

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 24, 2014

C. Filsfils, Ed.  
S. Previdi, Ed.  
A. Bashandy  
Cisco Systems, Inc.  
B. Decraene  
S. Litkowski  
Orange  
M. Horneffer  
Deutsche Telekom  
I. Milojevic  
Telekom Srbija  
R. Shakir  
British Telecom  
S. Ytti  
TDC Oy  
W. Henderickx  
Alcatel-Lucent  
J. Tantsura  
Ericsson  
E. Crabbe  
Google, Inc.  
October 21, 2013

Segment Routing Architecture  
draft-filsfils-rtgwg-segment-routing-01

Abstract

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

The Segment Routing architecture can be directly applied to the MPLS dataplane with no change on the forwarding plane. IGP-based segments require minor extension to the existing link-state routing protocols. Segment Routing can also be applied to IPv6 with a new type of routing extension 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

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 April 24, 2014.

#### Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1.	Introduction . . . . .	4
1.1.	Illustration . . . . .	4
1.2.	Terminology . . . . .	7
1.3.	Properties . . . . .	8
1.4.	Companion Documents . . . . .	9
1.5.	Relationship with MPLS and IPv6 . . . . .	9
2.	Abstract Routing Model . . . . .	10
2.1.	Traffic Engineering with SR . . . . .	12
2.2.	Segment Routing Database . . . . .	13
3.	Link-State IGP Segments . . . . .	13
3.1.	Illustration . . . . .	14
3.1.1.	Example 1 . . . . .	15
3.1.2.	Example 2 . . . . .	15
3.1.3.	Example 3 . . . . .	15
3.1.4.	Example 4 . . . . .	15
3.1.5.	Example 5 . . . . .	16
3.2.	IGP Segment Terminology . . . . .	16
3.2.1.	IGP Segment, IGP SID . . . . .	16
3.2.2.	IGP-Prefix Segment, Prefix-SID . . . . .	17
3.2.3.	IGP-Node Segment, Node-SID . . . . .	17
3.2.4.	IGP-Anycast Segment, Anycast SID . . . . .	18
3.2.5.	IGP-Adjacency Segment, Adj-SID . . . . .	18
3.2.6.	Finally . . . . .	19
3.3.	IGP Segment Allocation, Advertisement and SRDB Maintenance . . . . .	19
3.3.1.	Prefix-SID . . . . .	19
3.3.2.	Adj-SID . . . . .	20
3.4.	Inter-Area Considerations . . . . .	22
3.5.	IGP Mirroring Context Segment . . . . .	23
4.	Service Segments . . . . .	23
5.	OAM . . . . .	23
6.	Multicast . . . . .	24
7.	IANA Considerations . . . . .	24
8.	Manageability Considerations . . . . .	24
9.	Security Considerations . . . . .	24
10.	Acknowledgements . . . . .	24
11.	References . . . . .	25
11.1.	Normative References . . . . .	25
11.2.	Informative References . . . . .	25
	Authors' Addresses . . . . .	26

## 1. Introduction

In this section, we illustrate the key properties of the SR architecture, introduce the companion documents to this note and relate SR to the MPLS and IPv6 architectures.

Section 2 defines the SR abstract routing model. Section 3 defines the IGP-based segments. Section 4 defines the Service Segments.

### 1.1. Illustration

In the context of Figure 1 where all the links have the same IGP cost, let us assume that a packet P enters the SR domain at an ingress edge router I and that the operator requests the following requirements for packet P:

The local service S offered by node B must be applied to packet P.

The links AB and CE cannot be used to transport the packet P.

Any node N along the journey of the packet should be able to determine where the packet P entered the SR domain and where it will exit. The intermediate node should be able to determine the paths from the ingress edge router to itself, and from itself to the egress edge router.

Per-flow State for packet P should only be created at the ingress edge router.

State for packet P can only be created at the ingress edge router.

The operator can forbid, for security reasons, anyone outside the operator domain to exploit its intra-domain SR capabilities.

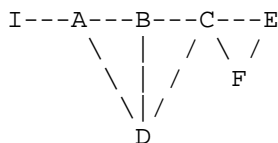


Figure 1: An illustration of SR properties

All these properties may be realized by instructing the ingress SR edge router I to push the following abstract SR header on the packet P.

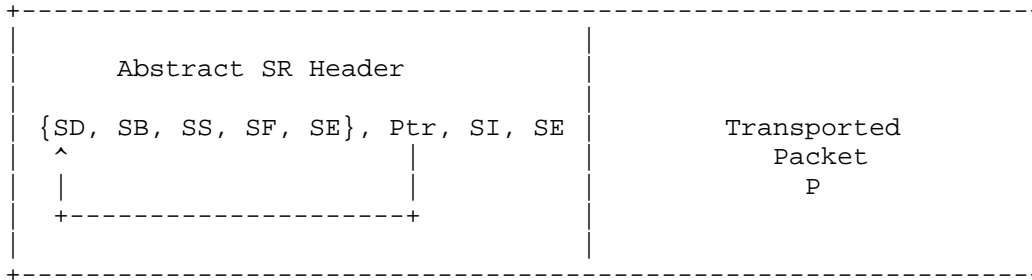


Figure 2: Packet P at node I

The abstract SR header contains a source route encoded as a list of segments {SD, SB, SS, SF, SE}, a pointer (Ptr) and the identification of the ingress and egress SR edge routers (segments SI and SE).

A segment is a 32-bit identification either for a topological instruction or a service instruction. A segment can either be global or local. The instruction associated with a global segment is recognized and executed by any SR-capable node in the domain. The instruction associated with a local segment is only supported by the specific node that originates it.

Let us assume some ISIS/OSPF extensions to define a "Node Segment" as a global instruction within the IGP domain to forward a packet along the shortest path to the specified node. Let us further assume that within the SR domain illustrated in Figure 1, segments SI, SD, SB, SE and SF respectively identify IGP node segments to I, D, B, E and F.

Let us assume that node B identifies its local service S with local segment SS.

With all of this in mind, let us describe the journey of the packet P.

The packet P reaches the ingress SR edge router. I pushes the SR header illustrated in Figure 2 and sets the pointer to the first segment of the list (SD).

SD is an instruction recognized by all the nodes in the SR domain which causes the packet to be forwarded along the shortest path to D.

Once at D, the pointer is incremented and the next segment is executed (SB).

SB is an instruction recognized by all the nodes in the SR domain which causes the packet to be forwarded along the shortest path to B.

Once at B, the pointer is incremented and the next segment is executed (SS).

SS is an instruction only recognized by node B which causes the packet to receive service S.

Once the service applied, the next segment is executed (SF) which causes the packet to be forwarded along the shortest path to F.

Once at F, the pointer is incremented and the next segment is executed (SE).

SE is an instruction recognized by all the nodes in the SR domain which causes the packet to be forwarded along the shortest path to E.

E then removes the SR header and the packet continues its journey outside the SR domain.

All of the requirements are met.

First, the packet P has not used links AB and CE: the shortest-path from I to D is I-A-D, the shortest-path from D to B is D-B, the shortest-path from B to F is B-C-F and the shortest-path from F to E is F-E, hence the packet path through the SR domain is I-A-D-B-C-F-E and the links AB and CE have been avoided.

Second, the service S supported by B has been applied on packet P.

Third, any node along the packet path is able to identify the service and topological journey of the packet within the SR domain. For example, node C receives the packet illustrated in Figure 3 and hence is able to infer where the packet entered the SR domain (SI), how it got up to itself {SD, SB, SS, SE}, where it will exit the SR domain (SE) and how it will do so {SF, SE}.

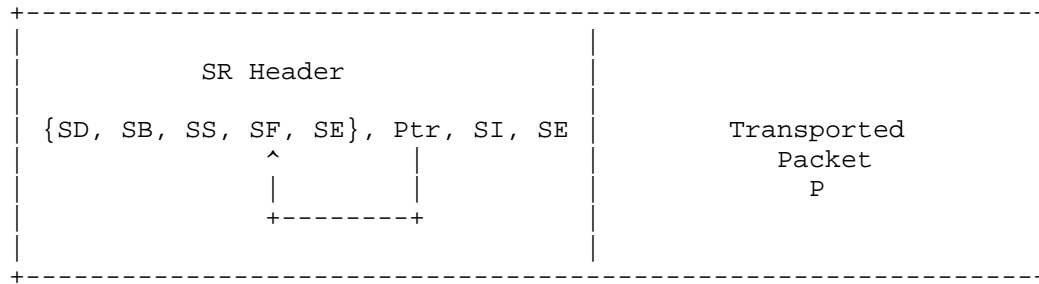


Figure 3: Packet P at node C

Fourth, only node I maintains per-flow state for packet P. The entire

program of topological and service instructions to be executed by the SR domain on packet P is encoded by the ingress edge router I in the SR header in the form of a list of segments where each segment identifies a specific instruction. No further per-flow state is required along the packet path. The per-flow state is in the SR header and travels with the packet. Intermediate nodes only hold states related to the IGP global node segments and the local IGP adjacency segments. These segments are not per-flow specific and hence scale very well. Typically, an intermediate node would maintain in the order of 100's to 1000's global node segments and in the order of 10's to 100 of local adjacency segments. Typically the SR IGP forwarding table is expected to be much less than 10000 entries.

Fifth, the SR header is inserted at the entrance to the domain and removed at the exit of the operator domain. For security reasons, the operator can forbid anyone outside its domain to use its intra-domain SR capability.

## 1.2. Terminology

The following terminology is defined:

Term	Definition
Segment	A segment that identifies an instruction
SID	A 32-bit identification for a segment
Segment List	Ordered list of segments encoding the topological and service source route of the packet
Active Segment	The segment that MUST be used by the receiving router to process the packet. It is identified by the pointer
SR-Pointer or pointer	In the SR header, it indicates the active segment in the segment list
Global Segment	The related instruction is supported by all the SR-capable nodes in the local domain
SRGB	SR Global Block: the set of global segments in the local SR domain
Local Segment	The related instruction is supported only by the node originating it

IGP Segment or IGP SID	The generic names 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 or Prefix-SID	An IGP-Prefix Segment is an IGP segment attached to an IGP prefix. An IGP-Prefix Segment is always global within the SR/IGP domain and identifies the ECMP-aware shortest-path computed by the IGP to the related prefix. The Prefix-SID is the SID of the IGP-Prefix Segment
IGP-Node Segment or Node Segment or Node-SID	An IGP-Node Segment is a an IGP-Prefix Segment which identifies a specific router (e.g. a loopback). The terms "Node Segment" or "Node-SID" are often used as an abbreviation
IGP-Anycast Segment or Anycast Segment or Anycast-SID	An IGP-Anycast Segment is an IGP-prefix segment which does not identify a specific router, but a set of routers. The terms "Anycast Segment" or "Anycast-SID" are often used as an abbreviation
IGP-Adjacency Segment or Adjacency Segment or Adj-SID	An IGP-Adjacency Segment is an IGP segment attached to an unidirectional adjacency or a set of unidirectional adjacencies. An IGP-Adjacency Segment is local to the node which advertises it
SRDB	The SR Database. Each entry is indexed by a segment value. Each entry must list the SR header operation to apply and the next-hop to forward the packet to
SR Header Operation	Push, Continue and Next are operations applied on the SR segment list

Table 1: Segment Routing Terminology

### 1.3. Properties

Assuming a packet flow F entering an SR domain at ingress SR edge router I, the properties offered by the SR architecture are:

Per-Flow state for F is only maintained by node I.



Any topological path through the SR domain can be enforced.

Any chain of services through the SR domain can be enforced.

Any mix of topological paths and chain of services can be enforced.

Any node along the flow path can determine where flow entered the SR domain, how it got up to that node, where it will exit the SR domain and how it will get there.

#### 1.4. Companion Documents

This document defines the SR architecture, its routing model, the IGP-based segments and the service segments.

Use cases are described in [I-D.filsfils-rtgwg-segment-routing-use-cases].

The support of SR by the MPLS dataplane is documented in [draft-filsfils-spring-segment-routing-mpls-00].

The support of SR on the Ipv6 dataplane will be documented in a future document.

IS-IS protocol extensions for Segment Routing are described in [I-D.previdi-isis-segment-routing-extensions].

OSPF protocol extensions for Segment Routing are described in [I-D.psenak-ospf-segment-routing-extensions] and [I-D.psenak-ospf-segment-routing-ospfv3-extension].

The FRR solution for SR is documented in [I-D.francois-sr-frr].

The PCEP protocol extensions for Segment Routing are defined in [I-D.sivabalan-pce-segment-routing].

The interaction between SR/MPLS with other MPLS Signaling planes is documented in [draft-filsfils-spring-segment-routing-ldp-interop-00].

#### 1.5. Relationship with MPLS and IPv6

The source routing model is inherited from the one proposed by and [RFC1940] and [RFC2460].

The notion of abstract segment identifier which can represent any instruction is inherited from MPLS ([RFC3031]).

Deployment experiences has shown the need to limit the number of per-flow states maintained in the network while preserving information on the topological and service journey of a packet (e.g. the ingress to the domain for accounting/billing purpose).

The main differences from the IPv6 source route model are:

The source route is encoded as an ordered list of segments instead of IP addresses.

A segment can represent any instruction either a service or a topological path. Topologically, the path to an IP address is often limited to the shortest-path to that address. A segment can represent any path (e.g. an adjacency segment forces a packet to a nexthop through a specific adjacency even if the shortest-path to the next-hop does not use that adjacency).

The ingress and egress edge routers are identified and always available, allowing for interesting accounting and policy applications.

The source route functionality cannot be controlled from outside the SR domain.

The main differences from the current MPLS model are:

Globally indexed segments are introduced (e.g. IGP Prefix segments).

LDP and RSVP MPLS signaling protocols are not required. If present, SR can coexist and interwork with LDP and RSVP. [draft-filsfils-spring-segment-routing-ldp-interop-00].

Per-flow states are only maintained at the ingress edge router.

SR can be instantiated on the IPv6 dataplane. A future document will detail the new routing extension header which carry all the elements of the abstract SR header. All the SR properties are preserved.

SR can be instantiated on the MPLS dataplane as detailed in [draft-filsfils-spring-segment-routing-mpls-00].

## 2. Abstract Routing Model

Segment Routing (SR) leverages the source routing paradigm.

At the entrance of the SR domain, the ingress SR edge router pushes

the SR header on top of the packet. At the exit of the SR domain, the egress SR edge router removes the SR header.

The SR header contains an ordered list of segments, a pointer identifying the next segment to process and the identifications of the ingress and egress SR edge routers on the path of this packet. The pointer identifies the segment that **MUST** be used by the receiving router to process the packet. This segment is called the active segment.

A property of the architecture is that the entire source route of the packet, including the identity of the ingress and egress edge routers is always available with the packet. This allows for interesting accounting and service applications.

We define three SR-header operations:

"PUSH": an SR header is pushed on an IP packet, or additional segments are added at the head of the segment list. The pointer is moved to the first entry of the added segments.

"NEXT": the active segment is completed, the pointer is moved to the next segment in the list.

"CONTINUE": the active segment is not completed, the pointer is left unchanged.

In the future, other SR-header management operations may be defined.

As the packet travels through the SR domain, the pointer is incremented through the ordered list of segments and the source route encoded by the SR ingress edge node is executed.

A node processes an incoming packet according to the instruction associated with the active segment.

Any instruction might be associated with a segment: for example, an intra or inter-domain topological strict or loose forwarding instruction, a service instruction, etc.

At minimum, a segment instruction must define two elements: the identity of the next-hop to forward the packet to (this could be the same node or a context within the node) and which SR-header management operation to execute.

Each segment is known in the network through a Segment Identifier (SID), a value allocated from the 32-bit Segment Identifier space. The first 16 values are reserved. The terms "segment" and "SID" are

interchangeable.

Within an SR domain, all the SR-capable nodes are configured with the Segment Routing Global Block (SRGB). The SRGB is a subset of the 32-bit SID space. SRGB can be a non-contiguous set of segments.

All global segments must be allocated from the SRGB. Any SR capable node MUST be able to process any global segment advertised by any other node within the SR domain.

Any segment outside the SRGB has a local significance and is called a "local segment". An SR-capable node MUST be able to process the local segments it originates. An SR-capable node MUST NOT support the instruction associated with a local segment originated by a remote node.

## 2.1. Traffic Engineering with SR

An SR Traffic Engineering policy is composed of two elements: a flow classification and a segment-list to prepend on the packets of the flow.

In the SR architecture, this per-flow state only exists at the ingress edge router whether the policy is defined and the SR header is pushed.

