Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2014

C. Filsfils, Ed.
Cisco Systems, Inc.
P. Francois, Ed.
IMDEA Networks
S. Previdi
Cisco Systems, Inc.
B. Decraene
S. Litkowski
Orange
M. Horneffer
Deutsche Telekom
I. Milojevic
Telekom Srbija
R. Shakir
British Telecom
S. Ytti
TDC Oy
W. Henderickx
Alcatel-Lucent
J. Tantsura
S. Kini
Ericsson
E. Crabbe
Google, Inc.

October 21, 2013

Segment Routing Use Cases
draft-filsfils-rtgwg-segment-routing-use-cases-02

Abstract

Segment Routing (SR) leverages the source routing and tunneling paradigms. A node steers a packet through a controlled set of instructions, called segments, by prepending the packet with an SR header. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any topological path and service chain while maintaining per-flow state only at the ingress node of the SR domain.

The Segment Routing architecture can be directly applied to the MPLS dataplane with no change on the forwarding plane. It requires minor extension to the existing link-state routing protocols. Segment Routing can also be applied to IPv6 with a new type of routing extension header.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 24, 2014.

Copyright Notice

Table of Contents

1. Introduction

The objective of this document is to illustrate the properties and benefits of the SR architecture, through the documentation of various SR use-cases.

Section 2 illustrates the ability to tunnel traffic towards remote service points without any other protocol than the IGP.

Section 3 reports various FRR use-cases leveraging the SR functionality.

Section 4 documents traffic-engineering use-cases, with and without support of bandwidth admission control.

Section 5 documents the use of SR to perform service chaining.

Section 6 illustrates OAM use-cases.

1.1. Companion Documents

The main reference for this document is the SR architecture defined in [draft-filsfils-rtgwg-segment-routing-01].

The SR instantiation in the MPLS dataplane is described in [I-D.gredler-isis-label-advertisement].

[draft-filsfils-spring-segment-routing-ldp-interop-00] documents the co-existence and interworking with MPLS Signaling protocols.

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

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

Fast-Reroute for Segment Routing is described in [I-D.francois-sr-frr].

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

The SR instantiation in the IPv6 dataplane will be described in a future draft.

1.2. Editorial simplification

A unique index is allocated to each IGP Prefix Segment. The related absolute segment associated to an IGP Prefix SID is determined by summing the index and the base of the SRGB. In the SR architecture, each node can be configured with a different SRGB and hence the absolute SID associated to an IGP Prefix Segment can change from node to node.

We have described the first use-case of this document in the most generic way, i.e. with different SRGB at each node in the SR IGP domain. We have detailed the packet path highlighting that the SID of a Prefix Segment may change hop by hop.

For editorial simplification purpose, we will assume for all the other use cases that the operator ensures a single consistent SRGB across all the nodes in the SR IGP domain. In that case, all the nodes associate the same absolute SID with the same index and hence one can use the absolute SID value instead of the index to refer to a Prefix SID.

Several operators have indicated that they would deploy the SR technology in this way: with a single consistent SRGB across all the nodes. They motivated their choice based on operational simplicity (e.g. troubleshooting across different nodes).

While this document notes this operator feedback and we use this deployment model to simplify the text, we highlight that the SR architecture is not limited to this specific deployment use-case (different nodes may have different SRGB thanks to the indexation of Prefix SID’s).

2. IGP-based MPLS Tunneling

SR, applied to the MPLS dataplane, offers the ability to tunnel services (VPN, VPLS, VPWS) from an ingress PE to an egress PE, without any other protocol than ISIS or OSPF. LDP and RSVP-TE signaling protocols are not required.

The operator only needs to allocate one node segment per PE and the SR IGP control-plane automatically builds the required MPLS forwarding constructs from any PE to any PE.

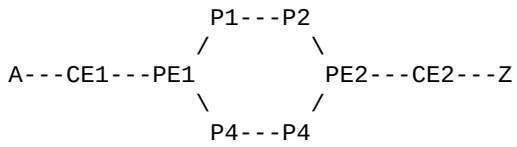


Figure 1: IGP-based MPLS Tunneling

In Figure 1 above, the four nodes A, CE1, CE2 and Z are part of the same VPN. CE2 advertises to PE2 a route to Z. PE2 binds a local label LZ to that route and propagates the route and its label via MPBGP to PE1 with nhop 192.168.0.2. PE1 installs the VPN prefix Z in the appropriate VRF and resolves the next-hop onto the node segment associated with PE2. Upon receiving a packet from A destined to Z, PE1 pushes two labels onto the packet: the top label is the Prefix SID attached to 192.168.0.2/32, the bottom label is the VPN label LZ attached to the VPN route Z.

The Prefix-SID attached to prefix 192.168.0.2 is a shared segment within the IGP domain, as such it is indexed.

