Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 1, 2021

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

October 28, 2020

# SR Replication Segment for Multi-point Service Delivery
## draft-ietf-spring-sr-replication-segment-01

Abstract

This document describes the SR Replication segment for Multi-point service delivery. A SR Replication segment allows a packet to be replicated from a replication node to downstream nodes.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on May 1, 2021.

Table of Contents



## 1. Introduction

We define a new type of segment for Segment Routing [RFC8402], called Replication segment, which allows a node (henceforth called as Replication Node) to replicate packets to a set of other nodes (called Downstream Nodes) in a Segment Routing Domain. Replication segments provide building blocks for Point-to-Multipoint Service delivery via SR Point-to-Multipoint (SR P2MP) policy. A Replication segment can replicate packet to directly connected nodes or to downstream nodes (without need for state on the transit routers). Replication segments apply equally to both SR-MPLS and SRv6 instantiations of Segment Routing. This document focuses on the

Replication Segment building block. The use of one or more stitched Replication Segments constructed for SR P2MP Policy tree is specified in [I-D.ietf-pim-sr-p2mp-policy].

## 2. Replication Segment

In a Segment Routing Domain, a Replication segment is a logical construct which 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 PCE.

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 that the Replication segment is for. Note that the root of a multi-point service is also a replication node.

In simplest case, Replication-ID can be a 32-bit number, but it can be extended or modified as required based on specific use of a Replication segment. When the PCE signals a Replication segment to its node, the <Replication-ID, Node-ID> tuple identifies the segment. Examples of such signaling and extension are described in [I-D.ietf-pim-sr-p2mp-policy].

A Replication segment includes the following elements:

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

- Downstream Nodes: Set of nodes in Segment Routing domain to which a packet is replicated by the Replication segment.

- Replication State: See below.

The Downstream Nodes and Replication State of a Replication segment can change over time, depending on the network state and leaf nodes of a multi-point service that the segment is part of.

Replication State 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. A Downstream Node is represented by a SID-list or a Segment Routing Policy [I-D.ietf-spring-segment-routing-policy] that specifies the explicit

path from the Replication Node to the Downstream Node, or even represented by another Replication segment. The SID-list MAY just have one SID. If a downstream node is adjacent to a Replication node, it MAY also be represented by an interface.

Replication SID identifies the Replication segment in the forwarding plane. At a Replication node, the Replication SID is the equivalent of Binding SID [I-D.ietf-spring-segment-routing-policy] of a Segment Routing Policy.

A packet steered into a Replication segment at a Replication node is replicated to each Downstream Node with the Downstream Replication SID that is relevant at that node. A packet is steered into a Replication Segment in two ways:

- When the Active Segment [RFC8402] is the Replication SID. In this case, the operation is NEXT followed by a PUSH for a replicated copy.

- On the root of a multi-point service, based on local policy-based routing. In this case, the operation for a replicated copy is PUSH.

If a Downstream Node is an egress (aka leaf) of the multi-point service, i.e. no further replication is needed, then that leaf node's Replication segment will not have any Replication State and the operation is NEXT. At an egress node, the Replication SID MAY be used to identify that portion of the multi-point 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 multi-point service at the same time [I-D.ietf-pim-sr-p2mp-policy]. In this case, the Replication segment's Replication State includes a branch with the Downstream Node being itself and the operation for the replicated copy is NEXT.

The Replication SID MUST be the last SID (at the bottom of stack for SR-MPLS) in a packet that is steered out from a Replication node of a Replication Segment. The behavior at Downstream nodes of a Replication Segment is undefined If there are any SIDs after the Replication SID and is outside the scope of this document.

## 2.1. SRv6

SRv6 network programming [I-D.ietf-spring-srv6-network-programming] introduces concept of functions. A function defines local behavior

on a node and is identified by opaque function part of a SRv6 SID. Familiarity with SRv6 Network Programming is expected.

