Abstract

This document describes the SR policy architecture for P2MP service delivery.

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 November 15, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.
This document defines variants of the SR Policy [I-D. ietf-spring-segment-routing-policy] to support Point-to-Multipoint service delivery.

We define a Point-to-Multipoint (P2MP) segment, which connects a Root node to a set of Leaf nodes in a Segment Routing Domain.

We also define a Replication Segment, which corresponds to the state of a P2MP segment on a particular node.
A P2MP segment consists of replication segments for the root, leaves, and optionally intermediate replication nodes. Note that a node may forward only one copy to a downstream node (be it a leaf or another intermediate node) or even just forward traffic off the p2mp segment (i.e. as a leaf), but we still call the forwarding behavior on the node a replication segment.

For a P2MP segment, a controller may be used to compute paths from a Root node to a set of Leaf nodes, optionally via a set of replication nodes. A packet is replicated at the root node and optionally on Replication nodes towards each Leaf node.

A Point-to-Multipoint service delivery could be via Ingress Replication (aka Spray in some SR context), i.e., the root unicasts individual copies of traffic to each leaf. The corresponding P2MP segment consists of replication segments only for the root and the leaves.

A Point-to-Multipoint service delivery could also be via Downstream Replication (aka TreeSID in some SR context), i.e., the root and some downstream replication nodes replicate the traffic along the way as it traverses closer to the leaves.

Notice that Spray is actually a special form of TreeSID. Also notice that, the explicit path from the root or a replication node to a leaf or a downstream replication node can optionally be partially or completely specified by the controller or determined locally.

2. SR Replication Policy

An SR Replication policy is a variant of an SR policy [I-D.ietf-spring-segment-routing-policy]. A replication policy corresponds to a replication segment, which defines the forwarding behavior on a particular node on a particular P2MP segment.

An SR Replication Policy can be either provisioned locally or programmed by a controller.

An SR Replication Policy is identified through the tuple <Node-ID, Root, Tree-ID>.

An SR Replication Policy is defined by following elements:

- Node-ID: The node that the replication segment is for.
- Root: The root of the P2MP segment that the replication segment is for.
o Tree-ID: Tree that the replication segment is part of.

o Replication-SID: Segment ID for this Replication Segment.

o Candidate Paths: See below.

The Replication-SID is instantiated into the forwarding plane at the node. An incoming packet with the SID is forwarded according to the replication branches. The Replication-SID may be the same on all nodes of the tree, and referred to as Tree-SID.

A SR Replication Policy may comprise of multiple candidate paths. The active candidate path is selected based on the tie breaking rules amongst the valid candidate-paths.

Each candidate path includes a list of replication branches. In this document, each branch is abstracted to a <Downstream Node, Downstream Replication-SID> tuple. For the signaling from a controller to a tree node, the Downstream Node in the tuple could be represented by its Node-SID (i.e. it does not matter how traffic gets to the downstream node, whether it’s directly connected or not), or in case of a directly connected Downstream Node it could be represented by one of this node’s Adjacency-SIDs (for the interface connecting to the directly connected Downstream Node). Alternatively, the Downstream Node could also be expanded to a SID-list that partially/fully specify the explicit path to it. In all cases, the node converts the signaled SIDs to its local forwarding representation (e.g., a Node/Adjacency-SID of a directly connected Downstream Node is translated to a local interface).

Each replication branch may also include one or more backup branches for protection purpose. Details will be added in a future revision.

3. SR P2MP Policy

The SR P2MP policy is a variant of an SR policy [I-D.ietf-spring-segment-routing-policy]. It correspond to an SR P2MP Segment.

A SR P2MP Policy is defined by following elements:

o Root node: This is the headend of the P2MP segment.

o Leaf nodes: A set of nodes that terminate the P2MP segment.

o Constraints/Objectives: Optional set of topological/resource constraints and optimization objectives to be satisfied by the P2MP segment.
A SR P2MP Policy is identified through the tuple <Root node, Tree-ID>.

An SR P2MP Policy has a BSID [I-D.ietf-spring-segment-routing-policy] instantiated into the forwarding plane. The BSID is applicable only at the Root node.

An SR P2MP policy can be either provisioned locally or programmed by a controller onto the root node of the segment, for the purpose of steering traffic into the segment. A controller calculates the tree and program corresponding replication segments on root, leaves and optional replication nodes.

