Internet Engineering Task Force (IETF)
Request for Comments: 8670
Category: Informational
ISSN: 2070-1721
C. Filsfils, Ed.
S. Previdi
Cisco Systems, Inc.
G. Dawra
LinkedIn
E. Aries
Arrcus, Inc.
P. Lapukhov
Facebook
December 2019

BGP Prefix Segment in Large-Scale Data Centers

Abstract

This document describes the motivation for, and benefits of, applying Segment Routing (SR) in BGP-based large-scale data centers. It describes the design to deploy SR in those data centers for both the MPLS and IPv6 data planes.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Segment Routing (SR), as described in [RFC8402], leverages the source-routing paradigm. A node steers a packet through an ordered list of instructions called "segments". A segment can represent any instruction, topological or service based. A segment can have a local semantic to an SR node or a global semantic within an SR domain. SR allows the enforcement of a flow through any topological path while maintaining per-flow state only from the ingress node to the SR domain. SR can be applied to the MPLS and IPv6 data planes.

The use cases described in this document should be considered in the context of the BGP-based large-scale data-center (DC) design described in [RFC7938]. This document extends it by applying SR both with IPv6 and MPLS data planes.

2. Large-Scale Data-Center Network Design Summary

This section provides a brief summary of the Informational RFC [RFC7938], which outlines a practical network design suitable for data centers of various scales:

* Data-center networks have highly symmetric topologies with multiple parallel paths between two server-attachment points. The well-known Clos topology is most popular among the operators (as described in [RFC7938]). In a Clos topology, the minimum number of parallel paths between two elements is determined by the

"width" of the "Tier-1" stage. See Figure 1 for an illustration of the concept.

* Large-scale data centers commonly use a routing protocol, such as BGP-4 [RFC4271], in order to provide endpoint connectivity. Therefore, recovery after a network failure is driven either by local knowledge of directly available backup paths or by distributed signaling between the network devices.

* Within data-center networks, traffic is load shared using the Equal Cost Multipath (ECMP) mechanism. With ECMP, every network device implements a pseudorandom decision, mapping packets to one of the parallel paths by means of a hash function calculated over certain parts of the packet, typically a combination of various packet header fields.

The following is a schematic of a five-stage Clos topology with four devices in the "Tier-1" stage. Notice that the number of paths between Node1 and Node12 equals four; the paths have to cross all of the Tier-1 devices. At the same time, the number of paths between Node1 and Node2 equals two, and the paths only cross Tier-2 devices. Other topologies are possible, but for simplicity, only the topologies that have a single path from Tier-1 to Tier-3 are considered below. The rest could be treated similarly, with a few modifications to the logic.

2.1. Reference Design

```
                                   Tier-1
                                  +-----+
                                  |NODE |
                               +->|  5  |--+
                               |  +-----+  |
                       Tier-2  |           |   Tier-2
                      +-----+  |  +-----+  |  +-----+
        +------------>|NODE |--+->|NODE |--+--|NODE |-------------+
        |       +-----|  3  |--+  |  6  |  +--|  9  |-----+       |
        |       |     +-----+     +-----+     +-----+     |       |
        |       |                                         |       |
        |       |     +-----+     +-----+     +-----+     |       |
        | +-----+---->|NODE |--+  |NODE |  +--|NODE |-----+-----+ |
        | |     | +---|  4  |--+->|  7  |--+--|  10 |---+ |     | |
        | |     | |   +-----+  |  +-----+  |  +-----+   | |     | |
      +-----+ +-----+          |  +-----+  |          +-----+ +-----+
      |NODE | |NODE | Tier-3   +->|NODE |--+   Tier-3 |NODE | |NODE |
      |  1  | |  2  |             |  8  |             | 11  | |  12 |
      +-----+ +-----+             +-----+             +-----+ +-----+
        | |     | |                                     | |     | |
        A O     B O            <- Servers ->            Z O     O O
```

Figure 1: 5-Stage Clos Topology

In the reference topology illustrated in Figure 1, it is assumed:

* Each node is its own autonomous system (AS) (Node X has AS X).

4-byte AS numbers are recommended ([RFC6793]).

- For simple and efficient route propagation filtering, Node5, Node6, Node7, and Node8 use the same AS; Node3 and Node4 use the same AS; and Node9 and Node10 use the same AS.

