

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.

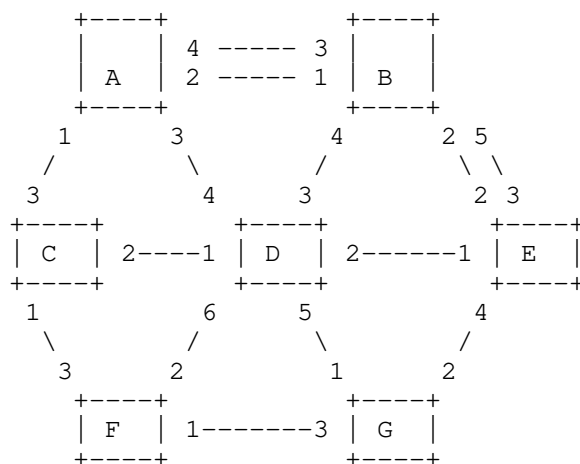


Figure 1 - simple 7 node network with local link identifiers.

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 SLSR 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 SLSR 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 SLSP to the replication point and then SLSPs 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 SLSP multicast etc. are FFS.

9. Failure Recovery

A variety of failure recovery techniques can be employed with SLSP. 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 SLSP 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
- [ONF] 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
- [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: December 30, 2013

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.
June 28, 2013

Segment Routing Architecture
draft-filsfils-rtgwg-segment-routing-00

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. 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. 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 December 30, 2013.

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	13
3.1.1. Example 1	14
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	17
3.2.5. IGP-Adjacency Segment, Adj-SID	18
3.2.6. Finally	18
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	22
4. Service Segments	23
5. MPLS	23
6. IPv6	24
7. OAM	24
8. Multicast	24
9. IANA Considerations	24
10. Manageability Considerations	25
11. Security Considerations	25
12. Acknowledgements	25
13. References	25
13.1. Normative References	25
13.2. 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. Section 5 and Section 6 define the instantiations of SR in MPLS and IPv6.

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.

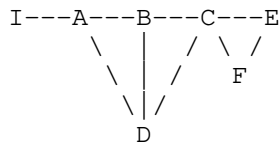


Figure 1: An illustration of SR properties

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

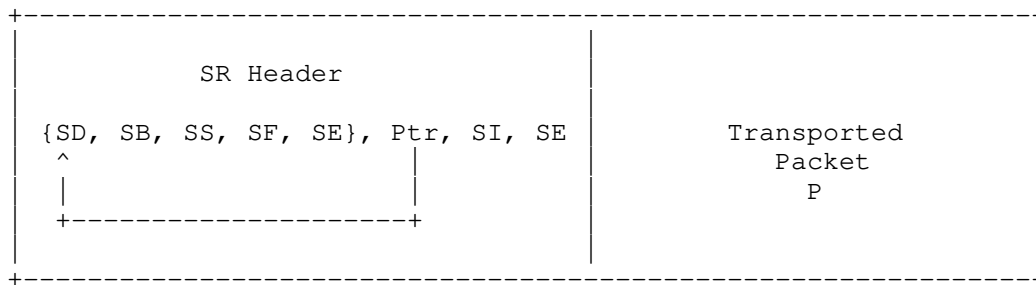


Figure 2: Packet P at node I

The 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}.

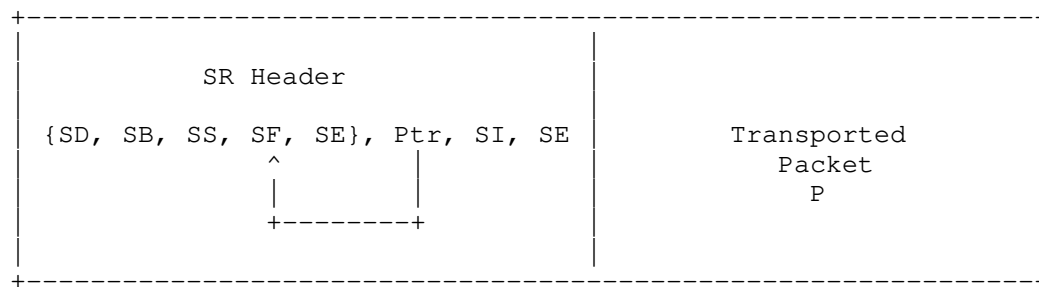


Figure 3: Packet P at node C

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
[draft-filsfils-rtgwg-segment-routing-use-cases-00].

IS-IS protocol extensions for Segment Routing are described in
[draft-previdi-isis-segment-routing-extensions-00].

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

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

In the future, it is expected that Section 5 and Section 6 of this document be removed and submitted as independent documents, respectively as instantiations of SR in MPLS and IPv6.

In the future, we will submit a SR-FRR specific document.

1.5. Relationship with MPLS and IPv6

The source routing model is inherited from the one proposed by
[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 MPLS model are:

Global segments are introduced (e.g. IGP node segments).

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

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

SR can be instantiated on the IPv6 dataplane. Section 6 details 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. In Section 5, we explain that the information present in the SR abstract header is encoded as a stack of labels. The notion of pointer and full segment list containing the full history of the path from ingress to egress edge routers is thus lost in the MPLS instantiation of SR. However all the other SR properties are preserved and especially the MPLS dataplane can be reused without any change.

2. Abstract Routing Model

Segment Routing (SR) leverages the source routing and tunneling paradigms.

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.

[draft-filsfils-rtgwg-segment-routing-use-cases-00] 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.

[draft-filsfils-rtgwg-segment-routing-use-cases-00] 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

[draft-msiva-pce-pcep-segment-routing-extensions-00].

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 [draft-filsfils-rtgwg-segment-routing-use-cases-00] 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.

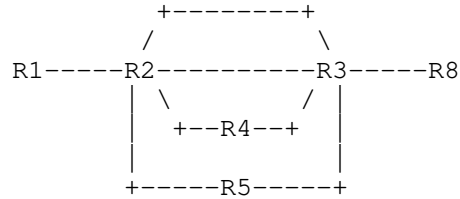


Figure 5: IGP Segments - Illustration

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.

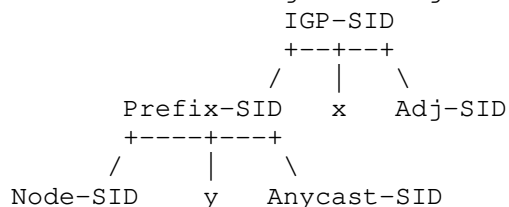


Figure 7: IGP SID Terminology

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 [draft-filsfils-rtgwg-segment-routing-use-cases-00] 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 [draft-filsfils-rtgwg-segment-routing-use-cases-00].

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 [draft-filsfils-rtgwg-segment-routing-use-cases-00].

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 [draft-filsfils-rtgwg-segment-routing-use-cases-00].

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 [draft-previdi-isis-segment-routing-extensions-00] and [draft-psenak-ospf-segment-routing-extensions-00].

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 [draft-previdi-isis-segment-routing-extensions-00] and [draft-psenak-ospf-segment-routing-extensions-00].

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.

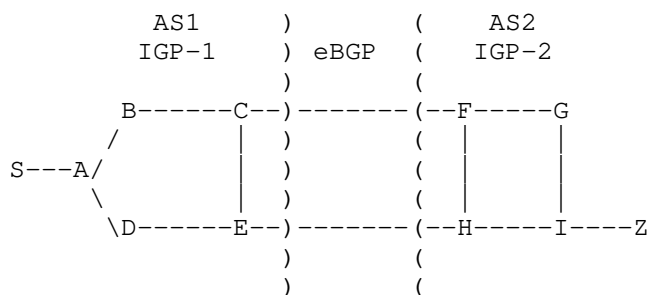


Figure 9: External Adjacency Example

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)}.

[draft-filsfils-rtgwg-segment-routing-use-cases-00] 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.

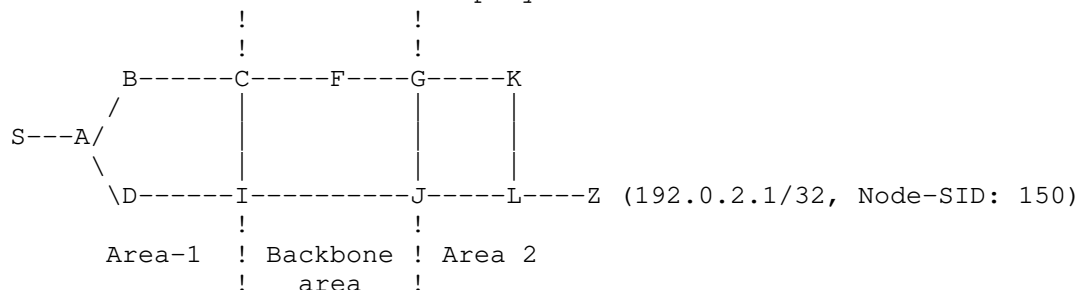


Figure 10: Inter-Area Topology Example

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.

[draft-filsfils-rtgwg-segment-routing-use-cases-00] 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 [draft-previdi-isis-segment-routing-extensions-00] and [draft-psenak-ospf-segment-routing-extensions-00].

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. MPLS

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.

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 where the outgoing label value is equal to the incoming label value.

The NEXT operation is implemented as an MPLS pop operation.

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

The SRGB label space is allocated to Segment Routing. The SR operator manages the SRGB space as a registry and ensures the unique allocation of the global resource.

A local segment is a locally allocated label.

In conclusion:

There are no changes in the operations of the data-plane currently used in MPLS networks.

The SR solution can co-exist and interwork with other MPLS control-plane protocols, see [draft-filsfils-rtgwg-segment-routing-use-cases-00] for more details.

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

6. IPv6

The text will be added in future revision.

7. 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 [draft-filsfils-rtgwg-segment-routing-use-cases-00].

8. Multicast

The text will be added in future revision.

9. IANA Considerations

TBD

10. Manageability Considerations

TBD

11. Security Considerations

TBD

12. 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.

13. References

13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [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.

13.2. Informative References

- [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.
- [draft-filsfils-rtgwg-segment-routing-use-cases-00]
Filsfils, C., "Segment Routing Use Cases", May 2013.

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

[draft-previdi-isis-segment-routing-extensions-00]
Previdi, S., Filsfils, C., and A. Bashandy, "IS-IS Segment
Routing Extensions", May 2013.

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

IS-IS for IP Internets
Internet-Draft
Intended status: Standards Track
Expires: November 22, 2013

H. Gredler, Ed.
Juniper Networks, Inc.
S. Amante
Level 3 Communications, Inc.
T. Scholl
Amazon
L. Jalil
Verizon
May 21, 2013

Advertising MPLS labels in IS-IS
draft-gredler-isis-label-advertisement-03

Abstract

Historically MPLS label distribution was driven by protocols like LDP, RSVP and LBGp. All of those protocols are session oriented. In order to obtain a label binding for a given destination FEC from a given router one needs first to establish an LDP/RSVP/LBGp session with that router.

Advertising MPLS labels in IGPs
[I-D.gredler-rtgwg-igp-label-advertisement] describes several use cases where utilizing the flooding machinery of link-state protocols for MPLS label distribution allows to obtain the binding without requiring to establish an LDP/RSVP/LBGp session with that router.

This document describes the protocol extension to distribute MPLS label bindings using the IS-IS protocol.

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 November 22, 2013.

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
2. Motivation, Rationale and Applicability	4
2.1. Issue: Bi-directionality semantics	5
2.2. Issue: IP path semantics	5
2.3. Issue: Lack of 'path' notion	5
2.4. Motivation	6
3. MPLS label TLV	6
3.1. Flags	7
3.2. subTLV support	7
3.3. IPv4 Prefix ERO subTLV	7
3.4. IPv6 Prefix ERO subTLV	8
3.5. Unnumbered Interface ID ERO subTLV	9
3.6. IPv4 Prefix Bypass ERO subTLV	10
3.7. IPv6 Prefix Bypass ERO subTLV	10
3.8. Unnumbered Interface ID Bypass ERO subTLV	11
3.9. Prefix ERO and Prefix Bypass ERO subTLV path semantics	12
3.10. All Router Block subTLV	12
3.11. All Router ID IPv4 Map subTLV	14
3.12. All Router ID IPv6 Map subTLV	15
4. Advertising Label Examples	15
4.1. Sample Topology	15
4.1.1. Transport IP addresses and Router-IDs	16
4.1.2. Link IP addresses	16
4.2. One-hop LSP to an adjacent Router	17
4.3. One-hop LSP to an adjacent Router using a specific link	17
4.4. Advertisement of Fast Re-Route LSP for One-Hop LSP	17
4.5. Advertisement of an RSVP LSP	18
4.6. Advertisement of an LDP LSP	18
4.7. Interarea advertisement of diverse paths	18
4.8. Advertisement of SPT labels using 'All Router Block' TLV	19
4.9. Expansion of an 'All Router Block' subTLV	20
5. Inter Area Protocol Procedures	21
5.1. Applicability	21
5.2. Data plane operations	21
5.3. Control plane operations	21
5.3.1. MPLS Label operations	21
5.3.2. MPLS Label Block operations	22
6. Acknowledgements	22
7. IANA Considerations	22
8. Security Considerations	23
9. References	23
9.1. Normative References	23
9.2. Informative References	24
Authors' Addresses	24

1. Introduction

MPLS label allocations are predominantly distributed by using the LDP [RFC5036], RSVP [RFC5151] or labeled BGP [RFC3107] protocol. All of those protocols have in common that they are session oriented, which means that in order to obtain label binding for a given destination FEC from a given router one needs first to establish a direct control plane (LDP/RSVP/LBGP) session with that router.