Let us assume that:

- the operator allocated the index 2 to the prefix 192.168.0.2/32

- the operator allocated SRGB [100, 199] at PE1

- the operator allocated SRGB [200, 299] at P1

- the operator allocated SRGB [300, 399] at P2

- the operator allocated SRGB [400, 499] at P3

- the operator allocated SRGB [500, 599] at P4

- the operator allocated SRGB [600, 699] at PE2

Thanks to this context, any SR-capable IGP node in the domain can determine what is the segment associated with the Prefix-SID attached to prefix 192.168.0.2/32:

- PE1’s SID is 100+2=102

- P1’s SID is 200+2=202

- P2's SID is 300+2=302

- P3's SID is 400+2=402

- P4's SID is 500+2=502

- PE2's SID is 600+2=602

Specifically to our example this means that PE1 load-balance the traffic to VPN route Z between P1 and P4. The packets sent to P1 have a top label 202 while the packets sent to P4 have a top label 502. P1 swaps 202 for 302 and forwards to P2. P2 pops 302 and forwards to PE2. The packets sent to P4 had label 502. P4 swaps 502 for 402 and forwards the packets to P3. P3 pops the top label and forwards the packets to PE2. Eventually all the packets reached PE2 with one single lable: LZ, the VPN label attached to VPN route Z.

This scenario illustrates how supporting MPLS services (VPN, VPLS, VPWS) with SR has the following benefits:

- Simple operation: one single intra-domain protocol to operate: the IGP. No need to support IGP synchronization extensions as described in [RFC5443] and [RFC6138].

- Excellent scaling: one Node-SID per PE.

3. Fast Reroute

Segment Routing aims at supporting services with tight SLA guarantees [draft-filsfils-rtgwg-segment-routing-01]. To meet this goal, local protection mechanisms can be useful to provide fast connectivity restoration after the sudden failure of network components. Protection mechanisms for segments aim at letting a point of local repair (PLR) pre-compute and install state allowing to locally recover the delivery of packets when the primary outgoing interface corresponding to the protected active segment is down.

This section describes use-cases leading to the definition of different protection mechanisms for node, adjacency, and service segments to be supported by the SR architecture.

3.1. Protecting node and adjacency segments

Node and adjacency segments are used to determine the path that a packet should follow from an ingress node to an egress node of the SR domain or a service node.

Ensuring fast recovery of the packet delivery service may wear different requirements depending on the application using the segment. For this reason, the SR architecture should be able to accomodate multiple protection mechanisms and provide means to the operator to configure the protection scheme applied for the segments that are advertised in the SR domain.

The operator may want to achieve fast recovery in case of failures with as little management effort as possible, using a protection mechanism provided by the Segment Routing architecture itself. In this case, a Segment Routing node is in charge of discovering "by default" protection paths for each of its adjacent network component, with minimal operational impact. Approaches for such applications, typically in line with classical IP-FRR solutions, are discussed in [I-D.francois-sr-frr].

The operator of a Segment Routing network may also have strict policies on how a given network component should be protected against failures. A typical case is the knowledge by an external controller (or through any other tool used by the operator) of shared risk among different components, which should not be used to protect each other. An operator could notably use [I-D.sivabalan-pce-segment-routing] for this purpose.

Third, some SR applications have strict requirements in terms of guaranteed performance, disjointness in the infrastructure components used for different services, or for redundant provisioning of such services. An approach for providing resiliency in these contexts is explained in [I-D.shakir-rtgwg-sr-performance-engineered-lsps]. It is basically aiming at letting the ingress node in the SR domain be in charge of the recovery of the Segment Routing paths that it uses to support these services.

The protection behavior applied to a given SID must be advertised in the routing information that is propagated in the SR domain for that SID, e.g., in [I-D.previdi-isis-segment-routing-extensions]. Nodes injecting traffic in the SR domain can hence select segments based on the protection mechanism that is required for their application.

3.2. Protecting a node segment upon the failure of its advertising node

Service segments can also benefit from a fast restoration mechanism provided by the SR architecture.

Referring to the below figure, let us assume:

A is identified by IP address 192.0.2.1/32 to which Node-SID 101 is attached.

B is identified by IP address 192.0.2.2/32 to which Node-SID 102 is attached

A and B host the same set of services.

Each service is identified by a local segment at each node: i.e. node A allocates a local service segment 9001 to identify a specific service S while the same service is identified by a local service segment 9002 at B. Specifically, for the sake of this illustration, let us assume that service S is a BGP-VPN service where A announces a VPN route V with BGP nhop 192.0.2.1/32 and local VPN label 9001 and B announces the same VPN route V with BGP nhop 192.0.2.2/32 and local VPN label 9002.

A generic mesh interconnects the three nodes M, Q and B.

N prefers to use the service S offered by A and hence sends its S-destined traffic with segment list {101, 9001}.

Q is a node connected to A.

Q has a method to detect the loss of node A within a few 10's of msec.

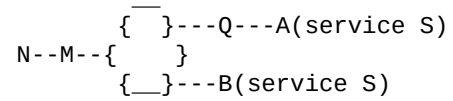


Figure 2: Service Mirroring

In that context, we would like to protect the traffic destined to service S upon the failure of node A.

The solution is built upon several components:

1. B advertises its mirroring capability for mirrored Node-SID 101
2. B pre-installs a mirroring table in order to process the packets originally destined to 101.
3. Q and any neighbor of A pre-install the Mirror_FRR LFA extension
4. All nodes implements a modified SRDB convergence upon Node-SID 101 deletion

3.2.1. Advertisement of the Mirroring Capability

B advertises a MIRROR sub-TLV in its IGP Link-State Router Capability TLV with the values (TTT=000, MIRRORED_OBJECT=101, CONTEXT_SEGMENT=10002),[draft-filsfils-rtgwg-segment-routing-01], [I-D.previdi-isis-segment-routing-extensions] and [draft-psenak-ospf-segment-routing-extensions-00] for more details in the encodings.

