SPRING Working Group
Internet-Draft
Intended status: Informational
Expires: November 22, 2018

C. Filsfils
S. Sivabalan
Cisco Systems, Inc.
S. Hegde
Juniper Networks, Inc.
D. Voyer
Bell Canada.
S. Lin
A. Bogdanov
P. Krol
Google, Inc.
M. Horneffer
Deutsche Telekom
D. Steinberg
Steinberg Consulting
B. Decraene
S. Litkowski
Orange Business Services
P. Mattes
Microsoft
Z. Ali
K. Talaulikar
J. Liste
F. Clad
K. Raza
Cisco Systems, Inc.
May 21, 2018

SR Policy Implementation and Deployment Considerations
draft-filsfils-spring-sr-policy-considerations-00.txt

Abstract

Segment Routing (SR) allows a headend node to steer a packet flow along any path. Intermediate per-flow states are eliminated thanks to source routing. SR Policy framework enables the instantiation and the management of necessary state on the headend node for flows along a source routed paths using an ordered list of segments associated with their specific SR Policies. This document describes some of the implementation and deployment aspects that are useful for operationalizing the SR Policy architecture.

Status of This Memo

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

This Internet-Draft will expire on November 22, 2018.

Table of Contents

1. Introduction

Segment Routing (SR) allows a headend node to steer a packet flow along any path. Intermediate per-flow states are eliminated with source routing [I-D.ietf-spring-segment-routing].

The headend node steers a flow into a Segment Routing Policy (SR Policy) by augmenting packet headers with the ordered list of segments associated with that SR Policy. [I-D.filsfils-spring-segment-routing-policy] defines the SR Policy architecture and details the concepts of SR Policy and steering into an SR Policy.

This document describes some of the implementation aspects for SR Policy framework which should be considered as suggestions. The same behavior, as defined in [I-D.filsfils-spring-segment-routing-policy], may in fact be realized with other alternate approaches. The deployment aspects described in this document are also meant to only serve as guidelines. This document describes these aspects and other considerations related to SR Policy concepts as they are important to facilitate multi-vendor interoperable deployments for various SR Policy use-cases.

These apply equally to the MPLS [I-D.ietf-spring-segment-routing-mpls] and SRv6 [I-D.filsfils-spring-srv6-network-programming] instantiations of segment routing.

For reading simplicity, the illustrations are provided for the MPLS instantiations.

2. SR Policy Headend Architecture

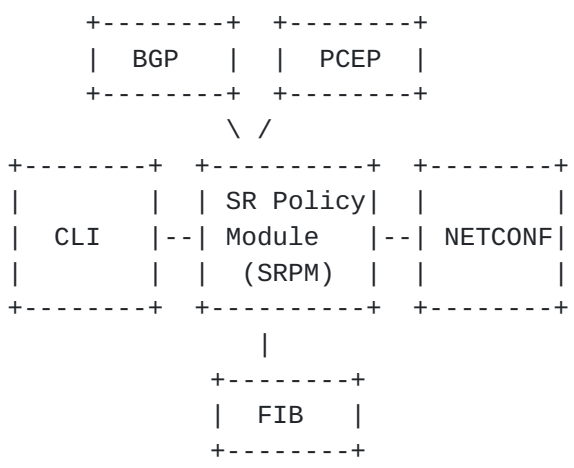

Figure 1: SR Policy Architecture at a Headend

The SR Policy functionality at a headend can be implemented in an SR Policy Module (SRPM) process as illustrated in Figure 1 .

The SRPM process interacts with other processes to learn candidate paths.

The SRPM process selects the active path of an SR Policy.

The SRPM process interacts with the RIB/FIB process to install an active SR Policy in the dataplane.

In order to validate explicit candidate paths and compute dynamic candidate paths, the SRPM process maintains an SR Database (SR-DB) as specified in [I-D.filsfils-spring-segment-routing-policy]. The SRPM process interacts with other processes as shown in Figure 2 to collect the SR-DB information.

Figure 2: Topology/link-state database architecture

The SR Policy architecture supports both centralized and distributed control-plane.

