Abstract

This document describes an application of Segment Routing to scale the network to support hundreds of thousands of network nodes, and tens of millions of physical underlay endpoints. This use-case can be applied to the interconnection of massive-scale DC’s and/or large aggregation networks. Forwarding tables of midpoint and leaf nodes only require a few tens of thousands of entries.
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 May 1, 2016.

Copyright Notice

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

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

Table of Contents

1. Introduction .................................................. 4
2. Reference Design .............................................. 4
3. Control Plane ................................................ 6
4. Illustration of the scale .................................... 7
5. Optional Designs ............................................. 7
6. Deployment Model .............................................. 7
7. Benefits ..................................................... 9
8. IANA Considerations ......................................... 10
1 Introduction

This document describes how SR can be used to interconnect 100s thousands of nodes and 10’s of millions of applications/humans. This version of the document focuses on the MPLS/SR instantiation. No new protocol extensions are required.

1.1. Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP</td>
<td>Border Gateway Protocol</td>
</tr>
<tr>
<td>BGP-LS</td>
<td>Border Gateway Protocol - Link State</td>
</tr>
<tr>
<td>DC</td>
<td>Data Center</td>
</tr>
<tr>
<td>ECMP</td>
<td>Equal Cost MultiPathing</td>
</tr>
<tr>
<td>FIB</td>
<td>Forwarding Information Base</td>
</tr>
<tr>
<td>LDP</td>
<td>Label Distribution Protocol</td>
</tr>
<tr>
<td>MPLS</td>
<td>Multi-Protocol Label Switching</td>
</tr>
<tr>
<td>PCE</td>
<td>Path Computation Element</td>
</tr>
<tr>
<td>PCEP</td>
<td>Path Computation Element Protocol</td>
</tr>
<tr>
<td>PW</td>
<td>Pseudowire</td>
</tr>
<tr>
<td>SR</td>
<td>Segment Routing</td>
</tr>
<tr>
<td>SRTE</td>
<td>Segment Routing Traffic Engineering</td>
</tr>
<tr>
<td>SRTE Policy</td>
<td>Segment Routing Traffic Engineering Policy</td>
</tr>
<tr>
<td>TI-LFA</td>
<td>Topology Independent Loop Free Alternate</td>
</tr>
</tbody>
</table>

2 Reference Design

+-----------+----------+----------+
| A         | X1a      | X2a      |
| | L1       | | C       |
| B         | X1b      | X2b      |
| | L2       | | D       |
+-----------+----------+----------+

A : PrefixSID 18001 is unique in L1
B : PrefixSID 18002 is unique in L1
X1a: Anycast PrefixSID 16001 is unique across all the domains
PrefixSID 16003 is unique across all the domains
X1b: Anycast PrefixSID 16001 is unique across all the domains
PrefixSID 16004 is unique across all the domains
X2a: Anycast PrefixSID 16002 is unique across all the domains
PrefixSID 16005 is unique across all the domains
X2b: Anycast PrefixSID 16002 is unique across all the domains
PrefixSID 16006 is unique across all the domains
C : PrefixSID 18001 is unique in L2
D : PrefixSID 18002 is unique in L2

We structure the network into leaf domains (L1, L2...) interconnected by a central core domain C. Each domain runs SR with its own independent routing protocol (e.g.: IS-IS, OSPF, BGP).

A common SRGB of [16000-23999] is assumed (any other common block choice is possible) across all of the domains. We further assume that [16000-17999] is solely used to provide prefix segments in the C domain (any other choice is possible) while [18000, 23999] is reused to provide prefix segments in any leaf domain.

For example, we see that A and C of the leaf domain L1 and L2 respectively, receive the prefix segment 18001 while prefix segment 16003 is allocated to node X1a in the C domain and is unique across the entire set of domains.

Each leaf domain Lk connects to the domain C with 2 or more nodes called Xka and Xkb. Each X node runs two independent SR routing protocols: one in the leaf domain and one in the core domain. Each X nodes is provided with two prefix segments allocated from the domain C: one uniquely identifies the node while the other (anycast prefix segment) identifies the pair number k of X nodes interconnecting the leaf domain k to the core domain.

In our reference diagram, X1a has prefix segment 16003 and anycast prefix segment 16001 while X1b has prefix segment 16004 and anycast prefix segment 16001.