- In the case in which 2-byte autonomous system numbers are used for efficient usage of the scarce 2-byte Private Use AS pool, different Tier-3 nodes might use the same AS.

- Without loss of generality, these details will be simplified in this document. It is to be assumed that each node has its own AS.

* Each node peers with its neighbors with a BGP session. If not specified, external BGP (EBGP) is assumed. In a specific use case, internal BGP (IBGP) will be used, but this will be called out explicitly in that case.

* Each node originates the IPv4 address of its loopback interface into BGP and announces it to its neighbors.

  - The loopback of Node X is 192.0.2.x/32.

In this document, the Tier-1, Tier-2, and Tier-3 nodes are referred to as "Spine", "Leaf", and "ToR" (top of rack) nodes, respectively. When a ToR node acts as a gateway to the "outside world", it is referred to as a "border node".

3. Some Open Problems in Large Data-Center Networks

The data-center-network design summarized above provides means for moving traffic between hosts with reasonable efficiency. There are few open performance and reliability problems that arise in such a design:

* ECMP routing is most commonly realized per flow. This means that large, long-lived "elephant" flows may affect performance of smaller, short-lived "mouse" flows and may reduce efficiency of per-flow load sharing. In other words, per-flow ECMP does not perform efficiently when flow-lifetime distribution is heavy tailed. Furthermore, due to hash-function inefficiencies, it is possible to have frequent flow collisions where more flows get placed on one path over the others.

* Shortest-path routing with ECMP implements an oblivious routing model that is not aware of the network imbalances. If the network symmetry is broken, for example, due to link failures, utilization hotspots may appear. For example, if a link fails between Tier-1 and Tier-2 devices (e.g., Node5 and Node9), Tier-3 devices Node1 and Node2 will not be aware of that since there are other paths available from the perspective of Node3. They will continue sending roughly equal traffic to Node3 and Node4 as if the failure didn't exist, which may cause a traffic hotspot.

* Isolating faults in the network with multiple parallel paths and

ECMP-based routing is nontrivial due to lack of determinism. Specifically, the connections from HostA to HostB may take a different path every time a new connection is formed, thus making consistent reproduction of a failure much more difficult. This complexity scales linearly with the number of parallel paths in the network and stems from the random nature of path selection by the network devices.

4. Applying Segment Routing in the DC with MPLS Data Plane

4.1. BGP Prefix Segment (BGP Prefix-SID)

A BGP Prefix Segment is a segment associated with a BGP prefix. A BGP Prefix Segment is a network-wide instruction to forward the packet along the ECMP-aware best path to the related prefix.

The BGP Prefix Segment is defined as the BGP Prefix-SID Attribute in [RFC8669], which contains an index. Throughout this document, the BGP Prefix Segment Attribute is referred to as the "BGP Prefix-SID" and the encoded index as the label index.

In this document, the network design decision has been made to assume that all the nodes are allocated the same SRGB (Segment Routing Global Block), e.g., [16000, 23999]. This provides operational simplification as explained in Section 8, but this is not a requirement.

For illustration purposes, when considering an MPLS data plane, it is assumed that the label index allocated to prefix 192.0.2.x/32 is X. As a result, a local label (16000+x) is allocated for prefix 192.0.2.x/32 by each node throughout the DC fabric.

When the IPv6 data plane is considered, it is assumed that Node X is allocated IPv6 address (segment) 2001:DB8::X.

4.2. EBGP Labeled Unicast (RFC 8277)

Referring to Figure 1 and [RFC7938], the following design modifications are introduced:

* Each node peers with its neighbors via an EBGP session with extensions defined in [RFC8277] (named "EBGP8277" throughout this document) and with the BGP Prefix-SID attribute extension as defined in [RFC8669].

* The forwarding plane at Tier-2 and Tier-1 is MPLS.

* The forwarding plane at Tier-3 is either IP2MPLS (if the host sends IP traffic) or MPLS2MPLS (if the host sends MPLS-encapsulated traffic).

Figure 2 zooms into a path from ServerA to ServerZ within the topology of Figure 1.

```
                  +-----+     +-----+     +-----+
       +---------->|NODE |     |NODE |     |NODE |
```

```
          |           |  4  |--+->|  7  |--+--|  10 |---+
          |           +-----+     +-----+     +-----+   |
          |                                             |
      +-----+                                         +-----+
      |NODE |                                         |NODE |
      |  1  |                                         | 11  |
      +-----+                                         +-----+
        |                                              |
        A                   <- Servers ->              Z
```

Figure 2: Path from A to Z via Nodes 1, 4, 7, 10, and 11

Referring to Figures 1 and 2, and assuming the IP address with the AS and label-index allocation previously described, the following sections detail the control-plane operation and the data-plane states for the prefix 192.0.2.11/32 (loopback of Node11).