There are a couple of practical use cases [I-D.gredler-rtgwg-igp-label-advertisement] where the consumer of a MPLS label binding may not be adjacent to the router that performs the binding. Bringing up an explicit session using the existing label distribution protocols between the non-adjacent router that binds the label and the router that acts as a consumer of this binding is the existing remedy for this dilemma.

This document describes an IS-IS protocol extension which allows routers to advertise MPLS label bindings within and beyond an IGP domain, and controlling inter-area distribution.

2. Motivation, Rationale and Applicability

One possible way of distributing MPLS labels using IS-IS has been described in Segment Routing [I-D.previdi-filsfils-isis-segment-routing]. The authors propose to re-use the IS-Reach TLVs (22, 23, 222) and Extended IP Prefix TLVs (135, 236) for carrying the label information. While retrofitting existing protocol machinery for new purposes is generally a good thing, Segment Routing [I-D.previdi-filsfils-isis-segment-routing] falls short of addressing some use-cases defined in [I-D.gredler-rtgwg-igp-label-advertisement].

The dominant issue around re-using IS-Reach TLVs and the extended IP Prefix TLVs is that both family of TLVs have existing protocol semantics, which might not be well suitable to advertising MPLS label switched paths in a generic fashion. These are specifically:

- o Bi-directionality semantics
- o IP path semantics
- o Lack of 'path' notion

2.1. Issue: Bi-directionality semantics

'Bi-directionality semantics', affects the complexity around advertisement of unidirectional LSPs. Label advertisement of per-link labels or 'Adj-SIDs' [I-D.previdi-filsfils-isis-segment-routing] is done using IS-reach TLVs. Usually implementations need to have an adjacency in 'Up' state prior to advertising this adjacency as IS-reach TLV in its Link State PDUs (LSPs). In order to advertise e.g. one-hop MPLS LSP in a given link an implementation first needs to have an adjacency, which only transitions to 'Up' state after passing the 3-way check. This implies bi-directionality. If an implementation wants to advertise per-link LSPs to e.g. outside the IGP domain then it would need to fake-up an adjacency. Changing existing IGP Adjacency code to support such cases defeats the purpose of re-using existing functionality as there is not much common functionality to be shared.

2.2. Issue: IP path semantics

LSPs pointing to a Node are advertised as 'Node-SIDs' [I-D.previdi-filsfils-isis-segment-routing] using the family of extended IP Reach TLVs. That means that in order to advertise a MPLS LSP, one is inheriting the semantics of advertising an IP path. Consider router A has got existing MPLS LSPs to its entire one-hop neighborhood and is re-advertising those MPLS LSPs using IP reachability semantics. Now we have two exact matching IP advertisements. One from the owning router (router B) which advertises its stable transport loopback address and another one from router A re-advertising a MPLS LSP path to router B. Existing routing software may get confused now as the 'stable transport' address shows up from multiple places in the network and more worse the IP forwarding path for control-plane protocols may get mingled with the MPLS data plane.

2.3. Issue: Lack of 'path' notion

Both IS-Reach TLVs and IP Prefix Reachability TLVs have a limited semantics describing MPLS label-switched paths in the sense of a 'path'. Both encoding formats allow to specify a pointer to some specific router, but not to describe a MPLS label switched path containing all of its path segments. [I-D.previdi-filsfils-isis-segment-routing] allows to define 'Forwarding Adjacencies' as per [RFC4206]. The way to describe a path of a given forwarding adjacency is to carry a list of "Segment IDs". That implies that nodes which do not yet participate in 'Segment routing' or are outside of a 'Segment routing' domain can not be expressed using those path semantics.

A protocol for advertising MPLS label switched paths, should be generic enough to express paths sourced by existing MPLS LSPs, such that ingress routers can flexibly combine them according to application needs.

2.4. Motivation

IGP advertisement of MPLS label switched paths requires a new set of protocol semantics (path paradigm), which hardly can be expressed using the existing IS-IS protocol. This document describes IS-IS protocol extensions which allows generic advertisement of MPLS label bindings in IS-IS.

The Protocol extensions described in this document are equally applicable to IPv4 and IPv6 carried over MPLS. Furthermore the proposed use of distributing MPLS Labels using IGP protocols adheres to the architectural principles laid out in [RFC3031].

3. MPLS label TLV

The MPLS Label TLV may be originated by any Traffic Engineering [RFC5305] capable router in an IS-IS domain. The router may advertise a single label binding or a block of label bindings. For single label binding advertisement a router needs to provide at least a single 'nexthop style' anchor. The protocol supports more than one 'nexthop style' anchor to be attached to a Label binding, which results into a simple path description language. In analogy to RSVP the terminology for this is called an 'Explicit Route Object' (ERO). Since ERO style path notation allows to anchor label bindings to both link and node IP addresses any label switched path, can be described. Furthermore also Label Bindings from other protocols can get easily re-advertised.

Due to the limited size of subTLV space (See [RFC5311] section 4.5 for details), The MPLS Label TLV has cumulative rather than canceling semantics. If a router originates more than one MPLS Label TLV with the same Label value, then the subTLVs of the second, third, etc. TLV are accumulated. Since some subTLVs represent an ordered set (e.g. ERO subTLVs) allocation and ordering of TLV space inside particular IS-IS LSP fragment is significant and needs to be tracked.

The MPLS Label TLV has type 149 and has the following format:

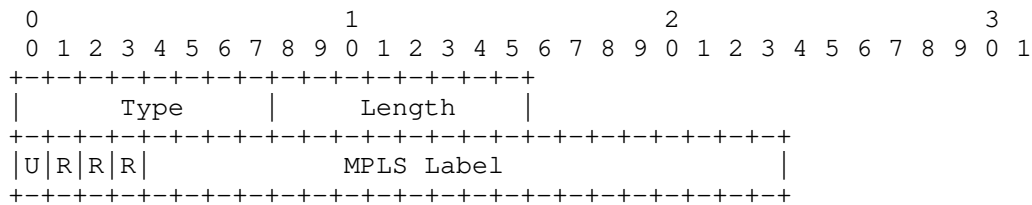


Figure 1: MPLS TLV format

- o 4 bits of flags, consisting of:
 - * 1 bit of up/down information (U bit)
 - * 3 bits are reserved for future use
- o 20 bits of MPLS label information
- o 0-252 octets of sub-TLVs, where each sub-TLV consists of a sequence of:
 - * 1 octet of sub-TLV type
 - * 1 octet of length of the value field of the sub-TLV
 - * 0-250 octets of value

3.1. Flags

Flags

Up/Down Bit: A router may flood MPLS label information across level boundaries. In order to prevent flooding loops, a router will Set the Up/Down (U-Bit) when propagating from Level 2 down to Level 1. This is done as per the procedures for IP Prefixes lined out in [RFC5302].

3.2. subTLV support

An originating router MAY want to attach one or more subTLVs to the MPLS label TLV. SubTLVs presence is inferred from the length of the MPLS Label TLV. If the MPLS Label TLV Length field is > 3 octets then one or more subTLVs may be present.

3.3. IPv4 Prefix ERO subTLV

The IPv4 ERO subTLV (Type 1) describes a path segment using IPv4 Prefix style of encoding. Its appearance and semantics have been

borrowed from Section 4.3.3.2 [RFC3209].

The 'Prefix Length' field contains the length of the prefix in bits. Only the most significant octets of the prefix are encoded. I.e. 1 octet for prefix length 1 up to 8, 2 octets for prefix length 9 to 16, 3 octets for prefix length 17 up to 24 and 4 octets for prefix length 25 up to 32, etc.

The 'L' bit in the subTLV is a one-bit attribute. If the L bit is set, then the value of the attribute is 'loose.' Otherwise, the value of the attribute is 'strict.'

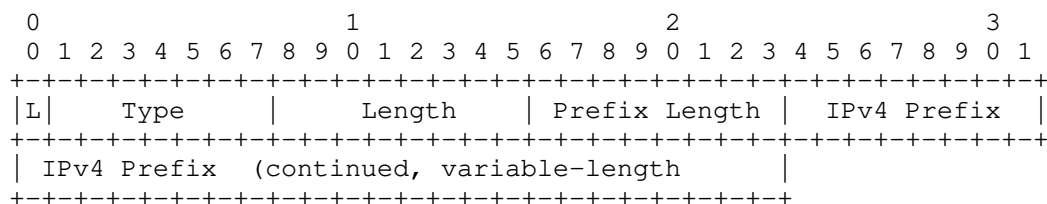


Figure 2: IPv4 Prefix ERO subTLV format

3.4. IPv6 Prefix ERO subTLV

The IPv6 ERO subTLV (Type 2) describes a path segment using IPv6 Prefix style of encoding. Its appearance and semantics have been borrowed from Section 4.3.3.3 [RFC3209].

The 'Prefix Length' field contains the length of the prefix in bits. Only the most significant octets of the prefix are encoded. I.e. 1 octet for prefix length 1 up to 8, 2 octets for prefix length 9 to 16, 3 octets for prefix length 17 up to 24 and 4 octets for prefix length 25 up to 32,, 16 octets for prefix length 113 up to 128.

The 'L' bit in the subTLV is a one-bit attribute. If the L bit is set, then the value of the attribute is 'loose.' Otherwise, the value of the attribute is 'strict.'

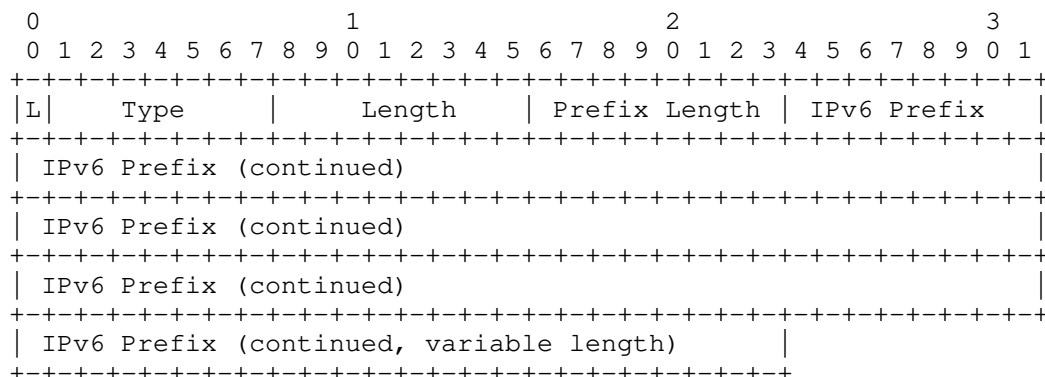


Figure 3: IPv6 Prefix ERO subTLV format

3.5. Unnumbered Interface ID ERO subTLV

The appearance and semantics of the 'Unnumbered Interface ID' have been borrowed from Section 4 [RFC3477].

The Unnumbered Interface-ID ERO subTLV (Type 9) describes a path segment that spans over an unnumbered interface. Unnumbered interfaces are referenced using the interface index. Interface indices are assigned local to the router and therefore not unique within a domain. All elements in an ERO path need to be unique within a domain and hence need to be disambiguated using a domain unique Router-ID.

The 'Router-ID' field contains the router ID of the router which has assigned the 'Interface ID' field. Its purpose is to disambiguate the 'Interface ID' field from other routers in the domain.

IS-IS supports two Router-ID formats:

- o (TLV 134, 32-Bit format) [RFC5305]
- o (TLV 140, 128-Bit format) [RFC6119]

The actual Router-ID format gets derived from the 'Length' field.

- o For 32-Bit Router-ID width the subTLV length is set to 8 octets.
- o For 128-Bit Router-ID width the subTLV length is set to 20 octets.

The 'Interface ID' is the identifier assigned to the link by the router specified by the router ID.

The 'L' bit in the subTLV is a one-bit attribute. If the L bit is set, then the value of the attribute is 'loose.' Otherwise, the value of the attribute is 'strict.'

