

Internet Research Task Force  
Internet Draft  
Intended status: Informational  
Expires: February 03, 2014

P. Ashwood-Smith  
Huawei  
M. Soliman  
Carleton University  
T. Wan  
Huawei  
July 03, 2013

## SDN State Reduction

draft-ashwood-sdnrg-state-reduction-00.txt

### Abstract

This document makes the argument that to support the centralized control of a substantial number of forwarding devices (as Software Defined Networking (SDN) proposes) that the scale, speed, cost and general quality of such a solution will be improved by reducing the state needed to be distributed into the network of devices by the controller(s). To this end we re-visit forms of Source Routing (SR), in particular Strict Link Source Routing (SLSR) and suggest that light weight SLSR could allow substantial reduction in controller burden while potentially reducing the costs/complexity on forwarding devices. We discuss some simulation results that demonstrate these advantages and how the advantages grow substantially as the network diameter grows. We also look at various implementation possibilities including existing IPV4, V6, MPLS, new/modified MPLS vs. something brand new that could possibly be implemented with new SDN technology like Protocol Oblivious Forwarding-POF.

### 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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

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

This Internet-Draft will expire on December 3, 2012.

#### 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.

#### Table of Contents

1. Terminology	3
2. Introduction	3
3. Logical Example	5
4. Expressing a Path	6
5. Computing a Path	7
6. Downloading Forwarding State	8
7. Logically Forwarding SLSR	10
7.1. Ingress Logical Unicast Forwarding	10
7.2. Tandem Logical Unicast Forwarding	11
7.3. Egress Logical Unicast Forwarding	12
8. Logical Multicast Forwarding SLSR Packets	13
9. Failure Recovery	14
10. Comparison of Logical Model to Existing Source Routing	15
10.1. MPLS as a SLSR	15
10.2. IPV4/6 Options as SLSR	18
10.3. Protocol Oblivious Forwarding as SLSR mechanism	19
11. Security Considerations	20
12. Conclusions and Future work	21
13. IANA Considerations	21
14. References	21
14.1. Informative References	21
15. Authors' Addresses	23
16. Contributors	23
17. Acknowledgements	23

## 1. Terminology

ATM	Asynchronous Transfer Mode (a cell based network)
BGP	Boarder Gateway Protocol
CSPF	Constrained Shortest Path First
DOS	Denial of Service (attack)
ECMP	Equal Cost Multi Path
flow	Logically related packets following the same path
IS-IS	Intermediate System to Intermediate System
LACP	Link Aggregation Control Protocol
LAG	Link Aggregation
Loose	A source route that enumerates only some of all hops
MPLS	Multi Protocol Label Switching
MPLS-TE	MPLS Traffic Engineering.
NPU	Network Processor Unit (programmable forwarding)
OpenFlow	Open data path programming protocol
OSPF	Open Shortest Path First
PCE	Path Computation Element (used with MPLS-TE)
PNNI	Private Network to Network Interface (link state ATM)
POF	Protocol Oblivious Forwarding - more generic OpenFlow)
RSVP-TE	Resource Reservation Protocol - Traffic Engineering
SDN	Software Defined Networking (as per [OPENFLOW])
SDN-domain	Set of forwarding devices controlled as a unit.
SR	Source Routing - enumerating hops to traverse
SLSR	Strict Link Source Routing - enumerating links [SLSR]
SPF	Shortest Path First - (Dijkstra etc.)
SSRR	Strict Source Record Route - IPV4 header option 9
Strict	Source route that enumerates every hop(unlike Loose)
TE	Traffic Engineering
VFI	Virtual Forwarding Instance (layer 2)
VPLS	Virtual Private Lan Service
VRF	Virtual Routing and Forwarding (layer 3)

## 2. Introduction

The centralized control of a network is not a new idea. Indeed centralized control was widely deployed in voice networks and some early data networks but of course gave way to distributed control for IP.

Centralized computation is however still widely used for traffic engineered networks, like MPLS-TE and GMPLS where a Path Computation Engine (PCE) makes use of a global view of a sub-network and its resource usage for the planning of new paths and their resources. The data path state distribution with these models is however not initiated centrally and relies on protocols like RSVP-TE to install the hop by hop state. In fact this form

of distributed control with centralized traffic engineering computations is the norm today.

Notwithstanding the massive deployment of this kind of hybrid distributed/central control, we have in the last several years seen a huge resurgence of interest in fully centralized control of at least a set of forwarding devices [ONF] [OPENFLOW] with Software Defined Networking (SDN). This SDN proposes a central controller (or controllers) using IP protocols such as TCP to talk to a set of arbitrarily interconnected (and cheap/dumb) forwarding devices (SDN-domain) and which is responsible for the configuration of the majority of forwarding state on those devices. This state may be produced either as a result of pro-active configuration, or based on re-active responses to packet flow indications from the forwarding devices themselves.

Since this central controller has knowledge of the entire sub-network of devices, and potentially of the traffic demands into/out of the sub-network, it can perform a variety of path optimization computations similar to CSPF/MPLS-TE/PCE/GMPLS, or even more elaborate forms of optimization (trading flows against each other rather than individually optimizing them, exploiting quiet areas of the network to offload busy areas etc), the output of which is forwarding state for all meta flows in the entire sub-network of devices and a sub network which more optimally meets the desired local constraints. One such deployment reports a substantial increase in network utilization from 30% to 70%-90% [SDNGOOG].

A central controller can also more effectively solve problems such as bin-packing and path blocking [SDNGOOG], which occur when flows are optimized individually with greedy type algorithms rather than considering other orderings of the flows. The finer grained ability to place traffic can also permit much more detailed placement of traffic after a failure, including traffic not directly affected by the failure but the replacement of which is critical to achieving fair/efficient use of the remaining bandwidth subsequent to the failure.

Since the output of the controller is much closer to a TE (Traffic Engineered) type solution from a PCE (Path Control Element) than an SPF (Shortest Path First) solution the controller cannot simply install destination based forwarding entries. A controller either needs to install tunnels that follow the explicit routes it wishes and then map traffic to those tunnels at the edges, or it must install n-tuple < <source IP> <destination IP> <source port> <destination port> etc.> state and configure these n-tuple matches on every hop along the desired path. Packets which fail to match an n-tuple are either discarded or sent to the controller.

In the normal case of SDN (as given in [OPENFLOW]) the controller is required to send configuration information to all devices along the path from ingress of the SDN-domain of this controller to the egress of that SDN-domain, alternatively a tunnel setup protocol like RSVP-TE is required to be triggered to distribute the per hop state between the ingress and egress.

This draft proposes that since the controller knows the exact end-to-end path (down to the level of the links it wishes the packets to traverse) and that the diameter of an SDN-domain is likely to be a reasonable number of hops, that the controller should instead simply insert into a header the exact links it wishes the packet to traverse and thereby not have to deal either with per hop n-tuple state installation (very expensive) or with MPLS tunnel installation via RSVP-TE (complex). Such a mechanism also eliminates any concerns about Equal Cost Multi Path (ECMP) and/or Link Aggregation (LAG) as the controller can place traffic on exact links.

Operations, Administration and Management (OAM) is also greatly simplified since data packets will flow on invariant paths that are known by both ends of the flow and can be the same as any OAM packets that probe the flow. This OAM "fate sharing" property is widely valued by network operators and considerable effort has already been expended to permit similar fate sharing between OAM and data paths with other carrier scale networking protocols such as 802.1ag and MPLS-TP. Of course if a controller does not wish to enforce symmetry and congruence it need not.

### 3. Logical Example

The following is an example of an idealized strict link based source routing (SLSR) forwarding. We talk about possible implementations including MPLS methods after looking at the logical ideal.

Consider the simple 7 node network shown in Figure 1 below. Here the nodes are named {A, B, C, D, E, F, G} and where each node has locally numbered interfaces named {1, 2, 3, 4, 5, 6}.

For example node A has interfaces named 1, 2, 3, 4 and where interfaces 4 and 2 both go to node B. Node B has local interfaces 1, 2, 3, 4 and 5 but the two interfaces going back to node A are locally named 1 and 3. Clearly node interface names are likely (but not necessarily) different at both ends of a link.

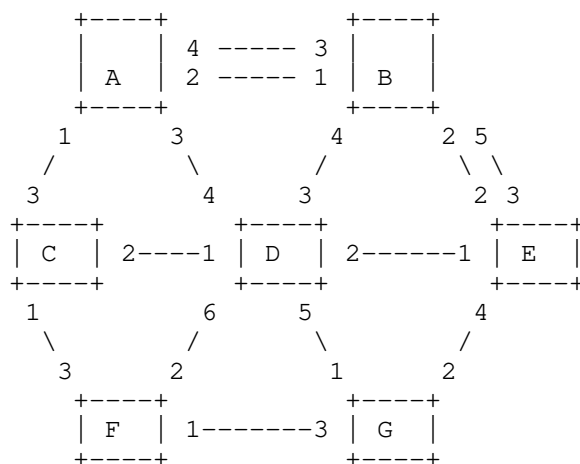


Figure 1 - simple 7 node network with local link identifiers.

#### 4. Expressing a Path

A path through a network labeled as per Figure 1 can clearly be expressed as a sequence of link names (an SLSR).

For example, between nodes C and E the following are all valid paths.

```
C.3 -> A.2 -> B.2
C.2 -> D.2
C.1 -> F.2 -> D.2
C.3 -> A.4 -> B.5
```

Now since the links lead unambiguously to a known node, the paths can be more compactly expressed without the node names as follows:

```
{3,2,2}
{2,2}
{1,2,2}
{3,4,5}
```

As long as we know the origin of the path (in this case node C), the list of link names unambiguously identifies a path and an egress point. In addition it identifies unambiguously which link from among parallel links between neighbors should be traversed. Of course it is possible to give a name to the set of links that all attach to the same neighbor and thereby leave the exact link in that path deliberately ambiguous and thereby subject to a local forwarding decision as to exactly which link in the set to follow.

Each path of course also has exactly one perfectly symmetric reverse. Note that the symmetric reverse path is not simply the same list of link names in reverse order. A reverse path has to be specified from the opposite end of the path so in this example the origin has to be E.

The forward and corresponding reverse paths are therefore.

C->E	E->C
{3,2,2}	{2,1,1}
{2,2}	{1,1}
{1,2,2}	{1,6,3}
{3,4,5}	{3,3,1}

Various very efficient encodings of these kinds of paths in source routed headers are possible. Even a simple encoding using 8 bits per hop can encode every path in a large 8 hop network with fewer bits than an IP in IP tunnel.

## 5. Computing a Path

It should be obvious that the output of any graph based computation which has as its goal various optimization criteria for flows can express its results as a series of such paths where each path is expressed as a Strict Link based Source Route (SLSR). This includes multiple different metric Dijkstra computations (i.e. shortest path, multi topology shortest path), CSPF type and of course more elaborate linear-programming or other convex type optimizations.

The expression of the path as an SLSR imposes no constraints on the type of computation being performed except possibly in path length. However in any real network under the control of a single controller it is not likely that path length would be a real issue unless perhaps unreasonably large link names are encoded.

Convex and linear-programming type solutions to traffic placement are of particular interest because to do a good job they must exploit a considerable number of paths through a network (many more than shortest). These algorithms take the matrix of ingress/egress flows in a network together with all the usable paths between all sources and destinations and will assign percentages of the ingress/egress flows to the available paths in ratios that can optimize a number of simultaneous constraints. For example they can optimize the network's total throughput, the average link utilization, the fairness of the bandwidth available to each flow and can even optimize different linear and non linear combinations of those goals. What is interesting about all of

these kinds of optimizations however is that they need access to all of the reasonable paths across the network since it is by making trade-offs between busy and less busy parts of the network that they achieve their goals. Unfortunately the number of paths (shortest or otherwise) in a network grows exponentially with network size and with it the state distribution problem (or burden) the controller must deal with.

It is important to make the distinction between a flow and a path. This draft concerns itself not with the immense numbers of micro flows but with the very large numbers of paths required to be supported onto which those micro flows are then aggregated. A set of micro flows can be treated as a single flow, and a single flow has a unique path through the network.

## 6. Downloading Forwarding State

A controller likely takes as input the fields that identify the flow and its various statistical attributes. The controller then likely computes an end to end path for this flow either based on the single flow's attributes (in a re-active manner), or on more global knowledge of multiple flow attributes (in a pro-active manner). Flows may be meta (many micro flows) or individual micro flows depending on the implementation and its scale. The output of course is just a list of links that must be traversed for this flow together with matching rules to identify the flow at the ingress.

The controller then delivers the flow matching rules and the Strict Link Based Source Route to the *\*single\** node where the flow is to be encapsulated (i.e. where the flow first enters the SDN-domain).

The fact of only having to communicate with the *\*single\** node at the head end of the path means that the controller experiences a reduction in its work load directly proportional to the number of hops in the path (as compared to traditional SDN which must program every hop along the path).

Intuitively this translates to the following I/O burden reduction at the controller based on the number of links that must be traversed per average path.



#Avg Path Len	% I/O Burden Reduction
1	0%
2	50%
3	66%
4	75%
5	80%
..	..
N	$(100-100/n) \%$

Since forwarding state download is typically a substantial part of a "normal" routers' re-convergence time, it seems reasonable that this will become a similar bottleneck for a central controller and quite possibly be further aggravated by the increased delays and larger amount of state that the central device must deal with.

As a result this reduction in state and I/O burden should have a marked impact on convergence times assuming there are appropriate forwarding mechanisms that can implement the Strict Link Source Route (SLSR). Note also that the position of the controller relative to the ingress/egress nodes is now more important than its position relative to all nodes. Therefore studies as to controller optimum placement as defined by the Controller Placement Problem [PLACEMENT] would require different optimization goals.

An additional 50% reduction can also be obtained should the implementation of the forwarding be able to reverse the path on the fly. Such a reversal permits the implicit communication of the desired reverse path to the receiver thereby eliminating communication with the controller to obtain a reverse path. Of course if symmetry is not desired this further optimization is not possible.

For example, consider a network with 1000 nodes. It therefore has  $O(1,000,000)$  meta flows and assuming 10 possible paths for each flow has  $O(10,000,000)$  ingress forwarding entries that must be centrally configured (its burden). If each path on average takes 5 hops then the burden on the controller grows 5 fold to  $O(50,000,000)$  entries but with SLSR the burden remains at  $O(10,000,000)$ . If path reversal is supported and symmetric routing is desired then the burden with SLSR drops further to  $O(5,000,000)$ .

Simulations done by one of us in [SRSDN] provide additional weight to the above arguments. In particular we simulated for various network sizes and diameters the differences between hop by hop SDN and SLSR and saw up to 3 x performance improvements in convergence times with SLSR. There were also a number of other benefits such as a markedly reduced standard deviation in convergence times for the different nodes (81% decreases) and a significantly reduced sensitivity to the placement of the controller (80% reduction in standard deviation). The performance improvements can perhaps better be understood by an analogy comparing the work required to fill in the area of an object (traditional SDN) vs. simply drawing the circumference of that object (SLSR). Since the circumference varies as a function of the diameter but the area varies as a function of the diameter squared the relative burden reduction with just dealing with the circumference (the edge of the network) becomes apparent. In fact in this simulation study we varied the radius and then plotted the relative convergence times of SLSR and traditional hop by hop forwarded SDN and saw a ratio of convergence times as a function of radius that indicated a trend towards  $1/R$  as expected. Simply stated, the bigger the centrally controlled network the better source routing performs compared to hop by hop.

## 7. Logically Forwarding SLSR

There are three distinct phases to be performed to logically forward unicast SLSR. These are similar to any tunnel technology and consist of 1) Ingress Encapsulation, 2) Tandem Forwarding, and 3) Egress Decapsulation and Forwarding. We address the generic concepts first before looking at possible existing or new encapsulations and their applicability.

Multicast SLSR is also possible (but with limitations to keep the header sizes from growing too large) and is briefly discussed after unicast.

### 7.1. Ingress Logical Unicast Forwarding

Here the flow information, likely IP header(s) + UDP/TCP header(s) is looked up and a sequence of link identifiers and a current hop must be placed on the packet, the packet must then be forwarded to the first of those links. This operation is identical to almost every tunnel protocol except that IP ECMP and/or LAG hash would potentially be unnecessary because the first link name would often resolve to a physical link not a LAG bundle. For example:

SrcIP	DstIP	SrcPrt	DstPrt	SLSR
192.0.2.4	192.0.2.9	1000	98	{3,2,5}
192.0.2.4	192.0.2.9	1001	99	{3,4,5}

Of course nothing precludes the use of LAG and the link identifier therefore identifying an entire LAG bundle rather than an element of that LAG. In fact it is possible to simultaneously support both concepts so that some traffic can be forwarded to the entire LAG while other traffic could be placed on a particular LAG bundle member at the discretion of the central computation.

## 7.2. Tandem Logical Unicast Forwarding

At tandem devices the operation would start by incrementing the current hop in the packet header (shown with a ^ symbol) and then forward to the link identified in the new current hop. If we support reversal, we change the previous link name to the local link name for that link. For example, referring to Figure 1 (and disregarding non relevant headers/options) after matching the first flow tuple at the ingress node C the packet is encapsulated with the SLSR header {3,2,5} and then leaves node C on interface 3 toward A. Then:

Packet arrives at node A on local interface 1 where it looks like this:

```

+-----+
| 3 | 2 | 5 | 192.0.2.9 | 192.0.2.4 | .. <payload> |
+-----+
| ^ |
+-----+

```

Current hop is incremented while previous hop is changed to local interface name (3 changes to a 1).

```

+-----+
| 1 | 2 | 5 | 192.0.2.9 | 192.0.2.4 | .. <payload> |
+-----+
| ^ |
+-----+

```

Packet is forwarded to interface for current hop i.e. 2.

Packet arrives at node B on local interface 1.

```
+-----+
| 1 | 2 | 5 | 192.0.2.9 | 192.0.2.4 | ..  <payload> |
+-----+
```

Current hop is incremented while previous hop is changed to local interface name 1 (2 changes to a 1).

```
+-----+
| 1 | 1 | 5 | 192.0.2.9 | 192.0.2.4 | ..  <payload> |
+-----+
```

Packet is forwarded to interface for current hop i.e. 5.

Packet arrives at node E on local interface 3.

```
+-----+
| 1 | 1 | 5 | 192.0.2.9 | 192.0.2.4 | ..  <payload> |
+-----+
```

Current hop is incremented while previous hop is changed to local interface name (5 changes to 3).

```
+-----+
| 1 | 1 | 3 | 192.0.2.9 | 192.0.2.4 | ..  <payload> |
+-----+
```

We are at the end of the path, so egress processing begins.

One additional step not described above is a reverse path check. Prior to substituting the reverse link identifier into the SLRSR header, the link identifier from the neighbor can be validated and the packet discarded if the neighbor link identifier in the packet is incorrect for the port the packet arrived on. This would reduce the chances of mis-delivery of the packet should a link identifier change or a link destination change while a packet is in flight.

### 7.3. Egress Logical Unicast Forwarding

Here the operation consists of normal IP/Ethernet etc. forwarding based on the IP destination / MAC or other ACL rules. Basically the SLRSR header is stripped and the packet is submitted to the Virtual Forwarding Instance, or Virtual Forwarding Function (VFI or VRF) for further processing.

Optionally the link identifier from the neighbor can be validated against what is expected and the packet discarded in the case of a

mismatch. This reduces the chance of mis-delivery as in the tandem case.

In addition, if path reversal is supported, the reverse path is compared against the current reverse path for this reverse flow and if it has changed the local forwarding state for the reverse flow would be updated. This would allow the head end to always dictate the forward and reverse path to be used for all packets in the flow without involving the controller on the egress side (and of course not needing to communicate with any tandem device).

Processing the reverse flow/path in this manner means that a flow is already present for the reverse direction without having to re-actively or pro-actively consult the controller. This results in a further 50% reduction in controller load. In the case of asymmetry this optimization is of course not possible.

## 8. Logical Multicast Forwarding SLSR Packets

Multicasting packets usually involve one of two approaches.

The first approach simply re-uses unicast and sends multiple copies to a pre-determined list of receivers. There is little to discuss with this approach as we can replicate SLSR based unicast packets just as easily as any other tunneling mechanism. Clearly such a serial unicast approach has nearly identical bandwidth overhead as other protocols like VPLS which also use this serial unicast mechanism.

It is therefore interesting to look at more efficient methods that involve the second multicast mechanism, which uses replication points in the network. These replication points are chosen so that copies are more efficiently made thereby eliminating multiple copies of the packet traversing any given link. Various logical tree structures are usually involved e.g. STP, SPB, TRILL, PIM, MOSPF etc.

These tree based mechanisms could in theory be implemented without requiring tandem state as an SLSR by introducing a branch point concept into the list of indexes. In this manner a complete tree as a pre-order traversal could be encoded along with the packet payload. It is not difficult to define a variety of different encodings that would accomplish this. The obvious objection to such a scheme is the sheer size of header required especially where a large network with many multicast receivers is concerned. It is therefore unlikely to be practical to encode any large tree of receivers and the SLSRs between them in any single header.

This leads to a hybrid approach which would encode a subset of the tree, say a single replication point and 5 or so recipients. This little tree or 'tree-let', would efficiently get a single packet to 5 (or some suitably small number) of recipients with an SLSR to the replication point and then SLSRs to each of the receivers. Such an encoding is much more reasonable than trying to encode all receivers and all replication points of a single tree in one packet. However, since this one packet would not reach all receivers, the head end would have to generate as many copies of the data packets as necessary to cover all recipients. As a result this approach would be a compromise between a full tree, and full head end replication. Variations in the size of the 'tree-let' header would allow for space v.s. bandwidth efficiency trade-offs while meeting the goal of remaining stateless in the core.

In the literature there are also non-exact methods to multicast without state such as with Bloom filters [BLOOM]. In this approach the links to be traversed are logically mapped into a field which is carried in the packet (for example if the links are given unique 128 bit sparse addresses then a 128 bit union of all the links to be traversed on the tree is encoded in the header). These mechanisms guarantee that all receivers will get a copy of the packet (because they check each link for inclusion in the Bloom Filter at each hop) however they do so at the expense of sending false positive copies to unintended receivers which must then filter the unwanted packets egress. Depending on the size of the Bloom Filter and the link identifiers various statistical trade-offs in false positives vs. packet header size can be made.

Other exact methods to encode and methods to compute SLSR multicast etc. are FFS.

## 9. Failure Recovery

A variety of failure recovery techniques can be employed with SLSR. The most obvious is to just re-compute all affected paths on indication of a link failure. This won't be discussed further.

More interesting are the so called fast restoration mechanisms. These can broadly be broken down into head end and tandem restoration.

Head end mechanisms that provide 1+1 protection have been around for a long time with MPLS-TP, PBT and SONET/DWDM. Similar mechanisms can be used with any tunnel type and of course SLSR is no exception. Probes can be sent down one source route, reflected back along the reverse source route and in this manner the forward and reverse paths can be simultaneously probed for failure. In the event of a failure a diverse alternate source route can rapidly be

added to the packet and the flow restored. The advantage of course with SLRS is that no state is required for either the primary or the backup path. As a result there is little added cost to having even greater redundancy than 1+1 with SLRS. The mechanisms to accomplish this are fairly obvious. Having the reverse path available at the egress means that state sharing the forward and reverse probes is easy.

In addition to 1+1 protection it is possible to do hop by hop fast reroute type detour protection. This can be done by substitution of a failed link identifier with a set of link identifiers that merge with the path downstream of the failure. An example is given further below for MPLS label stacks, however many other possibilities exist when a history of the packet's path is available to the detour mechanism. The history would permit the detour mechanism to spread the failing packets over different detours and thereby reduce the concentration of additional load imposed by the failure on the same set of links.

## 10. Comparison of Logical Model to Existing Source Routing

There are a number of existing protocols that support forms of source routing (or can be used to do something close to source routing). IPV4 and V6 had strict and loose node-by-node source routing options (now deprecated) and we'll discuss them briefly. Likewise MPLS behavior can be used to do strict link source routing where a label stack represents a list of link names, this has recently been called segment routing in [SEGMENT].

### 10.1. MPLS as a SLRS

MPLS is of course not a source routed forwarding protocol, at least not by design. Rather, packets follow an arbitrary path by substitution of a previous hop label with a next hop label and each hop must be pre-configured with the <incoming port, label> to <outgoing port, label> relationship. This is clearly not source routing because tandem configuration is required per path and per hop. However MPLS has a stacking mechanism that can be exploited to create a consumable list of link names to be traversed as they are popped.

The MPLS label stack can therefore be used to implement a flavor of SLRS. This is accomplished by pre-assigning a locally unique MPLS label to each outgoing link of a node. For example in figure 1, node D's link 3 would be assigned MPLS label 3 (but more likely a label value which is 1:1 related to link 3, however we stick with label=link for simplicity of explanation).

The tunnel encapsulation operation would therefore be to push a set of labels onto the frame where each label indicates which link to follow at that given exact strict hop. For example:

SrcIP	DstIP	SrcPrt	DstPrt	MPLS SLSR
192.0.2.4	192.0.2.9	1000	98	push(3,push(2,push(5)))
192.0.2.4	192.0.2.9	1001	99	push(3,push(4,push(5)))

The tunnel tandem operation would then be to pop the label on the incoming frame (after optionally validating its reverse link identifier) and forward to the interface specified by the just popped label value. Every tandem node would be pre-configured approximately as per below. Note that as with any source routing mechanism, this tandem pre-configuration is independent of the actual paths that traverse the node. A table like the one below, with a few hundred interfaces and hence a few hundred labels, could support the transit of an infinite number of TE (or SPF) paths. For clarity we use label  $N = \text{interface}/N$  but in reality it would be label  $N = F(\text{interface}/N)$  since a 1:1 mapping 'F' is almost certainly required.

Incoming Interface	Label	Actions
any	1	Pop, forward to interface/1
any	2	Pop, forward to interface/2
:	:	:
any	N	Pop, forward to interface/N

If reverse validation is required the tables would be a bit different because they must match the label to the incoming interface and then pop it and then forward based on the next label. Reverse validation therefore requires two label lookups per forwarding operation.

Finally the tunnel egress operation would be normal forwarding to a VFI or VRF.

MPLS in this manner could be made to do SLSR of unicast frames but cannot be made to reverse the route because the route is consumed in transit. This method also uses many more bits than are really necessary. Each label consumes 32 bits which is rather more than required to express the number of links/adjacencies on a typical switch or router. For example, if the average packet size is 512 bytes, a 5 hop MPLS source route imposes a 4% overhead (20/512) on



some links with the largest overhead on the first few links. For larger packets this is likely not an issue, for smaller packets it is possibly a concern.

A more realistic number actually required per hop is probably 8 or 12 bits (256-4K links) and if more bits are required two hops can be consumed by any node with such a large nodal degree. The MPLS label also has an 8 bit TTL which is of course redundant in any source routing mechanism. This begs the question of if a smaller MPLS label would not be more suitable?

There are other issues with the use of MPLS, in particular current hardware can usually not stack very many labels at a time (3 on some popular ASICs). This would limit the network diameter to 4 hops. Of course NPU's or new ASICs could be extended to allow further ingress stacking.

It does not seem possible to do SLSR multicast with MPLS except of course via head end replication.

The hop(ingress stack size) limit, lack of reverse, consumable route and lack of efficient multicast still do not invalidate use of MPLS source routing for many networks and its use would have a noticeable positive impact on the scale/speed of a central controller in such environments.

MPLS fast reroute mechanisms can also be implemented locally in a similar fashion thus further improving controller scale by alleviating the need for 50ms responses network wide from the controller and giving the controller more lee-way to recover after the fast reroutes have detoured traffic around the failed nodes and/or links.

Consider possible local actions when the link A.2 between nodes A and B in Figure 1 fails. Since there is still a link A.4 available, the node A can locally change the action associated with label 2 to instead send to interface 4 when interface 2 fails. If an entire adjacency fails, such as would happen when both A.2 and A.4 fail, then a link detour can be locally performed by reprogramming the actions for labels 2 and 4 to now push labels 3,3 and send to interface 3. This will cause a detour via D back to B. More elaborate kinds of detour are possible by processing two link names ahead instead of one, including nodal detours. These can be done locally without end to end path knowledge and hence scale independently to the number of paths. Eventually the controller will detect the failure and reconstruct the SLSRs at the head end and the use of the detour will stop without having to withdraw any state in the core.

If MPLS is of use in the context of SLSR then it would be worth considering a number of future extensions to MPLS. Some things to consider could be a smaller MPLS label option, say 16 bits with no TTL and the possibility of not popping but rotating the label to the bottom of the stack to preserve the path history for OAM and reversibility reasons. While these sorts of things are of course not possible with existing ASICs they are easy to do on existing NPU's and new work on Protocol Oblivious Forwarding [POF] allows near arbitrary bit pattern/action matches to be programmed by an SDN controller permitting a more optimal data path encoding of SLSR than can be obtained by simply reusing MPLS.

#### 10.2. IPV4/6 Options as SLSR

IP header option 9 [RFC791] defined (but now deprecated) the Strict Source and Record Route (SSRR) option for IPV4 packets. This option has(had) a 'length' field, a 'pointer' field and an array of 'route data' fields. The element in the array of 'route data' indexed by the 'pointer' field contains the IP address of the immediate next hop towards which the packet must be forwarded, the 'pointer' field is incremented, and the previous hop is filled in with the IP address of the current device prior to actually forwarding the packet. Up to 9 hops could be specified in this manner. IPV6 also had a similar option "RH0" which is also now deprecated [SRBAD].

IPV4 and V6 Strict Source and Record Route methods could be used to implement Strict Link Source Routing. This would be accomplished by assigning a 32 bit number to the link and then using the 32 bit number in place of the IPV4 or V6 address in the route list.

In both IPV4 and IPV6 the source routing options were found to be harmful to the Internet at large for a number of reasons. These reasons are described in [SRBAD] but briefly there were two broad classes of problem encountered. 1) Harm to intermediate links and 2) harm to end hosts. For example:

- Since it was possible to list a waypoint more than once in the route data, it was possible to loop traffic around multiple times (9 times in the case of IPV4 and 90 times in the case of IPV6). This looping allowed saturation of high speed links by hosts that had an order (or two) smaller bandwidth access to the Internet. A congestion style DOS was therefore possible from low speed access links against higher speed core links.

- Various schemes such as bypassing of firewalls etc. are of course easy to do when a host can specify waypoints that detour around a firewall.

- Spoofing using the reverse route. Since the reverse source route is installed against the IP SA by a host that receives it, it is possible to use a bogus IP SA in combination with a reverse source route that detours the packets to the imposter host.

### 10.3. Protocol Oblivious Forwarding as SLSR mechanism

The OpenFlow [OPENFLOW] protocol defines methods for an external controller to cause the manipulation of known packet headers and fields within those headers by a forwarding element. As such it is currently limited to matching on known fields like MPLS, IP, Ethernet etc. and taking actions on those fields. While flexible there are still many things at the data path level that OpenFlow cannot do including generic source routing such as SLSR.

The Protocol Oblivious Forwarding [POF] protocol is a proposed extension to OpenFlow which permits arbitrary bit pattern matching/actions and is therefore much more flexible. The goal of POF is to allow a controller to define a new data path in addition to a new control plane and to then program the data path on the forwarding elements to its specifications. POF is therefore not limited to existing IP, MPLS, Ethernet fields.

It would therefore be possible with POF to implement a highly flexible SDN tunnel data plane that closely resembles the idealized SLSR data path. Strictly by way of example POF could implement a flexible SLSR header along the following lines:

NextHop	Hop	Hop	Hop	Hop	~
Index:4	Count:4	Size:4	0	1	N

With only five bytes, this header could represent 3 hops with 256 links per hop, 4 hops with 64 links per hop, or 6 hops with 16 links per hop, etc. With additional bytes of course more/longer combinations are possible with very reasonable overhead. This is considerably more compact than the other described options and without sacrificing reversibility or giving up the OAM benefits of knowing the exact path the packet has taken.

POF however could also implement other variations of SLSR based on MPLS. For example POF could implement a smaller MPLS label, say a 16 bit label without a TTL. POF could theoretically also implement a rotating label list instead of a popping label stack.

POF appears to be ideally suited for SLSR developments beyond what can currently be done with MPLS.

## 11. Security Considerations

Source Routing security concerns are also discussed in the previous section related to IPV4 and IPV6 now deprecated nodal source routing.

This draft is proposing link based source routing and that it be used as a tunneling mechanism only. This means that only devices that are at the edge of an SDN sub-network would be allowed to insert strict link source routes. Note that an MPLS label can only be inserted by a Label Edge Router (LER) and processed by Label Switch Routers (LSR) and not by end hosts. Therefore SLSR should be no more or less secure than MPLS. In fact the absence of signaling protocols like RSVP-TE removes a point of attack. The fact that this mechanism is intended for use by a central controller further mitigates the possible attacks as encrypted communications are used to the edge devices which are the only device able to insert the strict link source routes.

There is however the possibility that an attacker could attach to a core device and inject strict link source routed packets. Methods to prevent this however are not hard, in particular the adjacency would have to be reported to the controller and the controller would have to enable packet forwarding. Unless the controller recognized both ends of the link as being part of its controlled domain it should not enable the strict link source routing capability on that interface thus preventing the threat.

Other interfaces, such as those facing a network of hosts or devices not in the domain of the controller would, as with current BCP's, drop any source routed frame in any format (new or old).

As previously mentioned there are ways to spread the link names into a 32 bit space such that the exact mappings are only known by the controller and the tandem node in question. This would prevent any easy form of guessing being used to construct an SLSR. One such example of this kind of secure source routing is given in [SANE].

Source Routing also is unique in that the packets themselves give details about slices/cuts through the topology, therefore with sufficient interception of packets from diverse sources and destinations in the network, an attacker could build up a detailed view of the network topology, this would be a concern for a carrier SDN network in particular where details of topology are considered a valuable asset, although exploiting knowledge of the topology would be more challenging given the secure protocols that exist between a controller and the forwarding entities.

In the SDN context there appears to be little need for a loose source route. Loose source routing adds additional security concerns because it does not require knowledge of the entire path to construct an attack. If loose source routing is included the security concerns should be addressed.