It is outside the scope of the document to define the process that leads to the instantiation at a node N of an SR Traffic Engineering policy.

[I-D.filsfils-rtgwg-segment-routing-use-cases] illustrates various alternatives:

- N is deriving this policy automatically (e.g. FRR).

- N is provisioned explicitly by the operator.

- N is provisioned by a stateful PCE server.

- N is provisioned by the operator with a high-level policy which is mapped into a path thanks to a local CSPF-based computation (e.g. affinity/SRLG exclusion).

Any architecture that involves the insertion of information onto a packet involves performance consideration.

[I-D.filsfils-rtgwg-segment-routing-use-cases] explains why the majority of use-cases require very short segment-lists.

A stateful PCE server, which desires to instantiate at node N an SR Traffic Engineering policy, collects the SR capability of node N such as to ensure that the policy meets its capability [I-D.sivabalan-pce-segment-routing].

## 2.2. Segment Routing Database

The Segment routing Database (SRDB) is a set of entries where each entry is identified by a segment value. The instruction associated with each entry at least defines the identity of the next-hop to which the packet should be forwarded and what operation should be performed on the SR header (PUSH, CONTINUE, NEXT).

Segment	Next-Hop	SR Header operation
Sk	M	CONTINUE
Sj	N	NEXT
Sl	NAT Srvc	NEXT
Sm	FW srvc	NEXT
Sn	Q	NEXT
etc.	etc.	etc.

Figure 4: SR Database

Each SR-capable node maintains its local SRDB. SRDB entries can either derive from local policy or or from protocol segment advertisement. The next section will detail segment advertisement by IGP protocols."

## 3. Link-State IGP Segments

Within a link-state IGP 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 the Segment Routing architecture and use-cases [I-D.filsfils-rtgwg-segment-routing-use-cases] as they enable the expression of any topological path throughout the IGP domain. Such a topological path is either expressed as a single IGP segment or a list of multiple IGP segments.

In the first sub-section, we introduce a terminology for a set of IGP segments which are very frequently seen in the SR use-cases. The second sub-section details the IGP segment allocation and SRDB construction rules.

## 3.1. Illustration

Assuming the network diagram of Figure 5 and the IP address and IGP Segment allocation of Figure 6, the following examples can be constructed.

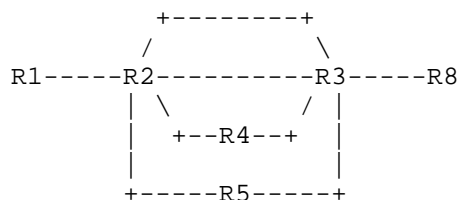


Figure 5: IGP Segments - Illustration

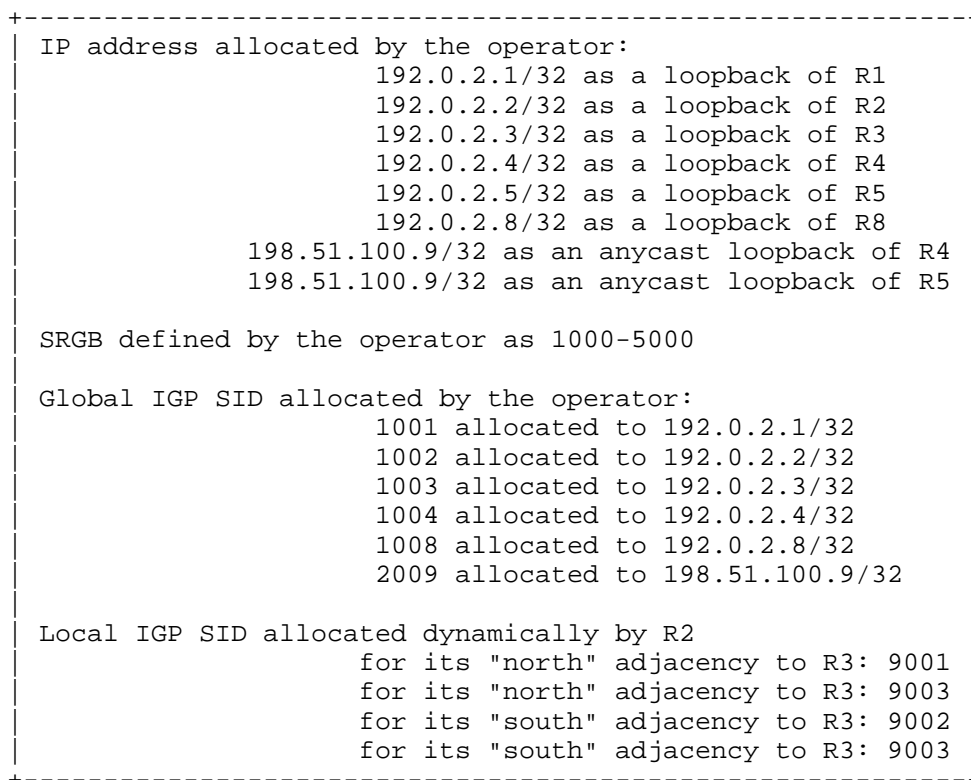


Figure 6: IGP Address and Segment Allocation - Illustration

### 3.1.1. Example 1

R1 may send a packet P1 to R8 simply by pushing an SR header with segment list {1008}.

1008 is a global IGP segment attached to the IP prefix 192.0.2.8/32. Its semantic is global within the IGP domain: any router forwards a packet received with active segment 1008 to the next-hop along the ECMP-aware shortest-path to the related prefix.

In conclusion, the path followed by P1 is R1-R2--R3-R8. The ECMP-awareness ensures that the traffic be load-shared between any ECMP path, in this case the two north and south links between R2 and R3.

### 3.1.2. Example 2

R1 may send a packet P2 to R8 by pushing an SR header with segment list {1002, 9001, 1008}.

1002 is a global IGP segment attached to the IP prefix 192.0.2.2/32. Its semantic is global within the IGP domain: any router forwards a packet received with active segment 1002 to the next-hop along the shortest-path to the related prefix.

9001 is a local IGP segment attached by node R2 to its north link to R3. Its semantic is local to node R2: R2 switches a packet received with active segment 9001 towards the north link to R3.

In conclusion, the path followed by P2 is R1-R2-north-link-R3-R8.

### 3.1.3. Example 3

R1 may send a packet P3 along the same exact path as P1 using a different segment list {1002, 9003, 1008}.

9003 is a local IGP segment attached by node R2 to both its north and south links to R3. Its semantic is local to node R2: R2 switches a packet received with active segment 9003 towards either the north or south links to R3 (e.g. per-flow loadbalancing decision).

In conclusion, the path followed by P3 is R1-R2-any-link-R3-R8.

### 3.1.4. Example 4

R1 may send a packet P4 to R8 while avoiding the links between R2 and R3 by pushing an SR header with segment list {1004, 1008}.

1004 is a global IGP segment attached to the IP prefix 192.0.2.4/32.

Its semantic is global within the IGP domain: any router forwards a packet received with active segment 1004 to the next-hop along the shortest-path to the related prefix.

In conclusion, the path followed by P4 is R1-R2-R4-R3-R8.

### 3.1.5. Example 5

R1 may send a packet P5 to R8 while avoiding the links between R2 and R3 while still benefitting from all the remaining shortest paths (via R4 and R5) by pushing an SR header with segment list {2009, 1008}.

2009 is a global IGP segment attached to the anycast IP prefix 198.51.100.9/32. Its semantic is global within the IGP domain: any router forwards a packet received with active segment 2009 to the next-hop along the shortest-path to the related prefix.

In conclusion, the path followed by P5 is either R1-R2-R4-R3-R8 or R1-R2-R5-R3-R8 .

## 3.2. IGP Segment Terminology

### 3.2.1. IGP Segment, IGP SID

The terms "IGP Segment" and "IGP SID" are the generic names for a segment attached to a piece of information advertised by a link-state IGP, e.g. an IGP prefix or an IGP adjacency.

The IGP signaling extension to advertise an IGP segment includes the G-Flag indicating whether the IGP segment is global or local.

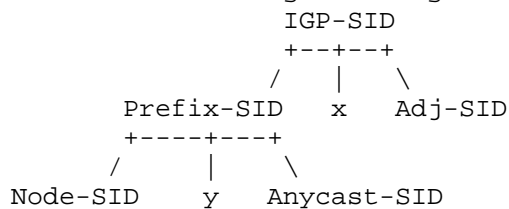


Figure 7: IGP SID Terminology

The IGP Segment terminology is introduced to ease the documentation of SR use-cases and hence does not propose a name for any possible variation of IGP segment supported by the architecture. For example, y in Figure 7 could represent a local IGP segment attached to an IGP Prefix. This variation, while supported by the SR architecture is not seen in the SR use-cases and hence does not receive a specific name.



In Figure 5 and Figure 6, SIDs 1001, 1002, 1003, 1004, 1008, 2009, 9001, 9002 and 9003 are called IGP SIDs.

### 3.2.2. IGP-Prefix Segment, Prefix-SID

An IGP-Prefix Segment is an IGP segment attached to an IGP prefix. An IGP-Prefix Segment is always global within the SR/IGP domain and identifies the ECMP-aware shortest-path computed by the IGP to the related prefix. The G-Flag MUST be set. The Prefix-SID is the SID of the IGP-Prefix Segment.

A packet injected anywhere within the SR/IGP domain with an active Prefix-SID will be forwarded along the shortest-path to that prefix.

The IGP signaling extension for IGP-Prefix segment includes the P-Flag. A Node N advertising a Prefix-SID SID-R for its attached prefix R resets the P-Flag to allow its connected neighbors to perform the NEXT operation while processing SID-R. This behavior is equivalent to Pen-ultimate Hop Popping in MPLS. When set, the neighbors of N must perform the CONTINUE operation while processing SID-R.

While the architecture allows to attach a local segment to an IGP prefix, we specifically assume that when the terms "IGP-Prefix Segment" and "Prefix-SID" are used then the segment is global (the SID is allocated from the SRGB). This is consistent with [I-D.filsfils-rtgwg-segment-routing-use-cases] as all the described use-cases require global segments attached to IGP prefix.

In Figure 5 and Figure 6, SIDs 1001, 1002, 1003, 1004, 1008, 2009 are called Prefix-SIDs.

### 3.2.3. IGP-Node Segment, Node-SID

An IGP-Node Segment is a an IGP-Prefix Segment which identifies a specific router (e.g. a loopback). The terms "Node Segment" or "Node-SID" are often used as an abbreviation.

A "Node Segment" or "Node-SID" is fundamental to the SR architecture. From anywhere in the network, it enforces the ECMP-aware shortest-path forwarding of the packet towards the related node as explained in [I-D.filsfils-rtgwg-segment-routing-use-cases].

In Figure 5 and Figure 6, SIDs 1001, 1002, 1003, 1004 and 1008 are called Node-SIDs.

#### 3.2.4. IGP-Anycast Segment, Anycast SID

An IGP-Anycast Segment is an IGP-prefix segment which does not identify a specific router, but a set of routers. The terms "Anycast Segment" or "Anycast-SID" are often used as an abbreviation.

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 as described in [I-D.filsfils-rtgwg-segment-routing-use-cases].

In Figure 5 and Figure 6, SID 2009 is called Anycast SID.

#### 3.2.5. IGP-Adjacency Segment, Adj-SID

An IGP-Adjacency Segment is an IGP segment attached to an unidirectional adjacency or a set of unidirectional adjacencies. An IGP-Adjacency Segment is local to the node which advertises it. The SID of the IGP-Adjacency Segment is called the Adj-SID. The G-Flag must be reset.

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 node (i.e.: an IGP neighbor).

- A non-adjacent neighbor (e.g.: a Forwarding Adjacency, [RFC4206]).

- A virtual neighbor outside the IGP domain (e.g.: an interface connecting another AS) as defined in [RFC5316].

A packet injected anywhere within the SR/IGP 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 along the various members of the set.

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 as explained in [I-D.filsfils-rtgwg-segment-routing-use-cases].

In Figure 5 and Figure 6, SIDs 9001, 9002 and 9003 are called Adj-

SIDs.

### 3.2.6. Finally

Figure 8 summarizes the different terms that can be used to refer to the SID's used in the example illustrated by Figure 5 and Figure 6. "Y" means that the term can be used to refer to the SID, "N" means that the term cannot be used to refer to the SID.

SID Value	IGP SID	Prefix-SID	Node-SID	Anycast SID	Adj-SID
1001	Y	Y	Y	N	N
1002	Y	Y	Y	N	N
1003	Y	Y	Y	N	N
1004	Y	Y	Y	N	N
1005	Y	Y	Y	N	N
1008	Y	Y	Y	N	N
2009	Y	Y	N	Y	N
9001	Y	N	N	N	Y
9002	Y	N	N	N	Y
9003	Y	N	N	N	Y

Figure 8: Terminology Example

## 3.3. IGP Segment Allocation, Advertisement and SRDB Maintenance

### 3.3.1. Prefix-SID

Multiple Prefix-SID's may be allocated to the same IGP Prefix (e.g. for class of service purpose). Typically a single Prefix-SID is allocated to an IGP Prefix.

A Prefix-SID is allocated from the SRGB according to a similar process to IP address allocation. Typically the Prefix-SID is allocated by policy by the operator (or NMS) and the SID very rarely changes.

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 warning for misconfiguration.

The required IGP protocol extensions are defined in [I-D.previdi-isis-segment-routing-extensions],

[I-D.psenak-ospf-segment-routing-extensions] and  
[I-D.psenak-ospf-segment-routing-ospfv3-extension].

A node N attaching a Prefix-SID SID-R to its attached prefix R MUST maintain the following SRDB entry:

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

A remote node M MUST maintain the following SRDB 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 shortest-path to prefix R.

### 3.3.2. Adj-SID

The Adjacency Segment SID (Adj-SID) identifies a unidirectional adjacency or a set of unidirectional adjacencies.

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

A node MAY allocate multiple Adj-SIDs to the same adjacency.

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

Adjacency suppression MUST NOT be performed by the IGP.

A node MUST install an SRDB entry for any Adj-SID of value V attached to data-link L:

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

When associated to a Forwarding Adjacency ([RFC4206]), the Adj-SID MAY also include the necessary information in order to describe the path to the remote end of the Forwarding Adjacency in the form of an Explicit Route Object.

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

### 3.3.2.1. Parallel Adjacencies

Adj-SIDs can be used in order to represent a set of parallel interfaces between two adjacent routers. For example, SID 9003 in figures 5 and 6 identify the set of interfaces between R2 and R3.

A node MUST install an SRDB entry for any locally originated Adjacency Segment (Adj-SID) of value W attached to a set of link B with:

Incoming Active Segment: W

Ingress Operation: NEXT

Egress interface: loadbalance between any data-link within set B

### 3.3.2.2. 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 other 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. These extensions are defined in [I-D.previdi-isis-segment-routing-extensions], [I-D.psenak-ospf-segment-routing-extensions] and [I-D.psenak-ospf-segment-routing-ospfv3-extension].

### 3.3.2.3. External Adjacencies Considerations

IGPs have been extended in order to advertise virtual adjacencies that represent external links ([RFC5316]).

Segment Routing allows to allocate an Adj-SID to these external links.

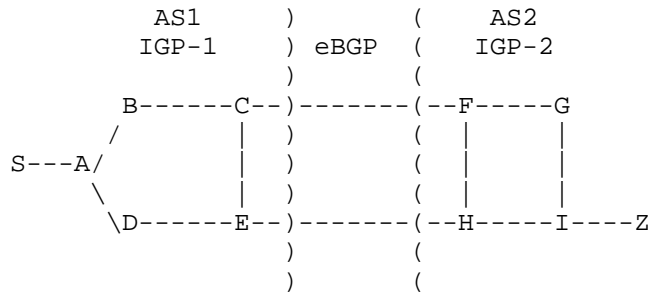


Figure 9: External Adjacency Example

In the diagram above, C advertises in the IGP an adjacency to peer F of AS2 together with an associated Adj-SID. When S wants to force an inter-domain path to Z via the peering link CF, S encapsulates the packets with the list {Prefix-SID(C), Adj-SID(C,F, AS2)}.

[I-D.filsfils-rtgwg-segment-routing-use-cases] provides an external-adjacency use-case.

### 3.4. Inter-Area Considerations

In the following example diagram we assume an IGP deployed using areas and where SR has been deployed.

