SPRING Working Group
Internet Draft
Intended status: Standards Track
Expires: August 2016

Dave Allan, Jeff Tantsura
Ericsson

February 2016

A Framework for Computed Multicast applied to MPLS based Segment Routing
draft-allan-spring-mpls-multicast-framework-00

Abstract

This document describes a multicast solution for Segment Routing with MPLS data plane. It is consistent with the Segment Routing architecture in that an IGP is augmented to distribute information in addition to the link state. In this solution it is multicast group membership information sufficient to synchronize state in a given network domain. Computation is employed to determine the topology of any loosely specified multicast distribution tree.

Table of Contents

1. Introduction.....3
1.1. Authors.....3
1.2. Requirements Language.....3
2. Conventions used in this document.....3
2.1. Terminology.....3
3. Solution Overview.....4
3.1. Mapping source specific trees onto the segment routing architecture.....5
3.2. Role of the Routing System.....5
3.3. MDT Construction Requirements.....6
3.4. Pruning - theory of operation.....6
4. Elements of Procedure.....7
4.1. Triggers for Computation.....7
4.2. FIB Determination.....7
4.2.1. Information in the IGP.....7
4.2.2. Computation of individual segments.....7
4.3. FIB Generation.....10
4.4. FIB installation.....10
5. Related work.....11
5.1. IGP Extensions.....11
5.2. BGP Extensions.....11
6. Observations.....11
7. Acknowledgements.....12
8. Security Considerations.....12
9. IANA Considerations.....12
10. References.....12
10.1. Normative References.....12
10.2. Informative References.....12
11. Authors' Addresses.....13

1. Introduction

This memo describes a solution for multicast for Segment Routing with MPLS data plane in which source specific multicast distribution trees (MDTs) are computed from information distributed via an IGP. Computation can use information in the IGP to determine if a given node in the network has a role as a root, leaf or replication point in a given MDT. Unicast tunnels are employed to interconnect the nodes determined to have a role. Therefore state only need be installed in nodes that have one of these three roles to fully instantiate an MDT.
Although this approach is computationally intensive, a significant amount of computation can be avoided when the computing agent determines that the node it is computing for has no role in a given MDT. This permits a computed approach to multicast convergence to be computationally tractable.

1.1. Authors

David Allan, Jeff Tantsura

1.2. 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 [RFC2119].

2. Conventions used in this document

2.1. Terminology

Candidate replication point - is a node that potentially needs to install state to replicate multicast traffic as determined at an intermediate step in multicast segment computation. It will either resolve to having no role or a role as a replication point once multicast has converged.

Candidate role - refers to any potential combination of roles on a given multicast segment as determined at some intermediate step in MDT computation. For example, a node with a candidate role may be a leaf and may be a candidate replication point.

Downstream - refers to the direction along the shortest path to one or more leaves for a given multicast distribution tree

Multicast convergence - is when all computation and state installation to ensure the FIB reflects the multicast information in the IGP is complete.

MDT - multicast distribution tree. Is a tree composed of one or more multicast segments.

Multicast segment - is a portion of the multicast tree where only the root and the leaves have been specified, and computation based upon the current state of the IGP database will be employed to determine and install the required state to implement the segment. A multicast segment is identified by a multicast SID.

Pinned path - Is a unique shortest path extending from a leaf upstream towards the root for a given multicast segment. Therefore is a component of the multicast segment that it has been determined must be there. It will not necessarily extend from the leaf all the way to the root during intermediate computation steps. A pinned path can result from pruning operations.

Role - refers specifically to a node that is either a root, a leaf, a replication node, or a pinned waypoint for a given MDT.

Unicast convergence - is when all computation and state installation to ensure the FIB reflects the unicast information in the IGP is complete.

Upstream - refers to the direction along the shortest path to the root of a given MDT.

3. Solution Overview

This memo describes a multicast architecture in which multicast state is only installed in those nodes that have roles as a root, leaves, and replication points for a given multicast segment. The a-priori established segment routing unicast tunnels are used as interconnect between the nodes that have a role in a given multicast SID.

A loosely specified MDT is composed of a single multicast segment and the routing of the MDT is delegated entirely to computation driven by information in the IGP database.

Explicitly routed MDTs are expressed as a tree of concatenated multicast segments where both the leaves of each segment and the waypoints coupling a given segment to the upstream and/or downstream segment(s) is specified in information flooded in the IGP by the

overall root of the MDT. The segments themselves will be computed as per a loosely specified MDT.

A PE acting as an overall root for a given tree is expected to be configured by management as to where to source multicast traffic from, be it an attachment circuit, interworking function for client technology or other. Similarly a leaf for a given tree is expected to be configured by management as to the disposition of received multicast traffic.

A computed segment is guaranteed to be loop free in a stable system. A concatenation of segments to construct an MDT will similarly be loop free as any collision of segments can be disambiguated in the data plane via the SIDs.

This architecture significantly reduces the amount of state that needs to be installed in the data plane to support multicast. This also means that the impact of many failures in the network on multicast traffic distribution will be recovered by unicast local repair or unicast convergence with subsequent multicast convergence acting in the role of network re-optimization (as opposed to restoration).

3.1. Mapping source specific trees onto the segment routing architecture

A computed source specific tree for a given multicast group corresponds to one or more multicast segments in the SR architecture, each of which is assigned a SID, typically by management configuration of the node that will be the overall root for the source specific tree, which then uses the IGP to advertise this information to the root"s peers.

A multicast group is implemented as the set of source specific trees from all nodes that have registered transmit interest to all nodes that have registered receive interest in a multicast group.

3.2. Role of the Routing System

The role of the IGP is to communicate topology information, multicast registrations, unicast to SID bindings, multicast to SID bindings and waypoints in multi-segment MDTs. No changes to topology or unicast to SID bindings advertisement are proposed by this memo.

The multicast registrations/bindings will be in the form of source, group, transmit/receive interest and the SID to use for the source specific multicast tree. Registrations are originated by any node that has send or receive interest in a given multicast group. Nodes

will use the combination of topology and multicast registrations to determine the nodes that have a role in each source specific tree and the SID information to then derive the required FIB state.

The definition of the required IGP TLVs is out of scope of this memo and will be done in relevant IGP drafts.

3.3. MDT Construction Requirements

A multicast segment in an MDT is constructed such that between any pair of nodes that have a role in the segment and are connected by a unicast tunnel, there is not another node on the shortest path between the two with a role in that segment. This ensures that copies of a packet forwarded by an multicast segment will traverse a link only once in a stable system.

Note that this can be satisfied by a minimum cost shortest path tree, but is not an absolute requirement. The pruning rules specified in this memo will meet this requirement without necessarily producing absolutely minimum cost multicast segment (or incurring the associated computational cost).

3.4. Pruning - theory of operation

The role of nodes in a given multicast segment is determined by first producing an inclusive shortest path tree with all possible paths between the root and leaves, and then applying a set of pruning rules repeatedly until an acyclic tree is produced or no further prunes are possible.

For the majority of multicast segments these rules will authoritatively produce a minimum cost tree. For those segments that have not yet been authoritatively resolved, there is a set of pruning operations applied that are not guaranteed to produce a tree that meets the requirements of 3.3, therefore these trees require auditing and potential correction according to a further set of agreed rules. This avoids the necessity of an exhaustive search of the solution space.

A node during computation of a segment may conclude that it will absolutely not have a role at any of numerous points in the computation process and abandon computation of that segment.

4. Elements of Procedure

4.1. Triggers for Computation

MDT computation is triggered by changes to the IGP database. These are in the form of either changes in registered multicast group interest, addition or removal of a multi-segment MDT descriptor, or topology changes.

A change in registered interest for a group will require re-computation of all MDTs that implement the multicast group.

A topology change will require the computation of some number of multicast segments, the actual number will depend on the implementation of tree computation but at a minimum will be all trees for which there is not an optimal shortest path solution as a result of the topology change.

4.2. FIB Determination

4.2.1. Information in the IGP

Group membership information for a multicast segment is obtained from the IGP. This is true for single segment MDTs as well as multi-segment MDTs. Included in the multi-segment MDT specification is the waypoint nodes in MDT and the upstream and downstream SIDs. The specified node is expected to cross connect the SIDs to join the segments together acting in the role of leaf for the upstream segment and root for the downstream segment.

When a waypoint in an MDT descriptor does not exist in the IGP, the assumption is that the node has failed. The response of the other nodes in the system in FIB determination is to add the leaves of the downstream segment to the upstream segment.

4.2.2. Computation of individual segments

FIB generation for a multicast segment is the result of computation, ultimately as applied to all source specific trees in the network. All computing nodes implement a common algorithm for tree generation, as all MUST agree on the solution.

One algorithm is as follows:

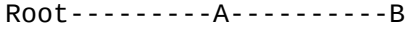
All possible shortest paths to the set of leaves for the MDT is determined. Then pruning rules are repeatedly applied until no further prunes are possible.

The philosophy of the application of these rules could be expressed as "simplify as much as possible, and prune that which cannot be". The rules are:

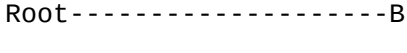
1) Eliminate any links and nodes not on a potential shortest path from the root to the leaves for the MDT under consideration.

2) Simplify via the replacement of any nodes that do not have a potential role in the MDT with links.

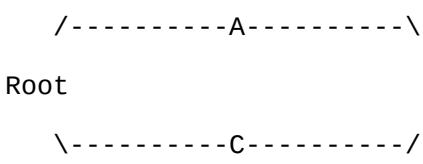
This will be nodes that are not a leaf, a root or a candidate replication point. For example:



B is a leaf. A is not but is in a potential shortest path from root to B. However A will have no role in the MDT that serves B as it provides simple transit therefore is replaced with a direct connection between the root and B.



Note that such pruning also needs to avoid the creation of duplicate links. For example:

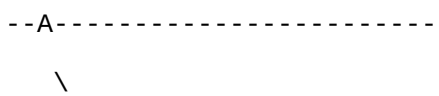


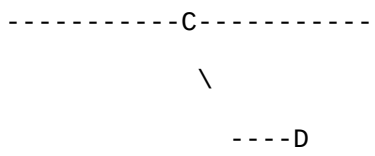
Where A and C have no role, they can be replaced with a single link from Root to B.

3) Simplify via the elimination of fewer hop paths

When for a given set of leaves, a node has multiple downstream links that converge on a common downstream point, and that set of leaves is only a subset of the leaves reachable on one or more of the links, any link that only serves that subset of leaves can be pruned.

For example:





B and D are leaves of a root upstream of A. From A, link AB can reach leaf B. Path AC can reach leaf B and D. In this case path A-B can be pruned from consideration. The set of leaves reachable via link A-B is a subset of that reachable by A-C, and the paths from A that serves that subset converges at B.

4) Prune via the elimination of upstream links where the nearest reachable leaf is further than the closest leaf or pinned path, and that path does not have a candidate replication point closer than the closet leaf or pinned path, as the resulting tree will require the shortest path to transit the closest upstream leaf or pinned path.

For each upstream link for each leaf in a segment the nearest leaf or pinned path is determined. Those links for which the nearest leaf is further upstream than the closest leaf are pruned.

If, at the end of pruning and simplification, all leaves in a multicast segment have a unique shortest path to the root, the tree is considered resolved, and the computation can progress directly to the FIB generation step.

If not all leaves have a unique shortest path, additional pruning steps are applied. These steps are NOT guaranteed to produce a lowest cost tree, and therefore require an additional audit and possible modification to ensure when forwarding a maximum of one copy of a packet will traverse an interface.

For segments not authoritatively resolved by the above rules, a prune that will not authoritatively result in a minimum cost tree is applied. For the purpose of interoperability, the following rule is proposed: A computing node will select the closest node to the root with a candidate role that does not have a unique shortest path to the root. Where more than one such node exists, the one with the lowest unicast SID is selected. For that node, the best upstream link is selected and all other upstream links pruned. The best upstream link is defined as the link with the closest node with a candidate role that potentially serves the highest number of leaves. Where there is a tie, once again the node with the lowest SID is selected.

Once the links have been pruned, rules 2 through 4 are repeatedly applied until either the tree is fully resolved, or again no further prunes are possible, in which case the next closest remaining unresolved node has the same prune applied.

For all segments not resolved by the initial prune rules, they are audited to ensure all nodes that have a role in the tree do not have a node with a role between them and their upstream node on the tree. If they do, the old upstream adjacency is removed, and the superior one added.

4.3. FIB Generation

The topology components that remain at the end of the pruning operation will reflect all nodes that have a role in a given multicast segment plus the necessary tunnels (as all intervening multi-path scenarios will have been simplified away). From this the FIB can be generated:

All nodes that have a role in a given multicast segment and have nodes upstream in the segment will need to accept the SID for the MDT from at minimum, all upstream interfaces.

All nodes that have a role in a given segment and have nodes immediately downstream in the segment will need to replicate packets simply labelled with the multicast SID onto those interfaces.

All nodes that have a role in a given segment and have nodes reachable via a tunnel downstream set the FIB to push the tunnel unicast SID for the downstream node onto any replicated copies of a received packet, and identify the set of interfaces on the shortest path for the tunnel SID.

4.4. FIB installation

FIB installation needs to acknowledge two aspects of the hybrid tunnel and role model of multicast tree construction. The first is that because of the sparse state model simple tree adds, moves, and changes may require the installation of state where it did not previously exist, and such changes may impact existing services. The second is that it is possible to retain the knowledge to prioritize computation of those trees impacted the failure of a node with a role.

To address this, there are three stages of state installation for multicast convergence:

1) Immediate:

a. Installation of state for multicast segments impacted by the failure of a node in the network, and installation of state for segments in nodes that have not previously had a role in the given segment.

b. Installation of state for waypoints in multi-segment MDTs.

2) After T1: Update state for nodes that both had and have a role in a given multicast segment.

3) After T2: Removal of state for nodes that transition from having a role to not having a role for a given multicast segment.

T1 and T2 will be network wide configurable values.

5. Related work

5.1. IGP Extensions

RFC 6329 provides a useful example of some of the type of IGP changes that will be required. There are two aspects in RFC 6329 that are worth emulating:

- The advertisement of multicast registrations

- The negotiation of the algorithm to be used for MDT computation

The required changes for both IS-IS and OSPF will be documented in separate WG targeted I-Ds.

5.2. BGP Extensions

This memo will require the specification of a new PMSI Tunnel Attribute (SPRING P2MP tunnel, tentatively 0x09) to order to integrate into the multicast framework documented in RFC 6514

6. Observations

This technique is not confined to segment routing, and with the provision of a global label space (to be employed as per a multicast SID), an MPLS-LDP network would also provide the requisite mesh of unicast tunnels and be capable of implementing this approach to multicast.

This memo focuses on an implementation based upon nodes that are IGP speakers and converge independently so is written in a form that assumes a node, computing node and IGP speaker are one in the same. It should be observed that the relative frugality of data plane state would suggest that separation of computation from nodes in the data plane combined with management or "software defined networking" based population of the multicast FIB entries may also be useful modes of network operation.

7. Acknowledgements

8. Security Considerations

For a future version of this document.

9. IANA Considerations

For a future version of this document.

10. References

10.1. Normative References

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

10.2. Informative References

[RFC6379] Ashwood-Smith et.al., "IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging", IETF RFC 6329, April 2012

[RFC6514] Aggarwal et.al., "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", IETF RFC 6514, February 2012

[RFC7385] Andersson & Swallow "IANA Registry for P-Multicast Service Interface (PMSI) Tunnel Type Code Points", IETF RFC 7385, October 2014

11. Authors' Addresses

Dave Allan (editor)
Ericsson
300 Holger Way
San Jose, CA 95134
USA
Email: david.i.allan@ericsson.com

Jeff Tantsura
Ericsson
200 Holger Way
San Jose, CA 95134
Email: jeff.tantsura@ericsson.com

SPRING Working Group
Internet Draft
Intended status: Standards Track
Expires: October 2017

Dave Allan
Ericsson
Jeff Tantsura

April 2017

A Framework for Computed Multicast applied to MPLS based Segment Routing
draft-allan-spring-mpls-multicast-framework-03

Abstract

This document describes a multicast solution for Segment Routing with MPLS data plane. It is consistent with the Segment Routing architecture in that an IGP is augmented to distribute information in addition to the link state. In this solution it is multicast group membership information sufficient to synchronize state in a given network domain. Computation is employed to determine the topology of any loosely specified multicast distribution tree.

Table of Contents

1. Introduction.....3
1.1. Authors.....3
1.2. Requirements Language.....3
2. Conventions used in this document.....3
2.1. Terminology.....3
3. Solution Overview.....4
3.1. Mapping source specific trees onto the segment routing architecture.....5
3.2. Role of the Routing System.....6
3.3. MDT Construction Requirements.....6
3.4. Pruning - theory of operation.....6
4. Elements of Procedure.....7
4.1. Triggers for Computation.....7
4.2. FIB Determination.....7
4.2.1. Information in the IGP.....7
4.2.2. Computation of individual segments.....8
4.3. FIB Generation.....11
4.4. FIB installation.....11
5. Related work.....12
5.1. IGP Extensions.....12
5.2. BGP Extensions.....12
6. Observations.....12
7. Acknowledgements.....13
8. Security Considerations.....13
9. IANA Considerations.....13
10. References.....13
10.1. Normative References.....13
10.2. Informative References.....13
11. Authors' Addresses.....14

1. Introduction

This memo describes a solution for multicast for Segment Routing with MPLS data plane in which source specific multicast distribution trees (MDTs) are computed from information distributed via an IGP. Computation can use information in the IGP to determine if a given node in the network has a role as a root, leaf or replication point in a given MDT. Unicast tunnels are employed to interconnect the nodes determined to have a role. Therefore state only need be installed in nodes that have one of these three roles to fully instantiate an MDT.
Although this approach is computationally intensive, a significant amount of computation can be avoided when the computing agent determines that the node it is computing for has no role in a given MDT. This permits a computed approach to multicast convergence to be computationally tractable.

1.1. Authors

David Allan, Jeff Tantsura

1.2. 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 [RFC2119].

2. Changes from the last version

Clarifications in the pruning and simplification algorithm

3. Conventions used in this document

3.1. Terminology

Candidate replication point (CRP) - is a node that potentially needs to install state to replicate multicast traffic as determined at an intermediate step in multicast segment computation. It will either resolve to having no role or a role as a replication point once multicast has converged.

Candidate role - refers to any potential combination of roles on a given multicast segment as determined at some intermediate step in MDT computation. For example, a node with a candidate role may be a leaf and may be a candidate replication point.

Downstream - refers to the direction along the shortest path to one or more leaves for a given multicast distribution tree

Multicast convergence - is when all computation and state installation to ensure the FIB reflects the multicast information in the IGP is complete.

MDT - multicast distribution tree. Is a tree composed of one or more multicast segments.

Multicast segment - is a portion of the multicast tree where only the root and the leaves have been specified, and computation based upon the current state of the IGP database is employed to determine and install the required state to implement the segment. For MPLS a multicast segment is implemented as a p2mp LSP. A multicast segment is identified by a multicast SID.

Multicast SID - Is the data plane identifier that is used to implement a multicast segment. As per a unicast MPLS segment, the rightmost 20 bits of a multicast SID is encoded as a label. It is drawn from an SRGB that is global to the SR domain.

Pinned path - Is a unique shortest path extending from a leaf upstream towards the root for a given multicast segment. Therefore is a component of the multicast segment that it has been determined must be there. It will not necessarily extend from the leaf all the way to the root during intermediate computation steps. A pinned path can result from pruning operations.

Role - refers specifically to a node that is either a root, a leaf, a replication node, or a pinned waypoint for a given MDT.

Unicast convergence - is when all computation and state installation to ensure the FIB reflects the unicast information in the IGP is complete.

Upstream - refers to the direction along the shortest path to the root of a given MDT.

4. Solution Overview

This memo describes a multicast architecture in which multicast state is only installed in those nodes that have roles as a root, leaves, and replication points for a given multicast segment. The a-priori established segment routing unicast tunnels are used as interconnect between the nodes that have a role in a given multicast SID.

A loosely specified MDT is composed of a single multicast segment and the routing of the MDT is delegated entirely to computation driven by information in the IGP database.

Explicitly routed MDTs are expressed as a tree of concatenated multicast segments where both the leaves of each segment and the waypoints coupling a given segment to the upstream and/or downstream segment(s) is specified in information flooded in the IGP by the overall root of the MDT. The segments themselves will be computed as per a loosely specified MDT.

A PE acting as an overall root for a given tree is expected to be configured by the operator as to where to source multicast traffic from, be it an attachment circuit, interworking function for client technology or other. Similarly a leaf for a given tree is expected to be configured by the operator as to the disposition of received multicast traffic.

A computed segment is guaranteed to be loop free in a stable system. A concatenation of segments to construct an MDT will similarly be loop free as any collision of segments can be disambiguated in the data plane via the SIDs.

This architecture significantly reduces the amount of state that needs to be installed in the data plane to support multicast. This also means that the impact of many failures in the network on multicast traffic distribution will be recovered by unicast local repair or unicast convergence with subsequent multicast convergence acting in the role of network re-optimization (as opposed to restoration).

4.1. Mapping source specific trees onto the segment routing architecture

A computed source specific tree for a given multicast group corresponds to one or more multicast segments in the SR architecture. Each multicast segment is assigned a SID, typically by management configuration of the node that will be the overall root for the source specific tree. The root node then uses the IGP to advertise this information to all nodes in the IGP area/domain.

A multicast group is implemented as the set of source specific trees from all nodes that have registered transmit interest to all nodes that have registered receive interest in a multicast group.

4.2. Role of the Routing System

The role of the IGP is to communicate topology information, multicast capability and associated algorithm, multicast registrations, unicast to SID bindings, multicast to SID bindings and waypoints in multi-segment MDTs. No changes to topology or unicast to SID binding advertisements are proposed by this memo.

The multicast registrations/bindings will be in the form of source, group, transmit/receive interest and the SID to use for the source specific multicast tree. Registrations are originated by any node that has send or receive interest in a given multicast group. Nodes will use the combination of topology and multicast registrations to determine the nodes that have a role in each source specific tree and the SID information to then derive the required FIB state.

4.3. MDT Construction Requirements

A multicast segment in an MDT is constructed such that between any pair of nodes that have a role in the segment and are connected by a unicast tunnel, there is not another node on the shortest path between the two with a role in that segment. This ensures that copies of a packet forwarded by an multicast segment will traverse a link only once in a stable system.

Note that this can be satisfied by a minimum cost shortest path tree, but is not an absolute requirement. The pruning rules specified in this memo will meet this requirement without necessarily producing absolutely minimum cost multicast segment (or incurring the associated computational cost).

4.4. Pruning - theory of operation

The role of nodes in a given multicast segment is determined by first producing an inclusive shortest path tree with all possible paths between the root and leaves, and then applying a set of pruning rules repeatedly until an acyclic tree is produced or no further prunes are possible.

For the majority of multicast segments these rules will authoritatively produce a minimum cost tree. For those segments that have not yet been authoritatively resolved, there is a set of pruning operations applied that are not guaranteed to produce a tree that meets the requirements of 3.3, therefore these trees require auditing and potential correction according to a further set of agreed rules. This avoids the necessity of an exhaustive search of the solution space.

A node during computation of a segment may conclude that it will absolutely not have a role at any of numerous points in the computation process and abandon computation of that segment.

5. Elements of Procedure

5.1. Triggers for Computation

MDT computation is triggered by changes to the IGP database. These are in the form of either changes in registered multicast group interest, addition or removal of a multi-segment MDT descriptor, or topology changes.

A change in registered interest for a group will require re-computation of all MDTs that implement the multicast group.

A topology change will require the computation of some number of multicast segments, the actual number will depend on the implementation of tree computation but at a minimum will be all trees for which there is not an optimal shortest path solution as a result of the topology change.

5.2. FIB Determination

5.2.1. Information in the IGP

Group membership information for a multicast segment is obtained from the IGP. This is true for single segment MDTs as well as multi-segment MDTs. Included in the multi-segment MDT specification is the waypoint nodes in MDT and the upstream and downstream SIDs. The specified node is expected to cross connect the SIDs to join the segments together acting in the role of leaf for the upstream segment and root for the downstream segment.

When a waypoint in an MDT descriptor does not exist in the IGP, the assumption is that the node identified by the waypoint SID has failed. The response of the other nodes in the system in FIB determination is to add the leaves of the downstream segment to the upstream segment.

An example of this would be consider a node "x", and another node "y". At some point in time, "x" advertises a tree that identifies "y" as a waypoint that cross connects upstream SID "a" to downstream SID "b". At some later point node "y" fails. The other nodes in the network will compute segment "a" as if it included all leaves and waypoints in segment "b". All apriori state installed for segment "b"

would be removed as the failure of "y" has required "b" to be subsumed by "a".

5.2.2. Computation of individual segments

FIB generation for a multicast segment is the result of computation, ultimately as applied to all source specific trees in the network. All computing nodes implement a common algorithm for tree generation, as all MUST agree on the solution.

One algorithm is as follows:

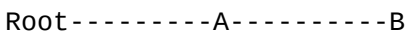
All possible shortest paths to the set of leaves for the MDT is determined. Then pruning rules are repeatedly applied until no further prunes are possible.

The philosophy of the application of these rules could be expressed as "simplify as much as possible, and prune that which cannot be". The rules are:

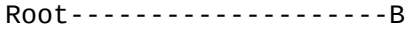
1) Eliminate any links and nodes not on a potential shortest path from the root to the leaves for the MDT under consideration.

2) Simplify via the replacement of any nodes that do not have a potential role in the MDT with links.

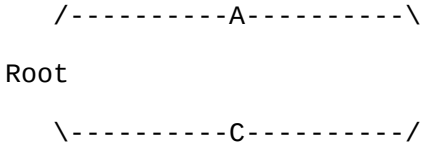
This will be nodes that are not a leaf, a root or a candidate replication point. For example:



B is a leaf. A is not but is in a potential shortest path from root to B. However A will have no role in the MDT that serves B as it provides simple transit therefore is replaced with a direct connection between the root and B.