## 12. Conclusions and Future work

SDN where a central controller creates either pro-actively or re-actively the state for a sub-network of forwarding devices will have performance limitations that are related to network diameter/size, network recovery requirements and the amount of state they need to distribute. Strict Link Source Routing mechanisms can alleviate these problems allowing greater scale and faster recovery. MPLS can be used to implement this on a small scale with some of the benefits. IPV4 and IPV6 source routing options can be used to implement this on a larger scale with more of the benefits but at much larger packet overhead but are however perceived as risky and have been deprecated from IP. These risks however can be mitigated in this specific use. No existing mechanism however is optimum, and therefore there is room for a new mechanism that addresses these requirements and includes multicast methods and more efficient encoding of link names than is currently possible. One possible solution is to look at a smaller MPLS label for this purpose and to look at ways to retain the popped labels for the purposes of end to end path reversal and OAM. New work in SDN, in particular Protocol Oblivious Forwarding may make these kinds of things possible in a generic manner.

## 13. IANA Considerations

This memo includes no request to IANA.

## 14. References

### 14.1. Informative References

- [BLOOM]           Active Bloom Filters for Multicast Addressing,  
                  Z. Heszberger et. al. Budapest  
                  University of Technology and Economics.
- [OPENFLOW]       [www.openflow.org](http://www.openflow.org)
- [ONF]             [www.opennetworking.org](http://www.opennetworking.org)
- [POF]             Protocol Oblivious Forwarding:  
                  <http://www.poforwarding.org/>

- [PLACEMENT] The Controller Placement Problem, Nick McKeon, Brandon Heller, Rob Sherwood. HotSDN'12, August 13, 2012, Helsinki, Finland. 2012 ACM 978-1-4503-1477-0/12/08, <http://conferences.sigcomm.org/sigcomm/2012/paper/hotsdn/p7.pdf>
- [RFC791] Internet Protocol, Information Sciences Institute, RFC 791, September 1981.
- [SANE] SANE: A Protection Architecture for Enterprise Networks, Martin Casado, Nick McKeown, <http://yuba.stanford.edu/~casado/sane.pdf>, Stanford and ICSI 2005.
- [SDNGOOG] SDN at Google - Opportunities for WAN optimization, E. Crabbe, V. Valancius, 8/1/2012. Presentation at IETF84 SDN BOF.
- [SEGMENT] Segment Routing with IS-IS, S.Previdi et. Al. <http://tools.ietf.org/html/draft-previdi-filsfils-isis-segment-routing-00>
- [SLSR] Software Defined Networking and Centralized Controller State Distribution Reduction, [www.ieee802.org/1/files/public/docs2012/new-ashwood-sdn-optimizations-0712-v01.pdf](http://www.ieee802.org/1/files/public/docs2012/new-ashwood-sdn-optimizations-0712-v01.pdf)
- [SRBAD] Deprecation of Source Routing Options in IPV4 <http://tools.ietf.org/html/draft-reitzel-ipv4-source-routing-is-evil-00>
- [SRSDN] Source Routed Forwarding with SDN, M. Soliman <http://conferences.sigcomm.org/co-next/2012/e proceedings/student/p43.pdf>

## 15. Authors' Addresses

Peter Ashwood-Smith  
Huawei Canada Inc.  
303 Terry Fox Drive, Suite 400, Kanata, Ontario K2K 3J1  
Email: Peter.AshwoodSmith@huawei.com

Mourad Soliman  
Carleton University,  
1125 Colonel By Drive Ottawa, Ontario K1S 5B6 Canada  
Email: MouradSoliman@cmail.carleton.ca

Tao Wan  
Huawei Canada Inc.  
303 Terry Fox Drive, Suite 400, Kanata, Ontario K2K 3J1  
Email: Tao.Wan@huawei.com

## 16. Contributors

We invite more contributors.

## 17. Acknowledgements

We gratefully appreciate the feedback of Nigel Bragg, Sue Hares, Peter Willis, Biswajit Nandy and Linda Dunbar.





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.

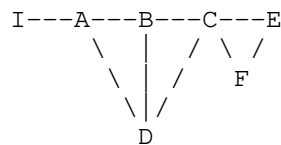


Figure 1: An illustration of SR properties

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.

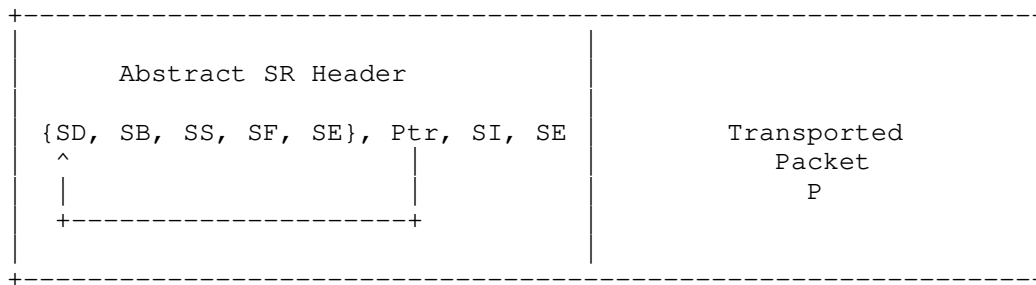


Figure 2: Packet P at node I

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}.

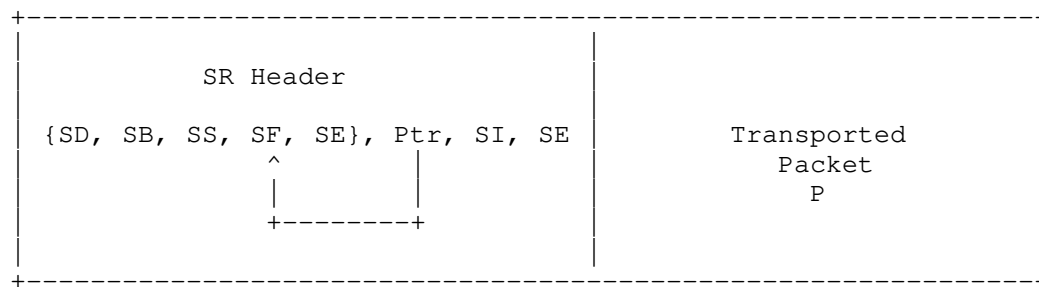


Figure 3: Packet P at node C

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.

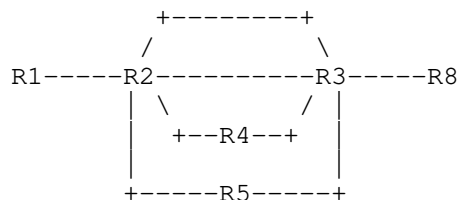


Figure 5: IGP Segments - Illustration

IP address allocated by the operator:	
192.0.2.1/32	as a loopback of R1
192.0.2.2/32	as a loopback of R2
192.0.2.3/32	as a loopback of R3
192.0.2.4/32	as a loopback of R4
192.0.2.5/32	as a loopback of R5
192.0.2.8/32	as a loopback of R8
198.51.100.9/32	as an anycast loopback of R4
198.51.100.9/32	as an anycast loopback of R5
SRGB defined by the operator as 1000-5000	
Global IGP SID allocated by the operator:	
1001	allocated to 192.0.2.1/32
1002	allocated to 192.0.2.2/32
1003	allocated to 192.0.2.3/32
1004	allocated to 192.0.2.4/32
1008	allocated to 192.0.2.8/32
2009	allocated to 198.51.100.9/32
Local IGP SID allocated dynamically by R2	
for its "north" adjacency to R3:	9001
for its "north" adjacency to R3:	9003
for its "south" adjacency to R3:	9002
for its "south" adjacency to R3:	9003

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.

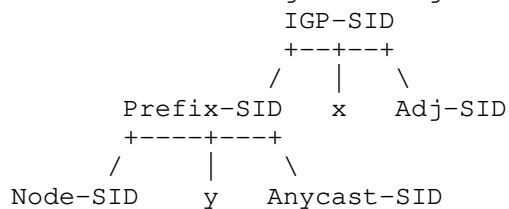


Figure 7: IGP SID Terminology

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.

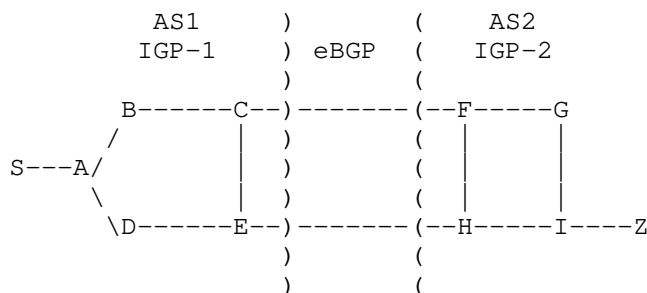


Figure 9: External Adjacency Example

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.

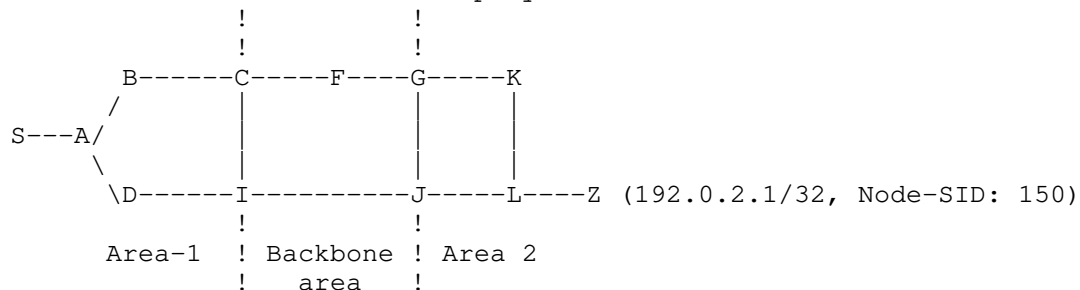


Figure 10: Inter-Area Topology Example

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: April 24, 2014

C. Filsfils, Ed.  
Cisco Systems, Inc.  
P. Francois, Ed.  
IMDEA Networks  
S. Previdi  
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  
S. Kini  
Ericsson  
E. Crabbe  
Google, Inc.  
October 21, 2013

Segment Routing Use Cases  
draft-filsfils-rtgwg-segment-routing-use-cases-02

Abstract

Segment Routing (SR) leverages the source routing and tunneling paradigms. 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. SR allows to enforce a flow through any topological path and service chain while maintaining per-flow state only at the ingress node of the SR domain.

The Segment Routing architecture can be directly applied to the MPLS dataplane with no change on the forwarding plane. It requires 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. Companion Documents . . . . .	4
1.2. Editorial simplification . . . . .	5
2. IGP-based MPLS Tunneling . . . . .	5
3. Fast Reroute . . . . .	7
3.1. Protecting node and adjacency segments . . . . .	7
3.2. Protecting a node segment upon the failure of its advertising node . . . . .	8
3.2.1. Advertisement of the Mirroring Capability . . . . .	10
3.2.2. Mirroring Table . . . . .	10
3.2.3. LFA FRR at the Point of Local Repair . . . . .	10
3.2.4. Modified IGP Convergence upon Node deletion . . . . .	11
3.2.5. Conclusions . . . . .	11
4. Traffic Engineering . . . . .	12
4.1. Traffic Engineering without Bandwidth Admission Control . . . . .	12
4.1.1. Anycast Node Segment . . . . .	12
4.1.2. Distributed CSPF-based Traffic Engineering . . . . .	17
4.1.3. Egress Peering Traffic Engineering . . . . .	18
4.1.4. Deterministic non-ECMP Path . . . . .	20
4.1.5. Load-balancing among non-parallel links . . . . .	21
4.2. Traffic Engineering with Bandwidth Admission Control . . . . .	22
4.2.1. Capacity Planning Process . . . . .	22
4.2.2. SDN/SR use-case . . . . .	25
4.2.3. Residual Bandwidth . . . . .	29
5. Service chaining . . . . .	29
6. OAM . . . . .	30
6.1. Monitoring a remote bundle . . . . .	30
6.2. Monitoring a remote peering link . . . . .	30
7. IANA Considerations . . . . .	30
8. Manageability Considerations . . . . .	31
9. Security Considerations . . . . .	31
10. Acknowledgements . . . . .	31
11. References . . . . .	31
11.1. Normative References . . . . .	31
11.2. Informative References . . . . .	32
Authors' Addresses . . . . .	33

## 1. Introduction

The objective of this document is to illustrate the properties and benefits of the SR architecture, through the documentation of various SR use-cases.

Section 2 illustrates the ability to tunnel traffic towards remote service points without any other protocol than the IGP.

Section 3 reports various FRR use-cases leveraging the SR functionality.

Section 4 documents traffic-engineering use-cases, with and without support of bandwidth admission control.

Section 5 documents the use of SR to perform service chaining.

Section 6 illustrates OAM use-cases.

### 1.1. Companion Documents

The main reference for this document is the SR architecture defined in [draft-filsfils-rtgwg-segment-routing-01].

The SR instantiation in the MPLS dataplane is described in [I-D.gredler-isis-label-advertisement].

[draft-filsfils-spring-segment-routing-ldp-interop-00] documents the co-existence and interworking with MPLS Signaling protocols.

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

OSPF protocol extensions for Segment Routing are defined in [draft-psenak-ospf-segment-routing-extensions-00].

Fast-Reroute for Segment Routing is described in [I-D.francois-sr-frr].

The PCEP protocol extensions for Segment Routing are defined in [draft-msiva-pce-pcep-segment-routing-extensions-00].

The SR instantiation in the IPv6 dataplane will be described in a future draft.



## 1.2. Editorial simplification

A unique index is allocated to each IGP Prefix Segment. The related absolute segment associated to an IGP Prefix SID is determined by summing the index and the base of the SRGB. In the SR architecture, each node can be configured with a different SRGB and hence the absolute SID associated to an IGP Prefix Segment can change from node to node.

We have described the first use-case of this document in the most generic way, i.e. with different SRGB at each node in the SR IGP domain. We have detailed the packet path highlighting that the SID of a Prefix Segment may change hop by hop.

For editorial simplification purpose, we will assume for all the other use cases that the operator ensures a single consistent SRGB across all the nodes in the SR IGP domain. In that case, all the nodes associate the same absolute SID with the same index and hence one can use the absolute SID value instead of the index to refer to a Prefix SID.

Several operators have indicated that they would deploy the SR technology in this way: with a single consistent SRGB across all the nodes. They motivated their choice based on operational simplicity (e.g. troubleshooting across different nodes).

While this document notes this operator feedback and we use this deployment model to simplify the text, we highlight that the SR architecture is not limited to this specific deployment use-case (different nodes may have different SRGB thanks to the indexation of Prefix SID's).

## 2. IGP-based MPLS Tunneling

SR, applied to the MPLS dataplane, offers the ability to tunnel services (VPN, VPLS, VPWS) from an ingress PE to an egress PE, without any other protocol than ISIS or OSPF. LDP and RSVP-TE signaling protocols are not required.

The operator only needs to allocate one node segment per PE and the SR IGP control-plane automatically builds the required MPLS forwarding constructs from any PE to any PE.

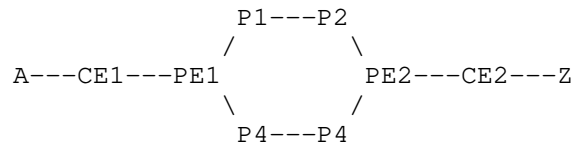


Figure 1: IGP-based MPLS Tunneling

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. Upon receiving a packet from A destined to Z, PE1 pushes two labels onto the packet: the top label is the Prefix SID attached to 192.168.0.2/32, the bottom label is the VPN label LZ attached to the VPN route Z.

The Prefix-SID attached to prefix 192.168.0.2 is a shared segment within the IGP domain, as such it is indexed.

Let us assume that:

- the operator allocated the index 2 to the prefix 192.168.0.2/32
- the operator allocated SRGB [100, 199] at PE1
- the operator allocated SRGB [200, 299] at P1
- the operator allocated SRGB [300, 399] at P2
- the operator allocated SRGB [400, 499] at P3
- the operator allocated SRGB [500, 599] at P4
- the operator allocated SRGB [600, 699] at PE2

Thanks to this context, any SR-capable IGP node in the domain can determine what is the segment associated with the Prefix-SID attached to prefix 192.168.0.2/32:

- PE1's SID is  $100+2=102$
- P1's SID is  $200+2=202$

- P2's SID is  $300+2=302$
- P3's SID is  $400+2=402$
- P4's SID is  $500+2=502$
- PE2's SID is  $600+2=602$

Specifically to our example this means that PE1 load-balance the traffic to VPN route Z between P1 and P4. The packets sent to P1 have a top label 202 while the packets sent to P4 have a top label 502. P1 swaps 202 for 302 and forwards to P2. P2 pops 302 and forwards to PE2. The packets sent to P4 had label 502. P4 swaps 502 for 402 and forwards the packets to P3. P3 pops the top label and forwards the packets to PE2. Eventually all the packets reached PE2 with one single label: LZ, the VPN label attached to VPN route Z.

This scenario illustrates how supporting MPLS services (VPN, VPLS, VPWS) with SR has the following benefits:

- Simple operation: one single intra-domain protocol to operate: the IGP. No need to support IGP synchronization extensions as described in [RFC5443] and [RFC6138].
- Excellent scaling: one Node-SID per PE.

### 3. Fast Reroute

Segment Routing aims at supporting services with tight SLA guarantees [draft-filsfils-rtgwg-segment-routing-01]. To meet this goal, local protection mechanisms can be useful to provide fast connectivity restoration after the sudden failure of network components. Protection mechanisms for segments aim at letting a point of local repair (PLR) pre-compute and install state allowing to locally recover the delivery of packets when the primary outgoing interface corresponding to the protected active segment is down.

This section describes use-cases leading to the definition of different protection mechanisms for node, adjacency, and service segments to be supported by the SR architecture.

#### 3.1. Protecting node and adjacency segments

Node and adjacency segments are used to determine the path that a packet should follow from an ingress node to an egress node of the SR domain or a service node.

Ensuring fast recovery of the packet delivery service may wear different requirements depending on the application using the segment. For this reason, the SR architecture should be able to accomodate multiple protection mechanisms and provide means to the operator to configure the protection scheme applied for the segments that are advertised in the SR domain.

The operator may want to achieve fast recovery in case of failures with as little management effort as possible, using a protection mechanism provided by the Segment Routing architecture itself. In this case, a Segment Routing node is in charge of discovering "by default" protection paths for each of its adjacent network component, with minimal operational impact. Approaches for such applications, typically in line with classical IP-FRR solutions, are discussed in [I-D.francois-sr-frr].

The operator of a Segment Routing network may also have strict policies on how a given network component should be protected against failures. A typical case is the knowledge by an external controller (or through any other tool used by the operator) of shared risk among different components, which should not be used to protect each other. An operator could notably use [I-D.sivabalan-pce-segment-routing] for this purpose.

Third, some SR applications have strict requirements in terms of guaranteed performance, disjointness in the infrastructure components used for different services, or for redundant provisioning of such services. An approach for providing resiliency in these contexts is explained in [I-D.shakir-rtgwg-sr-performance-engineered-lsps]. It is basically aiming at letting the ingress node in the SR domain be in charge of the recovery of the Segment Routing paths that it uses to support these services.

The protection behavior applied to a given SID must be advertised in the routing information that is propagated in the SR domain for that SID, e.g., in [I-D.previdi-isis-segment-routing-extensions]. Nodes injecting traffic in the SR domain can hence select segments based on the protection mechanism that is required for their application.

### 3.2. Protecting a node segment upon the failure of its advertising node

Service segments can also benefit from a fast restoration mechanism provided by the SR architecture.

Referring to the below figure, let us assume:

A is identified by IP address 192.0.2.1/32 to which Node-SID 101 is attached.

B is identified by IP address 192.0.2.2/32 to which Node-SID 102 is attached

A and B host the same set of services.

Each service is identified by a local segment at each node: i.e. node A allocates a local service segment 9001 to identify a specific service S while the same service is identified by a local service segment 9002 at B. Specifically, for the sake of this illustration, let us assume that service S is a BGP-VPN service where A announces a VPN route V with BGP nhop 192.0.2.1/32 and local VPN label 9001 and B announces the same VPN route V with BGP nhop 192.0.2.2/32 and local VPN label 9002.

A generic mesh interconnects the three nodes M, Q and B.

N prefers to use the service S offered by A and hence sends its S-destined traffic with segment list {101, 9001}.

Q is a node connected to A.

Q has a method to detect the loss of node A within a few 10's of msec.

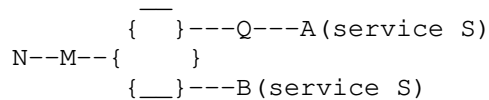


Figure 2: Service Mirroring

In that context, we would like to protect the traffic destined to service S upon the failure of node A.

The solution is built upon several components:

1. B advertises its mirroring capability for mirrored Node-SID 101
2. B pre-installs a mirroring table in order to process the packets originally destined to 101.
3. Q and any neighbor of A pre-install the Mirror\_FRR LFA extension
4. All nodes implements a modified SRDB convergence upon Node-SID 101 deletion

### 3.2.1. Advertisement of the Mirroring Capability

B advertises a MIRROR sub-TLV in its IGP Link-State Router Capability TLV with the values (TTT=000, MIRRORED\_OBJECT=101, CONTEXT\_SEGMENT=10002), [draft-filsfils-rtgwg-segment-routing-01], [I-D.previdi-isis-segment-routing-extensions] and [draft-psenak-ospf-segment-routing-extensions-00] for more details in the encodings.

Doing so, B advertises within the routing domain that it is willing to backup any traffic originally sent to Node-SID 101 provided that this rerouted traffic gets to B with the context segment 10002 directly preceding any local service segment advertised by A. 10002 is a local context segment allocated by B to identify traffic that was originally meant for A. This allows B to match the subsequent service segment (e.g. 9001) correctly.

### 3.2.2. Mirroring Table

We assume that B is able to discover all the local service segments allocated by A (e.g. BGP route reflection and add-path). B maps all the services advertised by A to its similar service representations. For example, service 9001 advertised by A is mapped to service 9002 advertised by B as both relate to the same service S (the same VPN route V). For example, B applies the same service treatment to a packet received with top segments {102, 10002, 9001} or with top segments {102, 9002}. Basically, B treats {10002, 9001} as a synonym of {9002}.

### 3.2.3. LFA FRR at the Point of Local Repair

In advance of any failure of A, Q (and any other node connected to A) learns the identity of the IGP Mirroring node for each Node-SID advertised by A (MIRROR\_TLV advertised by B) and pre-installs the following new MIRROR\_FRR entry:

- Trigger condition: the loss of nhop A
- Incoming active segment: 101 (a Node-SID advertised by A)
- Primary Segment processing: pop 101
  - Backup Segment processing: pop 101, push {102, 10002}
- Primary nhop: A
  - Backup nhop: primary path to node B

Upon detecting the loss of node A, Q intercepts any traffic destined to Node-SID 101, pops the segment to A (101) and push a repair tunnel {102, 10002}. Node-SID 102 steers the repaired traffic to B while context segment 10002 allows B to process the following service segment {9001} in the right context table.

#### 3.2.4. Modified IGP Convergence upon Node deletion

Upon the failure of A, all the neighbors of A will flood the loss of their adjacency to A and eventually every node within the IGP domain will delete 192.0.2.1/32 from their RIB.

The RIB deletion of 192.0.2.1/32 at N is beneficial as it triggers the BGP FRR Protection onto the precomputed backup next-hop [draft-rtgwg-bgp-pic-01.txt].

The RIB deletion at node M, if it occurs before the RIB deletion at N, would be disastrous as it would lead to the loss of the traffic from N to A before Q is able to apply the Mirroring protection.

The solution consists in delaying the deletion of the SRDB entry for 101 by 2 seconds while still deleting the IP RIB 192.0.2.1/32 entry immediately.

The RIB deletion triggers the BGP FRR and BGP Convergence. This is beneficial and must occur without delay.

The deletion of the SRDB entry to Node-SID101 is delayed to ensure that the traffic still in transit towards Node-SID 101 is not dropped.

The delay timer should be long enough to ensure that either the BGP FRR or the BGP Convergence has taken place at N.

#### 3.2.5. Conclusions

In our reference figure, N sends its packets towards A with the segment list {101, 9001}. The shortest-path from S to A transits via M and Q.

Within a few msec of the loss of A, Q activates its pre-installed Mirror\_FRR entry and reroutes the traffic to B with the following segment list {102, 10002, 9001}.

Within a few 100's of msec, any IGP node deletes its RIB entry to A but keeps its SRDB entry to Node-SID 101 for an extra 2 seconds.

Upon deleting its RIB entry to 192.0.2.1/32, N activates its BGP FRR entry and reroutes its S destined traffic towards B with segment list {102, 9002}.

By the time any IGP node deletes the SRDB entry to Node-SID 101, N no longer sends any traffic with Node-SID 101.

The deletion of the SRDB entry to Node-SID101 is delayed to ensure that the traffic still in transit towards Node-SID 101 is not dropped.

In conclusion, the traffic loss only depends on the ability of Q to detect the node failure of its adjacent node A.

#### 4. Traffic Engineering

In this section, we describe Traffic Engineering use-cases for SR, distinguishing use-cases for traffic engineering with bandwidth admission control from those without.

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

This section describes traffic-engineering use-cases which do not require bandwidth admission control.

The first sub-section illustrates the use of anycast segments to express macro policies. Two examples are provided: one involving a disjointness enforcement within a so-called dual-plane network, and the other involving CoS-based policies.

The second sub-section illustrate how a head-end router can combine a distributed CSPF computation with SR. Various examples are provided where the CSPF constraint or objective is either a TE affinity, an SRLG or a latency metric.

The third sub-section illustrates how SR can help traffic-engineer outbound traffic among different external peers, overriding the best installed IP path at the egress border routers.

The fourth sub-section describes how SR can be used to express deterministic non-ECMP paths. Several techniques to compress the related segment lists are also introduced.

The fifth sub-section describes a use-case where a node attaches an Adj-SID to a set of its interfaces however not sharing the same neighbor. The illustrated benefit relates to loadbalancing.

##### 4.1.1. Anycast Node Segment

The SR architecture defines an anycast segment as a segment attached to an anycast IP prefix ([RFC4786]).

The anycast node segment is an interesting tool for traffic engineering:



Macro-policy support: anycast segments allow to express policies such as "go via plane1 of a dual-plane network" (Section 4.1.1.1) or "go via Region3" (Section 4.1.3).

Implicit node resiliency: the traffic-engineering policy is not anchored to a specific node whose failure could impact the service. It is anchored to an anycast address/Anycast-SID and hence the flow automatically reroutes on any ECMP-aware shortest-path to any other router part of the anycast set.

The two following sub-sections illustrate to traffic-engineering use-cases leveraging Anycast-SID.

#### 4.1.1.1. Disjointness in dual-plane networks

Many networks are built according to the dual-plane design:

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.

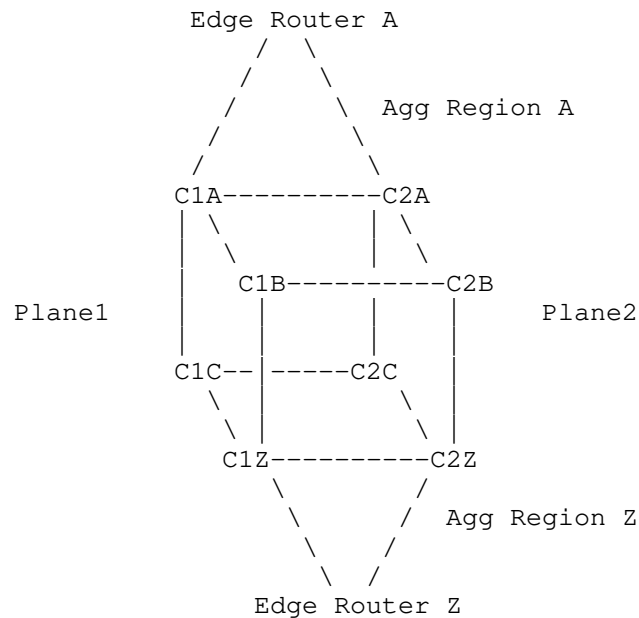


Figure 3: Dual-Plane Network and Disjointness

In the above network diagram, let us that the operator configures:

The four routers (C1A, C1B, C1C, C1Z) with an anycast loopback address 192.0.2.1/32 and an Anycast-SID 101.

The four routers (C2A, C2B, C2C, C2Z) with an anycast loopback address 192.0.2.2/32 and an Anycast-SID 102.

Edge router Z with Node-SID 109.

A can then use the three following segment lists to control its Z-destined traffic:

{109}: the traffic is load-balanced across any ECMP path through the network.

{101, 109}: the traffic is load-balanced across any ECMP path within the Plane1 of the network.

{102, 109}: the traffic is load-balanced across any ECMP path within the Plane2 of the network.

Most of the data traffic to Z would use the first segment list, such as to exploit the capacity efficiently. The operator would use the

two other segment lists for specific premium traffic that has requested disjoint transport.

For example, let us assume a bank or a government customer has requested that the two flows F1 and F2 injected at A and destined to Z should be transported across disjoint paths. The operator could classify F1 (F2) at A and impose an SR header with the second (third) segment list. Focusing on F1 for the sake of illustration, A would route the packets based on the active segment, Anycast-SID 101, which steers the traffic along the ECMP-aware shortest-path to the closest router part of the Anycast-SID 101, C1A is this example. Once the packets have reached C1A, the second segment becomes active, Node-SID 109, which steers the traffic on the ECMP-aware shortest-path to Z. C1A load-balances the traffic between C1B-C1Z and C1C-C1Z and then C1Z forwards to Z.

This SR use-case has the following benefits:

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

- Only two additional node segments (one Anycast-SID per plane).

- ECMP-awareness.

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

#### 4.1.1.2. CoS-based Traffic Engineering

Frequently, different classes of service need different path characteristics.

In the example below, a single-area international network with presence in four different regions of the world has lots of cheap network capacity from Region4 to Region1 via Region2 and some scarce expensive capacity via Region3.

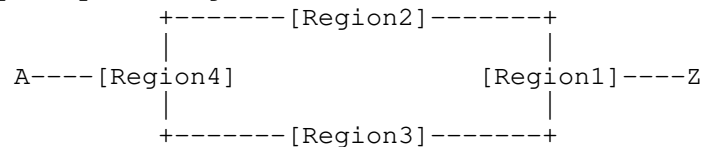


Figure 4: International Topology Example

In such case, the IGP metrics would be tuned to have a shortest-path from A to Z via Region2.

This would provide efficient capacity planning usage while fulfilling the requirements of most of the traffic demands. However, it may not suite the latency requirements of the voice traffic between the two cities.

Let us illustrate how this can be solved with Segment Routing.

The operator would configure:

- All the core routers in Region3 with an anycast loopback 192.0.2.3/32 to which Anycast-SID 333 is attached.
- A loopback 192.0.2.9/32 on Z and would attach Node-SID 109 to it.
- The IGP metrics such that the shortest-path from Region4 to Region1 is via Region2, from Region4 to Region3 is directly to Region3, the shortest-path from Region3 to Region1 is not back via Region4 and Region2 but straight to Region1.

With this in mind, the operator would instruct A to apply the following policy for its Z-destined traffic:

- Voice traffic: impose segment-list {333, 109}
  - Anycast-SID 333 steers the Voice traffic along the ECMP-aware shortest-path to the closest core router in Region3, then Node-SID 109 steers the Voice traffic along the ECMP-aware shortest-path to Z. Hence the Voice traffic reaches Z from A via the low-latency path through Region3.
- Any other traffic: impose segment-list {109}: Node-SID 109 steers the Voice traffic along the ECMP-aware shortest-path to Z. Hence the bulk traffic reaches Z from A via the cheapest path for the operator.

This SR use-case has the following benefits:

Zero per-service state and signaling at midpoint and tailend nodes.

One additional anycast segment per region.

ECMP-awareness.

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

#### 4.1.2. Distributed CSPF-based Traffic Engineering

In this section, we illustrate how a head-end router can map the result of its distributed CSPF computation into an SR segment list.

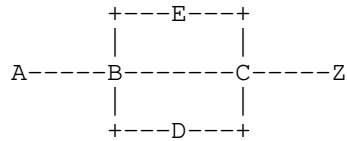


Figure 5: SRLG-based CSPF

Let us assume that in the above network diagram:

The operator configures a policy on A such that its Z-destined traffic must avoid SRLG1.

The operator configures SRLG1 on the link BC (or is learned dynamically from the IP/Optical interaction with the DWDM network).

The SRLG's are flooded in the link-state IGP.

The operator respectively configures the Node-SIDs 101, 102, 103, 104, 105 and 109 at nodes A, B, C, D, E and Z.

In that context, A can apply the following CSPF behavior:

- It prunes all the links affected by the SRLG1, computes an SPF on the remaining topology and picks one of the SPF paths.
  - In our example, A finds two possible paths ABECZ and ABDCZ and let's assume it takes the ABDCZ path.
- It translates the path as a list of segments
  - In our example, ABDCZ can be expressed as {104, 109}: a shortest path to node D, followed by a shortest-path to node Z.
- It monitors the status of the LSDB and upon any change impacting the policy, it either recomputes a path meeting the policy or update its translation as a list of segments.
  - For example, upon the loss of the link DC, the shortest-path to Z from D (Node-SID 109) goes via the undesired link BC. After a transient time immediately following such failure, the node A would figure out that the chosen path is no longer valid and instead select ABECZ which is translated as {103, 109}.
- This behavior is a local matter at node A and hence the details are outside the scope of this document.

The same use-case can be derived from any other C-SPF objective or constraint (TE affinity, TE latency, SRLG, etc.) as defined in [RFC5305] and [I-D.previdi-isis-te-metric-extensions]. Note that the bandwidth case is specific and hence is treated in Section 4.2.

#### 4.1.3. Egress Peering Traffic Engineering

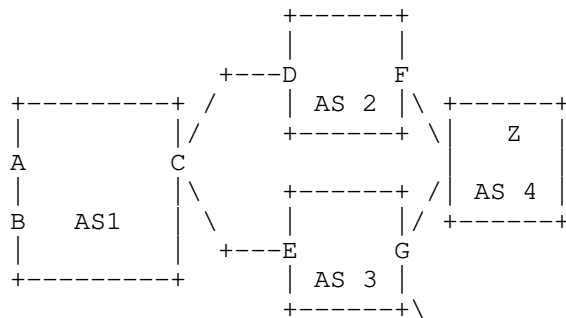


Figure 6: Egress peering traffic engineering

Let us assume that:

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

C sets next-hop-self before propagating the paths within AS1.