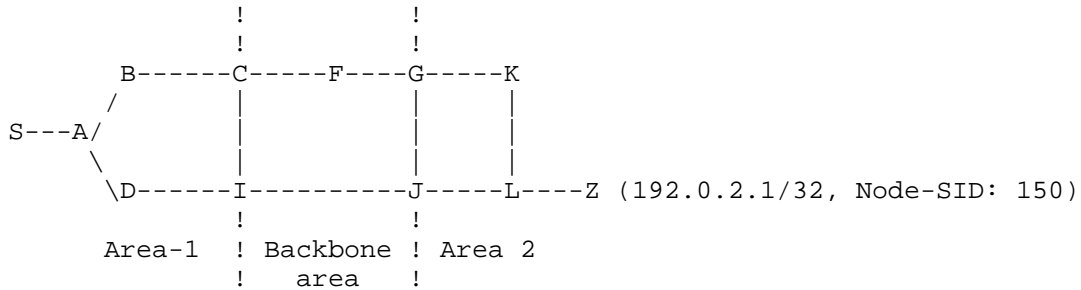


Figure 10: Inter-Area Topology Example

In area 2, node Z allocates Node-SID 150 to his local prefix 192.0.2.1/32. ABRs G and J will propagate the prefix 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 192.0.2.1/32 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.

When node S sends traffic to 192.0.2.1/32, 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.

When an ABR propagates a prefix from one area to another it MUST set the R-Flag.

### 3.5. IGP Mirroring Context Segment

It is beneficial for an IGP node to be able 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.

[I-D.filsfils-rtgwg-segment-routing-use-cases] illustrates such a use-case where two IGP nodes offer the same set of services (e.g. BGP VPN) and mirror each other upon their failure. A similar behavior is described in [I-D.minto-rsvp-lsp-egress-fast-protection].

IS-IS and OSPF Router Capability extensions are described in [I-D.previdi-isis-segment-routing-extensions], [I-D.psenak-ospf-segment-routing-extensions] and [I-D.psenak-ospf-segment-routing-ospfv3-extension].

## 4. Service Segments

A service segment refers to a service offered by a node (e.g. firewall, vpn, etc.).

Further informations will be included in future revisions.

## 5. OAM

SR offers an interesting capability to monitor SR domains:

Any path can be monitored by setting the segment list accordingly.

A path can be expressed with ECMP-awareness or not.

The probe travels along the desired path while staying at the forwarding level.

A monitoring system is able to check any element of the entire SR domain, even if it located multiple hops away.

Some elements of the SR/OAM functionality will require standardization and a related independent draft will eventually be submitted.

SR/OAM use-cases are described in [I-D.filsfils-rtgwg-segment-routing-use-cases].

## 6. Multicast

The text will be added in future revision.

## 7. IANA Considerations

TBD

## 8. Manageability Considerations

TBD

## 9. Security Considerations

TBD

## 10. Acknowledgements

We would like to thank Dave Ward, Dan Frost, Stewart Bryant, Pierre Francois, Thomas Telkamp, Les Ginsberg, Ruediger Geib and Hannes Gredler for their contribution to the content of this document.

## 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.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, October 2005.
- [RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, December 2008.

## 11.2. Informative References

- [I-D.filsfils-rtgwg-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., and E. Crabbe, "Segment Routing Use Cases", draft-filsfils-rtgwg-segment-routing-use-cases-01 (work in progress), July 2013.
- [I-D.francois-sr-frr]  
Francois, P., Filsfils, C., Bashandy, A., Previdi, S., and B. Decraene, "Segment Routing Fast Reroute", draft-francois-sr-frr-00 (work in progress), July 2013.
- [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-02 (work in progress), April 2013.
- [I-D.previdi-isis-segment-routing-extensions]  
Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., and S. Litkowski, "IS-IS Extensions for Segment Routing", draft-previdi-isis-segment-routing-extensions-02 (work in progress), July 2013.
- [I-D.psenak-ospf-segment-routing-extensions]  
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., and R.

Shakir, "OSPF Extensions for Segment Routing",  
draft-psenak-ospf-segment-routing-extensions-02 (work in  
progress), July 2013.

[I-D.psenak-ospf-segment-routing-ospfv3-extension]

Psenak, P. and S. Previdi, "OSPFv3 Extensions for Segment  
Routing", October 2013.

[I-D.sivabalan-pce-segment-routing]

Sivabalan, S., Medved, J., Filsfils, C., Crabbe, E., and  
R. Raszuk, "PCEP Extensions for Segment Routing",  
draft-sivabalan-pce-segment-routing-02 (work in progress),  
October 2013.

[RFC1940] Estrin, D., Li, T., Rekhter, Y., Varadhan, K., and D.  
Zappala, "Source Demand Routing: Packet Format and  
Forwarding Specification (Version 1)", RFC 1940, May 1996.

[draft-filsfils-spring-segment-routing-ldp-interop-00]

Filsfils, C. and S. Previdi, "Segment Routing  
interoperability with LDP", October 2013.

[draft-filsfils-spring-segment-routing-mpls-00]

Filsfils, C. and S. Previdi, "Segment Routing with MPLS  
data plane", October 2013.

#### Authors' Addresses

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

Email: cfilsfil@cisco.com

Stefano Previdi (editor)  
Cisco Systems, Inc.  
Via Del Serafico, 200  
Rome 00142  
Italy

Email: sprevidi@cisco.com

Ahmed Bashandy  
Cisco Systems, Inc.  
170, West Tasman Drive  
San Jose, CA 95134  
US

Email: bashandy@cisco.com

Bruno Decraene  
Orange  
FR

Email: bruno.decraene@orange.com

Stephane Litkowski  
Orange  
FR

Email: stephane.litkowski@orange.com

Martin Horneffer  
Deutsche Telekom  
Hammer Str. 216-226  
Muenster 48153  
DE

Email: Martin.Horneffer@telekom.de

Igor Milojevic  
Telekom Srbija  
Takovska 2  
Belgrade  
RS

Email: igormilojevic@telekom.rs

Rob Shakir  
British Telecom  
London  
UK

Email: rob.shakir@bt.com

Saku Ytti  
TDC Oy  
Mechelininkatu 1a  
TDC 00094  
FI

Email: saku@ytti.fi

Wim Henderickx  
Alcatel-Lucent  
Copernicuslaan 50  
Antwerp 2018  
BE

Email: wim.henderickx@alcatel-lucent.com

Jeff Tantsura  
Ericsson  
300 Holger Way  
San Jose, CA 95134  
US

Email: Jeff.Tantsura@ericsson.com

Edward Crabbe  
Google, Inc.  
1600 Amphitheatre Parkway  
Mountain View, CA 94043  
US

Email: edc@google.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 22, 2014

Pierre Francois  
Institute IMDEA Networks  
Clarence Filsfils  
Ahmed Bashandy  
Cisco Systems, Inc.  
Bruno Decraene  
Stephane Litkowski  
Orange  
November 18, 2013

Topology Independent Fast Reroute using Segment Routing  
draft-francois-segment-routing-ti-lfa-00

Abstract

This document presents a Fast Reroute (FRR) approach aimed at providing link and node protection of node and adjacency segments within the Segment Routing (SR) framework. This FRR behavior builds on proven IP-FRR concepts being LFAs, remote LFAs (RLFA), and remote LFAs with directed forwarding (DLFA). It extends these concepts to provide guaranteed coverage in any IGP network. We accommodate the FRR discovery and selection approaches in order to establish protection over post-convergence paths from the point of local repair, dramatically reducing the operator's need to control the tie-breaks among various FRR options.

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 May 22, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	4
3. Intersecting P-Space and Q-Space with post-convergence paths . . . . .	5
3.1. P-Space property computation for a resource X . . . . .	5
3.2. Q-Space property computation for a link S-F, over post-convergence paths . . . . .	5
3.3. Q-Space property computation for a node F, over post-convergence paths . . . . .	6
4. EPC Repair Tunnel . . . . .	6
4.1. The repair node is a direct neighbor . . . . .	6
4.2. The repair node is a PQ node . . . . .	6
4.3. The repair is a Q node, neighbor of the last P node . . . . .	7
4.4. Connecting distant P and Q nodes along post-convergence paths . . . . .	7
5. Protecting segments . . . . .	7
5.1. The active segment is a node segment . . . . .	7
5.2. The active segment is an adjacency segment . . . . .	7
5.2.1. Protecting [Adjacency, Adjacency] segment lists . . . . .	8
5.2.2. Protecting [Adjacency, Node] segment lists . . . . .	8
6. References . . . . .	8
Authors' Addresses . . . . .	9

1. Introduction

Segment Routing aims at supporting services with tight SLA guarantees [1]. This document provides local repair mechanisms using SR, capable of restoring end-to-end connectivity in the case of a sudden failure of a link or a node, with guaranteed coverage properties.

Using segment routing, there is no need to establish TLDP sessions with remote nodes in order to take advantage of the applicability of remote LFAs (RLFA) or remote LFAs with directed forwarding (DLFA) [2]. As a result, preferring LFAs over RLFAs or DLFAs, as well as minimizing the number of RLFA or DLFA repair nodes is not required. Using SR, there is no need to create state in the network in order to enforce an explicit FRR path. As a result, we can use optimized detour paths for each specific destination and for each possible failure in the network without creating additional forwarding state.

Building on such an easier forwarding environment, the FRR behavior suggested in this document tailors the repair paths over the post-convergence path from the PLR to the protected destination.

As the capacity of the post-convergence path is typically planned by the operator to support the post-convergence routing of the traffic for any expected failure, there is much less need for the operator to tune the decision among which protection path to choose. The protection path will automatically follow the natural backup path that would be used after local convergence. This also helps to reduce the amount of path changes and hence service transients: one transition (pre-convergence to post-convergence) instead of two (pre-convergence to FRR and then post-convergence).

We provide an EPC-FRR approach that achieves guaranteed coverage against link or node failure, in any IGP network, relying on the flexibility of SR.

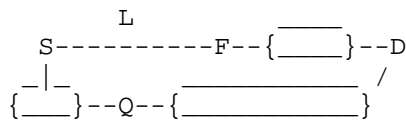


Figure 1: EPC Protection

We use Figure 1 to illustrate the EPC-FRR approach.

The Point of Local Repair (PLR),  $S$ , needs to find a node  $Q$  (a repair node) that is capable of safely forwarding the traffic to a destination  $D$  affected by the failure of the protected link  $L$ , or node  $F$ . The PLR also needs to find a way to reach  $Q$  without being affected by the convergence state of the nodes over the paths it wants to use to reach  $Q$ .

In Section 2 we define the main notations used in the document. They are in line with [2].

In Section 3, we suggest to compute the P-Space and Q-Space properties defined in Section 2, for the specific case of nodes lying over the post-convergence paths towards the protected destinations. The failure of a link  $S-F$  as well as the failure of a neighbor  $F$  is discussed.

Using the properties defined in Section 3, we describe how to compute protection lists that encode a loopfree post-convergence towards the destination, in Section 4.

Finally, we define the segment operations to be applied by the PLR to ensure consistency with the forwarding state of the repair node, in Section 5.

## 2. Terminology

We define the main notations used in this document as the following.

We refer to "old" and "new" topologies as the LSDB state before and after the considered failure.

$SPT\_old(R)$  is the Shortest Path Tree rooted at node  $R$  in the initial state of the network.

$SPT\_new(R, X)$  is the Shortest Path Tree rooted at node  $R$  in the state of the network after the resource  $X$  has failed.

$Dist\_old(A,B)$  is the distance from node  $A$  to node  $B$  in  $SPT\_old(A)$ .

$Dist\_new(A,B, X)$  is the distance from node  $A$  to node  $B$  in  $SPT\_new(A, X)$ .

The P-Space  $P(R,X)$  of a node  $R$  w.r.t. a resource  $X$  (e.g. a link  $S-F$ , or a node  $F$ ) is the set of nodes that are reachable from  $R$  without passing through  $X$ . It is the set of nodes that are not downstream of



X in  $SPT\_old(R)$ .

The Extended P-Space  $P'(R,X)$  of a node R w.r.t. a resource X is the set of nodes that are reachable from R or a neighbor of R, without passing through X.

The Q-Space  $Q(D,X)$  of a destination node D w.r.t. a resource X is the set of nodes which do not use X to reach D in the initial state of the network. In other words, it is the set of nodes which have D in their P-Space w.r.t. S-F (or F).

A symmetric network is a network such that the IGP metric of each link is the same in both directions of the link.

### 3. Intersecting P-Space and Q-Space with post-convergence paths

In this section, we suggest to determine the P-Space and Q-Space properties of the nodes along on the post-convergence paths from the PLR to the protected destination and compute an SR-based explicit path from P to Q when they are not adjacent. Such properties will be used in Section 4 to compute the EPC-FRR repair list.

#### 3.1. P-Space property computation for a resource X

A node N is in  $P(R, X)$  if it is not downstream of X in  $SPT\_old(R)$ .

A node N is in  $P'(R,X)$  if it is not downstream of X in  $SPT\_old(N)$ , for at least one neighbor N of R.

#### 3.2. Q-Space property computation for a link S-F, over post-convergence paths

We want to determine which nodes on the post-convergence from the PLR to the destination D are in the Q-Space of destination D w.r.t. link S-F.

This can be found by intersecting the post-convergence path to D, assuming the failure of S-F, with  $Q(D, S-F)$ .

The post-convergence path to D requires to compute  $SPT\_new(S, S-F)$ .

A node N is in  $Q(D,S-F)$  if it is not downstream of S-F in  $rSPT\_old(D)$ .

### 3.3. Q-Space property computation for a node F, over post-convergence paths

We want to determine which nodes on the post-convergence from the PLR to the destination D are in the Q-Space of destination D w.r.t. node F.

This can be found by intersecting the post-convergence path to D, assuming the failure of F with  $Q(D, F)$ .

The post-convergence path to D requires to compute  $SPT_{new}(S, F)$ .

A node N is in  $Q(D, F)$  if it is not downstream of F in  $rSPT_{old}(D)$ .

## 4. EPC Repair Tunnel

The EPC repair tunnel consists of an outgoing interface and a list of segments (repair list) to insert on the SR header. The repair list encodes the explicit post-convergence path to the destination, which avoids the protected resource X.

The EPC repair tunnel is found by intersecting  $P(S, X)$  and  $Q(D, X)$  with the post-convergence path to D and computing the explicit SR-based path  $EP(P, Q)$  from P to Q when these nodes are not adjacent along the post convergence path. The EPC repair list is expressed generally as  $(Node\_SID(P), EP(P, Q))$ .

Most often, the EPC repair list has a simpler form, as described in the following sections.

### 4.1. The repair node is a direct neighbor

When the repair node is a direct neighbor, the outgoing interface is set to that neighbor and the repair segment list is empty.

This is comparable to an LFA FRR repair.

### 4.2. The repair node is a PQ node

When the repair node is in  $P(S, X)$ , the repair list is made of a single node segment to the repair node.

This is comparable to an RLFA repair tunnel.

#### 4.3. The repair is a Q node, neighbor of the last P node

When the repair node is adjacent to P(S,X), the repair list is made of two segments: A node segment to the adjacent P node, and an adjacency segment from that node to the repair node.

This is comparable to a DLFA repair tunnel.

#### 4.4. Connecting distant P and Q nodes along post-convergence paths

In some cases, there is no adjacent P and Q node along the post-convergence path. However, the PLR can perform additional computations to compute a list of segments that represent a loopfree path from P to Q.

### 5. Protecting segments

In this section, we explain how a protecting router S processes the active segment of a packet upon the failure of its primary outgoing interface.

The behavior depends on the type of active segment to be protected.

#### 5.1. The active segment is a node segment

The active segment is kept on the SR header, unchanged (1). The repair list is inserted at the head of the list. The active segment becomes the first segment of the inserted repair list.

A future version of the document will describe the FRR behavior when the active segment is a node segment destined to F, and F has failed.

Note (1): If the SRGB at the repair node is different from the SRGB at the PLR, then the active segment must be updated to fit the SRGB of the repair node.

#### 5.2. The active segment is an adjacency segment

We define hereafter the FRR behavior applied by S for any packet received with an active adjacency segment S-F for which protection was enabled. We distinguish the case where this active segment is followed by another adjacency segment from the case where it is followed by a node segment.

### 5.2.1. Protecting [Adjacency, Adjacency] segment lists

If the next segment in the list is an Adjacency segment, then the packet has to be conveyed to F.

To do so, S applies a "NEXT" operation on Adj(S-F) and then two consecutive "PUSH" operations: first it pushes a node segment for F, and then it pushes a protection list allowing to reach F while bypassing S-F.

Upon failure of S-F, a packet reaching S with a segment list matching [adj(S-F),adj(M),...] will thus leave S with a segment list matching [RT(F),node(F),adj(M)], where RT(F) is the repair tunnel for destination F.