Doing so, B advertises within the routing domain that it is willing to backup any traffic originally sent to Node-SID 101 provided that this rerouted traffic gets to B with the context segment 10002 directly preceding any local service segment advertised by A. 10002 is a local context segment allocated by B to identify traffic that was originally meant for A. This allows B to match the subsequent service segment (e.g. 9001) correctly.

3.2.2. Mirroring Table

We assume that B is able to discover all the local service segments allocated by A (e.g. BGP route reflection and add-path). B maps all the services advertised by A to its similar service representations. For example, service 9001 advertised by A is mapped to service 9002 advertised by B as both relate to the same service S (the same VPN route V). For example, B applies the same service treatment to a packet received with top segments {102, 10002, 9001} or with top segments {102, 9002}. Basically, B treats {10002, 9001} as a synonym of {9002}.

3.2.3. LFA FRR at the Point of Local Repair

In advance of any failure of A, Q (and any other node connected to A) learns the identity of the IGP Mirroring node for each Node-SID advertised by A (MIRROR_TLV advertised by B) and pre-installs the following new MIRROR_FRR entry:

- Trigger condition: the loss of nhop A
- Incoming active segment: 101 (a Node-SID advertised by A)
- Primary Segment processing: pop 101
  - Backup Segment processing: pop 101, push {102, 10002}
- Primary nhop: A
  - Backup nhop: primary path to node B

Upon detecting the loss of node A, Q intercepts any traffic destined to Node-SID 101, pops the segment to A (101) and push a repair tunnel {102, 10002}. Node-SID 102 steers the repaired traffic to B while context segment 10002 allows B to process the following service segment {9001} in the right context table.

3.2.4. Modified IGP Convergence upon Node deletion

Upon the failure of A, all the neighbors of A will flood the loss of their adjacency to A and eventually every node within the IGP domain will delete 192.0.2.1/32 from their RIB.

The RIB deletion of 192.0.2.1/32 at N is beneficial as it triggers the BGP FRR Protection onto the precomputed backup next-hop [draft-rtgwg-bgp-pic-01.txt].

The RIB deletion at node M, if it occurs before the RIB deletion at N, would be disastrous as it would lead to the loss of the traffic from N to A before Q is able to apply the Mirroring protection.

The solution consists in delaying the deletion of the SRDB entry for 101 by 2 seconds while still deleting the IP RIB 192.0.2.1/32 entry immediately.

The RIB deletion triggers the BGP FRR and BGP Convergence. This is beneficial and must occur without delay.

The deletion of the SRDB entry to Node-SID101 is delayed to ensure that the traffic still in transit towards Node-SID 101 is not dropped.

The delay timer should be long enough to ensure that either the BGP FRR or the BGP Convergence has taken place at N.

3.2.5. Conclusions

In our reference figure, N sends its packets towards A with the segment list {101, 9001}. The shortest-path from S to A transits via M and Q.

Within a few msec of the loss of A, Q activates its pre-installed Mirror_FRR entry and reroutes the traffic to B with the following segment list {102, 10002, 9001}.

Within a few 100’s of msec, any IGP node deletes its RIB entry to A but keeps its SRDB entry to Node-SID 101 for an extra 2 seconds.

Upon deleting its RIB entry to 192.0.2.1/32, N activates its BGP FRR entry and reroutes its S destined traffic towards B with segment list {102, 9002}.

By the time any IGP node deletes the SRDB entry to Node-SID 101, N no longer sends any traffic with Node-SID 101.

The deletion of the SRDB entry to Node-SID101 is delayed to ensure that the traffic still in transit towards Node-SID 101 is not dropped.

In conclusion, the traffic loss only depends on the ability of Q to detect the node failure of its adjacent node A.

4. Traffic Engineering

In this section, we describe Traffic Engineering use-cases for SR, distinguishing use-cases for traffic engineering with bandwidth admission control from those without.

4.1. Traffic Engineering without Bandwidth Admission Control

This section describes traffic-engineering use-cases which do not require bandwidth admission control.

The first sub-section illustrates the use of anycast segments to express macro policies. Two examples are provided: one involving a disjointness enforcement within a so-called dual-plane network, and the other involving CoS-based policies.

The second sub-section illustrate how a head-end router can combine a distributed CSPF computation with SR. Various examples are provided where the CSPF constraint or objective is either a TE affinity, an SRLG or a latency metric.

The third sub-section illustrates how SR can help traffic-engineer outbound traffic among different external peers, overriding the best installed IP path at the egress border routers.

The fourth sub-section describes how SR can be used to express deterministic non-ECMP paths. Several techniques to compress the related segment lists are also introduced.

The fifth sub-section describes a use-case where a node attaches an Adj-SID to a set of its interfaces however not sharing the same neighbor. The illustrated benefit relates to loadbalancing.

4.1.1. Anycast Node Segment

The SR architecture defines an anycast segment as a segment attached to an anycast IP prefix ([RFC4786]).

The anycast node segment is an interesting tool for traffic engineering:

Macro-policy support: anycast segments allow to express policies such as "go via plane1 of a dual-plane network" (Section 4.1.1.1) or "go via Region3" (Section 4.1.3).

Implicit node resiliency: the traffic-engineering policy is not anchored to a specific node whose failure could impact the service. It is anchored to an anycast address/Anycast-SID and hence the flow automatically reroutes on any ECMP-aware shortest-path to any other router part of the anycast set.

The two following sub-sections illustrate to traffic-engineering use-cases leveraging Anycast-SID.