Note that such pruning also needs to avoid the creation of duplicate parallel links. For example:

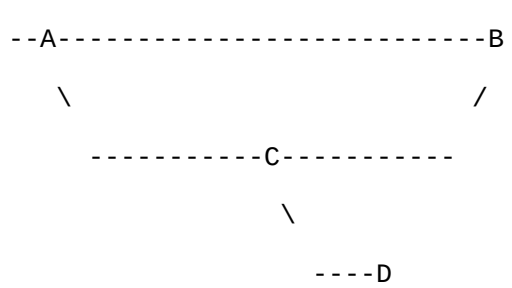


Where A and C have no role and the cost root-A-B = cost root-C-B, they can be replaced with a single link from Root to B.

3) Simplify via the elimination of fewer hop paths

When for a given set of leaves, a node has multiple downstream links that converge on a common downstream point, and that set of leaves is only a subset of the leaves reachable on one or more of the links, any link that only serves that subset of leaves can be pruned.

For example:



Link AB is cost 2, link AC and CB are cost 1 (cost of link CD does not affect the example).

B and D are leaves of a root upstream of A. From A, link AB can reach leaf B. Path AC can reach leaf B and D. In this case path A-B can be pruned from consideration. The set of leaves reachable via link A-B is a subset of that reachable by A-C, and the paths from A that serves that subset converges at B.

4) Prune upstream links.

The normal procedure is to determine the closest upstream leaf or pinned path and then compare all upstream adjacencies with that metric

a. If the upstream adjacency extends closer to the root than the closest leaf or pinned path, then that adjacency can be pruned.

b. If the upstream adjacency extends the same distance towards the root then

i. If it is to a non-leaf or pinned path candidate replication point, it can be pruned

ii. If it is to a pinned path, where there are equal upstream adjacencies that terminate on leaves, it can be pruned (considered inferior).

iii. If there is more than one "equal" upstream adjacency, that is all terminate on nodes that are on pinned paths, or all terminate on nodes that are leaves, than one is selected. This is via the lowest node ID.

c. If the upstream adjacency is a candidate replication point closer than the closest leaf, and upstream from it is a node that is a leaf or pinned path equidistant with the closest leaf, then all adjacencies that extend to leaves ranked lower than the leaf or pinned path behind the CRP may be pruned. Note that an upstream adjacency that has a CRP closer than the closest leaf or pinned path cannot be pruned.

d. When for a given node all possible upstream adjacencies that can be pruned have been identified, each is removed, and any simplifications that can be performed as a result of the prune are performed. This is the equivalent of a localized check for 2 and 3 above and is then performed iteratively in response to changes to the graph as a result of pruning.

The procedure is to implement 1, 2 and 3 above, then loop on 4 until such time as the MDT is fully resolved, or no further prunes are possible. Step 4 is performed in a specific order. The nodes are processed according to a ranking from closest to the root to the farthest, and from lowest node ID to the highest within a given distance from the root.

If, at the end of pruning and simplification, all leaves in a multicast segment have a unique shortest path to the root, the tree is considered resolved, and the computation can progress directly to the FIB generation step.

If not all leaves have a unique shortest path, additional pruning steps are applied. These steps are NOT guaranteed to produce a lowest cost tree, and therefore require an additional audit and possible modification to ensure when forwarding a maximum of one copy of a packet will traverse an interface.

For segments not authoritatively resolved by the above rules, a prune that will not authoritatively result in a minimum cost tree is applied. For the purpose of interoperability, the following rule is proposed: A computing node will select the closest node to the root with a candidate role that does not have a unique shortest path to

the root. Where more than one such node exists, the one with the lowest unicast SID is selected. For that node, the best upstream link is selected and all other upstream links pruned. The best upstream link is defined as the link with the closest node with a candidate role that potentially serves the highest number of leaves. Where there is a tie, once again the node with the lowest SID is selected.

Once the links have been pruned, rules 2 through 4 are repeatedly applied until either the tree is fully resolved, or again no further prunes are possible, in which case the next closest remaining unresolved node has the same prune applied.

For all segments not resolved by the initial prune rules, they are audited to ensure all nodes that have a role in the tree do not have a node with a role between them and their upstream node on the tree. If they do, the old upstream adjacency is removed, and the superior one added.

5.3. FIB Generation

The topology components that remain at the end of the pruning operation will reflect all nodes that have a role in a given multicast segment plus the necessary tunnels (as all intervening multi-path scenarios will have been simplified away). From this the FIB can be generated:

All nodes that have a role in a given multicast segment and have nodes upstream in the segment will need to accept the SID for the MDT from at minimum, all upstream interfaces.

All nodes that have a role in a given segment and have nodes immediately downstream in the segment will need to replicate packets simply labelled with the multicast SID onto those interfaces.

All nodes that have a role in a given segment and have nodes reachable via a tunnel downstream set the FIB to push the tunnel unicast SID for the downstream node onto any replicated copies of a received packet, and identify the set of interfaces on the shortest path for the tunnel SID.

5.4. FIB installation

FIB installation needs to acknowledge two aspects of the hybrid tunnel and role model of multicast tree construction. The first is that because of the sparse state model simple tree adds, moves, and changes may require the installation of state where it did not previously exist, and such changes may impact existing services. The

second is that it is possible to retain the knowledge to prioritize computation of those trees impacted the failure of a node with a role.

To address this, there are three stages of state installation for multicast convergence:

1) Immediate:

a. Installation of state for multicast segments impacted by the failure of a node in the network, and installation of state for segments in nodes that have not previously had a role in the given segment.

b. Installation of state for waypoints in multi-segment MDTs.

2) After T1: Update state for nodes that both had and have a role in a given multicast segment.

3) After T2: Removal of state for nodes that transition from having a role to not having a role for a given multicast segment.

T1 and T2 are network wide configurable values.

6. Related work

6.1. IGP Extensions

The required IGP changes are documented in [MCAST-ISIS] and [MCAST-OPSF].

6.2. BGP Extensions

This memo will require the specification of a new PMSI Tunnel Attribute (SPRING P2MP tunnel, tentatively 0x09) to order to integrate into the multicast framework documented in RFC 6514

7. Observations

This technique is not confined to segment routing, and with the provision of a global label space (to be employed as per a multicast SID), an MPLS-LDP network would also provide the requisite mesh of unicast tunnels and be capable of implementing this approach to multicast.

This memo focuses on an implementation based upon nodes that are IGP speakers and converge independently so is written in a form that

assumes a node, computing node and IGP speaker are one in the same. It should be observed that the relative frugality of data plane state would suggest that separation of computation from nodes in the data plane combined with management or "software defined networking" based population of the multicast FIB entries may also be useful modes of network operation.

8. Acknowledgements

Thanks to Uma Chunduri for his detailed review and suggestions.

9. Security Considerations

For a future version of this document.

10. IANA Considerations

This document requires the allocation of a PMSI tunnel type to identify a SPRING P2MP tunnel type from the P-Multicast Service Interface Tunnel (PMSI Tunnel) Tunnel Types registry.

11. References

11.1. Normative References

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

11.2. Informative References

[MCAST-ISIS] Allan et.al., "IS-IS extensions for Computed Multicast applied to MPLS based Segment Routing", IETF work in progress, draft-allan-isis-spring-multicast-00, July 2016

[MCAST-OSPF] Allan et.al., "OSPF extensions for Computed Multicast applied to MPLS based Segment Routing", IETF work in progress, draft-allan-ospf-spring-multicast-00, July 2016

[RFC6514] Aggarwal et.al., "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", IETF RFC 6514, February 2012

[RFC7385] Andersson & Swallow "IANA Registry for P-Multicast Service Interface (PMSI) Tunnel Type Code Points", IETF RFC 7385, October 2014

12. Authors' Addresses

Dave Allan (editor)
Ericsson
300 Holger Way
San Jose, CA 95134
USA
Email: david.i.allan@ericsson.com

Jeff Tantsura
Email: jefftant.ietf@gmail.com

SPRING Working Group
Internet-Draft
Intended Status: Informational

Expires: September 21, 2016

Madhukar Anand
Sanjoy Bardhan
Ramesh Subrahmaniam
Infinera Corporation
March 20, 2016

Packet-Optical Integration in Segment Routing
draft-anand-spring-poi-sr-00

Abstract

This document illustrates a way to integrate a new class of nodes and links in segment routing to represent networks in an opaque way for further extensibility of the link-state protocols that help with segment routing. An instance of the opaque definition would be optical networks that are typically transport centric. In the IP centric network, this will help in defining a common control protocol for packet optical integration that will include optical paths as opaque 'segments' or sub-paths as an augmentation to the defined extensions of segment routing. This opaque option defines a general mechanism to allow for future extensibility of segment 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 to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Table of Contents

1 Introduction 3
2. Reference Taxonomy 3
3. Use case - Packet Optical Integration 3
4. Mechanism overview 5
5. IS-IS extensions for supporting the opaque adjacency segment . 6
6. OSPF extensions for supporting the opaque adjacency segment . 8
7. OSPFv3 extensions for supporting the opaque adjacency segment 10
8. BGP-LS extensions for supporting the opaque adjacency segment 11
8.1 Link Attribute TLVs 11
8.2 Opaque Adjacency SID TLV 12
9. PCEP-LS extensions for supporting the opaque adjacency segment 12
10. Summary 13
11. Security Considerations 14
12 IANA Considerations 14
13 References 14
13.1 Normative References 14
13.2 Informative References 15
Authors' Addresses 15

1 Introduction

Packet and optical transport networks have evolved independently with different control plane mechanisms that have to be provisioned and maintained separately. Consequently, coordinating packet and optical networks for delivering services such as end-to-end traffic engineering or failure response has proved challenging. To address this challenge, a unified control and management paradigm that provides an incremental path to complete packet-optical integration while leveraging existing signaling and routing protocols in either domains is needed. This document introduces such a paradigm based on Segment Routing (SR) [I-D.ietf-spring-segment-routing].

This document introduces a new type of segment, Opaque Adjacency Segment. Opaque Adjacency Segment can be used to model abstracted paths through the optical transport domain and integrate it with the packet network for delivering end-to-end services. In addition, this also introduces a notion of a Packet optical gateway (POG). These are nodes in the network that map packet services to the optical domain that originate and terminate these opaque adjacency segments. Given an opaque adjacency, a POG will expand it to a path in the optical transport network.

2. Reference Taxonomy

POG - Packet optical gateway Device

SR Edge Router - The Edge Router which is the first SR capable device

CE - Customer Edge Device that is outside of the SR domain

PCE - Path Computation Engine

Controller - A network controller

3. Use case - Packet Optical Integration

Many operators build and operate their networks that are both multi-layer and multi-domain. Services are built around these layers and domains to provide end-to-end services. Due to the nature of the different domains, such as packet and optical, the management and service creation has always been problematic and time consuming. With segment routing, enabling a head-end node to select a path and embed the information in the packet is a powerful construct that would be

used in the Packet Optical Gateways (POG). The path is usually constructed for each domain that may be manually derived or through a stateful PCE which is run specifically in that domain.

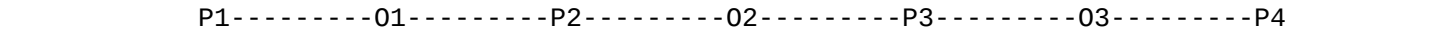


Figure 1: Representation of a packet-optical path

In Figure 1 above, the nodes represent a packet optical network. P1, P2, P3 and P4 are packet optical devices that are connected via optical paths O1, O2 and O3. Nodes P1 and P4 are edge devices that have customer facing devices (denoted as Border POGs) and P2 and P3 are core nodes (denoted as Transit POGs) in the network. A packet service is established by specifying a path between P1 and P4. Note that in defining this path, we will need to specify both the nodes and the links that make up this service. POGs advertise themselves along with their adjacencies and the domains they belong to. To leverage segment routing to define the above service, the ingress node P1 would append all outgoing packets in a SR header consisting of the SIDs that constitute the path. In the packet domain this would mean P1 would send its packets towards P4 using the segment list {P2, P4}. The operator would need to use a different mechanism in the optical domain to set up the optical paths denoted by O1, O2 and O3. Each POG would announce the active optical path as an opaque adjacency - for example, in the case of P1, the optical path O1 would represent an optical path that includes the optical nodes Om and On as shown on Figure 2. This path is not known to the packet SR domain and is only relevant to the optical domain D between P1 and P2. A PCE that is run in Domain D would be responsible for calculating path O1.

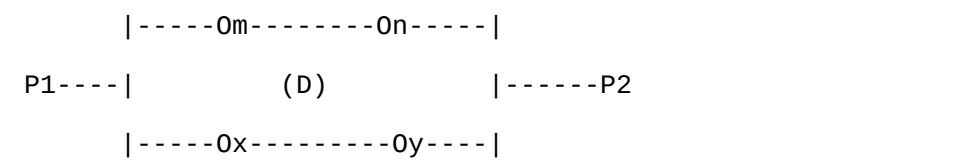


Figure 2: POG with multiple optical paths through an optical domain

Similarly, the transit POGs P2 and P3 in Figure 1 would announce opaque adjacencies O2 and O3. The border POG would include the optical paths O1, O2 and O3 to the segment list for P1 to P4. The expanded segment list would read as {O1, P2, O2, P3, O3, P4}.

There are potentially two locations for Borders POGs - one that has last-mile access nodes and the other being Data Center Interconnect nodes. The POGs that are in the core of the network which connect with long haul optical networks are usually Transit POGs.

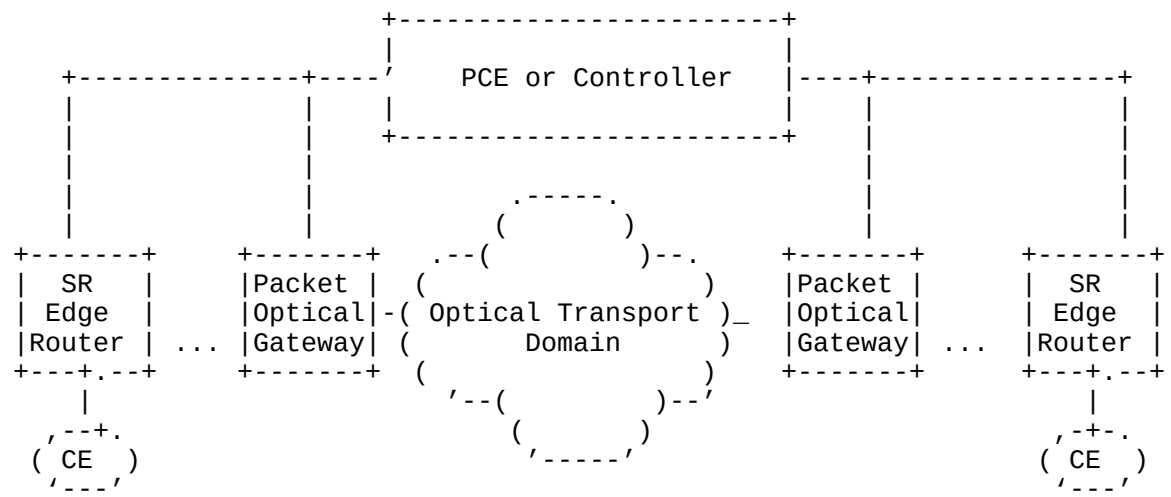


Figure 3. Reference Topology for Opaque Adjacency Segment

4. Mechanism overview

The current proposal assumes that the SR domains run standard IGP protocols to discover the topology and distribute labels without any modification. There are also no modifications to the control plane mechanisms in the Optical transport domains. The mechanism for supporting the opaque adjacency segment is as follows.

1. Firstly, the Packet Optical Gateway (POG) devices announce themselves in the SR domain. This is indicated by advertising a new SR node capability flag. The exact extensions to support this capability are described in the subsequent sections of this document.

2. Then, the POG devices announce paths to other POGs through the optical transport domain as an opaque adjacency segment (opaque adjacency SID) in the SR domain. The paths are announced with an appropriate transit domain type, optical transport domain ID, and a label to be used to bind to the opaque adjacency segment. The

appropriate IGP segment routing extensions to carry this information is described in the subsequent sections of this document.

3. The opaque adjacency segment can also optionally be announced with a set of attributes that characterizes the path in the optical transport domain between the two POG devices. For instance, those attributes could define the OTN mapping used (e.g., ODU4, ODU3,ODU3e1....ODU1), timeslots (1-8 or 4,6,7 or 1-2,5), or optical path protection schemes.

4. The POG device is also responsible for programming its forwarding table to map every opaque adjacency label entry into an appropriate forwarding action relevant in the optical domain, such as mapping it to a label-switched path.

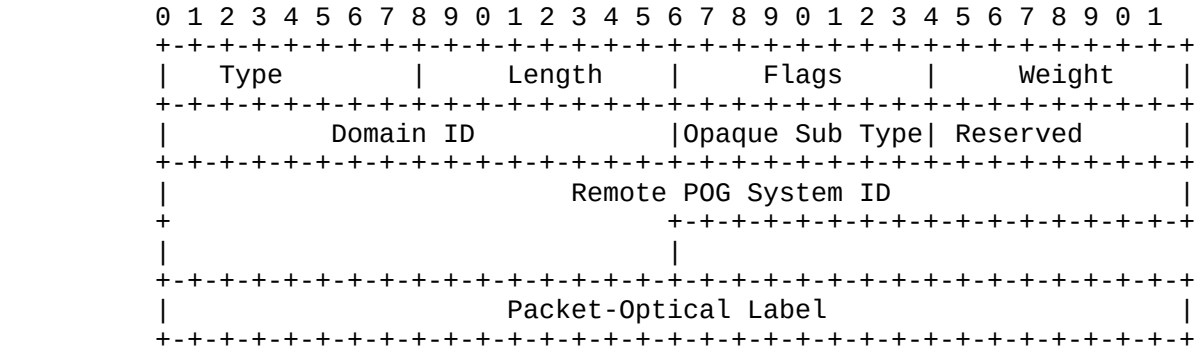
5. The opaque adjacency segment is communicated to the PCE or Controller using extensions to BGP-LS or PCEP-LS as described in subsequent sections of this document.

6. Finally, the PCE or Controller then uses the opaque adjacency segment label to influence the path leaving the SR domain into the optical domain, thereby defining the end-to-end path for a given service.

5. IS-IS extensions for supporting the opaque adjacency segment

A new IS-IS sub-TLV is defined: the Opaque Adjacency Segment Identifier sub-TLV (Opaque-Adj-SID sub-TLV). The Opaque-Adj-SID sub-TLV is an optional sub-TLV carrying the opaque adjacency SID with flags and fields that may be used, in future extensions of Segment Routing, for carrying other types of Opaque Adjacency SIDs.

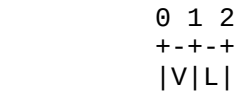
Multiple Opaque-Adj-SID sub-TLVs MAY be associated with a pair of POG devices to represent multiple paths within the optical domain with perhaps different characteristics.



Where:
Type: TBD, suggested value 33

Length: variable.

Flags: 1 octet field of following flags:
V - Value flag. If set, then the packet-optical label carries a value. By default the flag is SET.
L - Local. Local Flag. If set, then the value/index carried by the Adj-SID has local significance. By default the flag is SET.



Other bits: Reserved. These MUST be zero when sent and are ignored when received.

Weight: TBD

Domain ID: An identifier for the transport domain

Opaque Sub Type: TBD

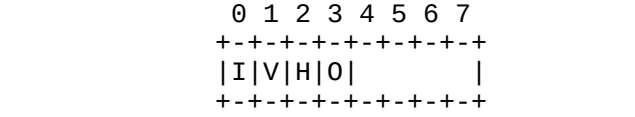
Remote POG System-ID: 6 octets of IS-IS System-ID of length "ID Length" as defined in [ISO10589].

Packet-Optical Label : according to the V and L flags, it contains either:

* A 3 octet local label where the 20 rightmost bits are used for encoding the label value. In this case the V and L flags MUST be set.

* A 4 octet index defining the offset in the label space advertised by this router. In this case V and L flags MUST be unset.

Further, to communicate the Packet-Optical Gateway capability of the device, we introduce a new flag O in the SR Node Capabilities sub-TLV:

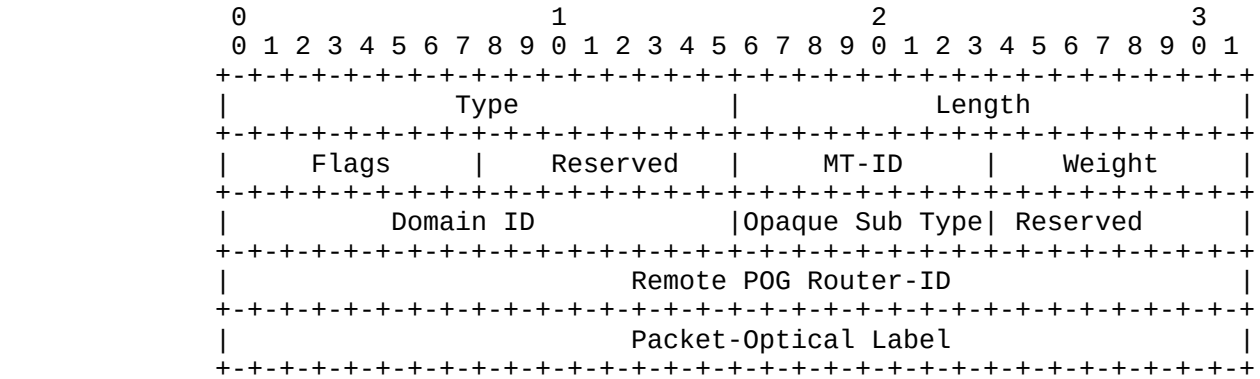


I, V, H flags are defined in [I-D.ietf-isis-segment-routing-extensions].
O-Flag: If set, then the router is capable of performing Packet Optical Gateway function.

6. OSPF extensions for supporting the opaque adjacency segment

A new OSPF sub-TLV is defined: the Opaque Adjacency Segment Identifier sub-TLV (Opaque-Adj-SID sub-TLV). The Opaque-Adj-SID sub-TLV is an optional sub-TLV of the Extended Link TLV carrying the opaque adjacency SID with flags and fields that may be used, in future extensions of Segment Routing, for carrying other types of Opaque Adjacency SIDs.

Multiple Opaque-Adj-SID sub-TLVs MAY be associated with a pair of POG devices to represent multiple paths within the optical domain with perhaps different characteristics.

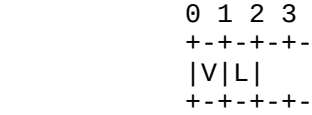


where:

Type: TBD, suggested value 3

Length: variable.

Flags: 1 octet field of following flags:
V - Value flag. If set, then the optical label carries a value. By default the flag is SET.
L - Local. Local Flag. If set, then the value/index carried by the Adj-SID has local significance. By default the flag is SET.



Other bits: Reserved. These MUST be zero when sent and are ignored when received.

MT-ID: Multi-Topology ID (as defined in [RFC4915]).

Weight: TBD

Domain ID: An identifier for the transport domain

Opaque Sub Type: TBD

Remote POG Router-ID: 4 octets of OSPF Router-ID

Packet-Optical Label : according to the V and L flags, it contains either:

* A 3 octet local label where the 20 rightmost bits are used for encoding the label value. In this case the V and L flags MUST be set.

* A 4 octet index defining the offset in the label space advertised by this router. In this case V and L flags MUST be unset.

Further, to communicate the Packet-Optical Gateway capability of the device, we introduce an new optical informational capability bit in the Router Information capabilities TLV (as defined in [RFC4970]).

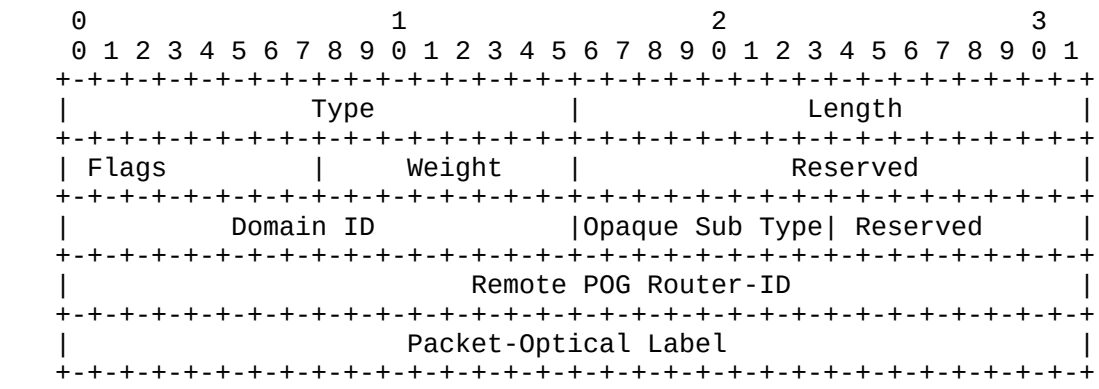
Bit-24 - Optical - If set, then the router is capable of performing Packet Optical Gateway function.

7. OSPFv3 extensions for supporting the opaque adjacency segment

The Opaque-Adj-SID Sub-TLV is an optional Sub-TLV of the Router-Link TLV as defined in [I-D.ietf-ospf-ospfv3-lsa-extend]. It MAY appear multiple times in Router-Link TLV.

Multiple Opaque-Adj-SID sub-TLVs MAY be associated with a pair of POG devices to represent multiple paths within the optical domain with perhaps different characteristics.

The Opaque-Adj-SID Sub-TLV has the following format:

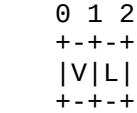


where:

Type: TBD, suggested value 6

Length: variable.

Flags: 1 octet field of following flags:
- V - Value flag. If set, then the optical label carries a value. By default the flag is SET.
- L - Local. Local Flag. If set, then the value/index carried by the Adj-SID has local significance. By default the flag is SET.



Other bits: Reserved. These MUST be zero when sent and are ignored when received.