### 5.2.2. Protecting [Adjacency, Node] segment lists

If the next segment in the stack is a node segment, say for node T, the packet segment list matches [adj(S-F),node(T),...].

A first solution would consist in steering the packet back to F while avoiding S-F, similarly to the previous case. To do so, S applies a "NEXT" operation on Adj(S-F) and then two consecutive "PUSH" operations: first it pushes a node segment for F, and then it pushes a repair list allowing to reach F while bypassing S-F.

Upon failure of S-F, a packet reaching S with a segment list matching [adj(S-F),node(T),...] will thus leave S with a segment list matching [RT(F),node(F),node(T)].

Another solution is to not steer the packet back via F but rather follow the new shortest path to T. In this case, S just needs to apply a "NEXT" operation on the Adjacency segment related to S-F, and push a repair list redirecting the traffic to a node Q, whose path to node segment T is not affected by the failure.

Upon failure of S-F, packets reaching S with a segment list matching [adj(L), node(T), ...], would leave S with a segment list matching [RT(Q),node(T), ...].

## 6. References

- [1] Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-filsfils-rtgwg-segment-routing-00 (work in progress), June 2013.

- [2] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, January 2010.
- [3] Filsfils, C., Francois, P., Shand, M., Decraene, B., Uttaro, J., Leymann, N., and M. Horneffer, "Loop-Free Alternate (LFA) Applicability in Service Provider (SP) Networks", RFC 6571, June 2012.
- [4] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and S. Ning, "Remote LFA FRR", draft-ietf-rtgwg-remote-lfa-02 (work in progress), May 2013.
- [5] Bryant, S., Filsfils, C., Previdi, S., and M. Shand, "IP Fast Reroute using tunnels", draft-bryant-ipfrr-tunnels-03 (work in progress), November 2007.

#### Authors' Addresses

Pierre Francois  
Institute IMDEA Networks  
Leganes  
ES

Email: pierre.francois@imdea.org

Clarence Filsfils  
Cisco Systems, Inc.  
Brussels  
BE

Email: cfilsfil@cisco.com

Ahmed Bashandy  
Cisco Systems, Inc.  
San Jose  
US

Email: bashandy@cisco.com

Bruno Decraene  
Orange  
Issy-les-Moulineaux  
FR

Email: [bruno.decraene@orange.com](mailto:bruno.decraene@orange.com)

Stephane Litkowski  
Orange  
FR

Email: [bruno.decraene@orange.com](mailto:bruno.decraene@orange.com)



Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: October 10, 2014

Pierre Francois  
IMDEA Networks  
Clarence Filsfils  
Cisco Systems, Inc.  
Bruno Decraene  
Orange  
Rob Shakir  
BT  
April 8, 2014

Use-cases for Resiliency in SPRING  
draft-francois-spring-resiliency-use-case-02

Abstract

This document describes the use cases for resiliency in SPRING networks.

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 October 10, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of



the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- 1. Introduction . . . . . 3
- 2. Path protection . . . . . 4
- 3. Management free local protection . . . . . 4
  - 3.1. Management free bypass protection . . . . . 5
  - 3.2. Management-free shortest path based protection . . . . . 5
- 4. Managed local protection . . . . . 6
  - 4.1. Managed bypass protection . . . . . 6
  - 4.2. Managed shortest path protection . . . . . 6
- 5. Co-existence . . . . . 7
- 6. References . . . . . 7
- Authors' Addresses . . . . . 7

## 1. Introduction

SPRING aims at providing a network architecture supporting services with tight SLA guarantees [1]. This document reviews various use cases for the protection of services in a SPRING network. Note that these use cases are in particular applicable to existing LDP based and pure IP networks.

Three key alternatives are described: path protection, local protection without operator management and local protection with operator management.

Path protection lets the ingress node be in charge of the failure recovery, as discussed in Section 2.

The rest of the document focuses on approaches where protection is performed by the node adjacent to the failed component, commonly referred to as local protection techniques or Fast Reroute techniques.

We discuss two different approaches to provide unmanaged local protection, namely link/node bypass protection and shortest path based protection, in Section 3.

A case is then made to allow the operator to manage the local protection behavior in order to accommodate specific policies, in Section 4.

The purpose of this document is to illustrate the different approaches and explain how an operator could combine them in the same network (see Section 5). Solutions are not defined in this document.

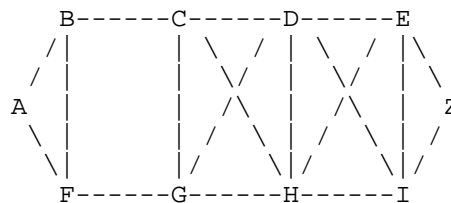


Figure 1: Reference topology

We use Figure 1 as a reference topology throughout the document. All link metrics are equal to 1, with the exception of the links from/to A and Z, which are configured with a metric of 100.

## 2. Path protection

A first protection strategy consists in excluding any local repair but instead use end-to-end path protection.

For example, a Pseudo Wire (PW) from A to Z can be "path protected" in the direction A to Z in the following manner: the operator configures two SPRING paths T1 and T2 from A to Z. The two paths are installed in the forwarding plane of A and hence are ready to forward packets. The two paths are made disjoint using the SPRING architecture.

T1 is established over path {AB, BC, CD, DE, EZ} and T2 over path {AF, FG, GH, HI, IZ}. When T1 is up, the packets of the PW are sent on T1. When T1 fails, the packets of the PW are sent on T2. When T1 comes back up, the operator either allows for an automated reversion of the traffic onto T1 or selects an operator-driven reversion. The solution to detect the end-to-end liveness of the path is out of the scope of this document.

From a SPRING viewpoint, we would like to highlight the following requirement: the two configured paths T1 and T2 MUST NOT benefit from local protection.

## 3. Management free local protection

This section describes two alternatives to provide local protection without requiring operator management, namely bypass protection and shortest-path based protection.

For example, a demand from A to Z, transported over the shortest paths provided by the SPRING architecture, benefits from management-free local protection by having each node along the path automatically pre-compute and pre-install a backup path for the destination Z. Upon local detection of the failure, the traffic is repaired over the backup path in sub-50msec.

The backup path computation should support the following requirements:

- o 100% link, node, and SRLG protection in any topology
- o Automated computation by the IGP
- o Selection of the backup path such as to minimize the chance for transient congestion and/or delay during the protection period, as reflected by the IGP metric configuration in the network.

3.1. Management free bypass protection

One way to provide local repair is to enforce a failover along the shortest path around the failed component, ending at the protected nexthop, so as to bypass the failed component and re-join the pre-convergence path at the nexthop. In the case of node protection, such bypass ends at the next-nexthop.

In our example, C protects Z, that it initially reaches via CD, by enforcing the traffic over the bypass {CH, HD}. The resulting end-to-end path between A and Z, upon recovery against the failure of C-D, is depicted in Figure 2.

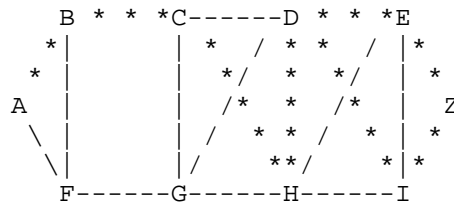


Figure 2: Bypass protection around link C-D

3.2. Management-free shortest path based protection

An alternative protection strategy consists in management-free local protection, aiming at providing a repair for the destination based on shortest path state for that destination.

In our example, C protects Z, that it initially reaches via CD, by enforcing the traffic over its shortest path to Z, considering the failure of the protected component. The resulting end-to-end path between A and Z, upon recovery against the failure of C-D, is depicted in Figure 3.

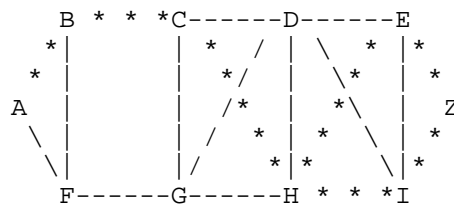


Figure 3: Reference topology

4. Managed local protection

There may be cases where a management free repair does not fit the policy of the operator. For example, in our illustration, the operator may want to not have C-D and C-H used to protect each other, in fear of a shared risk among the two links.

In this context, the protection mechanism must support the explicit configuration of the backup path either under the form of high-level constraints (end at the next-hop, end at the next-next-hop, minimize this metric, avoid this SRLG...) or under the form of an explicit path.

We discuss such aspects for both bypass and shortest path based protection schemes.

4.1. Managed bypass protection

Let us illustrate the case using our reference example. For the demand from A to B, the operator does not want to use the shortest failover path to the nexthop, {CH, HD}, but rather the path {CG,GH,HD}, as illustrated in Figure 4.

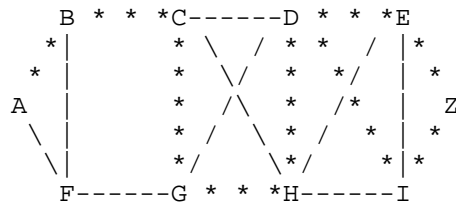


Figure 4: Managed bypass protection

4.2. Managed shortest path protection

In the case of shortest path protection, the case is the one of an operator who does not want to use the shortest failover via link C-H, but rather reach H via {CG, GH}.

The resulting end-to-end path upon activation of the protection is illustrated in Figure 5.

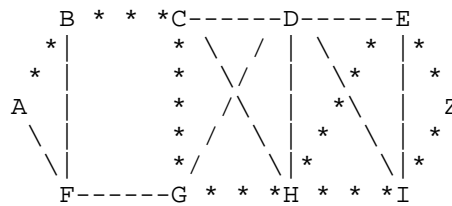


Figure 5: Managed shortest path protection

5. Co-existence

The operator may want to support several very-different services on the same packet-switching infrastructure. As a result, the SPRING architecture SHOULD allow for the co-existence of the different use cases listed in this document, in the same network.

Let us illustrate this with the following example.

- o Flow F1 is supported over path {C, C-D, E}
- o Flow F2 is supported over path {C, C-D, I}
- o Flow F3 is supported over path {C, C-D, Z}
- o Flow F4 is supported over path {C, C-D, Z}
- o It should be possible for the operator to configure the network to achieve path protection for F1, management free shortest path local protection for F2, managed protection over path {C-G, G-H, Z} for F3, and management free bypass protection for F4.

6. References

[1] Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-filsfils-rtgwg-segment-routing-01 (work in progress), October 2013.

Authors' Addresses

Pierre Francois  
 IMDEA Networks  
 Leganes  
 ES

Email: pierre.francois@imdea.org

Clarence Filsfils  
Cisco Systems, Inc.  
Brussels  
BE

Email: cfilsfil@cisco.com

Bruno Decraene  
Orange  
Issy-les-Moulineaux  
FR

Email: bruno.decraene@orange.com

Rob Shakir  
BT  
London  
UK

Email: rob.shakir@bt.com





spring  
Internet-Draft  
Intended status: Informational  
Expires: January 4, 2016

R. Geib, Ed.  
Deutsche Telekom  
C. Filsfils  
C. Pignataro  
N. Kumar  
Cisco Systems, Inc.  
July 3, 2015

Use case for a scalable and topology aware MPLS data plane monitoring  
system  
draft-geib-spring-oam-usecase-06

#### Abstract

This document describes features and a use case of a path monitoring system. Segment based routing enables a scalable and simple method to monitor data plane liveness of the complete set of paths belonging to a single domain. Compared with legacy MPLS ping and path trace, MPLS topology awareness reduces management and control plane involvement of OAM measurements while enabling new OAM features.

#### 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 January 4, 2016.

#### Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. An MPLS topology aware path monitoring system . . . . .	4
3. SR based path monitoring use case illustration . . . . .	5
3.1. Use-case 1 - LSP dataplane monitoring . . . . .	5
3.2. Use-case 2 - Monitoring a remote bundle . . . . .	7
3.3. Use-Case 3 - Fault localization . . . . .	8
4. Failure Notification from PMS to LERi . . . . .	8
5. Applying SR to monitor LDP paths . . . . .	9
6. PMS monitoring of different Segment ID types . . . . .	9
7. Connectivity Verification using PMS . . . . .	9
8. Extensions of related standards helpful for this use case . .	10
9. IANA Considerations . . . . .	10
10. Security Considerations . . . . .	10
11. Acknowledgement . . . . .	10
12. References . . . . .	10
12.1. Normative References . . . . .	10
12.2. Informative References . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

It is essential for a network operator to monitor all the forwarding paths observed by the transported user packets. The monitoring flow is expected to be forwarded in dataplane in a similar way as user packets. Segment Routing enables forwarding of packets along pre-defined paths and segments and thus a Segment Routed monitoring packet can stay in dataplane while passing along one or more segments to be monitored.

This document describes illustrates use-cases based on data plane path monitoring capabilities. The use case is limited to a single IGP MPLS domain.

The use case applies to monitoring of LDP LSP's as well as to monitoring of Segment Routed LSP's. As compared to LDP, Segment Routing is expected to simplify the use case by enabling MPLS topology detection based on IGP signaled segments as specified by [ID.sr-isis]. Thus a centralised and MPLS topology aware monitoring unit can be realized in a Segment Routed domain. This topology

awareness can be used for OAM purposes as described by this use case. The MPLS path monitoring system described by this document can be realised with pre-Segment based Routing (SR) technology. Making such a pre-SR MPLS monitoring system aware of a domains complete MPLS topology requires e.g. management plane access. To avoid the use of stale MPLS label information, IGP must be monitored and MPLS topology must be timely aligned with IGP topology. Obviously, enhancing IGPs to exchange of MPLS topology information as done by SR significantly simplifies and stabilises such an MPLS path monitoring system.

This document adopts the terminology and framework described in [ID.sr-archi]. It further adopts the editorial simplification explained in section 1.2 of the segment routing use-cases [ID.sr-use].

The use case offers several benefits for network monitoring. A single centralized monitoring device is able to monitor the complete set of a domains forwarding paths. Monitoring packets never leave data plane. MPLS path trace function (whose specification and features are not part of this use case) is required, if the actual data plane of a router should be checked against its control plane. SR capabilities allow to direct MPLS OAM packets from a centralized monitoring system to any router within a domain whose path should be traced.

In addition to monitoring paths, problem localization is required. Faults can be localized:

- o by IGP LSA analysis.
- o correlation between different SR based monitoring probes.
- o by any MPLS traceroute method (possibly in combination with SR based path stacks).

Topology awareness is an essential part of link state IGPs. Adding MPLS topology awareness to an IGP speaking device hence enables a simple and scalable data plane based monitoring mechanism.

MPLS OAM offers flexible features to recognise an execute data paths of an MPLS domain. By utilising the ECMP related tool set offered e.g. by RFC 4379 [RFC4379], a segment based routing LSP monitoring system may:

- o easily detect ECMP functionality and properties of paths at data level.

- o construct monitoring packets executing desired paths also if ECMP is present.
- o limit the MPLS label stack of an OAM packet to a minimum of 3 labels.

Alternatively, any path may be executed by building suitable label stacks. This allows path execution without ECMP awareness.

The MPLS path monitoring system may be a any server residing at a single interface of the domain to be monitored. It doesn't have to support any specialised protocol stack, it just should be capable of understanding the topology and building the probe packet with the right segment stack. As long as measurement packets return to this or another interface connecting such a server, the MPLS monitoring servers are the single entities pushing monitoring packet label stacks. If the depth of label stacks to be pushed by a path monitoring system (PMS) are of concern for a domain, a dedicated server based path monitoring architecture allows limiting monitoring related label stack pushes to these servers.

First drafts discussing SR OAM requirements and possible solutions to allow SR usage as described by this document have been submitted already, see [ID.sr-4379ext] and [ID.sr-oam\_detect].

## 2. An MPLS topology aware path monitoring system

An MPLS PMS which is able to learn the IGP LSDB (including the SID's) is able to execute arbitrary chains of label switched paths. It can send pure monitoring packets along such a path chain or it can direct suitable MPLS OAM packets to any node along a path segment. Segment Routing here is used as a means of adding label stacks and hence transport to standard MPLS OAM packets, which then detect correspondence of control and data plane of this (or any other addressed) path. Any node connected to an SR domain is MPLS topology aware (the node knows all related IP addresses, SR SIDs and MPLS labels). Thus a PMS connected to an MPLS SR domain just needs to set up a topology data base for monitoring purposes.

Let us describe how the PMS constructs a labels stack to transport a packet to LER i, monitor the path of it to LER j and then receive the packet back.