No route is redistributed from a leaf domain to the core domain. All the routes (and their prefix SID’s) of the X nodes are redistributed from the core domain into the leaf domains. No other route is redistributed from the core into the leaf domains. The FIB of an interior node within the C domain does not hold any entry for segments in the range [18000, 23999]. A node in a leaf domain only has FIB entries for all the segments in the local leaf domain and prefix segments towards all the X nodes in the network. For example, A of leaf L1 has a FIB entry for anycast segment 16002 which leads to the pair X2a and X2b and prefix segment 16005 which leads to X2a.

2.1 Examples

We use the notation A.L1 to represent the node A of leaf domain L1. Leveraging the above design, any leaf node can be interconnected with any other leaf node.

Intra-leaf, shortest-path: A.L1 uses the following SID list to reach
B.L1: (18002)
Inter-leaf, shortest-path through any X: A.L1 uses the following SID list to reach D.L2 via any intermediate X: (16002, 18002)
Inter-leaf, shortest-path through a specific X: A.L1 uses the following SID list to reach D.L2 via X2a: (16005, 18002)

It is out of the scope of this document to describe how the SID lists are computed and programmed at the source nodes. As an example, a centralized controller could be the source of the Prefix SID allocation. The controller could continuously collect the state of each domain (e.g. BGP-LS). Upon any new service request (e.g.: from V to W), it could check whether W is in the same leaf domain of V. If so, a single SID would be required (dynamically learned via IGP-SR (IS-IS-SR, OSPF-SR) within the domain and would not be added by the controller). Otherwise, if V and W resides on separate domains, the SID of the X gateway to W’s leaf domain would be inserted before W’s SID by the controller.

3. Control-plane

This section provides a high-level description of one example of an implemented control-plane. The example is for L2VPN PW service with certain SLA contract.

The service orchestration programs A with a PW to a remote next-hop C with a given SLA contract (low-latency path, be disjoint from a specific core plane, be disjoint from a different PW service, etc.).

A automatically detects that it does not have reachability to C. It then automatically sends a PCEP request to an SR PCE for an SRTE policy that provides reachability to C with the requested SLA.

The SR PCE is made of two components. A multi-domain topology and a compute block. The multi-domain topology is continuously refreshed from BGP-LS feeds from each domain. The compute block implements TE algorithms designed specifically for SR path expression. Upon receiving the PCEP request, the SR PCE computes the solution (e.g. (16002, 18001)) and provides it to A.

The SR PCE logs the request as a stateful query and hence recomputes another solution upon any multi-domain topology changes that invalidates the previous solution.

A receives the PCEP reply with the solution. A installs the received SRTE policy in the dataplane. A automatically steers the PW on that SRTE policy.
It is out of the scope of this document to describe how the SRTE Policies are computed and programmed at the source nodes.

4. Illustration of the scale

A review of a practical design use case can be used to determine the true scalability of this methodology in the real world. Such an example will demonstrate that over 50 million physical end points can be supported in as little as 12K of FIB space. Moreover, this example does not even suggest a maximum scale. Please refer to [Appendix] for details.

5 Optional Designs

Section 2 describes the reference model of the design. However, there could be multiple different design options depends on the network scale, network node hardware capability, SLA requirement, etc.

5.1 Leaf and Core Domains Sizing

The operator might choose to not redistribute the X routes into the leaf domains. In that case, one more segment is required in order to compose an end-to-end path. For example, to express an "inter-leaf, shortest-path through any X" path from A.L1 to D.L2, A.L1 uses {16001, 16002, 18002} instead of {16002, 18002}. This model gives the operator the ability to choose among a small number of larger leaf domains, a large number of small leaf domains or a mix of small and large domains.

5.2 Local Segments to Hosts/Servers

Local segments can be programmed at any leaf node in order to identify locally-attached hosts (or VM’s). For example, if D.L2 has bound a local segment 40001 to a local host DH1, then A uses the following SID list to reach that host: {16002, 18002, 40001} (assuming the reference design above). Such local segment could represent the NID (Network Interface Device) device in the context of the SP access network, or VM in the context of the DC network.