4.1.1.1. Disjointness in dual-plane networks

Many networks are built according to the dual-plane design:
Each access region k is connected to the core by two C routers (C(1,k) and C(2,k)).

C(1,k) is part of plane 1 and aggregation region K

C(2,k) is part of plane 2 and aggregation region K

C(1,k) has a link to C(2, j) iff k = j.

The core nodes of a given region are directly connected. Inter-region links only connect core nodes of the same plane.

{C(1,k) has a link to C(1, j)} iff {C(2,k) has a link to C(2, j)}.

The distribution of these links depends on the topological properties of the core of the AS. The design rule presented above specifies that these links appear in both core planes.

We assume a common design rule found in such deployments: the inter-plane link costs (Cik-Cjk where i<>j) are set such that the route to an edge destination from a given plane stays within the plane unless the plane is partitioned.

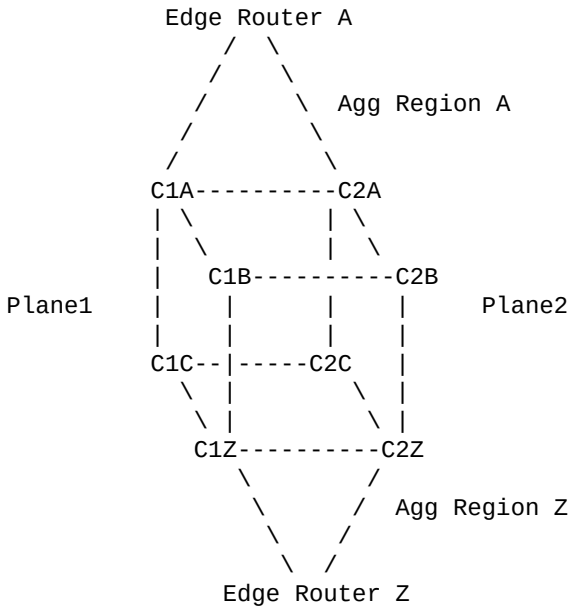


Figure 3: Dual-Plane Network and Disjointness

In the above network diagram, let us that the operator configures:

The four routers (C1A, C1B, C1C, C1Z) with an anycast loopback address 192.0.2.1/32 and an Anycast-SID 101.

The four routers (C2A, C2B, C2C, C2Z) with an anycast loopback address 192.0.2.2/32 and an Anycast-SID 102.

Edge router Z with Node-SID 109.

A can then use the three following segment lists to control its Z-destined traffic:

{109}: the traffic is load-balanced across any ECMP path through the network.

{101, 109}: the traffic is load-balanced across any ECMP path within the Plane1 of the network.

{102, 109}: the traffic is load-balanced across any ECMP path within the Plane2 of the network.

Most of the data traffic to Z would use the first segment list, such as to exploit the capacity efficiently. The operator would use the

two other segment lists for specific premium traffic that has requested disjoint transport.

For example, let us assume a bank or a government customer has requested that the two flows F1 and F2 injected at A and destined to Z should be transported across disjoint paths. The operator could classify F1 (F2) at A and impose and SR header with the second (third) segment list. Focusing on F1 for the sake of illustration, A would route the packets based on the active segment, Anycast-SID 101, which steers the traffic along the ECMP-aware shortest-path to the closest router part of the Anycast-SID 101, C1A is this example. Once the packets have reached C1A, the second segment becomes active, Node-SID 109, which steers the traffic on the ECMP-aware shortest-path to Z. C1A load-balances the traffic between C1B-C1Z and C1C-C1Z and then C1Z forwards to Z.

This SR use-case has the following benefits:

- Zero per-service state and signaling on midpoint and tail-end routers.
- Only two additional node segments (one Anycast-SID per plane).
- ECMP-awareness.
- Node resiliency property: the traffic-engineering policy is not anchored to a specific core node whose failure could impact the service.

4.1.1.2. CoS-based Traffic Engineering

Frequently, different classes of service need different path characteristics.

In the example below, a single-area international network with presence in four different regions of the world has lots of cheap network capacity from Region4 to Region1 via Region2 and some scarce expensive capacity via Region3.

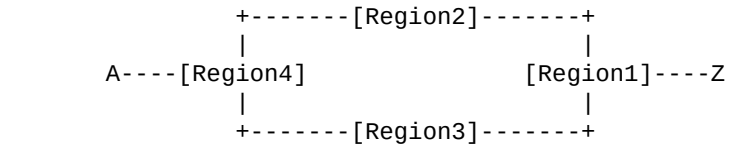


Figure 4: International Topology Example

In such case, the IGP metrics would be tuned to have a shortest-path from A to Z via Region2.

This would provide efficient capacity planning usage while fulfilling the requirements of most of the traffic demands. However, it may not suite the latency requirements of the voice traffic between the two cities.

Let us illustrate how this can be solved with Segment Routing.

The operator would configure:
- All the core routers in Region3 with an anycast loopback 192.0.2.3/32 to which Anycast-SID 333 is attached.
- A loopback 192.0.2.9/32 on Z and would attach Node-SID 109 to it.
- The IGP metrics such that the shortest-path from Region4 to Region1 is via Region2, from Region4 to Region3 is directly to Region3, the shortest-path from Region3 to Region1 is not back via Region4 and Region2 but straight to Region1.