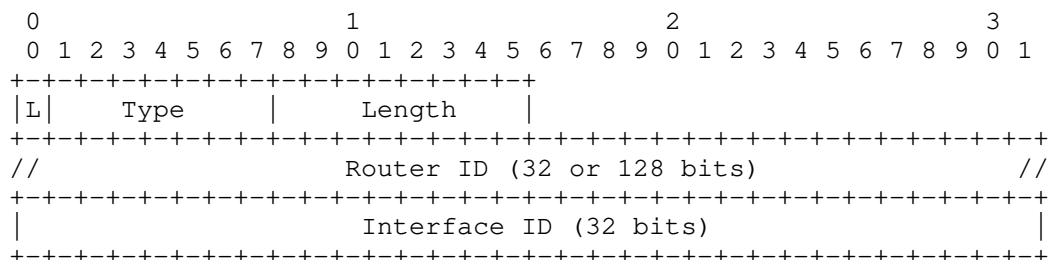


Figure 4: Unnumbered Interface ID ERO subTLV format

3.6. IPv4 Prefix Bypass ERO subTLV

The IPv4 Bypass ERO subTLV (Type 3) describes a Bypass LSP path segment using IPv4 Prefix style of encoding. Its appearance and semantics have been borrowed from Section 4.3.3.2 [RFC3209].

The 'Prefix Length' field contains the length of the prefix in bits. Only the most significant octets of the prefix are encoded, i.e. 1 octet for prefix length 1 up to 8, 2 octets for prefix length 9 to 16, 3 octets for prefix length 17 up to 24 and 4 octets for prefix length 25 up to 32, etc.

The 'L' bit in the subTLV is a one-bit attribute. If the L bit is set, then the value of the attribute is 'loose.' Otherwise, the value of the attribute is 'strict.'

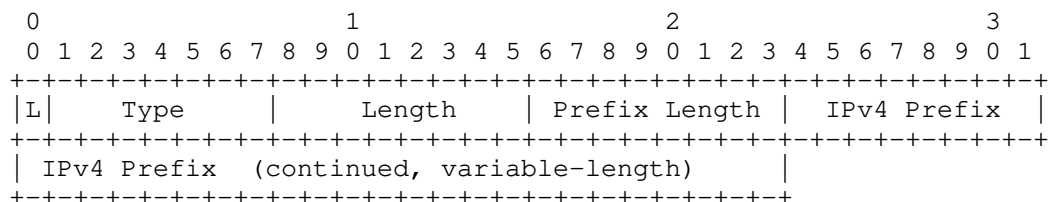


Figure 5: IPv4 Prefix Bypass ERO subTLV format

3.7. IPv6 Prefix Bypass ERO subTLV

The IPv6 ERO subTLV (Type 4) describes a Bypass LSP path segment using IPv6 Prefix style of encoding. Its appearance and semantics have been borrowed from Section 4.3.3.3 [RFC3209].

- o (TLV 134, 32-Bit format) [RFC5305]
- o (TLV 140, 128-Bit format) [RFC6119]

The actual Router-ID format gets derived from the 'Length' field.

- o For 32-Bit Router-ID width the subTLV length is set to 8 octets.
- o For 128-Bit Router-ID width the subTLV length is set to 20 octets.

The 'Interface ID' is the identifier assigned to the link by the router specified by the router ID.

The 'L' bit in the subTLV is a one-bit attribute. If the L bit is set, then the value of the attribute is 'loose.' Otherwise, the value of the attribute is 'strict.'

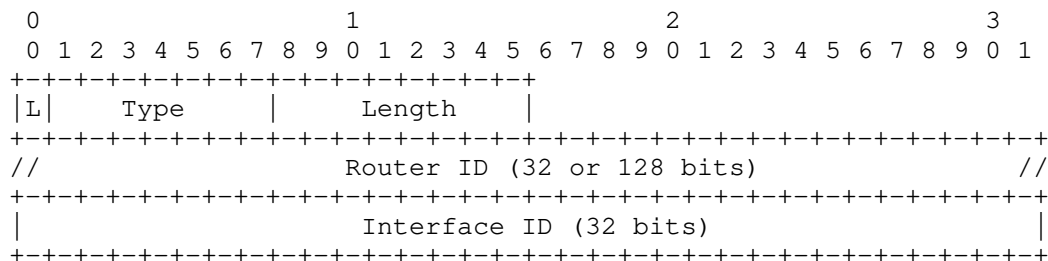


Figure 7: Unnumbered Interface ID Bypass ERO subTLV format

3.9. Prefix ERO and Prefix Bypass ERO subTLV path semantics

All 'Prefix ERO' and 'Prefix Bypass ERO' information represents an ordered set which describes the segments of a label-switched path. The last Prefix ERO subTLV describes the segment closest to the egress point of the LSP. Contrary the first Prefix ERO subTLV describes the first segment of a label switched path. If a router extends or stitches a label switched path it MUST prepend the new segments path information to the Prefix ERO list. The same ordering applies for the Bypass ERO labels. An implementation SHOULD first encode all primary path EROs followed by the bypass EROs.

3.10. All Router Block subTLV

The 'All Router Block' subTLV (Type 6) denominates the label block size of an MPLS Label advertisement and its semantics to connect to all routers in a given IS-IS domain using a local assigned [RFC3031] label range. Note that the actual mapping of a router within the label range is done using the subTLVs described in Section 3.11 and

Section 3.12. Since generation of an 'All Router ID IPv4 Map' or 'All Router ID IPv6 Map' subTLV is a local policy decision, it might be the case that connectivity is provided not to 'All' but rather a subset of 'All' routers. Keeping policy decisions aside, for simplicity reasons, assume that All Routers in a domain do generate either the 'All Router ID IPv4 Map' or 'All Router ID IPv6 Map' subTLVs and therefore all routers desire construction of a Label switched path from every source router in the network. The basic concept of using label blocks to provide connectivity to a set of routers has been borrowed from [RFC4761] which allows to advertise labels from multiple end-points using a single control-plane message. The difference to [RFC4761] is that rather than advertising where a particular packet came from (=source semantics), destination semantics (where a particular packet will be going to) is advertised.

Along with each label block a router advertises one for more 'IDs'. The 'ID' must be unique within a given domain. The 'ID' serves as ordinal to determine the actual label value inside the set of all advertised label ranges of a given router. A receiving router uses the ordinal to determine the actual label value in order to construct forwarding state to a particular destination router. The 'ID' is separately advertised using the subTLVs described in Section 3.11 and Section 3.12.

The ability to advertise more than one label block eases operational procedures for increasing the number of supported routers within a domain. For example consider a given domain has got support for <M> routers and runs out of ID space. It simply advertises one more label block to cover additional ordinals outside the range of the first label block. An example of label-block expansion is described in more detail in Section 4.9

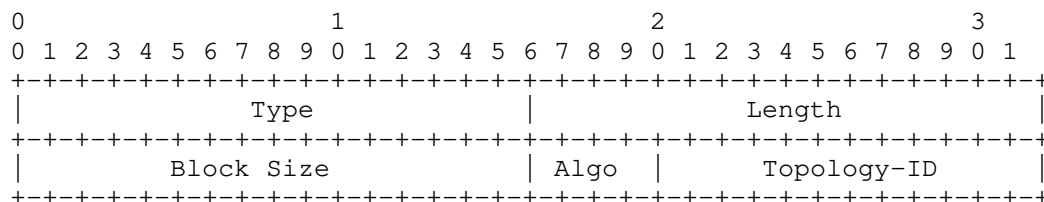


Figure 8: All Router Block subTLV format

The 'Block Size' value contains the size of the label advertisement. The 'value determines the amount of reachable router endpoints within a given Label block. It MUST contain a value greater or equal than two. Note that the label base is inferred from the Label Value in the carrying MPLS Label TLV. For example if a router wants to advertise a label range of 5000-5099 then it would need to generate a

MPLS Label TLV with a Label value of 5000 and a Block Size of 100.

The 'Algo' value denominates the path computation algorithm in order to calculate the forwarding topology. The basic SPF algorithm has an assigned 'Algo' code point of zero. The purpose of the 'Algo' field is to extend the notion of Label Block Signaling to arbitrary algorithms like for example 'MRT' ([I-D.ietf-rtgwg-mrt-frr-architecture]). Advertised Label Blocks with an unknown, unsupported or non-configured algorithm MUST be silently ignored.

The 'Reserved' bits are for future use. They should be zero on transmission and ignored on receipt.

The 'Topology-ID' field contains the Multi Topology ID ([RFC5120]) for which the advertised Label Block does apply. The basic IPv4 unicast Topology has an assigned 'Topology-ID' code point of zero. The basic IPv6 unicast Topology has an assigned 'Topology-ID' code point of 2. Advertised Label Blocks with an unknown, unsupported or non-configured Topology-ID MUST be silently ignored.

A MPLS Label TLV containing the 'All Router Block' subTLV MUST only contain the 'All Router IPv4 Map' subTLV (Section 3.11) or the 'All Router IPv6 Map' subTLV (Section 3.12).

3.11. All Router ID IPv4 Map subTLV

The 'All Router ID IPv4 Map' TLV (Type 7) maps an 'ID' to a given stable transport IPv4 address. Its purpose is to associate a given transport IPv4 IP address to the ordinal inside a label range as described in Section 3.10.

A router MAY advertise more than one 'ID' to 'IPv4 address' mapping pair, in case it has more than one stable transport IPv4 address.

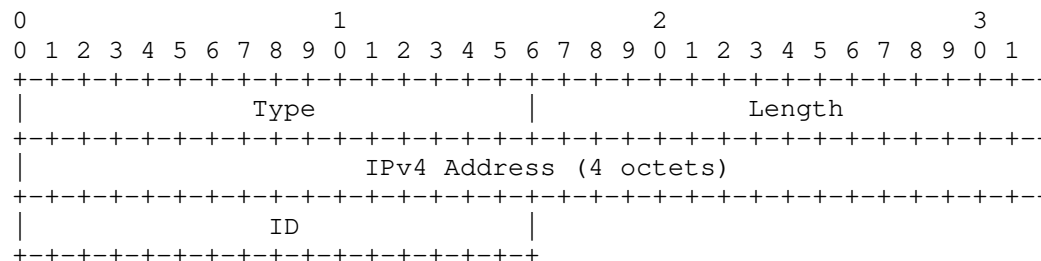


Figure 9: All Router ID IPv4 Map subTLV format

The 'IPv4 address' contains stable IPv4 transport address of a given

router.

The 'ID' contains the ordinal value of an advertising router inside the set of all advertised label blocks of a given router.

3.12. All Router ID IPv6 Map subTLV

The 'All Router ID IPv6 Map' TLV (Type 8) maps an 'ID' to a given stable transport IPv6 address. Its purpose is to associate a given transport IPv6 IP address to the ordinal inside a label range as described in Section 3.10.

A router MAY advertise more than one 'ID' to 'IPv6 address' mapping pair, in case it has more than one stable transport IPv6 address.

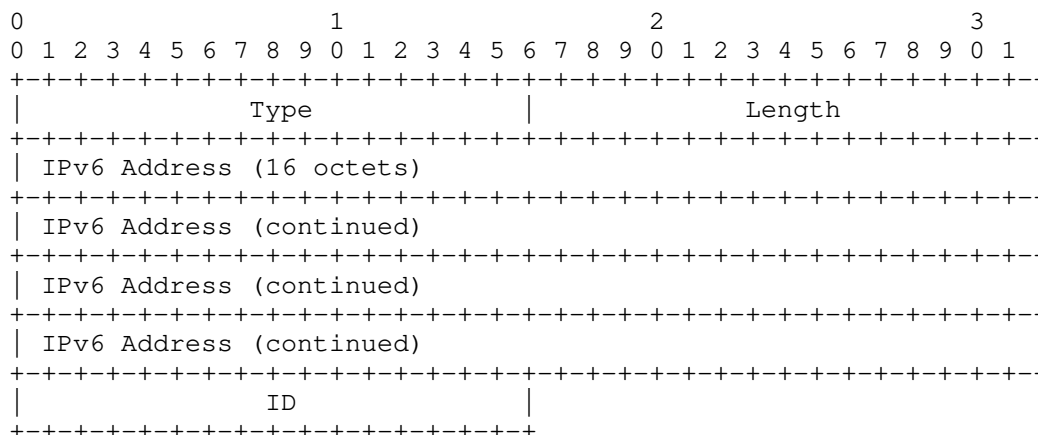


Figure 10: All Router ID IPv6 Map subTLV format

The 'IPv6 address' contains the stable IPv6 transport address of a given router.

The 'ID' contains the ordinal value of an advertising router inside the set of all advertised label blocks of a given router.

4. Advertising Label Examples

4.1. Sample Topology

The following topology (Figure 11) and IP addresses shall be used throughout the Label advertisement examples.

- o R2 to R3 link #1: 10.0.0.3, 10.0.0.4
- o R2 to R3 link #2: 10.0.0.5, 10.0.0.6
- o R2 to R5 link: 10.0.0.7, 10.0.0.8
- o R3 to R6 link: 10.0.0.13, 10.0.0.14
- o R3 to R7 link: 10.0.0.15, 10.0.0.16
- o R4 to R5 link: 10.0.0.19, 10.0.0.20
- o R5 to R6 link: 10.0.0.11, 10.0.0.12
- o R6 to R7 link: 10.0.0.17, 10.0.0.18

The IGP link metrics are displayed in the middle of the link. All of them are assumed to be bi-directional.