The PMS may do so by sending packets carrying the following MPLS label stack information:

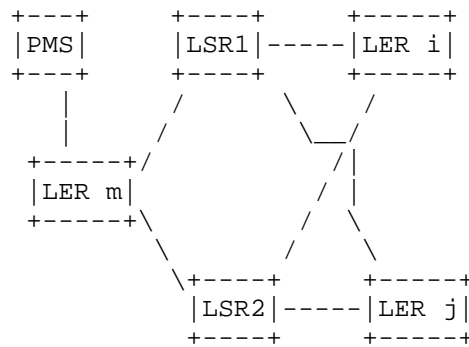
- o Top Label: a path from PMS to LER i, which is expressed as Node SID of LER i.

- o Next Label: the path that needs to be monitored from LER i to LER j. If this path is a single physical interface (or a bundle of connected interfaces), it can be expressed by the related AdjSID. If the shortest path from LER i to LER j is supposed to be monitored, the Node-SID (LER j) can be used. Another option is to insert a list of segments expressing the desired path (hop by hop as an extreme case). If LER i pushes a stack of Labels based on a SR policy decision and this stack of LSPs is to be monitored, the PMS needs an interface to collect the information enabling it to address this SR created path.
- o Next Label or address: the path back to the PMS. Likely, no further segment/label is required here. Indeed, once the packet reaches LER j, the 'steering' part of the solution is done and the probe just needs to return to the PMS. This is best achieved by popping the MPLS stack and revealing a probe packet with PMS as destination address (note that in this case, the source and destination addresses could be the same). If an IP address is applied, no SID/label has to be assigned to the PMS (if it is a host/server residing in an IP subnet outside the MPLS domain).

Note: if the PMS is an IP host not connected to the MPLS domain, the PMS can send its probe with the list of SIDs/Labels onto a suitable tunnel providing an MPLS access to a router which is part of the monitored MPLS domain.

### 3. SR based path monitoring use case illustration

#### 3.1. Use-case 1 - LSP dataplane monitoring



Example of a PMS based LSP dataplane monitoring

Figure 1

For the sake of simplicity, let's assume that all the nodes are configured with the same SRGB [ID.sr-archi], as described by section 1.2 of [ID.sr-use].

Let's assign the following Node SIDs to the nodes of the figure: PMS = 10, LER i = 20, LER j = 30.

To be able to work with the smallest possible SR label stack, first a suitable MPLS OAM method is used to detect the ECMP routed path between LER i to LER j which is to be monitored (and the required address information to direct a packet along it). Afterwards the PMS sets up and sends packets to monitor availability of the detected path. The PMS does this by creating a measurement packet with the following label stack (top to bottom): 20 - 30 - 10. The packet will only reliably use the monitored path, if the label and address information used in combination with the MPLS OAM method of choice is identical to that of the monitoring packet.

LER m forwards the packet received from the PMS to LSR1. Assuming Pen-ultimate Hop Popping to be deployed, LSR1 pops the top label and forwards the packet to LER i. There the top label has a value 30 and LER i forwards it to LER j. This will be done transmitting the packet via LSR1 or LSR2. The LSR will again pop the top label. LER j will forward the packet now carrying the top label 10 to the PMS (and it will pass a LSR and LER m).

A few observations on the example given in figure 1:

- o The path PMS to LER i must be available. This path must be detectable, but it is usually sufficient to apply a Shortest Path First algorithm based path.
- o If ECMP is deployed, it may be desired to measure along both possible paths which a packet may use between LER i and LER j. To do so, the MPLS OAM mechanism chosen to detect ECMP must reveal the required information (an example is a so called tree trace) between LER i and LER j. This method of dealing with ECMP based load balancing paths requires the smallest SR label stacks if monitoring of paths is applied after the tree trace completion.
- o The path LER j to PMS to must be available. This path must be detectable, but it is usually sufficient to apply an SPF based path.

Once the MPLS paths (Node SIDs) and the required information to deal with ECMP has been detected, the paths of LER i to LER j can be monitored by the PMS. Monitoring itself does not require MPLS OAM functionality. All monitoring packets stay on dataplane, hence path

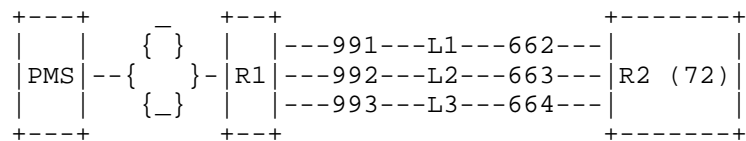
monitoring does no longer require control plane interaction in any LER or LSR of the domain. To ensure reliable results, the PMS should be aware of any changes in IGP or MPLS topology. Further changes in ECMP functionality at LER i will impact results. Either the PMS should be notified of such changes or they should be limited to planned maintenance. After a topology change, a suitable MPLS OAM mechanism may be useful to detect the impact of the change.

Determining a path to be executed prior to a measurement may also be done by setting up a label stack including all Node SIDs along that path (if LSR1 has Node SID 40 in the example and it should be passed between LER i and LER j, the label stack is 20 - 40 - 30 - 10). The advantage of this method is, that it does not involve MPLS OAM functionality and it is independent of ECMP functionalities. The method still is able to monitor all link combinations of all paths of an MPLS domain. If correct forwarding along the desired paths has to be checked, some suitable MPLS OAM mechanism may be applied also in this case.

In theory at least, a single PMS is able to monitor data plane availability of all LSPs in the domain. The PMS may be a router, but could also be dedicated monitoring system. If measurement system reliability is an issue, more than a single PMS may be connected to the MPLS domain.

Monitoring an MPLS domain by a PMS based on SR offers the option of monitoring complete MPLS domains with little effort and very excellent scalability. Data plane failure detection by circulating monitoring packets can be executed at any time. The PMS further could be enabled to send MPLS OAM packets with the label stacks and address information identical to those of the monitoring packets to any node of the MPLS domain. It does not require access to LSR/LER management interfaces or their control plane to do so.

3.2. Use-case 2 - Monitoring a remote bundle



SR based probing of all the links of a remote bundle

Figure 2

R1 addresses Lx by the Adjacency SID 99x, while R2 addresses Lx by the Adjacency SID 66(x+1).

In the above figure, the PMS needs to assess the dataplane availability of all the links within a remote bundle connected to routers R1 and R2.

The monitoring system retrieves the SID/Label information from the IGP LSDB and appends the following segment list/label stack: {72, 662, 992, 664} on its IP probe (whose source and destination addresses are the address of the PMS).

PMS sends the probe to its connected router. If the connected router is not SR compliant, a tunneling technique can be used to tunnel the probe and its MPLS stack to the first SR router. The MPLS/SR domain then forwards the probe to R2 (72 is the Node SID of R2). R2 forwards the probe to R1 over link L1 (Adjacency SID 662). R1 forwards the probe to R2 over link L2 (Adjacency SID 992). R2 forwards the probe to R1 over link L3 (Adjacency SID 664). R1 then forwards the IP probe to PMS as per classic IP forwarding.

### 3.3. Use-Case 3 - Fault localization

In the previous example, a uni-directional fault on the middle link in direction of R2 to R1 would be localized by sending the following two probes with respective segment lists:

- o 72, 662, 992, 664
- o 72, 663, 992, 664

The first probe would fail while the second would succeed. Correlation of the measurements reveals that the only difference is using the Adjacency SID 662 of the middle link from R1 to R2 in the non successful measurement. Assuming the second probe has been routed correctly, the fault must have been occurring in R2 which didn't forward the packet to the interface identified by its Adjacency SID 662.

### 4. Failure Notification from PMS to LERi

PMS on detecting any failure in the path liveness may use any out-of-band mechanism to signal the failure to LER i. This document does not propose any specific mechanism and operators can choose any existing or new approach.

Alternately, the Operator may log the failure in local monitoring system and take necessary action by manual intervention.



## 5. Applying SR to monitor LDP paths

A SR based PMS connected to a MPLS domain consisting of LER and LSR supporting SR and LDP in parallel in all nodes may use SR paths to transmit packets to and from start and end points of LDP paths to be monitored. In the above example, the label stack top to bottom may be as follows, when sent by the PMS:

- o Top: SR based Node-SID of LER i at LER m.
- o Next: LDP label identifying the path to LER j at LER i.
- o Bottom: SR based Node-SID identifying the path to the PMS at LER j

While the mixed operation shown here still requires the PMS to be aware of the LER LDP-MPLS topology, the PMS may learn the SR MPLS topology by IGP and use this information.

## 6. PMS monitoring of different Segment ID types

MPLS SR topology awareness should allow the SID to monitor liveness of most types of SIDs (this may not be recommendable if a SID identifies an inter domain interface).

To match control plane information with data plane information, MPLS OAM functions as defined by e.g. RFC4379 should be enhanced to allow collection of data relevant to check all relevant types of Segment IDs.

## 7. Connectivity Verification using PMS

While the PMS based use cases explained in Section 3 are sufficient to provide continuity check between LER i and LER j, it may not help perform connectivity verification. So in some cases like data plane programming corruption, it is possible that a transit node between LER i and LER j erroneously removes the top segment ID and forwards a monitoring packet to the PMS based on the bottom segment ID leading to a falsified path liveness indication by the PMS.

There are various method to perform basic connectivity verification like intermittently setting the TTL to 1 in bottom label so LER j selectively perform connectivity verification. Other methods are possible and may be added when requirements and solutions are specified.

## 8. Extensions of related standards helpful for this use case

The following activities are welcome enhancements supporting this use case, but they are not part of it:

RFC4379 functions should be extended to support Flow- and Entropy Label based ECMP.

## 9. IANA Considerations

This memo includes no request to IANA.

## 10. Security Considerations

As mentioned in the introduction, a PMS monitoring packet should never leave the domain where it originated. It therefore should never use stale MPLS or IGP routing information. Further, assigning different label ranges for different purposes may be useful. A well known global service level range may be excluded for utilisation within PMS measurement packets. These ideas shouldn't start a discussion. They rather should point out, that such a discussion is required when SR based OAM mechanisms like a SR are standardised.

## 11. Acknowledgement

The authors would like to thank Nobo Akiya for his contribution. Raik Leinnitz kindly provided an editorial review.

## 12. References

### 12.1. Normative References

[RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.

### 12.2. Informative References

[ID.sr-4379ext]  
IETF, "Label Switched Path (LSP) Ping/Trace for Segment Routing Networks Using MPLS Dataplane", IETF, <http://datatracker.ietf.org/doc/draft-kumar-mpls-spring-lsp-ping/>, 2013.

[ID.sr-archi]  
IETF, "Segment Routing Architecture", IETF, <https://datatracker.ietf.org/doc/draft-filsfils-spring-segment-routing/>, 2014.

## [ID.sr-isis]

IETF, "IS-IS Extensions for Segment Routing", IETF,  
[http://datatracker.ietf.org/doc/  
draft-previdi-isis-segment-routing-extensions/](http://datatracker.ietf.org/doc/draft-previdi-isis-segment-routing-extensions/), 2014.

## [ID.sr-oam\_detect]

IETF, "Detecting Multi-Protocol Label Switching (MPLS)  
Data Plane Failures in Source Routed LSPs", IETF,  
[http://datatracker.ietf.org/doc/  
draft-kini-spring-mpls-lsp-ping/](http://datatracker.ietf.org/doc/draft-kini-spring-mpls-lsp-ping/), 2013.

## [ID.sr-use]

IETF, "Segment Routing Use Cases", IETF,  
[http://datatracker.ietf.org/doc/  
draft-filsfils-rtgwg-segment-routing-use-cases/](http://datatracker.ietf.org/doc/draft-filsfils-rtgwg-segment-routing-use-cases/), 2013.

## Authors' Addresses

Ruediger Geib (editor)  
Deutsche Telekom  
Heinrich Hertz Str. 3-7  
Darmstadt 64295  
Germany

Phone: +49 6151 5812747  
Email: [Ruediger.Geib@telekom.de](mailto:Ruediger.Geib@telekom.de)

Clarence Filsfils  
Cisco Systems, Inc.  
Brussels  
Belgium

Email: [cfilsfil@cisco.com](mailto:cfilsfil@cisco.com)

Carlos Pignataro  
Cisco Systems, Inc.  
7200 Kit Creek Road  
Research Triangle Park, NC 27709-4987  
US

Email: [cpignata@cisco.com](mailto:cpignata@cisco.com)

Nagendra Kumar  
Cisco Systems, Inc.  
7200 Kit Creek Road  
Research Triangle Park, NC 27709  
US

Email: [naikumar@cisco.com](mailto:naikumar@cisco.com)

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: September 4, 2015

S. Kini, Ed.  
Ericsson  
K. Kompella  
Juniper  
S. Sivabalan  
Cisco  
S. Litkowski  
Orange  
R. Shakir  
B.T.  
X. Xu  
Huawei  
W. Hendrickx  
Alcatel-Lucent  
J. Tantsura  
Ericsson  
March 3, 2015

Entropy labels for source routed stacked tunnels  
draft-kini-mpls-spring-entropy-label-03

Abstract

Source routed tunnel stacking is a technique that can be leveraged to provide a method to steer a packet through a controlled set of segments. This can be applied to the Multi Protocol Label Switching (MPLS) data plane. Entropy label (EL) is a technique used in MPLS to improve load balancing. This document examines and describes how ELs are to be applied to source routed stacked tunnels.

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 4, 2015.

## Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Requirements Language . . . . .	3
2. Abbreviations and Terminology . . . . .	3
3. Use-case requiring multipath load balancing in source stacked tunnels . . . . .	4
4. Recommended EL solution for SPRING . . . . .	5
5. Options considered . . . . .	6
5.1. Single EL at the bottom of the stack of tunnels . . . . .	6
5.2. An EL per tunnel in the stack . . . . .	7
5.3. A re-usable EL for a stack of tunnels . . . . .	7
5.3.1. EL at top of stack . . . . .	8
5.4. ELs at readable label stack depths . . . . .	8
6. Acknowledgements . . . . .	9
7. IANA Considerations . . . . .	9
8. Security Considerations . . . . .	9
9. References . . . . .	9
9.1. Normative References . . . . .	9
9.2. Informative References . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

The source routed stacked tunnels paradigm is leveraged by techniques such as Segment Routing (SR) [I-D.filsfils-spring-segment-routing] to steer a packet through a set of segments. This can be directly applied to the MPLS data plane, but it has implications on label stack depth.

Clarifying statements on label stack depth have been provided in [RFC7325] but they do not address the case of source routed stacked MPLS tunnels as described in [I-D.gredler-spring-mpls] or

[I-D.filsfils-spring-segment-routing] where deeper label stacks are more prevalent.

Entropy label (EL) [RFC6790] is a technique used in the MPLS data plane to provide entropy for load balancing. When using LSP hierarchies there are implications on how [RFC6790] should be applied. One such issue is addressed by [I-D.ravisingh-mpls-el-for-seamless-mpls] but that is when different levels of the hierarchy are created at different LSRs. The current document addresses the case where the hierarchy is created at a single LSR as required by source stacked tunnels.

A use-case requiring load balancing with source stacked tunnels is given in Section 3. A recommended solution is described in Section 4 keeping in consideration the limitations of implementations when applying [RFC6790] to deeper label stacks. Options that were considered to arrive at the recommended solution are documented for historical purposes in Section 5.

### 1.1. 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].

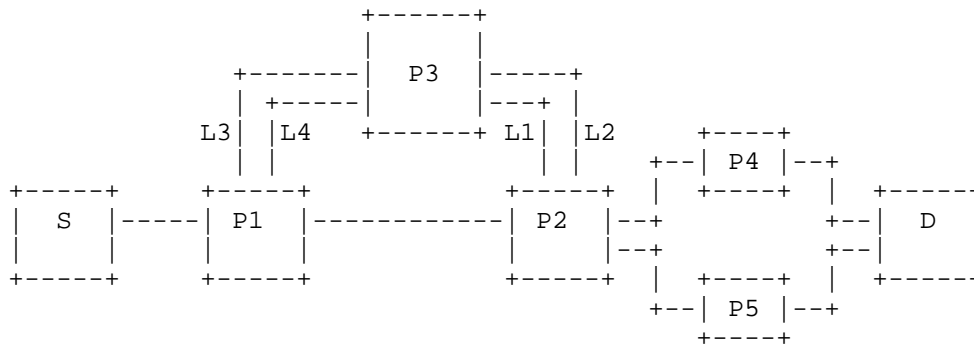
Although this document is not a protocol specification, the use of this language clarifies the instructions to protocol designers producing solutions that satisfy the requirements set out in this document.