3. Dynamic Path Computation

A dynamic candidate path for SR Policy is specified as an optimization objective and constraints and needs to be computed by either the headend or a Path Computation Element (PCE). The distributed or centralized computation aspect is described further in Section 5. This section describes the computation aspects of a dynamic path.

3.1. Optimization Objective

This document describes two optimization objectives:

- Min-Metric - requests computation of a solution SID-List optimized for a selected metric.

- Min-Metric with margin and maximum number of SIDs - Min-Metric with two changes: a margin of by which two paths with similar metrics would be considered equal, a constraint on the max number of SIDs in the SID-List.

The "Min-Metric" optimization objective requests to compute a solution SID-List such that packets flowing through the solution SID-List use ECMP-aware paths optimized for the selected metric. The "Min-Metric" objective can be instantiated for the IGP metric ([RFC1195] [RFC2328] [RFC5340]) xor the TE metric ([RFC5305] [RFC3630]) xor the latency extended TE metric ([RFC7810] [RFC7471]). This metric is called the O metric (the optimized metric) to distinguish it from the IGP metric. The solution SID-List must be computed to minimize the number of SIDs and the number of SID-Lists.

If the selected O metric is the IGP metric and the headend and tailend are in the same IGP domain, then the solution SID-List is made of the single prefix-SID of the tailend.

When the selected O metric is not the IGP metric, then the solution SID-List is made of prefix SIDs of intermediate nodes, Adjacency SIDs along intermediate links and potentially Binding SIDs (BSIDs) of intermediate policies.

In many deployments there are insignificant metric differences between mostly equal path (e.g. a difference of 100 usec of latency between two paths from NYC to SFO would not matter in most cases). The "Min-Metric with margin" objective supports such requirement.

The "Min-Metric with margin and maximum number of SIDs" optimization objective requests to compute a solution SID-List such that packets flowing through the solution SID-List do not use a path whose

cumulative O metric is larger than the shortest-path O metric + margin.

If this is not possible because of the number of SIDs constraint, then the solution SID-List minimizes the O metric while meeting the maximum number of SID constraints (i.e. path with the least value of O metric while using <= the number of SIDs specified).

3.2. Constraints

The following constraints can be described:

- Inclusion and/or exclusion of TE affinity.
- Inclusion and/or exclusion of IP address.
- Inclusion and/or exclusion of SRLG.
- Inclusion and/or exclusion of admin-tag.
- Maximum accumulated metric (IGP, TE and latency).
- Maximum number of SIDs in the solution SID-List.
- Maximum number of weighted SID-Lists in the solution set.
- Diversity to another service instance (e.g., link, node, or SRLG disjoint paths originating from different head-ends).

3.3. SR Native Algorithm

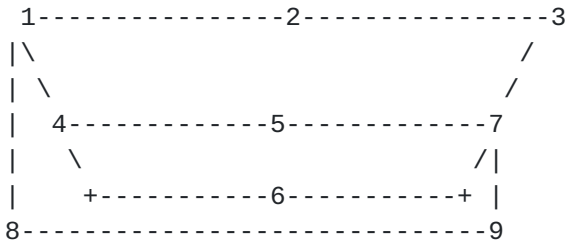

Figure 3: Illustration used to describe SR native algorithm

Let us assume that all the links have the same IGP metric of 10 and let us consider the dynamic path defined as: Min-Metric(from 1, to 3, IGP metric, margin 0) with constraint "avoid link 2-to-3".

A classical circuit implementation would do: prune the graph, compute the shortest-path, pick a single non-ECMP branch of the ECMP-aware

shortest-path and encode it as a SID-List. The solution SID-List would be <4, 5, 7, 3>.

An SR-native algorithm would find a SID-List that minimizes the number of SIDs and maximize the use of all the ECMP branches along the ECMP shortest path. In this illustration, the solution SID-List would be <7, 3>.

In the vast majority of SR use-cases, SR-native algorithms should be preferred: they preserve the native ECMP of IP and they minimize the dataplane header overhead.

In some specific use-case (e.g. TDM migration over IP where the circuit notion prevails), one may prefer a classic circuit computation followed by an encoding into SIDs (potentially only using non-protected Adj SIDs that pin the path to specific links and avoid ECMP to reflect the TDM paradigm).