Weight: TBD

Domain ID: An identifier for the transport domain

Opaque Sub Type: TBD

Remote POG Router-ID: 4 octets of OSPFv3 Router-ID

Packet-Optical Label : according to the V and L flags, it contains either:

* A 3 octet local label where the 20 rightmost bits are used for encoding the label value. In this case the V and L flags MUST be set.

* A 4 octet index defining the offset in the label space advertised by this router. In this case V and L flags MUST be unset.

Further, to communicate the Packet-Optical Gateway capability of the device, we introduce an new optical informational capability bit in the Router Information capabilities TLV (as defined in [RFC4970]).

Bit-24 - Optical - If set, then the router is capable of performing Packet Optical Gateway function.

8. BGP-LS extensions for supporting the opaque adjacency segment

8.1 Link Attribute TLVs

The following new Link Attribute TLVs are defined:

TLV Code Point	Description	Length	Section
1101	Opaque Adjacency Segment Identifier (Opq-Adj-SID)TLV	variable	

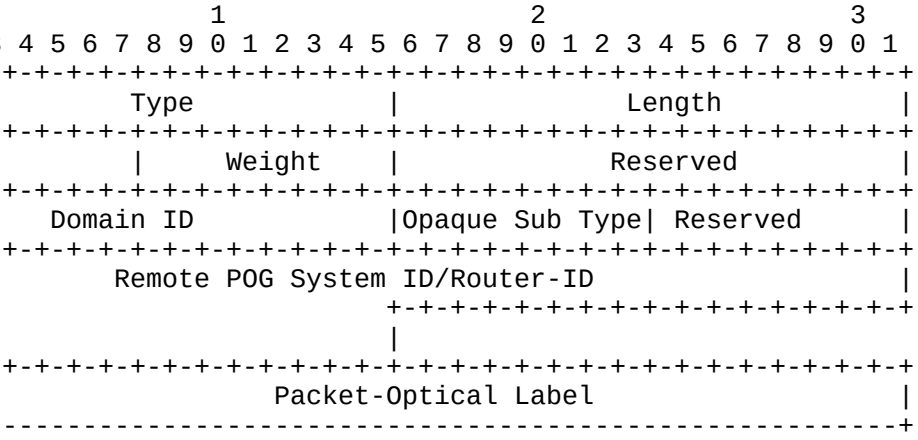
Table 1: BGP-LS Link Attribute TLVs

These TLVs can ONLY be added to the Link Attribute associated with the link whose local node originates the corresponding SR TLV.

The Opaque adjacency segment TLV allows a node to advertise an opaque adjacency within a single IGP domain.

8.2 Opaque Adjacency SID TLV

The Opaque Adjacency SID (Opq-Adj-SID) TLV has the following format:



Where:

Type: TBD, suggested value 1101.

Length: Variable.

Flags: 1 octet field of following flags as defined in the previous sections for IS-IS and OSPF.

Weight: TBD.

Domain ID: An identifier for the optical transport domain

Opaque Sub Type : TBD

Remote POG Router-ID/System-ID: 4 octets of OSPF Router-ID or 6 Octets of IS-IS System ID.

Packet-Optical Label: 4 octet field carrying the label as defined in the previous sections for IS-IS and OSPF.

9. PCEP-LS extensions for supporting the opaque adjacency segment

Changes similar to BGP-LS are needed for supporting the opaque adjacency segment in PCEP-LS. Details TBD.

10. Summary

The motivation for introducing an opaque adjacency segment that is separate from an IGP adjacency segment is to distinguish between a real IGP adjacency (which is typically a symmetric relationship between the devices that share a route flooding domain), and a relationship between devices in potentially two different domains such as packet and optical domains with no real IGP adjacency. Further, the opaque adjacency segment can carry optional information that is of significance only in the optical domain, and hence, opaque, to the IGPs. This is specifically useful if the optical domain is bridging the same IGP domain, then, the POG can attach both the adjacency SID and the opaque adjacency SID to influence the end-to-end path in the packet and optical domains respectively.

11. Security Considerations

This document does not introduce any new security considerations.

12 IANA Considerations

TBD.

13 References

13.1 Normative References

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and r. rjs@rob.sh, "Segment Routing Architecture", draft-ietf-spring-segment-routing-04 (work in progress), July 2015.

[I-D.ietf-isis-segment-routing-extensions]
Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-05 (work in progress), June 2015.

[I-D.ietf-ospf-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-ietf-ospf-segment-routing-extensions-05 (work in progress), June 2015.

[RFC4915] L. Nguyen, P. Psenak, S. Mirtorabi, P. Pillay-Esnault, and A. Roy, "Multi-Topology (MT) Routing in OSPF.", RFC4915, <http://tools.ietf.org/html/rfc4915>.

[I-D.ietf-ospf-ospfv3-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPFv3 Extensions for Segment Routing", draft-ietf-ospf-ospfv3-segment-routing-extensions-03 (work in progress), June 2015.

[I-D.ietf-idr-ls-distribution]
Gredler, H., Medved, J., Previdi, S., Farrel, A., and S.

Ray, "North-Bound Distribution of Link-State and TE Information using BGP", draft-ietf-idr-ls-distribution-13 (work in progress), October 2015.

[RFC4970] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 4970, DOI 10.17487/RFC4970, July 2007, <http://www.rfc-editor.org/info/rfc4970>.

13.2 Informative References

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

Authors' Addresses

Madhukar Anand
Infinera Corporation
169 W Java Dr, Sunnyvale, CA 94089

Email: manand@infinera.com

Sanjoy Bardhan
Infinera Corporation
169 W Java Dr, Sunnyvale, CA 94089

Email: sbardhan@infinera.com

Ramesh Subrahmaniam
Infinera Corporation
169 W Java Dr, Sunnyvale, CA 94089

Email: RSubrahmaniam@@infinera.com

SPRING Working Group
Internet-Draft
Intended Status: Standard Track

Madhukar Anand
Ciena Corporation

Sanjoy Bardhan
Infinera Corporation

Ramesh Subrahmaniam
Individual

Jeff Tantsura
Apstra

Utpal Mukhopadhyaya
Equinix Inc

Clarence Filsfils
Cisco Systems, Inc.

Expires: August 1, 2019 January 28, 2019

Packet-Optical Integration in Segment Routing
draft-anand-spring-poi-sr-07

Abstract

This document illustrates a way to integrate a new class of nodes and links in segment routing to represent transport/optical networks in an opaque way into the segment routing domain. An instance of this class would be optical networks that are typically transport centric by having very few devices with the capability to process packets. In the IP centric network, this will help in defining a common control protocol for packet optical integration that will include optical paths as 'transport segments' or sub-paths as an augmentation to packet paths. The transport segment option also defines a general mechanism to allow for future extensibility of segment routing into non-packet domains.

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

Copyright and License Notice

Table of Contents

1 Introduction 4
2. Reference Taxonomy 4
3. Use case - Packet Optical Integration 5
4. Mechanism overview 8
5. Transport Segments as SR Policy 9
6. PCEP extensions for supporting the transport segment 10
7. BGP-LS extensions for supporting the transport segment 11
7.1 Node Attributes TLV 11
7.2 SR-Optical-Node-Capability Sub-TLV 11
7.3 Prefix Attribute TLVs 12
7.3.1 Transport Segment SID Sub-TLV 12

8. Note about Transport Segments and Scalability 13
9. Summary 14
10. Security Considerations 14
11 IANA Considerations 14
11.1 PCEP 14
11.2 BGP-LS 15
12 Acknowledgements 15
13 References 15
13.1 Normative References 15
13.2 Informative References 16
Authors' Addresses 16

1 Introduction

Packet and optical transport networks have evolved independently with different control plane mechanisms that have to be provisioned and maintained separately. Consequently, coordinating packet and optical networks for delivering services such as end-to-end traffic engineering or failure response has proved challenging. To address this challenge, a unified control and management paradigm that provides an incremental path to complete packet-optical integration while leveraging existing signaling and routing protocols in either domains is needed. This document introduces such a paradigm based on Segment Routing (SR) [RFC8402].

This document introduces a new type of segment, Transport segment, as a special case of SR traffic engineering (SR-TE) policy (Type 1, Sec 5. [I-D.draft-ietf-spring-segment-routing-policy]). Specifically, the structure of SR-TE policy and constraints associated in the transport/optical network are different from those outlined for the packet networks. Transport segment can be used to model abstracted paths through the transport/optical domain and integrate it with the packet network for delivering end-to-end services. In addition, this also introduces a notion of a Packet optical gateway (POG). These are nodes in the network that map packet services to the optical domain that originate and terminate these transport segments. Given a transport segment, a POG will expand it to a path in the optical transport network. A POG can be viewed as SR traffic engineering policy headend.

The concept of POG introduced here allows for multiple instantiations of the concept. In one case, the packet device is distinct from the transport/optical device, and the POG is a logical entity that spans these two devices. In this case, the POG functionality is achieved with the help of external coordination between the packet and optical devices. In another case, the packet and optical components are integrated into one physical device, and the co-ordination required for functioning of the POG is performed by this integrated device. It must be noted that in either case, it is the packet/optical data plane that is either disaggregated or integrated. Control of the devices can be logically centralized or distributed in either scenario. The focus of this document is to define the logical functions of a POG without going into the exact instantiations of the concept.

2. Reference Taxonomy

POG - Packet optical gateway Device

SR Edge Router - The Edge Router which is the ingress device

CE - Customer Edge Device that is outside of the SR domain

PCE - Path Computation Engine

Controller - A network controller

3. Use case - Packet Optical Integration

Many operators build and operate their networks that are both multi-layer and multi-domain. Services are built around these layers and domains to provide end-to-end services. Due to the nature of the different domains, such as packet and optical, the management and service creation has always been problematic and time consuming. With segment routing, enabling a head-end node to select a path and embed the information in the packet is a powerful construct that would be used in the Packet Optical Gateways (POG). The path is usually constructed for each domain that may be manually derived or through a stateful PCE which is run specifically in that domain.

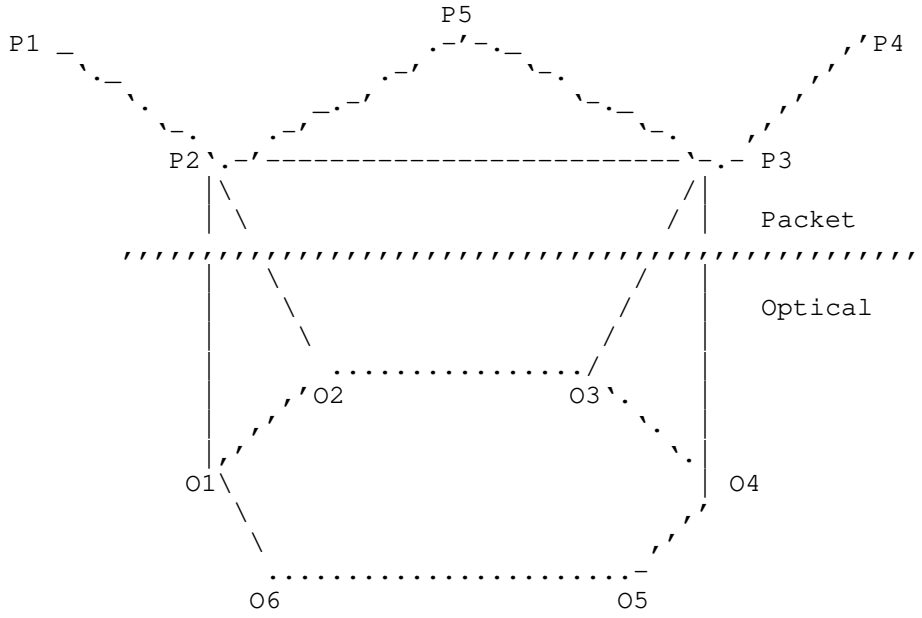


Figure 1: Representation of a packet-optical path

In Figure 1 above, the nodes represent a packet optical network. P1,...,P5 are packet devices. Nodes P2 and P3 are connected via optical network (e.g., DWDM) comprising of nodes O1,...,O6. Nodes P2 and P3 are POGs that communicate with other packet devices and also with the devices in the transport/optical domain. POGs P2 and P3 are connected to optical nodes O2/O3 and O3/O4 respectively via multiple links that are visible to the packet network. In defining a path between nodes P2 and P3, we will need to specify the nodes and the links in both the packet and transport/optical domains.

To leverage segment routing to define a service between P1 and P4, the ingress node P1 would append all outgoing packets in a SR header consisting of the SIDs that constitute the path. In the packet domain this would mean P1 would send its packets towards P4 using a segment list {P2, P3, P4} or {P2, P5, P3, P4} as the case may be. The operator would need to use a different mechanism in the optical domain to set up the paths between the two POGs P2 and P3. For instance, if the packet is forwarded on the link from P2 towards O1 with the expectation that it would come out on the link O4-P3, it could be routed in the optical network using either path {O1, O2, O3, O4} or {O1, O6, O5, O4}. Currently, this decision is made in the optical domain, and there are no mechanisms in the packet network to influence that. The transport segment mechanism proposed in this draft has been designed with an explicit goal of providing better control of optical path selection to the packet network and applications running on them.

Under the proposed scheme, each POG would announce active optical paths to the other POG as a transport segment - for example, the optical path from P2 to P3 comprising {O1, O2, O3, O4} could be represented as a transport segment label Om and the optical path from P2 to P3 comprising devices {O1, O5, O6, O4} could be represented as a transport segment label On. Both Om and On will be advertised by POG P2 as two optical paths between P2 and P3 with specific properties. The specifics of the optical paths, including specific intermediate devices, need not be exposed to the packet SR domain and are only relevant to the optical domain between P2 and P3. A PCE that is run in the optical domain would be responsible for calculating paths corresponding to label Om and On. The expanded segment list would read as {P2, Om, P3, P4} or {P2, On, P3, P4}. Multiple optical paths between P2 and P3 corresponding to different properties can be exposed as transport segments in the packet domain. For example, some optical paths can be low operational cost paths, some could be low-latency, and some others can be high-bandwidth paths. Transport segments for all these candidate viable alternative paths may be generated statically or dynamically.They may be pre-computed or may be generated on the fly when a customer at node P1 requests a service towards node P4. A discussion on transport segments and scalability can be found in Section 8.

Use-case examples of transport segments.

1. Consider the scenario where there are multiple fibers between two packet end points. The network operator may choose to route packet traffic on the first fiber, and reserve the second fiber only for maintenance or low priority traffic.

2. As a second use-case, consider the case where the packet end points are connected by transport/optical network provided by two different service providers. The packet operator wants to preferentially route traffic over one of the providers and use the second provider as a backup.

3. Finally, let the packet end points be connected by optical paths that may span multiple optical domains i.e. different administrative control. For instance, one transport/optical path may lie completely in one country while the other transport/optical path transits another country. Weather, tariffs, security considerations and other factors may determine how the packet operator wants to route different types of traffic on this network.

All of the above use-cases can be supported by first mapping distinct transport/optical paths to different transport segments and then, depending on the need, affixing appropriate transport segment identifier to the specific packet to route it appropriately through the transport domain.

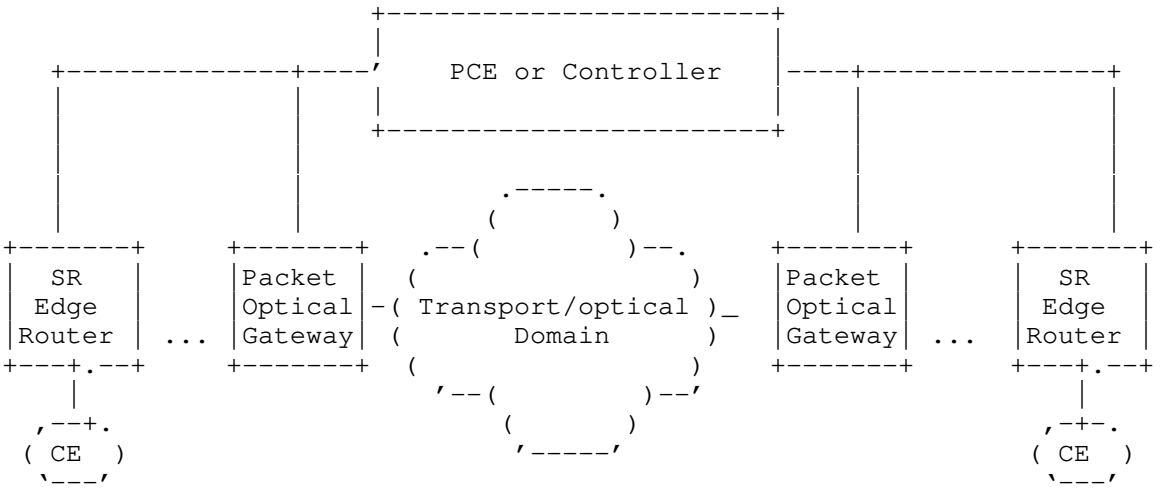


Figure 3. Reference Topology for Transport Segment Mechanism

4. Mechanism overview

The current proposal assumes that the SR domains run standard protocols without any modification to discover the topology and distribute labels. There are also no modifications necessary in the control plane mechanisms in the transport/optical domains. The only requirement of a transport segment is that the optical path be setup before they are announced to the packet network. For example, the optical paths may be setup using a domain-specific controller or a PCE based on requirements from the packet domain (such as bandwidth, QoS, latency and cost) taking into consideration the constraints in the optical network.

The mechanism for supporting the transport segment is as follows.

1. Firstly, the Packet Optical Gateway (POG) devices are announced in the packet domain. This is indicated by advertising a new SR node capability flag. The exact extensions to support this capability are described in the subsequent sections of this document.

2. Then, the POG devices announce candidate transport/optical paths between that POG (Source POG) and other POGs (Destination POG) via appropriate mechanisms in the packet domain. The paths are announced with an appropriate transport/optical domain ID and a Binding SID representing the transport segment from a source POG to a destination POG. The appropriate protocol-specific extensions to carry path characteristics and Binding SID corresponding to a optical path are described in the subsequent sections of this document.

3.The transport SR policy can also optionally be announced with a set of attributes that characterizes the path in the transport/optical domain between the two POG devices. For instance, those could define the path attributes such as path identifier, latency, bandwidth, quality, directionality, or optical path protection schemes. These attributes can be used to determine the "color" of the SR-TE policy in the tuple <Source POG, Destination POG, color> used to prioritize different candidate paths between the POGs.

4. The POG device is also responsible for programming its forwarding table to map every transport segment Binding SID entry

into an appropriate forwarding action relevant in the optical domain, such as mapping it to a optical label-switched path.

5. The transport SR policy is communicated to the PCE or Controller using extensions to BGP-LS or PCEP as described in subsequent sections of this document.

6. Finally, the PCE or Controller in the packet domain then uses the transport segment binding SID in the overall SR policy to influence the path traversed by the packet in the optical domain, thereby defining the end-to-end path for a given service.

In the next few sections, we outline a few representative protocol specific extensions to carry the transport segment.

5. Transport Segments as SR Policy

The Segment Routing Traffic Engineering (SRTE) [ietf-spring-segment-routing-policy] process installs the transport segment SR policy in the forwarding plane of the POG. The Transport SR policy is identified by using a transport segment Binding SID. Corresponding to each transport segment Binding SID, the SRTE process MAY learn about multiple candidate paths. The SRTE-DB includes information about the candidate paths including optical domain, topology and path characteristics. All of the information can be learned from different sources including but not limited to: Netconf/Restconf, PCEP and BGP-LS.

The information model for Transport SR policy is as follows:

```
    Transport SR Policy FO1
           Candidate-paths
             path preference 200 (selected)
                 BSID1
             path preference 100
                 BSID2
             path preference 100
                 BSID3
             path preference 50
                 BSID4
```

A transport SR policy is identified through the tuple <Source POG, Destination POG, color>. Each TSR policy is associated with one or more candidate paths, each of them associated with a (locally) unique Binding SID and a path preference. For each transport SR policy, the candidate path with the highest path preference (at most one) is selected and used for forwarding traffic that is being steered onto that policy. When candidate paths change (or a new candidate path is set up), the path

selection process can be re-executed. The validity of each path is to be verified by the POG before announcement in the packet domain. If there are no valid paths, then the transport SR policy is deemed invalid.

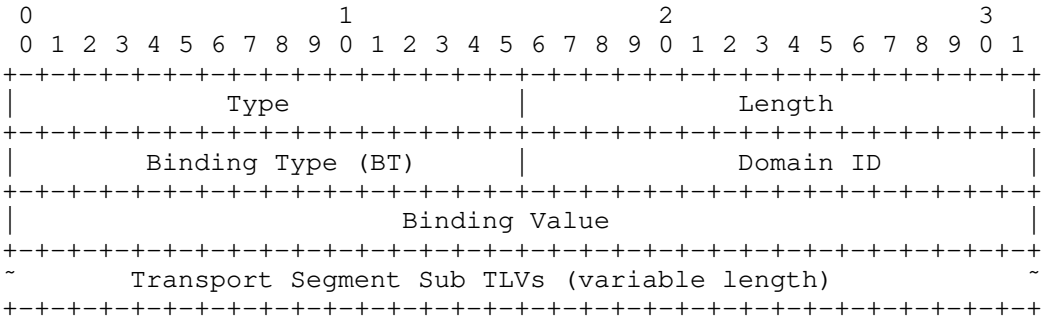
The allocation of BSID to a path can include dynamic, explicit or generic allocation strategies as discussed in [ietf-spring-segment-routing-policy]. We discuss PCEP and BGP-LS specific extensions in the subsequent section.

6. PCEP extensions for supporting the transport segment

To communicate the Packet-Optical Gateway capability of the device, we introduce a new PCEP capabilities TLV is defined as follows(extensions to [I-D.ietf-pce-segment-routing]):

Value	Meaning	Reference
TBD1	TRANSPORT-SR-PCE-CAPABILITY	This document

A new type of TLV to accommodate a transport segment is defined by extending Binding SIDs [I-D.sivabalan-pce-binding-label-sid]



where:

Type: TBD

Length: variable.

Binding Type: 0 or 1 as defined in
[I-D.sivabalan-pce-binding-label-sid]

Domain ID: An identifier for the transport domain

Binding Value: is the transport segment label

Transport Segment Sub TLVs: TBD

IANA will be requested to allocate a new TLV type for
TRANSPORT-SEGMENT-BINDING-TLV as specified in this document:

TBD Transport Segment Label (This document)

7. BGP-LS extensions for supporting the transport segment

7.1 Node Attributes TLV

To communicate the Packet-Optical Gateway capability of the device, we introduce an new optical informational capability the following new Node Attribute TLV is defined:

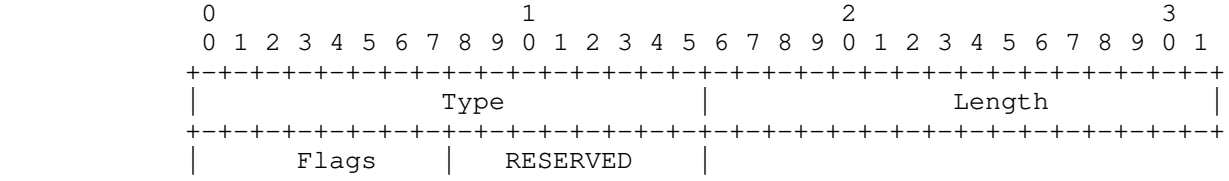
TLV Code Point	Description	Length	Section
TBD	SR-Optical-Node-Capability TLV	variable	

Table 1: Node Attribute TLVs

These TLVs can ONLY be added to the Node Attribute associated with the node NLRI that originates the corresponding SR TLV.

7.2 SR-Optical-Node-Capability Sub-TLV

The SR Capabilities sub-TLV has following format:



where:

Type : TBD, Suggested Value 1157

Length: variable.

Flags: The Flags field currently has only one bit defined. If the bit is set it has the capability of an Packet Optical Gateway.

7.3 Prefix Attribute TLVs

The following Prefix Attribute Binding SID Sub-TLVs have been added:

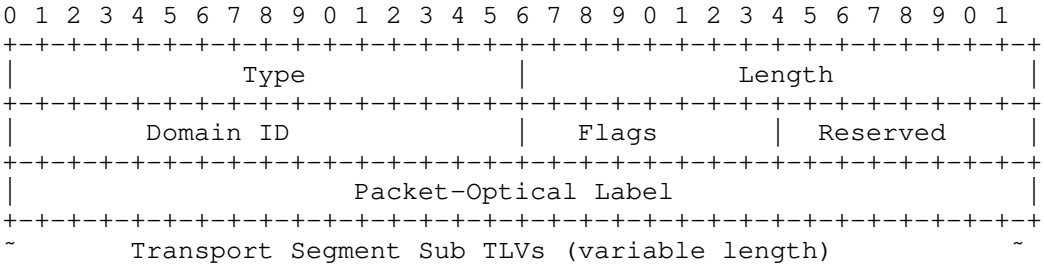
TLV Code Point	Description	Length	Section
TBD	TRANSPORT-SEGMENT-SID	12	

Table 4: Prefix Attribute - Binding SID Sub-TLVs

The Transport segment TLV allows a node to advertise an transport segment within a single IGP domain. The transport segment SID TLV TRANSPORT-SEGMENT-TLV has the following format:

7.3.1 Transport Segment SID Sub-TLV

Further, a new sub-TLV (similar to the IPV4 ERO SubTLV) of Binding SID Sub-TLV (TRANSPORT-SEGMENT-BINDING-SUBTLV) to carry the transport segment label is defined as follows.



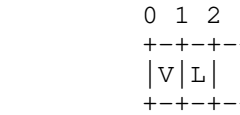
where:

Type : TBD

Length: variable.

Domain ID: An identifier for the transport domain

Flags: 1 octet field of following flags:
V - Value flag. If set, then the optical label carries a value. By default the flag is SET.
L - Local. Local Flag. If set, then the value/index carried by the Adj-SID has local significance. By default the flag is SET.



Packet-Optical Label : according to the V and L flags, it contains either:

* A 3 octet local label where the 20 rightmost bits are used for encoding the label value. In this case the V and L flags MUST be set.

* A 4 octet index defining the offset in the label space advertised by this router. In this case V and L flags MUST be unset.