4.2. One-hop LSP to an adjacent Router

If R1 would advertise a label <N> bound to a one-hop LSP from R1 to R2 it would encode as follows:

TLV 149: MPLS label <N>, Flags {}:

IPv4 Prefix ERO subTLV: 192.168.1.2/32, Strict

4.3. One-hop LSP to an adjacent Router using a specific link

If R2 would advertise a label <N> bound to a one-hop LSP from R2 to R3, using the link #2 it would encode as follows

TLV 149: MPLS label <N>, Flags {}:

IPv4 Prefix ERO subTLV: 10.0.0.6/32, Strict

4.4. Advertisement of Fast Re-Route LSP for One-Hop LSP

R2 may advertise a one-hop LSP from R2 to R3, along with a Link Protection Bypass for the directly adjacent links between those two nodes. The Link Protection Bypass would use the path: {R2, R5, R6, R3}. R2 would encode both the primary LSP and Link Protection Bypass LSP as follows:

TLV 149: MPLS label <N>, Flags {}:

IPv4 Prefix ERO subTLV: 192.168.1.3/32, Strict

IPv4 Prefix Bypass ERO subTLV: 192.168.1.5/32, Strict

IPv4 Prefix Bypass ERO subTLV: 192.168.1.6/32, Strict

IPv4 Prefix Bypass ERO subTLV: 192.168.1.3/32, Strict

4.5. Advertisement of an RSVP LSP

Consider a RSVP LSP name "R2-to-R6" traversing (R2 to R3 using link #1, R6):

If R2 would advertise a label <N> bound to the RSVP LSP named 'R2-to-R6', it would encode as follows

TLV 149: MPLS label <N>, Flags {}:

IPv4 Prefix ERO subTLV: 10.0.0.4/32, Strict

IPv4 Prefix ERO subTLV: 192.168.1.6/32, Strict

4.6. Advertisement of an LDP LSP

Consider R2 that creates a LDP label binding for FEC 172.16.0.0/12 using label <N>.

If R2 would re-advertise this binding in IS-IS it would encode as follows

TLV 149: MPLS label <N>, Flags {}:

IPv4 Prefix ERO subTLV: 172.16.0.0/12, Loose

4.7. Interarea advertisement of diverse paths

Consider two R2->R6 paths: {R2, R3, R6} and {R2, R5, R6}

Consider two R5->R3 paths: {R5, R2, R3} and {R5, R6, R3}

R2 encodes its two paths to R6 as follows:

TLV 149: MPLS label <N1>, Flags {}:

IPv4 Prefix ERO subTLV: 192.168.1.3, Strict

IPv4 Prefix ERO subTLV: 192.168.1.6, Strict

TLV 149: MPLS label <N2>, Flags {}:

IPv4 Prefix ERO subTLV: 192.168.1.5, Strict

IPv4 Prefix ERO subTLV: 192.168.1.6, Strict

R5 encodes its two paths to R3 as follows:

TLV 149: MPLS label <N1>, Flags {}:

IPv4 Prefix ERO subTLV: 192.168.1.2, Strict

IPv4 Prefix ERO subTLV: 192.168.1.3, Strict

TLV 149: MPLS label <N2>, Flags {}:

IPv4 Prefix ERO subTLV: 192.168.1.6, Strict

IPv4 Prefix ERO subTLV: 192.168.1.3, Strict

A receiving L1 router does see now all 4 paths and may decide to load-balance across all or a subset of them.

4.8. Advertisement of SPT labels using 'All Router Block' TLV

All routers within a given area MUST advertise their Label Blocks along with an 'ID'.

If R2 would advertise a label block <N1> with a size of 10, declaring SPT label forwarding support to all routers within a given domain, it would encode as follows:

TLV 149: MPLS Label <N1>, Flags {}:

All Router Block subTLV: Block Size 10, Algo 0, Topology 0

All Router ID IPv4 Map subTLV: ID 2, 192.168.1.2

If R3 would advertise a label block <N2> with a size of 10, declaring SPT label forwarding support to all routers within a given domain, it would encode as follows:

TLV 149, MPLS Label <N2>, Flags {}:

All Router Block subTLV: Block Size 10, Algo 0, Topology 0

All Router ID IPv4 Map subTLV: ID 3, 192.168.1.3

If R5 would advertise a label block <N3> with a size of 10, declaring SPT label forwarding support to all routers within a given domain, it would encode as follows:

TLV 149, MPLS Label <N3>, Flags {}:

All Router Block subTLV: Block Size 10, Algo 0, Topology 0

All Router ID IPv4 Map subTLV: ID 5, 192.168.1.5

If R6 would advertise a label block <N4> with a size of 10, declaring SPT label forwarding support to all routers within a given domain, it would encode as follows:

TLV 149, MPLS Label <N4>, Flags {}:

All Router Block subTLV: Block Size 10, Algo 0, Topology 0

All Router ID IPv4 Map subTLV: ID 6, 192.168.1.6

Consider now R2 constructing a SPT label for R6. R2s SPT to R6 is {R2, IP4, R3, R6}. R2 first determines if its downstream router (R3) has advertised a label-block. Since R3 has advertised a label block 'N2' and it has received R6 'ID' of 6 it will be picking the 6th label value inside the advertised range of its downstream neighbor. Specifically R2 MUST be program a MPLS SWAP for its own label range Label(N1+6) to Label(N2+6), NH 10.0.0.4 into its MPLS transit RIB. Furthermore R2 MAY program a MPLS PUSH operation for IP 192.168.1.6 to Label (N2+6), NH 10.0.0.4 into its IPv4 tunnel RIB.

Next walk down to R3, which is the next router on the SPT tree towards R6. R3s SPT to R6 is {R3, R6}. R3 determines if its downstream router (R6) has advertised a label-block. Since R6 has advertised a label block 'N4' and it has received R6 'ID' of 6 it will be picking the 6th label value inside the advertised range of its downstream neighbor. Since R3 is the penultimate router to R6 it MUST program a MPLS POP for its own label range Label(N2+6) NH 10.0.0.14 into its MPLS transit RIB. Furthermore R3 MAY program a MPLS NOP for IP 192.168.1.6, NH 10.0.0.14 into its IPv4 tunnel RIB.

4.9. Expansion of an 'All Router Block' subTLV

All routers within a given area MUST advertise their Label Blocks along with an 'ID'. Now assume that the initial label block size assignment is too small to support all routers which generate an ordinal within an IGP domain. Consider the seven routers in Figure 11, and assume that R7 advertises a new ID '15' using an 'All Router ID Map' subTLV. ID '15' is outside of the range of '10' as

per the previous example in Section 4.8. Now all the routers in an IGP domain need to advertise one more label block in order to map the ID '15' to an actual label value.

All routers would advertise in addition to their label block <N> with a size of 10, a second label block <N2> with a size sufficient enough that the new ordinal can get covered. In this example the same block size 10 is used also for the second label block. For example router R2 would advertise the following label bindings.

TLV 149: MPLS Label <N1>, Flags {}:

All Router Block subTLV: Block Size 10, Algo 0, Topology 0

All Router ID IPv4 Map subTLV: ID 2, 192.168.1.2

TLV 149: MPLS Label <N2>, Flags {}:

All Router Block subTLV: Block Size 10, Algo 0, Topology 0

Now the upstream router can map the new ID of R7 to an actual label value, as ID '15' corresponds to the 5th label inside the second Label block.

5. Inter Area Protocol Procedures

5.1. Applicability

Propagation of a MPLS LSP across a level boundary is a local policy decision.

5.2. Data plane operations

If local policy dictates that a given L1L2 router needs to re-advertise a MPLS LSPs from one Level to another then it MUST allocate a new label and program its label forwarding table to connect the new label to the path in the respective other level. Depending on how to reach the re-advertised LSP, this is typically done using a MPLS 'SWAP' or 'SWAP/PUSH' data plane operation.

5.3. Control plane operations

5.3.1. MPLS Label operations

If local policy dictates that a given L1L2 router re-advertises a MPLS LSPs into another Level then it MUST prepend its "Traffic-Engineering-ID" as a loose hop in the Prefix ERO subTLV list. If the

LSP is propagated from a higher Level to a lower Level then the 'Down' bit MUST be set.

5.3.2. MPLS Label Block operations

If local policy dictates that a given L1L2 router advertises its 'All Router Block' into another Level, then it also MUST re-advertise all known 'ID' ordinals (again gated by policy) to the respective other Level. Without knowledge of all 'ID's in the network no router is able to construct SPT label switched paths. If a Label Block and its ID mappings are propagated from a higher Level to a lower Level then the 'Down' bit MUST be set.

6. Acknowledgements

Many thanks to Yakov Rekhter and John Drake for their useful comments.

7. IANA Considerations

This documents request allocation for the following TLVs and subTLVs.

PDU	TLV	subTLV	Type	subType	#Occurence
LSP	MPLS Label		149		>=0
		IPv4 Prefix ERO		1	>=0
		IPv6 Prefix ERO		2	>=0
		Unnumbered Interface ID ERO		9	>=0
		IPv4 Prefix Bypass ERO		3	>=0
		IPv6 Prefix Bypass ERO		4	>=0
		Unnumbered Interface ID Bypass ERO		10	>=0
		All Router Block		6	>=0
		All Router ID IPv4 Map		7	>=0
		All Router ID IPv6 Map		8	>=0

Table 1: IANA allocations

The MPLS Label TLV requires a new sub-registry. Type value 149 has been assigned, with a starting sub-TLV value of 1, range from 1-127, and managed by Expert Review.

8. Security Considerations

This document does not introduce any change in terms of IS-IS security. It simply proposes to flood MPLS label information via the IGP. All existing procedures to ensure message integrity do apply here.

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.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RFC3107] Rekhter, Y. and E. Rosen, "Carrying Label Information in BGP-4", RFC 3107, May 2001.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.
- [RFC3477] Kompella, K. and Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)", RFC 3477, January 2003.
- [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.
- [RFC4761] Kompella, K. and Y. Rekhter, "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, January 2007.
- [RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 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.

- [RFC5151] Farrel, A., Ayyangar, A., and JP. Vasseur, "Inter-Domain MPLS and GMPLS Traffic Engineering -- Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 5151, February 2008.
- [RFC5302] Li, T., Smit, H., and T. Przygienda, "Domain-Wide Prefix Distribution with Two-Level IS-IS", RFC 5302, October 2008.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.
- [RFC5311] McPherson, D., Ginsberg, L., Previdi, S., and M. Shand, "Simplified Extension of Link State PDU (LSP) Space for IS-IS", RFC 5311, February 2009.
- [RFC6119] Harrison, J., Berger, J., and M. Bartlett, "IPv6 Traffic Engineering in IS-IS", RFC 6119, February 2011.

9.2. Informative References

- [I-D.gredler-rtgwg-igp-label-advertisement]
Gredler, H., Amante, S., Scholl, T., and L. Jalil, "Advertising MPLS labels in IGPs", draft-gredler-rtgwg-igp-label-advertisement-05 (work in progress), May 2013.
- [I-D.ietf-rtgwg-mrt-frr-architecture]
Atlas, A., Kebler, R., Envedi, G., Csaszar, A., Tantsura, J., Konstantynowicz, M., White, R., and M. Shand, "An Architecture for IP/LDP Fast-Reroute Using Maximally Redundant Trees", draft-ietf-rtgwg-mrt-frr-architecture-02 (work in progress), February 2013.
- [I-D.previdi-filsfils-isis-segment-routing]
Previdi, S., Filsfils, C., Bashandy, A., Horneffer, M., Decraene, B., Litkowski, S., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., and J. Tantsura, "Segment Routing with IS-IS Routing Protocol", draft-previdi-filsfils-isis-segment-routing-02 (work in progress), March 2013.

Authors' Addresses

Hannes Gredler (editor)
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: hannes@juniper.net

Shane Amante
Level 3 Communications, Inc.
1025 Eldorado Blvd
Broomfield, CO 80021
US

Email: shane@level3.net

Tom Scholl
Amazon
Seattle, WN
US

Email: tscholl@amazon.com

Luay Jalil
Verizon
1201 E Arapaho Rd.
Richardson, TX 75081
US

Email: luay.jalil@verizon.com

Open Shortest Path First IGP
Internet-Draft
Intended status: Standards Track
Expires: November 22, 2013

H. Gredler, Ed.
Juniper Networks, Inc.
S. Amante
Level 3 Communications, Inc.
T. Scholl
Amazon
L. Jalil
Verizon
May 21, 2013

Advertising MPLS labels in OSPF
draft-gredler-ospf-label-advertisement-03

Abstract

Historically MPLS label distribution was driven by protocols like LDP, RSVP and LBGp. All of those protocols are session oriented. In order to obtain label binding for a given destination FEC from a given router one needs first to establish an LDP/RSVP/LBGp session with that router.