SR-native algorithms are a local node behavior and are thus outside the scope of this document.

3.4. Path to SID

Let us assume the below diagram where all the links have an IGP metric of 10 and a TE metric of 10 except the link AB which has an IGP metric of 20 and the link AD which has a TE metric of 100. Let us consider the min-metric(from A, to D, TE metric, margin 0).

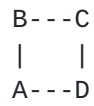

Figure 4: Illustration used to describe path to SID conversion

The solution path to this problem is ABCD.

This path can be expressed in SIDs as <B, D> where B and D are the IGP prefix SIDs respectively associated with nodes B and D in the diagram.

Indeed, from A, the IGP path to B is AB (IGP metric 20 better than ADCB of IGP metric 30). From B, the IGP path to D is BCD (IGP metric 20 better than BAD of IGP metric 30).

While the details of the algorithm remain a local node behavior, a high-level description follows: start at the headend and find an IGP prefix SID that leads as far down the desired path as possible(without using any link not included in the desired path).

If no prefix SID exists, use the Adj SID to the first neighbor along the path. Restart from the node that was reached.

4. Candidate Path Selection

An SR Policy may have multiple candidate paths that are provisioned or signaled [I-D.ietf-idr-segment-routing-te-policy] [I-D.ietf-pce-segment-routing] from one of more sources. The tie-breaker rules defined in [I-D.filsfils-spring-segment-routing-policy] result in determination of a single "active path" in a formal definition.

This section describe some examples for the candidate path selection based on the same rules.

Example 1:

Consider headend H where two candidate paths of the same SR Policy <color, endpoint> are signaled via BGP [I-D.ietf-idr-segment-routing-te-policy] and whose respective NLRIs have the same route distinguishers:

NLRI A with distinguisher = RD1, color = C, endpoint = N, preference P1.

NLRI B with distinguisher = RD1, color = C, endpoint = N, preference P2.

- Because the NLRIs are identical (same distinguisher), BGP will perform bestpath selection. Note that there are no changes to BGP best path selection algorithm.
- H installs one advertisement as bestpath into the BGP table.
- A single advertisement is passed to the SR Policy instantiation process.
- The SRPM process does not perform any path selection.

Note that the candidate path's preference value does not have any effect on the BGP bestpath selection process.

Example 2:

Consider headend H where two candidate paths of the same SR Policy <color, endpoint> are signaled via BGP and whose respective NLRIs have different route distinguishers:

NLRI A with distinguisher = RD1, color = C, endpoint = N, preference P1.

NLRI B with distinguisher = RD2, color = C, endpoint = N, preference P2.

- Because the NLRIs are different (different distinguisher), BGP will not perform bestpath selection.
- H installs both advertisements into the BGP table.
- Both advertisements are passed to the SR Policy instantiation process.
- SRPM process at H selects the candidate path advertised by NLRI B as the active path for the SR policy since P2 is greater than P1.

Note that the recommended approach is to use NLRIs with different distinguishers when several candidate paths for the same SR Policy (endpoint, color) are signaled via BGP to a headend.

Example 3:

Consider that a headend H learns two candidate paths of the same SR Policy <color, endpoint> one signaled via BGP and another via Local configuration.

NLRI A with distinguisher = RD1, color = C, endpoint = N, preference P1.

Local "foo" with color = C, endpoint = N, preference P2.

- H installs NLRI A into the BGP table.
- NLRI A and "foo" are both passed to the SRPM process.
- SRPM process at H selects the candidate path indicated by "foo" as the active path for the SR policy since P2 is greater than P1.

Now, let us consider cases, when an SR Policy has multiple valid candidate paths with the same best preference, the SRPM process at a

headend uses the rules described in [I-D.filsfils-spring-segment-routing-policy] section 2.9 to select the active path. This is explained in the following examples:

Example 4:

Consider headend H with two candidate paths of the same SR Policy <color, endpoint> and the same preference value received from the same controller R and where RD2 is higher than RD1.