Transport Segment Sub TLVs: TBD

Multiple TRANSPORT-SEGMENT-TLV MAY be associated with a pair of POG devices to represent multiple paths within the optical domain

8. Note about Transport Segments and Scalability

In most operational scenarios, there would be multiple, distinct paths between the POGs. There is no requirement that every distinct path in the optical domain be advertised as a separate transport segment. Transport segments are designed to be consumed in the packet domain, and the correspondence between transport segments and exact paths in the optical domain are determined by their utility to the packet world. Therefore, the number of transport segments is to be determined by the individual packet-optical use-case. The number of actual paths in the

optical domain between the POG is expected to be large (counting the number of active and passive devices in the optical network), it is likely that multiple actual paths are to be advertised as one transport segment. Of course, in the degenerate case, it is possible that there is a one-to-one correspondence between an optical path and a transport segment. Given this view of network operation, the POG is not expected to handle a large number of transport segments (and identifiers). This framework does leave open the possibility of handling a large number of transport segments in future. For instance, a hierarchical partitioning of the optical domain along with stacking of multiple transport segment identifiers could be explored towards reducing the overall number of transport segment identifiers.

9. Summary

The motivation for introducing a new type of segment - transport segment - is to integrate transport/optical networks with the segment routing domain and expose characteristics of the transport/optical domain into the packet domain. An end-to-end path across packet and transport/optical domains can then be specified by attaching appropriate SIDs to the packet. An instance of transport segments has been defined here for optical networks, where paths between packet-optical gateway devices have been abstracted using binding SIDs. Extensions to various protocols to announce the transport segment have been proposed in this document.

10. Security Considerations

This document does not introduce any new security considerations.

11 IANA Considerations

This documents request allocation for the following TLVs and subTLVs.

11.1 PCEP

Packet-Optical Gateway capability of the device

Value	Meaning	Reference
TBD1	TRANSPORT-SR-PCE-CAPABILITY	This document

A new type of TLV to accommodate a transport segment is defined by extending Binding SIDs [I-D.sivabalan-pce-binding-label-sid]

Value	Description	Reference

TBD2	TRANSPORT-SR-PCEP-TLV	This document

This document requests that a registry is created to manage the value of the Binding Type field in the TRANSPORT-SR-PCEP TLV.

Value	Description	Reference
TBD3	Transport Segment Label	This document

11.2 BGP-LS
Node Attributes TLV:

Value	Description	Reference
TBD4	TRANSPORT-SR-BGPLS-CAPABILITY	This document

Prefix Attribute Binding SID SubTLV:

Value	Description	Reference
TBD5	TRANSPORT-SR-BGPLS-TLV	This document

12 Acknowledgements
We would like to thank Peter Psenak, Bruno Decraene Ketan Talaulikar and Radhakrishna Valiveti for their comments and review of this document.

13 References

13.1 Normative References

[RFC8402]
Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, July 2018.

[I-D.sivabalan-pce-binding-label-sid]
Sivabalan, S., Tantsura, J., Filsfils, C., Previdi, S., Hardwick, J., and Dhody, D., "Carrying Binding Label/ Segment-ID in PCE-based Networks.", draft-sivabalan-pce-binding-label-sid-04 (work in progress), Mar 2018.

[I-D.ietf-pce-segment-routing]
Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W.,

and J. Hardwick, "PCEP Extensions for Segment Routing", draft-ietf-pce-segment-routing-12 (work in progress), June 2018.

[I-D.draft-ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., Voyer, D., Bogdanov, A., and Mattes, P., "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-01.txt (work in progress), June 2018.

13.2 Informative References

[RFC5513] Farrel, A., "IANA Considerations for Three Letter Acronyms", RFC 5513, DOI 10.17487/RFC5513, April 1 2009, <http://www.rfc-editor.org/info/rfc5513>.

[RFC5514] Vyncke, E., "IPv6 over Social Networks", RFC 5514, DOI 10.17487/RFC5514, April 1 2009, <http://www.rfc-editor.org/info/rfc5514>.

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

Authors' Addresses

Madhukar Anand
Ciena Corporation
3939, N 1st Street, San Jose, CA, 95134
Email: madanand@ciena.com

Sanjoy Bardhan
Infinera Corporation
169 W Java Dr, Sunnyvale, CA 94089
Email: sbardhan@infinera.com

Ramesh Subrahmaniam

Email: svr_fremont@yahoo.com

Jeff Tantsura
Apstra
333 Middlefield Road Suite 200
Menlo Park, CA 94025
Email: jefftant.ietf@gmail.com

Utpal Mukhopadhyaya
Equinix Inc
1188 E. Arques, Sunnyvale, CA 94085
Email: umukhopadhyaya@equinix.com

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

Networking Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 14, 2016

L. Ginsberg
P. Psenak
S. Previdi
Cisco Systems
M. Pilka
Pantheon Technologies

April 12, 2016

Segment Routing Conflict Resolution
draft-ginsberg-spring-conflict-resolution-01.txt

Abstract

In support of Segment Routing (SR) routing protocols advertise a variety of identifiers used to define the segments which direct forwarding of packets. In cases where the information advertised by a given protocol instance is either internally inconsistent or conflicts with advertisements from another protocol instance a means of achieving consistent forwarding behavior in the network is required. This document defines the policies used to resolve these occurrences.

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

Table of Contents

1. Introduction 2
2. SR Global Block Inconsistency 3
3. Segment Identifier Conflicts 5
3.1. Conflict Types 5
3.1.1. Prefix Conflict 6
3.1.2. SID Conflict 7
3.2. Processing conflicting entries 10
3.2.1. Policy: Ignore conflicting entries 10
3.2.2. Policy: Preference Algorithm/Quarantine 10
3.2.3. Policy: Preference algorithm/ignore overlap only . . 10
3.2.4. Preference Algorithm 11
3.2.5. Use of topology in preference 11
3.2.6. Example Behavior 12
3.2.7. Evaluation of Policy Alternatives 13
3.2.8. Guaranteeing Database Consistency 13
4. Security Considerations 14
5. IANA Consideration 14
6. Acknowledgements 14
7. References 14
7.1. Normative References 14
7.2. Informational References 15
Authors' Addresses 15

1. Introduction

Segment Routing (SR) as defined in [SR-ARCH] utilizes forwarding instructions called "segments" to direct packets through the network. Depending on the forwarding plane architecture in use, routing protocols advertise various identifiers which define the permissible values which can be used as segments, which values are assigned to specific prefixes, etc. Where segments have global scope it is

necessary to have non-conflicting assignments - but given that the advertisements may originate from multiple nodes the possibility exists that advertisements may be received which are either internally inconsistent or conflicting with advertisements originated by other nodes. In such cases it is necessary to have consistent resolution of conflicts network-wide in order to avoid forwarding loops.

The problem to be addressed is protocol independent i.e., segment related advertisements may be originated by multiple nodes using different protocols and yet the conflict resolution MUST be the same on all nodes regardless of the protocol used to transport the advertisements.

The remainder of this document defines conflict resolution policies which meet these requirements. All protocols which support SR MUST adhere to the policies defined in this document.

2. SR Global Block Inconsistency

In support of an MPLS dataplane routing protocols advertise an SR Global Block (SRGB) which defines a set of label ranges reserved for use by the advertising node in support of SR. The details of how protocols advertise this information can be found in the protocol specific drafts e.g., [SR-OSPF] and [SR-IS-IS]. However the protocol independent semantics are illustrated by the following example:

The originating router advertises the following ranges:

```
Range 1: (100, 199)
Range 2: (1000, 1099)
Range 3: (500, 5990
```

The receiving routers concatenate the ranges and build the Segment Routing Global Block (SRGB) as follows:

```
SRGB = (100, 199)
       (1000, 1099)
       (500, 599)
```

The indices span multiple ranges:

```
index=0 means label 100
...
index 99 means label 199
index 100 means label 1000
index 199 means label 1099
...
index 200 means label 500
...
```

Note that the ranges are an ordered set - what labels are mapped to a given index depends on the placement of a given label range in the set of ranges advertised.

For the set of ranges to be usable the ranges MUST be disjoint. Sender behavior is defined in various SR protocol drafts such as [SR-IS-IS] which specify that senders MUST NOT advertise overlapping ranges.

Receivers of SRGB ranges MUST validate the SRGB ranges advertised by other nodes. If the advertised ranges do not conform to the restrictions defined in the respective protocol specification receivers MUST ignore all advertised SRGB ranges from that node. Operationally the node is treated as though it did not advertise any SRGB ranges. [SR-MPLS] defines the procedures for mapping global SIDs to outgoing labels.

Note that utilization of local SIDs (e.g. adjacency SIDs) advertised by a node is not affected by the state of the advertised SRGB.

3. Segment Identifier Conflicts

In support of an MPLS dataplane Segment identifiers (SIDs) are advertised and associated with a given prefix. SIDs may be advertised in the prefix reachability advertisements originated by a routing protocol. SIDs may also be advertised by a Segment Routing Mapping Server (SRMS).

Mapping entries have an explicit context which includes the topology and the SR algorithm. A generalized mapping entry can be represented using the following definitions:

Pi - Initial prefix
Pe - End prefix
L - Prefix length
Lx - Maximum prefix length (32 for IPv4, 128 for IPv6)
Si - Initial SID value
Se - End SID value
R - Range value
T - Topology
A - Algorithm

A Mapping Entry is then the tuple: (Pi/L, Si, R, T, A)
Pe = (Pi + ((R-1) << (Lx-L))
Se = Si + (R-1)

Note that the SID advertised in a prefix reachability advertisement can be more generally represented as a mapping entry with a range of 1.

Conflicts in SID advertisements may occur as a result of misconfiguration. Conflicts may occur either in the set of advertisements originated by a single node or between advertisements originated by different nodes. When conflicts occur, it is not possible for routers to know which of the conflicting advertisements is "correct". If a router chooses to use one of the conflicting entries forwarding loops and/or blackholes may result unless it can be guaranteed that all other routers in the network make the same choice. Making the same choice requires that all routers have identical sets of advertisements and that they all use the same selection algorithm.

3.1. Conflict Types

Various types of conflicts may occur.

3.1.1. Prefix Conflict

When different SIDs are assigned to the same prefix we have a "prefix conflict". Prefix conflicts are specific to mapping entries sharing the same topology and algorithm. Consider the following sets of advertisements:

(192.0.2.120/32, 200, 1, 0, 0)
(192.0.2.120/32, 30, 1, 0, 0)

The prefix 192.0.2.120/32 has been assigned two different SIDs:
200 by the first advertisement
30 by the second advertisement

(2001:DB8::1/128, 400, 1, 2, 0)
(2001:DB8::1/128, 50, 1, 2, 0)

The prefix 2001:DB8::1/128 has been assigned two different SIDs:
400 by the first advertisement
50 by the second advertisement

Prefix conflicts may also occur as a result of overlapping prefix ranges. Consider the following sets of advertisements:

(192.0.2.1/32, 200, 200, 0, 0)
(192.0.2.121/32, 30, 10, 0, 0)

Prefixes 192.0.2.121/32 - 192.0.2.130/32 are assigned two different SIDs:
320 through 329 by the first advertisement
30 through 39 by the second advertisement

(2001:DB8::1/128, 400, 200, 2, 0)
(2001:DB8::121/128, 50, 10, 2, 0)

Prefixes 2001:DB8::121/128 - 2001:DB8::130/128 are assigned two different SIDs:
420 through 429 by the first advertisement
50 through 59 by the second advertisement

The second set of examples illustrate a complication - only part of the range advertised in the first advertisement is in conflict. It is logically possible to isolate the conflicting portion and try to use the non-conflicting portion(s) at the cost of increased implementation complexity.

A variant of the overlapping prefix range is a case where we have overlapping prefix ranges but no actual SID conflict.

(192.0.2.1/32, 200, 200, 0, 0)
(192.0.2.121/32, 320, 10, 0, 0)

(2001:DB8::1/128, 400, 200, 2, 0)
(2001:DB8::121/128, 520, 10, 2, 0)

Although there is prefix overlap between the two IPv4 entries (and the two IPv6 entries) the same SID is assigned to all of the shared prefixes by the two entries.

Given two mapping entries:

(P1/L1, S1, R1, T1, A1) and (P2/L2, S2, R2, T2, A2) where P1 <= P2

a prefix conflict exists if all of the following are true:

1)The prefixes are in the same address family.
2)(L1 == L2) && (T1 == T2) && (A1 == A2)
3)(P1e >= P2) && ((S1 + (P2 - P1)) != S2)

3.1.2. SID Conflict

When the same SID has been assigned to multiple prefixes we have a "SID conflict". SID conflicts are independent of address-family, independent of prefix len, independent of topology, and independent of algorithm. A SID conflict occurs when a mapping entry which has previously been checked to have no prefix conflict assigns one or more SIDs that are assigned by another entry which also has no prefix conflicts. Consider the following examples:

(192.0.2.1/32, 200, 1, 0, 0)
(192.0.2.222/32, 200, 1, 0, 0)
SID 200 has been assigned to 192.0.2.1/32 by the
first advertisement.
SID 200 has been assigned to 192.0.2.222/32 by the
second advertisement.

(2001:DB8::1/128, 400, 1, 0, 0)
(2001:DB8::1/128, 400, 1, 0, 1)
SID 400 has been assigned to 2001:DB8::1/128 for algorithm 0
by the first advertisement.
SID 400 has been assigned to 2001:DB8::1/128 for algorithm 1
by the second advertisement.

(192.0.2.1/32, 400, 1, 0, 0)
(2001:DB8::1/128, 400, 1, 0, 0)
SID 400 has been assigned to 192.0.2.1/32 by the
first advertisement.
SID 400 has been assigned to 2001:DB8::1/128 by the
second advertisement.

SID conflicts may also occur as a result of overlapping SID ranges.
Consider the following sets of advertisements:

(192.0.2.1/32, 200, 200, 0, 0)
(198.51.100.1/32, 300, 10, 0, 0)

SIDs 300 - 309 have been assigned to two different prefixes.
The first advertisement assigns these SIDs
to 192.0.2.101/32 - 192.0.2.110/32.
The second advertisement assigns these SIDs to
198.51.100.1/32 - 198.51.100.10/32.

(2001:DB8::1/128, 400, 200, 0, 0)
(2001:DB8:1::1/128, 500, 10, 0, 0)

SIDs 500 - 509 have been assigned to two different prefixes.
The first advertisement assigns these SIDs to
2001:DB8::65/128 - 2001:DB8::6E/128.
The second advertisement assigns these SIDs to
2001:DB8:1::1 - 2001:DB8:1::A/128.

(192.0.2.1/32, 200, 200, 0, 0)
(2001:DB8::1/128, 300, 10, 0, 0)
SIDs 300 - 309 have been assigned to two different prefixes.
The first advertisement assigns these SIDs
to 192.0.2.101/32 - 192.0.2.110/32.
The second advertisement assigns these SIDs to
2001:DB8::1/128 - 2001:DB8::A/128.

The second set of examples illustrate a complication - only part of the range advertised in the first advertisement is in conflict. It is logically possible to isolate the conflicting portion and try to use the non-conflicting portion(s) at the cost of increased implementation complexity.

Given two mapping entries:

(P1/L1, S1, R1, T1, A1) and (P2/L2, S2, R2, T2, A2) where S1 <= S2

a SID conflict exists if all of the following are true:

1)S1e >= S2
2)(AF1 != AF2) || (L1 != L2) || (T1 != T2) || (A1 != A2)
  || (P1 + ((S2-S1) << (Lx-L1)) != P2

NOTE: The last calculation is valid because it is only done when the two mapping entries are in the same address family and have the same prefix length.

3.2. Processing conflicting entries

Two general approaches can be used to process conflicting entries.

1. Conflicting entries can be ignored

2. A standard preference algorithm can be used to choose which of the conflicting entries will be used

The following sections discuss these two approaches in more detail.

Note: This document does not discuss any implementation details i.e. what type of data structure is used to store the entries (trie, radix tree, etc.) nor what type of keys may be used to perform lookups in the database.

3.2.1. Policy: Ignore conflicting entries

In cases where entries are in conflict none of the conflicting entries are used i.e., the network operates as if the conflicting advertisements were not present.

Implementations are required to identify the conflicting entries and ensure that they are not used.

3.2.2. Policy: Preference Algorithm/Quarantine

For entries which are in conflict properties of the conflicting advertisements are used to determine which of the conflicting entries are used in forwarding and which are "quarantined" and not used. The entire quarantined entry is not used.

This approach requires that conflicting entries first be identified and then evaluated based on a preference rule. Based on which entry is preferred this in turn may impact what other entries are considered in conflict i.e. if A conflicts with B and B conflicts with C - it is possible that A does NOT conflict with C. Hence if as a result of the evaluation of the conflict between A and B, entry B is not used the conflict between B and C will not be detected.

3.2.3. Policy: Preference algorithm/ignore overlap only

A variation of the preference algorithm approach is to quarantine only the portions of the less preferred entry which actually conflicts. The original entry is split into multiple ranges. The ranges which are in conflict are quarantined. The ranges which are not in conflict are used in forwarding. This approach adds complexity as the relationship between the derived sub-ranges of the

original mapping entry have to be associated with the original entry - and every time some change to the advertisement database occurs the derived sub-ranges have to be recalculated.

3.2.4. Preference Algorithm

The following algorithm is used to select the preferred mapping entry when a conflict exists. Evaluation is made in the order specified.

1. Smaller range wins
2. IPv6 entry wins over IPv4 entry
3. Smaller algorithm wins
4. Smaller prefix length wins
5. Smaller starting address (considered as an unsigned integer value) wins
6. Smaller starting SID wins

Using smaller range as the highest priority tie breaker makes advertisements with a range of 1 the most preferred. This associates a high priority to SID advertisements associated with protocol prefix advertisements as these always have an implicit range of one. SR mapping server advertisements (SRMS entries) may have any configured range - but in cases where they have a range greater than 1 they will be less preferred as compared to any SIDs in prefix advertisements. This has the nice property that a single misconfiguratoion of an SRMS entry with a large range will not be preferred over a large number of SIDs advertised in prefix reachability advertisements.

3.2.5. Use of topology in preference

The preference rule defined in the previous section does not include a comparison of topologies. When evaluating prefix conflicts this is only done when comparing mapping entries associated with the same topology - so this omission is not significant. However, when evaluating a SID conflict the topology associated with two mapping entries need not be the same. The question arises as to what should be done when all of the attributes specified in the preference rule are identical but the topologies are different?

The scope of topology identifiers is NOT global. A given routing protocol has topology identifiers which are consistent within the protocol area/domain, but if multiple routing protocols are in use in a network it cannot be guaranteed that the two routing protocols will

use the same identifier for a given topology. This is, in part, due to the fact that different routing protocols have different supported ranges for topology identifiers. It is then NOT possible to guarantee a consistent identifier for a topology on all routers in a network. Therefore no preference rule can be defined which will guarantee the same result on all routers when the topology is the only attribute which differs between two mapping entries. The following preference rule is defined to handle these cases:

When a SID conflict is detected between two mapping entries and the only difference between the two entries is the topology, both entries MUST be ignored in their entirety.

3.2.6. Example Behavior

The following mapping entries exist in the database. For brevity, Topology/Algorithm is omitted and assumed to be (0,0) in all entries.

1. (192.0.2.1/32, 100, 1)

2. (192.0.2.101/32, 200, 1)

3. (192.0.2.1/32, 400, 300) !Prefix conflict with entries 1 and 2

4. (198.51.100.40/32, 200,1) !SID conflict with entry 2

The table below shows what mapping entries will be used in the forwarding plane (Active) and which ones will not be used (Excluded) under the three candidate policies:

Policy	Active Entries	Excluded Entries
Ignore		(192.0.2.1/32,100,1) (192.0.2.101/32,200,1) (192.0.2.1/32,400,300) (198.51.100.40/32,200,1)
Quarantine	(192.0.2.1/32,100,1) (192.0.2.101/32,200,1)	(192.0.2.1/32,400,300) (198.51.100.40/32,200,1)
Overlap-Only	(192.0.2.1/32,100,1) (192.0.2.101/32,200,1) *(192.0.2.2/32,401,99) *(192.0.2.102/32,501,199)	(198.51.100.40/32,200,1) *(192.0.2.1/32,400,1) *(192.0.2.101/32,500,1)

* Derived from (192.0.2.1/32,400,300)

3.2.7. Evaluation of Policy Alternatives

The previous sections have defined three alternatives for resolving conflicts - ignore, quarantine, and ignore overlap-only.

The ignore policy impacts the greatest amount of traffic as forwarding to all destinations which have a conflict is affected.

Quarantine allows forwarding for some destinations which have a conflict to be supported. The bias is for mapping entries with the smallest range (typically - but not exclusively SIDs advertised in prefix reachability advertisements) to be forwarded while the destinations included in mapping entries with a larger range but NOT covered by entries with a smaller range will not be forwarded.

Ignore overlap-only maximizes the destinations which will be forwarded as all destinations covered by some mapping entry (regardless of range) will be able to use the SID assigned by the winning range. This alternative increases implementation complexity as comapred to quarantine. Mapping entries with a range greater than 1 which are in conflict with mapping entries having a smaller range have to internally be split into 2 or more "derived mapping entries". The derived mapping entries then fall into two categories - those that are in conflict with a mapping entry of smaller range - and those which are NOT in conflict with an entry with smaller range. The former are ignored and the latter are used. Each time the underived mapping database is updated the derived entries have to be recomputed based on the updated database. Internal data structures have to maintain the relationship between the advertised mapping entry and the set of derived mapping entries. All nodes in the network have to achieve the same behavior regardless of implementation internals.

There is then a tradeoff between a goal of maximizing traffic delivery and the risks associated with increased implementation complexity.

It is the opinion of the authors that "quarantine" is the best alternative.

3.2.8. Guaranteeing Database Consistency

In order to obtain consistent active entries all nodes in a network MUST have the same mapping entry database. Mapping entries can be obtained from a variety of sources.

- SIDs can be configured locally for prefixes assigned to interfaces on the router itself. Only SIDs which are advertised to protocol peers can be considered as part of the mapping entry database.

- SIDs can be received in prefix reachability advertisements from protocol peers. These advertisements may originate from peers local to the area or be leaked from other areas and/or redistributed from other routing protocols.

- SIDs can be received from SRMS advertisements - these advertisements can originate from routers local to the area or leaked from other areas

- In cases where multiple routing protocols are in use mapping entries advertised by all routing protocols MUST be included.

4. Security Considerations

TBD

5. IANA Consideration

This document has no actions for IANA.

6. Acknowledgements

The authors would like to thank Jeff Tantsura, Wim Henderickx, and Bruno Decraene for their careful review and content suggestions.

7. References

7.1. Normative References

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

[SR-IS-IS] "IS-IS Extensions for Segment Routing, draft-ietf-isis-segment-routing-extensions-06(work in progress)", December 2015.

[SR-MPLS] "Segment Routing with MPLS dataplane, draft-ietf-spring-segment-routing-mpls-04(work in progress)", March 2016.

[SR-OSPF] "OSPF Extensions for Segment Routing, draft-ietf-ospf-segment-routing-extensions-07(work in progress)", March 2016.

[SR-OSPFv3] "OSPFv3 Extensions for Segment Routing, draft-ietf-ospf-ospfv3-segment-routing-extensions-05(work in progress)", March 2016.

7.2. Informational References

[SR-ARCH] "Segment Routing Architecture, draft-ietf-spring-segment-routing-07(work in progress)", December 2015.

Authors' Addresses

Les Ginsberg
Cisco Systems
510 McCarthy Blvd.
Milpitas, CA 95035
USA

Email: ginsberg@cisco.com

Peter Psenak
Cisco Systems
Apollo Business Center Mlynske nivy 43
Bratislava 821 09
Slovakia

Email: ppsenak@cisco.com

Stefano Previdi
Cisco Systems
Via Del Serafico 200
Rome 0144
Italy

Email: sprevidi@cisco.com

Martin Pilka
Pantheon Technologies

Email: martin.pilka@pantheon.tech

Routing area
Internet-Draft
Intended status: Informational
Expires: September 21, 2016

S. Hegde
C. Bowers
Juniper Networks, Inc.
March 20, 2016

Node Protection for SR-TE Paths
draft-hegde-spring-node-protection-for-sr-te-paths-00

Abstract

Segment routing supports the creation of explicit paths using adjacency-sids, node-sids, and binding-sids. It is important to provide fast reroute (FRR) mechanisms to respond to failures of links and nodes in the Segment-Routed Traffic-Engineered(SR-TE) path. A point of local repair (PLR) can provide FRR protection against the failure of a link in an SR-TE path by examining only the first (top) label in the SR label stack. In order to protect against the failure of a node, a PLR may need to examine the second label in the stack as well in order to determine SR-TE path beyond the failed node. This document specifies how a PLR can use the first and second label in the label stack describing an SR-TE path to provide protection against node failures.

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

Table of Contents

1. Introduction 2
2. Node Failures Along SR-TE Paths 3
2.1. Node protection for node-sid explicit paths 3
2.2. Node-protection for adj-sid explicit paths 4
2.3. Node-protection of binding-sid explicit paths 5
3. Detailed Solution using Context Tables 5
3.1. Building Context Tables 5
3.2. Building node protecting paths for node-sids 5
3.2.1. Building node protecting paths for adjacency-sids . . 7
3.3. Node protection for binding sids 8
3.4. Node protection for edge nodes 10
4. Security Considerations 11
5. IANA Considerations 11
6. Acknowledgments 11
7. References 11
7.1. Normative References 11
7.2. Informative References 11
Authors' Addresses 13

1. Introduction

It is possible for a routing device to completely go out of service abruptly due to power failure, hardware failure or software crashes. Node protection is an important property of the Fast Reroute mechanism. It provides protection against a node failure by rerouting traffic around the failed node. For example, the mechanisms described in Loop Free Alternates [RFC5286] and Remote loop free alternates [I-D.ietf-rtgwg-rlfa-node-protection] can be used to provide node protection to ensure minimal traffic loss after a node failure. The solutions to provide node protection in this draft use SPF based local protection mechanisms.