Advertising MPLS labels in IGPs
[I-D.gredler-rtgwg-igp-label-advertisement] describes several use cases where utilizing the flooding machinery of link-state protocols for MPLS label distribution allows to obtain the binding without requiring to establish an LDP/RSVP/LBGp session with that router.

This document describes the protocol extension to distribute MPLS label bindings by the OSPFv2 and OSPFv3 protocol.

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 November 22, 2013.

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	5
2. Motivation, Rationale and Applicability	5
2.1. Issue: Bi-directionality semantics	6
2.2. Issue: IP path semantics	6
2.3. Issue: Lack of 'path' notion	6
2.4. Motivation	7
3. OSPF MPLS LSA Format	7
3.1. Common LSA Type	7
3.2. OSPFv2 LSA ID	7
3.3. OSPFv2 LSA Format Overview	8
3.4. OSPFv3 LSA ID	8
3.5. OSPFv3 LSA Format Overview	9
3.6. TLV Header	9
4. LSA payload details	10
4.1. ERO TLVs	10
4.1.1. IPv4 Prefix ERO TLV	10
4.1.2. IPv6 Prefix ERO TLV	11
4.1.3. Unnumbered Interface ID ERO TLV	12
4.1.4. IPv4 Prefix Bypass ERO TLV	13
4.1.5. IPv6 Prefix Bypass ERO TLV	13
4.1.6. Unnumbered Interface ID Bypass ERO TLV	14
4.2. Flags TLV	15
4.3. All Router Block TLV	15
4.4. All Router ID IPv4 Map TLV	17
4.5. All Router ID IPv6 Map TLV	18
5. Advertising Label Examples	18
5.1. Sample Topology	18
5.1.1. Transport IP addresses and router-IDs	19
5.1.2. Link IP addresses	19
5.2. One-hop LSP to an adjacent Router	20
5.3. One-hop LSP to an adjacent Router using a specific link	20
5.4. One-hop LSP to an adjacent external Router	20
5.5. Advertisement of an RSVP LSP	21
5.6. Advertisement of an LDP LSP	21
5.7. Interarea advertisement of diverse paths	21
5.8. Advertisement of SPT labels using 'All Router Block' TLV	22
5.9. Expansion of an 'All Router Block' TLV	23
6. Inter Area Protocol Procedures	24
6.1. Applicability	24
6.2. Data plane operations	24
6.3. Control plane operations	24
6.3.1. MPLS Label operations	24
6.3.2. MPLS Label Block operations	25
7. Acknowledgements	25
8. IANA Considerations	25

9. Security Considerations	26
10. References	26
10.1. Normative References	26
10.2. Informative References	27
Authors' Addresses	27

1. Introduction

MPLS label allocations are predominantly distributed by using the LDP [RFC5036], RSVP [RFC5151] or labeled BGP [RFC3107] protocol. All of those protocols have in common that they are session oriented, which means that in order to obtain label binding for a given destination FEC from a given router one needs first to establish a direct control plane (LDP/RSVP/LBGP) session with that router.

There are a couple of practical use cases [I-D.gredler-rtgwg-igp-label-advertisement] where the consumer of a MPLS label binding may not be adjacent to the router that performs the binding. Bringing up an explicit session using the existing label distribution protocols between the non-adjacent router that binds the label and the router that acts as a consumer of this binding is the existing remedy for this dilemma.

This document describes an OSPFv2 and OSPFv3 protocol extension which allows routers to advertise MPLS label bindings within and beyond an IGP domain, and controlling inter-area distribution.

2. Motivation, Rationale and Applicability

Distributing MPLS labels in an IGP (IS-IS) has been described in Segment Routing [I-D.previdi-filsfils-isis-segment-routing]. The authors propose to re-use existing traffic-engineering extensions for carrying the label information. While retrofitting existing protocol machinery for new purposes is generally a good thing, Segment Routing [I-D.previdi-filsfils-isis-segment-routing] falls short of addressing some use-cases defined in [I-D.gredler-rtgwg-igp-label-advertisement].

The dominant issue around re-using traffic-engineering extensions is that both have existing protocol semantics, which might not be applicable to advertising MPLS label switched paths in a generic fashion. These are specifically:

- o Bi-directionality semantics
- o IP path semantics
- o Lack of 'path' notion

2.1. Issue: Bi-directionality semantics

'Bi-directionality semantics', affects the complexity around advertisement of unidirectional LSPs. Label advertisement of per-link labels or 'Adj-SIDs' [I-D.previdi-filsfils-isis-segment-routing] is done by attaching label information to adjacency advertisement TLVs. Usually implementations need to have an adjacency in 'Up' state prior to advertising this adjacency as TE-Link in its Link State advertisement. In order to advertise a per-link LSP an implementation first needs to have an adjacency, which only transitions to 'Up' state after passing the 3-way check. This implies bi-directionality. If an implementation wants to advertise per-link MPLS LSPs to e.g. outside the IGP domain then it would need to fake-up an adjacency. Changing existing IGP Adjacency code to support such cases defeats the purpose of re-using existing functionality as there is not much common functionality to be shared.

2.2. Issue: IP path semantics

LSPs pointing to a Node are advertised as 'Node-SIDs' [I-D.previdi-filsfils-isis-segment-routing] using IP Prefix containers. That means that in order to advertise a MPLS LSP, one is inheriting the semantics of advertising an IP path. Consider router A has got existing LSPs to its entire one-hop neighborhood and is re-advertising those LSPs using IP reachability semantics. Now we have two exact matching IP advertisements. One from the owning router (router B) which advertises its stable transport loopback address and another one from router A re-advertising a LSP path to router B. Existing routing software may get confused now as the 'stable transport' address shows up from multiple places in the network and more worse the IP forwarding path for control-plane protocols may get mingled with the MPLS data plane.

2.3. Issue: Lack of 'path' notion

Both existing traffic-engineering extension containers have limited semantics describing MPLS label-switched paths in the sense of a 'path'. Both encoding formats allow to express a pointer to some specific router, but not to describe a MPLS label switched path containing all of its path segments. [I-D.previdi-filsfils-isis-segment-routing] allows to define 'Forwarding Adjacencies' as per [RFC4206]. The way to describe a path of a given forwarding adjacency is to enlist a list of "Segment IDs". That implies that nodes which do not yet participate in 'Segment routing' or are outside of a 'Segment routing' domain can not be expressed using those path semantics.

A protocol for advertising MPLS label switched paths, should be

generic enough to express paths sourced by existing MPLS LSPs, such that ingress routers can flexibly combine them according to application needs.

2.4. Motivation

IGP advertisement of MPLS label switched paths requires a new set of protocol semantics (undirectional paradigm, path paradigm), which hardly can be expressed using the existing OSPF and OSPF-TE protocol semantics. This document describes protocol extensions which allows generic advertisement of MPLS label bindings in the OSPFv2 and OSPFv3 protocol.

The Protocol extensions described in this document are equally applicable to IPv4 and IPv6 carried over MPLS. Furthermore the proposed use of distributing MPLS Labels using IGP protocols adheres to the architectural principles laid out in [RFC3031].

3. OSPF MPLS LSA Format

3.1. Common LSA Type

One new LSA is defined, the MPLS Label LSA. This LSA advertises MPLS labels along with their path information. The LSA contains more specific information encoded in TLVs. Those TLV extensions are shared between the OSPFv2 and OPSFv3 protocols.

3.2. OSPFv2 LSA ID

The LSA ID of an Opaque LSA is defined as having eight bits of type data and 24 bits of type-specific data. The MPLS Label LSA uses type 149. The remaining 24 bits are 4 zero bits followed by the MPLS Label or MPLS Label Base value as follows:

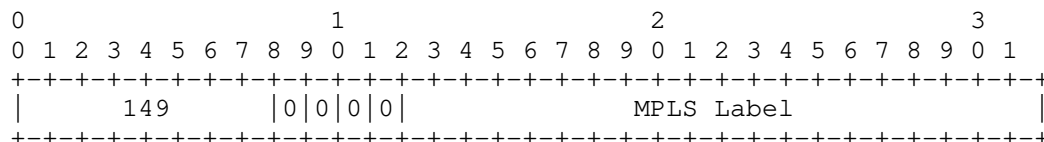


Figure 1: OSPFv2 MPLS Label LSA-ID format

The 'MPLS Label' field holds the 20 Bit MPLS label for MPLS Label base. Therefore a maximum of 2^{20} MPLS Label LSAs may be sourced by a single system.

3.3. OSPFv2 LSA Format Overview

This extension makes use of the Opaque LSAs [RFC5250].

Three types of Opaque LSAs exist, each of which has a different flooding scope. This proposal uses only Type 10 LSAs, which have an area flooding scope.

The MPLS Label LSA for OSPFv2 starts with the standard OSPFv2 LSA header:

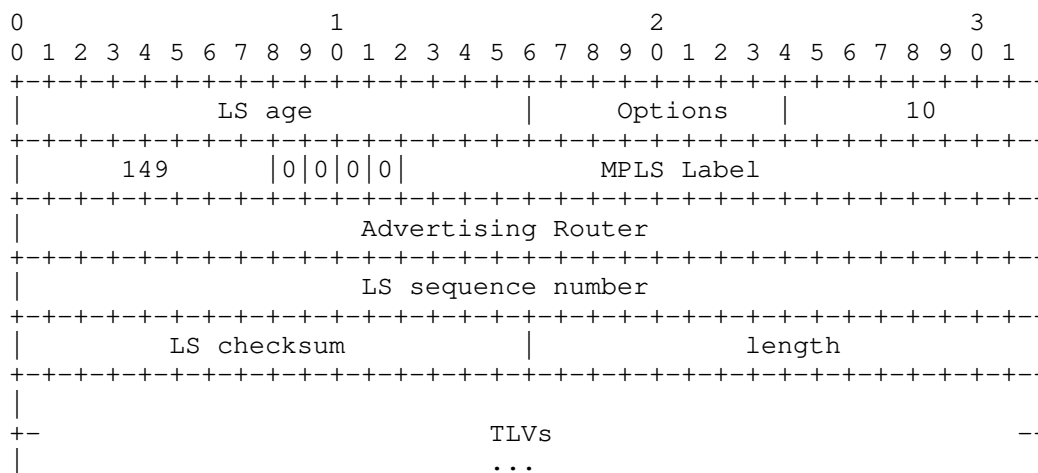


Figure 2: OSPFv2 MPLS Label LSA format

3.4. OSPFv3 LSA ID

The OSPFv3 LSA ID of an MPLS Label LSA is defined as having twelve bits of zero followed by the 20-Bit label MPLS Label value as follows:

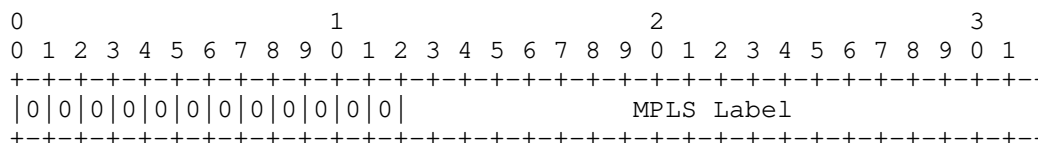


Figure 3: OSPFv3 MPLS Label LSA-ID format

The 'MPLS Label' field holds the 20 Bit MPLS label or MPLS label base value. Therefore a maximum of 2^{20} MPLS Label LSAs may be sourced by a single system.

3.5. OSPFv3 LSA Format Overview

The MPLS Label LSA for OSPFv3 starts with the standard OSPFv3 LSA header:

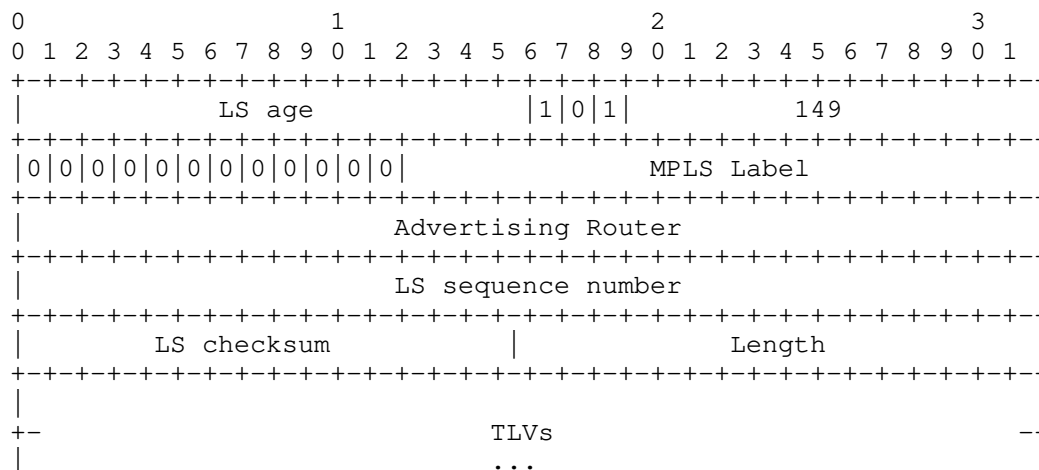


Figure 4: OSPFv3 MPLS Label LSA format

The OSPFv3 'U' Bit will always be set such that routers which do not understand the new MPLS Label LSA will store and forward it further.