- NLRI A with distinguisher RD1, color C, endpoint N, preference P1(selected as active path at time t0).
- NLRI B with distinguisher RD2 (RD2 is greater than RD1), color C, endpoint N, preference P1 (passed to SR Policy instatiation process at time t1 > t0).

After t1, SRPM process at H selects candidate path associated with NLRI B as active path of the SR policy since RD2 is higher than RD1. Here the time when the headend receives the candidate path via BGP is not a factor in the selection.

Note that, in such a scenario where there are redundant sessions to the same controller, the recommended approach is to use the same RD value for conveying the same candidate paths and let the BGP best path algorithm pick the best path.

Example 5:

Consider headend H with two candidate paths of the same SR Policy <color, endpoint> and the same preference value both received from the same controller R and where RD2 is higher than RD1.

Consider also that headend H is configured to override the discriminator tiebreaker specified in [I-D.filsfils-spring-segment-routing-policy] section 2.9

- NLRI A with distinguisher RD1, color C, endpoint N, preference P1 (selected as active path at time t0).
- NLRI B with distinguisher RD2, color C, endpoint N, preference P1 (passed to SR Policy instatiation process at time t1).

Even after t1, SRPM process at H retains candidate path associated with NLRI A as active path of the SR policy since the discriminator tiebreaker is disabled at H.

Example 6:

Consider headend H with two candidate paths of the same SR Policy <color, endpoint> and the same preference value.

- Local "foo" with color C, endpoint N, preference P1 (selected as active path at time t0).
- NLRI A with distinguisher RD1, color C, endpoint N, preference P1 (passed to SRPM process at time t1).

Even after t1, SRPM process at H retains candidate path associated with local candidate path "foo" as active path of the SR policy since the Local protocol is preferred over BGP by default based on its higher protocol identifier value.

Example 7:

Consider headend H with two candidate paths of the same SR Policy <color, endpoint> and the same preference value but received via NETCONF from two controllers R and S (where S > R)

- Path A from R with distinguisher D1, color C, endpoint N, preference P1 (selected as active path at time t0).
- Path B from S with distinguisher D2, color C, endpoint N, preference P1 (passed to SRPM process at time t1).

Note that the NETCONF process sends both paths to the SRPM process since it does not have any tiebreaker logic. After t1, SRPM process at H selects candidate path associated with Path B as active path of the SR policy.

5. Distributed and/or Centralized Control Plane

5.1. Distributed Control Plane within a single Link-State IGP area

Consider a single-area IGP with per-link latency measurement and advertisement of the measured latency in the extended-TE IGP TLV.

A head-end H is configured with a single dynamic candidate path for SR policy P with a low-latency optimization objective and endpoint E.

Clearly the SRPM process at H learns the topology (and extended TE latency information) from the IGP and computes the solution SID list providing the low-latency path to E.

No centralized controller is involved in such a deployment.

The SR-DB at H only uses the Link-State DataBase (LSDB) provided by the IGP.

5.2. Distributed Control Plane across several Link-State IGP areas

Consider a domain D composed of two link-state IGP single-area instances (I1 and I2) where each sub-domain benefits from per-link latency measurement and advertisement of the measured latency in the related IGP. The link-state information of each IGP is advertised via BGP-LS [RFC7752] towards a set of BGP-LS route reflectors (RR). H is a headend in IGP I1 sub-domain and E is an endpoint in IGP I2 sub-domain.

Using a BGP-LS session to any BGP-LS RR, H's SRPM process may learn the link-state information of the remote domain I2. H can thus compute the low-latency path from H to E as a solution SID list that spans the two domains I1 and I2.

The SR-DB at H collects the LSDB from both sub-domains (I1 and I2).

No centralized controller is required.

5.3. Centralized Control Plane

Considering the same domain D as in the previous section, let us now assume that H does not have a BGP-LS session to the BGP-LS RR's. Instead, let us assume a controller "C" has at least one BGP-LS session to the BGP-LS RR's.

The controller C learns the topology and extended latency information from both sub-domains via BGP-LS. It computes a low-latency path from H to E as a SID list <S1, S2, S3> and programs H with the related explicit candidate path.