4.2.1. Control Plane

Node11 originates 192.0.2.11/32 in BGP and allocates to it a BGP Prefix-SID with label-index: index11 [RFC8669].

Node11 sends the following EBGP8277 update to Node10:

IP Prefix: 192.0.2.11/32

Label: Implicit NULL

Next hop: Node11's interface address on the link to Node10

AS Path: {11}

BGP Prefix-SID: Label-Index 11

Node10 receives the above update. As it is SR capable, Node10 is able to interpret the BGP Prefix-SID; therefore, it understands that it should allocate the label from its own SRGB block, offset by the label index received in the BGP Prefix-SID (16000+11, hence, 16011) to the Network Layer Reachability Information (NLRI) instead of allocating a nondeterministic label out of a dynamically allocated portion of the local label space. The implicit NULL label in the NLRI tells Node10 that it is the penultimate hop and that it must pop the top label on the stack before forwarding traffic for this prefix to Node11.

Then, Node10 sends the following EBGP8277 update to Node7:

IP Prefix: 192.0.2.11/32

Label: 16011

Next hop: Node10's interface address on the link to Node7

AS Path: {10, 11}

BGP Prefix-SID: Label-Index 11

Node7 receives the above update. As it is SR capable, Node7 is able to interpret the BGP Prefix-SID; therefore, it allocates the local (incoming) label 16011 (16000 + 11) to the NLRI (instead of allocating a "dynamic" local label from its label manager). Node7 uses the label in the received EBGP8277 NLRI as the outgoing label (the index is only used to derive the local/incoming label).

Node7 sends the following EBGP8277 update to Node4:

IP Prefix: 192.0.2.11/32

Label: 16011

Next hop: Node7's interface address on the link to Node4

AS Path: {7, 10, 11}

BGP Prefix-SID: Label-Index 11

Node4 receives the above update. As it is SR capable, Node4 is able to interpret the BGP Prefix-SID; therefore, it allocates the local (incoming) label 16011 to the NLRI (instead of allocating a "dynamic" local label from its label manager). Node4 uses the label in the received EBGP8277 NLRI as an outgoing label (the index is only used to derive the local/incoming label).

Node4 sends the following EBGP8277 update to Node1:

IP Prefix: 192.0.2.11/32

Label: 16011

Next hop: Node4's interface address on the link to Node1

AS Path: {4, 7, 10, 11}

BGP Prefix-SID: Label-Index 11

Node1 receives the above update. As it is SR capable, Node1 is able to interpret the BGP Prefix-SID; therefore, it allocates the local (incoming) label 16011 to the NLRI (instead of allocating a "dynamic" local label from its label manager). Node1 uses the label in the received EBGP8277 NLRI as an outgoing label (the index is only used to derive the local/incoming label).

4.2.2. Data Plane

Referring to Figure 1, and assuming all nodes apply the same advertisement rules described above and all nodes have the same SRGB (16000-23999), here are the IP/MPLS forwarding tables for prefix 192.0.2.11/32 at Node1, Node4, Node7, and Node10.

Incoming Label or IP Destination	Outgoing Label	Outgoing Interface

16011	16011	ECMP{3, 4}
192.0.2.11/32	16011	ECMP{3, 4}

Table 1: Node1 Forwarding Table

Incoming Label or IP Destination	Outgoing Label	Outgoing Interface
16011	16011	ECMP{7, 8}
192.0.2.11/32	16011	ECMP{7, 8}

Table 2: Node4 Forwarding Table

Incoming Label or IP Destination	Outgoing Label	Outgoing Interface
16011	16011	10
192.0.2.11/32	16011	10

Table 3: Node7 Forwarding Table

Incoming Label or IP Destination	Outgoing Label	Outgoing Interface
16011	POP	11
192.0.2.11/32	N/A	11

Table 4: Node10 Forwarding Table

4.2.3. Network Design Variation

A network design choice could consist of switching all the traffic through Tier-1 and Tier-2 as MPLS traffic. In this case, one could filter away the IP entries at Node4, Node7, and Node10. This might be beneficial in order to optimize the forwarding table size.