## 2. Abbreviations and Terminology

EL - Entropy Label  
ELI - Entropy Label Identifier  
ELC - Entropy Label Capability  
SR - Segment Routing  
ECMP - Equal Cost Multi Paths  
MPLS - Multiprotocol Label Switching  
SID - Segment Identifier  
RLD - Readable Label Depth

OAM - Operation, Administration and Maintenance

- 3. Use-case requiring multipath load balancing in source stacked tunnels



S=Source LSR, D=Destination LSR, P1,P2,P3,P4,P5=Transit LSRs, L1,L2,L3,L4=Links

Figure 1: Traffic engineering use-case

Traffic-engineering (TE) is one of the applications of MPLS and is also a requirement for source stacked tunnels. Consider the topology shown in Figure 1. Lets say the LSR P1 has a limitation that it can only look four labels deep in the stack to do multipath decisions. All other transit LSRs in the figure can read deep label stacks and the LSR S can insert as many <ELI, EL> pairs as needed. The LSR S requires data to be sent to LSR D along a traffic-engineered path that goes over the link L1. Good load balancing is also required across equal cost paths (including parallel links). To engineer traffic along a path that takes link L1, the label stack that LSR S creates consists of a label to the node SID of LSR P3, stacked over the label for the adjacency SID of link L1 and that in turn is stacked over the label to the node SID of LSR D. For simplicity lets assume that all LSRs use the same label space for source stacked tunnels. Lets L<sub>N</sub>-P denote the label to be used to reach the node SID of LSR P. Let L<sub>A</sub>-Ln denote the label used for the adjacency SID for link Ln. The LSR S must use the label stack <L<sub>N</sub>-P3, L<sub>A</sub>-L1, L<sub>N</sub>-D> for traffic-engineering. However to achieve good load balancing over the equal cost paths P2-P4-D, P2-P5-D and the parallel links L3, L4, a mechanism such as Entropy labels [RFC6790] should be adapted for source stacked tunnels. Multiple ways to apply entropy labels were considered and are documented in Section 5 along with their tradeoffs. A recommended solution is described in Section 4.



#### 4. Recommended EL solution for SPRING

The solution described in this section follows [RFC6790].

An LSR may have a limitation in its ability to read and process the label stack in order to do multipath load balancing. This limitation expressed in terms of the number of label stack entries that the LSR can read is henceforth referred to as the Readable Label Depth (RLD) capability of that LSR. If an EL does not occur within the RLD of an LSR in the label stack of the MPLS packet that it receives, then it would lead to poor load balancing at that LSR. The RLD of an LSR is a characteristic of the forwarding plane of that LSR's implementation and determining it is outside the scope of this document.

In order for the EL to occur within the RLD of LSRs along the path corresponding to a label stack, multiple <ELI, EL> pairs MAY be inserted in the label stack as long as the tunnel's label below which they are inserted are advertised with entropy label capability enabled. The LSR that inserts <ELI, EL> pairs MAY have limitations on the number of such pairs that it can insert and also the depth at which it can insert them. If due to any limitation, the inserted ELs are at positions such that an LSR along the path receives an MPLS packet without an EL in the label stack within that LSR's RLD, then the load balancing performed by that LSR would be poor. Special attention should be paid when a forwarding adjacency LSP (FA-LSP) [RFC4206] is used as a link along the path of a source stacked LSP, since the labels of the FA-LSP would additionally count towards the depth of the label stack when calculating the appropriate positions to insert the ELs. The recommendations for inserting <ELI, EL> pairs are:

- o An LSR that is limited in the number of <ELI, EL> pairs that it can insert SHOULD insert such pairs deeper in the stack.
- o An LSR SHOULD try to insert <ELI, EL> pairs at positions so that for the maximum number of transit LSRs, the EL occurs within the RLD of the incoming packet to that LSR.
- o An LSR SHOULD try to insert the minimum number of such pairs while trying to satisfy the above criteria.

A sample algorithm to insert ELs is shown below. Implementations can choose any algorithm as long as it follows the above recommendations.

```

Initialize the current EL insertion point to the
  bottommost label in the stack that is EL-capable
while (local-node can push more <ELI,EL> pairs OR
      insertion point is not above label stack) {
  insert an <ELI,EL> pair below current insertion point
  move new insertion point up from current insertion point until
    ((last inserted EL is below the RLD) AND (RLD > 2)
     AND
     (new insertion point is EL-capable))
  set current insertion point to new insertion point
}

```

Figure 2: Algorithm to insert <ELI, EL> pairs in a label stack

When this algorithm is applied to the example described in Section 3 it will result in ELs being inserted in two positions, one below the label L<sub>N</sub>-D and another below L<sub>N</sub>-P3. Thus the resulting label stack would be <L<sub>N</sub>-P3, ELI, EL, L<sub>A</sub>-L1, L<sub>N</sub>-D, ELI, EL>

The RLD can be advertised via protocols and those extensions would be described in separate documents [I-D.xu-isis-mpls-elc] and [I-D.xu-ospf-mpls-elc].

The recommendations above are not expected to bring any additional OAM considerations beyond those described in section 6 of [RFC6790]. However, the OAM requirements and solutions for source stacked tunnels are still under discussion and future revisions of this document will address those if needed.

## 5. Options considered

### 5.1. Single EL at the bottom of the stack of tunnels

In this option a single EL is used for the entire label stack. The source LSR S encodes the entropy label (EL) below the labels of all the stacked tunnels. In the example described in Section 3 it will result in the label stack at LSR S to look like <L<sub>N</sub>-P3, L<sub>A</sub>-L1, L<sub>N</sub>-D, ELI, EL> <remaining packet header>. Note that the notation in [RFC6790] is used to describe the label stack. An issue with this approach is that as the label stack grows due an increase in the number of SIDs, the EL goes correspondingly deeper in the label stack. Hence transit LSRs have to access a larger number of bytes in the packet header when making forwarding decisions. In the example described in Section 3 the LSR P1 would poorly load-balance traffic on the parallel links L3, L4 since the EL is below the RLD of the packet received by P1. A load balanced network design using this approach must ensure that all intermediate LSRs have the capability

to traverse the maximum label stack depth as required for that application that uses source routed stacking.

In the case where the hardware is capable of pushing a single <ELI, EL> pair at any depth, this option is the same as the recommended solution in Section 4.

This option was discounted since there exist a number of hardware implementations which have a low maximum readable label depth. Choosing this option can lead to a loss of load-balancing using EL in a significant part of the network but that is a critical requirement in a service provider network.

#### 5.2. An EL per tunnel in the stack

In this option each tunnel in the stack can be given its own EL. The source LSR pushes an <ELI, EL> before pushing a tunnel label when load balancing is required to direct traffic on that tunnel. In the example described in Section 3, the source LSR S encoded label stack would be <L\_N-P3, ELI, EL, L\_A-L1, L\_N-D, ELI, EL> where all the ELs can be the same. Accessing the EL at an intermediate LSR is independent of the depth of the label stack and hence independent of the specific application that uses source stacking on that network. A drawback is that the depth of the label stack grows significantly, almost 3 times as the number of labels in the label stack. The network design should ensure that source LSRs should have the capability to push such a deep label stack. Also, the bandwidth overhead and potential MTU issues of deep label stacks should be accounted for in the network design.

In the case where the RLD is the minimum value (3) for all LSRs, all LSRs are EL capable and the LSR that is inserting <ELI, EL> pairs has no limit on how many it can insert then this option is the same as the recommended solution in Section 4.

This option was discounted due to the existence of hardware implementations that can push a limited number of labels on the label stack. Choosing this option would result in a hardware requirement to push two additional labels per tunnel label. Hence it would restrict the number of tunnels that can form a LSP and constrain the types of LSPs that can be created. This was considered unacceptable.

#### 5.3. A re-usable EL for a stack of tunnels

In this option an LSR that terminates a tunnel re-uses the EL of the terminated tunnel for the next inner tunnel. It does this by storing the EL from the outer tunnel when that tunnel is terminated and re-inserting it below the next inner tunnel label during the label swap

operation. The LSR that stacks tunnels SHOULD insert an EL below the outermost tunnel. It SHOULD NOT insert ELs for any inner tunnels. Also, the penultimate hop LSR of a segment MUST NOT pop the ELI and EL even though they are exposed as the top labels since the terminating LSR of that segment would re-use the EL for the next segment.

In Section 3 above, the source LSR S encoded label stack would be <L\_N-P3, ELI, EL, L\_A-L1, L\_N-D>. At P1 the outgoing label stack would be <L\_N-P3, ELI, EL, L\_A-L1, L\_N-D> after it has load balanced to one of the links L3 or L4. At P3 the outgoing label stack would be <L\_N-D, ELI, EL>. At P2 the outgoing label stack would be <L\_N-D, ELI, EL> and it would load balance to one of the nexthop LSRs P4 or P5. Accessing the EL at an intermediate LSR (e.g. P1) is independent of the depth of the label stack and hence independent of the specific use-case to which the stacked tunnels are applied.

This option was discounted due to the significant change in label swap operations that would be required for existing hardware.

#### 5.3.1. EL at top of stack

A slight variant of the re-usable EL option is to keep the EL at the top of the stack rather than below the tunnel label. In this case each LSR that is not terminating a segment should continue to keep the received EL at the top of the stack when forwarding the packet along the segment. An LSR that terminates a segment should use the EL from the terminated segment at the top of the stack when forwarding onto the next segment.

This option was discounted due to the significant change in label swap operations that would be required for existing hardware.

#### 5.4. ELs at readable label stack depths

In this option the source LSR inserts ELs for tunnels in the label stack at depths such that each LSR along the path that must load balance is able to access at least one EL. Note that the source LSR may have to insert multiple ELs in the label stack at different depths for this to work since intermediate LSRs may have differing capabilities in accessing the depth of a label stack. The label stack depth access value of intermediate LSRs must be known to create such a label stack. How this value is determined is outside the scope of this document. This value can be advertised using a protocol such as an IGP. For the same Section 3 above, if LSR P1 needs to have the EL within a depth of 4, then the source LSR S encoded label stack would be <L\_N-P3, ELI, EL, L\_A-L1, L\_N-D, ELI, EL> where all the ELs would typically have the same value.

In the case where the RLD has different values along the path and the LSR that is inserting <ELI, EL> pairs has no limit on how many pairs it can insert, and it knows the appropriate positions in the stack where they should be inserted, then this option is the same as the recommended solution in Section 4.

A variant of this solution was selected which balances the number of labels that need to be pushed against the requirement for entropy.

## 6. Acknowledgements

The authors would like to thank John Drake, Loa Andersson, Curtis Villamizar, Greg Mirsky, Markus Jork, Kamran Raza and Nobo Akiya for their review comments and suggestions.

## 7. IANA Considerations

This memo includes no request to IANA.

## 8. Security Considerations

This document does not introduce any new security considerations beyond those already listed in [RFC6790].

## 9. References

### 9.1. Normative References

[I-D.filsfils-spring-segment-routing]

Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-filsfils-spring-segment-routing-04 (work in progress), July 2014.

[I-D.gredler-spring-mpls]

Gredler, H., Rekhter, Y., Jalil, L., Kini, S., and X. Xu, "Supporting Source/Explicitly Routed Tunnels via Stacked LSPs", draft-gredler-spring-mpls-06 (work in progress), May 2014.

[I-D.ravisingh-mpls-el-for-seamless-mpls]

Singh, R., Shen, Y., and J. Drake, "Entropy label for seamless MPLS", draft-ravisingh-mpls-el-for-seamless-mpls-04 (work in progress), October 2014.

- [I-D.xu-isis-mpls-elc]  
Xu, X., Kini, S., Sivabalan, S., Filsfils, C., and S. Litkowski, "Signaling Entropy Label Capability Using IS-IS", draft-xu-isis-mpls-elc-01 (work in progress), September 2014.
- [I-D.xu-ospf-mpls-elc]  
Xu, X., Kini, S., Sivabalan, S., Filsfils, C., and S. Litkowski, "Signaling Entropy Label Capability Using OSPF", draft-xu-ospf-mpls-elc-01 (work in progress), October 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, October 2005.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012.

## 9.2. 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.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-03 (work in progress), October 2014.
- [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-04 (work in progress), February 2015.
- [RFC7325] Villamizar, C., Kompella, K., Amante, S., Malis, A., and C. Pignataro, "MPLS Forwarding Compliance and Performance Requirements", RFC 7325, August 2014.

Authors' Addresses

Sriganesh Kini (editor)  
Ericsson

Email: sriganesh.kini@ericsson.com

Kireeti Kompella  
Juniper

Email: kireeti@juniper.net

Siva Sivabalan  
Cisco

Email: msiva@cisco.com

Stephane Litkowski  
Orange

Email: stephane.litkowski@orange.com

Rob Shakir  
B.T.

Email: rob.shakir@bt.com

Xiaohu Xu  
Huawei

Email: xuxiaohu@huawei.com

Wim Hendrickx  
Alcatel-Lucent

Email: wim.henderickx@alcatel-lucent.com

Jeff Tantsura  
Ericsson

Email: jeff.tantsura@ericsson.com

spring  
Internet-Draft  
Intended status: Informational  
Expires: September 10, 2015

N. Kumar  
C. Pignataro  
N. Akiya  
Cisco Systems, Inc.  
R. Geib  
Deutsche Telekom  
G. Mirsky  
Ericsson  
S. Litkowski  
Orange  
March 9, 2015

OAM Requirements for Segment Routing Network  
draft-kumar-spring-sr-oam-requirement-03

Abstract

This document describes a list of functional requirement for OAM in Segment Routing (SR) based network.

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 10, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect



to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction . . . . .	2
2. Requirements notation . . . . .	2
3. Terminology . . . . .	2
4. Detailed Requirement list . . . . .	3
5. IANA Considerations . . . . .	4
6. Security Considerations . . . . .	4
7. Acknowledgement . . . . .	4
8. Contributing Authors . . . . .	5
9. References . . . . .	5
9.1. Normative References . . . . .	5
9.2. Informative References . . . . .	5
Authors' Addresses . . . . .	6

1. Introduction

[I-D.ietf-spring-segment-routing] introduces and explains Segment Routing architecture that leverages source routing and tunneling standards which can be applied directly to MPLS dataplane with no changes on forwarding plane and on IPv6 dataplane with new Routing Extension Header.

This document list the OAM requirements for Segment Routing based network which can further be used to produce OAM tools, either through enhancing existing OAM tools or constructing new OAM tools, for path liveliness and service validation.

2. Requirements notation

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].

3. Terminology

- SR OAM Packet: OAM probe originated and processed within SR domain(s)
- ECMP: Equal Cost Multipath
- SR: Segment Routing
- UCMP: Unequal Cost Multipath

Initiator: Centralized OAM initiator or PMS as referred in [I-D.geib-spring-oam-usecase]

#### 4. Detailed Requirement list

This section list the OAM requirement for Segment Routing based network. The below listed requirement MUST be supported with both MPLS and IPv6 dataplane:

- REQ#1: SR OAM MUST support both On-demand and Continuous OAM functionality.
- REQ#2: The SR OAM packet MUST follow exactly the same path as dataplane traffic.
- REQ#3: The SR OAM packet MUST have the ability to discover and exercise equal cost multipath (ECMP) paths.
- REQ#4: The SR OAM packet MUST have the ability to discover and exercise unequal cost multipath (UCMP) paths.
- REQ#5: The SR OAM packet MUST have ability to exercise any available paths, not just best path available.
- REQ#6: The forwarding semantic of adjacency Segment ID raises a need for additional consideration to detect any failure in forwarding to the right adjacency. SR OAM MUST have the ability to detect any failure in Node SID and adjacency segment based forwarding.
- REQ#7: SR OAM SHOULD have the ability to allow the Initiator to control the return path from any transit or egress responder.
- REQ#8: SR OAM MUST have the ability to be initialized from an arbitrary node to perform connectivity verification and continuity check to any other node within SR domain.
- REQ#9: In case of any failure with continuity check, SR OAM SHOULD support rapid Connectivity Fault localization to isolate the node on which the failure occurs.
- REQ#10: SR OAM SHOULD also have the ability to be initialized from a centralized controller.
- REQ#11: When SR OAM is initialized from centralized controller, it MUST have the ability to alert any edge node in SR domain about the corresponding path or service failure. The node

on receiving the alert MAY take a local protection action or pop an informational message.

