RIFT
Internet-Draft
Intended status: Standards Track
Expires: 27 November 2022

Z. Zhang
J. Tantsura
J. Head
Juniper Networks
D. Fedyk
Individual
26 May 2022

SRIFT: Segment Routing in Fat Trees
draft-zzhang-rift-sr-04

Abstract

This document specifies signaling procedures for Segment Routing in RIFT. Each node's loopback address, Segment Routing Global Block (SRGB) and Node Segment Identifier (Node-SID), which are typically assigned by a configuration management system and distibuted by routing protocols, are distributed southbound from the Top Of Fabric (TOF) nodes via RIFT's Key-Value distribution mechanism, so that each node can compute how to reach a segment represented by the active SID in a packet. An SR controller signals SR policies to ingress nodes so that they can send packets with a desired segment list to steer traffic.

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.

Status of This Memo

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

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

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

This Internet-Draft will expire on 27 November 2022.

Table of Contents

1. Introduction

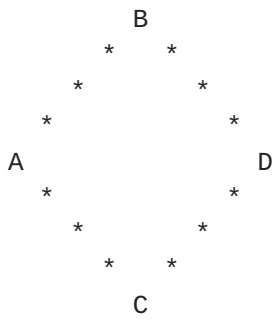
Before we discuss the SR procedures for RIFT, let us first review how SR works with OSPF [RFC8665] and IS-IS [RFC8667].

Each node is provisioned with a loopback address as well as SRGB and Node-SID values. The loopback address and Node-SID are centrally coordinated and are unique per-node within the SR network. These values are then communicated to each node out-of-band and stored as configuration information. Communication could be done via primitive pen and paper or via modern signaling (Netconf/YANG) from a configuration management system.

SRGB information represents the label range of the "global" labels that can be allocated on a particular node for SR. SRGB could have more than one contiguous range of labeks allocated to it. It is comprised of the first available label value and the total number of available labels per range. While in modern networks it is common for each node to have identical SRGB values so that a Node-SID will correspond to the same label on each node, this is not required as to allow for flexible label allocation. In either scenario, SRGB is part of each node's configuration. In today's networks, it is likely pushed to nodes by a configuration management system.

Each node then signals its SRGB and Node-SID to the other nodes. A Node-SID is an index value assigned to a node (say node X), and another node (say node Y) uses the Node-SID to derive (from Y's SRGB) the label to use when sending traffic to node X.

Consider the following example illustrating Node A's computed IP route and label values.



Node Name	Loopback	Node SID	SRGB Label Base	SRGB Label Range
A	10.1.1.1	1	100	50
B	10.1.1.2	2	100	50
C	10.1.1.3	3	200	50
D	10.1.1.4	4	100	50

Destination	Next Hop
10.1.1.1	local
10.1.1.2	if_ab
10.1.1.3	if_ac
10.1.1.4	if_ab, if_ac

Label	Next Hop
100 (La_a)	pop and look up next header
101 (Lb_a)	swap to 101 (Lb_b), via if_ab
102 (Lc_a)	swap to 202 (Lc_c), via if_ac
103 (Ld_a)	swap to 103 (Ld_b), via if_ab
	swap to 203 (Ld_c), via if_ac

The specific notation Lb_a refers to the label derived for node B, using B's Node-SID as index into A's SRGB. Similarly, Ld_c refers to the label derived for Node D, using D's Node-SID as index into C's SRGB.

Node A computes the route to Node D's loopback address. The next hops are Node B (via if_ab) and Node C (via if_ac). Node A uses Node D's Node-SID (which was advertised along with the loopback address) to index into its local SRGB to obtain a label value of 103 (Ld_a). Furthermore, Node A also uses Node D's Node-SID to derive label values for Node B and Node C, 103 (Ld_b) and 203 (Ld_c) respectively, using D's Node-SIDs as index into B and C' SRGBs respectively. Notice that Node C's SRGB is different from the other nodes. Node A can now program its label forwarding state with (Ld_a --> (via if_ab swap to Ld_b, via if_ac swap to Ld_c)).

Similarly, Node B computes the route to Node D's loopback address, but this time finds that the next hop is Node D itself (via if_bd). Node B will also use Node D's Node-SID (again, advertised with the loopback address) to index into its local SRGB and obtain a label value of 103 (Ld_b) and index into Node D's SRGB and obtain a label value of 103 (Ld_d). The label forwarding state can be programmed with (Ld_b --> via if_bd swap to Ld_d). Finally, Node D programs its label forwarding state with (Ld_d -> pop and lookup next header).

2. SR in RIFT (SRIFT)

In referring to the previous section, it is clear that each RIFT node participating in a SR domain requires the following information:

* SRGB values of all adjacent nodes

* Node-SID values of all nodes participating in the routing domain

* Loopback addresses or System IDs of all other nodes

In OSPF and IS-IS, each node's SR information is simply flooded. With RIFT, Node TIEs could be used to flood SR information, but each node would have to learn its own SR information first. With RIFT's Key-Value mechanism, KV-TIEs can be used for TOF nodes to flood all nodes' SR information that it learns from an SR controller, therefore accommodating both provisioning and signalling of SR. The non-TOF nodes do not need any SR related provisioning, which goes very well with RIFT's ZTP concept.

ToF nodes in an SR domain MUST populate KV South TIEs with the minimum required SR information for each node. Specifically SRGB Label Base, SRGB Label Range, Node-SID, RIFT System ID, and Loopback Address. While the Loopback Address must be included, it MAY be set to an empty value in cases if loopbacks are not configured for nodes.