A network design choice could consist of allowing the hosts to send MPLS-encapsulated traffic based on the Egress Peer Engineering (EPE) use case as defined in [SR-CENTRAL-EPE]. For example, applications at HostA would send their Z-destined traffic to Node1 with an MPLS label stack where the top label is 16011 and the next label is an EPE peer segment ([SR-CENTRAL-EPE]) at Node11 directing the traffic to Z.

4.2.4. Global BGP Prefix Segment through the Fabric

When the previous design is deployed, the operator enjoys global BGP Prefix-SID and label allocation throughout the DC fabric.

A few examples follow:

* Normal forwarding to Node11: A packet with top label 16011 received by any node in the fabric will be forwarded along the ECMP-aware BGP best path towards Node11, and the label 16011 is penultimate popped at Node10 (or at Node 9).

* Traffic-engineered path to Node11: An application on a host behind Node1 might want to restrict its traffic to paths via the Spine node Node5. The application achieves this by sending its packets with a label stack of {16005, 16011}. BGP Prefix-SID 16005 directs the packet up to Node5 along the path (Node1, Node3, Node5). BGP Prefix-SID 16011 then directs the packet down to Node11 along the path (Node5, Node9, Node11).

4.2.5. Incremental Deployments

The design previously described can be deployed incrementally. Let us assume that Node7 does not support the BGP Prefix-SID, and let us show how the fabric connectivity is preserved.

From a signaling viewpoint, nothing would change; even though Node7 does not support the BGP Prefix-SID, it does propagate the attribute unmodified to its neighbors.

From a label-allocation viewpoint, the only difference is that Node7 would allocate a dynamic (random) label to the prefix 192.0.2.11/32 (e.g., 123456) instead of the "hinted" label as instructed by the BGP Prefix-SID. The neighbors of Node7 adapt automatically as they always use the label in the BGP8277 NLRI as an outgoing label.

Node4 does understand the BGP Prefix-SID; therefore, it allocates the indexed label in the SRGB (16011) for 192.0.2.11/32.

As a result, all the data-plane entries across the network would be unchanged except the entries at Node7 and its neighbor Node4 as shown in the figures below.

The key point is that the end-to-end Label Switched Path (LSP) is preserved because the outgoing label is always derived from the received label within the BGP8277 NLRI. The index in the BGP Prefix-SID is only used as a hint on how to allocate the local label (the incoming label) but never for the outgoing label.

Incoming Label or IP Destination	Outgoing Label	Outgoing Interface
12345	16011	10

Table 5: Node7 Forwarding Table

Incoming Label or IP Destination	Outgoing Label	Outgoing Interface
16011	12345	7

Table 6: Node4 Forwarding Table

The BGP Prefix-SID can thus be deployed incrementally, i.e., one node at a time.

When deployed together with a homogeneous SRGB (the same SRGB across the fabric), the operator incrementally enjoys the global prefix segment benefits as the deployment progresses through the fabric.

4.3. IBGP Labeled Unicast (RFC 8277)

The same exact design as EBGP8277 is used with the following modifications:

* All nodes use the same AS number.

* Each node peers with its neighbors via an internal BGP session (IBGP) with extensions defined in [RFC8277] (named "IBGP8277" throughout this document).

* Each node acts as a route reflector for each of its neighbors and with the next-hop-self option. Next-hop-self is a well-known operational feature that consists of rewriting the next hop of a BGP update prior to sending it to the neighbor. Usually, it's a common practice to apply next-hop-self behavior towards IBGP peers for EBGP-learned routes. In the case outlined in this section, it is proposed to use the next-hop-self mechanism also to IBGP-learned routes.

```
                          Cluster-1
                       +-----------+
                       |  Tier-1   |
                       |  +-----+  |
                       |  |NODE |  |
                       |  |  5  |  |
             Cluster-2 |  +-----+  | Cluster-3
            +---------+|           |+---------+
            | Tier-2  ||           ||  Tier-2 |
            | +-----+ ||  +-----+  || +-----+ |
            | |NODE | ||  |NODE |  || |NODE | |
            | |  3  | ||  |  6  |  || |  9  | |
            | +-----+ ||  +-----+  || +-----+ |
            |         ||           ||         |
            |         ||           ||         |
            | +-----+ ||  +-----+  || +-----+ |
            | |NODE | ||  |NODE |  || |NODE | |
            | |  4  | ||  |  7  |  || |  10 | |
            | +-----+ ||  +-----+  || +-----+ |
```

```
                   +---------+ |           | +---------+
                               |           |
                               |  +-----+  |
                               |  |NODE |  |
             Tier-3            |  |  8  |  |      Tier-3
         +-----+ +-----+       |  +-----+  |     +-----+ +-----+
         |NODE | |NODE |       +-----------+     |NODE | |NODE |
         |  1  | |  2  |                         | 11  | |  12 |
         +-----+ +-----+                         +-----+ +-----+
```