- REQ#12: When SR OAM is initialized from centralized controller, it SHOULD support node redundancy. If primary Initiator fails, secondary one MUST take over the responsibility without having any impact on customer traffic.
- REQ#13: SR OAM MUST have the ability to measure Packet loss, Packet Delay or Delay variation using Active (using synthetic probe) and Passive (using data stream) mode.
- REQ#14: When a new path is instantiated, SR OAM SHOULD allow path verification without noticeable delay.
- REQ#15: The above listed requirements SHOULD be supported without any scalability limitation imposed and SHOULD be extensible to accommodate any new SR functionality.
- REQ#16: SR OAM SHOULD minimize the need to create or maintain per path state entry in any other nodes other than the Initiator.
- REQ#17: When traffic engineering is initiated by centralized controller device, and when SR OAM is performed by individual nodes, there MUST be a mechanism to communicate failure to centralized controller device.
- REQ#18: When service instruction is present in SR OAM packet header, there MUST be a method to disallow applying the service to the OAM packet to handle cases where that may result in unintended corruption of the OAM packet.

## 5. IANA Considerations

This document does not propose any IANA consideration.

## 6. Security Considerations

This document list the OAM requirement for Segment Routing network and does not raise any security considerations.

## 7. Acknowledgement

The authors would like to thank Stefano Previdi for his review.

## 8. Contributing Authors

Sriganesh Kini  
Ericsson  
Email: sriganesh.kini@ericsson.com

## 9. References

### 9.1. Normative References

- [I-D.geib-spring-oam-usecase]  
Geib, R., Filsfils, C., Pignataro, C., and N. Kumar, "Use case for a scalable and topology aware MPLS data plane monitoring system", draft-geib-spring-oam-usecase-04 (work in progress), March 2015.
- [I-D.ietf-spring-segment-routing]  
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-ietf-spring-segment-routing-01 (work in progress), February 2015.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

### 9.2. Informative References

- [RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.
- [RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, June 2011.
- [RFC6424] Bahadur, N., Kompella, K., and G. Swallow, "Mechanism for Performing Label Switched Path Ping (LSP Ping) over MPLS Tunnels", RFC 6424, November 2011.
- [RFC6425] Saxena, S., Swallow, G., Ali, Z., Farrel, A., Yasukawa, S., and T. Nadeau, "Detecting Data-Plane Failures in Point-to-Multipoint MPLS - Extensions to LSP Ping", RFC 6425, November 2011.

Authors' Addresses

Nagendra Kumar  
Cisco Systems, Inc.  
7200 Kit Creek Road  
Research Triangle Park, NC 27709  
US

Email: [naikumar@cisco.com](mailto:naikumar@cisco.com)

Carlos Pignataro  
Cisco Systems, Inc.  
7200 Kit Creek Road  
Research Triangle Park, NC 27709-4987  
US

Email: [cpignata@cisco.com](mailto:cpignata@cisco.com)

Nobo Akiya  
Cisco Systems, Inc.  
2000 Innovation Drive  
Kanata, ON K2K 3E8  
Canada

Email: [nobo@cisco.com](mailto:nobo@cisco.com)

Ruediger Geib  
Deutsche Telekom  
Heinrich Hertz Str. 3-7  
Darmstadt 64295  
Germany

Email: [Ruediger.Geib@telekom.de](mailto:Ruediger.Geib@telekom.de)

Greg Mirsky  
Ericsson

Email: [gregory.mirsky@ericsson.com](mailto:gregory.mirsky@ericsson.com)

Stephane Litkowski  
Orange

Email: [stephane.litkowski@orange.com](mailto:stephane.litkowski@orange.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: October 26, 2014

S. Previdi, Ed.  
C. Filsfils, Ed.  
Cisco Systems, Inc.  
B. Decraene  
S. Litkowski  
Orange  
M. Horneffer  
R. Geib  
Deutsche Telekom  
R. Shakir  
British Telecom  
R. Raszuk  
Individual  
April 24, 2014

SPRING Problem Statement and Requirements  
draft-previdi-spring-problem-statement-04

Abstract

The ability for a node to specify a forwarding path, other than the normal shortest path, that a particular packet will traverse, benefits a number of network functions. Source-based routing mechanisms have previously been specified for network protocols, but have not seen widespread adoption. In this context, the term 'source' means 'the point at which the explicit route is imposed'.

This document outlines various use cases, with their requirements, that need to be taken into account by the Source Packet Routing in Networking (SPRING) architecture for unicast traffic. Multicast use-cases and requirements are out of scope of this document.

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 October 26, 2014.

#### Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1. Introduction	3
2. Dataplanes	4
3. IGP-based MPLS Tunneling	4
3.1. Example of IGP-based MPLS Tunnels	4
4. Fast Reroute	5
5. Traffic Engineering	5
5.1. Examples of Traffic Engineering Use Cases	6
5.1.1. Traffic Engineering without Bandwidth Admission Control	6
5.1.2. Traffic Engineering with Bandwidth Admission Control	10
6. Interoperability with non-SPRING nodes	14
7. OAM	14
8. Security	14
9. IANA Considerations	15
10. Manageability Considerations	15
11. Security Considerations	15
12. Acknowledgements	15
13. References	15
13.1. Normative References	15
13.2. Informative References	15
Authors' Addresses	17

## 1. Introduction

The ability for a node to specify a unicast forwarding path, other than the normal shortest path, that a particular packet will traverse, benefits a number of network functions, for example:

- Some types of network virtualization, including multi-topology networks and the partitioning of network resources for VPNs

- Network, link, path and node protection such as fast re-route

- Network programmability

- OAM techniques

- Simplification and reduction of network signaling components

- Load balancing and traffic engineering

Source-based routing mechanisms have previously been specified for network protocols, but have not seen widespread adoption other than in MPLS traffic engineering.

These network functions may require greater flexibility and per packet source imposed routing than can be achieved through the use of the previously defined methods. In the context of this charter, 'source' means 'the point at which the explicit route is imposed'.

In this context, Source Packet Routing in Networking (SPRING) architecture is being defined in order to address the use cases and requirements described in this document.

SPRING architecture should allow incremental and selective deployment without any requirement of flag day or massive upgrade of all network elements.

SPRING architecture should allow optimal virtualization: put policy state in the packet header and not in the intermediate nodes along the path. Hence, the policy is completely virtualized away from midpoints and tail-ends.

SPRING architecture objective is not to replace existing source routing and traffic engineering mechanisms but rather complement them and address use cases where removal of signaling and path state in the core is a requirement.



## 2. Dataplanes

The SPRING architecture should be general in order to ease its applicability to different dataplanes.

MPLS dataplane doesn't require any modification in order to apply a source-based routed model (e.g.: [I-D.filsfils-spring-segment-routing-mpls]).

IPv6 specification [RFC2460], amended by [RFC6564] and [RFC7045], defines the Routing Extension Header which provides IPv6 source-based routing capabilities.

The SPRING architecture should leverage existing MPLS dataplane without any modification and leverage IPv6 dataplane with a new IPv6 Routing Header Type (IPv6 Routing Header is defined in [RFC2460]).

## 3. IGP-based MPLS Tunneling

The source-based routing model, applied to the MPLS dataplane, offers the ability to tunnel services (VPN, VPLS, VPWS) from an ingress PE to an egress PE, with or without the expression of an explicit path and without requiring forwarding plane or control plane state in intermediate nodes.

The source-based routing model, applied to the MPLS dataplane, offers the ability to tunnel unicast services (VPN, VPLS, VPWS) from an ingress PE to an egress PE, with or without the expression of an explicit path and without requiring forwarding plane or control plane state in intermediate nodes. p2mp and mp2mp tunnels are out of the scope of this document.

### 3.1. Example of IGP-based MPLS Tunnels

This section illustrates an example use-case taken from [I-D.filsfils-spring-segment-routing-use-cases].

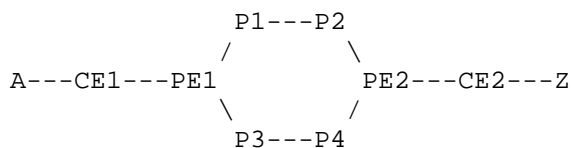


Figure 1: IGP-based MPLS Tunneling

In Figure 1 above, the four nodes A, CE1, CE2 and Z are part of the same VPN. CE2 advertises to PE2 a route to Z. PE2 binds a local label LZ to that route and propagates the route and its label via

MPBGP to PE1 with nhop 192.168.0.2. PE1 installs the VPN prefix Z in the appropriate VRF and resolves the next-hop onto the node segment associated with PE2.

In order to cope with the reality of current deployments, the SPRING architecture should allow PE to PE forwarding according to the IGP shortest path without the addition of any other signaling protocol. The packet each PE forwards across the network will contain (within their label stack) the necessary information derived from the topology database in order to deliver the packet to the remote PE.

#### 4. Fast Reroute

FRR technologies have been deployed by network operators in order to cope with link or node failures through pre-computation of backup paths.

The SPRING architecture should address following requirements:

- o support of FRR on any topology
- o pre-computation and setup of backup path without any additional signaling (other than the regular IGP/BGP protocols)
- o support of shared risk constraints
- o support of node and link protection
- o support of microloop avoidance

Further illustrations of the problem statement for FRR are to be found in [I-D.francois-spring-resiliency-use-case].

#### 5. Traffic Engineering

Traffic Engineering has been addressed using IGP protocol extensions (for resources information propagation) and RSVP-TE for signaling explicit paths. Different contexts and modes have been defined (single vs. multiple domains, with or without bandwidth admission control, centralized vs. distributed path computation, etc).

In all cases, one of the major components of the TE architecture is the soft state based signaling protocol (RSVP-TE) which is used in order to signal and establish the explicit path. Each path, once computed, need to be signaled and state for each path must be present in each node traversed by the path. This incurs a scalability problem especially in the context of SDN where traffic differentiation may be done at a finer granularity (e.g.: application

specific). Also the amount of state needed to be maintained and periodically refreshed in all involved nodes contributes significantly to complexity and the number of failures cases, and thus increases operational effort while decreasing overall network reliability.

The source-based routing model allows traffic engineering to be implemented without the need of a signaling component.

The SPRING architecture should support traffic engineering, including:

- o loose or strict options
- o bandwidth admission control
- o distributed vs. centralized model (PCE, SDN Controller)
- o disjointness in dual-plane networks
- o egress peering traffic engineering
- o load-balancing among non-parallel links
- o Limiting (scalable, preferably zero) per-service state and signaling on midpoint and tail-end routers.
- o ECMP-awareness
- o node resiliency property (i.e.: the traffic-engineering policy is not anchored to a specific core node whose failure could impact the service.

#### 5.1. Examples of Traffic Engineering Use Cases

As documented in [I-D.filsfils-spring-segment-routing-use-cases] here follows the description of two sets of use cases:

- o Traffic Engineering without Admission Control
- o Traffic Engineering with Admission Control

##### 5.1.1. Traffic Engineering without Bandwidth Admission Control

In this section, we describe Traffic Engineering use-cases without bandwidth admission control.

## 5.1.1.1. Disjointness in dual-plane networks

Many networks are built according to the dual-plane design, as illustrated in Figure 2:

Each access region  $k$  is connected to the core by two  $C$  routers ( $C(1,k)$  and  $C(2,k)$ ).

$C(1,k)$  is part of plane 1 and aggregation region  $K$

$C(2,k)$  is part of plane 2 and aggregation region  $K$

$C(1,k)$  has a link to  $C(2, j)$  iff  $k = j$ .

The core nodes of a given region are directly connected.  
Inter-region links only connect core nodes of the same plane.

$\{C(1,k) \text{ has a link to } C(1, j)\}$  iff  $\{C(2,k) \text{ has a link to } C(2, j)\}$ .

The distribution of these links depends on the topological properties of the core of the AS. The design rule presented above specifies that these links appear in both core planes.

We assume a common design rule found in such deployments: the inter-plane link costs ( $C_{ik}-C_{jk}$  where  $i \neq j$ ) are set such that the route to an edge destination from a given plane stays within the plane unless the plane is partitioned.

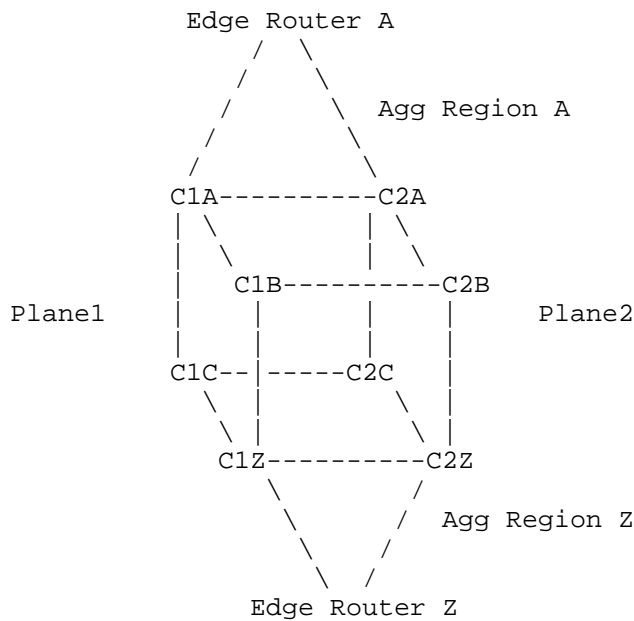


Figure 2: Dual-Plane Network and Disjointness

In this scenario, the operator requires the ability to deploy different strategies. For example, A should be able to use the three following options:

- o the traffic is load-balanced across any ECMP path through the network
- o the traffic is load-balanced across any ECMP path within the Plane1 of the network
- o the traffic is load-balanced across any ECMP path within the Plane2 of the network

Most of the data traffic from A to Z would use the first option, such as to exploit the capacity efficiently. The operator would use the two other choices for specific premium traffic that has requested disjoint transport.

The SPRING architecture should support this use case with the following requirements:

- o Zero per-service state and signaling on midpoint and tail-end routers.

- o ECMP-awareness.
- o Node resiliency property: the traffic-engineering policy is not anchored to a specific core node whose failure could impact the service.

5.1.1.2. Egress Peering Traffic Engineering

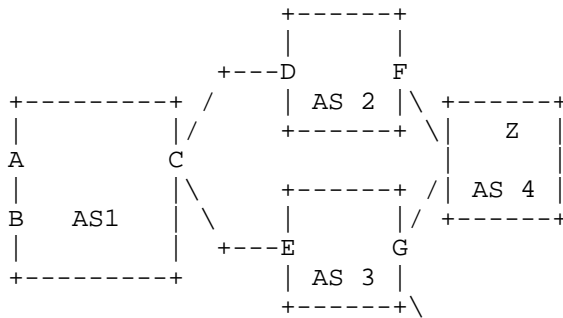


Figure 3: Egress peering traffic engineering

Let us assume, in the network depicted in Figure 3, that:

C in AS1 learns about destination Z of AS 4 via two BGP paths (AS2, AS4) and (AS3, AS4).

C may or may not be configured so to enforce next-hop-self behavior before propagating the paths within AS1.

C may propagate all the paths to Z within AS1 (add-path).

C may install in its FIB only the route via AS2, or only the route via AS3, or both.

In that context, SPRING should allow the operator of AS1 to apply the following traffic-engineering policy, regardless the configured behavior of next-hop-self:

Steer 60% of the Z-destined traffic received at A via AS2 and 40% via AS3.

Steer 80% of the Z-destined traffic received at B via AS2 and 20% via AS3.

While egress routers are known in the routing domain (generally through their loopback address), the SPRING architecture should enable following:

- o identify the egress interfaces of an egress node
- o identify the peering neighbors of an egress node
- o identify the peering ASes of an egress node

With these identifiers known in the domain, the SPRING architecture should allow an ingress node to select the exit point of a packet as any combination of an egress node, an egress interface, a peering neighbor, and a peering AS.

#### 5.1.1.3. Load-balancing among non-parallel links

The SPRING architecture should allow a given node should be able to load share traffic across multiple non parallel links even if these ones lead to different neighbors. This may be useful to support traffic engineering policies.

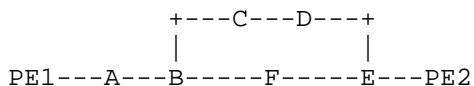


Figure 4: Multiple (non-parallel) Adjacencies

In the above example, the operator requires PE1 to load-balance its PE2-destined traffic between the ABCDE and ABFE paths.

#### 5.1.2. Traffic Engineering with Bandwidth Admission Control

The implementation of bandwidth admission control within a network (and its possible routing consequence which consists in routing along explicit paths where the bandwidth is available) requires a capacity planning process.

The spreading of load among ECMP paths is a key attribute of the capacity planning processes applied to packet-based networks.

##### 5.1.2.1. Capacity Planning Process

Capacity Planning anticipates the routing of the traffic matrix onto the network topology, for a set of expected traffic and topology variations. The heart of the process consists in simulating the placement of the traffic along ECMP-aware shortest-paths and accounting for the resulting bandwidth usage.