The headend H does not compute the solution SID list (it cannot). The headend only validates the received explicit candidate path. Most probably, the controller encodes the SID's of the SID-List with Type-1. In that case, The headend's validation simply consists in resolving the first SID on an outgoing interface and next-hop.

The SR-DB at H only includes the LSDB provided by the IGP I1.

The SR-DB of the controller collects the LSDB from both sub-domains(I1 and I2).

5.4. Distributed and Centralized Control Plane

Consider the same domain D as in the previous section.

H's SRPM process is configured to associate color C1 with a low-latency optimization objective.

H's BGP process is configured to steer a Route R/r of extended-color community C1 and of next-hop N via an SR policy (N, C1).

Upon receiving a first BGP route of color C1 and of next-hop N, H recognizes the need for an SR Policy (N, C1) with a low-latency objective to N. As N is outside the SRTE DB of H, H requests a controller to compute such SID list (e.g., PCEP [I-D.ietf-pce-segment-routing]).

This is an example of hybrid control-plane: the BGP distributed control plane signals the routes and their TE requirements. Upon receiving these BGP routes, a local headend either computes the solution SID list (entirely distributed when the endpoint is in the SR-DB of the headend) else delegates the computation to a controller (hybrid distributed/centralized control-plane).

The SR-DB at H only includes the LSDB provided by the IGP.

The SR-DB of the controller collects the LSDB from both sub-domains.

6. Binding SID Aspects

The Binding SID (BSID) is fundamental to Segment Routing. It provides scaling, network opacity and service independence.

This section describes implementation and operational aspects related to the Binding SID.

6.1. Benefits of Binding SID

A simplified illustration is provided on the basis of Figure 5 where it is assumed that S, A, B, Data Center Interconnect DCI1 and DCI2 share the same IGP-SR instance in the data-center 1 (DC1). DCI1, DCI2, C, D, E, F, G, DCI3 and DCI4 share the same IGP-SR domain in the core. DCI3, DCI4, H, K and Z share the same IGP-SR domain in the data-center 2 (DC2).

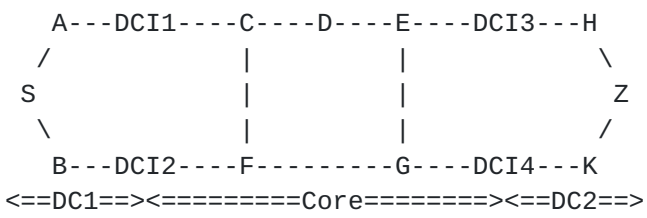

Figure 5: A Simple Datacenter Topology

In this example, it is assumed no redistribution between the IGP's and no presence of BGP-LU. The inter-domain communication is only provided by SR through SR Policies.

The latency from S to DCI1 equals to DCI2. The latency from Z to DCI3 equals to DCI4. All the intra-DC links have the same IGP metric 10.

The path DCI1, C, D, E, DCI3 has a lower latency and lower capacity than the path DCI2, F, G, DCI4.

The IGP metrics of all the core links are set to 10 except the links D-E which is set to 100.

A low-latency multi-domain policy from S to Z may be expressed as <DCI1, BSID, Z> where:

- DCI1 is the prefix SID of DCI1.
- BSID is the Binding SID bound to an SR policy <D, D2E, DCI3> instantiated at DCI1.
- Z is the prefix SID of Z.

Without the use of an intermediate core SR Policy (efficiently summarized by a single BSID), S would need to steer its low-latency flow into the policy <DCI1, D, D2E, DCI3, Z>.

The use of a BSID (and the intermediate bound SR Policy) decreases the number of segments imposed by the source.

A BSID acts as a stable anchor point which isolates one domain from the churn of another domain. Upon topology changes within the core of the network, the low-latency path from DCI1 to DCI3 may change. While the path of an intermediate policy changes, its BSID does not change. Hence the policy used by the source does not change, hence the source is shielded from the churn in another domain.

A BSID provides opacity and independence between domains. The administrative authority of the core domain may not want to share information about its topology. The use of a BSID allows keeping the service opaque. S is not aware of the details of how the low-latency service is provided by the core domain. S is not aware of the need of the core authority to temporarily change the intermediate path.