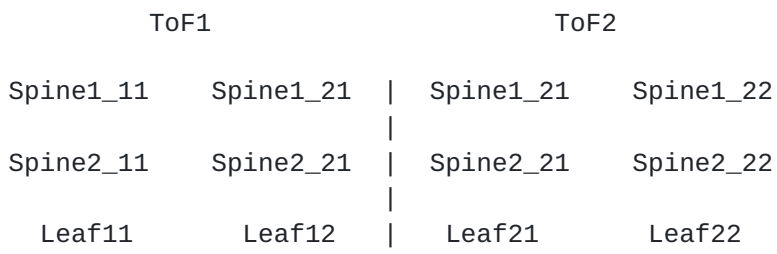
Traffic forwarding in an SR network is typically done in two ways.

The first option is to use Prefix-SIDs and allow traffic to follow the shortest paths for the prefixes. Prefix-SIDs for node prefixes, i.e. Node-SIDs (for loopback addresses), can be used both for encapsulating service traffic to service nodes (e.g. VPN PEs) and for SR-TE traffic steering purposes (see below), but the benefits of other Prefix-SIDs are not clear, so currently only Node-SIDs are supported with RIFT.

The second option is to use SR-TE and follow a specific segment list in the packet header. Each node in the path steers the packet to the currently active segment in the list, following the natural path for that segment (see above). Since a node only has the full topology south of it, and a leaf node does not have any south topology, the traffic steering information (i.e. the segment list) must be programmed by controllers into ingress nodes via SR policies.

Support for Adjacency SIDs will be considered in future revisions.

Consider the following 4-level topology:



Suppose the TE controller instructs Leaf11 to send a packet to Spine2_11 with label stack (Label_TOF2, Label_Spine2_21, Label_Leaf21). Spine2_11 recognizes that Label_TOF2 maps to node TOF2 and it should not simply follow the default route (because the default route could lead to an unintended path via TOF1). In other words, each node needs to have a specific route to every node (that may appear in the segment list). That means for RIFT the southbound distance vector routing needs to additionally advertise routes for the nodes in the north, and they must be propagated all the way down. Each node originates a route for its own loopback address and advertises it southbound, with a special marking that allows a south node to re-advertise it further south.

If loopback addresses are not used, similar "routes" for System IDs must be used. It is RECOMMENDED to use loopback addresses to reuse existing mechanisms.

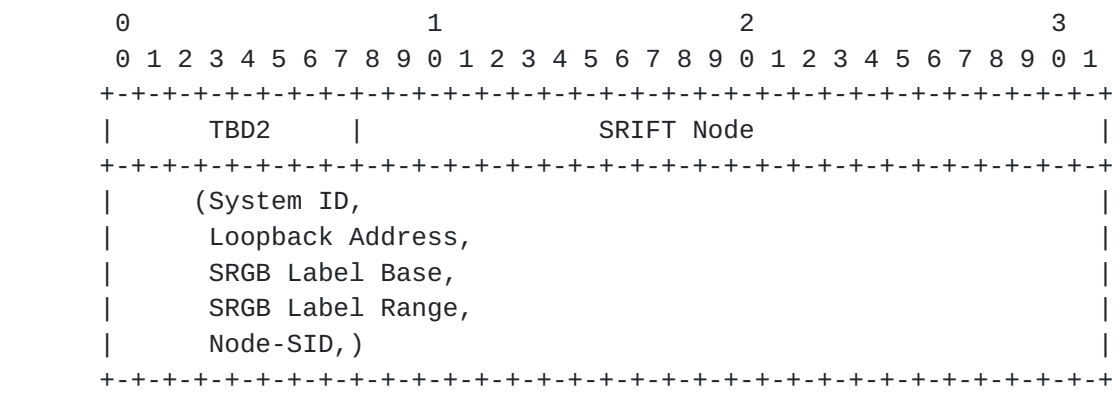
3. Well-Known KV Registry Values

This section requests an entry from the RIFT Well-Known Key-Type Registry for networks that use SR along with suggested values in accordance with RIFT-KV-REGISTRY [RIFT-KV-REGISTRY].

Name	Value	Description
SRIFT Node	TBD	Key-Type describing a SRIFT node

Table 1: Requested Entries

3.1. SRIFT Node Key-Type



where:

System ID:
A node's 64-bit RIFT System ID.

Loopback Address:
A node's loopback address. This MAY be set to 0 if loopback addresses are not used.

SRGB Label Base:
The first valid label within the corresponding node's SRGB.

SRGB Label Range:
The total number of valid labels in the corresponding node's SRGB.

Node-SID:
The corresponding node's Node-SID value.

4. Security Considerations

This document does not introduce any new security concerns with RIFT or any other referenced protocols. RIFT KV TIEs are already extensively secured via RIFT's specification.

5. Acknowledgements

The authors thank Bruno Rijsman and Antoni Przygenda for their review and suggestions.

6. References

6.1. Normative References

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

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

[RIFT] Przygienda, T., Sharma, A., Thubert, P., Rijsman, B., and D. Afanasiev, "RIFT: Routing in Fat Trees", Work in Progress, draft-ietf-rift-rift-12, May 2020.

[RIFT-KV-REGISTRY] Przygienda, T., "RIFT Keys Structure and Well-Known Registry in Key Value TIE", Work in Progress, draft-przygienda-rift-kv-registry-00, December 2020.

6.2. Informative References

[RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <https://www.rfc-editor.org/info/rfc8665>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <https://www.rfc-editor.org/info/rfc8667>.

Authors' Addresses

Zhaohui Zhang
Juniper Networks

Email: zzhang@juniper.net

Jeff Tantsura
Juniper Networks

Email: jefftant.ietf@gmail.com

Jordan Head
Juniper Networks

Email: jhead@juniper.net

Don Fedyk
Individual

Email: don.fedyk@gmail.com