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

C only installs the path via AS2 in its RIB.

In that context, the operator of AS1 cannot apply the following traffic-engineering policy:

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.

This traffic-engineering policy can be supported thanks to the following SR configuration.

The operator configures:

C with a loopback 192.0.2.1/32 and attach the Node-SID 101 to it.

C to bind an external adjacency segment ([draft-filsfils-rtgwg-segment-routing-01]) to each of its peering interface.

For the sake of this illustration, let us assume that the external adjacency segments bound by C for its peering interfaces to (D, AS2) and (E, AS3) are respectively 9001 and 9002.

These external adjacencies (and their attached segments) are flooded within the IGP domain of AS1 [RFC5316].

As a result, the following information is available within AS1: ISIS Link State Database:

- Node-SID 101 is attached to IP address 192.0.2.1/32 advertised by C.
  - C is connected to a peer D with external adjacency segment 9001.
  - C is connected to a peer E with external adjacency segment 9002.
- BGP Database:

- Z is reachable via 192.0.2.1 with AS Path {AS2, AS4}.
- Z is reachable via 192.0.2.1 with AS Path {AS3, AS4}.

The operator of AS1 can thus meet its traffic-engineering objective by enforcing the following policies:

A should apply the segment list {101, 9001} to 60% of the Z-destined traffic and the segment list {101, 9002} to the rest.

B should apply the segment list {101, 9001} to 80% of the Z-destined traffic and the segment list {101, 9002} to the rest.

Node segment 101 steers the traffic to C.

External adjacency segment 9001 forces the traffic from C to (D, AS2), without any IP lookup at C.

External adjacency segment 9002 forces the traffic from C to (E, AS3), without any IP lookup at C.

A and B can also use the described segments to assess the liveness of the remote peering links, see OAM section.

#### 4.1.4. Deterministic non-ECMP Path

The previous sections have illustrated the ability to steer traffic along ECMP-aware shortest-paths. SR is also able to express deterministic non-ECMP path: i.e. as a list of adjacency segments. We illustrate such an use-case in this section.

```

A-B-C-D-E-F-G-H-Z
  |           |
  +--I-J-K-L-M--+

```

Figure 7: Non-ECMP deterministic path

In the above figure, it is assumed all nodes are SR capable and only the following SIDs are advertised:

- A advertises Adj-SID 9001 for its adjacency to B
- B advertises Adj-SID 9002 for its adjacency to C
- C advertises Adj-SID 9003 for its adjacency to D
- D advertises Adj-SID 9004 for its adjacency to E
- E advertises Adj-SID 9001 for its adjacency to F
- F advertises Adj-SID 9002 for its adjacency to G
- G advertises Adj-SID 9003 for its adjacency to H
- H advertises Adj-SID 9004 for its adjacency to Z
- E advertises Node-SID 101
- Z advertises Node-SID 109

The operator can steer the traffic from A to Z via a specific non-ECMP path ABCDEFGHZ by imposing the segment list {9001, 9002, 9003, 9004, 9001, 9002, 9003, 9004}.



The following sub-sections illustrate how the segment list can be compressed.

#### 4.1.4.1. Node Segment

Clearly the same exact path can be expressed with a two-entry segment list {101, 109}.

This example illustrates that a Node Segment can also be used to express deterministic non-ECMP path.

#### 4.1.4.2. Forwarding Adjacency

The operator can configure Node B to create a forwarding-adjacency to node H along an explicit path BCDEFGH. The following behaviors can then be automated by B:

B attaches an Adj-SID (e.g. 9007) to that forwarding adjacency together with an ERO sub-sub-TLV which describes the explicit path BCDEFGH.

B installs in its Segment Routing Database the following entry:

Active segment: 9007.

Operation: NEXT and PUSH {9002, 9003, 9004, 9001, 9002, 9003}

As a result, the operator can configure node A with the following compressed segment list {9001, 9007, 9004}.

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

A given node may assign the same Adj-SID to multiple of its adjacencies, even if these ones lead to different neighbors. This may be useful to support traffic engineering policies.

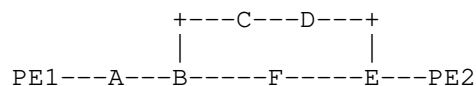


Figure 8: Adj-SID For Multiple (non-parallel) Adjacencies

In the above example, let us assume that the operator:

Requires PE1 to load-balance its PE2-destined traffic between the ABCDE and ABFE paths.

Configures B with Node-SID 102 and E with Node-SID 202.

Configures B to advertise an individual Adj-SID per adjacency (e.g. 9001 for BC and 9002 for BF) and, in addition, an Adj-SID for the adjacency set (BC, BF) (e.g. 9003).

With this context in mind, the operator achieves its objective by configuring the following traffic-engineering policy at PE1 for the PE2-destined traffic: {102, 9003, 202}:

Node-SID 102 steers the traffic to B.

Adj-SID 9003 load-balances the traffic to C or F.

From either C or F, Node-SID 202 steers the traffic to PE2.

In conclusion, the traffic is load-balanced between the ABCDE and ABFE paths, as desired.

#### 4.2. 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.

The first sub-section details the capacity planning process and the role of ECMP load-balancing. We highlight the relevance of SR in that context.

The next two sub-sections document two use-cases of SR-based traffic engineering with bandwidth admission control.

The second sub-section documents a concrete SR applicability involving centralized-based admission control. This is often referred to as the "SDN/SR use-case".

The third sub-section introduces a future research topic involving the notion of residual bandwidth introduced in [I-D.atlas-mpls-te-express-path].

##### 4.2.1. 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.

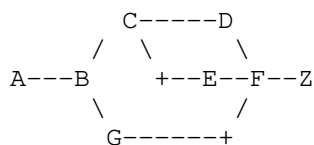


Figure 9: Capacity Planning an ECMP-based demand

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.

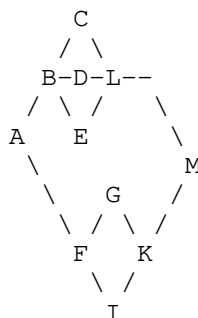


Figure 10: ECMP Topology Example

Segment Routing offers a simple support for such ECMP-based shortest-path placement: a node segment. A single node segment enumerates all the ECMP paths along the shortest-path.

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.

Segment Routing offers a simple support for explicit path policy. Let us provide two examples based on Figure 10.

First example: let us assume that the process has selected the flow AM for traffic-engineering away from its ECMP-enabled shortest path and flow AM must avoid consuming resources on the LM and the FG links.

The solution is straightforward: A sends its M-destined traffic towards the nhop F with a two-label stack where the top label is the adjacent segment FI and the next label is the node segment to M. Alternatively, a three-label stack with adjacency segments FI, IK and KM could have been used.

Second example: let us assume that AM is still the selected flow but the constraint is relaxed to only avoid using resources from the LM link.

The solution is straightforward: A sends its M-destined traffic towards the nhop F with a one-label stack where the label is the node segment to M. Note that while the AM flow has been traffic-engineered away from its natural shortest-path (ECMP across three paths), the traffic-engineered path is still ECMP-aware and leverages two of the three initial paths. This is accomplished with a single-label stack and without the enumeration of one tunnel per path.

Under the light of these examples, Segment Routing offers an interesting solution for Capacity Planning because:

- One node segment represents the set of ECMP-aware shortest paths.

- Adjacency segments allow to express any explicit path.

- The combination of node and adjacency segment allows to express any path without having to enumerate all the ECMP options.

The capacity planning process ensures that the majority of the traffic rides on node segments (ECMP-based shortest path), while a minority of the traffic is routed off its shortest-path.

The explicitly-engineered traffic (which is a minority) still benefits from the ECMP-awareness of the node segments within their segment list.

Only the head-end of a traffic-engineering policy maintains state. The midpoints and tail-ends do not maintain any state.

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

The heart of the application of SR to the SDN use-case lies in the SDN controller, also called Stateful PCE ([I-D.ietf-pce-stateful-pce]).

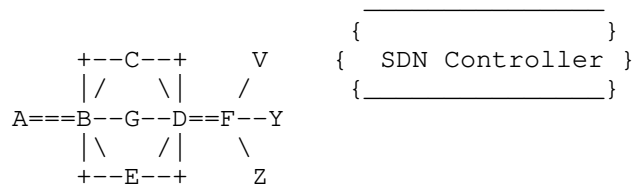
The SDN controller is responsible to control the evolution of the traffic matrix and topology. It accepts or denies the addition of new traffic into the network. It decides how to route the accepted traffic. It monitors the topology and upon failure, determines the minimum traffic that should be rerouted 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. [RFC5101] 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.ward-i2rs-framework], [I-D.crabbe-pce-pce-initiated-lsp] and [draft-msiva-pce-pcep-segment-routing-extensions-00]).

##### 4.2.2.1. Illustration



SDN/SR use-case

Let us assume that in the above network diagram:

An SDN Controller (SC) is connected to the network and is able to retrieve the topology and traffic information, as well as set traffic-engineering policies on the network nodes.

The operator (likely via the SDN Controller) as provisioned the Node-SIDs 101, 102, 103, 104, 105, 106, 107, 201, 202 and 203 respectively at nodes A, B, C, D, E, F, G, V, Y and Z.

All the links have the same BW (e.g. 10G) and IGP cost (e.g. 10) except the links BG and GD which have IGP cost 50.

Each described node connectivity is formed as a bundle of two links, except (B, G) and (G, D) which are formed by a single link each.

Flow FV is traveling from A to destinations behind V.

Flow FY is traveling from A to destinations behind Y.

Flow FZ is traveling from A to destinations behind Z.

The SDN Controller has admitted all these flows and has let A apply the default SR policy: "map a flow onto its ECMP-aware shortest-path".

In this example, this means that A respectively maps the flows FV onto segment list {201}, FY onto segment list {202} and FZ onto segment list {203}.

In this example, the reader should note that the SDN Controller knows what A would do and hence knows and controls that none of these flows are mapped through G.

Let us describe what happens upon the failure of one of the two links E-D.

The SDN Controller monitors the link-state database and detects a congestion risk due to the reduced capacity between E and D. Specifically, SC updates its simulation of the traffic according to the policies he instructed the network to use and discovers that too much traffic is mapped on the remaining link E-D.

The SDN Controller then computes the minimum number of flows that should be deviated from their existing path. For example, let us assume that the flow FZ is selected.

The SDN controller then computes an explicit path for this flow. For example, let us assume that the chosen path is ABGDFZ.

The SDN controller then maps the chosen path into an SR-based policy. In our example, the path ABGDFZ is translated into a segment list {107, 203}. Node-SID steers the traffic along ABG and then Node-SID 203 steers the traffic along GDFZ.

The SDN controller then applies the following traffic-engineering policy at A: "map any packet of the classified flow FZ onto segment-list {107, 203}". The SDN Controller uses PCEP extensions to instantiate that policy at A  
([draft-msiva-pcep-segment-routing-extensions-00]).

As soon as A receives the PCEP message, it enforces the policy and the traffic classified as FZ is immediately mapped onto segment list {107, 203}.

This immediately eliminate the congestion risk. Flows FV and FY were untouched and keep using the ECMP-aware shortest-path. The minimum amount of traffic was rerouted (FZ). No signaling hop-by-hop through the network from A to Z is required. No admission control hop-by-hop is required. No state needs to be maintained by B, G, D, F or Z. The only maintained state is within the SDN controller and the head-end node (A).

#### 4.2.2.2. Benefits

In the context of Centralized-Based Optimization and the SDN use-case, here are the benefits provided by the SR architecture:

- 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 (Section 3.

- Optimum virtualization: the policy state is in the packet header and not in the intermediate node along the policy. 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 lost programming the midpoints and tail-end along the policy.

#### 4.2.2.3. Dataset analysis

A future version of this document will report some analysis of the application of the SDN/SR use-case to real operator data sets.

A first, incomplete, report is available here below.

##### 4.2.2.3.1. Example 1

The first 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 riding on 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 riding off its shortest path.

We applied the SDN/SR use-case to this dataset. 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 SR segment-lists.

As a result, we measured that 98% of the traffic would be kept on its normal policy (ride 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 SR-capable ingress edge node maps the traffic on a single Node-ID 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.

#### 4.2.3. Residual Bandwidth

The notion of Residual Bandwidth (RBW) is introduced by [I-D.atlas-mps-te-express-path].

A future version of this document will describe the SR/RBW research opportunity.

### 5. Service chaining

Segment routing can be used to steer packets through services offered by middleboxes to perform specific actions such as DPI, accounting, etc.

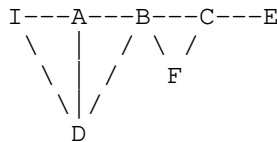


Figure 11

For example, as illustrated in Figure 11, an ingress node I selects an egress node E for a packet P. An application however requires that P undergoes a specific treatment (DPI, firewalling, ...) offered by a node D, reachable in the SR domain. In the SR architecture, this application can be supported through the use of a service segment with a local scope to D, say SS, following the nodal segment which corresponds to D. The Ingress box keeps the control of the egress node through which the packet needs to exit the network, by placing a nodal segment identifying the egress node after the service segment.

This would be achieved by letting I forward the packet P with the following sequence of segments: {D,SS,E}. D is a nodal segment, SS is the service segment corresponding to the service to apply to the packet P, and E is the nodal segment corresponding to the egress node selected by I for that packet.

## 6. OAM

### 6.1. Monitoring a remote bundle

This section documents a few representative SR/OAM use-cases.

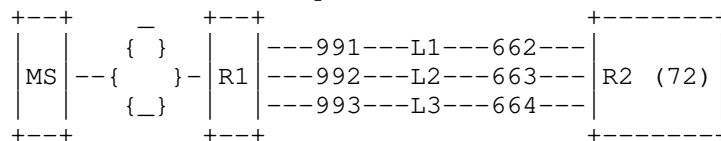


Figure 12: Probing all the links of a remote bundle

In the above figure, a monitoring system (MS) 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 segment information from the IGP LSDB and appends the following segment list: {72, 662, 992, 664} on its IP probe (whose source and destination addresses are the address of AA).

MS sends the probe to its connected router. If the connected router is not SR compliant, a tunneling technique can be used to tunnel the SR-based probe to the first SR router. The SR domain forwards the probe to R2 (72 is the node segment of R2). R2 forwards the probe to R1 over link L1 (adjacency segment 662). R1 forwards the probe to R2 over link L2 (adjacency segment 992). R2 forwards the probe to R1 over link L3 (adjacency segment 664). R1 then forwards the IP probe to AA as per classic IP forwarding.

### 6.2. Monitoring a remote peering link

In Figure 6, node A can monitor the dataplane liveness of the unidirectional peering link from C to D of AS2 by sending an IP probe with destination address A and segment list {101, 9001}. Node-SID 101 steers the probe to C and External Adj-SID 9001 steers the probe from C over the desired peering link to D of AS2. The SR header is removed by C and D receives a plain IP packet with destination address A. D returns the probe to A through classic IP forwarding. BFD Echo mode ([RFC5880]) would support such liveness unidirectional link probing application.

## 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, Thomas Telkamp, Ruediger Geib and Les Ginsberg 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.
- [RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast Services", BCP 126, RFC 4786, December 2006.
- [RFC5101] Claise, B., "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", RFC 5101, January 2008.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.
- [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.
- [RFC5443] Jork, M., Atlas, A., and L. Fang, "LDP IGP Synchronization", RFC 5443, March 2009.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.
- [RFC6138] Kini, S. and W. Lu, "LDP IGP Synchronization for Broadcast Networks", RFC 6138, February 2011.

## 11.2. Informative References

- [I-D.atlas-mppls-te-express-path]  
Atlas, A., Drake, J., Giacalone, S., Ward, D., Previdi, S., and C. Filsfils, "Performance-based Path Selection for Explicitly Routed LSPs",  
draft-atlas-mppls-te-express-path-02 (work in progress), February 2013.
- [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-01 (work in progress), April 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.gredler-isis-label-advertisement]  
Gredler, H., Amante, S., Scholl, T., and L. Jalil, "Advertising MPLS labels in IS-IS",  
draft-gredler-isis-label-advertisement-03 (work in progress), May 2013.
- [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-03 (work in progress), May 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-04 (work in progress), May 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-03 (work in progress), October 2013.
- [I-D.previdi-isis-te-metric-extensions]  
Previdi, S., Giacalone, S., Ward, D., Drake, J., Atlas, A., and C. Filsfils, "IS-IS Traffic Engineering (TE) Metric Extensions",

draft-previdi-isis-te-metric-extensions-03 (work in progress), February 2013.

[I-D.shakir-rtgwg-sr-performance-engineered-lsps]  
Shakir, R., Vernals, D., and A. Capello, "Performance Engineered LSPs using the Segment Routing Data-Plane", draft-shakir-rtgwg-sr-performance-engineered-lsps-00 (work in progress), July 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.

[I-D.ward-i2rs-framework]  
Atlas, A., Nadeau, T., and D. Ward, "Interface to the Routing System Framework", draft-ward-i2rs-framework-00 (work in progress), February 2013.

[draft-filsfils-rtgwg-segment-routing-01]  
Filsfils, C. and S. Previdi, "Segment Routing Architecture", October 2013.

[draft-filsfils-spring-segment-routing-ldp-interop-00]  
Filsfils, C. and A. Bashandy, "Segment Routing interoperability with LDP", October 2013.

[draft-msiva-pce-pcep-segment-routing-extensions-00]  
Filsfils, C. and S. Sivabalan, "PCEP Extensions for Segment Routing", May 2013.

[draft-psenak-ospf-segment-routing-extensions-00]  
Psenak, P. and S. Previdi, "OSPF Segment Routing Extensions", May 2013.

[draft-rtgwg-bgp-pic-01.txt]  
Filsfils, C., Bashandy, A., and P. Mohapatra, "BGP Prefix Independent Convergence", March 2013.

## Authors' Addresses

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

Email: cfilsfil@cisco.com

Pierre Francois (editor)  
IMDEA Networks  
Leganes,  
ES

Email: pierre.francois@imdea.org

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

Email: sprevidi@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

Sriganesh Kini  
Ericsson  
300 Holger Way  
San Jose, CA 95134  
US

Email: sriganesh.kini@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: September 10, 2015

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  
Individual Contributor  
March 9, 2015

Segment Routing interoperability with LDP  
draft-filsfils-spring-segment-routing-ldp-interop-03

Abstract

A Segment Routing (SR) 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. 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 data plane with no change in the forwarding plane. This drafts describes how Segment Routing operates in a network where LDP is deployed and in the case where SR-capable and non-SR-capable nodes coexist.

Requirements Language

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

## Status of This Memo

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

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

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

This Internet-Draft will expire on September 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 . . . . .	3
2. SR/LDP Ship-in-the-night coexistence . . . . .	3
2.1. MPLS2MPLS co-existence . . . . .	5
2.2. IP2MPLS co-existence . . . . .	6
3. Migration from LDP to SR . . . . .	6
4. SR and LDP Interworking . . . . .	7
4.1. LDP to SR . . . . .	7
4.2. SR to LDP . . . . .	8
5. Leveraging SR benefits for LDP-based traffic . . . . .	9
5.1. Eliminating Directed LDP Session . . . . .	11
5.2. Guaranteed FRR coverage . . . . .	12
6. Inter-AS Option C, Carrier's Carrier and Seamless MPLS . . . . .	13
7. IANA Considerations . . . . .	13
8. Manageability Considerations . . . . .	13

9. Security Considerations . . . . .	13
10. Acknowledgements . . . . .	13
11. References . . . . .	14
11.1. Normative References . . . . .	14
11.2. Informative References . . . . .	14
Authors' Addresses . . . . .	14

## 1. Introduction

Segment Routing, as described in [I-D.ietf-spring-segment-routing], can be used on top of the MPLS data plane without any modification as described in [I-D.ietf-spring-segment-routing-mpls].

Segment Routing control plane can co-exist with current label distribution protocols such as LDP.

This draft outlines the mechanisms through which SR provides interoperability with LDP in cases where a mix of SR-capable and non-SR-capable routers co-exist within the same network.

The first section describes the co-existence of SR with other MPLS Control Plane. The second section documents a method to migrate from LDP to SR-based MPLS tunneling. The third section documents the interworking of LDP and SR in the case of non-homogenous deployment. The fourth section describes how a partial SR deployment can be used to provide SR benefits to LDP-based traffic. The fifth section describes a possible application of SR in the context of inter-domain MPLS use-cases.

## 2. SR/LDP Ship-in-the-night coexistence

We call "MPLS Control Plane Client (MCC)" any control plane protocol installing forwarding entries in the MPLS data plane. SR, LDP, RSVP-TE, BGP 3107, VPNv4, etc. are examples of MCCs.

An MCC, operating at node N, must ensure that the incoming label it installs in the MPLS data plane of Node N has been uniquely allocated to himself.

Thanks to the defined segment allocation rule and specifically the notion of the SRGB, SR can co-exist with any other MCC.

This is clearly the case for the adjacency segment: it is a local label allocated by the label manager, as for any MCC.

This is clearly the case for the prefix segment: the label manager allocates the SRGB set of labels to the SR MCC client and the

operator ensures the unique allocation of each global prefix segment/label within the allocated SRGB set.

Note that this static label allocation capability of the label manager has been existing for many years across several vendors and hence is not new. Furthermore, note that the label-manager ability to statically allocate a range of labels to a specific application is not new either. This is required for MPLS-TP operation. In this case, the range is reserved by the label manager and it is the MPLS-TP NMS (acting as an MCC) that ensures the unique allocation of any label within the allocated range and the creation of the related MPLS forwarding entry.

Let us illustrate an example of ship-in-the-night (SIN) coexistence.

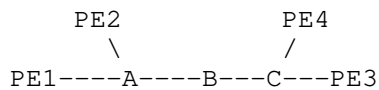


Figure 1: SIN coexistence

The EVEN VPN service is supported by PE2 and PE4 while the ODD VPN service is supported by PE1 and PE3. The operator wants to tunnel the ODD service via LDP and the EVEN service via SR.

This can be achieved in the following manner:

The operator configures PE1, PE2, PE3, PE4 with respective loopbacks 192.0.2.201/32, 192.0.2.202/32, 192.0.2.203/32, 192.0.2.204/32. These PE's advertised their VPN routes with next-hop set on their respective loopback address.

The operator configures A, B, C with respective loopbacks 192.0.2.1/32, 192.0.2.2/32, 192.0.2.3/32.

The operator configures PE2, A, B, C and PE4 with SRGB [100, 300].

The operator attaches the respective Node-SIDs 202, 101, 102, 103 and 204 to the loopbacks of nodes PE2, A, B, C and PE4. The Node-SID's are configured to request penultimate-hop-popping.

PE1, A, B, C and PE3 are LDP capable.

PE1 and PE3 are not SR capable.

PE3 sends an ODD VPN route to PE1 with next-hop 192.0.2.203 and VPN label 10001.

From an LDP viewpoint: PE1 received an LDP label binding (1037) for FEC 192.0.2.203/32 from its nhop A. A received an LDP label binding (2048) for that FEC from its nhop B. B received an LDP label binding (3059) for that FEC from its nhop C. C received implicit-null LDP binding from its next-hop PE3.

As a result, PE1 sends its traffic to the ODD service route advertised by PE3 to next-hop A with two labels: the top label is 1037 and the bottom label is 10001. A swaps 1037 with 2048 and forwards to B. B swaps 2048 with 3059 and forwards to C. C pops 3059 and forwards to PE3.

PE4 sends an EVEN VPN route to PE2 with next-hop 192.0.2.204 and VPN label 10002.

From an SR viewpoint: PE1 maps the IGP route 192.0.2.204/32 onto Node-SID 204; A swaps 204 with 204 and forwards to B; B swaps 204 with 204 and forwards to C; C pops 204 and forwards to PE4.

As a result, PE2 sends its traffic to the VPN service route advertised by PE4 to next-hop A with two labels: the top label is 204 and the bottom label is 10002. A swaps 204 with 204 and forwards to B. B swaps 204 with 204 and forwards to C. C pops 204 and forwards to PE4.

The two modes of MPLS tunneling co-exist.

The ODD service is tunneled from PE1 to PE3 through a continuous LDP LSP traversing A, B and C.

The EVEN service is tunneled from PE2 to PE4 through a continuous SR node segment traversing A, B and C.

## 2.1. MPLS2MPLS co-existence

We want to highlight that several MPLS2MPLS entries can be installed in the data plane for the same prefix.

Let us examine A's MPLS forwarding table as an example:

Incoming label: 1037

- outgoing label: 2048
- outgoing nhop: B
- Note: this entry is programmed by LDP for 192.0.2.203/32

Incoming label: 203

- outgoing label: 203
- outgoing nhop: B
- Note: this entry is programmed by SR for 192.0.2.203/32

These two entries can co-exist because their incoming label is unique. The uniqueness is guaranteed by the label manager allocation rules.

The same applies for the MPLS2IP forwarding entries.

## 2.2. IP2MPLS co-existence

By default, we propose that if both LDP and SR propose an IP2MPLS entry for the same IP prefix, then the LDP route is selected.

A local policy on a router MUST allow to prefer the SR-provided IP2MPLS entry.

Note that this policy may be locally defined. There is no requirement that all routers use the same policy.

## 3. Migration from LDP to SR

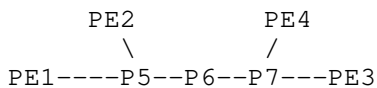


Figure 2: Migration

Several migration techniques are possible. We describe one technique inspired by the commonly used method to migrate from one IGP to another.

T0: all the routers run LDP. Any service is tunneled from an ingress PE to an egress PE over a continuous LDP LSP.

T1: all the routers are upgraded to SR. They are configured with the SRGB range [100, 300]. PE1, PE2, PE3, PE4, P5, P6 and P7 are respectively configured with the node segments 101, 102, 103, 104, 105, 106 and 107 (attached to their service-recurring loopback).

At this time, the service traffic is still tunneled over LDP LSP. For example, PE1 has an SR node segment to PE3 and an LDP LSP to PE3 but by default, as seen earlier, the LDP IP2MPLS encapsulation is preferred.

T2: the operator enables the local policy at PE1 to prefer SR IP2MPLS encapsulation over LDP IP2MPLS.

The service from PE1 to any other PE is now riding over SR. All other service traffic is still transported over LDP LSP.

T3: gradually, the operator enables the preference for SR IP2MPLS encapsulation across all the edge routers.

All the service traffic is now transported over SR. LDP is still operational and services could be reverted to LDP.

However, any traffic switched through LDP entries will still suffer from LDP-IGP synchronization.

T4: LDP is unconfigured from all routers.

#### 4. SR and LDP Interworking

In this section, we analyze a use-case where SR is available in one part of the network and LDP is available in another part. We describe how a continuous MPLS tunnel can be built throughout the network.

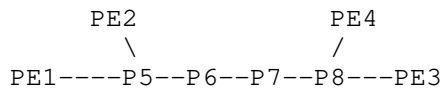


Figure 3: SR and LDP Interworking

Let us analyze the following example:

P6, P7, P8, PE4 and PE3 are LDP capable.

PE1, PE2, P5 and P6 are SR capable. PE1, PE2, P5 and P6 are configured with SRGB (100, 200) and respectively with node segments 101, 102, 105 and 106.

A service flow must be tunneled from PE1 to PE3 over a continuous MPLS tunnel encapsulation. We need SR and LDP to interwork.

##### 4.1. LDP to SR

In this section, we analyze a right-to-left traffic flow.

PE3 has learned a service route whose nhop is PE1. PE3 has an LDP label binding from the nhop P8 for the FEC "PE1". Hence PE3 sends its service packet to P8 as per classic LDP behavior.

P8 has an LDP label binding from its nhop P7 for the FEC "PE1" and hence P8 forwards to P7 as per classic LDP behavior.



P7 has an LDP label binding from its nhop P6 for the FEC "PE1" and hence P7 forwards to P6 as per classic LDP behavior.

P6 does not have an LDP binding from its nhop P5 for the FEC "PE1". However P6 has an SR node segment to the IGP route "PE1". Hence, P6 forwards the packet to P5 and swaps its local LDP-label for FEC "PE1" by the equivalent node segment (i.e. 101).

P5 pops 101 (assuming PE1 advertised its node segment 101 with the penultimate-pop flag set) and forwards to PE1.

PE1 receives the tunneled packet and processes the service label.

The end-to-end MPLS tunnel is built from an LDP LSP from PE3 to P6 and the related node segment from P6 to PE1.

#### 4.2. SR to LDP

In this section, we analyze the left-to-right traffic flow.

We assume that the operator configures P5 to act as a Segment Routing Mapping Server (SRMS) and advertise the following mappings: (P7, 107), (P8, 108), (PE3, 103) and (PE4, 104).

These mappings are advertised as Remote-Binding SID with Flag TBD.

The mappings advertised by an SR mapping server result from local policy information configured by the operator. If PE3 had been SR capable, the operator would have configured PE3 with node segment 103. Instead, as PE3 is not SR capable, the operator configures that policy at the SRMS and it is the latter which advertises the mapping. Multiple SRMS servers can be provisioned in a network for redundancy.

The mapping server advertisements are only understood by the SR capable routers. The SR capable routers install the related node segments in the MPLS data plane exactly like if the node segments had been advertised by the nodes themselves.

For example, PE1 installs the node segment 103 with nhop P5 exactly as if PE3 had advertised node segment 103.

PE1 has a service route whose nhop is PE3. PE1 has a node segment for that IGP route: 103 with nhop P5. Hence PE1 sends its service packet to P5 with two labels: the bottom label is the service label and the top label is 103.

P5 swaps 103 for 103 and forwards to P6.

P6's next-hop for the IGP route "PE3" is not SR capable (P7 does not advertise the SR capability). However, P6 has an LDP label binding from that next-hop for the same FEC (e.g. LDP label 1037). Hence, P6 swaps 103 for 1037 and forwards to P7.

P7 swaps this label with the LDP-label received from P8 and forwards to P8.

P8 pops the LDP label and forwards to PE3.

PE3 receives the tunneled packet and processes the service label.

The end-to-end MPLS tunnel is built from an SR node segment from PE1 to P6 and an LDP LSP from P6 to PE3.

Note: SR mappings advertisements cannot set Penultimate Hop Popping. In the previous example, P6 requires the presence of the segment 103 such as to map it to the LDP label 1037. For that reason, the P flag available in the Prefix-SID is not available in the Remote-Binding SID.

#### 5. Leveraging SR benefits for LDP-based traffic

SR can be deployed such as to enhance LDP transport. The SR deployment can be limited to the network region where the SR benefits are most desired.

In Figure 4, let us assume:

All link costs are 10 except FG which is 30.

All routers are LDP capable.

X, Y and Z are PE's participating to an important service S.

The operator requires 50msec link-based FRR for service S.

A, B, C, D, E, F and G are SR capable.

X, Y, Z are not SR capable, e.g. as part of a staged migration from LDP to SR, the operator deploys SR first in a sub-part of the network and then everywhere.

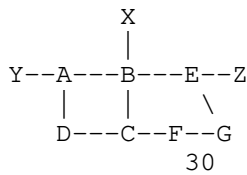


Figure 4: Leveraging SR benefits for LDP-based-traffic

The operator would like to resolve the following issues:

To protect the link BA along the shortest-path of the important flow XY, B requires an RLFA repair tunnel to D and hence a directed LDP session from B to D. The operator does not like these dynamically established multi-hop LDP sessions and would seek to eliminate them.

There is no LFA/RLFA solution to protect the link BE along the shortest path of the important flow XZ. The operator wants a guaranteed link-based FRR solution.

The operator can meet these objectives by deploying SR only on A, B, C, D, E and F:

The operator configures A, B, C, D, E, F and G with SRGB (100, 200) and respective node segments 101, 102, 103, 104, 105, 106 and 107.

The operator configures D as an SR Mapping Server with the following policy mapping: (X, 201), (Y, 202), (Z, 203).

Each SR node automatically advertises local adjacency segment for its IGP adjacencies. Specifically, F advertises adjacency segment 9001 for its adjacency FG.

A, B, C, D, E, F and G keep their LDP capability and hence the flows XY and XZ are transported over end-to-end LDP LSP's.

For example, LDP at B installs the following MPLS data plane entries:

Incoming label: local LDB label bound by B for FEC Y  
 Outgoing label: LDP label bound by A for FEC Y  
 Outgoing nhop: A

Incoming label: local LDB label bound by B for FEC Z  
 Outgoing label: LDP label bound by E for FEC Z  
 Outgoing nhop: E

The novelty comes from how the backup chains are computed for these LDP-based entries. While LDP labels are used for the primary nhop and outgoing labels, SR information is used for the FRR construction. In steady state, the traffic is transported over LDP LSP. In transient FRR state, the traffic is backup thanks to the SR enhanced capabilities.

This helps meet the requirements of the operator:

- Eliminate directed LDP session.

- Guaranteed FRR coverage.

- Keep the traffic over LDP LSP in steady state.

- Partial SR deployment only where needed.

#### 5.1. Eliminating Directed LDP Session

B's MPLS entry to Y becomes:

- Incoming label: local LDB label bound by B for FEC Y  
Outgoing label: LDP label bound by A for FEC Y  
Backup outgoing label: SR node segment for Y {202}  
Outgoing nhop: A  
Backup nhop: repair tunnel: node segment to D {104}  
with outgoing nhop: C

In steady-state, X sends its Y-destined traffic to B with a top label which is the LDP label bound by B for FEC Y. B swaps that top label for the LDP label bound by A for FEC Y and forwards to A. A pops the LDP label and forwards to Y.

Upon failure of the link BA, B swaps the incoming top-label with the node segment for Y (202) and sends the packet onto a repair tunnel to D (node segment 104). Thus, B sends the packet to C with the label stack {104, 202}. C pops the node segment 104 and forwards to D. D swaps 202 for 202 and forwards to A. A's nhop to Y is not SR capable and hence A swaps the incoming node segment 202 to the LDP label announced by its next-hop (in this case, implicit null).