Figure 3: IBGP Sessions with Reflection and Next-Hop-Self

* For simple and efficient route propagation filtering and as illustrated in Figure 3:

  - Node5, Node6, Node7, and Node8 use the same Cluster ID (Cluster-1).

  - Node3 and Node4 use the same Cluster ID (Cluster-2).

  - Node9 and Node10 use the same Cluster ID (Cluster-3).

* The control-plane behavior is mostly the same as described in the previous section; the only difference is that the EBGP8277 path propagation is simply replaced by an IBGP8277 path reflection with next hop changed to self.

* The data-plane tables are exactly the same.

5. Applying Segment Routing in the DC with IPv6 Data Plane

The design described in [RFC7938] is reused with one single modification. It is highlighted using the example of the reachability to Node11 via Spine node Node5.

Node5 originates 2001:DB8::5/128 with the attached BGP Prefix-SID for IPv6 packets destined to segment 2001:DB8::5 ([RFC8402]).

Node11 originates 2001:DB8::11/128 with the attached BGP Prefix-SID advertising the support of the Segment Routing Header (SRH) for IPv6 packets destined to segment 2001:DB8::11.

The control-plane and data-plane processing of all the other nodes in the fabric is unchanged. Specifically, the routes to 2001:DB8::5 and 2001:DB8::11 are installed in the FIB along the EBGP best path to Node5 (Spine node) and Node11 (ToR node) respectively.

An application on HostA that needs to send traffic to HostZ via only Node5 (Spine node) can do so by sending IPv6 packets with a Segment Routing Header (SRH, [IPv6-SRH]). The destination address and active segment is set to 2001:DB8::5. The next and last segment is set to 2001:DB8::11.

The application must only use IPv6 addresses that have been advertised as capable for SRv6 segment processing (e.g., for which the BGP Prefix Segment capability has been advertised). How applications learn this (e.g., centralized controller and orchestration) is outside the scope of this document.

6. Communicating Path Information to the Host

There are two general methods for communicating path information to the end-hosts: "proactive" and "reactive", aka "push" and "pull" models. There are multiple ways to implement either of these methods. Here, it is noted that one way could be using a centralized controller: the controller either tells the hosts of the prefix-to-path mappings beforehand and updates them as needed (network event driven push) or responds to the hosts making requests for a path to a specific destination (host event driven pull). It is also possible to use a hybrid model, i.e., pushing some state from the controller in response to particular network events, while the host pulls other state on demand.

Note also that when disseminating network-related data to the end-hosts, a trade-off is made to balance the amount of information vs. the level of visibility in the network state. This applies to both push and pull models. In the extreme case, the host would request path information on every flow and keep no local state at all. On the other end of the spectrum, information for every prefix in the network along with available paths could be pushed and continuously updated on all hosts.

7. Additional Benefits

7.1. MPLS Data Plane with Operational Simplicity

As required by [RFC7938], no new signaling protocol is introduced. The BGP Prefix-SID is a lightweight extension to BGP Labeled Unicast [RFC8277]. It applies either to EBGP- or IBGP-based designs.

Specifically, LDP and RSVP-TE are not used. These protocols would drastically impact the operational complexity of the data center and would not scale. This is in line with the requirements expressed in [RFC7938].

Provided the same SRGB is configured on all nodes, all nodes use the same MPLS label for a given IP prefix. This is simpler from an operation standpoint, as discussed in Section 8.

7.2. Minimizing the FIB Table

The designer may decide to switch all the traffic at Tier-1 and Tier-2 based on MPLS, thereby drastically decreasing the IP table size at these nodes.

This is easily accomplished by encapsulating the traffic either directly at the host or at the source ToR node. The encapsulation is done by pushing the BGP Prefix-SID of the destination ToR for intra-DC traffic, or by pushing the BGP Prefix-SID for the border node for inter-DC or DC-to-outside-world traffic.