In analogy to the OSPFv2 opaque LSA 10 the flooding scope will be set to 'Area scoping'.

3.6. TLV Header

The LSA payload consists of one or more nested Type/Length/Value (TLV) triplets for extensibility. The format of each TLV is:

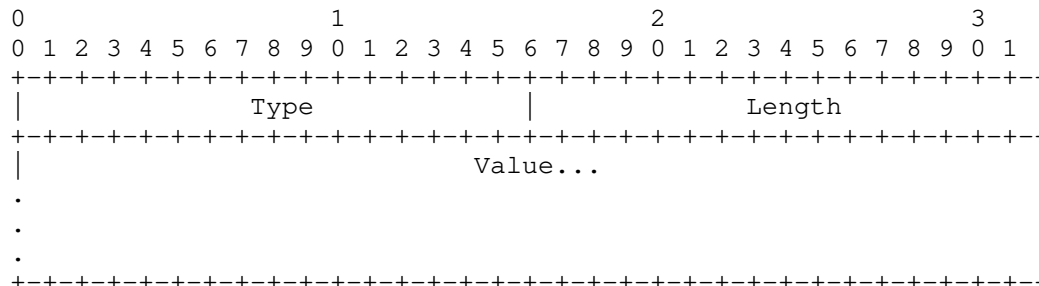


Figure 5: TLV format

The Length field defines the length of the value portion in octets (thus a TLV with no value portion would have a length of zero). The TLV is padded to four-octet alignment; padding is not included in the length field (so a three octet value would have a length of three, but the total size of the TLV would be eight octets). Nested TLVs are also 32-bit aligned. Unrecognized types are ignored.

This memo defines TLV Types 1 through 8. See the IANA Considerations section for allocation of new Types.

4. LSA payload details

The MPLS Label LSA may be originated by any Traffic Engineering [RFC3630] capable router in an OSPF domain. The router may advertise a single label binding or a block of label bindings. For single label binding advertisement a router needs to provide at least a single 'nexthop style' anchor. The protocol supports more than one 'nexthop style' anchor to be attached to a Label binding, which results into a simple path description language. In analogy to RSVP the terminology for this is called an 'Explicit Route Object' (ERO). Since ERO style path notation allows to anchor label bindings to both link and node IP addresses any label switched path, can be described. Furthermore also Label Bindings from other protocols can get easily re-advertised.

An LSA contains one or more TLVs, describing properties of the advertised MPLS label.

The following TLV extensions may be shared in both OSPV2 and OSPFv3. Passing an IP address of the other address family (IPv4 in OPSFv3 or IPv6 in OSPFv2) is possible as the information carried are related describing the hops along a path. The receiver of this information is a protocol agnostic path computation module.

4.1. ERO TLVs

All 'ERO' information represents an ordered set which describes the segments of a label-switched path. The last ERO TLV describes the segment closest to the egress point of the LSP. Contrary the first ERO TLV describes the first segment of a label switched path. If a router extends or stitches a label switched path it MUST prepend the new segments path information to the ERO list.

4.1.1. IPv4 Prefix ERO TLV

The IPv4 ERO TLV (Type 1) describes a path segment using IPv4 Prefix style of encoding. Its appearance and semantics have been borrowed

from Section 4.3.3.2 [RFC3209].

the 'IPv4 Address' is treated as a prefix based on the prefix length value below. Bits beyond the prefix are ignored on receipt and SHOULD be set to zero on transmission.

The 'Prefix Length' field contains the length of the prefix in bits.

The 'L' bit in the TLV is a one-bit attribute. If the L bit is set, then the value of the attribute is 'loose.' Otherwise, the value of the attribute is 'strict.'

The 'Reserved' bits are for future use. They should be zero on transmission and ignored on receipt.

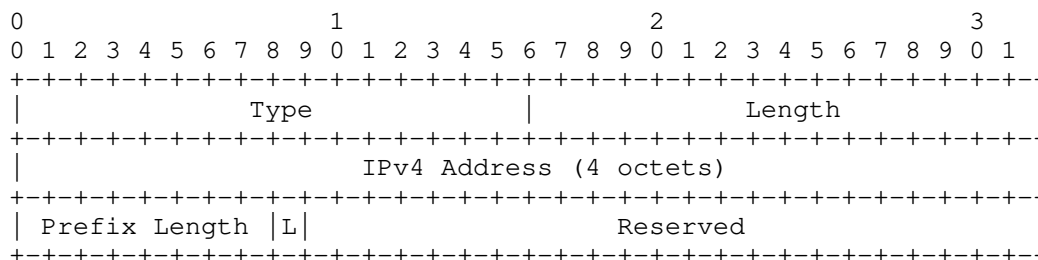


Figure 6: IPv4 Prefix ERO TLV format

4.1.2. IPv6 Prefix ERO TLV

The IPv6 ERO TLV (Type 2) describes a path segment using IPv6 Prefix style of encoding. Its appearance and semantics have been borrowed from Section 4.3.3.2 [RFC3209].

the 'IPv6 Address' is treated as a prefix based on the prefix length value below. Bits beyond the prefix are ignored on receipt and SHOULD be set to zero on transmission.

The 'Prefix Length' field contains the length of the prefix in bits.

The 'L' bit in the TLV is a one-bit attribute. If the L bit is set, then the value of the attribute is 'loose.' Otherwise, the value of the attribute is 'strict.'

The 'Reserved' bits are for future use. They should be zero on transmission and ignored on receipt.

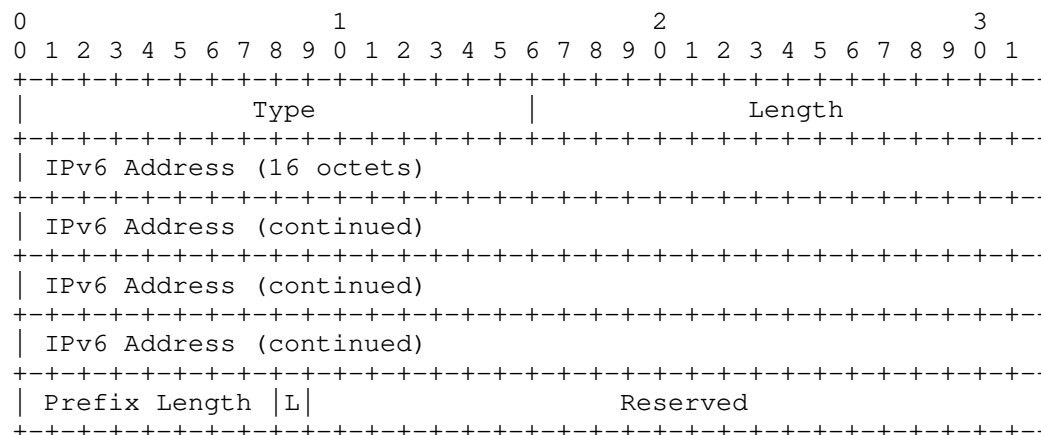


Figure 7: IPv6 Prefix ERO TLV format

4.1.3. Unnumbered Interface ID ERO TLV

The appearance and semantics of the 'Unnumbered Interface ID' have been borrowed from Section 4 [RFC3477].

The Unnumbered Interface-ID ERO TLV (Type 9) describes a path segment that spans over an unnumbered interface. Unnumbered interfaces are referenced using the interface index. Interface indices are assigned local to the router and therefore not unique within a domain. All elements in an ERO path need to be unique within a domain and hence need to be disambiguated using a domain unique Router-ID.

The 'Router-ID' field contains the router ID of the router which has assigned the 'Interface ID' field. Its purpose is to disambiguate the 'Interface ID' field from other routers in the domain.

The 'Interface ID' is the identifier assigned to the link by the router specified by the router ID.

The 'L' bit in the TLV is a one-bit attribute. If the L bit is set, then the value of the attribute is 'loose.' Otherwise, the value of the attribute is 'strict.'

The 'Reserved' bits are for future use. They should be zero on transmission and ignored on receipt.

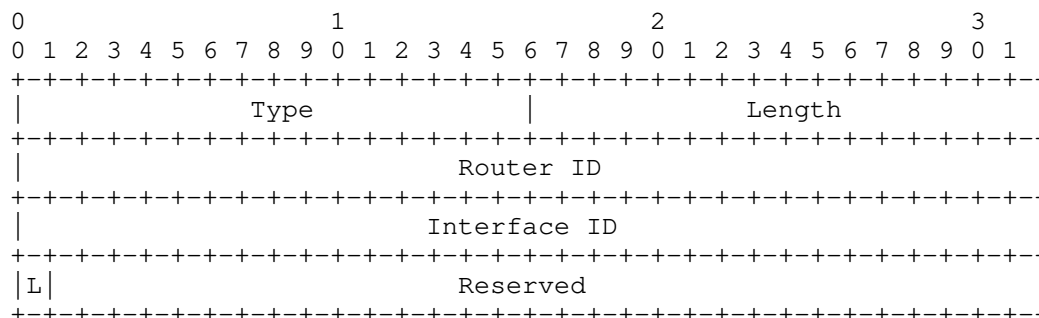


Figure 8: Unnumbered Interface ID ERO TLV format

4.1.4. IPv4 Prefix Bypass ERO TLV

The IPv4 Bypass ERO TLV (Type 3) describes a Bypass LSP path segment using IPv4 Prefix style of encoding. Its appearance and semantics have been borrowed from Section 4.3.3.2 [RFC3209].

The 'Prefix Length' field contains the length of the prefix in bits.

The 'L' bit in the TLV is a one-bit attribute. If the L bit is set, then the value of the attribute is 'loose.' Otherwise, the value of the attribute is 'strict.'

The 'Reserved' bits are for future use. They should be zero on transmission and ignored on receipt.

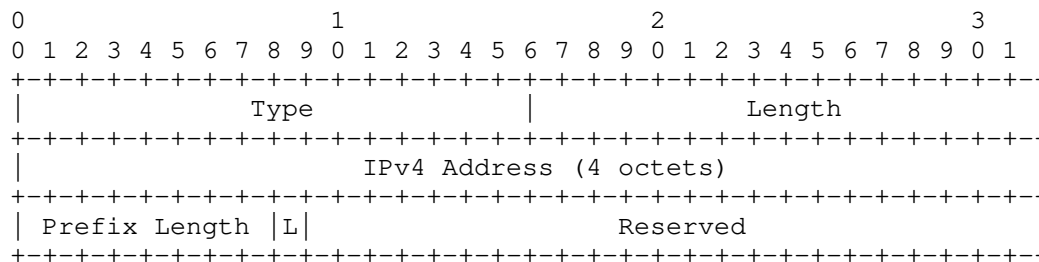


Figure 9: IPv4 Prefix Bypass ERO TLV format

4.1.5. IPv6 Prefix Bypass ERO TLV

The IPv6 ERO TLV (Type 4) describes a Bypass LSP path segment using IPv6 Prefix style of encoding. Its appearance and semantics have been borrowed from Section 4.3.3.3 [RFC3209].

The 'Prefix Length' field contains the length of the prefix in bits.

The 'L' bit in the TLV is a one-bit attribute. If the L bit is set, then the value of the attribute is 'loose.' Otherwise, the value of the attribute is 'strict.'

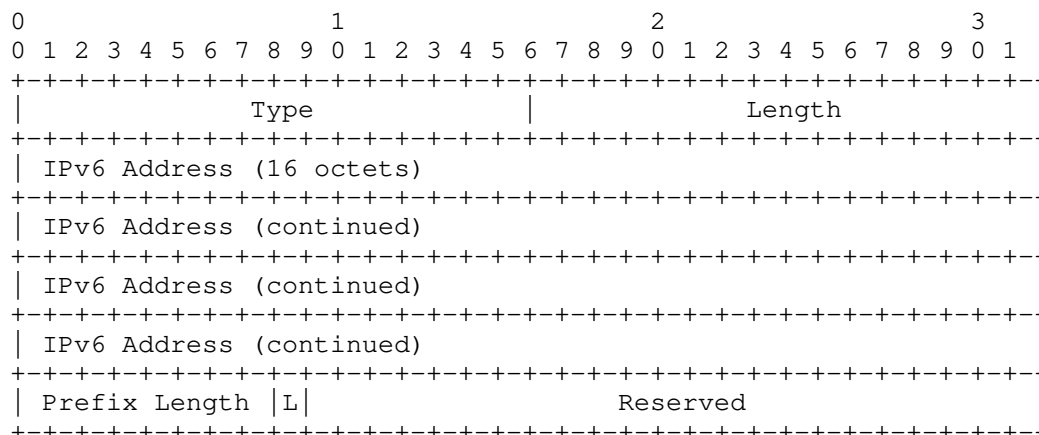


Figure 10: IPv6 Prefix Bypass ERO TLV format

4.1.6. Unnumbered Interface ID Bypass ERO TLV

The appearance and semantics of the 'Unnumbered Interface ID' have been borrowed from Section 4 [RFC3477].