6.2. Centralized Discovery of available BSID

This section explains how controllers can discover the local SIDs available at a node N so as to pick an explicit BSID for a SR Policy to be instantiated at headend N.

Any controller can discover the following properties of a node N (e.g., via BGP-LS , NETCONF etc.):

- its local topology [RFC7752].
- its topology-related SIDs (Prefix SIDs, Adj SID and EPE SID [I-D.ietf-idr-bgp-ls-segment-routing-ext] [I-D.ietf-idr-bgpls-segment-routing-epe]).
- its Segment Routing Label Block (SRLB).
- its SR Policies and their BSID ([I-D.ietf-pce-segment-routing] [I-D.sivabalan-pce-binding-label-sid] [I-D.ietf-idr-te-lsp-distribution]).

Any controller can thus infer the available SIDs in the SRLB of any node.

As an example, a controller discovers the following characteristics of N: SRLB (4000, 8000), 3 Adj SIDs (4001, 4002, 4003), 2 EPE SIDs (4004, 4005) and 3 SRTE policies (whose BSIDs are respectively 4006, 4007 and 4008). This controller can deduce that the SRLB sub-range (4009, 5000) is free for allocation.

A controller is not restricted to use the next numerically available SID in the available SRLB sub-range. It can pick any label in the subset of available labels. This random pick make the chance for a collision unlikely.

An operator could also sub-allocate the SRLB between different controllers (e.g. (4000-4499) to controller 1 and (4500-5000) to controller 2).

Inter-controller state-synchronization may be used to avoid/detect collision in BSID.

All these techniques make the likelihood of a collision between different controllers very unlikely.

In the unlikely case of a collision, the controllers will detect it through system alerts, BGP-LS reporting using [I-D.ietf-idr-te-lsp-distribution] or PCEP notification [RFC8231]. They then have the choice to continue the operation of their SR Policy with the dynamically allocated BSID or re-try with another explicit pick.

Note: in deployments where PCE Protocol (PCEP) is used between head-end and controller (PCE), a head-end can report BSID as well as policy attributes (e.g., type of disjointness) and operational and administrative states to controller. Similarly, a controller can also assign/update the BSID of a policy via PCEP when instantiating or updating SR Policy.

7. Flex-Algorithm Based SR Policies

SR allows for association of algorithms to Prefix SIDs [I-D.ietf-spring-segment-routing]. [I-D.ietf-lsr-flex-algo] defines the IGP based Flex-Algorithm solution which allows IGPs themselves to compute constraint based paths over the network. Prefix SIDs for the specific flex-algorithm and associated with a node are used in the forwarding plane to steer along the specific constraint path to that node.

As specified in [I-D.ietf-spring-segment-routing] these IGP Flex Algo Prefix SIDs can be used as segments within SR Policies thereby leveraging the underlying IGP Flex Algo solution.

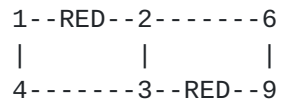

Figure 6: Illustration for Flex-Alg SID

Now let us assume that

- 1, 2, 3 and 4 are part of IGP 1.
- 2, 6, 9 and 3 are part of IGP 2.
- All the IGP link costs are 10.
- Links 1to2 and 3to9 are colored with IGP Link Affinity Red.

- Flex-Alg1 is defined in both IGPs as: avoid red, minimize IGP metric.
- All nodes of each IGP domain are enabled for FlexAlg1
- SID(k, 0) represents the PrefixSID of node k according to Alg=0.
- SID(k, FlexAlg1) represents the PrefixSID of node k according to Flex-Alg1.

A controller can steer a flow from 1 to 9 through an end-to-end path that avoids the RED links of both IGP domains thanks to the explicit SR Policy <SID(2, FlexAlg1), SID9(FlexAlg1)>.