The bandwidth accounting of a demand along its shortest-path is a basic capability of any planning tool or PCE server.

For example, in the network topology described below, and assuming a default IGP metric of 1 and IGP metric of 2 for link GF, a 1600Mbps A-to-Z flow is accounted as consuming 1600Mbps on links AB and FZ, 800Mbps on links BC, BG and GF, and 400Mbps on links CD, DF, CE and EF.

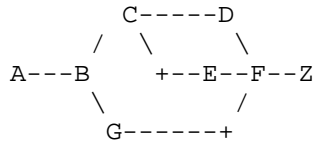


Figure 5: Capacity Planning an ECMP-based demand

ECMP is extremely frequent in SP, Enterprise and DC architectures and it is not rare to see as much as 128 different ECMP paths between a source and a destination within a single network domain. It is a key efficiency objective to spread the traffic among as many ECMP paths as possible.

This is illustrated in the below network diagram which consists of a subset of a network where already 5 ECMP paths are observed from A to M.

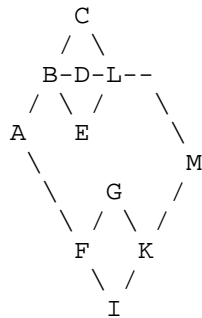


Figure 6: ECMP Topology Example

When the capacity planning process detects that a traffic growth scenario and topology variation would lead to congestion, a capacity increase is triggered and if it cannot be deployed in due time, a traffic engineering solution is activated within the network.

A basic traffic engineering objective consists of finding the smallest set of demands that need to be routed off their shortest path to eliminate the congestion, then to compute an explicit path for each of them and instantiating these traffic-engineered policies in the network.



SPRING architecture should offer a simple support for ECMP-based shortest path placement as well as for explicit path policy without incurring additional signaling in the domain. This includes:

- o the ability to steer a packet across a set of ECMP paths
- o the ability to diverge from a set of ECMP shortest paths to one or more paths not in the set of shortest paths

#### 5.1.2.2. SDN/SR use-case

The SDN use-case lies in the SDN controller, (e.g.: Stateful PCE as described in [I-D.ietf-pce-stateful-pce]).

The SDN controller is responsible to control the evolution of the traffic matrix and topology. It accepts or denies the addition of new traffic into the network. It decides how to route the accepted traffic. It monitors the topology and upon topological change, determines the minimum traffic that should be rerouted on an alternate path to alleviate a bandwidth congestion issue.

The algorithms supporting this behavior are a local matter of the SDN controller and are outside the scope of this document.

The means of collecting traffic and topology information are the same as what would be used with other SDN-based traffic-engineering solutions (e.g. [RFC7011] and [I-D.ietf-idr-ls-distribution]).

The means of instantiating policy information at a traffic-engineering head-end are the same as what would be used with other SDN-based traffic-engineering solutions (e.g.: [I-D.ietf-i2rs-architecture], [I-D.crabbe-pce-pce-initiated-lsp] and [I-D.sivabalan-pce-segment-routing]).

In the context of Centralized-Based Optimization and the SDN use-case, here are the benefits that the SPRING architecture should deliver:

Explicit routing capability with or without ECMP-awareness.

No signaling hop-by-hop through the network.

State is only maintained at the policy head-end. No state is maintained at mid-points and tail-ends.

Automated guaranteed FRR for any topology.

Optimum virtualization: the policy state is in the packet header and not in the intermediate nodes along the path. The policy is completely virtualized away from midpoints and tail-ends.

Highly responsive to change: the SDN Controller only needs to apply a policy change at the head-end. No delay is introduced due to programming the midpoints and tail-end along the path.

#### 5.1.2.2.1. SDN Example

The data-set consists in a full-mesh of 12000 explicitly-routed tunnels observed on a real network. These tunnels resulted from distributed headend-based CSPF computation.

We measured that only 65% of the traffic is forwarded over its shortest path.

Three well-known defects are illustrated in this data set:

The lack of ECMP support in explicitly routed tunnels: ATM-alike traffic-steering mechanisms steer the traffic along a non-ECMP path.

The increase of the number of explicitly-routed non-ECMP tunnels to enumerate all the ECMP options.

The inefficiency of distributed optimization: too much traffic is forwarded off its shortest path.

We applied the SDN use-case to this dataset implying a source route model where the path of the packet is encoded within the packet itself. This means that:

The distributed CSPF computation is replaced by centralized optimization and BW admission control, supported by the SDN Controller.

As part of the optimization, we also optimized the IGP-metrics such as to get a maximum of traffic load-spread among ECMP paths by default.

The traffic-engineering policies are supported by a source route model (e.g.: [I-D.filsfils-rtgwg-segment-routing]).

As a result, we measured that 98% of the traffic would be kept on its normal policy (over the shortest-path) and only 2% of the traffic requires a path away from the shortest-path.

Let us highlight a few benefits:

98% of the traffic-engineering head-end policies are eliminated.

Indeed, by default, an ingress edge node capable of injecting source routed packets steers the traffic to the egress edge node. No configuration or policy needs to be maintained at the ingress edge node to realize this.

100% of the states at mid/tail nodes are eliminated.

#### 6. Interoperability with non-SPRING nodes

SPRING must inter-operate with non-SPRING nodes.

An illustration of interoperability between SPRING and other MPLS Signalling Protocols (LDP) is described here in [I-D.filsfils-spring-segment-routing-ldp-interop].

Interoperability with IPv6 non-SPRING nodes will be described in a future document.

#### 7. OAM

The SPRING WG should provide OAM and the management needed to manage SPRING enabled networks. The SPRING procedures may also be used as a tool for OAM in SPRING enabled networks.

OAM use cases and requirements are described in [I-D.geib-spring-oam-usecase] and [I-D.kumar-spring-sr-oam-requirement].

#### 8. Security

There is an assumed trust model such that any node imposing an explicit route on a packet is assumed to be allowed to do so. In such context trust boundaries should strip explicit routes from a packet.

For each data plane technology that SPRING specifies, a security analysis must be provided showing how protection is provided against an attacker disrupting the network by for example, maliciously injecting SPRING packets.

## 9. IANA Considerations

TBD

## 10. Manageability Considerations

TBD

## 11. Security Considerations

TBD

## 12. Acknowledgements

The authors would like to thank Yakov Rekhter for his contribution to this document.

## 13. References

## 13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, April 2012.
- [RFC7011] Claise, B., Trammell, B., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, September 2013.
- [RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, December 2013.

## 13.2. Informative References

- [I-D.crabbe-pce-pce-initiated-lsp] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-crabbe-pce-pce-initiated-lsp-03 (work in progress), October 2013.

## [I-D.filsfils-rtgwg-segment-routing]

Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-filsfils-rtgwg-segment-routing-01 (work in progress), October 2013.

## [I-D.filsfils-spring-segment-routing-ldp-interop]

Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing interoperability with LDP", draft-filsfils-spring-segment-routing-ldp-interop-01 (work in progress), April 2014.

## [I-D.filsfils-spring-segment-routing-mpls]

Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing with MPLS data plane", draft-filsfils-spring-segment-routing-mpls-01 (work in progress), April 2014.

## [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-00 (work in progress), March 2014.

## [I-D.francois-spring-resiliency-use-case]

Francois, P., Filsfils, C., Decraene, B., and R. Shakir, "Use-cases for Resiliency in SPRING", draft-francois-spring-resiliency-use-case-02 (work in progress), April 2014.

## [I-D.geib-spring-oam-usecase]

Geib, R. and C. Filsfils, "Use case for a scalable and topology aware MPLS data plane monitoring system", draft-geib-spring-oam-usecase-01 (work in progress), February 2014.

## [I-D.ietf-i2rs-architecture]

Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", draft-ietf-i2rs-architecture-02 (work in progress), February 2014.

## [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-04 (work in progress), November 2013.

## [I-D.ietf-pce-stateful-pce]

Crabbe, E., Medved, J., Minei, I., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-08 (work in progress), February 2014.

## [I-D.kumar-spring-sr-oam-requirement]

Kumar, N., Pignataro, C., Akiya, N., Geib, R., and G. Mirsky, "OAM Requirements for Segment Routing Network", draft-kumar-spring-sr-oam-requirement-00 (work in progress), February 2014.

## [I-D.sivabalan-pce-segment-routing]

Sivabalan, S., Medved, J., Filsfils, C., Crabbe, E., and R. Raszuk, "PCEP Extensions for Segment Routing", draft-sivabalan-pce-segment-routing-02 (work in progress), October 2013.

## Authors' Addresses

Stefano Previdi (editor)  
Cisco Systems, Inc.  
Via Del Serafico, 200  
Rome 00142  
Italy

Email: sprevidi@cisco.com

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

Email: cfilsfil@cisco.com

Bruno Decraene  
Orange  
FR

Email: bruno.decraene@orange.com

Stephane Litkowski  
Orange  
FR

Email: stephane.litkowski@orange.com

Martin Horneffer  
Deutsche Telekom  
Hammer Str. 216-226  
Muenster 48153  
DE

Email: Martin.Horneffer@telekom.de

Ruediger Geib  
Deutsche Telekom  
Heinrich Hertz Str. 3-7  
Darmstadt 64295  
DE

Email: Ruediger.Geib@telekom.de

Rob Shakir  
British Telecom  
London  
UK

Email: rob.shakir@bt.com

Robert Raszuk  
Individual

Email: robert@raszuk.net

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: September 8, 2016

X. Xu  
Huawei  
R. Raszuk  
Bloomberg LP  
U. Chunduri  
Ericsson  
L. Contreras  
Telefonica I+D  
L. Jalil  
Verizon  
March 7, 2016

Connecting MPLS-SPRING Islands over IP Networks  
draft-xu-spring-islands-connection-over-ip-05

Abstract

MPLS-SPRING is an MPLS-based source routing paradigm in which a sender of a packet is allowed to partially or completely specify the route the packet takes through the network by imposing stacked MPLS labels to the packet. To facilitate the incremental deployment of this new technology, this document describes a mechanism which allows the outermost LSP be replaced by an IP-based tunnel.

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 8, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.



This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Requirements Language . . . . .	3
2. Terminology . . . . .	3
3. Packet Forwarding Procedures . . . . .	3
4. Acknowledgements . . . . .	4
5. IANA Considerations . . . . .	4
6. Security Considerations . . . . .	4
7. References . . . . .	4
7.1. Normative References . . . . .	4
7.2. Informative References . . . . .	4
Authors' Addresses . . . . .	5

## 1. Introduction

MPLS-SPRING [I-D.ietf-spring-segment-routing-mpls] is a MPLS-based source routing paradigm in which a sender of a packet is allowed to partially or completely specify the route the packet takes through the network by imposing stacked MPLS labels to the packet. To facilitate the incremental deployment of this new technology, this document describes a mechanism which allows the outermost LSP to be replaced by an IP-based tunnel (e.g., MPLS-in-IP/GRE tunnel [RFC4023], MPLS-in-UDP tunnel [RFC7510] or MPLS-in-L2TPv3 tunnel [RFC4817] and etc) when the nexthop along the LSP is not MPLS-SPRING-enabled. The tunnel destination address would be the address of the egress of the outmost LSP (e.g., the egress of the active segment).

This mechanism is much useful in the MPLS-SPRING-based Service Function Chaining (SFC) case [I-D.xu-sfc-using-mpls-spring] where only a few specific routers (e.g., Service Function Forwarders (SFF) and classifiers) are required to be MPLS-SPRING-capable while the remaining routers are just required to support IP forwarding capability. In addition, this mechanism is also useful in some specific Traffic Engineering scenarios where only a few routers (e.g., the entry and exit nodes of each plane in the dual-plane network ) are specified as segments of explicit paths. In this way, only a few routers are required to support the MPLS-SPRING capability

while all the other routers just need to support IP forwarding capability, which would significantly reduce the deployment cost of this new technology. Furthermore, since there is no need to run any other label distribution protocol (e.g., LDP), the network provisioning is greatly simplified, which is one of the major claimed benefits of the MPLS-SPRING technology.

### 1.1. 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].

## 2. Terminology

This memo makes use of the terms defined in [RFC3031], [I-D.ietf-spring-segment-routing-mpls] and [I-D.xu-sfc-using-mpls-spring] .

## 3. Packet Forwarding Procedures

Assume an MPLS-SPRING-enabled router X prepares to forward an MPLS packet to the next segment (i.e., the node segment of MPLS-SPRING-enabled router Y) which is identified by the top label of the MPLS packet. If the next-hop router of the best path to Y is a non-MPLS router, X couldn't map the packet's top label into an Next Hop Label Forwarding Entry (NHLFE) , even though the top label itself is a valid incoming label. If the label is not a Penultimate Hop Popping (PHP) label (i.e., the NP-flag [I-D.ietf-isis-segment-routing-extensions] associated with the corresponding prefix SID of that top label is set), X SHOULD swap the top label to the corresponding label significant to Y and then encapsulate the MPLS packet into an IP-based tunnel. The tunnel destination address is the IP address of Y (e.g., the /32 or /128 prefix FEC associated with that top label) and the tunnel source address is the IP address of X. If the top label is a PHP label and not at the bottom of the label stack, X SHOULD pop that top label before performing the above encapsulation. The IP encapsulated packet would be forwarded according to the IP forwarding table. Upon receipt of that IP encapsulated packet, Y would decapsulate it and then process the decapsulated MPLS packet accordingly.

As for which tunnel encapsulation type should be used by X, it can be manually specified on X or learnt from Y's advertisement of its tunnel encapsulation capability. How to advertise the tunnel encapsulation capability using IS-IS or OSPF are specified in [I-D.xu-isis-encapsulation-cap] and [I-D.ietf-ospf-encapsulation-cap] respectively.

#### 4. Acknowledgements

Thanks Joel Halpern, Bruno Decraene and Loa Andersson for their insightful comments on this draft.

#### 5. IANA Considerations

No action is required for IANA.

#### 6. Security Considerations

TBD.

#### 7. References

##### 7.1. Normative References

- [I-D.ietf-spring-segment-routing-mpls]  
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J., and E. Crabbe, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-03 (work in progress), February 2016.
- [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>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<http://www.rfc-editor.org/info/rfc3031>>.

##### 7.2. Informative References

- [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-06 (work in progress), December 2015.
- [I-D.ietf-ospf-encapsulation-cap]  
Xu, X., Decraene, B., Raszuk, R., Chunduri, U., Contreras, L., and L. Jalil, "Advertising Tunnelling Capability in OSPF", draft-ietf-ospf-encapsulation-cap-00 (work in progress), October 2015.

- [I-D.xu-isis-encapsulation-cap]  
Xu, X., Decraene, B., Raszuk, R., Chunduri, U., Contreras, L., and L. Jalil, "Advertising Tunnelling Capability in IS-IS", draft-xu-isis-encapsulation-cap-06 (work in progress), November 2015.
- [I-D.xu-sfc-using-mpls-spring]  
Xu, X., Li, Z., Shah, H., and L. Contreras, "Service Function Chaining Using MPLS-SPRING", draft-xu-sfc-using-mpls-spring-04 (work in progress), September 2015.
- [RFC4023] Worster, T., Rekhter, Y., and E. Rosen, Ed., "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, DOI 10.17487/RFC4023, March 2005, <<http://www.rfc-editor.org/info/rfc4023>>.
- [RFC4817] Townsley, M., Pignataro, C., Wainner, S., Seely, T., and J. Young, "Encapsulation of MPLS over Layer 2 Tunneling Protocol Version 3", RFC 4817, DOI 10.17487/RFC4817, March 2007, <<http://www.rfc-editor.org/info/rfc4817>>.
- [RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <<http://www.rfc-editor.org/info/rfc7510>>.

## Authors' Addresses

Xiaohu Xu  
Huawei

Email: [xuxiaohu@huawei.com](mailto:xuxiaohu@huawei.com)

Robert Raszuk  
Bloomberg LP

Email: [robert@raszuk.net](mailto:robert@raszuk.net)

Uma Chunduri  
Ericsson

Email: [uma.chunduri@ericsson.com](mailto:uma.chunduri@ericsson.com)

Luis M. Contreras  
Telefonica I+D  
Ronda de la Comunicacion, s/n  
Sur-3 building, 3rd floor  
Madrid, 28050  
Spain

Email: [luismiguel.contrerasmurillo@telefonica.com](mailto:luismiguel.contrerasmurillo@telefonica.com)  
URI: <http://people.tid.es/LuisM.Contreras/>

Luay Jalil  
Verizon

Email: [luay.jalil@verizon.com](mailto:luay.jalil@verizon.com)