After IGP convergence, B's MPLS entry to Y will become:

- Incoming label: local LDB label bound by B for FEC Y  
Outgoing label: LDP label bound by C for FEC Y  
Outgoing nhop: C

And the traffic XY travels again over the LDP LSP.

Conclusion: the operator has eliminated its first problem: directed LDP sessions are no longer required and the steady-state traffic is still transported over LDP. The SR deployment is confined to the area where these benefits are required.

## 5.2. Guaranteed FRR coverage

B's MPLS entry to Z becomes:

- Incoming label: local LDB label bound by B for FEC Z
- Outgoing label: LDP label bound by E for FEC Z
- Backup outgoing label: SR node segment for Z {203}
- Outgoing nhop: E
- Backup nhop: repair tunnel to G: {106, 9001}

G is reachable from B via the combination of a node segment to F {106} and an adjacency segment FG {9001}

Note that {106, 107} would have equally work. Indeed, in many case, P's shortest path to Q is over the link PQ. The adjacency segment from P to Q is required only in very rare topologies where the shortest-path from P to Q is not via the link PQ.

In steady-state, X sends its Z-destined traffic to B with a top label which is the LDP label bound by B for FEC Z. B swaps that top label for the LDP label bound by E for FEC Z and forwards to E. E pops the LDP label and forwards to Z.

Upon failure of the link BE, B swaps the incoming top-label with the node segment for Z (203) and sends the packet onto a repair tunnel to G (node segment 106 followed by adjacency segment 9001). Thus, B sends the packet to C with the label stack {106, 9001, 203}. C pops the node segment 106 and forwards to F. F pops the adjacency segment 9001 and forwards to G. G swaps 203 for 203 and forwards to E. E's nhop to Z is not SR capable and hence E swaps the incoming node segment 203 for the LDP label announced by its next-hop (in this case, implicit null).

After IGP convergence, B's MPLS entry to Z will become:

- Incoming label: local LDB label bound by B for FEC Z
- Outgoing label: LDP label bound by C for FEC Z
- Outgoing nhop: C

And the traffic XZ travels again over the LDP LSP.

Conclusion: the operator has eliminated its second problem: guaranteed FRR coverage is provided. The steady-state traffic is still transported over LDP. The SR deployment is confined to the area where these benefits are required.

## 6. Inter-AS Option C, Carrier's Carrier and Seamless MPLS

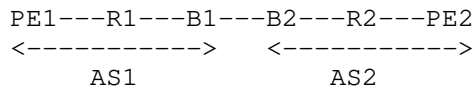


Figure 5: Inter-AS Option C

In Inter-AS Option C [RFC4364], B2 advertises to B1 a BGP3107 route for PE2 and B1 reflects it to its internal peers, such as PE1. PE1 learns from a service route reflector a service route whose nhop is PE2. PE1 resolves that service route on the BGP3107 route to PE2. That BGP3107 route to PE2 is itself resolved on the AS1 IGP route to B1.

If AS1 operates SR, then the tunnel from PE1 to B1 is provided by the node segment from PE1 to B1.

PE1 sends a service packet with three labels: the top one is the node segment to B1, the next-one is the BGP3107 label provided by B1 for the route "PE2" and the bottom one is the service label allocated by PE2.

The same straightforward SR applicability is derived for CsC and Seamless MPLS ([I-D.ietf-mpls-seamless-mpls]).

## 7. IANA Considerations

TBD

## 8. Manageability Considerations

TBD

## 9. Security Considerations

TBD

## 10. Acknowledgements

We would like to thank Pierre Francois and Ruediger Geib 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.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

### 11.2. Informative References

- [I-D.ietf-mpls-seamless-mpls]  
Leymann, N., Decraene, B., Filsfils, C., Konstantynowicz, M., and D. Steinberg, "Seamless MPLS Architecture", draft-ietf-mpls-seamless-mpls-07 (work in progress), June 2014.
- [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.
- [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-00 (work in progress), December 2014.

### 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  
Individual Contributor

Email: edward.crabbe@gmail.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: February 1, 2015

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.  
July 31, 2014

Segment Routing with MPLS data plane  
draft-filsfils-spring-segment-routing-mpls-03

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. 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.

Segment Routing can be directly applied to the MPLS architecture with no change in the forwarding plane. This drafts describes how Segment Routing operates on top of the MPLS data plane.

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 February 1, 2015.

## 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. Illustration . . . . .	3
3. MPLS Instantiation of Segment Routing . . . . .	4
4. IGP Segments Examples . . . . .	6
4.1. Example 1 . . . . .	7
4.2. Example 2 . . . . .	7
4.3. Example 3 . . . . .	7
4.4. Example 4 . . . . .	7
4.5. Example 5 . . . . .	8
5. Other Examples of MPLS Segments . . . . .	8
5.1. LDP LSP segment combined with IGP segments . . . . .	8
5.2. RSVP-TE LSP segment combined with IGP segments . . . . .	9
6. Segment List History . . . . .	10
7. IANA Considerations . . . . .	10

8. Manageability Considerations . . . . .	11
9. Security Considerations . . . . .	11
10. Acknowledgements . . . . .	11
11. References . . . . .	11
11.1. Normative References . . . . .	11
11.2. Informative References . . . . .	11
Authors' Addresses . . . . .	12

## 1. Introduction

The Segment Routing architecture [I-D.filsfils-spring-segment-routing] can be directly applied to the MPLS architecture with no change in the MPLS forwarding plane. This drafts describes how Segment Routing operates on top of the MPLS data plane.

The Segment Routing use cases are described in in [I-D.filsfils-spring-segment-routing-use-cases].

Link State protocol extensions for Segment Routing are described in [I-D.ietf-isis-segment-routing-extensions], [I-D.ietf-ospf-segment-routing-extensions] and [I-D.psenak-ospf-segment-routing-ospfv3-extension].

## 2. Illustration

Segment Routing, applied to the MPLS data plane, offers the ability to tunnel services (VPN, VPLS, VPWS) from an ingress PE to an egress PE, without any other protocol than ISIS or OSPF ([I-D.ietf-isis-segment-routing-extensions] and [I-D.ietf-ospf-segment-routing-extensions]). LDP and RSVP-TE signaling protocols are not required.

Note that [I-D.filsfils-spring-segment-routing-ldp-interop] documents SR co-existence and interworking with other MPLS signaling protocols, if present in the network during a migration, or in case of non-homogeneous deployments.

The operator only needs to allocate one node segment per PE and the SR IGP control-plane automatically builds the required MPLS forwarding constructs from any PE to any PE.

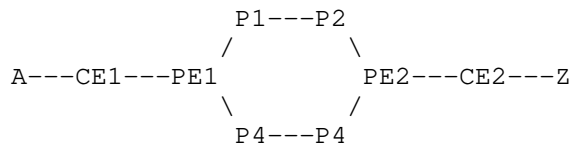


Figure 1: IGP-based MPLS Tunneling

In Figure 1 above, the four nodes A, CE1, CE2 and Z are part of the same VPN.

PE2 advertises (in the IGP) a host address 192.0.2.2/32 with its attached node segment 102.

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 192.0.2.2 (PE2 loopback address).

PE1 installs the VPN prefix Z in the appropriate VRF and resolves the next-hop onto the node segment 102. Upon receiving a packet from A destined to Z, PE1 pushes two labels onto the packet: the top label is 102, the bottom label is LZ. 102 identifies the node segment to PE2 and hence transports the packet along the ECMP-aware shortest-path to PE2. PE2 then processes the VPN label LZ and forwards the packet to CE2.

Supporting MPLS services (VPN, VPLS, VPWS) with SR has the following benefits:

Simple operation: one single intra-domain protocol to operate: the IGP. No need to support IGP synchronization extensions as described in [RFC5443] and [RFC6138].

Excellent scaling: one Node-SID per PE.

### 3. MPLS Instantiation of Segment Routing

MPLS instantiation of Segment Routing fits in the MPLS architecture as defined in [RFC3031] both from a control plane and forwarding plane perspective:

- o From a control plane perspective [RFC3031] does not mandate a single signaling protocol. Segment Routing proposes to use the Link State IGP as its use of information flooding fits very well with label stacking on ingress.
- o From a forwarding plane perspective, Segment Routing does not require any change to the forwarding plane.

When applied to MPLS, a Segment is a LSP and the 20 right-most bits of the segment are encoded as a label. This implies that, in the MPLS instantiation, the SID values are allocated within a reduced 20-bit space out of the 32-bit SID space.

The notion of indexed global segment fits the MPLS architecture [RFC3031] as the absolute value allocated to any segment (global or local) can be managed by a local allocation process (similarly to other MPLS signaling protocols).

If present, SR can coexist and interwork with LDP and RSVP [I-D.filsfils-spring-segment-routing-ldp-interop].

The source routing model described in [I-D.filsfils-spring-segment-routing] is inherited from the ones proposed by [RFC1940] and [RFC2460]. The source routing model offers the support for explicit routing capability.

Contrary to RSVP-based explicit routes where tunnel midpoints maintain states, SR-based explicit routes only require per-flow states at the ingress edge router where the traffic engineer policy is applied.

Contrary to RSVP-based explicit routes which consist in non-ECMP circuits (similar to ATM/FR), SR-based explicit routes can be built as list of ECMP-aware node segments and hence ECMP-aware traffic engineering is natively supported by SR.

When Segment Routing is instantiated over the MPLS data plane the following applies:

A list of segments is represented as a stack of labels.

The active segment is the top label.

The CONTINUE operation is implemented as an MPLS swap operation. When the same Segment Routing Global Block (SRGB, defined in [I-D.filsfils-spring-segment-routing] is used throughout the SR domain, the outgoing label value is equal to the incoming label value . Else, the outgoing label value is [SRGB(next\_hop)+index]

The NEXT operation is implemented as an MPLS pop operation.

The PUSH operation is implemented as an MPLS push of a label stack.

In conclusion, there are no changes in the operations of the data-plane currently used in MPLS networks.

#### 4. IGP Segments Examples

Assuming the network diagram of Figure 2 and the IP address and IGP Segment allocation of Figure 3, the following examples can be constructed.

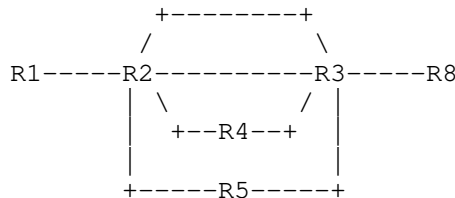


Figure 2: IGP Segments - Illustration

IP address allocated by the operator:	
	192.0.2.1/32 as a loopback of R1
	192.0.2.2/32 as a loopback of R2
	192.0.2.3/32 as a loopback of R3
	192.0.2.4/32 as a loopback of R4
	192.0.2.5/32 as a loopback of R5
	192.0.2.8/32 as a loopback of R8
	198.51.100.9/32 as an anycast loopback of R4
	198.51.100.9/32 as an anycast loopback of R5
SRGB defined by the operator as 1000-5000	
Global IGP SID allocated by the operator:	
	1001 allocated to 192.0.2.1/32
	1002 allocated to 192.0.2.2/32
	1003 allocated to 192.0.2.3/32
	1004 allocated to 192.0.2.4/32
	1008 allocated to 192.0.2.8/32
	2009 allocated to 198.51.100.9/32
Local IGP SID allocated dynamically by R2	
	for its "north" adjacency to R3: 9001
	for its "north" adjacency to R3: 9003
	for its "south" adjacency to R3: 9002
	for its "south" adjacency to R3: 9003

Figure 3: IGP Address and Segment Allocation - Illustration

#### 4.1. 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.

#### 4.2. 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.

#### 4.3. 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.

#### 4.4. 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.

#### 4.5. 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 .

### 5. Other Examples of MPLS Segments

In addition to the IGP segments previously described, the SPRING source routing policy applied to MPLS can include MPLS LSP's signaled by LDP, RSVPTE and BGP. The list of examples is non exhaustive. Other form of segments combination can be instantiated through Segment Routing (e.g.: RSVP LSPs combined with LDP or IGP or BGP LSPs).

#### 5.1. LDP LSP segment combined with IGP segments

The example illustrates a segment-routing policy including IGP segments and LDP LSP segments.

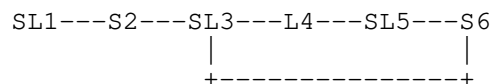


Figure 4: LDP LSP segment combined with IGP segments

We assume that:

- o All links have an IGP cost of 1 except SL3-S6 link which has cost 2.
- o All nodes are in the same IGP area.

- o Nodes SL1, S2, SL3, SL5 and S6 are IGP-SR capable.
- o SL3 and S6 have, respectively, index 3 and 6 assigned to them.
- o All SR nodes have the same SRGB consisting of: [1000, 1999]
- o SL1, SL3, L4 and SL5 are LDP capable.
- o SL1 has a directed LDP session with SL3 and is able to retrieve the SL3 local LDP mapping for FEC SL5: 35
- o The following source-routed policy is defined in S1 for the traffic destined to S6: use path SL1-S2-SL3-L4-SL5-S6 (instead of shortest-path SL1-S2-SL3-S6).

This is realized by programming the following segment-routing policy at S1: for traffic destined to S6, push the ordered segment list: {1003, 35, 1006}, where:

- o 1003 gets the packets from S1 to SL3 via S2.
- o 35 gets the packets from SL3 to SL5 via L4.
- o 1006 gets the packets from SL5 to S6.

The above allows to steer the traffic into path SL1-S2-SL3-L4-SL5-S6 instead of the shortest path SL1-S2-SL3-S6.

## 5.2. RSVP-TE LSP segment combined with IGP segments

The example illustrates a segment-routing policy including IGP segments and RSVP-TE LSP segments.

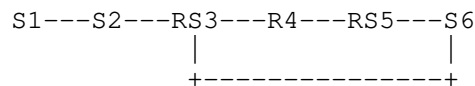


Figure 5: RSVP-TE LSP segment combined with IGP segments

We assume that:

- o All links have an IGP cost of 1 except link RS3-S6 which has cost 2.
- o All nodes are IGP-SR capable except R4.
- o RS3 and R6 have, respectively, index 3 and 6 assigned to them.

- o All SR nodes have the same SRGB consisting of: [1000, 1999]
- o RS3, R4 and RS5 are RSVP-TE capable.
- o An RSVP-TE LSP has been provisioned from RS3 to RS5 via R4.
- o RS3 allocates a binding SID (with value of 135) for this RSVP-TE LSP and signals it in the igp.
- o The following source-routed policy is defined at S1 for the traffic destined to S6: use path S1-S2-RS3-R4-RS5-S6 instead of shortest-path S1-S2-RS3-S6.

This is realized by programming the following segment-routing policy at S1: - for traffic destined to S6, push the ordered segment list: {1003, 135, 1006}, where:

- o 1003 gets the packets from S1 to RS3 via S2.
- o 135 gets the packets from RS3 into the RSVP-TE LSP to RS5 via R4.
- o 1006 gets the packets from RS5 to S6.

The above allows to steer the traffic into path S1-S2-RS3-R4-RS5-S6 instead of the shortest path S1-S2-RS3-S6.

## 6. Segment List History

In the abstract SR routing model [I-D.filsfils-spring-segment-routing], any node N along the journey of the packet is able to determine where the packet P entered the SR domain and where it will exit. The intermediate node is also able to determine the paths from the ingress edge router to itself, and from itself to the egress edge router.

In the MPLS instantiation, as the packet travels through the SR domain, the stack is depleted and the segment list history is gradually lost.

Future version of this document will describe how this information can be preserved in MPLS domains.

## 7. IANA Considerations

TBD

## 8. Manageability Considerations

TBD

## 9. Security Considerations

TBD

## 10. Acknowledgements

## 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.

## 11.2. Informative 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.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-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.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-02 (work in progress), June 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-01 (work in progress), July 2014.
- [I-D.psenak-ospf-segment-routing-ospfv3-extension]  
Psenak, P., Previdi, S., Filsfils, C., Gredler, H.,  
Shakir, R., Henderickx, W., and J. Tantsura, "OSPFv3  
Extensions for Segment Routing", draft-psenak-ospf-  
segment-routing-ospfv3-extension-02 (work in progress),  
July 2014.
- [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.
- [RFC5443] Jork, M., Atlas, A., and L. Fang, "LDP IGP  
Synchronization", RFC 5443, March 2009.
- [RFC6138] Kini, S. and W. Lu, "LDP IGP Synchronization for Broadcast  
Networks", RFC 6138, February 2011.
- [draft-filsfils-rtgwg-segment-routing-ldp-interop-00]  
Filsfils, C. and S. Previdi, "Segment Routing  
interoperability with LDP", 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: January 2, 2014

Pierre Francois  
IMDEA Networks  
Clarence Filsfils  
Ahmed Bashandy  
Stefano Previdi  
Cisco Systems, Inc.  
Bruno Decraene  
Orange  
July 1, 2013

Segment Routing Fast Reroute  
draft-francois-sr-frr-00

Abstract

This document presents a Fast Reroute approach aimed at providing link protection of nodal and adjacency segments to the Segment Routing framework. This FRR behavior builds on proven IP-FRR concepts being LFAs, remote LFAs (RLFA), and remote LFAs with directed forwarding (DLFA). We describe their implementation using SR segments. We then analyze the benefits brought by Segment Routing to the scalability of such IP-FRR approaches.

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 2, 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. Protection lists . . . . .	4
2.1. LFA based protection list . . . . .	4
2.2. RLFA based protection list . . . . .	4
2.3. DLFA based protection list . . . . .	5
3. Protecting segments . . . . .	5
3.1. The active segment is a node segment . . . . .	5
3.2. The active segment is an adjacency segment . . . . .	6
3.2.1. Protecting [Adjacency, Adjacency] segment lists . . . . .	6
3.2.2. Protecting [Adjacency, Nodal] segment lists . . . . .	6
4. SR FRR benefits in LDP environments . . . . .	7
4.1. Eliminating Directed LDP Sessions . . . . .	9
4.2. Guaranteed FRR coverage . . . . .	9
5. References . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

Segment Routing aims at supporting services with tight SLA guarantees [1]. Acknowledging this fact, this document provides local repair mechanisms capable of restoring end-to-end connectivity in case of a sudden failure of a link. The FRR behavior builds on proven IP-FRR concepts; we leverage LFAs, remote LFAs (RLFA), and remote LFAs with directed forwarding (DLFA) [2].

In the SR context, not all flows are being routed along the shortest paths defined by the IGP, but also along explicit paths containing Adjacency Segments. We thus accommodate the IP-FRR behavior for SR Adjacency Segments.

Through the document, we will observe that performing FRR with SR has the following benefits:

- The simplicity properties of LFA FRR [3] are preserved.
- The capacity planning properties are preserved [3]. Unlike SDH and other FRR solutions, the repaired packet does not go back to the next-hop or next-next-hop but uses shortest-path forwarding from a much closer release point.
- The RLFA operation is simplified: dynamically established directed LDP sessions to the repair nodes are no longer required.
- The scalable support for DLFA provides guaranteed coverage for symmetric networks, i.e. networks configured with symmetric link metrics: the repair tunnel in a symmetric network can be encoded efficiently with only two segments. We will observe that only one segment is needed in most cases.

A future version of this document will analyze the protection upon node failure.

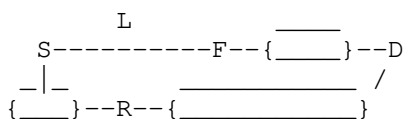


Figure 1: Link Protection

We use Figure 1 to illustrate the three objectives that have to be met when implementing SR FRR.

First, the protecting router S needs to find a detour path around the protected link. Intuitively, the Point of Local Repair (PLR) needs to find a node R (a repair node) that is capable of safely forwarding the traffic affected by the failure of the protected link L. We leverage the algorithms defined in the IP-FRR framework to achieve this first goal, as explained in Section 2.

Second, S must ensure proper forwarding behavior once the packet reaches the repair node R. We define the segment operations to be applied by the protecting node to ensure consistency with the forwarding state of the repair node in Section 3.

We will observe, in Section 4, that the MPLS instantiation of SR improves the scalability and operation of the FRR solution by not requiring multi-hop LDP sessions to distant repair nodes.

## 2. Protection lists

The protection list is the list of segments encoding a detour path from the protecting node S to the repair node R, avoiding the protected link L. In this section, we define how to encode the LFA, RLFA, and DLFA repair paths with protection lists. These protection lists contain at most two segments.

### 2.1. LFA based protection list

According to the LFA FRR approach, if a path to a destination D from a neighbor N of S does not contain S (i.e. N is a loop-free alternate of S for the failure of link S-F), then S can pre-install a repair forwarding information, in order to deviate the packet to N upon the failure of S-F.

In the case of LFA applicability, the SR protection list is thus empty. All what a protecting router S needs to do is to send the protected packet as is to its LFA neighbor N.

### 2.2. RLFA based protection list

If there is no such LFA neighbor, then S may be able to create a virtual LFA by using a tunnel to carry the packet to a point in the network that is not a direct neighbor of S, and from which the packet will be delivered to the destination without looping back to S. The Remote LFA proposal [4] calls such a tunnel a repair tunnel. The tail-end of this tunnel (R in figure 1) is called a "remote LFA" or a "PQ node". We refer to the RLFA document for the definitions of the P and Q sets.

In the case of RLFA applicability for the protection of a segment, the protection list is made of a nodal segment to the PQ node. It thus matches [nodal(PQ), ...]

### 2.3. DLFA based protection list

There are some cases where there is no remote LFA coverage for some links/destinations, due to topological properties in the neighborhood of the protecting node. If there is no such RLFA PQ node, we propose to use a Directed LFA (DLFA) repair tunnel to a Q node that is adjacent to the P space [5].

In the case of applicability of RLFA with directed forwarding (DLFA), the protection list is made of a nodal segment to the P node followed by an Adjacency segment to the Q node. It thus matches [nodal(P), Adj(P-->Q), ...]

In networks with symmetric IGP metrics (the metric of a link AB is the same as the metric of the reverse link BA), we can prove that either the P and the Q sets intersect or there is at least one P node that is adjacent to a Q node. Thanks to the DLFA extension, we thus have a guaranteed LFA-based FRR technique for any network with symmetric IGP metrics.

Future versions of the document will describe the solutions leveraging SR capabilities to provide guaranteed FRR applicability in any IGP topology.

## 3. Protecting segments

In this section, we explain how a protecting router S processes the active segment of a packet upon the failure of the primary adjacency along which the packet should be forwarded. The behavior depends on the type of active segment to be protected.

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

The definition of the protection of a nodal segment is a direct translation of IP-FRR behaviors into the SR terminology. That is, traffic for nodal segment D will be rerouted to a safe node R whose shortest paths for D do not contain the failed component.

As nodal segments semantics are known by all nodes of the domain, no specific signaling needs to be done to let R correctly process the detoured packet. A packet whose active segment matches [nodal(D),...], arriving at a protecting node S will leave S with a segment list matching [PS(R), nodal(D),...]. The actual value used

to encode `nodal(D)` is set by `S` based on the `SRSB` obtained from the IGP [1].

`PS(R)` is the computed Protection list to reach `R`, as discussed in section 2, and depends on the available type of protection: per-prefix LFA, Remote LFA or Directed LFA. The packet will follow the detour path defined by `PS(R)`, and will finally reach `R`. When reaching `R`, the active segment of the packet is `nodal(D)`, and the packet resumes its course along the original segment list.

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

The operator may forbid protection of an adjacency segment by policy (?-Flag in [ISIS]/[OSPF]). For example, this is useful when the operator prefers an end-to-end protection mechanism triggered by the source of a multi-hop transport conduit.

We define hereafter the FRR behavior applied by `S` for any packet received with an active segment `L` 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 nodal segment.

#### 3.2.1. 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(L)` and then two consecutive "PUSH" operations: first it pushes a nodal Segment for `F`, and then it pushes a protection list allowing to reach `F` while bypassing `L`.

Upon failure of `L`, a packet reaching `S` with a segment list matching `[adj(L),adj(M),...]` will thus leave `S` with a segment list matching `[PS(F),nodal(F),adj(M)]`.

The protection list `PS(F)` will define the course of the packet from `S` to `F`, and `F` will resume the the course of the original segment list, receiving it with an active segment list matching `[nodal(F),adj(M),...]`.

#### 3.2.2. Protecting [Adjacency, Nodal] segment lists

If the next segment in the stack is a nodal segment, say for node `T`, the packet segment list matches `[adj(L),nodal(T),...]`.

A first solution would consist in steering the packet back to `F` while

avoiding L, similarly to the previous case. To do so, S applies a "NEXT" operation on Adj(L) and then two consecutive "PUSH" operations: first it pushes a nodal Segment for F, and then it pushes a protection list allowing to reach F while bypassing L.

Upon failure of L, a packet reaching S with a segment list matching  $[\text{adj}(L), \text{nodal}(T), \dots]$  will thus leave S with a segment list matching  $[\text{PS}(F), \text{nodal}(F), \text{nodal}(T)]$ .

Another solution is to not steer the packet back via F. In this case, S just needs to apply a "NEXT" operation on the Adjacency segment related to L, and push a protection segment list redirecting the traffic to a node R, capable of whose path to nodal segment T is not affected by the failure.

Upon failure of L, packets reaching S with a segment list matching [adj(L), nodal(T), ...], would leave S with a segment list matching [PS(R), nodal(T), ...].

#### 4. SR FRR benefits in LDP environments

In this section, we describe the operational and scaling benefits of SR when used to implement RLFA and DLFA protection for LDP-based transport. We will also observe that a partial SR deployment, limited to the network region where the SR benefits are most desired, already provides the mentioned scaling benefits.

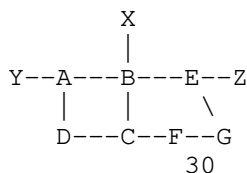


Figure 2: Leveraging SR benefits for LDP-based traffic

In Figure 2, let us assume:

- All link costs are 10 except FG which is 30.
- All routers are LDP capable.
- X, Y and Z are PEs participating to an important service S.
- The operator requires 50msec link-based FRR for service S.
- A, B, C, D, E, F and G are SR capable.

- X, Y, Z are not SR capable. As part of a staged migration from LDP to SR, the operator deploys SR first in a sub-part of the network and then everywhere.

The operator would like to resolve the following issues:

- to protect the link BA along the shortest-path of the important flow XY, B requires an RLFA repair tunnel to D and hence a directed LDP session from B to D. The operator does not like these dynamically established multi-hop LDP sessions and would seek to eliminate them.
- there is no LFA/RLFA solution to protect the link BE along the shortest path of the important flow XZ. The operator wants a guaranteed link-based FRR solution.

The operator can meet these objectives by deploying SR only on A, B, C, D, E and F:

- The operator configures A, B, C, D, E, F and G with SRGB [100, 200] and respective node segments 101, 102, 103, 104, 105, 106 and 107.
- The operator configures D as an SR Mapping Server with the following policy mapping: (X, 201), (Y, 202), (Z, 203).
- Each SR node automatically advertises local adjacency segment for its IGP adjacencies. Specifically, F advertises adjacency segment 9001 for its adjacency FG.

A, B, C, D, E, F and G keep their LDP capability and hence the flows XY and XZ are transported over end-to-end LDP LSP's.

For example, LDP at B installs the following MPLS dataplane entries:

- Incoming label: local LDB label bound by B for FEC Y
  - o Outgoing label: LDP label bound by A for FEC Y
  - o Outgoing nhop: A
- Incoming label: local LDB label bound by B for FEC Z
  - o Outgoing label: LDP label bound by E for FEC Z
  - o Outgoing nhop: E

The novelty comes from how the backup chains are computed for these LDP-based entries. While LDP labels are used for the primary nhop and outgoing labels, SR information is used for the FRR construction. In steady state, the traffic is transported over LDP LSP. In transient FRR state, the traffic is backed up thanks to the SR capabilities.

This helps meet the requirements of the operator:

- Eliminate directed LDP session
- Guaranteed FRR coverage

- Keep the traffic over LDP LSP in steady state
- Partial SR deployment only where needed

#### 4.1. Eliminating Directed LDP Sessions

B's MPLS entry to Y becomes:

- Incoming label: local LDB label bound by B for FEC Y
  - o Outgoing label: LDP label bound by A for FEC Y
  - Backup outgoing label: SR node segment for Y {202}
  - o Outgoing nhop: A
  - Backup nhop: repair tunnel: node segment to D {104} with outgoing nhop: C

In steady-state, X sends its Y-destined traffic to B with a top label which is the LDP label bound by B for FEC Y. B swaps that top label for the LDP label bound by A for FEC Y and forwards to A. A pops the LDP label and forwards to Y.

Upon failure of the link BA, B swaps the incoming top-label with the node segment for Y (202) and sends the packet onto a repair tunnel to D (node segment 104). Thus, B sends the packet to C with the label stack {104, 202}. C pops the node segment 104 and forwards to D. D swaps 202 for 202 and forwards to A. A's nhop to Y is not SR capable and hence A swaps the incoming node segment 202 to the LDP label announced by its next-hop (in this case, implicit null).

After IGP convergence, B's MPLS entry to Y will become:

- Incoming label: local LDB label bound by B for FEC Y
- Outgoing label: LDP label bound by C for FEC Y
- Outgoing nhop: C

And the traffic XY travels again over the LDP LSP.

The operator has eliminated its first problem: dynamically established directed LDP sessions are no longer required and the steady-state traffic is still transported over LDP. The SR deployment is confined to the area where these benefits were required.

#### 4.2. Guaranteed FRR coverage

B's MPLS entry to Z becomes:

- Incoming label: local LDB label bound by B for FEC Z
  - Outgoing label: LDP label bound by E for FEC Z
  - o Backup outgoing label: SR node segment for Z {203}
  - Outgoing nhop: E



- o Backup nhop: repair tunnel to G: {106, 9001}
  - G is reachable from B via the combination of a node segment to F {106} and an adjacency segment FG {9001}
  - Note that {106, 107} would have equally work. Indeed, in many case, P's shortest path to Q is over the link PQ. The adjacency segment from P to Q is required only in very rare topologies where the shortest-path from P to Q is not via the link PQ.

In steady-state, X sends its Z-destined traffic to B with a top label which is the LDP label bound by B for FEC Z. B swaps that top label for the LDP label bound by E for FEC Z and forwards to E. E pops the LDP label and forwards to Z.

Upon failure of the link BE, B swaps the incoming top-label with the node segment for Z (203) and sends the packet onto a repair tunnel to G (node segment 106 followed by adjacency segment 9001). Thus, B sends the packet to C with the label stack {106, 9001, 203}. C pops the node segment 106 and forwards to F. F pops the adjacency segment 9001 and forwards to G. G swaps 203 for 203 and forwards to E. E's nhop to Z is not SR capable and hence E swaps the incoming node segment 203 for the LDP label announced by its next-hop (in this case, implicit null).

After IGP convergence, B's MPLS entry to Z will become:

- Incoming label: local LDB label bound by B for FEC Z
  - o Outgoing label: LDP label bound by C for FEC Z
  - o Outgoing nhop: C

And the traffic XZ travels again over the LDP LSP.

The operator has eliminated its second problem: guaranteed FRR coverage is provided. The steady-state traffic is still transported over LDP. The SR deployment is confined to the area where these benefits are required.

## 5. 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  
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

Stefano Previdi  
Cisco Systems, Inc.  
Rome  
IT

Email: sprevidi@cisco.com

Bruno Decraene  
Orange  
Issy-les-Moulineaux  
FR

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



Network Working Group  
Internet Draft  
Intended status: Standards Track  
Expires: November 2014

Hannes Gredler  
Juniper Networks

Yakov Rekhter  
Juniper Networks

Luay Jalil  
Verizon

Sriganesh Kini  
Ericsson

Xiaohu Xu  
Huawei

May 21 2014

Supporting Source/Explicitly Routed Tunnels via Stacked LSPs

draft-gredler-spring-mpls-06.txt

Status of this Memo

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

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

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

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>.

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

## 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.

## Abstract

This document describes how source/explicitly routed tunnels could be realized using stacked Label Switched Paths (LSPs).

This document also describes how use of IS-IS/OSPF as a label distribution protocol fits into the MPLS architecture.

## Table of Contents

1	Specification of Requirements .....	3
2	Terminology .....	3
3	Introduction .....	4
4	Constructing Explicitly Routed Tunnels by using Stacked LSPs	5
4.1	Examples of Constructing Explicitly Routed Tunnels by Stacked LSPs	8
4.1.1	Explicitly Routed Tunnel with Single Hops .....	8
4.1.2	Explicitly Routed Tunnel with Multi-Hops .....	11
5	IS-IS or OSPF as Label Distribution Protocol .....	13
5.1	Example of IS-IS/OSPF as Label Distribution Protocols	15
6	IANA Considerations .....	16
7	Security Considerations .....	16
8	Acknowledgements .....	16
9	Normative References .....	16
10	Informative References .....	16
11	Authors' Addresses .....	17

## 1. Specification of Requirements

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

We use the term "explicitly routed tunnels" as a synonym for such terms as "source routed tunnels" and "source-initiated routed tunnels".

Note that the term "source routed tunnel", or "source-initiated routed tunnel" does not imply that intermediate nodes of such a tunnel forward packets traversing the tunnel based upon source addresses of these packets. In the context of "source routed tunnels" and "source-initiated routed tunnels" the term "source" refers to the

tunnels' ingress.

This document assumes that a reader is familiar with the MPLS architecture [RFC3031] terminology.

For a given Label Switched Path (LSP) of level m, as defined in section 3.15 of [RFC3031]:

- + the ingress node/router is the LSR that pushes the level m label,
- + intermediate nodes/routers are the LSRs making their forwarding decision on a level m label

### 3. Introduction

MPLS architecture [RFC3031] defines the concept of explicitly routed tunnel as follows:

If a Tunneled Packet travels from Ru to Rd over a path other than the Hop-by-hop path, we say that it is in an "Explicitly Routed Tunnel"

where Ru and Rd are Label Switch Routers (LSRs).

To realize explicitly routed tunnels [RFC3031] proposes to use explicitly routed Label Switched Paths (LSPs):

An "Explicitly Routed LSP Tunnel" is a LSP Tunnel that is also an Explicitly Routed LSP

Up until now there have been two possible protocols to instantiate/signal such explicitly routed LSPs - RSVP-TE ([RFC3209]) and CR-LDP ([RFC3212]).

MPLS architecture ([RFC3031]) defines the notion of LSP hierarchy, as LSP tunnels within LSPs. Use of MPLS label stack mechanism allows LSP hierarchy to nest to any depth.