7.3. Egress Peer Engineering

It is straightforward to combine the design illustrated in this document with the Egress Peer Engineering (EPE) use case described in [SR-CENTRAL-EPE].

In such a case, the operator is able to engineer its outbound traffic on a per-host-flow basis, without incurring any additional state at intermediate points in the DC fabric.

For example, the controller only needs to inject a per-flow state on the HostA to force it to send its traffic destined to a specific Internet destination D via a selected border node (say Node12 in Figure 1 instead of another border node, Node11) and a specific egress peer of Node12 (say peer AS 9999 of local PeerNode segment 9999 at Node12 instead of any other peer that provides a path to the destination D). Any packet matching this state at HostA would be encapsulated with SR segment list (label stack) {16012, 9999}. 16012 would steer the flow through the DC fabric, leveraging any ECMP, along the best path to border node Node12. Once the flow gets to border node Node12, the active segment is 9999 (because of Penultimate Hop Popping (PHP) on the upstream neighbor of Node12). This EPE PeerNode segment forces border node Node12 to forward the packet to peer AS 9999 without any IP lookup at the border node. There is no per-flow state for this engineered flow in the DC fabric. A benefit of SR is that the per-flow state is only required at the source.

As well as allowing full traffic-engineering control, such a design also offers FIB table-minimization benefits as the Internet-scale FIB at border node Node12 is not required if all FIB lookups are avoided there by using EPE.

7.4. Anycast

The design presented in this document preserves the availability and load-balancing properties of the base design presented in [RFC8402].

For example, one could assign an anycast loopback 192.0.2.20/32 and associate segment index 20 to it on the border nodes Node11 and Node12 (in addition to their node-specific loopbacks). Doing so, the EPE controller could express a default "go-to-the-Internet via any border node" policy as segment list {16020}. Indeed, from any host in the DC fabric or from any ToR node, 16020 steers the packet towards the border nodes Node11 or Node12 leveraging ECMP where available along the best paths to these nodes.

8. Preferred SRGB Allocation

In the MPLS case, it is recommended to use the same SRGBs at each node.

Different SRGBs in each node likely increase the complexity of the solution both from an operational viewpoint and from a controller viewpoint.

From an operational viewpoint, it is much simpler to have the same global label at every node for the same destination (the MPLS troubleshooting is then similar to the IPv6 troubleshooting where this global property is a given).

From a controller viewpoint, this allows us to construct simple policies applicable across the fabric.

Let us consider two applications, A and B, respectively connected to Node1 and Node2 (ToR nodes). Application A has two flows, FA1 and FA2, destined to Z. B has two flows, FB1 and FB2, destined to Z. The controller wants FA1 and FB1 to be load shared across the fabric while FA2 and FB2 must be respectively steered via Node5 and Node8.

Assuming a consistent unique SRGB across the fabric as described in this document, the controller can simply do it by instructing A and B to use {16011} respectively for FA1 and FB1 and by instructing A and B to use {16005 16011} and {16008 16011} respectively for FA2 and FB2.

Let us assume a design where the SRGB is different at every node and where the SRGB of each node is advertised using the Originator SRGB TLV of the BGP Prefix-SID as defined in [RFC8669]: SRGB of Node K starts at value K*1000, and the SRGB length is 1000 (e.g., Node1's SRGB is [1000, 1999], Node2's SRGB is [2000, 2999], ...).

In this case, the controller would need to collect and store all of these different SRGBs (e.g., through the Originator SRGB TLV of the BGP Prefix-SID); furthermore, it would also need to adapt the policy for each host. Indeed, the controller would instruct A to use {1011} for FA1 while it would have to instruct B to use {2011} for FB1 (while with the same SRGB, both policies are the same {16011}).

Even worse, the controller would instruct A to use {1005, 5011} for FA1 while it would instruct B to use {2011, 8011} for FB1 (while with the same SRGB, the second segment is the same across both policies: 16011). When combining segments to create a policy, one needs to carefully update the label of each segment. This is obviously more error prone, more complex, and more difficult to troubleshoot.

9. IANA Considerations

This document has no IANA actions.

10. Manageability Considerations

The design and deployment guidelines described in this document are based on the network design described in [RFC7938].

