Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 1, 2016

C. Filsfils, Ed.
D. Cai, Ed.
S. Previdi
Cisco

W. Henderickx
Alcatel-Lucent

R. Shakir
BT

D. Cooper
F. Ferguson
Level3

T. LaBerge
S. Lin
Microsoft

B. Decraene
Orange

L. Jalil
Verizon

J. Tantsura
Ericsson

November 1, 2015

Interconnecting Millions Of Endpoints With Segment Routing
draft-filsfils-spring-large-scale-interconnect-01

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.

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 will expire on May 1, 2016.

Copyright Notice

Table of Contents

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.

2 Reference Design

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

2.2 Scale Examples

1 core domain and 100 leaf domains

Core domain has 200 core nodes. Assume two nodes per each leaf domain, with specific node segment and anycast segments, it's 300 prefix segments in total.
Assume a core node connects only one leaf domain.

Each leaf domain has 6,000 leaf node segments.
Each leaf-node has 500 endpoints attached, thus 500 adjacency segments.
In total, it is 3M endpoints per leaf domain.

Network wide scale:
6,000x100=600,000 nodes
6,000x100x500=300M endpoints

Per-node segment scale:
Leaf node segment scale: 6,000 (leaf node segments) + 300 (core node segments) + 500 (adj segments) = 6,800
Core node segment scale: 6,000 (leaf domain segments) + 300 (core domain segments) = 6,300

In the above calculation, it didn't count the link adjacency segments, which is local to the node. Typically it should be <100.

Note, depends on the leaf node FIB capability, we could split the leaf domain into multiple smaller domains. For the above example, we can split the leaf domain to 6 smaller leaf domains. So each leaf node only need to learn 1000 (leaf node segments) + 300 (core node segments) + 500 (adj segments)= 1,800 segments.

3 Optional Designs

3.1 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.

3.2 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.

3.3 Sub-leaf Domains

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

```
     +---------+ +---------+ +---------+ +--------+
    A         X1a         X2a         Y21a      E
    |   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 splitted 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 segmemt 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}.

3.4 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 by A 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.

4 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).

5 Benefit

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.

Sub-50msec FRR: Topology-Independent FRR using SR [draft-francois-spring-segment-routing-ti-lfa-01] ensures sub-50msec upon any link or node failure, in any topology.

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.

6. IANA Considerations

None

7. Manageability Considerations

TBD

8. Security Considerations

TBD

9. Acknowledgements

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

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.

10.2. Informative References

[draft-ietf-mpls-seamless-mpls] Leymann, et al., "Seamless MPLS Architecture", draft-ietf-mpls-seamless-mpls-07, (work in progress), July 2015

[draft-francois-spring-segment-routing-ti-lfa-01] Pierre Francois, et al., "Topology Independent Fast Reroute using Segment Routing", draft-francois-spring-segment-routing-ti-lfa-01,

(work in progress), April 2015

Authors' Addresses

Clarence Filsfils (editor)
Cisco Systems, Inc.
Brussels
BE
Email: cfilsfil@cisco.com

Dennis Cai (editor)
Cisco Systems, Inc.
170, West Tasman Drive
San Jose, CA 95134
US
Email: dcai@cisco.com

Stefano Previdi
Cisco Systems, Inc.
Via Del Serafico, 200
Rome 00142
Italy
Email: sprevidi@cisco.com

Wim Henderickx
Alcatel-Lucent
Email: wim.henderickx@alcatel-lucent.com

Rob Shakir
BT
Email: rob.shakir@bt.com

Dave Cooper
Level 3
Email: Dave.Cooper@Level3.com

Francis Ferguson
Level 3
Email: Francis.Ferguson@level3.com

Tim LaBerge
Microsoft
Email: Tim.LaBerge@microsoft.com

Steven Lin
Microsoft
Email: slin@microsoft.com

Bruno Decraene
Orange
Email: bruno.decraene@orange.com

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

Jeff Tantsura
Ericsson
jeff.tantsura@ericsson.com