In SRv6, a Replication Segment can be realized by defining a SRv6 Segment Endpoint behavior for replication. End.Replicate is an Endpoint function for replicating packets and, if required, decapsulation and processing of next header. This function is bound to a local SRv6 Replication SID at the Replication Node and Downstream Nodes of a Replication segment. FUNCT part of a Replication SID can represent both replication function as well the Replication State of a specific Replication Segment, or the Replication state MAY be represented by ARG part of Replication SID. For example, assuming two Replication Segments, RS1 and RS2 at a node, the node can bind two functions 0x00F1 and 0x00F2 (F=16, A=0) to End.Replicate function on Replication Segments RS1 and RS2 respectively. The node can also choose to bind one function 0x00FA with End.Replicate and ARGs 0x0001 and 0x0002 (F=16, A=16) to RS1 and RS2 respectively.

A Replication Node will replicate packet matching local SRv6 Replication SID to all Downstream Nodes. Each replication is equivalent to pushing segment list of an SRv6 policy to a Downstream Node, If there is only one SID, the Downstream Replication SID and there is no need to use any Flag, Tag or TLV, the SRH MAY be omitted and the Downstream Replication SID is set as IPv6 DA in replicated copy of packet. In this case, the LOC part of routed Downstream Replication SID takes packet from Replication Node to the Downstream Node. If an SRH is inserted in a replicated copy of packet, the Downstream Replication SID MUST be the last Segment in SRH i.e at Segment List index 0.

If a Downstream Node is an egress (aka leaf) of the multi-point service, i.e. no further replication is needed, then that leaf node's Replication segment will not have any Replication State and the operation on packet wtih local Replication SID is decapsulation with processing of next header equivalent to End.DT46.

A bud node performs both the replication and decapsulation part of End.Replicate function on a packet with local Replication SID.

H.Encaps.Replicate is behavior on the root of a multipoint service to steer a packet into a SRv6 Replication Segment.

Considerations of SRv6 Small SID/Compresion SID for SRv6 Replication SID will be addressed in future revision of this document.

### 2.1.1. End.Replicate: Replicate and/or Decapsulate

The "Endpoint with replication and/or decapsulate behavior (End.Replicate for short) is variant of End behavior.

We define a generic Replicate function on a packet for Replication State (RS).

```
S01. Replicate(RS, packet)
S02. {
S03.    For each Replication R with Downstream Replication SID, R-SID {
S04.       Make copy of packet
S05.       If (NumSID(R)== 1) {
S06.          Set IPv6 DA = R-SID
S07.          Set NH-Header in copy to Next-Header value of packet
S08.       } Else {
S09.         Insert SRH with R-SID at SID List[0] followed by other SIDS
S10.         Set NH-Header of SRH to Next-Header value of packet
S10.         Set IPv6 DA = First SID of R
S11.         Set NH-Header in copy to SRH
S12.       }
S13.       Submit the packet to the egress IPv6 FIB lookup and
           transmission to the new destination
S14. }
```

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

```
S01.   Lookup FUNCT OR (FUNCT,ARG) portion of S to get Replication State RS
S02.   Call Replicate(RS, packet)
S03.   If NH==SRH and SL != 0 {
S04.      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.
S05.   } Else If "decap check" success: {
S06.      Process packet according to End.DT46 behavior in SRv6 Network
Programming
S07.   } Else {
S08.      Drop packet
S09.   }
```

Notes:
The "decap check" would succeed on egress or bud node. The SRv6 Replication SID is bound to a specific tenant table at these nodes.

### 2.1.2. H.Encaps.Replicate: SR Headend encapsulation in Replication Segment

Node N receives two packets P1=(A, B2) and P2=(A,B2)(B3, B2, B1; SL=1). B2 is neither a local address nor SID of N.

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

N steers the transit packets P1 and P2 into an SRv6 Replication Segment, R, with a Source Address T and Replication State RS..