The Unnumbered Interface-ID Bypass ERO TLV (Type 10) describes a Bypass path segment that spans over an unnumbered interface. Unnumbered interfaces are referenced using the interface index. Interface indices are assigned local to the router and therefore not unique within a domain. All elements in an ERO path need to be unique within a domain and hence need to be disambiguated using a domain unique Router-ID.

The 'Router-ID' field contains the router ID of the router which has assigned the 'Interface ID' field. Its purpose is to disambiguate the 'Interface ID' field from other routers in the domain.

The 'Interface ID' is the identifier assigned to the link by the router specified by the router ID.

The 'L' bit in the TLV is a one-bit attribute. If the L bit is set, then the value of the attribute is 'loose.' Otherwise, the value of the attribute is 'strict.'

The 'Reserved' bits are for future use. They should be zero on transmission and ignored on receipt.

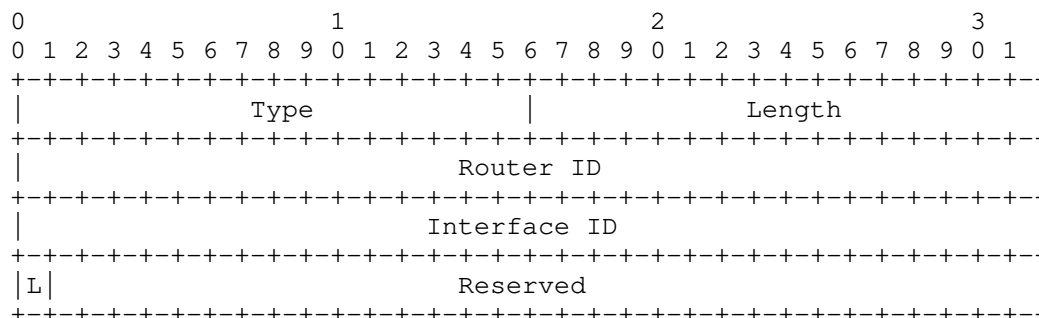


Figure 11: Unnumbered Interface ID Bypass ERO TLV format

4.2. Flags TLV

The Flags TLV (Type 5) describes Flags for further MPLS LSA treatment.

Up/Down 'U' Bit: A router may flood MPLS label information across area boundaries. In order to prevent flooding loops, a router will Set the Up/Down (U-Bit) when propagating MPLS labels from Area 0 to a non-zero Area.

The 'Reserved' bits are for future use. They should be zero on transmission and ignored on receipt.

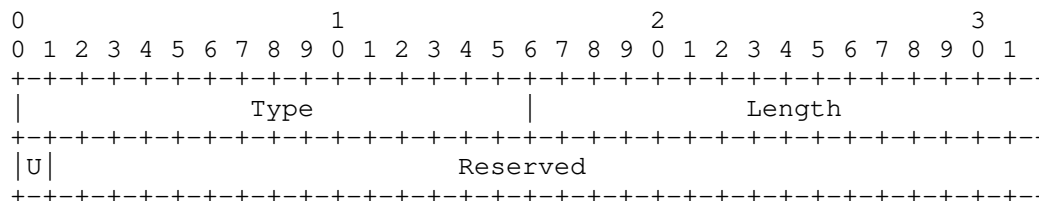


Figure 12: Flags TLV format

4.3. All Router Block TLV

The 'All Router Block' TLV (Type 6) denominates the label block size of an MPLS Label advertisement and its semantics to connect to all routers in a given OSPF domain using a local assigned [RFC3031] label range. Note that the actual mapping of a router within the label range is done using the TLVs described in Section 4.4 and Section 4.5. Since generation of an IPv4 or IPv6 Map TLV is a local policy decision, it might be the case that connectivity is provided not to 'All' but rather a subset of 'All' routers. Keeping policy decisions aside, for simplicity reasons, assume that All Routers in a

domain do generate either the 'All Router ID IPv4 Map' or 'All Router ID IPv6 Map' TLVs and therefore all routers desire construction of a Label switched path from every source router in the network. The basic concept of using label blocks to provide connectivity to a set of routers has been borrowed from [RFC4761] which allows to advertise labels from multiple end-points using a single control-plane message. The difference to [RFC4761] is that rather than advertising where a particular packet came from (=source semantics), destination semantics (where a particular packet will be going to) is advertised.

Along with each label block a router advertises one for more 'IDs'. The 'ID' must be unique within a given domain. The 'ID' serves as ordinal to determine the actual label value inside the set of all advertised label ranges of a given router. A receiving router uses the ordinal to determine the actual label value in order to construct forwarding state to a particular destination router. The 'ID' is separately advertised using the TLVs described in Section 4.4 and Section 4.5.

The ability to advertise more than one label block eases operational procedures for increasing the number of supported routers within a domain. For example consider a given domain has got support for <M> routers and runs out of ID space. It simply advertises one more label block to cover additional ordinals outside the range of the first label block. An example of label-block expansion is described in more detail in Section 5.9

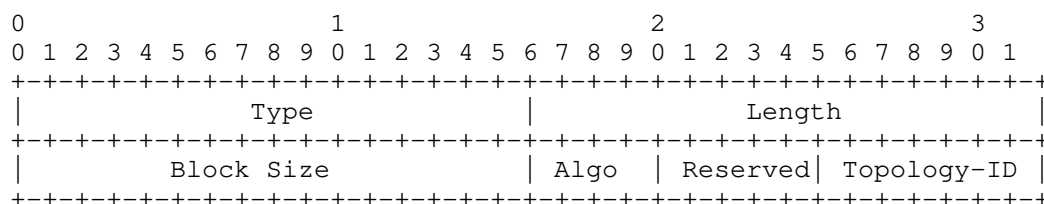


Figure 13: All Router Block TLV format

The 'Block Size' value contains the size of the label advertisement. The 'value determines the amount of reachable router endpoints within a given Label block. It MUST contain a value greater or equal than two. Note that the label base is inferred from the LSA-ID in the LSA header. For example if a router wants to advertise a label range of 5000-5099 then it would need to generate a LSA-ID of 5000 (= 0.0.19.136) and a Block Size of 100.

The 'Algo' value denominates the path computation algorithm in order to calculate the forwarding topology. The basic SPF algorithm has an assigned 'Algo' code point of zero. The purpose of the 'Algo' field

is to extend the notion of Label Block Signaling to arbitrary algorithms like for example 'MRT' ([I-D.ietf-rtgwg-mrt-frr-architecture]). Advertised Label Blocks with an unknown, unsupported or non-configured algorithm MUST be silently ignored.

The 'Reserved' bits are for future use. They should be zero on transmission and ignored on receipt.

The 'Topology-ID' field contains the Multi Topology ID ([RFC4915]) for which the advertised Label Block does apply. The basic IPv4 unicast Topology has an assigned 'Topology-ID' code point of zero. Advertised Label Blocks with an unknown, unsupported or non-configured Topology-ID MUST be silently ignored.

An LSA containing the 'All Router Block' TLV MUST only contain the Flags TLV (Section 4.2, the 'All Router IPv4 Map' TLV (Section 4.4) or the 'All Router IPv6 Map' TLV (Section 4.5).

4.4. All Router ID IPv4 Map TLV

The 'All Router ID IPv4 Map' TLV (Type 7) maps an 'ID' to a given stable transport IPv4 address. Its purpose is to associate a given transport IPv4 IP address to the ordinal inside a label range as described in Section 4.3.

A router MAY advertise more than one 'ID' to 'IPv4 address' mapping pair, in case it has more than one stable transport IPv4 address.

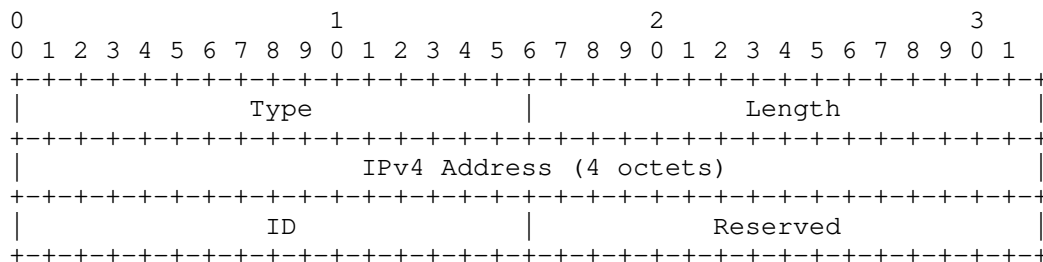


Figure 14: All Router ID IPv4 Map TLV format

The 'IPv4 address' contains stable IPv4 transport address of a given router.

The 'ID' contains the ordinal value of an advertising router inside the set of all advertised label blocks of a given router.

The 'Reserved' bits are for future use. They should be zero on

transmission and ignored on receipt.

4.5. All Router ID IPv6 Map TLV

The 'All Router ID IPv6 Map' TLV (Type 8) maps an 'ID' to a given stable transport IPv6 address. Its purpose is to associate a given transport IPv6 IP address to the ordinal inside a label range as described in Section 4.3.

A router MAY advertise more than one 'ID' to 'IPv6 address' mapping pair, in case it has more than one stable transport IPv6 address.

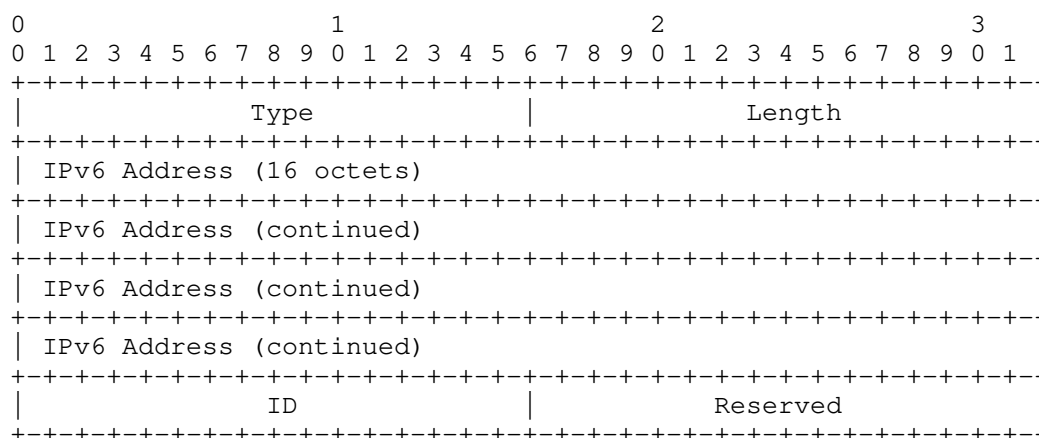


Figure 15: All Router ID IPv6 Map TLV format

The 'IPv6 address' contains the stable IPv6 transport address of a given router.

The 'ID' contains the ordinal value of an advertising router inside the set of all advertised label blocks of a given router.

The 'Reserved' bits are for future use. They should be zero on transmission and ignored on receipt.

5. Advertising Label Examples

5.1. Sample Topology

The following topology (Figure 16) and IP addresses shall be used throughout the Label advertisement examples.