With this in mind, the operator would instruct A to apply the following policy for its Z-destined traffic:
- Voice traffic: impose segment-list {333, 109}
  - Anycast-SID 333 steers the Voice traffic along the ECMP-aware shortest-path to the closest core router in Region3, then Node-SID 109 steers the Voice traffic along the ECMP-aware shortest-path to Z. Hence the Voice traffic reaches Z from A via the low-latency path through Region3.

- Any other traffic: impose segment-list {109}: Node-SID 109 steers the Voice traffic along the ECMP-aware shortest-path to Z. Hence the bulk traffic reaches Z from A via the cheapest path for the operator.

This SR use-case has the following benefits:

Zero per-service state and signaling at midpoint and tailend nodes.

One additional anycast segment per region.

ECMP-awareness.

Node resiliency property: the traffic-engineering policy is not anchored to a specific core node whose failure could impact the service.

4.1.2. Distributed CSPF-based Traffic Engineering

In this section, we illustrate how a head-end router can map the result of its distributed CSPF computation into an SR segment list.

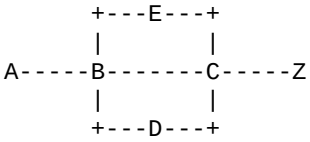


Figure 5: SRLG-based CSPF

Let us assume that in the above network diagram:

The operator configures a policy on A such that its Z-destined traffic must avoid SRLG1.

The operator configures SRLG1 on the link BC (or is learned dynamically from the IP/Optical interaction with the DWDM network).

The SRLG's are flooded in the link-state IGP.

The operator respectively configures the Node-SIDs 101, 102, 103, 104, 105 and 109 at nodes A, B, C, D, E and Z.

In that context, A can apply the following CSPF behavior:

- It prunes all the links affected by the SRLG1, computes an SPF on the remaining topology and picks one of the SPF paths.
  - In our example, A finds two possible paths ABECZ and ABDCZ and let's assume it takes the ABDCZ path.

- It translates the path as a list of segments
  - In our example, ABDCZ can be expressed as {104, 109}: a shortest path to node D, followed by a shortest-path to node Z.

- It monitors the status of the LSDB and upon any change impacting the policy, it either recomputes a path meeting the policy or update its translation as a list of segments.
  - For example, upon the loss of the link DC, the shortest-path to Z from D (Node-SID 109) goes via the undesired link BC. After a transient time immediately following such failure, the node A would figure out that the chosen path is no longer valid and instead select ABECZ which is translated as {103, 109}.

- This behavior is a local matter at node A and hence the details are outside the scope of this document.

The same use-case can be derived from any other C-SPF objective or constraint (TE affinity, TE latency, SRLG, etc.) as defined in [RFC5305] and [I-D.previdi-isis-te-metric-extensions]. Note that the bandwidth case is specific and hence is treated in Section 4.2.

4.1.3. Egress Peering Traffic Engineering

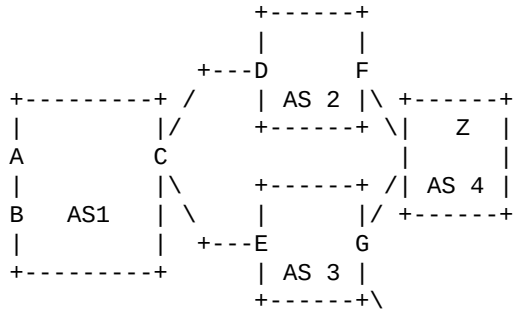


Figure 6: Egress peering traffic engineering

Let us assume that:

C in AS1 learns about destination Z of AS 4 via two BGP paths (AS2, AS4) and (AS3, AS4).

C sets next-hop-self before propagating the paths within AS1.

C propagates all the paths to Z within AS1 (add-path).

C only installs the path via AS2 in its RIB.

In that context, the operator of AS1 cannot apply the following traffic-engineering policy:

Steer 60% of the Z-destined traffic received at A via AS2 and 40% via AS3.

Steer 80% of the Z-destined traffic received at B via AS2 and 20% via AS3.

This traffic-engineering policy can be supported thanks to the following SR configuration.

The operator configures:

C with a loopback 192.0.2.1/32 and attach the Node-SID 101 to it.

C to bind an external adjacency segment ([draft-filsfils-rtgwg-segment-routing-01]) to each of its peering interface.

For the sake of this illustration, let us assume that the external adjacency segments bound by C for its peering interfaces to (D, AS2) and (E, AS3) are respectively 9001 and 9002.

These external adjacencies (and their attached segments) are flooded within the IGP domain of AS1 [RFC5316].

As a result, the following information is available within AS1:
ISIS Link State Database:

- Node-SID 101 is attached to IP address 192.0.2.1/32 advertised by C.
- C is connected to a peer D with external adjacency segment 9001.
- C is connected to a peer E with external adjacency segment 9002.

BGP Database:

- Z is reachable via 192.0.2.1 with AS Path {AS2, AS4}.
- Z is reachable via 192.0.2.1 with AS Path {AS3, AS4}.

The operator of AS1 can thus meet its traffic-engineering objective by enforcing the following policies:

A should apply the segment list {101, 9001} to 60% of the Z-destined traffic and the segment list {101, 9002} to the rest.

B should apply the segment list {101, 9001} to 80% of the Z-destined traffic and the segment list {101, 9002} to the rest.

Node segment 101 steers the traffic to C.

External adjacency segment 9001 forces the traffic from C to (D, AS2), without any IP lookup at C.