Section 2 describes problems with SR-TE paths and need for a specialized mechanism to provide node protection for the SR-TE paths. Section 3 describes the solution applied to paths built using adjacency-sids, node-sids and binding-sids. Section 3.4 describes the solution applied to egress node protection.

2. Node Failures Along SR-TE Paths

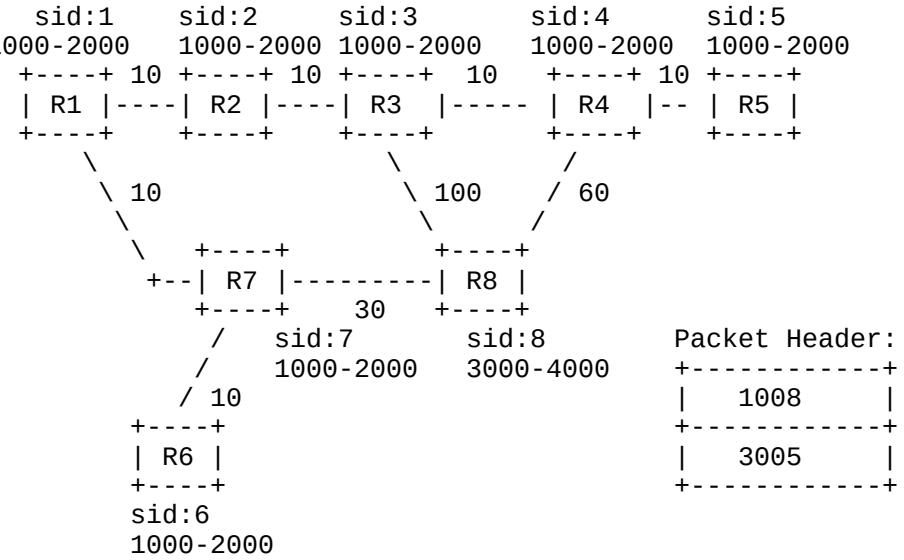


Figure 1: Sample Network

The topology shown in Figure 1. illustrates a sample network topology with SPRING enabled on each node. The SRGB and the segment index corresponding to each node is described in the topology diagram.

2.1. Node protection for node-sid explicit paths

Consider an explicit path from R1->R5 via R1->R7->R8->R4->R5. This path can be built using R1->R8 and R8->R5 shortest paths. The label stack contains two node-sids 1008 and 3005. The 1008 label would take the packet to R8 and get popped. The next label in the stack 3005 would take the packet to the destination R5. If the node R8 goes down, it is not possible for R7 to perform FRR without examining the second label in the incoming label stack (3005). R7 does not need to understand the meaning of label 3005 in order to perform normal forwarding in the absence of a failure. However, in order to support node protection, R7 will need to understand the meaning of label 3005 in order to determine where the packet is headed after R8.

Anycast addresses are in general advertised by more than one node and if per-prefix LFA calculation [RFC5286] is used node protecting paths can be found for the anycast sids. If a node protecting path is available for the anycast sid then the context table lookup mechanism would not be required. Otherwise, the anycast label has to be popped and next label looked up to find where the packet should be forwarded.

2.2. Node-protection for adj-sid explicit paths

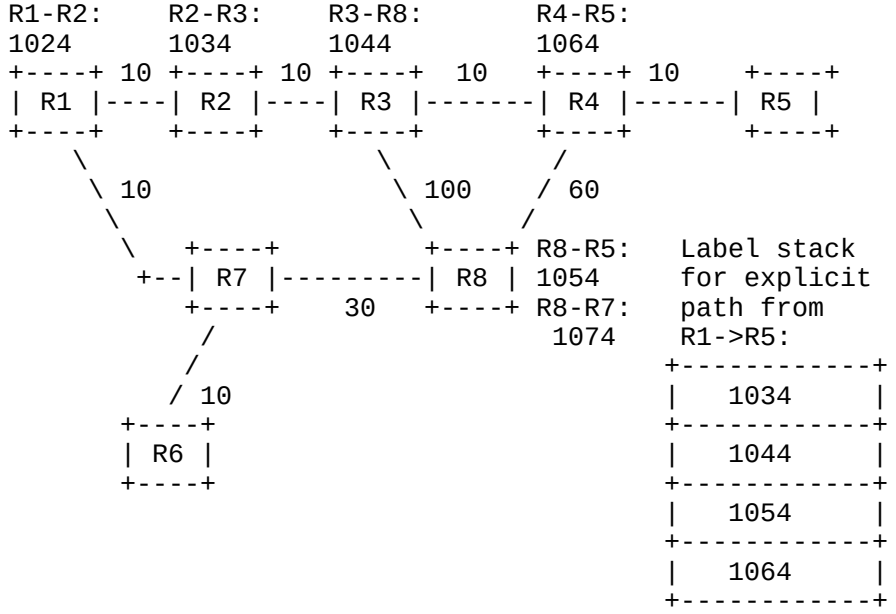


Figure 2: Explicit path using adjacency sids

Consider an explicit path from R1->R5 via R1->R2->R3->R8->R4->R5. This path can be built using adjacency sids, as shown in Figure 2. The diagram shows the adjacency sids advertised by each node required to realize this path, as well as the complete label stack. When a packet leaving R1 with this label stack reaches R3, the top of stack contains the label 1044 which will take the packet to R8. The next-next-hop in the path is R4. To provide protection for the failure of node R8, R3 would need to send the the packet to R4 without going through R8. However, the only way R3 can learn that the packet needs to go to the R4 is to examine the next label in the stack, label 1054.

2.3. Node-protection of binding-sid explicit paths

Binding sids (defined in SR architecture [I-D.ietf-spring-segment-routing]) allow the SR-TE path to be built using a hierarchy of sub-paths. The binding sid provides a single label to represent a set of nodes and links. If the node advertising the binding sid goes down, the traffic needs to be protected. The label stack involving the binding-sid contains next label in the stack which corresponds to the end point represented by the binding-sid. The penultimate node of the binding-sid advertiser cannot know the meaning of the next label in the stack.

3. Detailed Solution using Context Tables

3.1. Building Context Tables

[RFC5331] introduced the concept of Context Specific Label Spaces and there are various applications making use of this concept.A context label table on a router represents the Label Information Base (LIB) from the point of view of a particular neighbor . Context tables are built by constructing incoming label mappings advertised by the neighbor and the actions corresponding to those labels. The labels advertised by each node are local to the node and may not be unique across the segment routing domain. The context tables are separate tables built on a per-neighbor basis on every node to ensure they represent LIBs of a particular neighbor.

When a node learns the node-sid, SRGB, and adjacency-sids or binding-sids from a neighbor, the label mapping is added to the context table corresponding to that neighbor. The output actions for the label mapping are derived based on the actions that the neighbor would perform on receipt of the label.

The following section illustrates how the context table is constructed to allow the PLR to provide node-protecting paths for the next-next hops in the previous examples

3.2. Building node protecting paths for node-sids

R7's Transit Routing table

in-label	Out label
1001	Fwd to R1,
1002	swap 1002, Fwd to R1
1003	swap 1003, Fwd to R1
1004	swap 1004, Fwd to R1
1005,	swap 1005, Fwd to R1
1008,	pop, fwd to r8 *pop,lookup context.r8

* - Indicates backup path.

R7's Context Table for R8

in-label	Out label
3001	Fwd to R1,
3002	swap 1002, Fwd to R1
3003	swap 1003, Fwd to R1
3004	swap 1004, Fwd to R1
3005,	swap 1005, Fwd to R1

Figure 3: Transit routing table and Context Table at R7

The above Figure 3 shows the transit routing table and the context table of neighbor R8 built at R7 for the example network shown in

Figure 1. When the adjacency with R8 comes up, R7 builds the context table for R8 and adds the label mappings to the context table by adding the node-sid index of all the nodes to the SRGB advertised by R8. The output action is constructed by looking into the R7's SPF and backup SPF computations for the next-nexthop. The backup SPF computations as defined in LFA [RFC5286] are applicable here. The R7's SPF and backup SPF computations for the next-nexthop may provide multiple loop free primary or backup paths. A loop free path that does not include the failure node (R8 in this example) is chosen and downloaded to the context table.

R7's routing table entry for R8's sid i.e label 1008 will have a pop and forward action and the backup path SHOULD have action pop and lookup into the context table of R8. When the node R7 detects R8 goes down, R7's forwarding plane does a local repair and points to the backup path. When a packet whose top label is 1008 arrives at R7, the top label is popped, and the next label is looked up in the context table for R8. As shown in Figure 3, if the next label is 3005, the packet will be directed to R5 along a path that avoids R8.

3.2.1. Building node protecting paths for adjacency-sids

R3's Transit Routing table (partial)

in-label	Out label
1044	pop,Fwd to R8, *pop, lookup context.r8
1004	pop, Fwd to R4 *push 3004, fwd to R8

* - Indicates backup path.

R3's Context Table for R8 (partial)

in-label	Out label
1054	pop,Fwd to R4,
1074	swap 1007, Fwd to R2

Figure 4: Context Table at R3

The processing for the packet is similar to mechanism explained for node sids in section Section 3.2.

Figure 4 shows the context table constructed at R3 corresponding to R8 for the sample network shown in Figure 2. Adjacency sids are attached to the link advertisements in IGPs and the link advertisements contain the node information of the remote end. When R3 learns adjacency sids from R8, it builds context table for R8 which contains the adjacency labels advertised by R8 and the output action is built by looking at R3's own SPF and backup SPF computations for the remote end point of the link. Among the multiple primary/backup paths to the remote end of the link, a loop free path that does not pass through R8 is chosen.

3.3. Node protection for binding sids

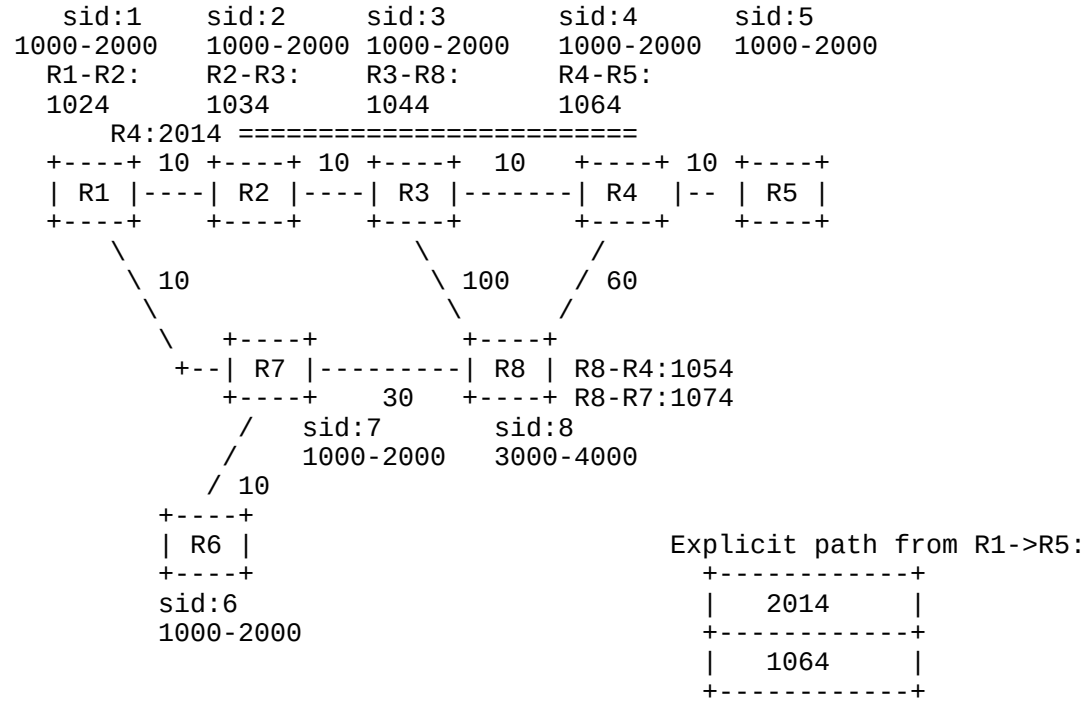


Figure 5: Node Protection for Binding SID

Figure Section 3.3 describes a sample network where R2 advertises a binding sid 2014 for the path R2->R3->R4. This mechanism is very useful in compressing the label stack depth as a sub-path can be represented using a single label. The explicit path R1->R2->R3->R4->R5 can be represented by 2 label stack as shown in above figure. If the node that advertises the binding-sid goes down, protection mechanisms are needed for the binding sid that the node advertised. A receiving node that programs a forwarding path for the binding sid should find a node protecting path to the last node of the path represented by the binding sid. In the above sample network, R1 programs a backup path for binding sid 2014 with the node protecting R-LFA path to R4 which consists of two labels [1008, 1004]. When the packet reached R4, it has the label 1064 in the label stack and can recognize this label and forward to R5. The node protecting path could be computed using various FRR technologies like LFA [RFC5286], Remote-LFA [RFC7490] , TI-LFA [I-D.francois-rtgwg-segment-routing-ti-lfa] etc.

3.4. Node protection for edge nodes

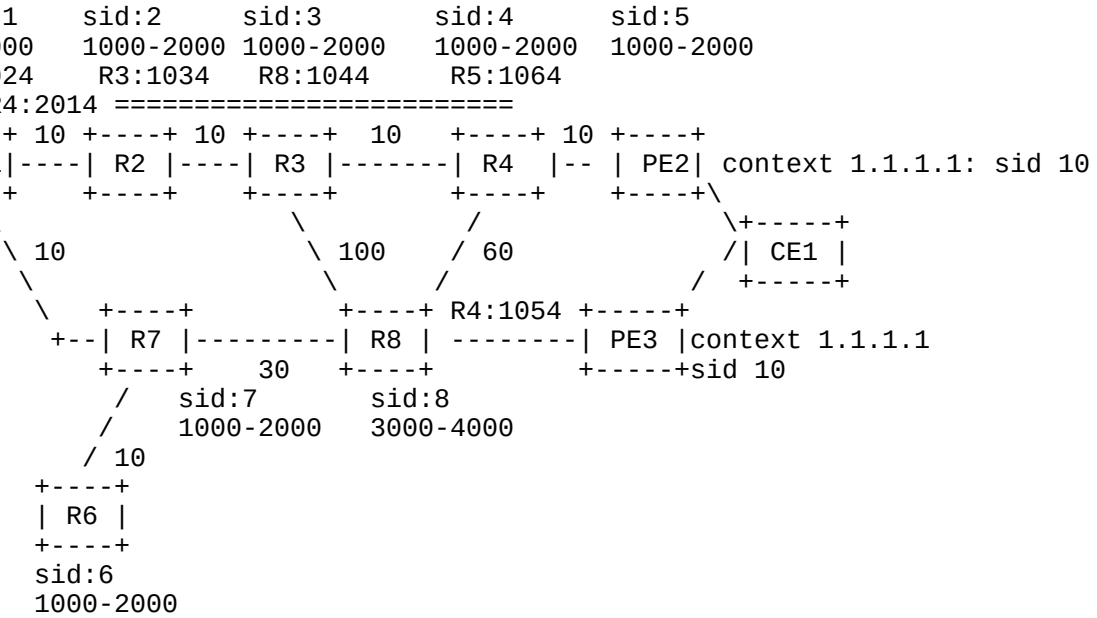


Figure 6: Node Protection for edge nodes

The node protection mechanisms that are described in previous sections depend on the assumption that the label below the top label in the label stack are understood in the IGP domain. If the edge node goes down, the label below the top label representing the edge node could be BGP service label or labels representing other applications. Service mirroring use case is described in [I-D.filsfils-spring-segment-routing-use-cases] The Customer edges are multi-homed to provider edges and one of the PE's acts in primary role and the other in protector role. The two PEs advertise a context ip address for each customer site and attaches a prefix-sid to the context. The protector PE advertises a binding sid with M bit set which implies mirroring capability for the context. Protector PE builds the context table for the BGP service labels advertised by the primary PE for the same context. The BGP service is built using stack of labels with context-sid at the bottom of the label stack.when the label ranges advertised by the PE2 and the penultimate node, Penultimate node does not understand the bottom label which is advertised by the node PE2. Any penultimate node of PE2 builds a context table for PE2 as explained in the section Section 3.1. This context table contains the sid for the context-id and output action is to pop the top label and replace with the binding sid that the

protector PE advertised for the context 1.1.1.1. The binding sid directs the protector PE to lookup the context table of Primary PE for the BGP service labels. The node protection mechanisms described in this document also ensure the edge node protection when uniform label range is not assigned across the entire IGP domain.

4. Security Considerations

TBD

5. IANA Considerations

6. Acknowledgments

Thanks to Eric Rosen for valuable inputs on the document and Chandrasekar Ramachandran for discussions on the topic.

7. References

7.1. Normative References

[RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, DOI 10.17487/RFC5286, September 2008, <http://www.rfc-editor.org/info/rfc5286>.

[RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, DOI 10.17487/RFC5331, August 2008, <http://www.rfc-editor.org/info/rfc5331>.

[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, <http://www.rfc-editor.org/info/rfc7490>.

7.2. Informative References

[I-D.filsfils-spring-segment-routing-use-cases] Filsfils, C., Francois, P., Previdi, S., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., Kini, S., and E. Crabbe, "Segment Routing Use Cases", draft-filsfils-spring-segment-routing-use-cases-01 (work in progress), October 2014.

[I-D.francois-rtgwg-segment-routing-ti-lfa]
Francois, P., Filsfils, C., Bashandy, A., and B. Decraene, "Topology Independent Fast Reroute using Segment Routing", draft-francois-rtgwg-segment-routing-ti-lfa-00 (work in progress), August 2015.

[I-D.ietf-rtgwg-rlfa-node-protection]
Sarkar, P., Hegde, S., Bowers, C., Gredler, H., and S. Litkowski, "Remote-LFA Node Protection and Manageability", draft-ietf-rtgwg-rlfa-node-protection-05 (work in progress), December 2015.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-07 (work in progress), December 2015.

[I-D.minto-rsvp-lsp-egress-fast-protection]
Jeganathan, J., Gredler, H., and Y. Shen, "RSVP-TE LSP egress fast-protection", draft-minto-rsvp-lsp-egress-fast-protection-03 (work in progress), November 2013.

[ISO10589]
"Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473), ISO/IEC 10589:2002, Second Edition.", Nov 2002.

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <http://www.rfc-editor.org/info/rfc1195>.

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

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <http://www.rfc-editor.org/info/rfc2328>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <http://www.rfc-editor.org/info/rfc5340>.

Authors' Addresses

Shraddha Hegde
Juniper Networks, Inc.
Exora Business Park
Bangalore, KA 560103
India

Email: shraddha@juniper.net

Chris Bowers
Juniper Networks, Inc.

Email: cbowers@juniper.net

Routing area
Internet-Draft
Intended status: Informational
Expires: April 20, 2019

S. Hegde
C. Bowers
Juniper Networks Inc.
S. Litkowski
Orange
X. Xu
Alibaba Inc.
F. Xu
Tencent

October 17, 2018

Node Protection for SR-TE Paths
draft-hegde-spring-node-protection-for-sr-te-paths-04

Abstract

Segment routing supports the creation of explicit paths using adjacency-sids, node-sids, and binding-sids. It is important to provide fast reroute (FRR) mechanisms to respond to failures of links and nodes in the Segment-Routed Traffic-Engineered(SR-TE) path. A point of local repair (PLR) can provide FRR protection against the failure of a link in an SR-TE path by examining only the first (top) label in the SR label stack. In order to protect against the failure of a node, a PLR may need to examine the second label in the stack as well in order to determine SR-TE path beyond the failed node. This document specifies how a PLR can use the first and second label in the label stack describing an SR-TE path to provide protection against node failures.

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

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 20, 2019.

Table of Contents

1. Introduction 2
2. Node Failures Along SR-TE Paths 3
2.1. Node protection for node-sid explicit paths 3
2.2. Node-Protection for Anycast-SIDs 4
2.3. Node-protection for adj-sid explicit paths 5
3. Detailed Solution using Context Tables 6
3.1. Building Context Tables 6
3.2. Node protection for node SIDs 7
3.3. Node protection for adjacency SIDs 8
3.4. Node protection for edge nodes 9
4. Security Considerations 10
5. IANA Considerations 10
6. Acknowledgments 10
7. References 10
7.1. Normative References 10
7.2. Informative References 10
Authors' Addresses 11

1. Introduction

It is possible for a routing device to completely go out of service abruptly due to power failure, hardware failure or software crashes. Node protection is an important property of the Fast Reroute mechanism. It provides protection against a node failure by rerouting traffic around the failed node. For example, the mechanisms described in Loop Free Alternates ([RFC5286]), Remote Loop

Free Alternates ([RFC8102]), and
[I-D.bashandy-rtgwg-segment-routing-ti-lfa] can be used to provide node protection to ensure minimal traffic loss after a node failure.

Section 2 describes problems with SR-TE paths and the need for a specialized mechanism to provide node protection for SR-TE paths. Section 3 describes the solution applied to paths built using adjacency-sids and node-sids.

2. Node Failures Along SR-TE Paths

The topology shown in Figure 1. illustrates a example network topology with SPRING enabled on each node.

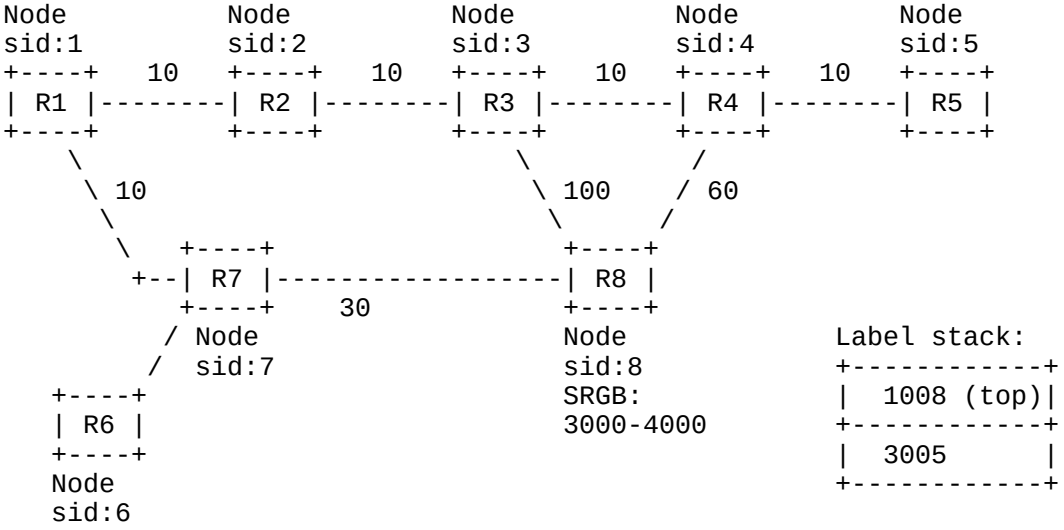


Figure 1: Example topology. The segment index for each node is shown in the diagram. All nodes have SRGB = [1000-2000], except for R8 which has SRGB = [3000-4000]. A label stack that represents the path R1->R7->R8->R4->R5 is shown as well.

2.1. Node protection for node-sid explicit paths

Consider an explicit path in the topology in Figure 1 from R1->R5 via R1->R7->R8->R4->R5. This path can be built using the shortest paths from R1-to-R8 and R8-to-R5. The label stack to instantiate this path contains two node-sids 1008 and 3005. The 1008 label will take the packet from R1 to R8 via R7 and get popped. The next label in the stack 3005 will take the packet from R8 to the destination R5 via R4. If the node R8 goes down, it is not possible for R7 to perform FRR

without examining the second label in the incoming label stack (3005).

Note that in the absence of a failure, R7 does not need to understand the meaning of the second label (3005) in order to perform normal forwarding. However, in order to support node protection, R7 will need to understand the meaning of label 3005 in order to determine where the packet is headed after R8.

2.2. Node-Protection for Anycast-SIDs

A prefix segment advertised as a node SID may only be advertised by one node in the network. Instead, an anycast prefix segment may be advertised by more than one node. In some situations, one can use anycast SIDs to construct SR-TE paths that are protected against node failure, without the need for the mechanism described in this document.

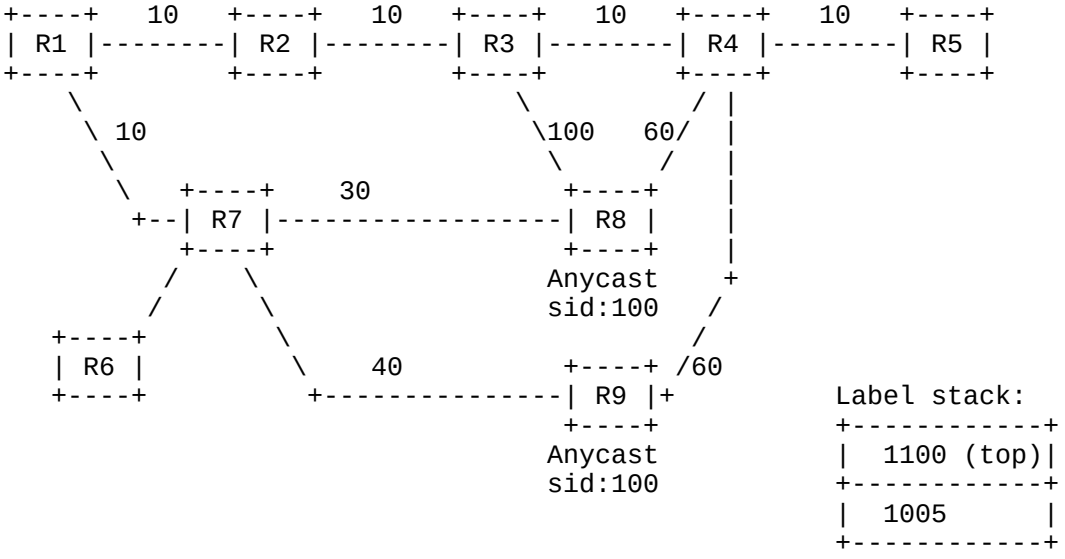


Figure 2: Topology illustrating use of anycast-sids to protect against node failures. All nodes have SRGB = [1000-2000].