The deployment model assumed in this document is based on a single domain where the interconnected DCs are part of the same administrative domain (which, of course, is split into different autonomous systems). The operator has full control of the whole domain, and the usual operational and management mechanisms and procedures are used in order to prevent any information related to internal prefixes and topology to be leaked outside the domain.

As recommended in [RFC8402], the same SRGB should be allocated in all nodes in order to facilitate the design, deployment, and operations of the domain.

When EPE ([SR-CENTRAL-EPE]) is used (as explained in Section 7.3), the same operational model is assumed. EPE information is originated and propagated throughout the domain towards an internal server, and unless explicitly configured by the operator, no EPE information is leaked outside the domain boundaries.

11. Security Considerations

This document proposes to apply SR to a well-known scalability requirement expressed in [RFC7938] using the BGP Prefix-SID as defined in [RFC8669].

It has to be noted, as described in Section 10, that the design illustrated in [RFC7938] and in this document refer to a deployment model where all nodes are under the same administration. In this context, it is assumed that the operator doesn't want to leak outside of the domain any information related to internal prefixes and topology. The internal information includes Prefix-SID and EPE information. In order to prevent such leaking, the standard BGP mechanisms (filters) are applied on the boundary of the domain.

Therefore, the solution proposed in this document does not introduce any additional security concerns from what is expressed in [RFC7938] and [RFC8669]. It is assumed that the security and confidentiality of the prefix and topology information is preserved by outbound filters at each peering point of the domain as described in Section 10.

12. References

12.1. Normative References

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC7938] Lapukhov, P., Premji, A., and J. Mitchell, Ed., "Use of BGP for Routing in Large-Scale Data Centers", RFC 7938, DOI 10.17487/RFC7938, August 2016, <https://www.rfc-editor.org/info/rfc7938>.

[RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017, <https://www.rfc-editor.org/info/rfc8277>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[RFC8669] Previdi, S., Filsfils, C., Lindem, A., Ed., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix Segment Identifier Extensions for BGP", RFC 8669, DOI 10.17487/RFC8669, December 2019, <https://www.rfc-editor.org/info/rfc8669>.

12.2. Informative References

[IPv6-SRH] Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", Work in Progress, Internet-Draft, draft-ietf-6man-segment-routing-header-26, 22 October 2019, <https://tools.ietf.org/html/draft-ietf-6man-segment-routing-header-26>.

[RFC6793] Vohra, Q. and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", RFC 6793, DOI 10.17487/RFC6793, December 2012, <https://www.rfc-editor.org/info/rfc6793>.

[SR-CENTRAL-EPE] Filsfils, C., Previdi, S., Dawra, G., Aries, E., and D. Afanasiev, "Segment Routing Centralized BGP Egress Peer Engineering", Work in Progress, Internet-Draft, draft-ietf-spring-segment-routing-central-epe-10, 21 December 2017, <https://tools.ietf.org/html/draft-ietf-spring-segment-routing-central-epe-10>.

Acknowledgements

The authors would like to thank Benjamin Black, Arjun Sreekantiah, Keyur Patel, Acee Lindem, and Anoop Ghanwani for their comments and review of this document.

Contributors

Gaya Nagarajan
Facebook
United States of America

Email: gaya@fb.com

Gaurav Dawra
Cisco Systems
United States of America

Email: gdawra.ietf@gmail.com

Dmitry Afanasiev
Yandex
Russian Federation

Email: fl0w@yandex-team.ru

Tim Laberge
Cisco
United States of America

Email: tlaberge@cisco.com

Edet Nkposong
Salesforce.com Inc.
United States of America

Email: enkposong@salesforce.com

Mohan Nanduri
Microsoft
United States of America

Email: mohan.nanduri@oracle.com

James Uttaro
ATT
United States of America

Email: ju1738@att.com

Saikat Ray
Unaffiliated
United States of America

Email: raysaikat@gmail.com

Jon Mitchell
Unaffiliated
United States of America

Email: jrmitche@puck.nether.net

Authors' Addresses

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

Email: cfilsfil@cisco.com

Stefano Previdi
Cisco Systems, Inc.
Italy

Email: stefano@previdi.net

Gaurav Dawra
LinkedIn
United States of America

Email: gdawra.ietf@gmail.com

Ebben Aries
Arrcus, Inc.
2077 Gateway Place, Suite #400
San Jose, CA 95119
United States of America

Email: exa@arrcus.com

Petr Lapukhov
Facebook
United States of America

Email: petr@fb.com