External adjacency segment 9002 forces the traffic from C to (E, AS3), without any IP lookup at C.

A and B can also use the described segments to assess the liveness of the remote peering links, see OAM section.

4.1.4. Deterministic non-ECMP Path

The previous sections have illustrated the ability to steer traffic along ECMP-aware shortest-paths. SR is also able to express deterministic non-ECMP path: i.e. as a list of adjacency segments. We illustrate such an use-case in this section.

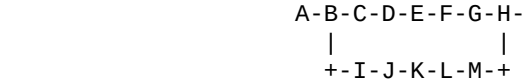


Figure 7: Non-ECMP deterministic path

In the above figure, it is assumed all nodes are SR capable and only the following SIDs are advertised:

- A advertises Adj-SID 9001 for its adjacency to B
- B advertises Adj-SID 9002 for its adjacency to C
- C advertises Adj-SID 9003 for its adjacency to D
- D advertises Adj-SID 9004 for its adjacency to E
- E advertises Adj-SID 9001 for its adjacency to F
- F advertises Adj-SID 9002 for its adjacency to G
- G advertises Adj-SID 9003 for its adjacency to H
- H advertises Adj-SID 9004 for its adjacency to Z
- E advertises Node-SID 101
- Z advertises Node-SID 109

The operator can steer the traffic from A to Z via a specific non-ECMP path ABCDEFGHZ by imposing the segment list {9001, 9002, 9003, 9004, 9001, 9002, 9003, 9004}.

The following sub-sections illustrate how the segment list can be compressed.

4.1.4.1. Node Segment

Clearly the same exact path can be expressed with a two-entry segment list {101, 109}.

This example illustrates that a Node Segment can also be used to express deterministic non-ECMP path.

4.1.4.2. Forwarding Adjacency

The operator can configure Node B to create a forwarding-adjacency to node H along an explicit path BCDEFGH. The following behaviors can then be automated by B:

B attaches an Adj-SID (e.g. 9007) to that forwarding adjacency together with an ERO sub-sub-TLV which describes the explicit path BCDEFGH.

B installs in its Segment Routing Database the following entry:

Active segment: 9007.

Operation: NEXT and PUSH {9002, 9003, 9004, 9001, 9002, 9003}

As a result, the operator can configure node A with the following compressed segment list {9001, 9007, 9004}.

4.1.5. Load-balancing among non-parallel links

A given node may assign the same Adj-SID to multiple of its adjacencies, even if these ones lead to different neighbors. This may be useful to support traffic engineering policies.

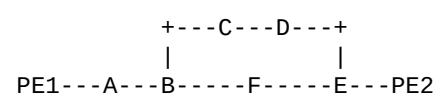


Figure 8: Adj-SID For Multiple (non-parallel) Adjacencies

In the above example, let us assume that the operator:

Requires PE1 to load-balance its PE2-destined traffic between the ABCDE and ABFE paths.

Configures B with Node-SID 102 and E with Node-SID 202.

Configures B to advertise an individual Adj-SID per adjacency (e.g. 9001 for BC and 9002 for BF) and, in addition, an Adj-SID for the adjacency set (BC, BF) (e.g. 9003).

With this context in mind, the operator achieves its objective by configuring the following traffic-engineering policy at PE1 for the PE2-destined traffic: {102, 9003, 202}:

Node-SID 102 steers the traffic to B.

Adj-SID 9003 load-balances the traffic to C or F.

From either C or F, Node-SID 202 steers the traffic to PE2.

In conclusion, the traffic is load-balanced between the ABCDE and ABFE paths, as desired.

4.2. Traffic Engineering with Bandwidth Admission Control

The implementation of bandwidth admission control within a network (and its possible routing consequence which consists in routing along explicit paths where the bandwidth is available) requires a capacity planning process.

The spreading of load among ECMP paths is a key attribute of the capacity planning processes applied to packet-based networks.

The first sub-section details the capacity planning process and the role of ECMP load-balancing. We highlight the relevance of SR in that context.

The next two sub-sections document two use-cases of SR-based traffic engineering with bandwidth admission control.

The second sub-section documents a concrete SR applicability involving centralized-based admission control. This is often referred to as the "SDN/SR use-case".

The third sub-section introduces a future research topic involving the notion of residual bandwidth introduced in [I-D.atlas-mpls-te-express-path].

4.2.1. Capacity Planning Process

Capacity Planning anticipates the routing of the traffic matrix onto the network topology, for a set of expected traffic and topology

variations. The heart of the process consists in simulating the placement of the traffic along ECMP-aware shortest-paths and accounting for the resulting bandwidth usage.

The bandwidth accounting of a demand along its shortest-path is a basic capability of any planning tool or PCE server.

For example, in the network topology described below, and assuming a default IGP metric of 1 and IGP metric of 2 for link GF, a 1600Mbps A-to-Z flow is accounted as consuming 1600Mbps on links AB and FZ, 800Mbps on links BC, BG and GF, and 400Mbps on links CD, DF, CE and EF.

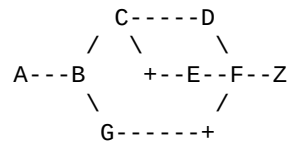


Figure 9: Capacity Planning an ECMP-based demand

ECMP is extremely frequent in SP, Enterprise and DC architectures and it is not rare to see as much as 128 different ECMP paths between a source and a destination within a single network domain. It is a key efficiency objective to spread the traffic among as many ECMP paths as possible.