8. Layer 2 and Optical Transport

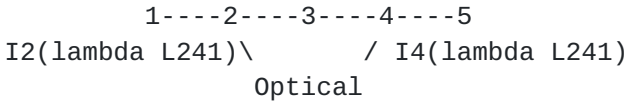

Figure 7: SR Policy with integrated DWDM

An explicit candidate path can express a path through a transport layer beneath IP (ATM, FR, DWDM). The transport layer could be ATM, FR, DWDM, back-to-back Ethernet etc. The transport path is modelled as a link between two IP nodes with the specific assumption that no distributed IP routing protocol runs over the link. The link may have IP address or be IP unnumbered. Depending on the transport protocol case, the link can be a physical DWDM interface and a lambda (integrated solution), an Ethernet interface and a VLAN, an ATM interface with a VPI/VCI, a FR interface with a DLCI etc.

Using the DWDM integrated use-case of Figure 7 as an illustration, let us assume

- nodes 1, 2, 3, 4 and 5 are IP routers running an SR-enable IGP on the links 1-2, 2-3, 3-4 and 4-5.
- The SRGB is homogeneous (16000, 24000).
- Node K's prefix SID is 16000+K.
- node 2 has an integrated DWDM interface I2 with Lambda L1.
- node 4 has an integrated DWDM interface I4 with Lambda L2.

- the optical network is provisioned with a circuit from 2 to 4 with continuous lambda L241 (details outside the scope of this document).

- Node 2 is provisioned with an SR policy with SID list <I2(L241)> and Binding SID B where I2(L241) is of type 5 (IPv4) or type 7 (IPv6), see section 4.

- node 1 steers a packet P1 towards the prefix SID of node 5 (16005).

- node 1 steers a packet P2 on the SR policy <16002, B, 16005>.

In such a case, the journey of P1 will be 1-2-3-4-5 while the journey of P2 will be 1-2-lambda(L241)-4-5. P2 skips the IP hop 3 and leverages the DWDM circuit from node 2 to node 4. P1 follows the shortest-path computed by the distributed routing protocol. The path of P1 is unaltered by the addition, modification or deletion of optical bypass circuits.

The salient point of this example is that the SR Policy architecture seamlessly support explicit candidate paths through any transport sub-layer.

BGP-LS Extensions to describe the sub-IP-layer characteristics of the SR Policy are out of scope of this document (e.g. in Figure 7, the DWDM characteristics of the SR Policy at node 2 in terms of latency, loss, security, domain/country traversed by the circuit etc.).

Further details of the SR Policy use-case for Packet Optical networks are specified in [I-D.anand-spring-poi-sr] .

9. Security Considerations

The security considerations related to Segment Routing architecture are described in [I-D.ietf-spring-segment-routing] and for SR Policy architecture are described in [I-D.filsfils-spring-segment-routing-policy] and they apply to this document as well.

10. IANA Considerations

This document has no actions for IANA.

11. Acknowledgement

The authors like to thank Tarek Saad, Dhanendra Jain and Muhammad Durrani for their valuable comments and suggestions.

12. References

12.1. Normative References

[I-D.filsfils-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., Raza, K., Liste, J., Clad, F., Talaulikar, K., Ali, Z., Hegde, S., daniel.voyer@bell.ca, d., Lin, S., bogdanov@google.com, b., Krol, P., Horneffer, M., Steinberg, D., Decraene, B., Litkowski, S., and P. Mattes, "Segment Routing Policy for Traffic Engineering", draft-filsfils-spring-segment-routing-policy-05 (work in progress), February 2018.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-15 (work in progress), January 2018.

12.2. Informative References

[I-D.anand-spring-poi-sr]
Anand, M., Bardhan, S., Subrahmaniam, R., Tantsura, J., Mukhopadhyaya, U., and C. Filsfils, "Packet-Optical Integration in Segment Routing", draft-anand-spring-poi-sr-05 (work in progress), February 2018.

[I-D.filsfils-spring-srv6-network-programming]
Filsfils, C., Li, Z., Leddy, J., daniel.voyer@bell.ca, d., daniel.bernier@bell.ca, d., Steinberg, D., Raszuk, R., Matsushima, S., Lebrun, D., Decraene, B., Peirens, B., Salsano, S., Naik, G., Elmalky, H., Jonnalagadda, P., and M. Sharif, "SRv6 Network Programming", draft-filsfils-spring-srv6-network-programming-04 (work in progress), March 2018.