An example of this is shown in Figure 2. In this example, R8 and R9 advertise an anycast SID of 100. The label stack in this example = [1100, 1005];. The top label (1100) corresponds to the anycast SID advertised by both R8 and R9. In the absence of a failure, the packet sent by R1 with this label stack will follow the path from R1->R5 along R1->R7->R8->R4->R5.

If R7 is performing a per-prefix LFA calculation [RFC5286], then R7 will install a backup next-hop to R9 for this anycast SID, protecting against the failure of the primary next-hop to R8. This backup path does not pass through R8, so it is would not be affected by a complete failure of node R8. As illustrated by this example, for some topologies node-protecting SR-TE paths can constructed through the use of anycast SIDs, as opposed to the mechanism described in this document.

2.3. Node-protection for adj-sid explicit paths

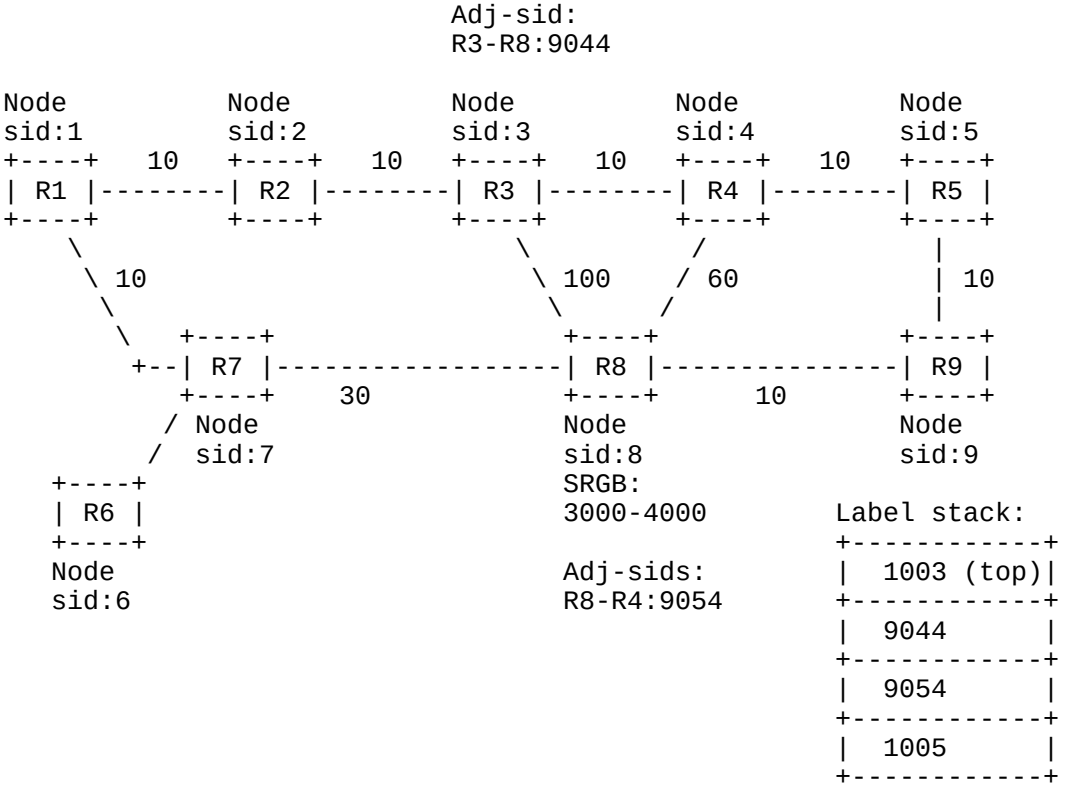


Figure 3: Explicit path using an adjacency sid. All nodes have SRGB = [1000-2000], except for R8 which has SRGB = [3000-4000].

Consider an explicit path from R1->R5 via R1->R2->R3->R8->R4->R5. This path can be built using a combination of node sids and adjacency sids, as shown in Figure 3. The diagram shows the label stack needed to instantiate this path, as well as several adjacency sids advertised by nodes involved in this path. When a packet leaving R1 with this label stack reaches R3, the top label is 9044, which will

take the packet to R8. The next-next-hop in the path is R4. To provide protection for the failure of node R8, R3 would need to send the the packet to R4 without going through R8. However, the only way R3 can learn that the packet needs to go to the R4 is to examine the next label in the stack, label 9054. Since R3 knows that R8 has advertised label 9054 as the adjacency segment for the link from R8 to R4, R3 knows that a backup path can merge back into the original explicit path at R4.

3. Detailed Solution using Context Tables

This section provides a detailed description of how to construct node-protecting backup paths for SR-TE paths using context tables. The end result of this description is externally visible forwarding behavior that can be specified as a packet arriving at a PLR with a particular incoming label stack and leaving the PLR on a particular outgoing interface with a particular outgoing label stack. There may be other methods of arriving at the same externally visible forwarding behavior as described in draft [I-D.bashandy-rtgwg-segment-routing-ti-lfa]. It is not the intent of this document to exclude other methods, as long as the externally visible forwarding behavior is the same as produced by this method.

3.1. Building Context Tables

[RFC5331] introduced the concept of Context Specific Label Spaces and there are various applications making use of this concept.A context label table on a router represents the Label Forwarding Information Base (LFIB) from the point of view of a particular neighbor . Context tables are built by constructing incoming label mappings advertised by the neighbor and the actions corresponding to those labels. The labels advertised by each node are local to the node and may not be unique across the segment routing domain. The context tables are separate tables built on a per-neighbor basis on every node to ensure they represent LFIBs of a particular neighbor.

When a PLR needs to protect an SR-TE path against the failure of a neighbor N, it creates a context table associated with N. This context table is populated with the following segment routing forwarding entries:

- All the Prefix-SIDs of the network. The programmed incoming label map uses the SRGB of N to compute the input label value. The NHLFE (Next Hop Label Forwarding Entry) is then constructed by looking into all the nexthops for the prefix-SID and choosing a loop-free path as explained in Section 3.2

- All the Adjacency SIDs advertised by N. The NHLFE is constructed as explained in Section 3.3

The following section illustrates how the context table is constructed to allow the PLR to provide node-protecting paths for the next-next hops in the topology shown in Figure 1 and Figure 3.

3.2. Node protection for node SIDs

Figure 4 shows the routing table entries on R7 corresponding to the node SIDs to reach R1 and R8 for the topology in Figure 1. In the absence of a failure, a packet with a label stack whose top label is 1008 will have its top label popped by R7 (assuming PHP behavior), and R7 will forward the packet to R8. When the interface to R8 is down, the backup next-hop entry is used. R7 will pop the top label of 1008, and use the context table that R7 computed for R8 to evaluate the next label on the stack.

R7's Routing Table (partial)
Transits routes for Node SIDs for R1 and R8

In label	Outgoing label action
1001	Primary: pop, fwd to R1
	Backup: pop, lookup context.r1
1008	Primary: pop, fwd to R8
	Backup: pop, lookup context.r8

R7's Context Table for R8 (context.r8, partial)

In label	Outgoing label action
3004	swap 1004, fwd to R1
3005	swap 1005, fwd to R1
3008	drop

Figure 4: Building node-protecting backup paths for SR-TE paths involving node SIDs

R7 builds context table for R8 using the following process. R7 computes the mapping of incoming label to node-sid that R8 expects to see based on the SRGB advertised by R8. In the example in Figure 1,

R7 can determine that R8 interprets in incoming label of 3005 as mapping to the the node SID for R5.

R7 then computes a loop-free backup path to reach R5 which is node-protecting with respect to the failure of R8. In this example, the backup path computed by R7 to reach R5 without passing through R8 can be achieved forwarding the packet to R1 with a top label of 1005, corresponding to the node SID for R5 in the context of R1's SRGB. The loop-free path computation may be based on a mechanism such as LFA, R-LFA, TI-LFA, or constraint based SPF avoiding failure. To populate the context table for R8, R7 maps the out label actions corresponding to the backup path to R5 to the incoming label 3005. This results in the entry for label 3005 showin in context.r8 in Figure 4.

Therefore, when a packet arrives at R7 with label stack = [1008, 3005], and the link from R7 to R8 has recently failed, R7 will use backup next-hop entry for label 1008 in its main routing table. Based on this entry, R7 will pop label 1008, and use context.r8 to lookup the new top label = 3005. R7 will swap label 3005 for 1005 and forward the packet to R1. This will get the packet to R5 on a node protecting backup path.

Note that R7 activates the node-protecting backup path when it detects that the link to R8 has failed. R7 does not know that node R8 has actually failed. However, the node-protecting backup path is computed assuming that the failure of the link to R8 implies that R8 has failed.

3.3. Node protection for adjacency SIDs

This section gives an example of how to constuct node-protecting backup paths when the SR-TE path uses adjacency SIDs. Figure 5 shows some of the routing table entries for R3 corresponding to the sample network shown in Figure 3. When the top label of the label stack is an adjacency SID, the PLR needs to recognize that in order to provide a node-protecting backup path, it needs to pop the top label and examine the next label in the context of the next-hop router identified by the top label adjacency SID. In this example, when R3 is constructing its routing table, it recognizes that label 9044 corresponds to a next-hop of R8, so it installs a backup entry, corresponding to the failure of the link to R8, when pops label 9044, and then examines the new top label in the context of R8.

R3's Routing Table (partial)
Transit route for Adj SID

In label	Outgoing label action
9044	Primary: pop, fwd to R8
	Backup: pop, lookup context.r8

R3's Context Table for R8 (context.r8, partial)

In label	Outgoing label action
3005	swap 1005, fwd to R4
9054	pop, fwd to R4

Figure 5: Building node-protecting backup paths for SR-TE paths involving adjacency SIDs

R3 constructs its context table for R8 by determining which labels R8 expects to receive to accomplish different forwarding actions. The entry for incoming label 3005 in context.r8 in Figure 5 corresponds to a node SID. This entry is computed using the methods described in Section 3.2

The entry for incoming label 9054 in context.r8 corresponds to an adjacency SID. R3 recognizes that R8 has advertised this adjacency SID for the link from R8 to R4 in Figure 3. So R3 determines the outgoing label action needed to reach R4 without passing through R8. This can be accomplished by popping the label 9054, and forwarding the packet directly on the link from R3 to R4.

3.4. Node protection for edge nodes

The node protection mechanism described in the previous sections depends on the assumption that the label immediately below the top label in the label stack is understood in the IGP domain. When the provider edge routers exchange service labels via BGP or some other non-IGP mechanism the bottom label is not understood in the IGP domain.

The egress node protection mechanisms described in the draft [I-D.ietf-mpls-egress-protection-framework] is applicable to this usecase and no additional changes will be required for SR based networks

4. Security Considerations

TBD

5. IANA Considerations

6. Acknowledgments

The authors would like to thank Peter Psenak and Bruno Decreane for their review and suggestions.

7. References

7.1. Normative References

[RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, DOI 10.17487/RFC5286, September 2008, <https://www.rfc-editor.org/info/rfc5286>.

[RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, DOI 10.17487/RFC5331, August 2008, <https://www.rfc-editor.org/info/rfc5331>.

7.2. Informative References

[I-D.bashandy-rtgwg-segment-routing-ti-lfa] Bashandy, A., Filsfils, C., Decraene, B., Litkowski, S., Francois, P., daniel.voyer@bell.ca, d., Clad, F., and P. Camarillo, "Topology Independent Fast Reroute using Segment Routing", draft-bashandy-rtgwg-segment-routing-ti-lfa-05 (work in progress), October 2018.

[I-D.ietf-mpls-egress-protection-framework] Shen, Y., Jeganathan, J., Decraene, B., Gredler, H., Michel, C., Chen, H., and Y. Jiang, "MPLS Egress Protection Framework", draft-ietf-mpls-egress-protection-framework-02 (work in progress), July 2018.

[I-D.ietf-spring-segment-routing] Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-15 (work in progress), January 2018.

[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>.

[RFC8102] Sarkar, P., Ed., Hegde, S., Bowers, C., Gredler, H., and S. Litkowski, "Remote-LFA Node Protection and Manageability", RFC 8102, DOI 10.17487/RFC8102, March 2017, <https://www.rfc-editor.org/info/rfc8102>.

Authors' Addresses

Shraddha Hegde
Juniper Networks Inc.
Exora Business Park
Bangalore, KA 560103
India

Email: shraddha@juniper.net

Chris Bowers
Juniper Networks Inc.

Email: cbowers@juniper.net

Stephane Litkowski
Orange

Email: stephane.litkowski@orange.com

Xiaohu Xu
Alibaba Inc.
Beijing
China

Email: xiaohu.xxh@alibaba-inc.com

Feng Xu
Tencent
China

Email: oliverxu@tencent.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 29, 2018

C. Filsfils, Ed.
S. Previdi, Ed.
Cisco Systems, Inc.
L. Ginsberg
Cisco Systems, Inc
B. Decraene
S. Litkowski
Orange
R. Shakir
Google, Inc.
January 25, 2018

Segment Routing Architecture
draft-ietf-spring-segment-routing-15

Abstract

Segment Routing (SR) leverages the source routing paradigm. A node steers a packet through an ordered list of instructions, called segments. A segment can represent any instruction, topological or service-based. A segment can have a semantic local to an SR node or global within an SR domain. SR allows to enforce a flow through any topological path while maintaining per-flow state only at the ingress nodes to the SR domain.

Segment Routing can be directly applied to the MPLS architecture with no change on the forwarding plane. A segment is encoded as an MPLS label. An ordered list of segments is encoded as a stack of labels. The segment to process is on the top of the stack. Upon completion of a segment, the related label is popped from the stack.

Segment Routing can be applied to the IPv6 architecture, with a new type of routing header. A segment is encoded as an IPv6 address. An ordered list of segments is encoded as an ordered list of IPv6 addresses in the routing header. The active segment is indicated by the Destination Address of the packet. The next active segment is indicated by a pointer in the new routing 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

Copyright Notice

Table of Contents

1. Introduction 3
2. Terminology 5
3. Link-State IGP Segments 8
3.1. IGP-Prefix Segment, Prefix-SID 8
3.1.1. Prefix-SID Algorithm 9
3.1.2. SR-MPLS 10
3.1.3. SRv6 11
3.2. IGP-Node Segment, Node-SID 12
3.3. IGP-Anycast Segment, Anycast SID 12
3.3.1. Anycast SID in SR-MPLS 12
3.4. IGP-Adjacency Segment, Adj-SID 15
3.4.1. Parallel Adjacencies 16
3.4.2. LAN Adjacency Segments 17
3.5. Inter-Area Considerations 18

4. BGP Peering Segments 19
4.1. BGP Prefix Segment 19
4.2. BGP Peering Segments 19
5. Binding Segment 20
5.1. IGP Mirroring Context Segment 20
6. Multicast 21
7. IANA Considerations 21
8. Security Considerations 21
8.1. SR-MPLS 21
8.2. SRv6 23
8.3. Congestion Control 24
9. Manageability Considerations 24
10. Contributors 25
11. Acknowledgements 26
12. References 26
12.1. Normative References 26
12.2. Informative References 27
Authors' Addresses 30

1. Introduction

Segment Routing (SR) leverages the source routing paradigm. A node steers a packet through an SR Policy instantiated as an ordered list of instructions called segments. A segment can represent any instruction, topological or service-based. A segment can have a semantic local to an SR node or global within an SR domain. SR supports per-flow explicit routing while maintaining per-flow state only at the ingress nodes to the SR domain.

A segment is often referred to by its Segment Identifier (SID).

A segment may be associated with a topological instruction. A topological local segment may instruct a node to forward the packet via a specific outgoing interface. A topological global segment may instruct an SR domain to forward the packet via a specific path to a destination. Different segments may exist for the same destination, each with different path objectives (e.g., which metric is minimized, what constraints are specified).

A segment may be associated with a service instruction (e.g. the packet should be processed by a container or VM associated with the segment). A segment may be associated with a QoS treatment (e.g., shape the packets received with this segment at x Mbps).

The SR architecture supports any type of instruction associated with a segment.

The SR architecture supports any type of control-plane: distributed, centralized or hybrid.

In a distributed scenario, the segments are allocated and signaled by IS-IS or OSPF or BGP. A node individually decides to steer packets on a source-routed policy (e.g., pre-computed local protection [I-D.ietf-spring-resiliency-use-cases]) . A node individually computes the source-routed policy.

In a centralized scenario, the segments are allocated and instantiated by an SR controller. The SR controller decides which nodes need to steer which packets on which source-routed policies. The SR controller computes the source-routed policies. The SR architecture does not restrict how the controller programs the network. Likely options are NETCONF, PCEP and BGP. The SR architecture does not restrict the number of SR controllers. Specifically multiple SR controllers may program the same SR domain. The SR architecture allows these SR controllers to discover which SID’s are instantiated at which nodes and which sets of local (SRLB) and global labels (SRGB) are available at which node.

A hybrid scenario complements a base distributed control-plane with a centralized controller. For example, when the destination is outside the IGP domain, the SR controller may compute a source-routed policy on behalf of an IGP node. The SR architecture does not restrict how the nodes which are part of the distributed control-plane interact with the SR controller. Likely options are PCEP and BGP.

Hosts MAY be part of an SR Domain. A centralized controller can inform hosts about policies either by pushing these policies to hosts or responding to requests from hosts.

The SR architecture can be instantiated on various dataplanes. This document introduces two dataplane instantiations of SR: SR over MPLS (SR-MPLS) and SR over IPv6 (SRv6).

Segment Routing can be directly applied to the MPLS architecture with no change on the forwarding plane [I-D.ietf-spring-segment-routing-mpls] A segment is encoded as an MPLS label. An SR Policy is instantiated as a stack of labels. The segment to process (the active segment) is on the top of the stack. Upon completion of a segment, the related label is popped from the stack.

Segment Routing can be applied to the IPv6 architecture with a new type of routing header called the SR header (SRH) [I-D.ietf-6man-segment-routing-header] . An instruction is associated with a segment and encoded as an IPv6 address. An SRv6 segment is

also called an SRv6 SID. An SR Policy is instantiated as an ordered list of SRv6 SID's in the routing header. The active segment is indicated by the Destination Address(DA) of the packet. The next active segment is indicated by the SegmentsLeft (SL) pointer in the SRH. When an SRv6 SID is completed, the SL is decremented and the next segment is copied to the DA. When a packet is steered on an SR policy, the related SRH is added to the packet.

In the context of an IGP-based distributed control-plane, two topological segments are defined: the IGP adjacency segment and the IGP prefix segment.

In the context of a BGP-based distributed control-plane, two topological segments are defined: the BGP peering segment and the BGP prefix segment.

The headend of an SR Policy binds a SID (called Binding segment or BSID) to its policy. When the headend receives a packet with active segment matching the BSID of a local SR Policy, the headend steers the packet into the associated SR Policy.

This document defines the IGP, BGP and Binding segments for the SR-MPLS and SRv6 dataplanes.

Note: This document defines the architecture for Segment Routing, including definitions of basic objects and functions and a description of the overall design. It does NOT define the means of implementing the architecture - that is contained in numerous referencing documents, some of which are mentioned in this document as a convenience to the reader.

2. Terminology

SR-MPLS: the instantiation of SR on the MPLS dataplane

SRv6: the instantiation of SR on the IPv6 dataplane.

Segment: an instruction a node executes on the incoming packet (e.g., forward packet according to shortest path to destination, or, forward packet through a specific interface, or, deliver the packet to a given application/service instance).

SID: a segment identifier. Note that the term SID is commonly used in place of the term Segment, though this is technically imprecise as it overlooks any necessary translation.

SR-MPLS SID: an MPLS label or an index value into an MPLS label space explicitly associated with the segment.

SRv6 SID: an IPv6 address explicitly associated with the segment.

Segment Routing Domain (SR Domain): the set of nodes participating in the source based routing model. These nodes may be connected to the same physical infrastructure (e.g., a Service Provider's network). They may as well be remotely connected to each other (e.g., an enterprise VPN or an overlay). If multiple protocol instances are deployed, the SR domain most commonly includes all of the protocol instances in a network. However, some deployments may wish to sub-divide the network into multiple SR domains, each of which includes one or more protocol instances. It is expected that all nodes in an SR Domain are managed by the same administrative entity.

Active Segment: the segment that is used by the receiving router to process the packet. In the MPLS dataplane it is the top label. In the IPv6 dataplane it is the destination address. [I-D.ietf-6man-segment-routing-header].

PUSH: the instruction consisting of the insertion of a segment at the top of the segment list. In SR-MPLS the top of the segment list is the topmost (outer) label of the label stack. In SRv6, the top of the segment list is represented by the first segment in the Segment Routing Header as defined in [I-D.ietf-6man-segment-routing-header].

NEXT: when the active segment is completed, NEXT is the instruction consisting of the inspection of the next segment. The next segment becomes active. In SR-MPLS, NEXT is implemented as a POP of the top label. In SRv6, NEXT is implemented as the copy of the next segment from the SRH to the Destination Address of the IPv6 header.

CONTINUE: the active segment is not completed and hence remains active. In SR-MPLS, CONTINUE instruction is implemented as a SWAP of the top label. [RFC3031] In SRv6, this is the plain IPv6 forwarding action of a regular IPv6 packet according to its Destination Address.

SR Global Block (SRGB): the set of global segments in the SR Domain. If a node participates in multiple SR domains, there is one SRGB for each SR domain. In SR-MPLS, SRGB is a local property of a node and identifies the set of local labels reserved for global segments. In SR-MPLS, using identical SRGBs on all nodes within the SR Domain is strongly recommended. Doing so eases operations and troubleshooting as the same label represents the same global segment at each node. In SRv6, the SRGB is the set of global SRv6 SIDs in the SR Domain.

SR Local Block (SRLB): local property of an SR node. If a node participates in multiple SR domains, there is one SRLB for each SR domain. In SR-MPLS, SRLB is a set of local labels reserved for local segments. In SRv6, SRLB is a set of local IPv6 addresses reserved

for local SRv6 SID's. In a controller-driven network, some controllers or applications may use the control plane to discover the available set of local segments.

Global Segment: a segment which is part of the SRGB of the domain. The instruction associated to the segment is defined at the SR Domain level. A topological shortest-path segment to a given destination within an SR domain is a typical example of a global segment.

Local Segment: In SR-MPLS, this is a local label outside the SRGB. It may be part of the explicitly advertised SRLB. In SRv6, this can be any IPv6 address i.e., the address may be part of the SRGB but used such that it has local significance. The instruction associated to the segment is defined at the node level.

IGP Segment: the generic name for a segment attached to a piece of information advertised by a link-state IGP, e.g. an IGP prefix or an IGP adjacency.

IGP-Prefix Segment: an IGP-Prefix Segment is an IGP Segment representing an IGP prefix. When an IGP-Prefix Segment is global within the SR IGP instance/topology it identifies an instruction to forward the packet along the path computed using the routing algorithm specified in the algorithm field, in the topology and the IGP instance where it is advertised. Also referred to as Prefix Segment.

Prefix SID: the SID of the IGP-Prefix Segment.

IGP-Anycast Segment: an IGP-Anycast Segment is an IGP-Prefix Segment which identify an anycast prefix advertised by a set of routers.

Anycast-SID: the SID of the IGP-Anycast Segment.

IGP-Adjacency Segment: an IGP-Adjacency Segment is an IGP Segment attached to a unidirectional adjacency or a set of unidirectional adjacencies. By default, an IGP-Adjacency Segment is local (unless explicitly advertised otherwise) to the node that advertises it. Also referred to as Adjacency Segment.

Adj-SID: the SID of the IGP-Adjacency Segment.

IGP-Node Segment: an IGP-Node Segment is an IGP-Prefix Segment which identifies a specific router (e.g., a loopback). Also referred to as Node Segment.

Node-SID: the SID of the IGP-Node Segment.

SR Policy: an ordered list of segments. The headend of an SR Policy steers packets onto the SR policy. The list of segments can be specified explicitly in SR-MPLS as a stack of labels and in SRv6 as an ordered list of SRv6 SID's. Alternatively, the list of segments is computed based on a destination and a set of optimization objective and constraints (e.g., latency, affinity, SRLG, ...). The computation can be local or delegated to a PCE server. An SR policy can be configured by the operator, provisioned via NETCONF [RFC6241] or provisioned via PCEP [RFC5440] . An SR policy can be used for traffic-engineering, OAM or FRR reasons.

Segment List Depth: the number of segments of an SR policy. The entity instantiating an SR Policy at a node N should be able to discover the depth insertion capability of the node N. For example, the PCEP SR capability advertisement described in [I-D.ietf-pce-segment-routing] is one means of discovering this capability.

Forwarding Information Base (FIB): the forwarding table of a node

3. Link-State IGP Segments

Within an SR domain, an SR-capable IGP node advertises segments for its attached prefixes and adjacencies. These segments are called IGP segments or IGP SIDs. They play a key role in Segment Routing and use-cases as they enable the expression of any path throughout the SR domain. Such a path is either expressed as a single IGP segment or a list of multiple IGP segments.

Advertisement of IGP segments requires extensions in link-state IGP protocols. These extensions are defined in
[I-D.ietf-isis-segment-routing-extensions]
[I-D.ietf-ospf-segment-routing-extensions]
[I-D.ietf-ospf-ospfv3-segment-routing-extensions]

3.1. IGP-Prefix Segment, Prefix-SID

An IGP-Prefix segment is an IGP segment attached to an IGP prefix. An IGP-Prefix segment is global (unless explicitly advertised otherwise) within the SR domain. The context for an IGP-Prefix segment includes the prefix, topology, and algorithm. Multiple SIDs MAY be allocated to the same prefix so long as the tuple <prefix, topology, algorithm> is unique.

Multiple instances and topologies are defined in IS-IS and OSPF in: [RFC5120], [RFC8202], [RFC6549] and [RFC4915].

3.1.1. Prefix-SID Algorithm

Segment Routing supports the use of multiple routing algorithms i.e, different constraint based shortest path calculations can be supported. An algorithm identifier is included as part of a Prefix-SID advertisement. Specification of how an algorithm specific path calculation is done is required in the document defining the algorithm.