In this document we specify the procedures to realize explicitly routed point-to-point tunnels by using LSP hierarchy, thus defining yet another possible mechanism to realize such tunnels. (Note though that the idea of using LSP hierarchy to realize explicitly routed tunnels is not new - e.g., Remote LFA [R-LFA] uses explicitly routed tunnels constructed by LSP hierarchy.)



An essential part of MPLS is the notion of label distribution protocol. On the subject of whether it should be one, or more than one label distribution protocol, MPLS architecture ([RFC3031]) said the following:

THE ARCHITECTURE DOES NOT ASSUME THAT THERE IS ONLY A SINGLE LABEL DISTRIBUTION PROTOCOL. In fact, a number of different label distribution protocols are being standardized.

Up until now IETF standardized the following label distribution protocols for unicast: LDP ([RFC5036]), CR-LDP ([RFC3212]), RSVP-TE [RFC3209] and BGP ([RFC3107], [RFC4364], [RFC4761]).

Recently there have been proposals ([gredler-isis], [gredler-ospf], [previdi-isis], [psenak-ospf]) to extend IS-IS [RFC1142] and OSPF [RFC1583] to make them yet another label distribution protocols.

This document describes how use of IS-IS or OSPF as label distribution protocols fits into the MPLS architecture. This document also describes the benefits of using IS-IS/OSPF as label distribution protocols for the purpose of constructing explicitly routed tunnels with stacked LSPs.

#### 4. Constructing Explicitly Routed Tunnels by using Stacked LSPs

Instead of explicitly routed LSPs, one can use LSP hierarchy (stack of LSPs) to construct explicitly routed point-to-point tunnels as follows.

Consider an explicitly routed point-to-point tunnel with an explicit route  $\langle R(0), R(1), R(2), \dots R(n) \rangle$ , where  $R(0)$  is the ingress of the tunnel and  $R(n)$  is the egress of the tunnel. Denote the LSPs needed to realize such a tunnel via an LSP stack as  $\langle LSP(1), LSP(2), \dots LSP(n) \rangle$ , where  $LSP(1)$  is the topmost and  $LSP(n)$  is the bottommost LSP in the stack. These LSPs are constructed as follows:

- + All the LSPs in the stack are constructed with the same ingress -  $R(0)$ . (See further down on why this is needed.)
- +  $LSP(i)$  is constructed with  $R(i)$  as its egress (e.g.,  $LSP(1)$  is constructed with  $R(1)$  as its egress,  $LSP(2)$  with  $R(2)$  as its egress, etc...  $LSP(n)$  with  $R(n)$  as its egress).
- + For every  $0 < i < n$ , the first intermediate router of  $LSP(i+1)$  is constructed to be the same as the egress router of  $LSP(i)$ .

- + The first intermediate router of LSP(1) is constructed to be one hop away from R(0). If R(1) is one hop away from R(0), then this intermediate router is also the egress of LSP(1) (in which case LSP(1) is a one-hop LSP). If R(1) is more than one hop away from R(0), then this intermediate router is some router other than R(1), and R(1) is still the egress of that LSP.
- + The first intermediate router of any LSP in the stack, could be either single or multi-hop away from the egress of that LSP.
- + All the LSPs in the stack are constructed with the penultimate hop popping. That is, for each LSP in the stack the penultimate router of that LSP pops the label corresponding to that LSP off the label stack before sending data to the egress router of that LSP.

When R(i) and R(i+1) are single hop away from each other, the first intermediate router of LSP(i+1) is one hop away from the egress of that LSP. When R(i) and R(i+1) are multi-hop away from each other, the first intermediate router of LSP(i+1) is multi-hop away from the egress of that LSP.

Following the above procedures, the LSPs in the stack satisfy the following properties:

- + All LSPs in the stack have the same ingress.
- + The egress of a given LSP in the stack is the first intermediate router of the next LSP in the stack.
- + The first intermediate router of the LSP at the top of the stack is one hop away from the ingress.
- + The first intermediate router of any LSP in the stack could be either single or multi-hop away from the egress of that LSP. (Thus the egress of a given LSP in the stack could be either single or multi-hop away from the egress of the next LSP in the stack.)

Such stack of LSPs provides the functionality to forward a packet through a sequence of egresses of the LSPs on the stack - the sequence of these egresses represents the explicit route of the explicitly routed point-to-point tunnel constructed by using these stacked LSPs. The ingress of all these LSPs is the ingress of the tunnel.

When the first intermediate router of a given LSP in the stack is multi-hop away from the egress of that LSP, the existing label distribution protocols (LDP, RSVP-TE, etc. ) can be used to establish a multi-hop LSP fragment for this LSP. When IS-IS or OSPF, in addition to being a routing protocol, is also used as a label distribution protocol (see section "IS-IS or OSPF as Label Distribution Protocol"), it can also be used to establish such multi-hop LSP fragment.

To construct the label stack associated with the stack of LSPs the ingress of all these LSPs, R0, uses the following procedures:

- + For  $(n > i > 0)$  R(0) obtains from R(i) label binding for LSP(i+1) and places the label onto the stack, starting from the bottommost label (the label that corresponds to LSP(n)).

In section "IS-IS or OSPF as Label Distribution Protocol" we describe how IS-IS or OSPF with appropriate extensions could be used as a label distribution protocol to obtain such label bindings.

If the first intermediate router of LSP(i+1) is either (a) a single hop away from the egress of that LSP, or (b) multi-hop away, and LDP is used as a label distribution protocol to establish a multi-hop LSP fragment between the first intermediate router and the egress of that LSP, then R(0) can use targeted LDP session with R(i) to obtain such label bindings.

- + For LSP(1) if R(1) is one hop away from R(0), then no label is needed (as LSP(1), just like all other LSPs in the stack, is constructed with the penultimate hop popping), and the label stack construction terminates with the topmost label that R(0) obtains from R(1) for LSP(2).
- + Otherwise, if R(1) is more than one hop away from R(0), then R(0) obtains label binding for LSP(1) from the first intermediate router of LSP(1), and places this label at the top of the stack.

Note that the above procedures require all the LSPs in the stack to have the same ingress - R(0). This requirement comes from the observation that (a) R(0) is the router that constructs the whole label stack needed to realize the explicitly routed tunnel, and (b) according to MPLS architecture [RFC3031] when a router wants to create a label stack, the router has to be the head-end of all the LSPs corresponding to the labels in the stack.

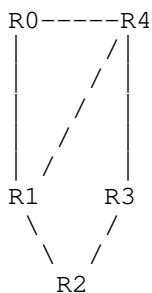
Since the MPLS label stack mechanism allows stack of LSPs to nest to any depth, use of LSP hierarchy for explicitly routed tunnels does not place any protocol restrictions on the number of entries in the explicit route of an explicitly routed tunnel. Note though that there may be some other restrictions (e.g., due to MTU, or hardware) that would place an upper bound on the depth of the label stack, and thus on the number of entries in the explicit route. Also, the depth of the label stack may have implications on ECMP, and specifically on the use of the Entropy label (see [kini] for more).

#### 4.1. Examples of Constructing Explicitly Routed Tunnels by Stacked LSPs

In this section we illustrate how to construct an explicitly routed tunnel by using stacked LSPs. The first example illustrates this for an explicitly routed tunnel where consecutive hops that define the tunnel are one hop away from each other. The second example illustrates this when these hops are more than one hop away from each other.

##### 4.1.1. Explicitly Routed Tunnel with Single Hops

Consider a network topology shown below:



Assume that R0 wants to construct an explicitly routed tunnel with (R0, R1, R2, R3, R4). The consecutive hops that define the tunnel are one hop away from each other. That is, R0 is one hop away from R1, R2 is one hop away from R1, R3 is one hop away from R2, and R4 is one hop away from R3.

R0 constructs this tunnel using the following stack of LSPs:

LSP1: (R0, R1) - top of the stack  
LSP2: (R0, R1, R2)

LSP3: (R0, R2, R3)

LSP4: (R0, R3, R4) - bottom of the stack

Note that this stack of LSPs meets the requirements specified in section "Constructing Explicitly Routed Tunnels by using Stacked LSPs". Specifically,

- + All four LSPs in the stack have the same ingress - R0, which is also the ingress of the explicitly routed tunnel.
- + The egress of LSP1, R1, is the first intermediate router of the next LSP in the stack, LSP2. The egress of LSP2, R2, is the first intermediate router of the next LSP in the stack, LSP3. Likewise, the egress of LSP3, R3, is the first intermediate router of the next (and the last) LSP in the stack, LSP4.
- + The LSP at the top of the stack, LSP1, has its first intermediate router, R1, one hop away from its ingress, R0. Because of that, this intermediate router is also the egress of that LSP, and that LSP is a one-hop LSP.
- + In that particular example the first intermediate router of every LSP in the stack is one hop away from the egress of that LSP. That is, the first intermediate router of LSP2, R1, is one hop away from the egress of that LSP, R2; the first intermediate router of LSP3, R2, is one hop away from the egress of that LSP, R3; and the first intermediate router of LSP4, R3, is one hop away from the egress of that LSP, R4. As a result, the egress of a given LSP in the stack is one hop away from the egress of the next LSP in the stack.

The first intermediate router of each of these LSPs creates label bindings for these LSPs as follows. R3 creates label binding for LSP4 by binding a particular label, L1, to the address of R4, creating a Next Hop Label Forwarding Entry (NHLFE) whose next hop is the link from R3 to R4, and setting the Incoming Label Map (ILM) so that L1 maps to that NHLFE. Likewise, R2 creates label binding for LSP3 by binding a particular label, L2, to the address of R3, creating an NHLFE whose next hop is the link from R2 to R3, and setting the ILM so that L2 maps to that NHLFE. Finally, R1 creates label binding for LSP2 by binding a particular label, L3, to the address of R2, creating an NHLFE whose next hop is the link from R1 to R2, and setting the ILM so that L3 maps to that NHLFE.

To get from the first hop of LSP4, R0, to the second hop of LSP4, R3, the packet has to go through the LSP tunnel provided by LSP3. To get from the first hop of LSP3, R0, to the second hop of LSP3, R2, a

packet has to go through the LSP tunnel provided by LSP2. To get from the first hop of LSP2, R0, to the second hop of LSP2, R1, a packet has to go through the LSP tunnel provided by LSP1.

In order to accomplish this R0 constructs the label stack for the explicitly routed tunnel as follows:

- + Step 1: R0 obtains label binding L1 created by R3 for LSP4 (R0, R3, R4), and starts building the label stack by pushing L1 onto the label stack.
- + Step 2: R0 obtains label binding L2 created by R2 for LSP3 (R0, R2, R3), and pushes L2 into the stack. At this point the stack contains (L2, L1).
- + Step 3: R0 obtains label binding L3 created by R1 for LSP2 (R0, R1, R2), and pushes L3 into the stack. At this point the stack contains (L3, L2, L1).
- + Step 4: Since R0 and R1 are one hop away from each other the label stack construction is completed (R0 does not need a label for one-hop LSP1, as all the LSPs use penultimate hop popping).

So far we did not say anything about how R0 obtains from R3 label binding for LSP4, from R2 label binding for LSP3, and from R1 label binding for LSP2.

At least in principle, these label bindings could be obtain by such already defined label distribution protocols as LDP (to be more precise, targeted LDP if the two routers are more than one hop away from each other). E.g., if one uses targeted LDP, then R0 would need to dynamically establish and maintain a targeted LDP session with R3 and another targeted LDP session with R2 (R0 would maintain a "vanilla" LDP session with R1). Using these LDP sessions R0 would obtain from R3 label binding for LSP4, from R2 label binding for LSP3, and from R1 label binding for LSP2. Note that obtaining such labels bindings with targeted LDP may also require defining a new FEC to be used by targeted LDP.

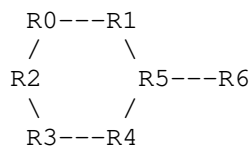
In section "IS-IS or OSPF as Label Distribution Protocol" we describe how IS-IS or OSPF with appropriate extensions could be used as a label distribution protocol to obtain such label bindings.

When R0 wants to forward a packet along the explicitly constructed tunnel (R0, R1, R2, R3, R4), R0 pushes (L3, L2, L1) onto the label stack of the packet, and forwards the packet to R1. R1 performs the lookup on the topmost label, L3, and based on this lookup forwards

the packet to R2. Prior to forwarding the packet to R2, R1 (acting as a penultimate hop for LSP2) pops the topmost label, L3. When R2 receives the packet, R2 performs the lookup on the topmost label, L2, and based on this lookup forwards the packet to R3. Prior to forwarding the packet to R3, R2 (acting as a penultimate hop for LSP3) pops the topmost label, L2. When R3 receives the packet, R3 performs the lookup on the topmost label, L1, and based on this lookup forwards the packet to R4. Prior to forwarding the packet to R4, R3 (acting as a penultimate hop for LSP4) pops the topmost label, L1.

#### 4.1.2. Explicitly Routed Tunnel with Multi-Hops

Consider a network topology shown below:



Assume that R0 wants to construct an explicitly routed tunnel with (R0, R4, R6) as hops. Note that R4 is multi-hop away from R0, and R6 is multi-hop away from R4.

R0 constructs this tunnel using the following stack of LSPs:

LSP1: (R0, R2, R3, R4) - top of the stack  
LSP2: (R0, R4, R5, R6) - bottom of the stack

Note that this stack of LSPs meets the requirements specified in section "Constructing Explicitly Routed Tunnels by using Stacked LSPs". Specifically,

- + Both LSPs in the stack have the same ingress - R0, which is also the ingress of the explicitly routed tunnel.
- + The egress of LSP1, R4, is the first intermediate router of the next LSP in the stack, LSP2.
- + The LSP at the top of the stack, LSP1, has its first intermediate router, R2, one hop away from its ingress, R0. However, this intermediate router is not the egress of that LSP, and therefore this LSP is a multi-hop LSP.

- + In that particular example the first intermediate router of every LSP in the stack is multi-hop away from the egress of that LSP. That is, the first intermediate router of LSP1, R2, is multi-hop away from the egress of that LSP, R4; the first intermediate router of LSP2, R4, is multi-hop away from the egress of that LSP, R6. As a result, the egress of a given LSP in the stack is multi-hop away from the egress of the next LSP in the stack.

In this example we assume that LDP is used as a label distribution protocol for both LSP1 and LSP2. Since R0 and R4 are not IGP neighbors, they are remote label distribution peers. Thus R0 and R4 use targeted LDP for label distribution. All other routers use "vanilla" LDP procedures.

To get from the first hop of LSP2, R0, to its second hop, R4, the packet has to go through the LSP tunnel provided by LSP1.

In order to accomplish this R0 constructs the label stack for the explicitly routed tunnel as follows:

- + Step 1: R0 (using targeted LDP) obtains label binding L1 created by R4 for LSP2 (R0, R4, R5, R6), and starts building the label stack by pushing L1 onto the label stack.
- + Step 2: R0 (using "vanilla" LDP procedures) obtains label binding L2 created by R2 for LSP1 (R0, R2, R3, R4), and pushes L2 into the stack. At this point the stack contains (L2, L1).
- + Step 3: Since R0 and R1 are one hop away from each other the label stack construction is completed (R0 does not need a label for one-hop LSP1).

A reader familiar with Remote LFA FRR [R-LFA] should be able to notice that the example described in this section is nothing more than an instance of Remote LFA FRR, where Remote LFA FRR provides fast reroute to the traffic going from R0 to R6 in the presence of the (R0, R2) link failure, with R0 being the Point of Local Repair (PLR), R4 being the PQ-node, and R6 being the ultimate destination. The explicitly routed tunnel (R0, R4, R6) consists of the PLR as the head-node, the PQ-node as the next hop, and the ultimate destination as yet another hop.



## 5. IS-IS or OSPF as Label Distribution Protocol

When OSPF or IS-IS, in addition to being a routing protocol, is also used as a label distribution protocol (as proposed in [gredler-isis], [gredler-ospf], [previdi-isis], [psenak-ospf]), the OSPF/IS-IS Link State Advertisements originated by a router carry label bindings for LSPs that either transit or originated by the router. Doing this allows to extend such LSPs. The criteria for selecting among all these LSPs a subset for which the router would originate label binding advertisements in IS-IS/OSPF are purely local to the router. The router could be either single or multi-hop away from the egresses of the LSPs in the subset. Existing label distribution protocols (LDP, RSVP-TE, etc.) can be used to establish multi-hop LSP fragments if the router is multi-hop away from the egress of a particular LSP in the subset. When IS-IS or OSPF, in addition to being a routing protocol, is also used as a label distribution protocol, it can also be used to establish such multi-hop LSP fragments.

Use of IS-IS or OSPF as a label distribution protocol supports advertisements of label mappings for such FECs as:

- + IPv4/IPv6 prefix FEC via a hop-by-hop LSP established using IS-IS/OSPF as a label distribution protocol.
- + IPv4/IPv6 prefix FEC via a hop-by-hop LSP established using LDP as a label distribution protocol.
- + IPv4/IPv6 address FEC where the address identifies the remote end of one of the advertising router's links via an LSP that traverses the link and terminates on the remote end of the link.
- + IPv4/IPv6 address FEC where the address identifies the remote end of one of the advertising router's point-to-point links via an LSP that traverses the link and terminates on the remote end of the link.
- + IPv4/IPv6 address FEC where the address identifies a (remote) router connected to the advertising router by a broadcast link via an LSP that traverses the link and terminates on the remote router identified by its node-id.
- + IPv4/IPv6 prefix FEC via an explicitly routed LSP established using RSVP-TE, where path computation for such LSP is done by either distributed CSPF, or by PCE.

When a router obtains label binding for a given FEC from more than one label distribution protocol (e.g., one binding from targeted LDP

and another from IS-IS/OSPF), deciding which label binding to use is a matter of policy local to the router. In the scenario where a router obtains label binding for a given FEC from both (targeted) LDP and IS-IS/OSPF, the default behavior RECOMMENDED in this document is to prefer the one obtained from (targeted) LDP. An implementation MUST support the ability to override the default behavior via configuration.

MPLS architecture [RFC3031] defines the notion of local/remote label distribution peers as follows:

When two LSRs are IGP neighbors, we will refer to them as "local label distribution peers". When two LSRs may be label distribution peers, but are not IGP neighbors, we will refer to them as "remote label distribution peers."

Following OSPF/IS-IS procedures each router passes Link State Advertisements originated by other routers unmodified. When these advertisements carry label binding information, this information is also passed unmodified. Therefore, the router that originates label bindings advertisements in IS-IS/OSPF can be either single or multi-hop away from the routers that receive and use these bindings. In the former case the IGP neighbors of the router that originates the advertisements will be the local label distribution peers of the router. In the latter case other routers in the same IGP domain will be the remote label distribution peers of the router.

Use of OSPF or IS-IS as a label distribution protocol provides scalable support for remote label distribution peering in terms of the number of label distribution peers a given router has to maintain. This is because label distribution protocol messages (Link State Advertisements) are exchanged only between IGP neighbors, without requiring control plane peering between a router that originates Link State Advertisements and each of its remote label distribution peers.

It is important to note that the existing MPLS control plane already has mechanisms/protocols to support remote label distribution peering (using BGP or targeted LDP [RFC5036]). Thus the practical relevance of the ability to provide scalable support for remote label distribution peering with IS-IS or OSPF as a label distribution protocol depends on a particular use case.

If for a given subset of routers within an MPLS network each router within the subset is assigned a distinct index, then one could compress announcements of labels bound to the LSPs whose FECs are the IP addresses of these routers by (a) advertising these indices in IS-IS/OSPF, and (b) making each router advertise a label block in IS-

IS/OSPF as well. A router R1 that advertises a given label block algorithmically binds a FEC associated with an IP address of some other router R2 to the label from that block that is identified by the index that R2 advertises in IGP. A router R1 that receives label block originated by some other router R2 can determine the label bound to a FEC associated with an IP address of some other router R3 by using the index advertised by R3 as an offset into the label block advertised by R2. Note that to avoid wasting labels this scheme requires a fairly dense assignment of indices. Also note that to expand the number of labels that a router advertises using label blocks, the router may advertise more than one label block.

Note though, that the benefits of scaling improvements in terms of label distribution peering come at a cost, as every router in the domain ends up keeping all the labels assigned/bounded by every other router in the domain, whether it really needs to know them or not. Whether this cost is of practical significance depends on (a) the number of label bindings being advertised, and (b) the encoding of label bindings (e.g., use of label blocks vs enumerating each label binding).

#### 5.1. Example of IS-IS/OSPF as Label Distribution Protocols

In this section we illustrate how IS-IS/OSPF with extensions, as defined in [gredler-isis], [gredler-ospf], [previ-di-isis], [psenak-ospf] could be used as a label distribution protocol to support explicitly routed tunnels realized by stacked LSPs. For the purpose of this illustration we assume the scenario described in section "Example of Constructing Explicitly Routed Tunnels by Stacked LSPs". In that example one of the key issues is the ability of R0 to obtain from R3 label binding for LSP4, from R2 label binding for LSP3, and from R1 label binding for LSP2.

To obtains such label bindings, the Link State Advertisement originated by R3 carries label L1 (this is the label that R3 binds to LSP4). Using IS-IS/OSPF procedures this Link State Advertisement is propagated by R2 and R1 (as well as by R4) to R0. This is how R0 obtains from R3 label binding for LSP4. In a similar fashion, the Link State Advertisement originated by R2 carries label L2 (which is the label that R2 binds to LSP3). Using IS-IS/OSPF procedures this Link State Advertisement is propagated by R1 (as well as by R3 and R4) to R0. This is how R0 obtains from R2 label binding for LSP3. Likewise, the Link State Advertisement originated by R1 carries label L3 (which is the label that R1 binds to LSP2). Using IS-IS/OSPF procedures this Link State Advertisement is delivered to R0. This is how R0 obtains from R1 label binding for LSP2.

Note that while from R0's perspective both R2 and R3 are remote label distribution peers, R0 does not maintain any control plane peering (e.g., targeted LDP) with either R2 or R3.

## 6. IANA Considerations

This document introduces no new IANA Considerations.

## 7. Security Considerations

TBD

## 8. Acknowledgements

We would like to thank John Drake (Juniper Networks) and John Scudder (Juniper Networks) for their review and comments.

We would also like to thank Bruno Decraene (Orange) and Robert Raszuk for their review and comments.

## 9. Normative References

[RFC3031] Rosen, E., et. al., "Multiprotocol Label Switching Architecture", RFC3031, January 2001

## 10. Informative References

[kini] Kini, S., et. al., "Entropy labels for source routed stacked tunnels", draft-kini-mpls-entropy-label-src-stacked-tunnels, work in progress

[gredler] Gredler, H., et. al., "Advertising MPLS labels in IGPs" draft-gredler-rtgwg-igp-label-advertisement, work in progress

[gredler-isis] Gredler, H., et. al., "Advertising MPLS labels in IS-IS" draft-gredler-isis-label-advertisement, work in progress

[gredler-ospf] Gredler, H., et. al., "Advertising MPLS labels in OSPF" draft-gredler-ospf-label-advertisement, work in progress

[previdi-isis] Previdi, S., et. al., "IS-IS Extensions for Segment Routing" draft-previdi-isis-segment-routing-extensions, work in progress

[psenak-ospf] Psenak, P., et. al., "OSPF Extensions for Segment Routing" draft-psenak-ospf-segment-routing-extensions, work in progress

[R-LFA] Bryant, S., et. al., "Remote LFA FRR", draft-ietf-rtgwg-remote-lfa, work in progress

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

[RFC3107] Rekhter, Y., et. al., "Carrying Label Information in BGP-4", RFC3107, May 2001

[RFC3209] Awduche, D., et. al., "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC3209, December 2001

[RFC3212] Jamoussi, B., et. al., "Constraint-Based LSP Setup using LDP", RFC3212, January 2002

[RFC4364] Rosen E., et. al., "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC4364, February 2006

[RFC4761] Kompella, K., et. al., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC4761, January 2007

[RFC5036] L. Andersson, et. al., "LDP Specification", RFC5036, October 2007

## 11. Authors' Addresses

Hannes Gredler  
Juniper Networks  
e-mail: hannes@juniper.net

Yakov Rekhter  
Juniper Networks  
e-mail: yakov@juniper.net

Luay Jalil  
Verizon  
e-mail: luay.jalil@verizon.com

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

Xiaohu Xu

Huawei Technologies,  
Email: xuxiaohu@huawei.com



Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: March 02, 2014

S. Kini, Ed.  
Ericsson  
K. Kompella  
Juniper  
S. Sivabalan  
Cisco  
August 29, 2013

Entropy labels for source routed stacked tunnels  
draft-kini-mpls-entropy-label-src-stacked-tunnels-01

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 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 March 02, 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 . . . . .	2
1.1. Requirements Language . . . . .	2
2. Abbreviations and Terminology . . . . .	2
3. Entropy Labels for source routed stacked tunnels . . . . .	3
3.1. Single EL at the bottom of the stack of tunnels . . . . .	4
3.2. An EL per tunnel in the stack . . . . .	4
3.3. A re-usable EL for a stack of tunnels . . . . .	4
3.4. ELs at readable label stack depths . . . . .	5
4. Acknowledgements . . . . .	5
5. IANA Considerations . . . . .	5
6. Security Considerations . . . . .	5
7. References . . . . .	5
7.1. Normative References . . . . .	5
7.2. Informative References . . . . .	6
Authors' Addresses . . . . .	6

## 1. Introduction

The source routed stacked tunnels paradigm is leveraged by techniques such as Segment Routing (SR) [I-D.filsfils-rtgwg-segment-routing] to steer a packet through a set of segments. This can be directly applied to the MPLS data plane. Entropy labels (EL) [RFC6790] is a technique used by the MPLS data plane to do load balancing. Applying ELs to stacked tunnels brings up some issues and these are documented in Section 3.

### 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. Abbreviations and Terminology

EL - Entropy Label

ELI - Entropy Label Identifier

SR - Segment Routing

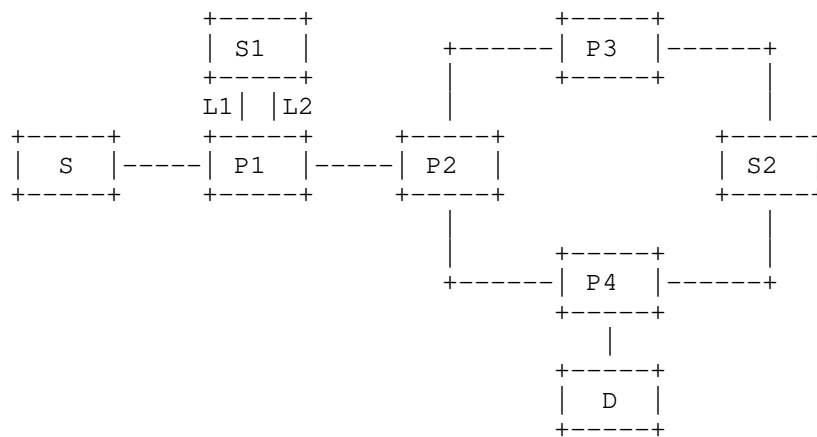
ECMP - Equal Cost Multi Paths

MPLS - Multi Protocol Label Switching

SID - Segment Identifier

### 3. Entropy Labels for source routed stacked tunnels

Stacked tunnels have several use-cases, one of which is service chaining [I-D.filsfils-rtgwg-segment-routing-use-cases]. Consider a service-chaining network in Figure 1. The source LSR S wants to send traffic to destination LSR D. This traffic is required to go through service nodes S1 and S2 to produce the service chain S-S1-S2-D. Segment Routing can be used to achieve this. Load balancing is required across the parallel links between P1 and S1. Load balancing is also required between the ECMP paths from S1 to S2, S1-P1-P2-P3-S2 and S1-P1-P2-P4-S2. The source LSR wants the intermediate LSRs P1 and P2 to take local load balancing decisions and does not specify the Segment Identifiers (SIDs) of specific interfaces. Entropy labels should be used to achieve the desired load balancing. Two possible ways to use the entropy labels and their associated tradeoffs are discussed below. We denote SN to be the node segment identifier (SID) of LSR N and SN{L1,L2,...} to denote the SID of the adjacency set for links {L1,L2,...} of LSR N and S-N to denote the SID for a service at service node N. The label stack that the source LSR S uses for the service chain can be <SS1, S-S1, SS2, S-S2, SD> or <SP1, SP1{L1,L2}, S-S1, SS2, S-S2, SD>. The issues discussed in this document are equally applicable to both of these options.



S=Source LSR, D=Destination LSR, S1,S2=service-nodes, L1,L2=links,  
P1,P2,P3,P4=Transit LSRs

Figure 1: Service chaining use-case

### 3.1. 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 Figure 1 label stack at LSR S would look like `<SP1, SP1{L1,L2}, SS1, S-S1, SS2, S-S2, SD, 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 correspondingly goes deeper in the label stack. As a result, intermediate LSRs (such as P1) that have to walk the label stack at least until the EL to perform load balancing decisions have to access a larger number of bytes in the packet header when making forwarding decisions. A network design using this approach, should ensure that all intermediate LSRs have the capability to traverse the maximum label stack depth in order to do effective load balancing. The use-case for which the tunnel stacking is applied would determine the maximum label stack depth.

### 3.2. 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. For the same Figure 1 above, the source LSR S encoded label stack would be `<SP1, SP1{L1,L2}, ELI, EL1, SS1, S-S1, SS2, ELI, EL2, SD>` where all the ELs would typically have the same value. Accessing the EL at an intermediate LSR is independent of the depth of the label stack and hence independent of the specific use-case to which the stacked tunnels are applied. 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.

### 3.3. 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. For the same Figure 1 above, the source LSR S encoded label stack would be `<SP1, ELI, EL, SP1{L1,L2}, SS1, S-S1, SS2, SD>`. At P1 the

outgoing label stack would be <SS1, ELI, EL, S-S1, SS2, SD> after it has load balanced to one of the links L1 or L2. At S1 the outgoing label stack would be <SS2, ELI, EL, SD>. At P2 the outgoing label stack would be <SS2, ELI, EL, SD> and it would load balance to one of the nexthop LSRs P3 or P4. Accessing the EL at an intermediate LSR is independent of the depth of the label stack and hence independent of the specific use-case to which the stacked tunnels are applied.

### 3.4. 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. Details of this will follow in subsequent versions if this option is found to be worth pursuing. For the same Figure 1 above, if LSR P1 needs to have the EL within a depth of 4, then the source LSR S encoded label stack would be <SP1, SP1{L1,L2}, ELI, EL1, SS1, S-S1, SS2, ELI, EL2, SD> where all the ELs would typically have the same value.

## 4. Acknowledgements

The authors would like to thank Rob Shakir and TBD for their comments.

## 5. IANA Considerations

This memo includes no request to IANA.

## 6. Security Considerations

## 7. References

### 7.1. Normative 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.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-00 (work in progress), June 2013.

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

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012.

## 7.2. Informative References

[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.

## 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

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

Z. Li  
Q. Zhao  
X. Chen  
Huawei Technologies  
T. Yang  
China Mobile  
R. Raszuk  
Individual  
July 3, 2014

A Framework of MPLS Global Label  
draft-li-mpls-global-label-framework-02

Abstract

The document defines the framework of MPLS global label including the label allocation method for MPLS global label, the representation of MPLS global label and the process of control plane and data plane for MPLS global label.

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 January 4, 2015.

## 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. Terminology . . . . .	3
3. MPLS Global Label Allocation Methods . . . . .	3
3.1. Special-Purpose MPLS Label . . . . .	3
3.2. Domain Wide Labels . . . . .	3
3.2.1. Label Allocation Methods . . . . .	4
4. Representation of MPLS Global Label . . . . .	4
4.1. Per-platform Label Space . . . . .	4
4.2. Context-Specific Label Space . . . . .	5
5. Control Plane for MPLS Global Label . . . . .	5
5.1. Architecture . . . . .	5
5.2. In-Band Global Label Allocation . . . . .	7
5.2.1. Label Allocation in Per-Platform MPLS Label Space . .	7
5.2.2. Label Allocation in Context-Specific Label Space . .	9
5.3. Label Mapping Distribution . . . . .	9
5.4. Inter-Domain Label Negotiation . . . . .	9
5.5. Protocol Extensions Requirement . . . . .	10
5.5.1. IGP Protocol Extensions . . . . .	10
5.5.2. BGP Protocol Extensions . . . . .	10
5.5.3. PECP Protocol Extensions . . . . .	10
6. Data Plane of MPLS Global Label . . . . .	11
6.1. Global Label in Per-Platform Label Space . . . . .	11
6.2. Global Label in Context-Specific Label Space . . . . .	11
6.3. Global Process of Inner Global Label . . . . .	11
7. IANA Considerations . . . . .	12
8. Security Considerations . . . . .	12
9. References . . . . .	12
9.1. Normative References . . . . .	12
9.2. Informative References . . . . .	13
Authors' Addresses . . . . .	13

## 1. Introduction

[I-D.li-mpls-global-label-usecases] proposes possible usecases of MPLS global label. MPLS global label can be used for identification of the location, the service and the network in different application scenarios.

Several MPLS global label allocation mechanisms has been proposed in [RFC5331], [I-D.raszuk-mpls-domain-wide-labels], etc.. This document is to define the framework for MPLS global label based on the existing work and more emerging applications. The framework includes the label allocation method for MPLS global label, the representation of MPLS global label and the process of control plane and data plane for MPLS global label.

## 2. Terminology

FEC: Forward Equivalence Class

MVPN: Multicast VPN

PCE: Path Computation Element

SRGB: Global Segment Routing Block

## 3. MPLS Global Label Allocation Methods

MPLS global label is the label which meaning can be understood by all nodes or part of nodes in the network. These nodes can be nodes in one domain or nodes spanning multiple domains.

### 3.1. Special-Purpose MPLS Label

Special-purpose MPLS label defined in [RFC7274] is a type of special global label. These labels have specific well-known meaning which can be understood and processed accordingly by all MPLS nodes in the network. These labels are allocated and retired by IANA. How to allocate and retire these labels is specified in [RFC7274].