5.3 Sub-leaf Domains

A third level of hierarchy called "Sub-Leaf" can be introduced for further scale.

```
+---------+ +---------+ +---------+ +-------+
|   L1    | |    C    | |    L2   | | SL21  |
B         X1b         X2b         Y21b      F
+---------+ +---------+ +---------+ +-------+
```

In the above diagram, a sub-leaf "SL21" has been added to the leaf domain L2. SL21 is connected to L2 via two (or more) Y nodes. The SRGB sub-space [18000, 23999] initially allocated for the leaf is split into two sub-spaces: [18000-19999] for the leaf allocation and [20000-23999] for the sub-leaf allocation. Each Y node is allocated with a unique anycast prefix segment and a unique prefix segment within the leaf block. For example, Y21a receives anycast SID 19021 and prefix SID 19211. Each node within a subleaf domain receives a unique prefix SID from that domain (e.g. E receives 20001).

For example, to express an "inter-leaf, shortest-path, through any X, through any Y" path to E.L2.SL21, A.L1 uses {16002, 19021, 20001}.

Alternatively, the operator may decide not to distribute any X route down into leaf domains, but instead, distribute Y gateways up to the C domain. In this case, A.L1 would express the "inter-leaf, shortest-path, through any X, through any Y" path to E.L2.SL21 with SID list:{16001, 19021, 20001}.

5.4 Traffic Engineering

Traffic Engineering: Any leaf or core domain can use SR in order to traffic engineer its traffic locally within the domain. For example, a flow from A.L1 to X1a within L1 domain could be steered via B using the SR policy {18002, 16003}. Similarly a flow from X1a to X2a within the core domain could be steered via X2b with the SR policy {16006, 16005}.

Similarly, a flow can be engineered across domains. For example, a flow from A.L1 to C.L2 could be steered via B then X1a then X2b then X2a then C using the SR policy {18002, 16003, 16006, 16005, 18001}.

The SR policy at the source can be "compressed" (in terms of number of segments) by leveraging binding segments bound to SR policy. For example, assuming that the local binding segment 30000 is bound to the policy (18002, 16003) and that the local binding segment 30001 is bound by X1a to the policy (16006, 16005), then the previous inter-domain policy can also be expressed at A (or any node connected to A) as (30000, 30001, 18001). Using a binding segment to refer to a remote SR policy provides other benefits such as decreasing the need for the centralized controller in order to reflect a change from one domain to another.

For example, let us assume that something changes within the core domain such that the path followed by the policy 30001 at X1a changes. The SR policy associated with 30001 is updated at X1a without any change at A. The binding segment 30001 remains "stable" from the viewpoint of L1 leaf domain. Updating a remote domain becomes necessary only when the headend of the binding segment
becomes unavailable (X1a becomes unavailable) or when the policy attached to the binding segment is no longer achievable. An example could be: upon a double and independent failure, a policy avoiding some resources (e.g. another plane of the backbone) might no longer be possible. In only these cases, the policy at A needs to be changed. It is out of the scope of this document to describe how the SID lists are computed in order to realize a specific traffic-engineering objective, with or without the use of binding SID. For example, an application could request a specific treatment via a north-bound API to a centralized controller. The centralized controller might collect the topology of all the domains. It might also translate the application requirement into an end-to-end path through the domains. Finally, it might then translate that end-to-end path in a list of segments. It might create intermediate per-domain policies (e.g. using PCEP provisioning) and learn their associated binding segments (e.g. PCEP or BGP-LS) and return to the application the resulting SID list where some of the SID’s are binding segments.

6 Deployment Model

It is expected that this design be deployed as a greenfield but as well in interworking (brownfield) with seamless-mpls design (draft-ietf-mpls-seamless-mpls).

7 Benefit

There are some notable benefits that this proposal can bring to the large scale network deployment:

ECMP: each policy (intra or inter-domain, with or without TE) is expressed as a list of segments. As each segment is optimized for ECMP, therefore the entire policy is optimized for ECMP. The ECMP gain of anycast prefix segment should also be considered (e.g. 16001 load-shares across any gateway from L1 leaf domain to Core and 16002 load-shares across any gateway from Core to L2 leaf domain.


Simple and better node redundancy: furthermore the use of anycast segment provides for an additional high-availability mechanism (e.g.: flows directed to 16001 can either go via X1a or X1b).

No new protocol extensions are required to support this.
8. IANA Considerations

None

9. Manageability Considerations

TBD

10. Security Considerations

TBD

11. Acknowledgements

We would like to thank Giles Heron, Alexander Preusche and Steve Braaten for their contribution to the content of this document.