This is illustrated in the below network diagram which consists of a subset of a network where already 5 ECMP paths are observed from A to M.

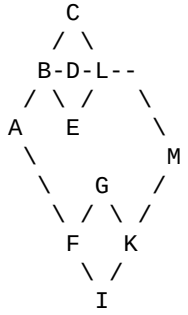


Figure 10: ECMP Topology Example

Segment Routing offers a simple support for such ECMP-based shortest-path placement: a node segment. A single node segment enumerates all the ECMP paths along the shortest-path.

When the capacity planning process detects that a traffic growth

scenario and topology variation would lead to congestion, a capacity increase is triggered and if it cannot be deployed in due time, a traffic engineering solution is activated within the network.

A basic traffic engineering objective consists of finding the smallest set of demands that need to be routed off their shortest path to eliminate the congestion, then to compute an explicit path for each of them and instantiating these traffic-engineered policies in the network.

Segment Routing offers a simple support for explicit path policy. Let us provide two examples based on Figure 10.

First example: let us assume that the process has selected the flow AM for traffic-engineering away from its ECMP-enabled shortest path and flow AM must avoid consuming resources on the LM and the FG links.

The solution is straightforward: A sends its M-destined traffic towards the nhop F with a two-label stack where the top label is the adjacent segment FI and the next label is the node segment to M. Alternatively, a three-label stack with adjacency segments FI, IK and KM could have been used.

Second example: let us assume that AM is still the selected flow but the constraint is relaxed to only avoid using resources from the LM link.

The solution is straightforward: A sends its M-destined traffic towards the nhop F with a one-label stack where the label is the node segment to M. Note that while the AM flow has been traffic-engineered away from its natural shortest-path (ECMP across three paths), the traffic-engineered path is still ECMP-aware and leverages two of the three initial paths. This is accomplished with a single-label stack and without the enumeration of one tunnel per path.

Under the light of these examples, Segment Routing offers an interesting solution for Capacity Planning because:

One node segment represents the set of ECMP-aware shortest paths.

Adjacency segments allow to express any explicit path.

The combination of node and adjacency segment allows to express any path without having to enumerate all the ECMP options.

The capacity planning process ensures that the majority of the traffic rides on node segments (ECMP-based shortest path), while a minority of the traffic is routed off its shortest-path.

The explicitly-engineered traffic (which is a minority) still benefits from the ECMP-awareness of the node segments within their segment list.

Only the head-end of a traffic-engineering policy maintains state. The midpoints and tail-ends do not maintain any state.

4.2.2. SDN/SR use-case

The heart of the application of SR to the SDN use-case lies in the SDN controller, also called Stateful PCE ([I-D.ietf-pce-stateful-pce]).

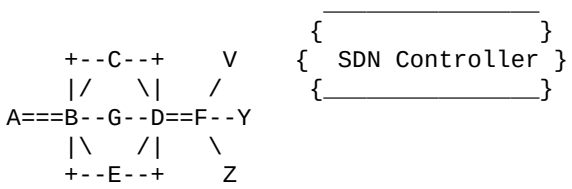
The SDN controller is responsible to control the evolution of the traffic matrix and topology. It accepts or denies the addition of new traffic into the network. It decides how to route the accepted traffic. It monitors the topology and upon failure, determines the minimum traffic that should be rerouted on an alternate path to alleviate a bandwidth congestion issue.

The algorithms supporting this behavior are a local matter of the SDN controller and are outside the scope of this document.

The means of collecting traffic and topology information are the same as what would be used with other SDN-based traffic-engineering solutions (e.g. [RFC5101] and [I-D.ietf-idr-ls-distribution].

The means of instantiating policy information at a traffic-engineering head-end are the same as what would be used with other SDN-based traffic-engineering solutions (e.g.: [I-D.ward-i2rs-framework], [I-D.crabbe-pce-pce-initiated-lsp] and [draft-msiva-pce-pcep-segment-routing-extensions-00]).

4.2.2.1. Illustration



SDN/SR use-case

Let us assume that in the above network diagram:

An SDN Controller (SC) is connected to the network and is able to retrieve the topology and traffic information, as well as set traffic-engineering policies on the network nodes.

The operator (likely via the SDN Controller) as provisioned the Node-SIDs 101, 102, 103, 104, 105, 106, 107, 201, 202 and 203 respectively at nodes A, B, C, D, E, F, G, V, Y and Z.

All the links have the same BW (e.g. 10G) and IGP cost (e.g. 10) except the links BG and GD which have IGP cost 50.

Each described node connectivity is formed as a bundle of two links, except (B, G) and (G, D) which are formed by a single link each.

Flow FV is traveling from A to destinations behind V.

Flow FY is traveling from A to destinations behind Y.

Flow FZ is traveling from A to destinations behind Z.

The SDN Controller has admitted all these flows and has let A apply the default SR policy: "map a flow onto its ECMP-aware shortest-path".

In this example, this means that A respectively maps the flows FV onto segment list {201}, FY onto segment list {202} and FZ onto segment list {203}.

In this example, the reader should note that the SDN Controller knows what A would do and hence knows and controls that none of these flows are mapped through G.

Let us describe what happens upon the failure of one of the two links E-D.

The SDN Controller monitors the link-state database and detects a congestion risk due to the reduced capacity between E and D. Specifically, SC updates its simulation of the traffic according to the policies he instructed the network to use and discovers that too much traffic is mapped on the remaining link E-D.