[I-D.ietf-idr-bgp-ls-segment-routing-ext]
Previdi, S., Talaulikar, K., Filsfils, C., Gredler, H., and M. Chen, "BGP Link-State extensions for Segment Routing", draft-ietf-idr-bgp-ls-segment-routing-ext-07 (work in progress), May 2018.

[I-D.ietf-idr-bgpls-segment-routing-epe]
Previdi, S., Filsfils, C., Patel, K., Ray, S., and J. Dong, "BGP-LS extensions for Segment Routing BGP Egress Peer Engineering", draft-ietf-idr-bgpls-segment-routing-epe-15 (work in progress), March 2018.

[I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Jain, D., Mattes, P., Rosen, E., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-03 (work in progress), May 2018.

[I-D.ietf-idr-te-lsp-distribution]
Previdi, S., Dong, J., Chen, M., Gredler, H., and J. Tantsura, "Distribution of Traffic Engineering (TE) Policies and State using BGP-LS", draft-ietf-idr-te-lsp-distribution-08 (work in progress), December 2017.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-00 (work in progress), May 2018.

[I-D.ietf-pce-segment-routing]
Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "PCEP Extensions for Segment Routing", draft-ietf-pce-segment-routing-11 (work in progress), November 2017.

[I-D.ietf-spring-segment-routing-mpls]
Bashandy, A., Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-13 (work in progress), April 2018.

[I-D.sivabalan-pce-binding-label-sid]
Sivabalan, S., Tantsura, J., Filsfils, C., Previdi, S., Hardwick, J., and D. Dhody, "Carrying Binding Label/ Segment-ID in PCE-based Networks.", draft-sivabalan-pce-binding-label-sid-04 (work in progress), March 2018.

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <https://www.rfc-editor.org/info/rfc1195>.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <https://www.rfc-editor.org/info/rfc3630>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <https://www.rfc-editor.org/info/rfc5340>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <https://www.rfc-editor.org/info/rfc7471>.

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <https://www.rfc-editor.org/info/rfc7752>.

[RFC7810] Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 7810, DOI 10.17487/RFC7810, May 2016, <https://www.rfc-editor.org/info/rfc7810>.

[RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <https://www.rfc-editor.org/info/rfc8231>.

Authors' Addresses

Clarence Filsfils
Cisco Systems, Inc.
Pegasus Parc
De kleetlaan 6a, DIEGEM BRABANT 1831
BELGIUM

Email: cfilsfil@cisco.com

Siva Sivabalan
Cisco Systems, Inc.
2000 Innovation Drive
Kanata, Ontario K2K 3E8
Canada

Email: msiva@cisco.com

Shraddha Hegde
Juniper Networks, Inc.
Embassy Business Park
Bangalore, KA 560093
India

Email: shraddha@juniper.net

Daniel Voyer
Bell Canada.
671 de la gauchetiere W
Montreal, Quebec H3B 2M8
Canada

Email: daniel.voyer@bell.ca

Steven Lin
Google, Inc.

Email: stevenlin@google.com

Alex Bogdanov
Google, Inc.

Email: bogdanov@google.com

Przemyslaw Krol
Google, Inc.

Email: pkrol@google.com

Martin Horneffer
Deutsche Telekom

Email: martin.horneffer@telekom.de

Dirk Steinberg
Steinberg Consulting

Email: dws@steinbergnet.net

Bruno Decraene
Orange Business Services

Email: bruno.decraene@orange.com

Stephane Litkowski
Orange Business Services

Email: stephane.litkowski@orange.com

Paul Mattes
Microsoft
One Microsoft Way
Redmond, WA 98052-6399
USA

Email: pamattes@microsoft.com

Zafar Ali
Cisco Systems, Inc.

Email: zali@cisco.com

Ketan Talaulikar
Cisco Systems, Inc.

Email: ketant@cisco.com

Jose Liste
Cisco Systems, Inc.
821 Alder Drive
Milpitas, California 95035
USA

Email: jliste@cisco.com

Francois Clad
Cisco Systems, Inc.

Email: fclad@cisco.com

Kamran Raza
Cisco Systems, Inc.
2000 Innovation Drive
Kanata, Ontario K2K 3E8
Canada

Email: skraza@cisco.com