### 3.2. Domain Wide Labels

Besides the special-purpose labels which have the global meaning and are defined by the IANA, it is necessary to provide dynamic allocation mechanisms to allocate global labels to satisfy requirements of emerging possible applications ([I-D.li-mpls-global-label-usecases]). Such global labels may be not possible to be understood by all network nodes like the special-purpose label. That is, these labels may be only understood by all



nodes or part of nodes in one domain or multiple domains. This type of global label can also be called as Domain Wide Label. The scope of domains for Domain Wide Label is service-specific or management-specific which is out of scope of this document.

Note: In the following sections of this document, the global label always means Domain Wide Label. That is, the global label and the Domain Wide Label have the same meaning.

### 3.2.1. Label Allocation Methods

There are two types of label allocation methods for Domain Wide Labels: out-of-band label allocation and in-band label allocation.

Out-of-band label allocation means that the global labels are planned and designated manually for special usage. The typical scenario is Segment Routing. When MPLS is applied for Segment Routing, the global labels allocated for node segments is based on the reserved SRGB and the designated unique Segment ID. In essence the global uniqueness of these label is guaranteed by manual planning. So this method can be seen as the out-of-band label allocation.

In-band label allocation means that the global labels are requested and allocated dynamically through control protocols in the domains. The typical example is the upstream MPLS label assignment defined in [RFC5331]. The method has been adopted in BGP-based MVPN ([RFC6514]) in which the root PE allocates labels to represent MVPN instances and advertise the label binding to leaf PEs for the scenario that multiple MVPNs shares one P-multicast tree.

Choice of the two methods is related with scalability of the possible applications. If the scale of the application is limited, the out-of-band method is enough. Otherwise, the in-band method must be taken into account.

## 4. Representation of MPLS Global Label

### 4.1. Per-platform Label Space

The labels in the per-platform label space can be used for Domain Wide label. The advantage of this method is that the existing MPLS forward plane which is used for widely deployed applications based on the per-platform label can be reused well to support global labels. The challenge for the method is that the existing MPLS protocols such as LDP, BGP and RSVP-TE are always allocate local labels from the space which may cause the confliction with the global label allocation. This confliction could be prevented through division of

the per-platform space into multiple segments used for local label and global label respectively.

#### 4.2. Context-Specific Label Space

The concept of Context-Specific Label Space is defined in [RFC5331]. The labels in Context-Specific label space can also be used for Domain Wide label. The Context-Specific label space is isolated from the per-platform label space and the confliction issue of label allocation can be avoided naturally. The challenge for the method is the possible complexity of both control plane and forward plane introduced by multiple label spaces management.

If the Context-Specific label space is used for global labels, it is necessary to determine the Context Identifier for the label space. There are two methods as follows:

-- Service-specific Context Identifier: The Context Identifier is determined by the service. For example, in [RFC6514], the Tunnel-Specific label space is introduced in which the P-Tunnel Identifier becomes the Context Identifier for the label space.

-- MPLS Global Label Indicator: It is to define a well-known Context-Specific label space for global label. The label space is indicated by the MPLS Global Label Indicator which can be seen as a well-known Context Identifier. In the forwarding plane, the MPLS Global Label Indicator is a special-purpose label to indicate that next label in the MPLS label stack of each transported packet is Domain Wide Labels. The value of the special-purpose label needs to be allocated by IANA according to [RFC7274].

### 5. Control Plane for MPLS Global Label

#### 5.1. Architecture

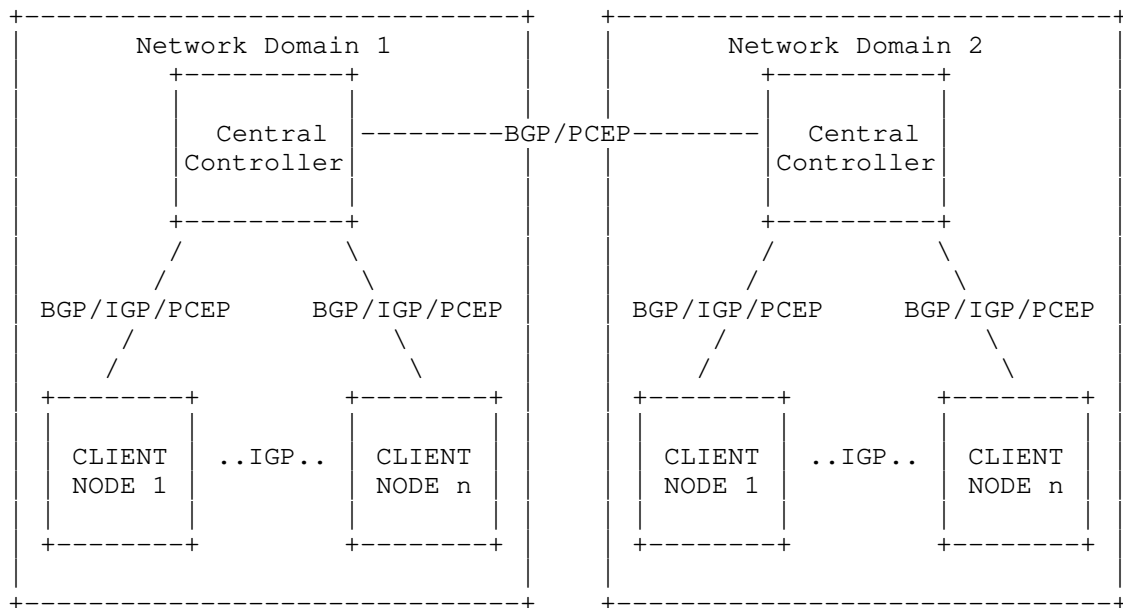


Figure 1: Architecture of In-band Domain-Wide Label Allocation

MPLS global label should be allocated centrally to guarantee all nodes can understand the same meaning for a specific global label. It is natural to adopt the central control architecture for the in-band label allocation. In the architecture the central controller is responsible for allocating the global labels and advertising to the client nodes in the network. When client nodes receives the label binding, it will install the corresponding forwarding entry for the global label.

The applications based on global labels are different: they may need advertise global label to all nodes of a domain, edge nodes of a domain or part of nodes of a domain. IGP extensions, BGP extensions and PCEP extensions are appropriate for these applications respectively. In addition, the global label may be negotiated across multiple domains, it will adopt BGP extensions and PCEP extensions.

Central Control of global labels is the logical functionality which can be deployed in the independent server or in the network device. For example, the upstream label assignment for BGP-based MVPN is done by the root node of MVPN which can be seen as the central controller for the global label.

## 5.2. In-Band Global Label Allocation

### 5.2.1. Label Allocation in Per-Platform MPLS Label Space

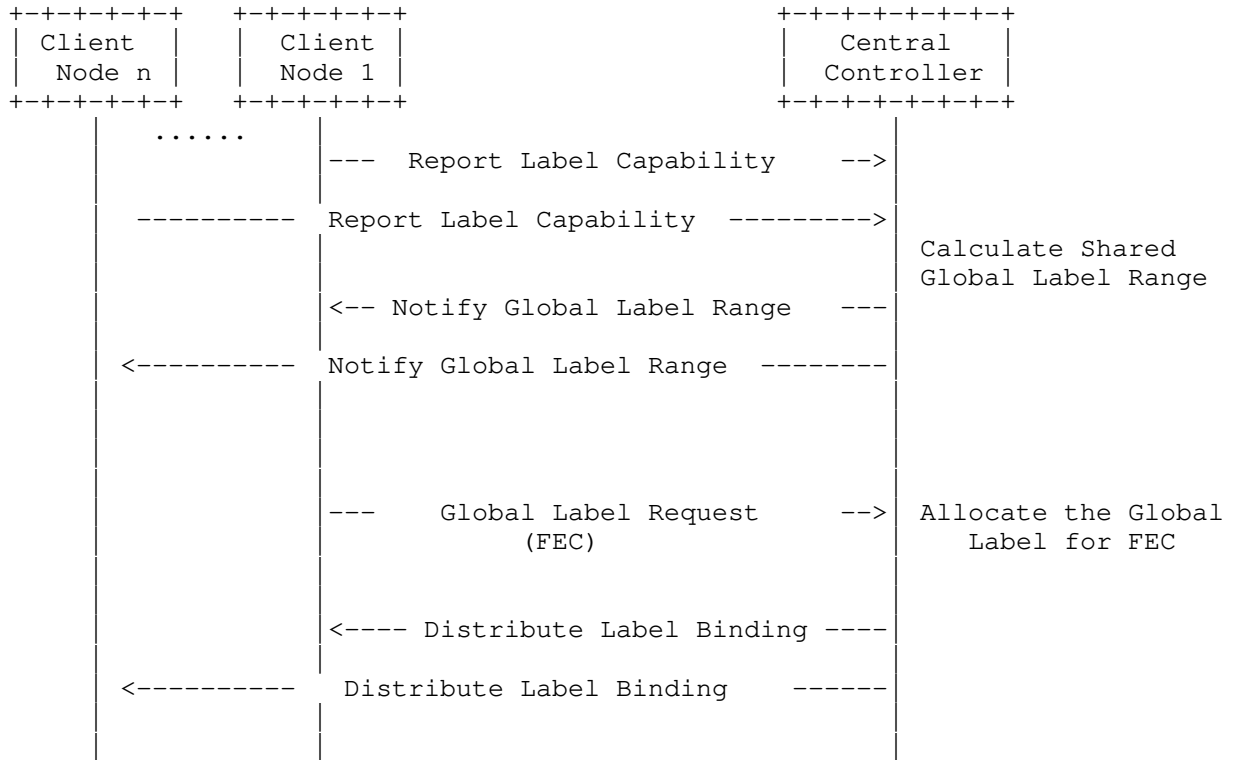


Figure 2: Procedures of Global Label Allocation

Procedures of global label allocation from per-platform label space is shown in the Figure 2. There are two import phases for these procedures: Shared MPLS global label range calculation and MPLS global label allocation.

#### 5.2.1.1. Shared MPLS Global Label Range Calculation

1. Clients nodes should report MPLS label capability to the central controller.
2. The central controller collects MPLS label capability of all nodes. Then it can calculate the shared MPLS global label range for all nodes.

3. The central controller should notify the shared global label range to all client nodes.

Report of label capability and notification of shared MPLS global range can be done by IGP, BGP or PCEP extensions.

#### 5.2.1.2. Label Allocation

There are two methods for the global label allocation: On-demand label allocation and Unsolicited label allocation.

##### 1. On-demand allocation

This method is that the global label allocation is done by the central controller based on the label requirement from client nodes. The procedures of on-demand allocation are as follows:

- 1) The client node should send the global label request for specific usage to the central controller. FEC (Forward Equivalence Class) should be incorporated in the MPLS global label request message.
- 2) When the central controllers receives the MPLS global label request, it should allocate the label from the shared MPLS global label range of all nodes.
- 3) The central controller distributes the MPLS global label mapping message to all client nodes. Thus the MPLS global label for specific usage can be understood by all client nodes.
- 4) The client nodes receive the MPLS global label mapping message and install the corresponding MPLS forwarding entry for the global label.

Label request and distribution of label mapping which are used in on-demand allocation can be done by BGP extensions or PCEP extensions.

##### 2. Unsolicited allocation

This method is that the central controller directly allocates the global label without receiving the label request. The procedures of unsolicited allocation are as follows:

- 1) Discovery of service: this can be implemented by configuration or auto discovery which is service-specific and out of scope of this document.
- 2) The central controller allocates the global label from the global label space for the service.

3) The central controller distributes the MPLS global label mapping message to all client nodes. Thus the MPLS global label for specific usage can be understood by all related client nodes.

4) The client nodes receive the MPLS global label mapping message and install the corresponding MPLS forwarding entry for the global label.

Distribution of label mapping which is used in unsolicited allocation can be done by IGP extensions, BGP extensions or PCEP extensions.

#### 5.2.2. Label Allocation in Context-Specific Label Space

As mentioned in previous section, there can be two types of Context-Specific label space for global label allocation. For the Context-Specific label space identified by service-specific context identifier, the label allocation procedures are service-specific and these procedures are out of scope of this document. For the Context-Specific label space identified by MPLS Global Label Indicator, since the label space is well-known, it is necessary to calculate the share global label range like the global allocation in the per-platform label space. Except this, other procedures for global label allocation are similar as the global label allocation in per-platform label space.

#### 5.3. Label Mapping Distribution

After allocating the global label by the central controller, the label mapping must be distributed to all involved nodes of the specific global-label-based service. If the central controller connects to all involved nodes, the label mapping can be directly advertised to these nodes. But if the central controller only connects part of the involved nodes, it not only needs to distribute the label mapping to the connected client nodes, but also the label mapping should be distributed to other client nodes by the clients nodes which receive the label mapping from the central controller. The distribution of label mapping among client nodes can be implemented by IGP extensions.

#### 5.4. Inter-Domain Label Negotiation

If the global label for the service needs to be allocated across multiple domains, PCEP extensions or BGP extensions can be introduced for label negotiation across multiple domains.

## 5.5. Protocol Extensions Requirement

### 5.5.1. IGP Protocol Extensions

REQ 01. Report Label Capability from client nodes to the central controller.

REQ 02. Notify the shared global label range from the central controller to client nodes.

REQ 03: Distribute label mapping from the central controller to client node.

REQ 04: Distribute label mapping among client nodes.

### 5.5.2. BGP Protocol Extensions

REQ 11. Report Label Capability from client nodes to the central controller.

REQ 12. Notify the shared global label range from the central controller to client nodes.

REQ 13: Send global label request from client nodes to the central controller.

REQ 14: Distribute label mapping from the central controller to client node.

REQ 15: Inter-domain global label negotiation

### 5.5.3. PECP Protocol Extensions

REQ 21. Report Label Capability from client nodes to the central controller.

REQ 22. Notify the shared global label range from the central controller to client nodes.

REQ 23: Send global label request from client nodes to the central controller.

REQ 24: Distribute label mapping from the central controller to client node.

REQ 25: Inter-domain global label negotiation

## 6. Data Plane of MPLS Global Label

### 6.1. Global Label in Per-Platform Label Space

For global label allocated from the per-platform label space, the existing MPLS forwarding mechanism can be reused without modification.

### 6.2. Global Label in Context-Specific Label Space

For a global label allocated within the Context-Specific label space, it is necessary to maintain multiple MPLS label forwarding table in the forwarding plane. When forwarding packets with global label encapsulation, it must decapsulate the label for the Context Identifier firstly to determine the MPLS label forwarding table of the corresponding Context-Specific label space. Then it will decapsulate the next label and search the corresponding MPLS forwarding entry in the Context-Specific label space. The encapsulation of the global label from the Context-Specific label space is shown as follows:

Global Label Indicator	Global Label
---------------------------	--------------

### 6.3. Global Process of Inner Global Label

Because the label forwarding entry for the global label can be created in multiple nodes in the network, there may be one application scenario for which the global label is located in the middle of the label stack of the transported packet and should be processed by all possible node. For example, the Entropy Label for ECMP can be encapsulated multiple times following multiple node segments in Segment Routing. This method may cause the depth of the label stack of the packet is too deep to process. In order to solve this issue, the global label can be introduced to represent the same process of all possible nodes. Thus the depth of the label stack can be reduced. This method can be implemented by introducing a special-purpose label which is named as Global Process Indicator (GPI). When the Global Process Indicator is encapsulated in the packet, it indicates that the next global label SHOULD be process by each node along the path.

The encapsulation of the global label allocated from the per-platform label space which needs to be globally processed is as follows:



Global Process Indicator	Global Label
-----------------------------	--------------

The encapsulation of the global label allocated from the Context-Specific label space indicated by MPLS Global Label Indicator which needs to be globally processed is as follows:

Global Process Indicator	Global Label Indicator	Global Label
-----------------------------	---------------------------	--------------

## 7. IANA Considerations

Following two special-purpose labels defined in this document needs to be allocated by IANA:

- Global Label Indicator
- Global Process Indicator

## 8. Security Considerations

TBD.

## 9. References

### 9.1. Normative References

- [I-D.li-mpls-global-label-usecases]  
Li, Z., Zhao, Q., and T. Yang, "Useases of MPLS Global Label", draft-li-mpls-global-label-usecases-01 (work in progress), February 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, August 2008.
- [RFC7274] Kompella, K., Andersson, L., and A. Farrel, "Allocating and Retiring Special-Purpose MPLS Labels", RFC 7274, June 2014.

## 9.2. Informative References

- [I-D.raszuk-mpls-domain-wide-labels]  
Raszuk, R., "MPLS Domain Wide Labels", draft-raszuk-mpls-domain-wide-labels-01 (work in progress), January 2014.
- [RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, February 2012.

## Authors' Addresses

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

Email: lizhenbin@huawei.com

Quintin Zhao  
Huawei Technologies  
125 Nagog Technology Park  
Acton, MA 01719  
US

Email: quintin.zhao@huawei.com

Xia Chen  
Huawei Technologies  
Huawei Bld., No.156 Beiqing Rd.  
Beijing 100095  
China

Email: jescia.chenxia@huawei.com

Tianle Yang  
China Mobile  
32, Xuanwumenxi Ave.  
Beijing 01719  
China

Email: yangtianle@chinamobile.com

Robert Raszuk  
Individual

Email: robert@raszuk.net

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 19, 2016

Z. Li  
Q. Zhao  
Huawei Technologies  
T. Yang  
China Mobile  
R. Raszuk  
Individual  
L. Fang  
Microsoft  
October 17, 2015

Usecases of MPLS Global Label  
draft-li-mpls-global-label-usecases-03

Abstract

As the MPLS technologies develop, MPLS label is not only used with the local meaning which is always be understood by the upstream node and the downstream node, but also used with the global meaning which can be understood by all nodes or part of nodes in the network. The document defines the latter as the global label and proposes the possible use cases of global label. In these usecases MPLS global label can be used for location identification, VPN identification, segment routing, etc.

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 19, 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. Terminology . . . . .	3
3. Use Cases . . . . .	3
3.1. Location Identification . . . . .	3
3.2. VPN Identification . . . . .	4
3.2.1. Flow Label of VPN LSP . . . . .	4
3.2.2. Aggregate MVPN/VPLS over Single P-Tunnel . . . . .	5
3.3. Segment Routing . . . . .	5
4. Discussion . . . . .	6
5. IANA Considerations . . . . .	8
6. Security Considerations . . . . .	8
7. References . . . . .	8
7.1. Normative References . . . . .	8
7.2. Informative References . . . . .	8
Authors' Addresses . . . . .	11

## 1. Introduction

In the traditional MPLS architecture, MPLS label is always distributed from the downstream node to the upstream node by LDP, RSVP-TE and MP-BGP. These label mappings always have the local meaning which can only be understood by the upstream node and the downstream node. As the MPLS technologies develop, there proposes possible usecases in which MPLS label mapping can be advertised to all nodes or part of nodes in the network. That is, the meaning of the label mapping will be understood by all nodes or part of nodes in the network other than the local upstream node and downstream node. This document defines such type of MPLS label as global label as the opposite of local label.

In the MPLS world there are another pair of label related concepts: per-platform label space [RFC3031] and context-specific label space [RFC5331]. According to [RFC3031] MPLS local label can be allocated from per-platform label space and per-interface label space (in [RFC5331], per-interface label space is generalized as one type of context-specific label space). MPLS global label can also be allocated from per-platform label space or context-specific label space.

The document proposes the possible usecases of MPLS global label. In these usecases MPLS global label can be used for location identification, VPN identification, segment routing, etc.

## 2. Terminology

CE: Customer Edge

MP2P: Multi-Point to Point

MP2MP: Multi-point to Multi-point

MVPN: Multicast VPN

P2MP: Point to Multi-Point

P2P: Point to Point

PE: Provider Edge

## 3. Use Cases

### 3.1. Location Identification

[I-D.bryant-mpls-flow-ident] and [I-D.bryant-mpls-synonymous-flow-labels] propose the challenge of the measurement of packet loss for the multi-point to point LSP. In this case the same label is normally used by multiple ingress or upstream LSRs for specific prefixes and hence source identification is not possible by inspection of the top label by the egress LSRs. Thus [I-D.bryant-mpls-synonymous-flow-labels] proposes the synonymous flow label to be used to introduce some source specific information encapsulated in the packet to identify packet batches from a specific source.

MPLS LDP LSP is one type of multi-point to point LSP. As the network convergence develops, MPLS LDP network needs to interwork with MPLS TE/MPLS-TP network and unified MPLS OAM becomes the realistic requirement. In this usecase, MPLS global label can be allocated for

each network node and advertised in the network. When implement the measurement of packet loss for LDP LSP, such MPLS global label can be used as the flow label to identify the source node of the LDP LSP. When the destination receives the packets it can differentiate flows from specific source node based on the advertised global label binding information for network nodes. In this usecase, MPLS global label is used as the unique identification of source nodes in the network and may save the complex flow label negotiation process between the source node and the destination node.

### 3.2. VPN Identification

MPLS global label can be allocated for VPN and advertised in the network. In this usecase, MPLS global label is used as the unique identification of VPN in the network and can be used for multiple purposes.

#### 3.2.1. Flow Label of VPN LSP

BGP VPN LSP is another type of multi-point to point LSP which faces the challenge of the measurement of packet loss proposed by [I-D.bryant-mpls-flow-ident] and [I-D.bryant-mpls-synonymous-flow-labels]. In this usecase, the flow label should be introduced to identification of the source VPN. There are two possible ways to use global label as the flow label:

Option 1: The global label is allocated for the same VPN on all PE nodes and advertised in the network. And global labels can be allocated for PE nodes and advertised in the network. Then the flow label should be the source PE label + the VPN label shown in the figure 1.

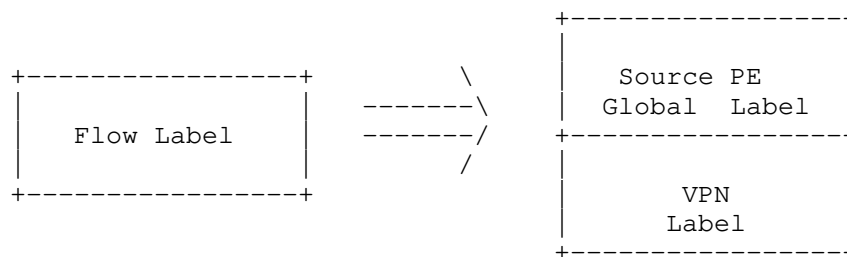


Figure 1: Flow Label using Two Layers of Global Label

Option 2: The global label is allocated directly for source VPN (identified by the pair of { Source PE, VPN }) and advertised in the network. We call such label as Source VPN label. The flow label should be the source VPN label shown in the figure 2.

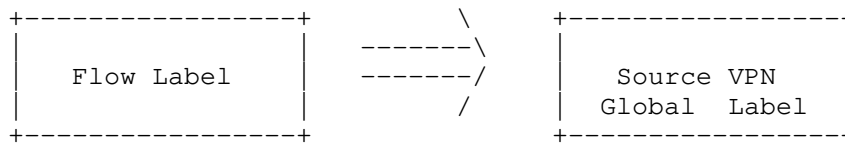


Figure 2: Flow Label using One Layer of Global Label

No matter option 1 or option 2 is adopted, when the destination receives the packets it can differentiate flows from specific source VPN based on the advertised global label binding information.

### 3.2.2. Aggregate MVPN/VPLS over Single P-Tunnel

In BGP-base Multicast VPN ([RFC6513]) and VPLS Multicast ([RFC7117]), in order to implement aggregating multiple MVPN/VPLS Instances on a single P-Tunnel (i.e. sharing one P2MP LSP), context-specific label is introduced to identify the MVPN/VPLS instance and the label binding is allocated by the root PE and advertised to the leaf PEs. In this usecase the context-specific label is one type of global label to uniquely identify the MVPN/VPLS instance in the network.

The context-specific label can solve the issue of aggregating multiple MVPNs or VPLS instances over a single P2MP LSP. But if the MP2MP LSP is adopted for aggregating multiple MVPN/VPLS instances the solution does not work since there are multiple root PEs which may allocate the same context-specific label for different MVPN/VPLS instances. In order to solve the issue the global label can be allocated to the same MVPN/VPLS instance on all PEs and advertised in the network. Then the global label will become the unique identification of VPN instance in the network. When aggregating multiple MVPNs or VPLS instances over one MP2MP LSP, the corresponding MPLS global label binding with the MVPN/VPLS instance can be encapsulated by the root PE. Then the leaf PEs can determine the MVPN or VPLS instance the received packets belong to based on the advertised global label binding information for MVPN/VPLS instances. The solution can provide the unified solution for aggregating multiple MVPN/VPLS instances over P2MP LSP and MP2MP LSP. And the solution can save the complex control plane and forwarding plane process of context-specific label.

### 3.3. Segment Routing

Segment Routing [I-D.ietf-spring-segment-routing] is introduced to leverage the source routing paradigm for traffic engineering, fast re-route, etc. A node can steer a packet through an ordered list of segments. A segment can represent any instruction, topological or



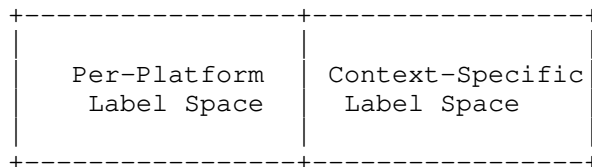
service-based. Segment Routing can be directly applied to the MPLS architecture with no change on the forwarding plane in which a segment can be encoded as an MPLS label and an ordered list of segments can be encoded as a stack of labels.

Segment Routing [I-D.ietf-spring-segment-routing] introduces some segments such as node segment, adjacency segment, etc. SR Global Block (SRGB) is also introduced for allocation of segment. In the MPLS architecture, SRGB is the set of local labels reserved for global segments. When the global segment index is advertised, it can be transited to MPLS label based on the SRGB. According to [I-D.ietf-ospf-segment-routing-extensions] and [I-D.ietf-isis-segment-routing-extensions] MPLS global label binding information can also be directly advertised in the network. For example, in the section 2.1 of [I-D.ietf-ospf-segment-routing-extensions], when the Length field of SID/Label Sub-TLV is set as 3, it will represent the label which can be flooded in the whole network. By this method MPLS global label can be directly allocated for specific node or adjacency, etc. and advertised in the network. The solution can save the complex process of SRGB advertisement and transition from the global Segment ID to MPLS label.

#### 4. Discussion

In the MPLS world, we can adopt the dichotomy to divide it into per-platform label space and context-specific label space.

##### MPLS World



When we adopt another dichotomy to divide the MPLS world into local label and global label, we may face more challenges.

MPLS World			
Local Label		vs.	Global Label
LDP (RFC 5036) RSVP-TE (RFC 3209) BGP LSP (RFC 3107) L3VPN (RFC 4364) LDP-based L2VPN (RFC 4762) EVPN (RFC 7432)			Special Purpose Label (RFC 7274)
			MPLS Upstream Label Assignment /Context-Specific Label Space (RFC 5331)
			Entropy Label (RFC 6790)
			Flow Label (RFC 6391)
			BGP-base VPLS (RFC 4761)
			Segment Routing (draft-ietf-spring-segment-routing)
			Domain-Wide Label (Usecases: Synonymous Label/ Segment Routing, etc.)

Figure 3: Division of MPLS World Using Local Label and Global Label

In the figure 3, we can easily understand the local label using for LDP, RSVP-TE, label BGP, L3VPN, LDP-based L2VPN, EVPN, etc. But for the opposite of these applications there may be many usecases which are different from each other, but share the common characteristic that the label meaning can be understood by all network nodes or part of network nodes instead of only the local downstream nodes and upstream nodes for which in this document such label is defined as global label :

-- For special purpose labels, their meaning can be understood by all nodes in the MPLS network. Should they belong to global label?

-- For MPLS upstream label assignment in context-specific label space, all downstream nodes can understand the meaning of the label allocated by the upstream node binding for specific MVPN/VPLS instance. We can see the root PE as one type to central controlled node to allocate label to all leaf nodes. And thinking about the uniqueness of the context determine by the shared P-tunnel, these labels in fact are also unique in the network. Should they belong to global label?

-- For entropy label and flow label, the label is calculated by the ingress node based on specific hash algorithms which is totally different from the local label distributed in the MPLS control plane.

And all nodes along the path will parse the label and according to the uniform meaning to use the label for ECMP. But the label values can be duplicate since they are calculated by different ingress nodes. Should they belong to global label?

-- For BGP-based VPLS and Segment Routing, they can adopt the local label block. But they introduce the global ID and transit them into the local label. Especially for segment routing, when all nodes in the network adopts the same SRGB, the global segment ID is easily transited to a unique global label value in the network. Should they belong to global label?

-- This document proposes some usecases to directly allocate the unique label value and advise the label binding in the network. Should they be directly called as global label or Domain-Wide label as one type of global label?

Since above applications which are different from the traditional MPLS local label, can we define all of them as global label or define some of them as global label and bring some use cases to the local label field? Or maybe such dichotomy using local label and global label does not exist.

## 5. IANA Considerations

This document makes no request of IANA.

## 6. Security Considerations

TBD.

## 7. References

### 7.1. Normative References

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

### 7.2. Informative References

[I-D.bryant-mpls-flow-ident]  
Bryant, S., Pignataro, C., Chen, M., Li, Z., and G. Mirsky, "MPLS Flow Identification", draft-bryant-mpls-flow-ident-02 (work in progress), September 2015.

- [I-D.bryant-mpls-synonymous-flow-labels]  
Bryant, S., Swallow, G., Sivabalan, S., Mirsky, G., Chen, M., and Z. Li, "RFC6374 Synonymous Flow Labels", draft-bryant-mpls-synonymous-flow-labels-01 (work in progress), July 2015.
- [I-D.ietf-isis-segment-routing-extensions]  
Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-05 (work in progress), June 2015.
- [I-D.ietf-ospf-segment-routing-extensions]  
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-ietf-ospf-segment-routing-extensions-05 (work in progress), June 2015.
- [I-D.ietf-spring-segment-routing]  
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and r. rjs@rob.sh, "Segment Routing Architecture", draft-ietf-spring-segment-routing-06 (work in progress), October 2015.
- [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>>.
- [RFC3107] Rekhter, Y. and E. Rosen, "Carrying Label Information in BGP-4", RFC 3107, DOI 10.17487/RFC3107, May 2001, <<http://www.rfc-editor.org/info/rfc3107>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<http://www.rfc-editor.org/info/rfc3209>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<http://www.rfc-editor.org/info/rfc4364>>.
- [RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <<http://www.rfc-editor.org/info/rfc4761>>.

- [RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <<http://www.rfc-editor.org/info/rfc4762>>.
- [RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <<http://www.rfc-editor.org/info/rfc5036>>.
- [RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, DOI 10.17487/RFC5331, August 2008, <<http://www.rfc-editor.org/info/rfc5331>>.
- [RFC6391] Bryant, S., Ed., Filsfils, C., Drafz, U., Kompella, V., Regan, J., and S. Amante, "Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network", RFC 6391, DOI 10.17487/RFC6391, November 2011, <<http://www.rfc-editor.org/info/rfc6391>>.
- [RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <<http://www.rfc-editor.org/info/rfc6513>>.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <<http://www.rfc-editor.org/info/rfc6790>>.
- [RFC7117] Aggarwal, R., Ed., Kamite, Y., Fang, L., Rekhter, Y., and C. Kodeboniya, "Multicast in Virtual Private LAN Service (VPLS)", RFC 7117, DOI 10.17487/RFC7117, February 2014, <<http://www.rfc-editor.org/info/rfc7117>>.
- [RFC7274] Kompella, K., Andersson, L., and A. Farrel, "Allocating and Retiring Special-Purpose MPLS Labels", RFC 7274, DOI 10.17487/RFC7274, June 2014, <<http://www.rfc-editor.org/info/rfc7274>>.
- [RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<http://www.rfc-editor.org/info/rfc7432>>.

## Authors' Addresses

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

Email: lizhenbin@huawei.com

Quintin Zhao  
Huawei Technologies  
125 Nagog Technology Park  
Acton, MA 01719  
US

Email: quintin.zhao@huawei.com

Tianle Yang  
China Mobile  
32, Xuanwumenxi Ave.  
Beijing 01719  
China

Email: yangtianle@chinamobile.com

Robert Raszuk  
Individual

Email: robert@raszuk.net

Luyuan Fang  
Microsoft  
5600 148th Ave NE  
Redmond, WA 98052  
USA

Email: lufang@microsoft.com

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

Z. Li  
M. Li  
Huawei Technologies  
October 21, 2013

Framework of Network Virtualization Based on MPLS Global Label  
draft-li-mpls-network-virtualization-framework-00

Abstract

As the virtual network operators develop, it is desirable to provide better network virtualization solutions to facilitate the service provision. In the past years, MPLS plays a key role in the process of implementing network virtualization. This document introduces a new framework to implement network virtualization based on MPLS global label. It can provide the virtualized network topology, nodes and links using MPLS global label which can make up the virtual network.

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 . . . . .	2
2. Terminology . . . . .	3
3. Framework . . . . .	4
4. MPLS Virtualization of Network Topology . . . . .	6
5. MPLS Virtualization of Network Nodes . . . . .	8
6. MPLS Virtualization of Network Links . . . . .	9
7. Forwarding in Virtual Network . . . . .	10
8. IANA Considerations . . . . .	12
9. Security Considerations . . . . .	12
10. References . . . . .	12
10.1. Normative References . . . . .	12
10.2. Informative References . . . . .	12
Authors' Addresses . . . . .	13

## 1. Introduction

The virtual network operators are in fast development. They can deploy possible services based on the virtual network which is provided by the underlying network. Owing to the technology limitation, the virtual network operators face following challenges:

- It is hard to get the traffic and data information of internal nodes. So it is hard to develop value-added services.
- Traditional VPN technology is just to provide a transparent pipe for virtual network operators which cannot control and manage the internal nodes.
- Traditional technologies can not implement virtualization of network nodes and links. It is hard to provide flexible virtual networks.



-- It is unable to implement central control, which is hard to provide customized virtual networks based on policies and open APIs.

For the virtual network operators, in order to provide better services it is necessary to get more control on the internal network nodes. Traditional VPN solutions is just to provide virtual networks on the network edge. This can not satisfy the new network virtualization requirement. On the other hand, the underlying network operators do not hope to expose much internal network details to the virtual network operators. Furthermore, it also exerts much burden on the virtual network operation and management if there is much internal network details for the virtual network operators.