The H.Encaps.Replicate encapsulation behavior is defined as follows:

S01. Push an IPv6 header
S02. Set outer IPv6 SA = T
S03. Set outer Payload Length, Traffic Class, Hop Limit and Flow Label fields
S04. Set the outer Next-Header value
S05. Decrement inner IPv6 Hop Limit or IPv4 TTL
S06. Call Replicate(RS, Outer packet)

After the H.Encaps behavior, assuming a directly adjacent Downstream Node with Downstream Replication SID, D-RSID, P1' and P2' respectively look like:

- (T, D-RSID) (A, B2)

- (T, D-RSID) (A, B2) (B3, B2, B1; SL=1)

After the H.Encaps behavior, assuming a non-adjacent Downstream Node with Downstream Replication SID, D-RSID and a Segment list <S1, S2> to reach Downstream Node, P1' and P2' respectively look like:

- (T, S1) (D-RSID, S2, S1; SL=2) (A, B2)

- (T, S1) (D-RSID, S2, S1; SL=2) (A, B2) (B3, B2, B1; SL=1)

## 3. Use Cases

In the simplest use case, a single Replication segment includes the root node of a multi-point service and the egress/leaf nodes of the the service as all the Downstream Nodes. This achieves Ingress Replication [RFC7988] that has been widely used for MVPN [RFC6513] and EVPN [RFC7432] BUM (Broadcast, Unknown and Multicast) traffic.

Replication segments can also be used as building blocks for replication trees when Replication segments on the root, intermediate replication nodes and leaf nodes are stitched together to achieve efficient replication. That is specified in [I-D.ietf-pim-sr-p2mp-policy].

## 4. IANA Considerations

This document requires registration of End.Replicate behavior in "SRv6 Endpoint Behaviors" sub-registry of "Segment Routing Parameters" top-level registry.

| Value | Hex | Endpoint behavior | Reference |
|---|---|---|---|
| TBD | TBD | End.Replicate | [This.ID] |
| TBD | TBD | End.Replicate with ARG | [This.ID] |

Table 1: IETF - SRv6 Endpoint Behaviors

## 5. Security Considerations

There are no additional security risks introduced by this design.

## 6. Acknowledgements

The authors would like to acknowledge Siva Sivabalan, Mike Koldychev, Vishnu Pavan Beeram, Alexander Vainshtein, Bruno Decraene and Joel Halpern for their valuable inputs.

## 7. 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
US

Email: arvvenka@cisco.com

Zafar Ali
Cisco Systems, Inc.
US

Email: zali@cisco.com

Swadesh Agrawal
Cisco Systems, Inc.
San Jose
US

Email: swaagraw@cisco.com

Jayant Kotalwar
Nokia
Mountain View
US

Email: jayant.kotalwar@nokia.com

Tanmoy Kundu
Nokia
Mountain View
US

Email: tanmoy.kundu@nokia.com

Andrew Stone
Nokia
Ottawa
Canada

Email: andrew.stone@nokia.com

Tarek Saad
Juniper Networks
Canada

Email:tsaad@juniper.net

# 8. References

## 8.1. Normative 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-08 (work in progress), July 2020.

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

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

[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>.

## 8.2. Informative References

[I-D.filsfils-spring-srv6-net-pgm-illustration]
Filsfils, C., Camarillo, P., Li, Z., Matsushima, S., Decraene, B., Steinberg, D., Lebrun, D., Raszuk, R., and J. Leddy, "Illustrations for SRv6 Network Programming", draft-filsfils-spring-srv6-net-pgm-illustration-03 (work in progress), September 2020.

[I-D.ietf-pim-sr-p2mp-policy]
Voyer, D., Filsfils, C., Parekh, R., Bidgoli, H., and Z. Zhang, "Segment Routing Point-to-Multipoint Policy", draft-ietf-pim-sr-p2mp-policy-00 (work in progress), July 2020.

[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>.

## Appendix A. Illustration of a Replication Segment

This section illustrates an example of a single Replication Segment. Examples showing Replication Segment stitched together to form P2MP tree (based on SR P2MP policy) are in [I-D.ietf-pim-sr-p2mp-policy].