This document defines two algorithms:

- "Shortest Path": this algorithm is the default behavior. The packet is forwarded along the well known ECMP-aware SPF algorithm employed by the IGPs. However it is explicitly allowed for a midpoint to implement another forwarding based on local policy. The "Shortest Path" algorithm is in fact the default and current behavior of most of the networks where local policies may override the SPF decision.

- "Strict Shortest Path (Strict-SPF)": This algorithm mandates that the packet is forwarded according to ECMP-aware SPF algorithm and instructs any router in the path to ignore any possible local policy overriding the SPF decision. The SID advertised with Strict-SPF algorithm ensures that the path the packet is going to take is the expected, and not altered, SPF path. Note that Fast Reroute (FRR) [RFC5714] mechanisms are still compliant with the Strict Shortest Path. In other words, a packet received with a Strict-SPF SID may be rerouted through a FRR mechanism. Strict-SPF uses the same topology used by "Shortest Path". Obviously, nodes which do not support Strict-SPF will not install forwarding entries for this algorithm. Restricting the topology only to those nodes which support this algorithm will not produce the desired forwarding paths since the desired behavior is to follow the path calculated by "Shortest Path". Therefore, a source SR node MUST NOT use a source-routing policy containing a strict SPF segment if the path crosses a node not supporting the strict-SPF algorithm.

An IGP-Prefix Segment identifies the path, to the related prefix, computed as per the associated algorithm. A packet injected anywhere within the SR domain with an active Prefix-SID is expected to be forwarded along a path computed using the specified algorithm. For this to be possible, a fully connected topology of routers supporting the specified algorithm is required.

3.1.2. SR-MPLS

When SR is used over the MPLS dataplane SIDs are an MPLS label or an index into an MPLS label space (either SRGB or SRLB).

Where possible, it is recommended that identical SRGBs be configured on all nodes in an SR Domain. This simplifies troubleshooting as the same label will be associated with the same prefix on all nodes. In addition, it simplifies support for anycast as detailed in Section 3.3.

The following behaviors are associated with SR operating over the MPLS dataplane:

- the IGP signaling extension for IGP-Prefix segment includes a flag to indicate whether directly connected neighbors of the node on which the prefix is attached should perform the NEXT operation or the CONTINUE operation when processing the SID. This behavior is equivalent to Penultimate Hop Popping (NEXT) or Ultimate Hop Popping (CONTINUE) in MPLS.

- A Prefix-SID is allocated in the form of an MPLS label (or an index in the SRGB) according to a process similar to IP address allocation. Typically, the Prefix-SID is allocated by policy by the operator (or NMS) and the SID very rarely changes.

- While SR allows to attach a local segment to an IGP prefix, it is specifically assumed that when the terms "IGP-Prefix Segment" and "Prefix-SID" are used, the segment is global (the SID is allocated from the SRGB or as an index into the advertised SRGB). This is consistent with all the described use-cases that require global segments attached to IGP prefixes.

- The allocation process MUST NOT allocate the same Prefix-SID to different IP prefixes.

- If a node learns a Prefix-SID having a value that falls outside the locally configured SRGB range, then the node MUST NOT use the Prefix-SID and SHOULD issue an error log reporting a misconfiguration.

- If a node N advertises Prefix-SID SID-R for a prefix R that is attached to N, if N specifies CONTINUE as the operation to be performed by directly connected neighbors, N MUST maintain the following FIB entry:

```
   Incoming Active Segment: SID-R
   Ingress Operation: NEXT
   Egress interface: NULL
```

- A remote node M MUST maintain the following FIB entry for any learned Prefix-SID SID-R attached to IP prefix R:

```
  Incoming Active Segment: SID-R
  Ingress Operation:
     If the next-hop of R is the originator of R
     and instructed to remove the active segment: NEXT
     Else: CONTINUE
  Egress interface: the interface towards the next-hop along the
                    path computed using the algorithm advertised with
                    the SID toward prefix R.
```

As Prefix-SIDs are specific to a given algorithm, if traffic associated with an algorithm arrives at a node which does not support that algorithm the traffic will be dropped as there will be no forwarding entry matching the incoming label.

3.1.3. SRv6

When SR is used over the IPv6 dataplane:

- A Prefix-SID is an IPv6 address.

- An operator MUST explicitly instantiate an SRv6 SID. IPv6 node addresses are not SRv6 SIDs by default.

A node N advertising an IPv6 address R usable as a segment identifier MUST maintain the following FIB entry:

```
   Incoming Active Segment: R
   Ingress Operation: NEXT
   Egress interface: NULL
```

Note that forwarding to R does not require an entry in the FIBs of all other routers for R. Forwarding can be and most often will be achieved by a shorter mask prefix which covers R.

Independent of Segment Routing support, any remote IPv6 node will maintain a plain IPv6 FIB entry for any prefix, no matter if the prefix represents a segment or not. This allows forwarding of packets to the node which owns the SID even by nodes which do not support Segment Routing.

Support of multiple algorithms applies to SRv6. Since algorithm specific SIDs are simply IPv6 addresses, algorithm specific forwarding entries can be achieved by assigning algorithm specific subnets to the (set of) algorithm specific SIDs which a node allocates.

Nodes which do not support a given algorithm may still have a FIB entry covering an algorithm specific address even though an algorithm specific path has not been calculated by that node. This is mitigated by the fact that nodes which do not support a given algorithm will not be included in the topology associated with that algorithm specific SPF and so traffic using the algorithm specific destination will normally not flow via the excluded node. If such traffic were to arrive and be forwarded by such a node, it will still progress towards the destination node. The nexthop will either be a node which supports the algorithm - in which case the packet will be forwarded along algorithm specific paths (or be dropped if none are available) - or the nexthop will be a node which does NOT support the algorithm - in which case the packet will continue to be forwarded along Algorithm 0 paths towards the destination node.

3.2. IGP-Node Segment, Node-SID

An IGP Node-SID MUST NOT be associated with a prefix that is owned by more than one router within the same routing domain.

3.3. IGP-Anycast Segment, Anycast SID

An "Anycast Segment" or "Anycast SID" enforces the ECMP-aware shortest-path forwarding towards the closest node of the anycast set. This is useful to express macro-engineering policies or protection mechanisms.

An IGP-Anycast segment MUST NOT reference a particular node.

Within an anycast group, all routers in an SR domain MUST advertise the same prefix with the same SID value.

3.3.1. Anycast SID in SR-MPLS

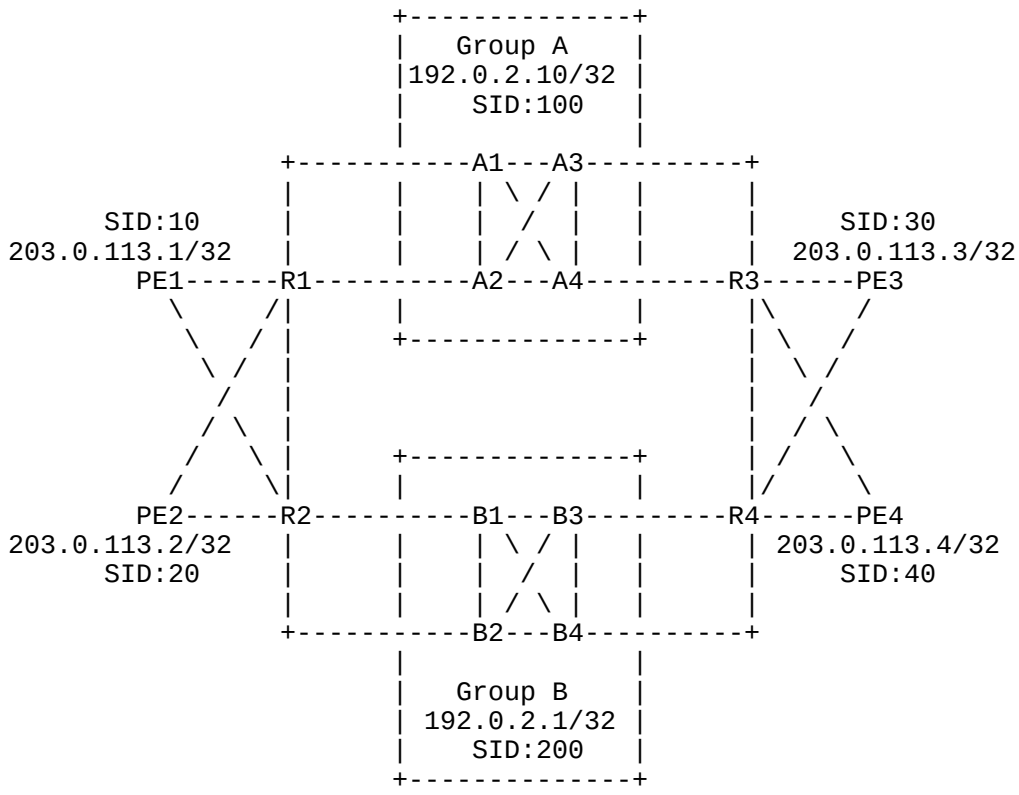


Figure 1: Transit device groups

The figure above describes a network example with two groups of transit devices. Group A consists of devices {A1, A2, A3 and A4}. They are all provisioned with the anycast address 192.0.2.10/32 and the anycast SID 100.

Similarly, group B consists of devices {B1, B2, B3 and B4} and are all provisioned with the anycast address 192.0.2.1/32, anycast SID 200. In the above network topology, each PE device has a path to each of the groups A and B.

PE1 can choose a particular transit device group when sending traffic to PE3 or PE4. This will be done by pushing the anycast SID of the group in the stack.

Processing the anycast, and subsequent segments, requires special care.

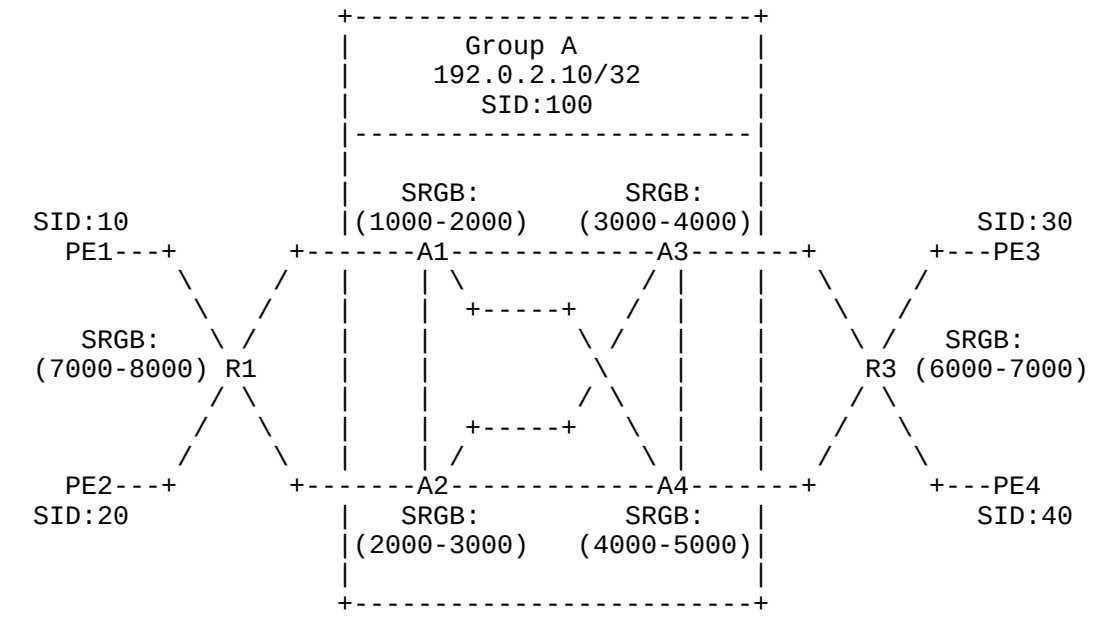


Figure 2: Transit paths via anycast group A

Considering an MPLS deployment, in the above topology, if device PE1 (or PE2) requires to send a packet to the device PE3 (or PE4) it needs to encapsulate the packet in an MPLS payload with the following stack of labels.

- Label allocated by R1 for anycast SID 100 (outer label).
- Label allocated by the nearest router in group A for SID 30 (for destination PE3).

While the first label is easy to compute, in this case since there are more than one topologically nearest devices (A1 and A2), unless A1 and A2 allocated the same label value to the same prefix, determining the second label is impossible. Devices A1 and A2 may be devices from different hardware vendors. If both don't allocate the same label value for SID 30, it is impossible to use the anycast group "A" as a transit anycast group towards PE3. Hence, PE1 (or PE2) cannot compute an appropriate label stack to steer the packet exclusively through the group A devices. Same holds true for devices PE3 and PE4 when trying to send a packet to PE1 or PE2.

To ease the use of anycast segment, it is recommended to configure identical SRGBs on all nodes of a particular anycast group. Using

this method, as mentioned above, computation of the label following the anycast segment is straightforward.

Using anycast segment without configuring identical SRGBs on all nodes belonging to the same device group may lead to misrouting (in an MPLS VPN deployment, some traffic may leak between VPNs).

3.4. IGP-Adjacency Segment, Adj-SID

The adjacency is formed by the local node (i.e., the node advertising the adjacency in the IGP) and the remote node (i.e., the other end of the adjacency). The local node MUST be an IGP node. The remote node may be an adjacent IGP neighbor or a non-adjacent neighbor (e.g., a Forwarding Adjacency, [RFC4206]).

A packet injected anywhere within the SR domain with a segment list {SN, SNL}, where SN is the Node-SID of node N and SNL is an Adj-SID attached by node N to its adjacency over link L, will be forwarded along the shortest-path to N and then be switched by N, without any IP shortest-path consideration, towards link L. If the Adj-SID identifies a set of adjacencies, then the node N load-balances the traffic among the various members of the set.

Similarly, when using a global Adj-SID, a packet injected anywhere within the SR domain with a segment list {SNL}, where SNL is a global Adj-SID attached by node N to its adjacency over link L, will be forwarded along the shortest-path to N and then be switched by N, without any IP shortest-path consideration, towards link L. If the Adj-SID identifies a set of adjacencies, then the node N does load-balance the traffic among the various members of the set. The use of global Adj-SID allows to reduce the size of the segment list when expressing a path at the cost of additional state (i.e.: the global Adj-SID will be inserted by all routers within the area in their forwarding table).

An "IGP Adjacency Segment" or "Adj-SID" enforces the switching of the packet from a node towards a defined interface or set of interfaces. This is key to theoretically prove that any path can be expressed as a list of segments.

The encodings of the Adj-SID include a set of flags supporting the following functionalities:

- Eligible for Protection (e.g., using IPFRR or MPLS-FRR). Protection allows that in the event the interface(s) associated with the Adj-SID are down, that the packet can still be forwarded via an alternate path. The use of protection is clearly a policy

based decision i.e., for a given policy protection may or may not be desirable.

- Indication whether the Adj-SID has local or global scope. Default scope SHOULD be Local.

- Indication whether the Adj-SID is persistent across control plane restarts. Persistence is a key attribute in ensuring that an SR Policy does not temporarily result in misforwarding due to reassignment of an Adj-SID.

A weight (as described below) is also associated with the Adj-SID advertisement.

A node SHOULD allocate one Adj-SID for each of its adjacencies.

A node MAY allocate multiple Adj-SIDs for the same adjacency. An example is to support an Adj-SID which is eligible for protection and an Adj-SID which is NOT eligible for protection.

A node MAY associate the same Adj-SID to multiple adjacencies.

In order to be able to advertise in the IGP all the Adj-SIDs representing the IGP adjacencies between two nodes, parallel adjacency suppression MUST NOT be performed by the IGP.

When a node binds an Adj-SID to a local data-link L, the node MUST install the following FIB entry:

```
Incoming Active Segment: V
Ingress Operation: NEXT
Egress Interface: L
```

The Adj-SID implies, from the router advertising it, the forwarding of the packet through the adjacency(ies) identified by the Adj-SID, regardless of its IGP/SPF cost. In other words, the use of adjacency segments overrides the routing decision made by the SPF algorithm.

3.4.1. Parallel Adjacencies

Adj-SIDs can be used in order to represent a set of parallel interfaces between two adjacent routers.

A node MUST install a FIB entry for any locally originated adjacency segment (Adj-SID) of value W attached to a set of links B with:

```
   Incoming Active Segment: W
   Ingress Operation: NEXT
   Egress interface: load-balance between any data-link within set B
```

When parallel adjacencies are used and associated to the same Adj-SID, and in order to optimize the load balancing function, a "weight" factor can be associated to the Adj-SID advertised with each adjacency. The weight tells the ingress (or an SDN/orchestration system) about the load-balancing factor over the parallel adjacencies. As shown in Figure 3, A and B are connected through two parallel adjacencies

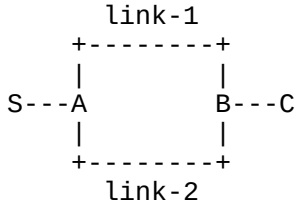


Figure 3: Parallel Links and Adj-SIDs

Node A advertises following Adj-SIDs and weights:

- Link-1: Adj-SID 1000, weight: 1

- Link-2: Adj-SID 1000, weight: 2

Node S receives the advertisements of the parallel adjacencies and understands that by using Adj-SID 1000 node A will load-balance the traffic across the parallel links (link-1 and link-2) according to a 1:2 ratio i.e., twice as many packets will flow over Link-2 as compared to Link-1.

3.4.2. LAN Adjacency Segments

In LAN subnetworks, link-state protocols define the concept of Designated Router (DR, in OSPF) or Designated Intermediate System (DIS, in IS-IS) that conduct flooding in broadcast subnetworks and that describe the LAN topology in a special routing update (OSPF Type2 LSA or IS-IS Pseudonode LSP).

The difficulty with LANs is that each router only advertises its connectivity to the DR/DIS and not to each of the individual nodes in the LAN. Therefore, additional protocol mechanisms (IS-IS and OSPF) are necessary in order for each router in the LAN to advertise an Adj-SID associated to each neighbor in the LAN.

3.5. Inter-Area Considerations

In the following example diagram it is assumed that the all areas are part of a single SR Domain.

The example here below assumes the IPv6 control plane with the MPLS dataplane.

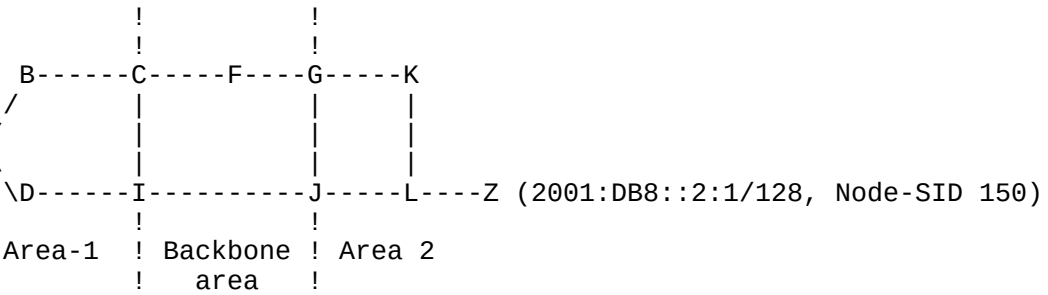


Figure 4: Inter-Area Topology Example

In area 2, node Z allocates Node-SID 150 to his local IPv6 prefix 2001:DB8::2:1/128.

Area Border Routers (ABR) G and J will propagate the prefix and its SIDs into the backbone area by creating a new instance of the prefix according to normal inter-area/level IGP propagation rules.

Nodes C and I will apply the same behavior when leaking prefixes from the backbone area down to area 1. Therefore, node S will see prefix 2001:DB8::2:1/128 with Prefix-SID 150 and advertised by nodes C and I.

It therefore results that a Prefix-SID remains attached to its related IGP Prefix through the inter-area process, which is the expected behavior in a single SR Domain.

When node S sends traffic to 2001:DB8::2:1/128, it pushes Node-SID(150) as active segment and forward it to A.

When packet arrives at ABR I (or C), the ABR forwards the packet according to the active segment (Node-SID(150)). Forwarding continues across area borders, using the same Node-SID(150), until the packet reaches its destination.

4. BGP Peering Segments

BGP segments may be allocated and distributed by BGP.

4.1. BGP Prefix Segment

A BGP-Prefix segment is a BGP segment attached to a BGP prefix.

A BGP-Prefix segment is global (unless explicitly advertised otherwise) within the SR domain.

The BGP Prefix SID is the BGP equivalent to the IGP Prefix Segment.

A likely use-case for the BGP Prefix Segment is an IGP-free hyper-scale spine-leaf topology where connectivity is learned solely via BGP [RFC7938]

4.2. BGP Peering Segments

In the context of BGP Egress Peer Engineering (EPE), as described in [I-D.ietf-spring-segment-routing-central-epe], an EPE enabled Egress PE node MAY advertise segments corresponding to its attached peers. These segments are called BGP peering segments or BGP peering SIDs. They enable the expression of source-routed inter-domain paths.

An ingress border router of an AS may compose a list of segments to steer a flow along a selected path within the AS, towards a selected egress border router C of the AS and through a specific peer. At minimum, a BGP peering Engineering policy applied at an ingress PE involves two segments: the Node SID of the chosen egress PE and then the BGP peering segment for the chosen egress PE peer or peering interface.

Three types of BGP peering segments/SIDs are defined: PeerNode SID, PeerAdj SID and PeerSet SID.

- PeerNode SID: a BGP PeerNode segment/SID is a local segment. At the BGP node advertising it, its semantics is:
  - SR header operation: NEXT.
  - Next-Hop: the connected peering node to which the segment is related.
- PeerAdj SID: a BGP PeerAdj segment/SID is a local segment. At the BGP node advertising it, the semantic is:
  - SR header operation: NEXT.

    * Next-Hop: the peer connected through the interface to which the segment is related.

  o PeerSet SID. a BGP PeerSet segment/SID is a local segment. At the BGP node advertising it, the semantic is:

    * SR header operation: NEXT.

    * Next-Hop: load-balance across any connected interface to any peer in the related group.

    A peer set could be all the connected peers from the same AS or a subset of these. A group could also span across AS. The group definition is a policy set by the operator.

  The BGP extensions necessary in order to signal these BGP peering segments are defined in [I-D.ietf-idr-bgpls-segment-routing-epe]

5. Binding Segment

In order to provide greater scalability, network opacity, and service independence, SR utilizes a Binding SID (BSID). The BSID is bound to an SR policy, instantiation of which may involve a list of SIDs. Any packets received with active segment = BSID are steered onto the bound SR Policy.

A BSID may either be a local or a global SID. If local, a BSID SHOULD be allocated from the SRLB. If global, a BSID MUST be allocated from the SRGB.

Use of a BSID allows the instantiation of the policy (the SID list) to be stored only on the node(s) which need to impose the policy. Direction of traffic to a node supporting the policy then only requires imposition of the BSID. If the policy changes, this also means that only the nodes imposing the policy need to be updated. Users of the policy are not impacted.

5.1. IGP Mirroring Context Segment

One use case for a Binding Segment is to provide support for an IGP node to advertise its ability to process traffic originally destined to another IGP node, called the Mirrored node and identified by an IP address or a Node-SID, provided that a "Mirroring Context" segment be inserted in the segment list prior to any service segment local to the mirrored node.

When a given node B wants to provide egress node A protection, it advertises a segment identifying node's A context. Such segment is called "Mirror Context Segment" and identified by the Mirror SID.

The Mirror SID is advertised using the binding segment defined in SR IGP protocol extensions [I-D.ietf-isis-segment-routing-extensions] .

In the event of a failure, a point of local repair (PLR) diverting traffic from A to B does a PUSH of the Mirror SID on the protected traffic. B, when receiving the traffic with the Mirror SID as the active segment, uses that segment and processes underlying segments in the context of A.

6. Multicast

Segment Routing is defined for unicast. The application of the source-route concept to Multicast is not in the scope of this document.

7. IANA Considerations

This document does not require any action from IANA.

8. Security Considerations

Segment Routing is applicable to both MPLS and IPv6 data planes.

Segment Routing adds some meta-data (instructions) to the packet, with the list of forwarding path elements (e.g., nodes, links, services, etc.) that the packet must traverse. It has to be noted that the complete source routed path may be represented by a single segment. This is the case of the Binding SID.

SR by default operates within a trusted domain. Traffic MUST be filtered at the domain boundaries.

The use of best practices to reduce the risk of tampering within the trusted domain is important. Such practices are discussed in [RFC4381] and are applicable to both SR-MPLS and SRv6.

8.1. SR-MPLS

When applied to the MPLS data plane, Segment Routing does not introduce any new behavior or any change in the way MPLS data plane works. Therefore, from a security standpoint, this document does not define any additional mechanism in the MPLS data plane.

SR allows the expression of a source routed path using a single segment (the Binding SID). Compared to RSVP-TE which also provides explicit routing capability, there are no fundamental differences in term of information provided. Both RSVP-TE and Segment Routing may express a source routed path using a single segment.

When a path is expressed using a single label, the syntax of the meta-data is equivalent between RSVP-TE [RFC3209] and SR.

When a source routed path is expressed with a list of segments additional meta-data is added to the packet consisting of the source routed path the packet must follow expressed as a segment list.

When a path is expressed using a label stack, if one has access to the meaning (i.e.: the Forwarding Equivalence Class) of the labels, one has the knowledge of the explicit path. For the MPLS data plane, as no data plane modification is required, there is no fundamental change of capability. Yet, the occurrence of label stacking will increase.

SR domain boundary routers MUST filter any external traffic destined to a label associated with a segment within the trusted domain. This includes labels within the SRGB of the trusted domain, labels within the SRLB of the specific boundary router, and labels outside either of these blocks. External traffic is any traffic received from an interface connected to a node outside the domain of trust.