In order to solve the problems of existing solutions and satisfy new virtual network requirements, it is desirable to provide a central controlled network virtualization solution which can provide flexible customized virtual networks easily. This document introduces a new framework to implement network virtualization based on MPLS global label. It can provide the virtualized network topology, nodes and links using MPLS global label which can make up the virtual network easily.

## 2. Terminology

**Underlying Network:** It is the network which the virtual network is built based on. The underlying network can be the physical network or the virtual network.

**MPLS Virtual Network:** The virtual network is built based on the underlying network. It is composed by virtual nodes and virtual links which are identified by MPLS global label. In this document, the concept of virtual network is the same as that of MPLS virtual network.

**MPLS Virtual Network Topology:** It is the topology of the MPLS virtual network. It can be identified by multi-topology ID of corresponding virtual network. MPLS global label is allocated to represent the virtual network topology.

**Underlying Link:** It is the link in the underlying network which the virtual link is built based on. The underlying link can be physical link or the virtual link.

**MPLS Virtual Link:** The virtual link is built based on the underlying link with specific attribute requirement. It can be identified by MPLS global label. In this document, the concept of virtual link is the same as that of MPLS virtual link.

**Underlying Node:** It is the node in the underlying network which the virtual node is built based on. The underlying node can be physical node or the virtual node.

**MPLS Virtual Node:** The virtual node is built based on the underlying node with specific attribute requirement. It can be identified by MPLS global label. In this document, the concept of virtual node is the same as that of MPLS virtual node.

### 3. Framework

MPLS is always a basic technology to implement network virtualization. L3VPN and VPLS are typical network virtualization solutions based on MPLS technologies. VPN technologies provides virtual network at the network edge based on BGP or T-LDP. In order to provide better virtual network services the internal network should be virtualized to be provided to the virtual network operators. Then IGP is a better choice to combine with MPLS technologies to provide these virtual networks.

In the MPLS virtual network, virtual nodes and virtual links are basic components. They can be represented by unique MPLS global label values. In addition, in order to differentiate virtual networks, the virtual network topology can be identified by multi-topology ID and the unique MPLS global label value can also be allocated to represent the virtual network topology. Thus the network topology, the node and the link can be virtualized by MPLS. They can hide the details of the underlying network.

The architecture to construct virtual network is shown in the following figure. There is a central controller to control network nodes. The controller can construct different virtual networks according to the requirements proposed by the virtual network operators. IGP runs among the controller and the network nodes. MPLS global labels can be allocated by the IGP controller for the virtual network topologies, the virtual nodes and the virtual links. The label binding between the MPLS global label and the virtual network topology/node/link are flooded among the controller and the network nodes. When the network nodes receive the label mapping messages, they will install corresponding MPLS forwarding entries accordingly.

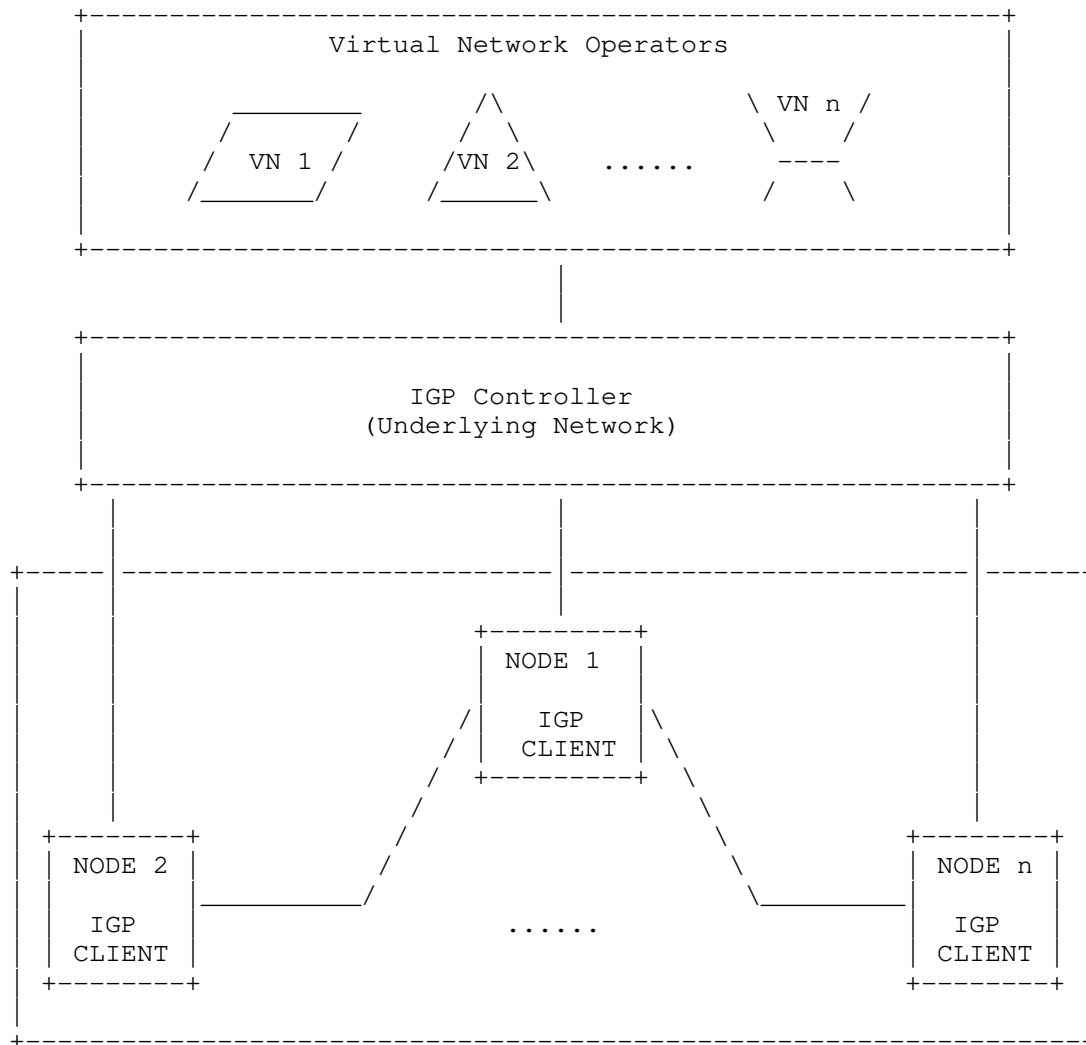


Figure 1 Architecture of MPLS Virtual Network

Figure 2 shows an example of the virtual network built based on the underlying network. The virtual network topology is represented by the Virtual Topology Global Label (VT-GL). The virtual node is represented by the Node Global Label (N-GL). The virtual link is represented by the Link Global Label (L-GL). In the virtual network shown in the figure 2, there are three virtual links identified by L-GL 1, L-GL 2 and L-GL 3 and there are three virtual nodes identified by N-GL 1, N-GL 2 and N-GL 3. All virtual nodes and links

constructs a triangle virtual topology identified by VT-GL 1. The virtual network operators can provision their own service based on the virtual network. Especially, for the virtual link, it can have common attributes such as bandwidth, MTU, etc. like the physical link. The virtual network operators need not care about the physical details of links of the virtual network. For example, the bandwidth for the virtual link is 10G. It may be an independent physical interface, or a virtual link allocating 10G bandwidth from a physical interface, or a virtual interface constructed by compositing several physical interfaces. All the details of the underlying network are hidden from the virtual network operators. This can simplify the network operation and management for the virtual network operators which can focus more on their own service provision. On the other hand, the hidden details can improve security of the underlying network to some extent.

Virtual Network 1: VT-GL 1

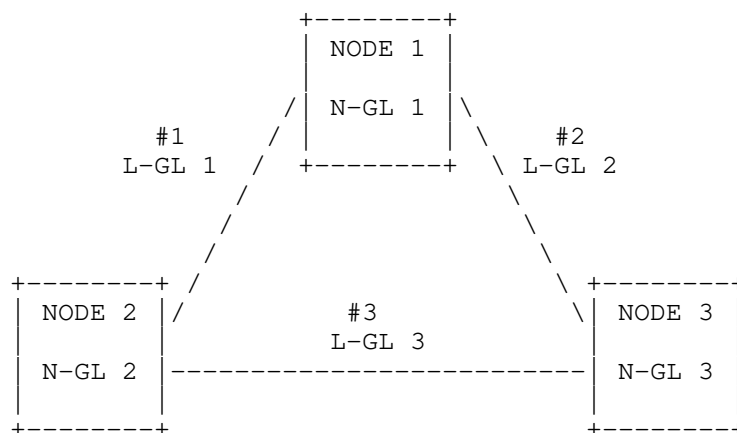


Figure 2 An Example of MPLS Virtual Network

#### 4. MPLS Virtualization of Network Topology

In essence, constructing virtual networks is to construct different virtual network topologies based on the underlying network. The virtual network topology can be identified by the Multi-Topology ID. The global label for the virtual network topology is allocated by the IGP controller. The label binding between the Multi-Topology ID and the Global Label are flooded from the IGP controller to the network nodes.

The network nodes should support the multi-topology. It can install FIBs for multi-topologies. That is, there are multiple forwarding instances in one network node. Each forwarding instance is corresponding to a virtual network topology.

When network nodes receive the label binding between the Multi-Topology ID and the Global Label, it will install one MPLS forwarding entry: The incoming label is the Global Label. It will be mapped to the forwarding instance corresponding to the Multi-Topology.

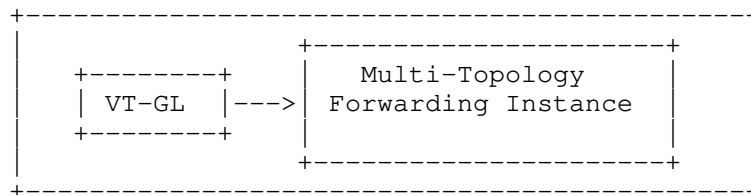
When packets of different virtual networks are forwarded in the network nodes, they must encapsulate the global label binded with the Multi-Topology, Thus the network node receiving the packet will get the label from the MPLS encapsulation and find the corresponding MPLS forwarding entry. Then the packet will be mapped to the corresponding forwarding instance to determine how to forward in the corresponding virtual network. If the packet is to be forwarded to the next hop in the virtual network, when it leaves the network node, the global label must be encapsulated again.

Step 1:

Incoming Packet

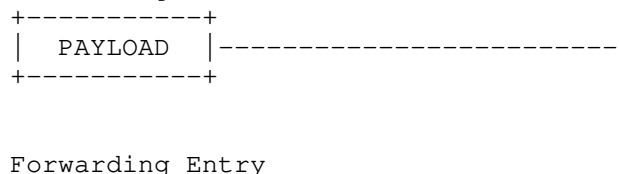


Forwarding Entry

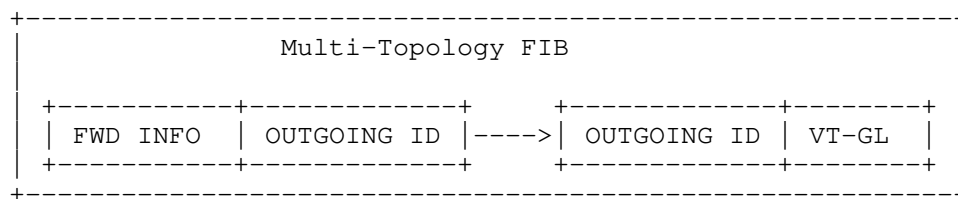


Step 2:

Transiting Packet



Forwarding Entry



Step 3:

Outgoing Packet

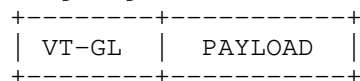


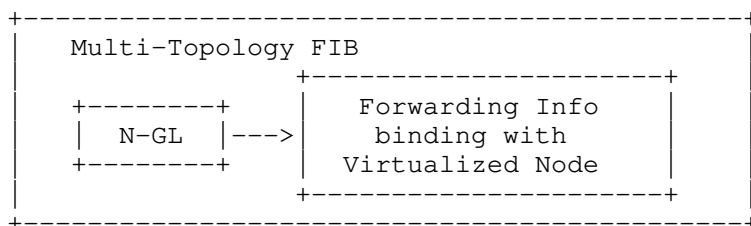
Figure 3 Forwarding Process for MPLS Virtual Topology

## 5. MPLS Virtualization of Network Nodes

MPLS Virtual nodes can be built based on the underlying node in a specific underlying network. They can be identified by unique MPLS global label allocated for the tuple { Multi-Topology ID, Underlying Node Identification, Attributes of the Virtualized Node }. Multi-topology ID is the identification of the corresponding multi-topology of the underlying network. The underlying node can be identified by the node's address (typically the loopback address) if the underlying node is the physical network node or it can be identified by another global label corresponding to the underlying virtual node. When implement virtual nodes, IGP controller will allocate the global label for the tuple { Multi-Topology ID, Underlying Node Identification, Attributes of the Virtualized Node }. Then the label binding between the tuple and the Global Label are flooded from the IGP controller to the network nodes.

When network nodes receive the label binding between the tuple and the Global Label, it will install one MPLS forwarding entry in the forwarding instance corresponding to the Multi-Topology ID: The

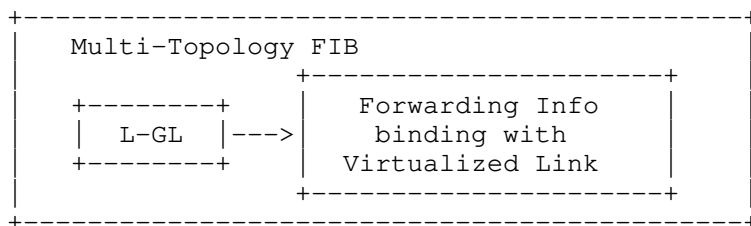
incoming label is the Global Label. It will be mapped to the forwarding information related with the virtualized nodes. The forwarding information is derived according to the specific application requirement. For example, in Segment Routing, the forwarding information can be the shortest path to the underlying node. In addition, the forwarding identification for the specified attributes to the virtual node can also be provided in the forwarding information.



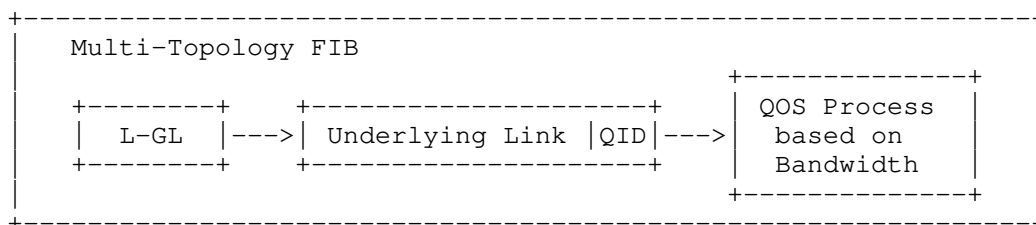
## 6. MPLS Virtualization of Network Links

MPLS Virtual links can be built based on the underlying link in a specific underlying network. They can be identified by unique MPLS global label allocated for the tuple { Multi-Topology ID, Underlying Link Identification, Attributes of the Virtualized Link }. Multi-topology ID is the identification of the corresponding multi-topology of the underlying network. The underlying link can be identified by the link ID or the link's address (typically the pair of the addresses of two end-points of the link) if the underlying link is the physical network link or it can be identified by another global label corresponding to the underlying virtual link. When implement virtual links, IGP controller will allocate the global label for the tuple { Multi-Topology ID, Underlying Link Identification, Attributes of the Virtualized Link }. Then the label binding between the tuple and the Global Label are flooded from the IGP controller to the network nodes.

When network nodes receive the label binding between the tuple and the Global Label, it will install one MPLS forwarding entry in the forwarding instance corresponding to the Multi-Topology ID: The incoming label is the Global Label. It will be mapped to the forwarding information related with the virtualized links. The forwarding information is derived according to the specific application requirement.



The typical attribute for the virtualized link is the bandwidth. When the virtual network need a virtual link with specific bandwidth requirement, IGP controller will create the virtual link by allocating the global label for the tuple {Multi-Topology ID, Underlying Link Identification, Bandwidth} and flood the label binding to the network nodes. When network nodes receive the label binding, it will reserve the bandwidth firstly based on the underlying link to provide QoS service of bandwidth guarantee. Then it will create the MPLS forwarding entry shown in the following figure:



## 7. Forwarding in Virtual Network

If the packet is forwarded in a specific virtual network, the global label binding with the virtual network topology should be encapsulated in the packet. Thus the network node receiving the packet will get the VT-GL to map to the corresponding forwarding instance to determine how to forward the packet in the virtual network.

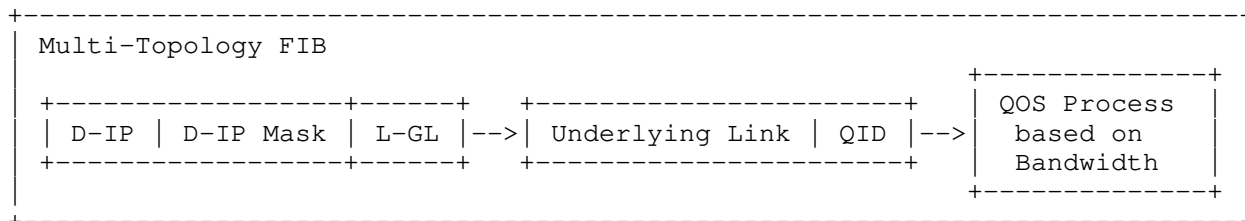
There are two ways to use the virtualized nodes and links for forwarding.

### 1. Traditional SPF or CSPF Path Calculation

The virtualized nodes and links can be added to the LSDB or be added to the TEDB after applying specific MPLS TE attributes. Then these nodes and links can be involved in the path calculation based on SPF



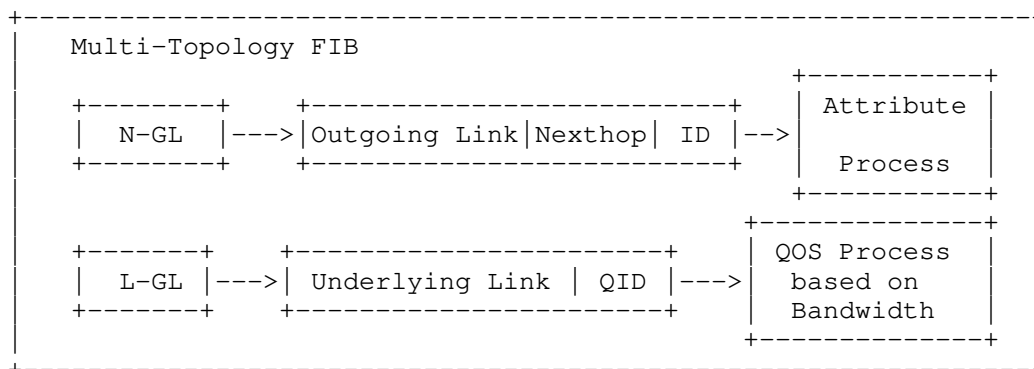
or CSPF. Then the IP forwarding entry or MPLS TE forwarding entry may be created which can use the virtual link as the outgoing link. A typical IP Routing forwarding entry is shown in the following figure:



In this case, the forwarding entry related with L-GL is not an independent entry. It is combined with other information (Destination IP address and destination IP mask in the example) to compose the forwarding entry. For packets which may use the forwarding entry, they need not encapsulate the L-GL. The L-GL is just like an internal index to link different parts of the forwarding information.

## 2. Segment Routing

The MPLS virtual nodes and links can also be used for Segment Routing. The MPLS forwarding entry for the virtualized nodes and links can be created for the Segment Routing. The MPLS virtual node is just like the Node Segment in the Segment Routing. The MPLS virtual link is just like the Adjacency Segment in the Segment Routing. The difference is that MPLS global label is used for the Adjacency instead of the local label since in the virtual network the unique identification based on the MPLS global label can simplify the network operation and management. In addition, there are specific attributes for the virtual link and virtual node, there should be forwarding process identification of the corresponding attribute in the forwarding entry. The typical Segment Routing forwarding entry is shown in the following figure:



In this case, the forwarding entry related with N-GL or L-GL is the independent MPLS forwarding entry. For packets which may use the forwarding entry, they must encapsulate the N-GL or the L-GL.

## 8. IANA Considerations

This document makes no request of IANA.

## 9. Security Considerations

TBD.

## 10. References

### 10.1. Normative References

[I-D.li-rtgwg-cc-igp-arch]

Li, Z., Chen, H., and G. Yan, "An Architecture of Central Controlled Interior Gateway Protocol (IGP)", draft-li-rtgwg-cc-igp-arch-00 (work in progress), October 2013.

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

### 10.2. Informative References

[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-00 (work in progress), June 2013.

[I-D.li-mpls-global-label-framework]

Li, Z., Zhao, Q., and T. Yang, "A Framework of MPLS Global Label", draft-li-mpls-global-label-framework-00 (work in progress), July 2013.

Authors' Addresses

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

Email: lizhenbin@huawei.com

Ming Li  
Huawei Technologies  
2330 Central Expressway  
Santa Clara, CA 95050  
USA

Email: mli@huawei.com

Routing Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 16, 2014

R. Shakir  
BT  
D. Vernalis  
Vodafone  
A. Capello  
Telecom Italia  
July 15, 2013

Performance Engineered LSPs using the Segment Routing Data-Plane  
draft-shakir-rtgwg-sr-performance-engineered-lsps-00

Abstract

A number of applications and services running over IP/MPLS networks have strict requirements relating to their routing, or the performance of the path supporting their traffic flow, for instance, in terms of characteristics such as latency, loss, or bandwidth availability. Segment routing provides a means by which the data-plane of an IP/MPLS network can be programmed to support such "performance engineered" paths. This document describes an architecture for the use of such performance engineered label switched paths, and the control-plane functionality required to allow both distributed and centralised computation of acceptable forwarding paths.

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 16, 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. Motivation . . . . .	3
2. Conventions Used in This Document . . . . .	6
3. Data-plane Path Selection . . . . .	7
3.1. SID Selection for Non-Revertive Services . . . . .	7
3.2. SID Selection for Revertive Services . . . . .	8
3.2.1. Procedure for Link Protection of Adj-SIDs . . . . .	8
3.2.2. Procedure for Node Protection of Adj-SIDs . . . . .	9
3.2.3. Example of Revertive Adj-SID Protection . . . . .	10
3.3. Path Re-Optimisation and Re-Routing . . . . .	11
4. Distributed Path Computation via Constrained Shortest-Path Algorithms . . . . .	13
4.1. Path Selection based on Static IGP Path Attributes . . . . .	13
4.2. Path Selection based on Performance-Related IGP Path Attributes . . . . .	13
4.2.1. Requirements for IGP Attributes Pertaining to Adjacency Performance . . . . .	14
5. Centralised Path Computation using PCE . . . . .	16
5.1. Use of Path Computation Element to Provide Inter-Area SR LSPs . . . . .	16
5.2. Providing Co-routed or Multi-Layer Aware LSPs using PCE . . . . .	17
5.2.1. Co-Routed LSPs . . . . .	17
5.2.2. Resource Reservation and Admission Control through a Stateful PCE . . . . .	18
5.2.3. Multi-Layer Calculation through a Common PCE . . . . .	18
6. Security Considerations . . . . .	20
7. Acknowledgements . . . . .	21
8. Normative References . . . . .	22
Authors' Addresses . . . . .	24

## 1. Motivation

For numerous applications running over IP/MPLS networks, there is a requirement to provide paths that have guaranteed network performance. These resources guarantees may be in terms of sufficient bandwidth being available for a traffic flow, but also can be in terms of other characteristics (such as latency, packet delay variation, and packet loss). In addition to such characteristics of the underlying network, requirements related to path routing can exist to ensure that a path offers characteristics such as affinity to particular infrastructure, or disjointness to another service. For instance:

- o Where two services provided by an IP/MPLS network make up part of an active/backup or live/live service pair for a transported application it is required that the paths are wholly disjoint (shared risk, link, and node) to ensure that they do not fail simultaneously.
- o If the service a network provides supports an application that requires a particular end-to-end latency budget, then the service must be constrained to a path, or paths, meeting this budget and the path made unavailable if these characteristics cannot be met.
- o Where a service provided by the IP/MPLS network makes up part of another network's topology (e.g., an ATM PWE3 service provided within an IP/MPLS network may form a part of a wider client ATN network), then an affinity to particular links within the network (such as particular sub-sea cable systems) may be required. Where such a path is not available, it can be preferable to utilise protection within the client network rather than re-route the IP/MPLS service.
- o Where services, such as paths for real-time voice and video trunking delivered over IP/MPLS services are routed according to paths with guaranteed resource availability (such as available bandwidth).
- o Where the service requires that both directions of the network path are co-routed, such as where an IP/MPLS network path is used to carry IEEE 1588 synchronisation traffic. In this case, two uni-directional LSPs must be routed in a co-ordinated manner, which may diverge from the shortest-path within the network.

These requirements can be generalised into a need to support arbitrary constrained paths within an IP/MPLS network - with the constraints being both in terms of the path selected in the network (and its underlying characteristics) and the treatment of packets

forwarding onto this path during network events (such as during link failures, or re-convergence).

It is important to note that these routing requirements are inherently related to a particular service (e.g., customer service A must be disjoint to customer service B, or a customer service must only be available if paths with an end-to-end latency of less than 200 milliseconds are available between node X and node Y) - and apply to all traffic (as may be the case within a pair of PWE3 services) or a subset of traffic within the service (as may be the case with particular treatment of voice traffic within an L3VPN service). This results in the number of such routed paths in the network being dependent upon the number of services supported by the network (order of tens or hundreds of thousands), rather than to the number of edge devices within a network (typically of the order of hundreds, or thousands).

It is possible to utilise Segment Routing (SR) as described in [I-D.filsfils-rtgwg-segment-routing] to provide a means to support services with constrained path requirements via label-switched paths (LSPs) in an IP/MPLS network without requiring devices within the core of the network to maintain per-LSP state. Since in the limits described above, this number of LSPs is proportional to the number of services on the network utilising a stateless mechanism (such as SR) to provide such forwarding paths equates to avoiding per-service state on transit devices. The forwarding paths utilised to support such services are referred to as performance engineered LSPs within this document.

For example, considering the topology shown in Figure 1, if the ingress label edge router (LER) requires forwarding of traffic on a path to the egress LER which does not exceed 40 milliseconds, it must ensure that traffic traverses the iLER-A, A-B, and B-eLER links.

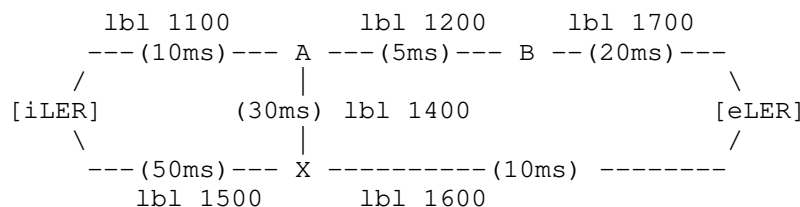


Figure 1

With segment routing, iLER can apply a label stack of {1200,1700} and next-hop of A to influence A to forward packets to B, and B to

forward to eLER, regardless of the shortest path selection due to the IGP metrics within the network topology.



## 2. Conventions Used in This Document

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. Data-plane Path Selection

When considering services that are to be carried via performance engineered paths within a network, constraints are introduced relating to protection during failures of the explicit path selected through the network. Particularly, where a path is provided with particular link affinity, or as part of an active/active service pair where both primary and backup LSPs are routed within certain performance constraints, it is not desirable to perform dynamic re-routing or fast re-route protection for the traffic within the service, as a path violating the performance constraints may be introduced, where the alternate route made available continues to be compliant with the original routing criteria. In order to allow an operator to route services not requiring reversion from the primary path, an implementation **MUST** allow the advertisement of segment identifiers which are explicitly excluded from fast re-route, and reversion away from the primary path.

Where services are specified as being suitable for reversion, some consideration is required as to the means by which they are re-routed. For instance, where with some IP FRR mechanisms the protection path may follow the shortest-path tree from the point of local repair (PLR) to the packet destination (effectively making the LSP tail-end the merge point - MP), such re-routing behaviour is not desirable for all performance engineered paths. In such cases, it is more preferable to provide means by which (during protection events) traffic is routed back on to the original LSP path, as soon as possible, essentially minimising the divergence away from the path calculated to meet service constraints. An implementation **SHOULD** provide means by which an operator can influence MP selection to support such requirements.

#### 3.1. SID Selection for Non-Revertive Services

Following the calculation of a route meeting a particular set of constraints, the ingress service edge device should select the data path through determining the relevant SIDs to be pushed to the packet. During the translation of a route to a set of SIDs the ingress device **MUST** consider the service requirements in order to ensure that protection within the network does not result in path constraints being violated where this is unacceptable. Where a service is specified to be non-revertive, the iLER **MUST** utilise only segment identifiers which are explicitly identified to be non-revertive. Since protection requirements vary on per-service basis, the control of reversion to other paths during failures **SHOULD** be specified on a per-LSP basis on the ingress router.

It should be noted that where strict path constraints are required,

this results in the number of SIDs applied to a packet being proportion to the number of links traversed through the topology (through disallowing use of Node-SIDs), which may have implications for certain hardware implementations.

A network operator MAY create forwarding adjacencies consisting of multiple SIDs utilising the Explicit Route Object encoding of the Adjacency-SID specified in [I-D.previdi-isis-segment-routing-extensions] for IS-IS and [I-D.psenak-ospf-segment-routing-extensions] for OSPF, such that the total SID-stack to be imposed is minimised. Where such forwarding adjacency LSPs (FA-LSPs), or other paths consisting of multiple segments are re-advertised, the path characteristics of the underlying links MUST be included, such that other constraints used for calculation (such as shared risk link groups) can be considered by the calculating iLER. In order that the routing of non-segment routed traffic and devices not supporting SR is not influenced by the advertisement of such adjacencies:

- o It MUST be possible for an operator to specify policies as to whether such forwarding-adjacencies are utilised for non-Segment Routed traffic within the IP/MPLS network.
- o The advertisement of FA-LSPs for the use of performance engineered LSPs SHOULD NOT negatively impact the scaling and performance of devices running vanilla SPF calculations of the network topology (in order to avoid introducing additional computational overhead to legacy devices within the IGP domain within which such LSPs are to be introduced).

### 3.2. SID Selection for Revertive Services

#### 3.2.1. Procedure for Link Protection of Adj-SIDs

By default, Adj-SID values refer to an particular individual or set of physical or logical adjacencies between two devices. It is therefore linked specifically to a specific path between two nodes , and hence (by default) does not have a viable alternate route. Where a revertive Adj-SID is advertised, specified through the 'B' flag of the Adj-SID advertisement described in [I-D.previdi-isis-segment-routing-extensions] and [I-D.psenak-ospf-segment-routing-extensions] the advertising LSR MUST calculate a backup path for this adjacency.

By default an LSR SHOULD calculate a link-protecting tunnel to the node to which the adjacency is received on - this can be achieved through mechanisms such as Loop-Free Alternates [RFC5286]. During failure of a path advertised with a revertive Adj-SID, the LSR

detecting the adjacency failure should act as the point of local repair (PLR) and SHOULD pop the adjacency segment (as per the default Adj-SID action). In order to reach the merge point (MP), it is possible for the PLR to utilise either:

- o A set of SIDs relating to the loop free alternate path to reach the MP - in this case, it should be noted that such a set of SIDs may relate to multiple node and/or adjacency SIDs, where a Remote or Directed LFA is required to reach the MP.
- o The adjacency segments relating to the calculated path between the PLR and the MP. Utilising Adj-SIDs requires the PLR to perform no calculation of the path between its neighbours and the MP, however, may result in a less survivable service, in cases where simultaneous failures result in the backup SR-LSP specified by the set of Adj-SIDs becoming unavailable.

In cases where particular policies should be enforced for the protection path for an Adj-SID, an implementation SHOULD utilise a set of Adj-SIDs that indicate the links to be traversed between the PLR and the MP, based on characteristics of these adjacencies (e.g., the maximum total link bandwidth path). Where such Adj-SID based backup path selection is utilised, the path selected SHOULD be influenced by operator policy in a similar manner as the LFA selection considered in [I-D.ietf-rtgwg-lfa-manageability].

It should be noted that since the selection of the protecting set of SIDs is calculated on a per-Adj-SID basis, no particular backup path selection can be performed by a transit LSR on a per-service basis. Therefore, where revertive SIDs are utilised an operator SHOULD recognise that during protection events, no path characteristics, or resource constraints can be met whilst re-routing results in the service diverging from the specified explicit path.

### 3.2.2. Procedure for Node Protection of Adj-SIDs

An LSR MAY provide node-protection for an Adj-SID if such a node-protecting path exists within the network topology.

In the case where such a path is available, the LSR acting as the PLR must be capable of programming its forwarding plane based on the tuple of the top two labels of the SID stack. Where this can be achieved, a backup action to push the corresponding set of SIDs to reach the next-next-hop node (indicated by the advertising entity of the second entry in the label stack) during the failure of the primary Adj-SID. The PLR must therefore program an action to pop the first two labels of the ingress packet, and subsequently push the SIDs relating to this path.

Again, it is possible for a node to utilise either a set of Node or Adjacency SIDs to reach the next-next-hop node (MP). Where Node-SIDs are utilised, means to determine a loop-free path MUST be used to determine the set of Node-SIDs required. Where Adj-SIDs are utilised for such functionality, no such calculation is required. Where certain policies are to be enforced for the protecting path, an implementation SHOULD allow the use of Adj-SIDs to determine the path utilised, and the selection of these SIDs SHOULD be influenced by operator policy.

### 3.2.3. Example of Revertive Adj-SID Protection

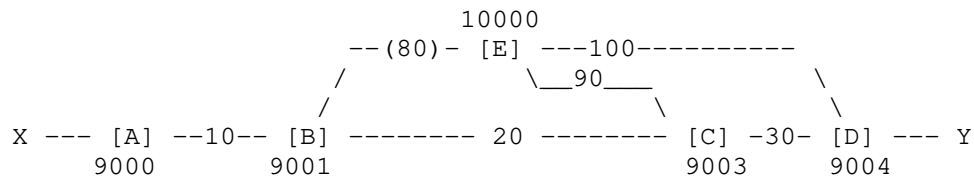


Figure 2

In Figure 2 a source X sends traffic utilising Adj-SIDs to a destination Y utilising through pushing the relevant Adj-SIDs to traverse A-B, B-C, C-D. A therefore receives a packet with {10,20,30} segments applied. B's FIB is programmed with a pop() operation for Adj-SID 20, along with a next-hop of the B-C link.