Consider the following topology:

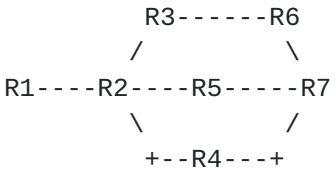


Figure 1

### A.1. SR-MPLS

In this example, the Node-SID of a node Rn is N-SIDn and Adjacency-SID from node Rm to node Rn is A-SIDmn. Interface between Rm and Rn is 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 Segment state at nodes R1, R2, R6 and R7 is shown below. Note nodes R3, R4 and R5 do not have state for the 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 packet directly to R2 on interface L12. Replication to R6, using N-SID6, steers packet via IGP 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:

- Since R1 is directly connected to R2, R1 performs PUSH operation with just <R-SID2> label for the replicated copy and sends it to R2 on interface L12. R2, as Leaf, performs NEXT operation, pops R-SID2 label and delivers the payload.

- R1 performs PUSH operation with <N-SID6, R-SID6> label stack for the replicated copy to R6 and sends it to R2, the nexthop on IGP shortest path to R6. R2 performs 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 and the packet is then sent to R6 with <R-SID6> in the label stack. R6, as Leaf, performs NEXT operation, pops R-SID6 label and delivers the payload.

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

pops A-SID47, and delivers packet to R7 with <R-SID7> in the label stack. R7, as Leaf, performs NEXT operation, pops R-SID7 label and delivers the payload.

## A.2. SRv6

For SRv6 , we use SID allocation scheme, reproduced below, from Illustrations for SRv6 Network Programming [I-D.filsfils-spring-srv6-net-pgm-illustration]

2001:db8::/32 is an IPv6 block allocated by a 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 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 PSP support

Function :Cn:: (function Cn, for short) represents the End.X function from to Node n

Each node k has:

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

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

An explicit SID instantiation 2001:db8:cccc:k:Fk::/128 bound to an End.Replcate 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.Replcate function, is 2001:db8:cccc:k:Fk::/128

with ARG value 0. A packet replicated from R1 to R7 has to traverse R4.

The Replication Segment state at nodes R1, R2, R6 and R7 is shown below. Note nodes R3, R4 and R5 do not have state for the Replication Segment.

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, 2001:db8:cccc:7:F7::0>
```

Replication to R2 steers packet directly to R2 on interface L12. Replication to R6, using 2001:db8:cccc:6:F6::0, steers packet via IGP shortest path to that node. Replication to R7 is steered via R4, using 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>
```

At R1, a H.Encaps.Replicate behavior is associated with the replication segment. When a packet, (A,B2), is steered into the replication segment at R1:

- Since R1 is directly connected to R2, R1 creates encapsulated replicated copy (2001:db8::1, 2001:db8:cccc:2:F2::0) (A, B2), and

sends it to R2 on interface L12. R2, as Leaf, executes decapsulation operation of End.Replicate, removes outer IPv6 header and delivers the payload.

- R1 creates 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, executes decapsulation operation of End.Replicate, removes outer IPv6 header and delivers the payload.

- R1 created 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, the nexthop on IGP shortest path to 2001:db8:cccc:4::/64. R2 forwards packet to R4 using 2001:db8:cccc:4::/64. R4 executes End.X function on 2001:db8:cccc:4:C7::0, performs PSP action, removes SRH and sends resulting packet (2001:db8::1, 2001:db8:cccc:7:F7::0) (A, B2) to R4. R7, as Leaf, executes decapsulation operation of End.Replicate, removes outer IPv6 header and delivers the payload.

## Authors' Addresses

Daniel Voyer (editor)
Bell Canada
Montreal
CA

Email: daniel.voyer@bell.ca

Clarence Filsfils
Cisco Systems, Inc.
Brussels
BE

Email: cfilsfil@cisco.com

Rishabh Parekh
Cisco Systems, Inc.
San Jose
US

Email: riparekh@cisco.com

Hooman Bidgoli
Nokia
Ottawa
CA

Email: hooman.bidgoli@nokia.com

Zhaohui Zhang
Juniper Networks

Email: zzhang@juniper.net