Traffic is steered into a SR P2MP Policy in two ways:

- Based on a local policy-based routing at the Root node.
- Based on remote classification and steering via the BSID of the SR P2MP Policy at the Root node.

Traffic is then forwarded toward the leaves following the replication segments.

4. Using Controller to build a P2MP Segment

A P2MP segment can be built using a Path Computation Element (PCE) and PCE Protocol (PCEP). This section outlines a high-level architecture for such an approach.
4.1. SR P2MP Policy Creation

A SR P2MP policy can be instantiated and maintained in a centralized fashion using a Path Computation Element (PCE).

4.1.1. API

North-bound APIs on a PCE can be used to:

1. Create P2MP SR policy
2. Delete P2MP SR policy
3. Update P2MP SR policy

4.1.2. Invoking API

Operator shall interact with a PCE via REST, Netconf, gRPC, CLI. Yang model shall be developed for this purpose as well.
4.2. P2MP Segment Computation

Network operator passes the addresses of the root (R) and set of leaves {L} as well as Traffic Engineering (TE) attributes (e.g., constraints such as link color, optimization criteria such as latency) of the P2MP segment to PCE via a suitable North-Bound API. The PCE computes the tree instantiates the P2MP segment on Root, Replication, and Leaf nodes.

Path constraints shall include link color affinity, bandwidth, disjointness (link, node, SRLG), delay bound, link loss, etc. Path shall be optimized based on IGP or TE metric or link latency.

Ideally, same P2MP SID SHOULD be used for forwarding entries at Root, Mid, and Leaf nodes. Different P2MP SIDs MAY be used at different node(s) if it is not feasible to use same P2MP SID. SIDs (BSID as well as P2MP SID) can also be assigned by operator.

A PCE can modify a P2MP segment following network element failure or in case a better path can be found based on the new network state. In this case, the PCE may want to setup the new tree and remove the old tree from the network in order to minimize traffic loss. As such, a separate P2MP SID can be used for the new tree.

A PCE shall be capable of computing paths across multiple IGP areas or levels as well as Autonomous Systems (ASs).

4.2.1. Topology Discovery

A PCE shall learn network topology, TE attributes of link/node as well as SIDs via dynamic routing protocols (IGP and/or BGP-LS). It may be possible for operators to pass topology information to PCE via north-bound API.

4.2.2. Capability and Attribute Discovery

It shall be possible for a node to advertise TreeSID capability via IGP and/or BGP-LS. Similarly, a PCE can also advertise its TreeSID capability via IGP and/or BGP-LS. Capability advertisement allows a network node to dynamically choose one or more PCE(s) to obtain services pertaining to SR P2MP policies, as well a PCE to dynamically identify TreeSID capable nodes.

4.3. Instantiating P2MP segment nodes

Once a PCE computes a tree for P2MP segment, it needs to instantiate the segment on the relevant network nodes. The PCE can use various
protocols to program the forwarding entries, and these protocols are described below.

4.3.1. PCEP

PCE Protocol (PCEP) has been traditionally used:

1. For a head-end to obtain paths from a PCE.
2. A PCE to instantiate SR policies.

PCEP protocol can be stateful in that a PCE can have a stateful control of an SR policy on a head-end which has delegated the control of the SR policy to the PCE. PCEP shall be extended to provision and maintain forwarding entries in a stateful fashion.

4.3.2. BGP

BGP has been extended to instantiate and report SR policies. It shall be used to instantiate and maintain forwarding entries for SR P2MP policies.

4.3.3. NetConf

TBD

4.4. Protection

4.4.1. Local Protection

A network link/node on the tree of a P2MP segment can be protected using SR policies computed by PCE. The backup SR policies shall be programmed in forwarding plane in order to minimize traffic loss when the protected link/node fails.

4.4.2. Path Protection

It is possible for PCE create a disjoint backup tree for providing end-to-end path protection.

5. IANA Considerations

This document makes no request of IANA.
6. Security Considerations

There are no additional security risks introduced by this design.

7. Acknowledgements

The authors would like to acknowledge Siva Sivabalan.

8. 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
9. Normative References

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d.,
bogdanov@google.com, b., and P. Mattes, "Segment Routing
Policy Architecture", draft-ietf-spring-segment-routing-
policy-03 (work in progress), May 2019.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,

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