From a network protection standpoint, there is an assumed trust model such that any node imposing a label stack on a packet is assumed to be allowed to do so. This is a significant change compared to plain IP offering shortest path routing but not fundamentally different compared to existing techniques providing explicit routing capability such as RSVP-TE. By default, the explicit routing information MUST NOT be leaked through the boundaries of the administered domain. Segment Routing extensions that have been defined in various protocols, leverage the security mechanisms of these protocols such as encryption, authentication, filtering, etc.

In the general case, a segment routing capable router accepts and install labels only if these labels have been previously advertised by a trusted source. The received information is validated using existing control plane protocols providing authentication and security mechanisms. Segment Routing does not define any additional security mechanism in existing control plane protocols.

Segment Routing does not introduce signaling between the source and the mid points of a source routed path. With SR, the source routed path is computed using SIDs previously advertised in the IP control

plane. Therefore, in addition to filtering and controlled advertisement of SIDs at the boundaries of the SR domain, filtering in the data plane is also required. Filtering MUST be performed on the forwarding plane at the boundaries of the SR domain and may require looking at multiple labels/instruction.

For the MPLS data plane, there are no new requirements as the existing MPLS architecture already allows such source routing by stacking multiple labels. And for security protection, [RFC4381] and [RFC5920] already call for the filtering of MPLS packets on trust boundaries.

8.2. SRv6

When applied to the IPv6 data plane, Segment Routing does introduce the Segment Routing Header (SRH, [I-D.ietf-6man-segment-routing-header]) which is a type of Routing Extension header as defined in [RFC8200].

The SRH adds some meta-data to the IPv6 packet, with the list of forwarding path elements (e.g., nodes, links, services, etc.) that the packet must traverse and that are represented by IPv6 addresses. A complete source routed path may be encoded in the packet using a single segment (single IPv6 address).

SR domain boundary routers MUST filter any external traffic destined to an address within the SRGB of the trusted domain or the SRLB of the specific boundary router. External traffic is any traffic received from an interface connected to a node outside the domain of trust.

From a network protection standpoint, there is an assumed trust model such that any node adding an SRH to the packet is assumed to be allowed to do so. Therefore, by default, the explicit routing information MUST NOT be leaked through the boundaries of the administered domain. Segment Routing extensions that have been defined in various protocols, leverage the security mechanisms of these protocols such as encryption, authentication, filtering, etc.

In the general case, an SR IPv6 router accepts and install segments identifiers (in the form of IPv6 addresses), only if these SIDs are advertised by a trusted source. The received information is validated using existing control plane protocols providing authentication and security mechanisms. Segment Routing does not define any additional security mechanism in existing control plane protocols.

Problems which may arise when the above behaviors are not implemented or when the assumed trust model is violated (e.g., through a security breach) include:

- Malicious looping
- Evasion of access controls
- Hiding the source of DOS attacks

Security concerns with source routing at the IPv6 data plane are more completely discussed in [RFC5095]. The new IPv6-based segment routing header is defined in [I-D.ietf-6man-segment-routing-header]. This document also discusses the above security concerns.

8.3. Congestion Control

SR does not introduce new requirements for congestion control. By default, traffic delivery is assumed to be best effort. Congestion control may be implemented at endpoints. Where SR policies are in use bandwidth allocation may be managed by monitoring incoming traffic associated with the binding SID identifying the SR policy. Other solutions such as [RFC8084] may be applicable.

9. Manageability Considerations

In SR enabled networks, the path the packet takes is encoded in the header. As the path is not signaled through a protocol, OAM mechanisms are necessary in order for the network operator to validate the effectiveness of a path as well as to check and monitor its liveness and performance. However, it has to be noted that SR allows to reduce substantially the number of states in transit nodes and hence the number of elements that a transit node has to manage is smaller.

SR OAM use cases for the MPLS data plane are defined in [I-D.ietf-spring-oam-usecase]. SR OAM procedures for the MPLS data plane are defined in [RFC8287].

SR routers receive advertisements of SIDs (index, label or IPv6 address) from the different routing protocols being extended for SR. Each of these protocols have monitoring and troubleshooting mechanisms to provide operation and management functions for IP addresses that must be extended in order to include troubleshooting and monitoring functions of the SID.

SR architecture introduces the usage of global segments. Each global segment MUST be bound to a unique index or address within an SR

domain. The management of the allocation of such index or address by the operator is critical for the network behavior to avoid situations like mis-routing. In addition to the allocation policy/tooling that the operator will have in place, an implementation SHOULD protect the network in case of conflict detection by providing a deterministic resolution approach.

When a path is expressed using a label stack, the occurrence of label stacking will increase. A node may want to signal in the control plane its ability in terms of size of the label stack it can support.

A YANG data model [RFC6020] for segment routing configuration and operations has been defined in [I-D.ietf-spring-sr-yang].

When Segment Routing is applied to the IPv6 data plane, segments are identified through IPv6 addresses. The allocation, management and troubleshooting of segment identifiers is no different than the existing mechanisms applied to the allocation and management of IPv6 addresses.

The DA of the packet gives the active segment address. The segment list in the SRH gives the entire path of the packet. The validation of the source routed path is done through inspection of DA and SRH present in the packet header matched to the equivalent routing table entries.

In the context of SR over the IPv6 data plane, the source routed path is encoded in the SRH as described in [I-D.ietf-6man-segment-routing-header]. The SR IPv6 source routed path is instantiated into the SRH as a list of IPv6 address where the active segment is in the Destination Address (DA) field of the IPv6 packet header. Typically, by inspecting in any node the packet header, it is possible to derive the source routed path it belongs to. Similar to the context of SR over MPLS data plane, an implementation may originate path control and monitoring packets where the source routed path is inserted in the SRH and where each segment of the path inserts in the packet the relevant data in order to measure the end to end path and performance.

10. Contributors

The following people have substantially contributed to the definition of the Segment Routing architecture and to the editing of this document:

Ahmed Bashandy
Cisco Systems, Inc.
Email: bashandy@cisco.com

Martin Horneffer
Deutsche Telekom
Email: Martin.Horneffer@telekom.de

Wim Henderickx
Nokia
Email: wim.henderickx@nokia.com

Jeff Tantsura
Email: jefftant@gmail.com

Edward Crabbe
Email: edward.crabbe@gmail.com

Igor Milojevic
Email: milojevicigor@gmail.com

Saku Ytti
TDC
Email: saku@ytti.fi

11. Acknowledgements

We would like to thank Dave Ward, Peter Psenak, Dan Frost, Stewart Bryant, Pierre Francois, Thomas Telkamp, Ruediger Geib, Hannes Gredler, Pushpasis Sarkar, Eric Rosen, Chris Bowers and Alvaro Retana for their comments and review of this document.

12. References

12.1. 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>.

[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>.

[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>.

12.2. Informative References

[I-D.ietf-6man-segment-routing-header]
Previdi, S., Filsfils, C., Raza, K., Dukes, D., Leddy, J., Field, B., daniel.voyer@bell.ca, d., daniel.bernier@bell.ca, d., Matsushima, S., Leung, I., Linkova, J., Aries, E., Kosugi, T., Vyncke, E., Lebrun, D., Steinberg, D., and R. Raszuk, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-08 (work in progress), January 2018.

[I-D.ietf-idr-bgpls-segment-routing-epe]
Previdi, S., Filsfils, C., Patel, K., Ray, S., and J. Dong, "BGP-LS extensions for Segment Routing BGP Egress Peer Engineering", draft-ietf-idr-bgpls-segment-routing-epe-14 (work in progress), December 2017.

[I-D.ietf-isis-segment-routing-extensions]
Previdi, S., Ginsberg, L., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-15 (work in progress), December 2017.

[I-D.ietf-ospf-ospfv3-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPFv3 Extensions for Segment Routing", draft-ietf-ospf-ospfv3-segment-routing-extensions-10 (work in progress), September 2017.

[I-D.ietf-ospf-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-ietf-ospf-segment-routing-extensions-24 (work in progress), December 2017.

[I-D.ietf-pce-segment-routing]
Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "PCEP Extensions for Segment Routing", draft-ietf-pce-segment-routing-11 (work in progress), November 2017.

[I-D.ietf-spring-oam-usecase]
Geib, R., Filsfils, C., Pignataro, C., and N. Kumar, "A Scalable and Topology-Aware MPLS Dataplane Monitoring System", draft-ietf-spring-oam-usecase-10 (work in progress), December 2017.

[I-D.ietf-spring-resiliency-use-cases]
Filsfils, C., Previdi, S., Decraene, B., and R. Shakir, "Resiliency use cases in SPRING networks", draft-ietf-spring-resiliency-use-cases-12 (work in progress), December 2017.

[I-D.ietf-spring-segment-routing-central-epe]
Filsfils, C., Previdi, S., Dawra, G., Aries, E., and D. Afanasiev, "Segment Routing Centralized BGP Egress Peer Engineering", draft-ietf-spring-segment-routing-central-epe-10 (work in progress), December 2017.

[I-D.ietf-spring-segment-routing-mpls]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-11 (work in progress), October 2017.

[I-D.ietf-spring-sr-yang]
Litkowski, S., Qu, Y., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", draft-ietf-spring-sr-yang-08 (work in progress), December 2017.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <https://www.rfc-editor.org/info/rfc3209>.

[RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, DOI 10.17487/RFC4206, October 2005, <https://www.rfc-editor.org/info/rfc4206>.

[RFC4381] Behringer, M., "Analysis of the Security of BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4381, DOI 10.17487/RFC4381, February 2006, <https://www.rfc-editor.org/info/rfc4381>.

[RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <https://www.rfc-editor.org/info/rfc4915>.

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <https://www.rfc-editor.org/info/rfc5095>.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <https://www.rfc-editor.org/info/rfc5120>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/info/rfc5440>.

[RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, DOI 10.17487/RFC5714, January 2010, <https://www.rfc-editor.org/info/rfc5714>.

[RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <https://www.rfc-editor.org/info/rfc5920>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6549] Lindem, A., Roy, A., and S. Mirtorabi, "OSPFv2 Multi-Instance Extensions", RFC 6549, DOI 10.17487/RFC6549, March 2012, <https://www.rfc-editor.org/info/rfc6549>.

[RFC7938] Lapukhov, P., Premji, A., and J. Mitchell, Ed., "Use of BGP for Routing in Large-Scale Data Centers", RFC 7938, DOI 10.17487/RFC7938, August 2016, <https://www.rfc-editor.org/info/rfc7938>.

[RFC8084] Fairhurst, G., "Network Transport Circuit Breakers", BCP 208, RFC 8084, DOI 10.17487/RFC8084, March 2017, <https://www.rfc-editor.org/info/rfc8084>.

[RFC8202] Ginsberg, L., Previdi, S., and W. Henderickx, "IS-IS Multi-Instance", RFC 8202, DOI 10.17487/RFC8202, June 2017, <https://www.rfc-editor.org/info/rfc8202>.

[RFC8287] Kumar, N., Ed., Pignataro, C., Ed., Swallow, G., Akiya, N., Kini, S., and M. Chen, "Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes", RFC 8287, DOI 10.17487/RFC8287, December 2017, <https://www.rfc-editor.org/info/rfc8287>.

Authors' Addresses

Clarence Filsfils (editor)
Cisco Systems, Inc.
Brussels
BE

Email: cfilsfil@cisco.com

Stefano Previdi (editor)
Cisco Systems, Inc.
Italy

Email: stefano@previdi.net

Les Ginsberg
Cisco Systems, Inc

Email: ginsberg@cisco.com

Bruno Decraene
Orange
FR

Email: bruno.decraene@orange.com

Stephane Litkowski
Orange
FR

Email: stephane.litkowski@orange.com

Rob Shakir
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: robjs@google.com

Network Working Group Z. Li
Internet-Draft N. Wu
Intended status: Informational Huawei
Expires: September 14, 2016 March 13, 2016

Tunnel Segment in Segment Routing
draft-li-spring-tunnel-segment-01

Abstract

This document introduces a new type of segment, Tunnel Segment, for the segment routing (SR). Tunnel segment can be used to reduce SID stack depth of SR path, span the non-SR domain or provide differentiated services. Forwarding mechanisms and requirements of control plane and data models for tunnel segments are also defined.

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 http://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 September 14, 2016.

Copyright Notice

Table of Contents

1. Introduction 2
2. Terminology 2
3. Usecases 3
3.1. Reducing SID Stack Depth 3
3.2. Passing through Non-SR Domain 4
3.3. Differentiated Services 5
4. Comparison with Agency Segment 6
5. Forwarding Mechanisms 6
6. Requirement of Control Plane and Yang Models 7
7. IANA Considerations 7
8. Security Considerations 8
9. References 8
9.1. Normative References 8
9.2. Informative References 8
Authors' Addresses 8

1. Introduction

Segment Routing (SR), introduced by [I-D.ietf-spring-segment-routing], leverages the source routing paradigm. A packet can be steered through an ordered list of instructions, which are also called segments. The node segment, adjacency segment, etc. have been proposed for different usecases.

This document introduces a new type of segment, Tunnel Segment, for the segment routing. Tunnel segment can be used to reduce SID stack depth of SR path, span the non-SR domain or provide differentiated services. Forwarding mechanisms and requirements of control plane and data models for tunnel segments are also defined.

2. Terminology

- SID: Segment ID
- SR: Segment Routing
- SR Path: Segment Routing Path
- SR-TE Path: Segment Routing Traffic Engineering Path

- o MSD: Maximally SID Depth

The terms "Tunnel Segment" and "Tunnel SID" are the generic names for a segment attached to a specific tunnel. A tunnel segment can be used to steer traffic into the corresponding tunnel along the SR path.

3. Usecases

3.1. Reducing SID Stack Depth

It is possible that a SR path has to take an explicit path with multiple hops instead of the shortest path for the purpose of traffic engineering. As a result, the ingress node has to push lots of segments to steer the packet, which could be a challenge for the forwarding plane, since the depth of this segment stack may be beyond the capability of their forwarding engines. The tunnel segment introduced in this document will be helpful to mitigate the pain in these scenarios.

Taking Figure 1 below as an example, the SR-TE path is created from SR-Node-1(ingress) to SR-Node-2(egress). The original SID stack, {A, B, X, E, F, G, H, Y, J, K}, is too overwhelming for the path MSD. With help of the tunnel segment, the tunnel from Gateway-Node-1 to Gateway-Node-2 can be represented by a dedicated SID, saying Z. So the SR-TE path can be represented as {A, B, X, Z, J, K}. Comparing with the original SR-TE path, the SID stack depth is reduced.

The SR-TE tunnel can be created through two ways:

1. Manually configure on ingress node (Gateway-Node-1) and designate the SID binding to it. This binding relationship needs to be propagated to PCE/controller or advertised to other nodes in the network.

2. With the knowledge of all MSD along the path, a PCE/controller can calculate SR-TE tunnels using for reduce SID stack depth and determine ingress/egress gateway nodes dynamically. Those SR-TE tunnels can be created through PCE initiated style. The corresponding tunnel segment and the binding relationship can be propagated to ingress nodes and other nodes if necessary. As shown in Figure 1, ingress (SR-Node-1) can receive update messages from PCE/controller about the binding relationship. And SR-Node-1 can calculate the SR-TE path with the SR-TE tunnel segment without the help of PCE/controller in a centralized manner.

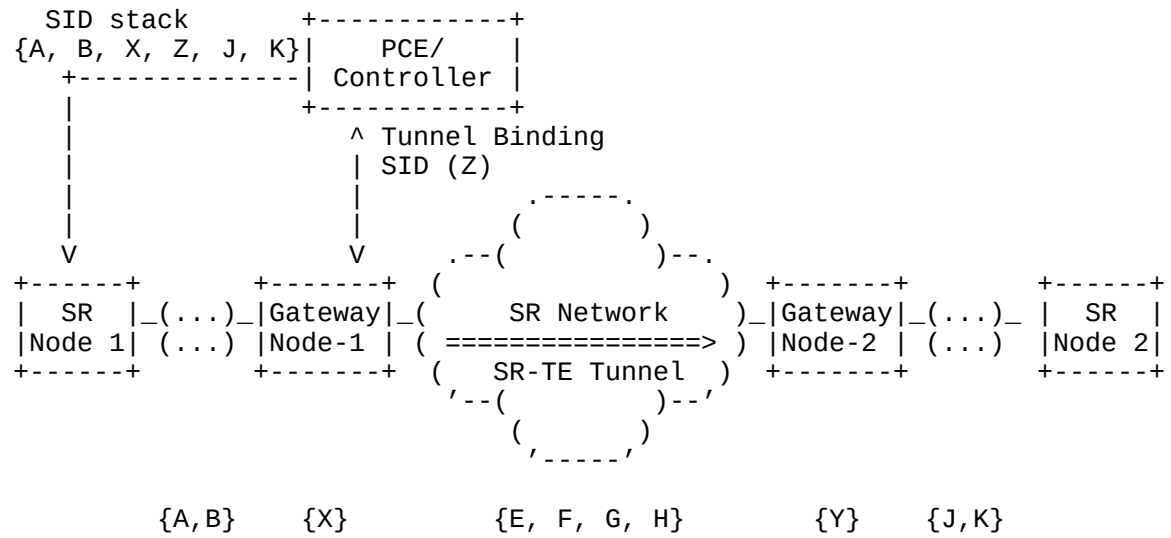


Figure 1 Usecase for Reducing SID Stack Depth

3.2. Passing through Non-SR Domain

The tunnel segment can also be used in those scenarios that traffic has to pass through non-SR domains. In another word, tunnel segment can be used to connect SR islands.

As shown in Figure 2, traffic from SR-Node-1 to SR-Node-2 has to pass through a traditional IP/MPLS network. Usually a RSVP-TE tunnel or IP tunnel will be created between two gateway nodes. By allocating SID for this tunnel, saying Z, the SR path from SR-Node-1 to SR-Node-2 can be represented as {A, B, X, Z, J, K}.

In this scenario, the RSVP-TE tunnel or IP tunnel can be involved into SR networks through two ways:

1. Manually configure on ingress node (Gateway-Node-1) and designate the SID binding to it. This binding relationship needs to be propagated to PCE/controller or advertised to other nodes in the network.

2. With the knowledge of topology of non-SR domain, a PCE/controller can calculate RSVP-TE tunnels or IP tunnels and determine ingress/egress gateway nodes dynamically. Those RSVP-TE tunnels or IP tunnels can be created through PCE initiated style. The corresponding tunnel segment and the binding relationship can be propagated to ingress nodes and other nodes if necessary. As shown in Figure 2, ingress (SR-Node-1) can receive update

messages from PCE/controller about the binding relationship. And SR-Node-1 can calculate the SR-TE path which can pass through non-SR domain without the help of PCE/controller in a centralized manner.

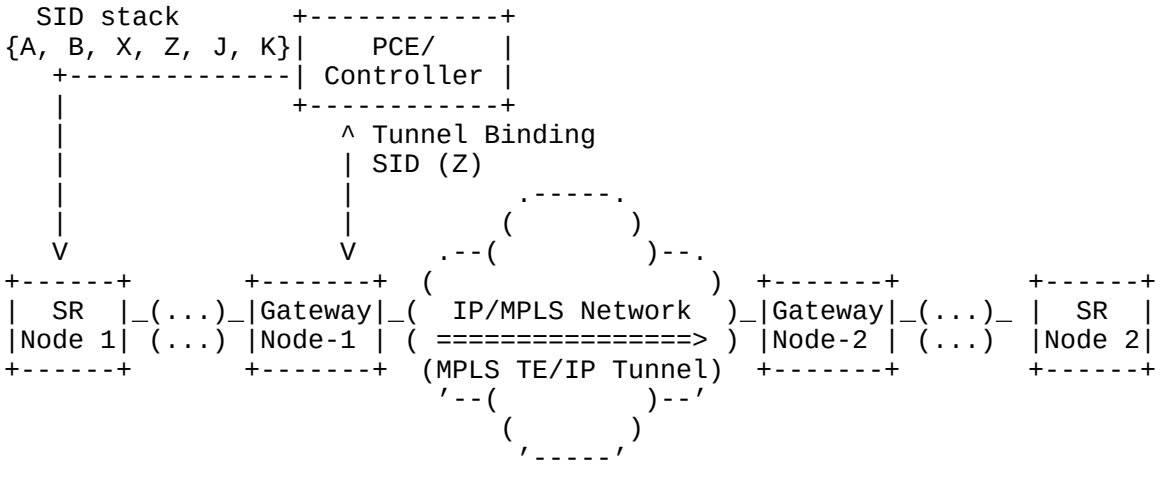


Figure 2 Usecase for Passing through Non-SR Domain

3.3. Differentiated Services

It is necessary to create multiple tunnels between the same pair of gateway nodes to support different services, since different tunnels can have different attributes. As a result, different SIDs have to be assigned per tunnel. Then an End-to-End SR path can choose different SIDs at ingress according to the service requirement when passing through the network between gateway nodes.

As depicted in Figure 3, two RSVP-TE tunnels, say RSVP-TE-tunnel1 and RSVP-TE-tunnel2, are created in MPLS network to provide different bandwidth guarantee services. And two SIDs, Z1 and Z2, are allocated and mapped to these two tunnels separately. These two SIDs can be utilized by a PCE/controller when defining the SR path at ingress. Since different traffic will transport through different tunnels, differentiated services can be guaranteed.

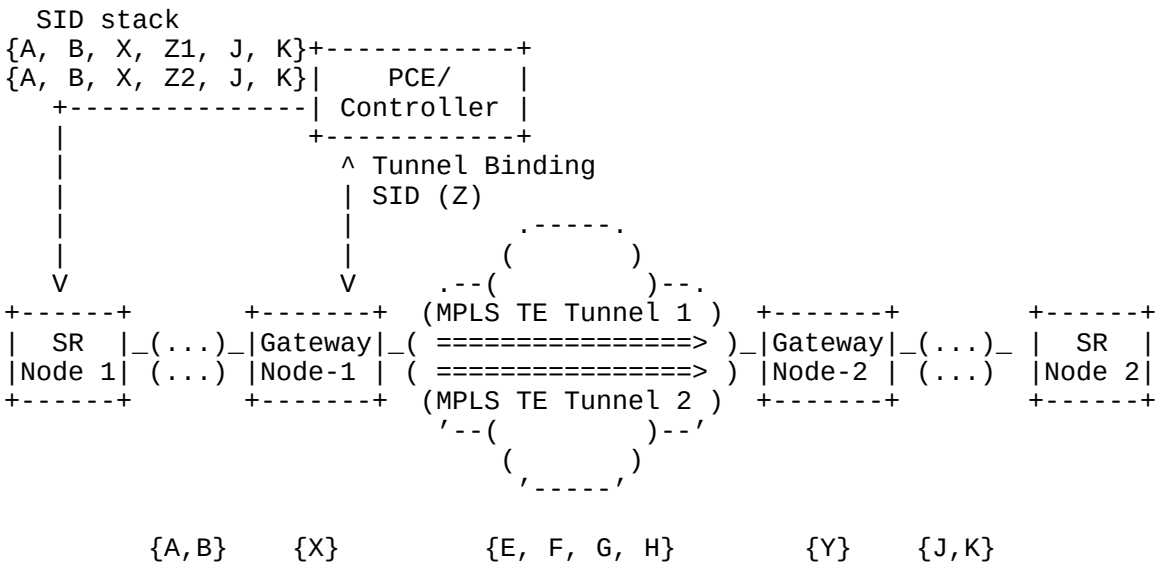


Figure 3 Usecase for Differentiated Services

4. Comparison with Agency Segment

As described in [I-D.ietf-spring-segment-routing], a tunnel can be represented by an Adj-SID or as a Forwarding Adjacency. One obvious benefit of the method is to unify the process. But it may be necessary to differentiate a tunnel segment from other adjacency segment in some scenarios since there are more attributes attached to a tunnel.

By introducing the tunnel segment, this document expects not only to inform the binding relationship between a tunnel and a SID but also to learn tunnel information as much as possible. For example, it will be helpful for SR-capable nodes to know the detail of an explicit path that passes through non-SR networks.

In addition, one tunnel will need an IP address if handled as an adjacency (a borrowed IP address at least). While a tunnel binding to a Tunnel-SID does not have to contain an IP address, only an ingress node and an egress node is enough.

5. Forwarding Mechanisms

In the gateway node, when received the packet with the tunnel segment SID as the topmost SID, it will use the forwarding mechanism shown in the following figure to steering the traffic to the corresponding tunnel.

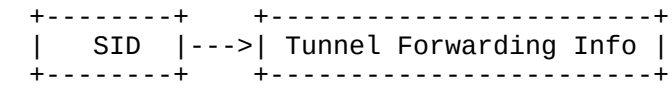


Figure 4 Forwarding Mechanisms for Tunnel Segment

6. Requirement of Control Plane and Yang Models

According to the procedures of the above usecases, following requirements of control plane and Yang models for Tunnel Segment are proposed:

- REQ 01: IGP extensions SHOULD be introduced to advertise the binding relationship between a SID/label and the corresponding tunnel. Attributes of the tunnel MAY be carried optionally.
- REQ 02: BGP Link-State extension SHOULD be introduced to advertise the binding relationship between a label and the corresponding tunnel. Attributes of the tunnel MAY be carried optionally.
- REQ 03: PCEP extensions SHOULD be introduced to advertise the binding relationship between a SID/label and the corresponding tunnel from a PCC to a PCE. Attributes of the tunnel MAY be carried optionally.
- REQ 04: PCE SHOULD support initiated IP tunnel.
- REQ 05: PCE SHOULD support to allocate SID/label for the corresponding tunnel dynamically.
- REQ 06: PCEP extensions SHOULD be introduced to distribute the binding relationship between a SID/label and the corresponding tunnel from a PCE to a PCC. Attributes of the tunnel MAY be carried optionally.
- REQ 07: An I2RS interface SHOULD be available for allocating SID/label to the corresponding tunnel. And augmentation on segment routing YANG models SHOULD be introduced.

7. IANA Considerations

This document makes no request of IANA.

8. Security Considerations

This document does not introduce new security threat.

9. References

9.1. Normative References

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-07 (work in progress), December 2015.

9.2. Informative References

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

Authors' Addresses

Zhenbin Li
Huawei
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: lizhenbin@huawei.com

Nan Wu
Huawei
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: eric.wu@huawei.com