To provide a link-protecting backup, if E has been calculated as a link-protecting LFA for segment 20, then B programs a backup action of push(9003) for the ingress label of 20, with an egress interface of the B-E link. When E receives this labelled packet, it swaps label 9003 for label 9003 (as per standard behaviour for a Node-SID), and forwards directly to C, who receives the packet with the label 30 exposed, and hence acts as the MP.

Clearly, an alternative is that B programs a backup action to push label 90 to this stack (with a next-hop of E) to ensure that the adjacency between E-C is utilised, rather than E's IGP shortest-path to C.

To provide node protection for the Adj-SID, then additional complexity is introduced. If B receives a packet destined for the Adj-SID indicating link B-C, B can examine the following label within the stack (in this case, label 30) which is advertised by D within the topology. Since there is a node-protecting LFA via E to D, B may therefore pop() the subsequent Adj-SID and push the Node-SID of D (9004), whilst forwarding the packet via the B-E link.

Alternatively, it is possible for B to utilise an explicitly derived path to reach D (namely, the Adj-SID of the E-D link, with a next-hop of E), to reach the MP. In this case, no calculation as to the routing behaviour of E is required to determine this protection path (trading computational complexity for increasing the length of the protection SID stack). Through this behaviour, a packet can be forwarded during the complete failure of C. It should be noted that this requires two look-ups on the PLR rather than the single look-up required in the link protecting case.

### 3.3. Path Re-Optimisation and Re-Routing

Where performance-engineered SR paths are selected by a head-end, the calculation of the path is based on the information that is available to the computing entity (be it centralised or distributed) at the time of calculation. In order to ensure that such paths are re-routed onto more optimal paths where available, an ingress LER MUST perform periodic re-optimisation whereby the path selected for a service is recalculated. The period between such re-optimisations SHOULD be configurable by a network operator.

Unlike other network technologies which can be utilised to specify explicit paths within an IP/MPLS network, the mid-point network elements are unaware of the LSPs that traverse them. There is therefore a requirement for an SR head-end to determine when specific segments are no longer valid to be utilised for a service to be routed. In order to ensure that traffic is forwarded onto paths that remain valid, a head-end device MUST trigger re-calculation of explicit paths within the network when it receives an IGP update relating to the segment utilised within a particular service topology. In order to provide means to tolerate short-lived failures (particularly where such services are revertive), it SHOULD be possible to delay such recalculation on a per-service basis. Such triggered re-optimisation MUST be performed for IGP updates that withdraw segments from the topology and MAY be triggered based on updates to other attributes within the network. Where updates are triggered on information that may rapidly change within the IGP (e.g., information relating to bandwidth reservation or utilisation) an iLER device SHOULD provide means to limit the period between re-optimisations, or provide thresholds over which re-optimisation is triggered.

In addition to re-optimisation based on failures, an iLER SHOULD provide means by which per-service OAM measuring performance or liveness characteristics of a particular path can trigger a path to be withdrawn from use, and/or re-optimisation of the SID selection for the path. Such per-service OAM is critical within multi-area environments where it cannot be guaranteed that a head-end device

will have all routing information propagated to it - in such deployments, an implementation MUST support per-service OAM. In all environments, per-service OAM can be utilised to ensure that a service can be withdrawn more quickly than IGP-updates relating to segment failures can be propagated, or a head-end is able to react to "grey" failure events, where data-plane traffic forwarding has failed, but no IGP update is generated.

#### 4. Distributed Path Computation via Constrained Shortest-Path Algorithms

##### 4.1. Path Selection based on Static IGP Path Attributes

To determine the relevant set of segment identifiers to be utilised for a service, existing constrained shortest-path (CSPF) functions such as those used for other route selection mechanisms can be utilised. This can provide route selection based on IGP traffic engineering metrics (such as those specified for OSPF in RFC3630, or IS-IS in RFC5305), or GMPLS IGP extensions such as shared risk link groups (SRLGs) carried in attributes such as those that are described for OSPF in RFC5307 and IS-IS in RFC4203. An implementation providing distributed CSPF to provide performance-engineered SR paths SHOULD support path selection through consideration of such traffic engineering IGP attributes.

Where adjacency segments are created for use as forwarding adjacency LSPs, or as a means to provide compression of SID stacks, an implementation MUST include the relevant IGP traffic engineering attributes indicating the characteristics of the underlying Adj-SIDs within IGP attributes relating to such segments.

##### 4.2. Path Selection based on Performance-Related IGP Path Attributes

In addition to considering such static attributes of links within the IGP topology, distributed path computation can be triggered based on performance monitoring information propagated into IGP attributes such as those described in [I-D.giacalone-ospf-te-express-path] and [I-D.ietf-isis-te-metric-extensions]. Consideration of such attributes allows paths to be calculated based on the underlying loss and delay characteristics of a network path. Through monitoring updates to these attributes advertised through IGP update messages, re-routes based on changes in the performance characteristics of a path can be achieved. An iLER supporting performance engineered LSPs utilising the SR dataplane SHOULD allow consideration of these attributes when performing ERO calculation, and SHOULD provide means to trigger re-routes based on changes in their values.

In many implementations of MPLS Traffic Engineering, a mechanism referred to as "auto-bandwidth" is implemented. In this case, the traffic forwarded via a particular label switched path signalled by RSVP-TE is monitored, and the utilisation observed over a set period of time utilised as the bandwidth requested when a service is periodically re-optimised. Whilst a SR-based implementation cannot provide control-plane resource reservation based on this approach, through monitoring these attributes, three forms of bandwidth-aware routing can be achieved:



- o Least-Fill - When selecting an particular path for a service to be routed by, where the service has affinity to an individual link within an ECMP, a typical means to ensure balancing of traffic between the different candidate links is to route the service via the link within the ECMP that is least utilised. During the computation of a Segment ERO, an ingress LSR SHOULD provide means by which such services can select the least utilised link from an set of ECMP candidate links through consideration of the Available Bandwidth sub-TLV within such IGP extensions.
- o Reaction to bandwidth utilisation within the network to re-route services based on load of links. In this case, through monitoring a set of particular (potentially high-bandwidth) services against the bandwidth utilisation of the links that they follow, it is possible to re-optimize the routing of services such that traffic is re-routed away from links experiencing congestion in a reactive manner.
- o Reaction to the bandwidth consumed per-service - for instance, in cases where traffic is routed via a network with mixed maximum link bandwidths (e.g., some paths may have a maximum of 2.5Gbps where others have a maximum of 10Gbps) it is advantageous for a head-end device to split traffic flows into multiple sub-elements, with some diverging from the SPT. In this case, no knowledge of the utilisation of the network is required, however, the maximum available bandwidth of adjacencies within the SPT combined with explicit routed LSPs can be utilised to achieve traffic balance across the network.

Through utilising the uni-directional residual and available bandwidth TLVs described in the aforementioned performance attributes, the current utilisation (or available bandwidth remaining on a link) can be considered within a path calculation. An SR implementation providing performance engineered LSPs SHOULD provide means by which residual or available bandwidth can be utilised as a means to calculate an ERO, and trigger subsequent re-routing. Where re-routes are triggered based on available bandwidth an iLER MUST provide means by which the time between re-optimisations can be limited, and SHOULD provide means by which such recalculations can be jittered, such that periodic re-optimisation is not performed simultaneously for all LSPs on a particular iLER.

#### 4.2.1. Requirements for IGP Attributes Pertaining to Adjacency Performance

The use of extended IGP attributes to determine underlying path characteristics for the selection of performance engineered paths requires some considerations to ensure that the routing information

utilised is sufficient and timely - and to balance this accuracy against resource utilisation of the systems within the IGP.

Where dynamically measured performance statistics are advertised into the IGP - such as latency measurement, or bandwidth utilisation - there is a requirement to ensure that a head-end performing re-routing of an LSP calculates performance in a manner which balances:

- o Accuracy of the resource consideration at the time of routing - ensuring that the current performance of the adjacency meets the path selection criteria. This requirement may lead to frequent updates of performance information into the IGP - hence, in order to minimise the impact to the overall network system, a receiving implementation in such a network SHOULD provide means by which such updates do not result in a recalculation of (the complete, or a subset of) the network's topology. In some networks, especially those with legacy systems, it is not possible to make such changes to all elements within the IGP, and therefore an advertising implementation MUST provide means by which the flooding of bandwidth information can be limited to cases where particular (operator specified) thresholds in performance are exceeded.
- o Consideration of the medium-term performance of the network link - for instance, where residual bandwidth-based path selection is to be performed, it is of advantage to consider both the instantaneous bandwidth utilisation, along with the measured average over a previous time period such that the longer-term performance guarantees can be considered during route selection. Whilst such criteria does not provide strict admission control for services, it provides means by which further accuracy can be added to calculations based on instantaneous measures. To this end, an advertising implementation SHOULD provide a moving average performance measure when advertising real-time performance information within the IGP, where such attributes are not available.

In some cases, it may be advantageous for distributed path selection to consider per-forwarding class performance - in these cases an advertising implementation MAY provide performance measures on a per-configured forwarding class basis for a particular adjacency.

## 5. Centralised Path Computation using PCE

In addition to the utilisation of distributed computation, existing PCE mechanisms can be provided to provide centralised path computation for performance engineered services. Such PCE-based computation have utility both for providing inter-area or multi-layer aware information, alongside providing globally aware service functions.

It is envisaged that the interface between the PCE and head-end LSR utilises interfaces which may:

- o Exploit the Path Computation Element Protocol described in RFC5440.
- o Utilise other real-time protocols providing interaction between forwarding elements and centralised routing entities such as those described in [I-D.amante-i2rs-topology-use-cases].

Where an implementation provides support for performance engineered LSPs it SHOULD provide means by which a remote path calculation entity can be utilised to provide both explicit route object consisting of IP addresses that can be translated into SIDs by the iLER and SHOULD support receiving a set of SID values directly from the PCE.

### 5.1. Use of Path Computation Element to Provide Inter-Area SR LSPs

In a multi-area network deployment, where there is restricted information propagated into stub areas, an iLER within the stub area does not have full visibility of the Adj-SIDs required to build a particular network path. Such a LER is therefore unable to determine (based on distributed computation) which SIDs should be utilised for a path to a remote node. Whilst one solution to providing such visibility is to implement a single-area IGP, or propagate all topology information to all areas, non-engineering constraints can prevent such implementations. Through utilising a PCE which has information relating to the SIDs within the network, any iLER may be provided with the relevant SIDs create a particular path through the network.

In such a deployment, the PCE element MUST have a live view of the IGP topology for all areas. This allows knowledge of the SIDs that are to be utilised to the head-end. It is envisaged that such information be provided to the PCE through interfaces such as BGP-LS as described in [I-D.ietf-idr-ls-distribution], with extensions to encode Segment Routing IGP attributes within the information propagated.

Since it is not only during signalling that visibility into the IGP topology is required, a PCE supporting such SR LSPs MUST offer functionality to inform the ingress LER supporting the SR LSP of a change to the underlying path of the LSP. For non-revertive LSPs, an iLER SHOULD offer a mechanism by which a secondary (backup) path can be requested for a service, which can be switched to based on local failure detection mechanisms (such as in-band OAM) to allow fast restoration of a service independent on interaction with the PCE at the time of failure.

## 5.2. Providing Co-routed or Multi-Layer Aware LSPs using PCE

### 5.2.1. Co-Routed LSPs

For a number of use-cases, there is a requirement for a path (be it a complete service, or a subset of traffic within a service) to be routed according to the route of another service within the network. For example:

- o Where there is a requirement diversity between a pair of services within a network - particularly where there services are instantiated on different iLER devices. In such cases, global visibility is required in order to jointly route two the services in a manner such that they are diverse (SRLG, link, and node) to one another (in addition to meeting any other performance constraints required of them).
- o Where two services form a bi-directional service within an IP/MPLS network. In this case, some services (e.g., those supporting BFD for end-to-end monitoring) may have a requirement for a return path from the eLER to the iLER which requires similar performance characteristics to the path from the iLER to eLER. Other services have tighter coupling requirements, such that the forwarding path used from iLER to eLER is symmetrical (i.e., utilises the exact same links) to the return path.

In both cases, there is a requirement for association between a set of LSPs, which span multiple head-end LERs. An implementation supporting such co-routing requirements MUST support the use of a stateful PCE, such as that described in [I-D.ietf-pce-stateful-pce] to provide calculated paths for such services.

In cases where an LSR initiates a path computation request, it MUST be possible to communicate associated paths, and relationship between the calculated path and the associated paths (in terms of co-routing or disjointness) to the PCE device. Both PCE and LSRs SHOULD provide means by which the associated paths can be specified in terms of an arbitrary service identifier (e.g., the service provider's "circuit

identifiant").

Since associated LSPs may also have performance requirements, it MUST be possible for an LSR to communicate performance constraints along with associations. Where a bi-directional service is specified, path computation SHOULD support the communication of common, or differing, performance requirements for each LSP within the path.

Such PCE functions MUST provide means by which the protection requirements for a particular service can be specified - both in terms of the expected reversion behaviour for the instantiated SR-LSPs and requirements for any supporting paths (e.g., path protection).

#### 5.2.2. Resource Reservation and Admission Control through a Stateful PCE

Implementing explicit path routing via the segment routing data-plane, rather than alternatives such as RSVP-TE, results in the inability of transit LSR devices to provide admission control (since it is unaware of the existing flows and their resource reservations). For some premium applications - such as carrying broadcast traffic - the reservation of bandwidth (and its guarantee via the relevant data-plane queueing configuration) continues to be a requirement.

A stateful PCE can be utilised to perform admission control into one or more forwarding classes - allowing reservation to be achieved for these services. Where reservations are required, an iLER MUST provide means by which a bandwidth reservation in (one or more) classes can be requested of the PCE. LERs supporting such requests MUST provide means by which the resources requested for a particular SR-LSP can be statically configured by an operator, and SHOULD provide means by which dynamic observation of traffic forwarded via an LSP can be used in the subsequent requests (in order to achieve PCE-controlled auto-bandwidth functionality).

#### 5.2.3. Multi-Layer Calculation through a Common PCE

A further case in which such PCE-based computation can be utilised is to provide correlation between layers of network infrastructure. For instance, where a common network model is available to a PCE across an optical and IP/MPLS network infrastructure, SRLG diverse paths can be provided without the requirement to encode this information (or communicate it) between the layers of the network. Through utilising a PCE able to support such multi-layer optimisations SRLG-disjoint paths can be computed and provided to iLERs for use as performance engineered paths.

Implementations providing PCE-based calculation with multi-layer awareness SHOULD provide means by which an arbitrary SRLG identifier can be provided to the PCE to allow path calculation.

## 6. Security Considerations

TBC

## 7. Acknowledgements

The authors would like to thank Clarence Filsfils, Pierre Francois, Hannes Gredler, and Siva Sivabalan for their feedback and suggestions pertaining to this document.



## 8. Normative References

- [I-D.amante-i2rs-topology-use-cases]  
Amante, S., Medved, J., Previdi, S., and T. Nadeau,  
"Topology API Use Cases",  
draft-amante-i2rs-topology-use-cases-00 (work in  
progress), February 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-00 (work in  
progress), June 2013.
- [I-D.giacalone-ospf-te-express-path]  
Giacalone, S., Ward, D., Drake, J., Atlas, A., and S.  
Previdi, "OSPF Traffic Engineering (TE) Express Path",  
draft-giacalone-ospf-te-express-path-02 (work in  
progress), September 2011.
- [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-03  
(work in progress), May 2013.
- [I-D.ietf-isis-te-metric-extensions]  
Previdi, S., Giacalone, S., Ward, D., Drake, J., Atlas,  
A., and C. Filsfils, "IS-IS Traffic Engineering (TE)  
Metric Extensions",  
draft-ietf-isis-te-metric-extensions-00 (work in  
progress), June 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-05 (work in progress),  
July 2013.
- [I-D.ietf-rtgwg-lfa-manageability]  
Litkowski, S., Decraene, B., Filsfils, C., and K. Raza,  
"Operational management of Loop Free Alternates",  
draft-ietf-rtgwg-lfa-manageability-00 (work in progress),  
May 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.

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

[RFC5286] Atlas, A. and A. Zinin, "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, September 2008.

Authors' Addresses

Rob Shakir  
BT  
pp. C3L, BT Centre  
81, Newgate Street  
London EC1A 7AJ  
UK

Email: [rob.shakir@bt.com](mailto:rob.shakir@bt.com)  
URI: <http://www.bt.com/>

Danny Vernals  
Vodafone  
Melbourne Street  
Leeds LS2 7PS  
UK

Email: [danny.vernals@vodafone.com](mailto:danny.vernals@vodafone.com)  
URI: <http://www.vodafone.com/>

Alessandro Capello  
Telecom Italia  
Via Reiss Romoli, 274  
Torino 10148  
Italy

Email: [alessandro.capello@telecomitalia.it](mailto:alessandro.capello@telecomitalia.it)  
URI: <http://www.telecomitalia.com/>



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: March 10, 2014

X. Xu  
Huawei  
S. Kini  
Ericsson  
S. Sivabalan  
C. Filsfils  
Cisco  
September 06, 2013

Signaling Entropy Label Capability Using Interior Gateway Protocols  
draft-xu-mpls-el-capability-signaling-igp-00

Abstract

Multi Protocol Label Switching (MPLS) has defined a mechanism to load balance traffic flows using Entropy Labels (EL). An LSR inserts the EL Indicator and the EL label only if the LSR that pops them has the capability of processing them. This draft defines a mechanism to signal that capability using link state Interior Gateway Protocols (IGP). This mechanism is useful when the label advertisement is also done via that IGP.

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 March 10, 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 . . . . .	2
1.1. Requirements Language . . . . .	2
2. Abbreviations and Terminology . . . . .	3
3. Advertising ELC using OSPF . . . . .	3
4. Advertising ELC using ISIS . . . . .	3
5. Acknowledgements . . . . .	3
6. IANA Considerations . . . . .	3
7. Security Considerations . . . . .	3
8. References . . . . .	3
8.1. Normative References . . . . .	4
8.2. Informative References . . . . .	4
Authors' Addresses . . . . .	4

## 1. Introduction

Multi Protocol Label Switching (MPLS) has defined a method in [RFC6790] to load balance traffic flows using Entropy Labels (EL). An LSR inserts the EL Indicator and the EL only if the LSR that pops those labels has the capability of recognizing and processing them. [RFC6790] defines the signaling of this capability (a.k.a Entropy Label Capability - ELC) via signaling protocols. Recently, mechanisms are being defined to signal labels via link state Interior Gateway Protocols (IGP) such as OSPF [I-D.psenak-ospf-segment-routing-extensions] and ISIS [I-D.previdi-isis-segment-routing-extensions]. In such scenarios the signaling mechanisms defined in [RFC6790] are inadequate. This draft defines mechanisms to signal the ELC using the link state advertisements (LSA) of the IGPs OSPF and ISIS. These capabilities are advertised for the entire router and not just a single prefix. This mechanism is useful when the label advertisement is also done via that IGP.

### 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. Abbreviations and Terminology

This memo makes use of the terms defined in [RFC6790], [RFC4970] and [RFC4971].

## 3. Advertising ELC using OSPF

The OSPF Router Information (RI) Opaque LSA defined in [RFC4970] is used by OSPF routers to announce their capabilities. A new TLV within the body of this LSA, called ELC TLV is defined to advertise the capability of the router to process the ELI and EL. Its formatting follows that described in sec 2.1 of [RFC4970]. This TLV is applicable to both OSPFv2 and OSPFv3. The Type for the ELC TLV needs to be assigned by IANA and it has a Length of zero. The scope of the advertisement depends on the application but it is recommended that it SHOULD be AS-scoped.

## 4. Advertising ELC using ISIS

The IS-IS Router CAPABILITY TLV defined in [RFC4971] is used by IS-IS routers to announce their capabilities. A new sub-TLV of this TLV, called ELC sub-TLV is defined to advertise the capability of the router to process the ELI and EL. It is formatted as described in [RFC5305] with a Type code to be assigned by IANA and a Length of zero. The scope of the advertisement depends on the application but it is recommended that it SHOULD be domain-wide.

## 5. Acknowledgements

The authors would like to thank TBD for their comments.

## 6. IANA Considerations

This memo includes requests to IANA to allocate a TLV type from the OSPF RI TLVs registry and a sub-TLV type within the IS-IS Router Capability TLV.

## 7. Security Considerations

This document does not introduce any new security considerations.

## 8. References

## 8.1. Normative References

- [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.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4970] Lindem, A., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 4970, July 2007.
- [RFC4971] Vasseur, JP., Shen, N., and R. Aggarwal, "Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information", RFC 4971, July 2007.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012.

## 8.2. Informative References

- [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-00 (work in progress), June 2013.

## Authors' Addresses

Xiaohu Xu  
Huawei

Email: xuxiaohu@huawei.com



Sriganesh Kini  
Ericsson

Email: [sriganesh.kini@ericsson.com](mailto:sriganesh.kini@ericsson.com)

Siva Sivabalan  
Cisco

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

Clarence Filsfils  
Cisco

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

Expires: January 2014

July 11, 2013

Advertising Global Labels Using IGP

draft-xu-rtgwg-global-label-adv-00

Abstract

Segment Routing (SR) [SR-ARCH] is a new MPLS paradigm in which each SR-capable router is required to independently advertise global MPLS labels for its attached prefixes using IGP [SR-ISIS-EXT][SR-OSPF-EXT]. One major challenge associated with such label advertisement mechanism is how to avoid a given global MPLS label from being allocated by different routers to different prefixes. Although manual allocation can address such label allocation collision problem, it is error-prone and therefore may not be suitable for large SR network environments. This document proposes an alternative approach for advertising global labels without any risk of label allocation collision.

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 11, 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.

## Conventions used in this document

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].

## Table of Contents

1. Introduction .....	3
2. Terminology .....	3
3. Advertising Label Bindings for Prefixes .....	3
3.1. Extension to ISIS .....	3
3.2. Extension to OSPFv2 .....	4
3.3. Extension to OSPFv3 .....	6
4. Requesting Label Bindings for Prefixes .....	6
4.1. Extension to ISIS .....	6
4.2. Extension to OSPFv2 .....	7
4.3. Extension to OSPFv3 .....	7
5. Mapping Server Redundancy and Election .....	7
5.1. Extension to ISIS .....	8
5.2. Extension to OSPFv2 .....	8
5.3. Extension to OSPFv3 .....	8
6. Security Considerations .....	9
7. IANA Considerations .....	9
8. Acknowledgements .....	9
9. References .....	9
9.1. Normative References .....	9
9.2. Informative References .....	9
Authors' Addresses .....	10

## 1. Introduction

Segment Routing (SR) [SR-ARCH] is a new MPLS paradigm in which each SR-capable router is required to independently advertise global MPLS labels for its attached prefixes using IGP [SR-ISIS-EXT][SR-OSPF-EXT]. One major challenge associated with such label advertisement mechanism is how to avoid a given global MPLS label from being allocated by different routers to different prefixes. Although manual allocation can address such label allocation collision problem, it is error-prone therefore may not be suitable for large SR network environments.

This document proposes an alternative approach for advertising global labels without any risk of label allocation collision. The basic idea of this approach is that a single mapping server would, on behalf of all SR-capable routers within an IGP domain, allocate global labels for prefixes attached to those SR-capable routers and then advertise the label bindings in the IGP domain scope. Those prefixes which need to be allocated with global labels can be manually configured on the mapping servers or be advertised by the corresponding SR-capable routers to which those prefixes are attached. In the multi-area/level scenario where route summary between areas/levels is required, the IP longest-match algorithm SHOULD be used by SR-capable routers when processing label bindings advertised by the mapping server.

As for the scenario where the scope of label advertisement is set to area/level-scoped, it will be discussed in a future version of this document.

## 2. Terminology

This memo makes use of the terms defined in [RFC1195] [RFC2328] [SR-ARCH].

## 3. Advertising Label Bindings for Prefixes

### 3.1. Extension to ISIS

A mapping server could use one or more of the following TLVs to advertise global labels for those prefixes which need to be allocated with global labels:

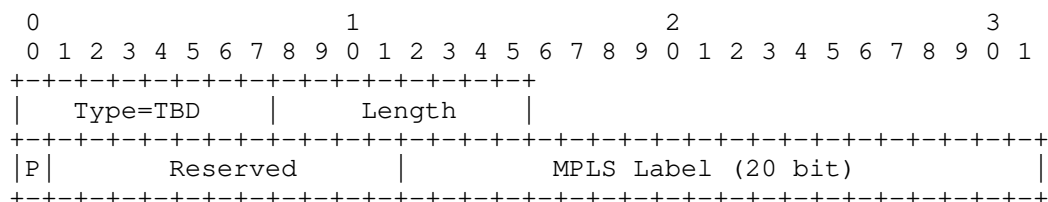
TLV-135 (IPv4) [RFC5305]

TLV-235 (MT-IPv4) [RFC5120]

TLV-236 (IPv6) [RFC5308]

TLV-237 (MT-IPv6) [RFC5120]

A Label Binding Sub-TLV (TBD) as shown below is associated with a prefix which is contained in one of the above TLVs:



Type: TBD

Length: 4

P-Flag: if set, the penultimate hop router MUST perform PHP action on the allocated MPLS label. For a given prefix, the P-Flag in the Label Binding Sub-TLV MUST be set to the same value as that of the P-Flag in the Label Request Sub-TLV if a label request message (see section 4 of this document) for that prefix is received by the mapping server.

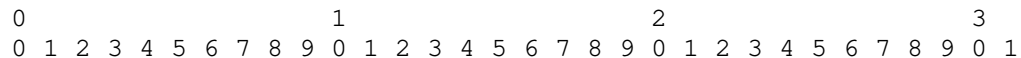
MPLS Label: a global label for the prefix which is carried in the TLV containing this sub-TLV.

Since the mapping server uses these TLVs for label binding advertisement purpose other than building the normal IP routing table, the Metric field MUST be set to a value larger than MAX\_PATH\_METRIC (i.e., 0xFE000000).

### 3.2. Extension to OSPFv2

A new Opaque LSA [RFC5250] of type 11 (with domain-wide flooding scope), referred to as Prefix Opaque LSA, is defined. The opaque type of this Prefix Opaque LSA is TBD. A mapping server could use one or more Prefix Opaque LSAs to advertise label bindings for those prefixes which need to be allocated with global labels.

One or more Prefix TLV (type code=TBD) as shown below could be contained in a Prefix Opaque LSA.



```

+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type=TBD                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      MT-ID      | Prefix-Len | Sub-TLV-Len |      Reserved      |
+-----+-----+-----+-----+-----+-----+-----+
|                                     IPv4 Prefix (0-4 octets)                                     |
+-----+-----+-----+-----+-----+-----+-----+
//                                     Sub-TLVs (Variable)                                     //
+-----+-----+-----+-----+-----+-----+-----+

```

```

?
|      MT-ID      | Prefix-Len | Sub-TLV-Len |      Reserved      |
+-----+-----+-----+-----+-----+-----+-----+
|                                     IPv4 Prefix (0-4 octets)                                     |
+-----+-----+-----+-----+-----+-----+-----+
//                                     Sub-TLVs (Variable)                                     //
+-----+-----+-----+-----+-----+-----+-----+

```

Type: TBD.

Length: Variable.

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

Prefix-Len: the length of the prefix in bits (i.e., 0-32).

Sub-TLV-Len: the length of Sub-TLVs.

IPv4 Prefix: the prefix is encoded in the minimal number of octets (i.e., 0-4) for the given number of significant bits.

A Label Binding Sub-TLV (type code=TBD) as shown below is associated with a prefix which is contained in the Prefix TLV.

```

0                                     1                                     2                                     3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type=TBD                                     |
+-----+-----+-----+-----+-----+-----+-----+
|P|      Reserved      |                                     MPLS Label (20 bit)                                     |
+-----+-----+-----+-----+-----+-----+-----+

```

Type: TBD.

Length: 4.

P-Flag: if set, the penultimate hop router MUST perform PHP action on the allocated MPLS label. For a given prefix, the P-Flag in the Label Binding Sub-TLV MUST be set to the same value as that of the P-Flag in the Label Request Sub-TLV if a label request message (see section 4 of this document) for that prefix is received by the mapping server.

MPLS Label: a global label which is allocated to the prefix which is contained in the Prefix TLV.

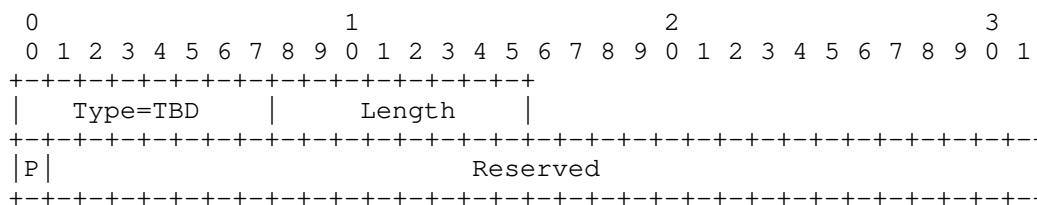
### 3.3. Extension to OSPFv3

TBD.

## 4. Requesting Label Bindings for Prefixes

### 4.1. Extension to ISIS

When advertising IP reachability information by using one of the Extended IP Reachability TLVs (i.e., TLV-135, TLV-235, TLV-236 and TLV-237), SR-capable ISIS routers SHOULD mark those among their attached prefixes which need to be allocated with a global label by associating each of these prefixes with a Label Request sub-TLV (type code=TBD) as shown below.



Type: TBD

Length: 4

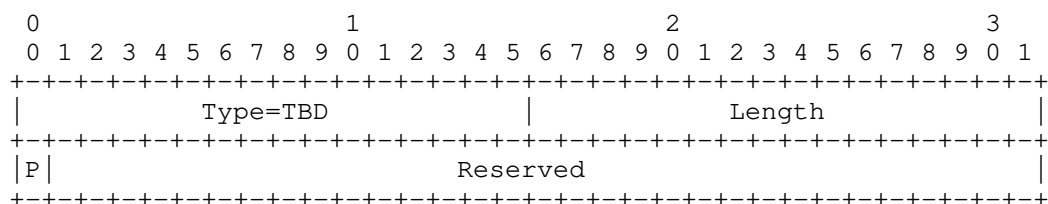
P-Flag: if set, the penultimate hop router MUST perform PHP action on the required label.

In the multi-level scenario where route summary between levels is required, separate Extended IP Reachability TLVs other than those for IP reachability advertisement purpose SHOULD be used for label binding advertisement purpose. Since these separate TLVs are not used for the purpose of building the normal IP routing table, the Metric field MUST be set to a value larger than MAX\_PATH\_METRIC (i.e., 0xFE000000).

## 4.2. Extension to OSPFv2

SR-capable OSPF routers could use one or more Prefix Opaque LSAs as defined in section 3.2 of this document to advertise those among their attached prefixes which need to be allocated with global labels.

A new Sub-TLV of the Prefix TLV, referred to as Label Request Sub-TLV (type code=TBD) as shown below is associated with a prefix which is contained in a Prefix TLV.



Type: TBD

Length: 4

P-Flag: if set, the penultimate hop router MUST perform PHP action.

## 4.3. Extension to OSPFv3

TBD.

## 5. Mapping Server Redundancy and Election

For redundancy purpose, more than one router could be configured as candidates for mapping servers. Each candidate for mapping servers SHOULD advertise its capability of being a mapping servers by using IS-IS or OSPF Router Capability TLV. The one with the highest priority SHOULD be elected as the primary mapping server which is eligible to allocate and advertise global labels for prefixes on behalf of SR-capable routers. The comparison of Router ID of ISIS or OSPF routers breaks the tie between two or more candidates with the same highest priority. Meanwhile, the one with the second highest priority SHOULD be elected as a backup mapping server. This backup mapping server is responsible for advertising the same label bindings as those advertised by the primary mapping server. In this way, it's possible to avoid unnecessary changes to the data plane (i.e., MPLS forwarding table) of SR-capable routers in the event of mapping server failover.



## 5.1. Extension to ISIS

Each candidate mapping server SHOULD advertise its capability of being a mapping server and the corresponding priority for mapping server election by attaching a Mapping Server Capability Sub-TLV (type code=TBD) shown as below to an IS-IS Router Capability TLV [RFC4971] with the S flag set (with domain-wide flooding scope).

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Type   |         Length        |      Priority      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type: TBD

Length: 1

Priority: the priority for mapping server election.

## 5.2. Extension to OSPFv2

Each candidate mapping server SHOULD advertise its capability of being mapping servers by using an OSPF Router Information Capabilities TLV [RFC4970] contained in an Opaque LSA of type 11 (with domain-wide flooding scope). One of the unreserved OSPF Router Information Capabilities Bits is reserved for this purpose. Furthermore, a sub-TLV (type code=TBD) as shown below is used to convey the priority value for mapping server election.

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Type   |         Length        |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Priority   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Type: TBD

Length: 1

Priority: the priority for mapping server election.

## 5.3. Extension to OSPFv3

TBD.

## 6. Security Considerations

TBD.

## 7. IANA Considerations

TBD.

## 8. Acknowledgements

Thanks to.

## 9. References

### 9.1. Normative References

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

### 9.2. Informative References

[SR-ARCH] 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.

[SR-ISIS-EXT] Previdi, S., Filsfils, C., and A. Bashandy, "IS-IS Segment Routing Extensions", May 2013.

[SR-OSPF-EXT] Psenak, P. and S. Previdi, "OSPF Segment Routing Extensions", May 2013.

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, October 2008.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

[RFC4971] Vasseur, J-P., Ed., Shen, N., Ed., and R. Aggarwal, Ed., "Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information", RFC 4971, July 2007.

- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, February 2008.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.
- [RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, July 2008.
- [RFC4970] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 4970, July 2007.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, June 2007.

#### Authors' Addresses

Xiaohu Xu  
Huawei Technologies,  
Beijing, China  
Phone: +86-10-60610041  
Email: xuxiaohu@huawei.com

Mach(Guoyi) Chen  
Huawei Technologies,  
Beijing, China  
Phone: +86-10-60610041  
Email: mach.chen@huawei.com