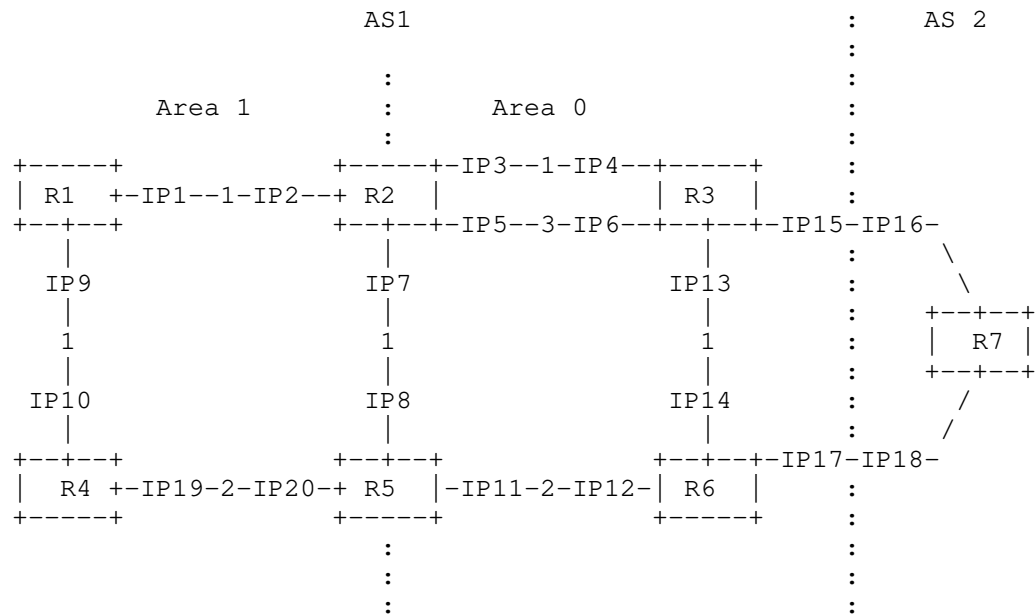


Figure 16: Sample Topology

5.1.1. Transport IP addresses and router-IDs

- o R1: 192.168.1.1
- o R2: 192.168.1.2
- o R3: 192.168.1.3
- o R4: 192.168.1.4
- o R5: 192.168.1.5
- o R6: 192.168.1.6
- o R7: 192.168.1.7

5.1.2. Link IP addresses

- o R1 to R2 link: 10.0.0.1, 10.0.0.2
- o R1 to R4 link: 10.0.0.9, 10.0.0.10
- o R2 to R3 link #1: 10.0.0.3, 10.0.0.4

- o R2 to R3 link #2: 10.0.0.5, 10.0.0.6
- o R2 to R5 link: 10.0.0.7, 10.0.0.8
- o R3 to R6 link: 10.0.0.13, 10.0.0.14
- o R3 to R7 link: 10.0.0.15, 10.0.0.16
- o R4 to R5 link: 10.0.0.19, 10.0.0.20
- o R5 to R6 link: 10.0.0.11, 10.0.0.12
- o R6 to R7 link: 10.0.0.17, 10.0.0.18

The IGP link metrics are displayed in the middle of the link. All of them are assumed to be bi-directional.

5.2. One-hop LSP to an adjacent Router

If R1 would advertise a label <N> bound to a one-hop LSP from R1 to R2 it would encode as follows:

LSA 149, LSA-ID <N>:

IPv4 Prefix ERO TLV: 192.168.1.2/32, Strict

5.3. One-hop LSP to an adjacent Router using a specific link

If R2 would advertise a label <N> bound to a one-hop LSP from R2 to R3, using the link #2 it would encode as follows

LSA 149, LSA-ID <N>:

IPv4 Prefix ERO TLV: 10.0.0.6/32, Strict

5.4. One-hop LSP to an adjacent external Router

If R3 would advertise a label <N> bound to a one-hop LSP from R3 to R7 (which is outside of the IGP domain), it would encode as follows:

LSA 149, LSA-ID <N>:

IPv4 Prefix ERO TLV: 192.168.1.7/32, Strict

As you can see the representation of an MPLS label crossing an external link is identical as an internal link Section 5.2.

5.5. Advertisement of an RSVP LSP

Consider a RSVP LSP name "R2-to-R6" traversing (R2 to R3 using link #1, R6):

If R2 would advertise a label <N> bound to the RSVP LSP named 'R2-to-R6', it would encode as follows

LSA 149, LSA-ID <N>:

IPv4 Prefix ERO TLV: 10.0.0.4/32, Strict

IPv4 Prefix ERO TLV: 192.168.1.6/32, Strict

5.6. Advertisement of an LDP LSP

Consider R2 that creates a LDP label binding for FEC 172.16.0.0./12 using label <N>.

If R2 would re-advertise this label binding in OSPF it would encode as follows

LSA 149, LSA-ID <N>:

IPv4 Prefix ERO TLV: 172.16.0.0/12, Loose

5.7. Interarea advertisement of diverse paths

Consider two R2->R6 paths: {R2, R3, R6} and {R2, R5, R6}

Consider two R5->R3 paths: {R5, R2, R3} and {R5, R6, R3}

R2 encodes its two paths to R6 as follows:

LSA 149, LSA-ID <N1>:

IPv4 Prefix ERO TLV: 192.168.1.3, Loose

IPv4 Prefix ERO TLV: 192.168.1.6, Loose

Flags TLV: Down

LSA 149, LSA-ID <N2>:

IPv4 Prefix ERO TLV: 192.168.1.5, Loose

IPv4 Prefix ERO TLV: 192.168.1.6, Loose

Flags TLV: Down

R5 encodes its two paths to R3 as follows:

LSA 149, LSA-ID <N1>:

IPv4 Prefix ERO TLV: 192.168.1.2, Loose

IPv4 Prefix ERO TLV: 192.168.1.3, Loose

Flags TLV: Down

LSA 149, LSA-ID <N2>:

IPv4 Prefix ERO TLV: 192.168.1.6, Loose

IPv4 Prefix ERO TLV: 192.168.1.3, Loose

Flags TLV: Down

A receiving non-backbone router does see now all 4 paths and may decide to load-balance across all or a subset of them.

5.8. Advertisement of SPT labels using 'All Router Block' TLV

All routers within a given area MUST advertise their Label Blocks along with an 'ID'.

If R2 would advertise a label block <N1> with a size of 10, declaring SPT label forwarding support to all routers within a given domain, it would encode as follows:

LSA 149, LSA-ID <N1>:

All Router Block TLV: Block Size 10, Algo 0, Topology-ID 0

All Router ID IPv4 Map TLV: ID 2, 192.168.1.2

If R3 would advertise a label block <N2> with a size of 10, declaring SPT label forwarding support to all routers within a given domain, it would encode as follows:

LSA 149, LSA-ID <N2>:

All Router Block TLV: Block Size 10, Algo 0, Topology-ID 0

All Router ID IPv4 Map TLV: ID 3, 192.168.1.3

If R5 would advertise a label block <N3> with a size of 10, declaring SPT label forwarding support to all routers within a given domain, it would encode as follows:

LSA 149, LSA-ID <N3>:

All Router Block TLV: Block Size 10, Algo 0, Topology-ID 0

All Router ID IPv4 Map TLV: ID 5, 192.168.1.5

If R6 would advertise a label block <N4> with a size of 10, declaring SPT label forwarding support to all routers within a given domain, it would encode as follows:

LSA 149, LSA-ID <N4>:

All Router Block TLV: Block Size 10, Algo 0, Topology-ID 0

All Router ID IPv4 Map TLV: ID 6, 192.168.1.6

Consider now R2 constructing a SPT label for R6. R2s SPT to R6 is {R2, IP4, R3, R6}. R2 first determines if its downstream router (R3) has advertised a label-block. Since R3 has advertised a label block 'N2' and it has received R6 'ID' of 6 it will be picking the 6th label value inside the advertised range of its downstream neighbor. Specifically R2 MUST be program a MPLS SWAP for its own label range Label(N1+6) to Label(N2+6), NH 10.0.0.4 into its MPLS transit RIB. Furthermore R2 MAY program a MPLS PUSH operation for IP 192.168.1.6 to Label (N2+6), NH 10.0.0.4 into its IPv4 tunnel RIB.

Next walk down to R3, which is the next router on the SPT tree towards R6. R3s SPT to R6 is {R3, R6}. R3 determines if its downstream router (R6) has advertised a label-block. Since R6 has advertised a label block 'N4' and it has received R6 'ID' of 6 it will be picking the 6th label value inside the advertised range of its downstream neighbor. Since R3 is the penultimate router to R6 it MUST program a MPLS POP for its own label range Label(N2+6) NH 10.0.0.14 into its MPLS transit RIB. Furthermore R3 MAY program a MPLS NOP for IP 192.168.1.6, NH 10.0.0.14 into its IPv4 tunnel RIB.

5.9. Expansion of an 'All Router Block' TLV

All routers within a given area MUST advertise their Label Blocks along with an 'ID'. Now assume that the initial label block size assignment is too small to support all routers which generate an ordinal within an IGP domain. Consider the seven routers in Figure 16, and assume that R7 advertises a new ID '15' using an 'All Router ID Map' TLV. ID '15' is outside of the range of '10' as per

the previous example in Section 5.8. Now all the routers in an IGP domain need to advertise one more label block in order to map the ID '15' to an actual label value.

All routers would advertise in addition to their label block <N> with a size of 10, a second label block <N2> with a size sufficient enough, that the new ordinal can get covered. In this example the same block size 10 is used also for the second label block. For example router R2 would advertise the following label bindings.

LSA 149: LSA-ID <N1>:

All Router Block TLV: Block Size 10, Algo 0, Topology 0

All Router ID IPv4 Map TLV: ID 2, 192.168.1.2

LSA 149: LSA-ID <N2>:

All Router Block TLV: Block Size 10, Algo 0, Topology 0

Now the upstream router can map the new ID of R7 to an actual label value, as ID '15' corresponds to the 5th label inside the second Label block.

6. Inter Area Protocol Procedures

6.1. Applicability

Propagation of a MPLS LSPs and MPLS Block LSPs across an area boundary is a local policy decision.

6.2. Data plane operations

If local policy dictates that a given ABR router needs to re-advertise a MPLS LSPs from one area to another then it MUST allocate a new label and program its label forwarding table to connect the new label to the path in the respective other area. Depending on how to reach the re-advertised LSP, this is typically done using a MPLS 'SWAP' or 'SWAP/PUSH' data plane operation.

6.3. Control plane operations

6.3.1. MPLS Label operations

If local policy dictates that a given ABR router needs to re-advertise MPLS LSPs from one area to another then it must prepend its "Traffic-Engineering-ID" as a loose hop in the Prefix ERO TLV list.

Furthermore it MUST append the Flags TLV and set the 'Down' Bit.

6.3.2. MPLS Label Block operations

If local policy dictates that a given ABR router advertises its 'All Router Block' into another area, then it also MUST re-advertise all known 'ID' ordinals (again gated by policy) to the respective other area. Without knowledge of all 'ID's in the network no router is able to construct SPT label switched paths. Furthermore an ABR MUST append the Flags TLV and set the 'Down' Bit for all re-advertised 'CE' IDs.

7. Acknowledgements

Many thanks to Yakov Rekhter, John Drake and Shraddha Hedge for their useful comments.

8. IANA Considerations

This documents request allocation for one common OSPFv2 and OSPFv3 LSA Type and TLVs contained within.

LSA Type	TLV	TLV Type	#Occurence
149	IPv4 Prefix ERO	1	>=0
	IPv6 Prefix ERO	2	>=0
	Unnumbered Interface ID ERO	9	>=0
	IPv4 Prefix Bypass ERO	3	>=0
	IPv6 Prefix Bypass ERO	4	>=0
	Unnumbered Bypass Interface ID ERO	10	>=0
	Flags	5	0,1
	All Router Block	6	>=0
	All Router ID IPv4 Map	7	>=0
	All Router ID IPv6 Map	8	>=0

Table 1: IANA allocations

The MPLS Label LSA requires a new sub-registry, with a starting TLV value of 1, and managed by IETF consensus.

9. Security Considerations

This document does not introduce any change in terms of OSPF security. It simply proposes to flood MPLS label information via the IGP. All existing procedures to ensure message integrity do apply here.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RFC3107] Rekhter, Y. and E. Rosen, "Carrying Label Information in BGP-4", RFC 3107, May 2001.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.
- [RFC3477] Kompella, K. and Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)", RFC 3477, January 2003.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.
- [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.
- [RFC4761] Kompella, K. and Y. Rekhter, "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, January 2007.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, June 2007.
- [RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.

- [RFC5151] Farrel, A., Ayyangar, A., and JP. Vasseur, "Inter-Domain MPLS and GMPLS Traffic Engineering -- Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 5151, February 2008.
- [RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, July 2008.

10.2. Informative References

- [I-D.gredler-rtgwg-igp-label-advertisement]
Gredler, H., Amante, S., Scholl, T., and L. Jalil,
"Advertising MPLS labels in IGPs",
draft-gredler-rtgwg-igp-label-advertisement-05 (work in
progress), May 2013.
- [I-D.ietf-rtgwg-mrt-frr-architecture]
Atlas, A., Kebler, R., Envedi, G., Csaszar, A., Tantsura,
J., Konstantynowicz, M., White, R., and M. Shand, "An
Architecture for IP/LDP Fast-Reroute Using Maximally
Redundant Trees", draft-ietf-rtgwg-mrt-frr-architecture-02
(work in progress), February 2013.
- [I-D.previdi-filsfils-isis-segment-routing]
Previdi, S., Filsfils, C., Bashandy, A., Horneffer, M.,
Decraene, B., Litkowski, S., Milojevic, I., Shakir, R.,
Ytti, S., Henderickx, W., and J. Tantsura, "Segment
Routing with IS-IS Routing Protocol",
draft-previdi-filsfils-isis-segment-routing-02 (work in
progress), March 2013.

Authors' Addresses

Hannes Gredler (editor)
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: hannes@juniper.net

Shane Amante
Level 3 Communications, Inc.
1025 Eldorado Blvd
Broomfield, CO 80021
US

Email: shane@level3.net

Tom Scholl
Amazon
Seattle, WN
US

Email: tscholl@amazon.com

Luay Jalil
Verizon
1201 E Arapaho Rd.
Richardson, TX 75081
US

Email: luay.jalil@verizon.com