12. References

12.1. Normative References


12.2. Informative References


Appendix - Scale Example

A review of a practical design use case can be used to determine the true scalability of this methodology in the real world. This example does not suggest a maximum scale, but it provides a simple framework that can be deployed on emerging silicon products from a variety of sources. In doing so, it also demonstrates huge scale potential on the order of 50 million end-points.

The design

The practical design is based on the reference topology above, but
differs in that each data center layer is enumerated for addressing and the SRGB is necessarily expanded beyond the default (more on that follows). Otherwise, the topology is the same. For the sake of brevity, the design is described only at a high level here.

For terminology’s sake, each leaf domain in the original topology is shown below as a data center (DC). Further, the X nodes above are referred to here as data center routers (DCR’s). For example:

```
+---------+ +---------+ +---------+
|         DCR         DCR         |
|   DC1   | |   Core  | |   DC2   |
|         DCR         DCR         |
+---------+ +---------+ +---------+
```

The design leverages well-worn physical deployment practices (such as Clos-based fabric principles and relatively low oversubscription) in conjunction with emerging silicon options supporting 25G/50G/100G interfaces.

Each data center is composed of three layers of switching: Top-of-Rack (ToR), leaf, and spine. Sixteen ToR’s and eight leaf devices combine to create clusters supporting 1,536 physical servers each (connected at 25G). All switching devices in the cluster are based on 32x100G switches (leveraging 25G breakout where required).

The spine layer interconnects 272 clusters using a total of 128 288x100G modular switches. To leave each data center, 16 DCR devices (using 256 ports of 100G) provide connectivity to 16 core devices (also using 256 ports of 100G). From a server’s perspective, north-south oversubscription in this model is 51:1 and east-west oversubscription is 9:1.

All told, the data center (including its portion of the core) is composed of 6,688 networking devices:

- 4,352 ToR’s (16 ToR x 272 clusters)
- 2,176 Leaf nodes (8 leaf nodes x 272 clusters)
- 128 spine switches
- 16 DCR’s
- 16 Core routers

For the sake of Segment Routing scale calculations, each of these nodes will advertise its node SID. The Leaf domain FIB size is 6,672, consisting of the spine, leaf, and ToR switching layers as well as the DCR layer.

The core domain consists of the DCR’s and the Core routers (32
Finally, each data center site will advertise one anycast SID into the core domain, so the per-site SID count in the core domain is 33.

### Scalability

Each data center site has the following scale characteristics:

- 417,792 physical server ports (at 25G per port)
- 6,672 SID entries in the Leaf domain
- 33 SID entries in the Core domain

Each device in the Leaf domain must be aware of all other devices in the data center, as well as all SID’s advertised on the core. Therefore, the FIB in these devices must support:

- 7K segments for local connectivity (from the SRGB)
- <300 adjacency segments for large switches (locally installed)
- Up to 104 adjacency segments on small switches (locally installed)
- 4K Core segments (from the SRGB, to be covered shortly)

This represents a total FIB requirement of about 12K (rounded up). Most of this addressing involves global addressing. This means that the SRGB must be expanded beyond its 8K default configuration (values 16000 - 23999). For this design, the SRGB is expanded to 12K (16000 - 27999) and apportioned in the following way:

- 16000 - 19999 (4K) Core and inter-site connectivity (truly global)
- 20000 - 27999 (8K) Leaf domain (reused per DC site)

The core domain size of 4K is somewhat arbitrary and was chosen to demonstrate huge scale without requiring a large FIB. Assuming every data center site consumes 33 SID’s in the Core domain space, a 4K Core domain results in the following system scalability:

- 120 data center sites (4K / 33)
- 50M servers at 25G (417,792 * 120)

It should be appreciated that this example covers infrastructure only. Further, design options are too numerous to describe succinctly. Therefore, subjects such as an application-based addressing architecture, multi-tenancy, service chaining, and so forth are not accounted for. Nonetheless, deploying 50M endpoints using a maximum FIB of 12K should generally provide super-granular traffic steering and SDN services on low-cost hardware.

### Authors’ Addresses
Luay Jalil  
Verizon  
400 International Pkwy  
Richardson, TX 75081  
Email: luay.jalil@verizon.com

Jeff Tantsura  
Ericsson  
jeff.tantsura@ericsson.com