>: Replication-SID: 2001:db8:cccc:2:F2::0 Replication state: R2: <Leaf>

Replication segment at R6:

Replication segment
<R-ID,R6>: Replication-SID: 2001:db8:cccc:6:F6::0 Replication state: R6: <Leaf>

Replication segment at R7:

Replication segment
<R-ID,R7>: Replication-SID: 2001:db8:cccc:7:F7::0 Replication state: R7: <Leaf>

When a packet, (A,B2), is steered into the Replication segment at R1:

* R1 creates an encapsulated replicated copy (2001:db8::1, 2001:db8:cccc:2:F2::0) (A, B2), and sends it to R2 on interface L12, since R1 is directly connected to R2. R2, as leaf, removes the outer IPv6 header and delivers the payload.

* R1 creates an encapsulated replicated copy (2001:db8::1, 2001:db8:cccc:6:F6::0) (A, B2) then forwards the resulting packet on the shortest path to 2001:db8:cccc:6::/64. R2 and R3 forward the packet using 2001:db8:cccc:6::/64. R6, as leaf, removes the outer IPv6 header and delivers the payload.

* R1 has to steer the packet to downstream node R7 via node R4. It can do this in one of two ways:

  - R1 creates an encapsulated replicated copy (2001:db8::1, 2001:db8:cccc:7:F7::0) (A, B2) and then performs H.Encaps.Red using the SL to create the (2001:db8::1, 2001:db8:cccc:4:C7::0) (2001:db8::1, 2001:db8:cccc:7:F7::0) (A, B2) packet. It sends this packet to R2, which is the nexthop on the shortest path to 2001:db8:cccc:4::/64. R2 forwards the packet to R4 using 2001:db8:cccc:4::/64. R4 executes the End.X function on 2001:db8:cccc:4:C7::0, performs a USD action, removes the outer IPv6 encapsulation, and sends the resulting packet (2001:db8::1, 2001:db8:cccc:7:F7::0) (A, B2) to R7. R7, as

leaf, removes the outer IPv6 header and delivers the payload.

- R1 is the root of the Replication segment. Therefore, it can combine above encapsulations to create an encapsulated replicated copy (2001:db8::1, 2001:db8:cccc:4:C7::0) (2001:db8:cccc:7:F7::0; SL=1) (A, B2) and sends it to R2, which is the nexthop on the shortest path to 2001:db8:cccc:4::/64. R2 forwards the packet to R4 using 2001:db8:cccc:4::/64. R4 executes the End.X function on 2001:db8:cccc:4:C7::0, performs a PSP action, removes the SRH, and sends the resulting packet (2001:db8::1, 2001:db8:cccc:7:F7::0) (A, B2) to R7. R7, as leaf, removes the outer IPv6 header and delivers the payload.

A.2.1. Pinging a Replication-SID

This section illustrates the ping of a Replication-SID.

Node R1 pings the Replication-SID of node R6 directly by sending the following packet:

1. R1 to R6: (2001:db8::1, 2001:db8:cccc:6:F6::0; NH=ICMPv6) (ICMPv6 Echo Request).

2. Node R6 as a leaf processes the upper-layer ICMPv6 Echo Request and responds with an ICMPv6 Echo Reply.

Node R1 pings the Replication-SID of R7 via R4 by sending the following packet with the SRH:

1. R1 to R4: (2001:db8::1, 2001:db8:cccc:4:C7::0) (2001:db8:cccc:7:F7::0; SL=1; NH=ICMPV6) (ICMPv6 Echo Request).

2. R4 to R7: (2001:db8::1, 2001:db8:cccc:7:F7::0; NH=ICMPv6) (ICMPv6 Echo Request).

3. Node R7 as a leaf processes the upper-layer ICMPv6 Echo Request and responds with an ICMPv6 Echo Reply.

Assume node R4 is a transit replication node with Replication-SID 2001:db8:cccc:4:F4::0 replicating to R7. Node R1 pings the Replication-SID of R7 via the Replication-SID of R4 as follows:

1. R1 to R4: (2001:db8::1, 2001:db8:cccc:4:F4::0; NH=ICMPv6) (ICMPv6 Echo Request).

2. R4 replicates to R7 by replacing the IPv6 DA with the Replication-SID of R7 from its Replication state.

3. R4 to R7: (2001:db8::1, 2001:db8:cccc:7:F7::0; NH=ICMPv6) (ICMPv6 Echo Request).

4. Node R7 as a leaf processes the upper-layer ICMPv6 Echo Request and responds with an ICMPv6 Echo Reply.

Acknowledgements

The authors would like to acknowledge Siva Sivabalan, Mike Koldychev, Vishnu Pavan Beeram, Alexander Vainshtein, Bruno Decraene, Thierry Couture, Joel Halpern, Ketan Talaulikar, Darren Dukes and Jingrong Xie for their valuable inputs.

Contributors

Clayton Hassen
Bell Canada
Vancouver

Canada
Email: clayton.hassen@bell.ca

Kurtis Gillis
Bell Canada
Halifax
Canada
Email: kurtis.gillis@bell.ca

Arvind Venkateswaran
Cisco Systems, Inc.
San Jose, CA
United States of America
Email: arvvenka@cisco.com

Zafar Ali
Cisco Systems, Inc.
United States of America
Email: zali@cisco.com

Swadesh Agrawal
Cisco Systems, Inc.
San Jose, CA
United States of America
Email: swaagraw@cisco.com

Jayant Kotalwar
Nokia
Mountain View, CA
United States of America
Email: jayant.kotalwar@nokia.com

Tanmoy Kundu
Nokia
Mountain View, CA
United States of America
Email: tanmoy.kundu@nokia.com

Andrew Stone
Nokia
Ottawa
Canada
Email: andrew.stone@nokia.com

Tarek Saad
Cisco Systems, Inc.
Canada
Email: tsaad@cisco.com

Kamran Raza
Cisco Systems, Inc.
Canada
Email: skraza@cisco.com

Jingrong Xie
Huawei Technologies

Beijing
China
Email: xiejingrong@huawei.com

Authors’ Addresses

Daniel Voyer (editor)
Bell Canada
Montreal
Canada
Email: daniel.voyer@bell.ca

Clarence Filsfils
Cisco Systems, Inc.
Brussels
Belgium
Email: cfilsfil@cisco.com

Rishabh Parekh
Cisco Systems, Inc.
San Jose, CA
United States of America
Email: riparekh@cisco.com

Hooman Bidgoli
Nokia
Ottawa
Canada
Email: hooman.bidgoli@nokia.com

Zhaohui Zhang
Juniper Networks
Email: zzhang@juniper.net