The SDN Controller then computes the minimum number of flows that should be deviated from their existing path. For example, let us assume that the flow FZ is selected.

The SDN controller then computes an explicit path for this flow. For example, let us assume that the chosen path is ABGDFZ.

The SDN controller then maps the chosen path into an SR-based policy. In our example, the path ABGDFZ is translated into a segment list {107, 203}. Node-SID steers the traffic along ABG and then Node-SID 203 steers the traffic along GDFZ.

The SDN controller then applies the following traffic-engineering policy at A: "map any packet of the classified flow FZ onto segment-list {107, 203}". The SDN Controller uses PCEP extensions to instantiate that policy at A ([draft-msiva-pce-pcep-segment-routing-extensions-00]).

As soon as A receives the PCEP message, it enforces the policy and the traffic classified as FZ is immediately mapped onto segment list {107, 203}.

This immediately eliminate the congestion risk. Flows FV and FY were untouched and keep using the ECMP-aware shortest-path. The minimum amount of traffic was rerouted (FZ). No signaling hop-by-hop through the network from A to Z is required. No admission control hop-by-hop is required. No state needs to be maintained by B, G, D, F or Z. The only maintained state is within the SDN controller and the head-end node (A).

4.2.2.2. Benefits

In the context of Centralized-Based Optimization and the SDN use-case, here are the benefits provided by the SR architecture:

- Explicit routing capability with or without ECMP-awareness.
- No signaling hop-by-hop through the network.
- State is only maintained at the policy head-end. No state is maintained at mid-points and tail-ends.
- Automated guaranteed FRR for any topology (Section 3.
- Optimum virtualization: the policy state is in the packet header and not in the intermediate node along the policy. The policy is completely virtualized away from midpoints and tail-ends.
- Highly responsive to change: the SDN Controller only needs to apply a policy change at the head-end. No delay is lost programming the midpoints and tail-end along the policy.

4.2.2.3. Dataset analysis

A future version of this document will report some analysis of the application of the SDN/SR use-case to real operator data sets.

A first, incomplete, report is available here below.

4.2.2.3.1. Example 1

The first data-set consists in a full-mesh of 12000 explicitly-routed tunnels observed on a real network. These tunnels resulted from distributed headend-based CSPF computation.

We measured that only 65% of the traffic is riding on its shortest path.

Three well-known defects are illustrated in this data set:

- The lack of ECMP support in explicitly--routed tunnels: ATM-alike traffic-steering mechanisms steer the traffic along a non-ECMP path.
- The increase of the number of explicitly-routed non-ECMP tunnels to enumerate all the ECMP options.
- The inefficiency of distributed optimization: too much traffic is riding off its shortest path.

We applied the SDN/SR use-case to this dataset. This means that:

- The distributed CSPF computation is replaced by centralized optimization and BW admission control, supported by the SDN Controller.
  - As part of the optimization, we also optimized the IGP-metrics such as to get a maximum of traffic load-spread among ECMP-paths by default.
- The traffic-engineering policies are supported by SR segment-lists.

As a result, we measured that 98% of the traffic would be kept on its normal policy (ride shortest-path) and only 2% of the traffic requires a path away from the shortest-path.

Let us highlight a few benefits:

98% of the traffic-engineering head-end policies are eliminated.

Indeed, by default, an SR-capable ingress edge node maps the traffic on a single Node-ID to the egress edge node. No configuration or policy needs to be maintained at the ingress edge node to realize this.

100% of the states at mid/tail nodes are eliminated.

4.2.3. Residual Bandwidth

The notion of Residual Bandwidth (RBW) is introduced by [I-D.atlas-mpls-te-express-path].

A future version of this document will describe the SR/RBW research opportunity.

5. Service chaining

Segment routing can be used to steer packets through services offered by middleboxes to perform specific actions such as DPI, accounting, etc.

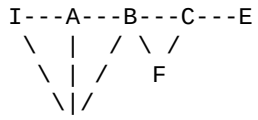


Figure 11

For example, as illustrated in Figure 11, an ingress node I selects an egress node E for a packet P. An application however requires that P undergoes a specific treatment (DPI, firewalling, ...) offered by a node D, reachable in the SR domain. In the SR architecture, this application can be supported through the use of a service segment with a local scope to D, say SS, following the nodal segment which corresponds to D. The Ingress box keeps the control of the egress node through which the packet needs to exit the network, by placing a nodal segment identifying the egress node after the service segment.

This would be achieved by letting I forward the packet P with the following sequence of segments: {D,SS,E}. D is a nodal segment, SS is the service segment corresponding to the service to apply to the packet P, and E is the nodal segment corresponding to the egress node selected by I for that packet.

6. OAM

6.1. Monitoring a remote bundle

This section documents a few representative SR/OAM use-cases.

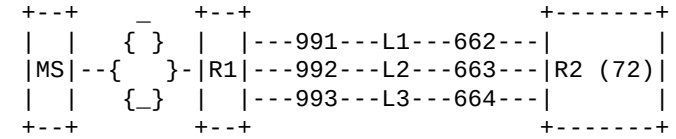


Figure 12: Probing all the links of a remote bundle

In the above figure, a monitoring system (MS) needs to assess the dataplane availability of all the links within a remote bundle connected to routers R1 and R2.

The monitoring system retrieves the segment information from the IGP LSDB and appends the following segment list: {72, 662, 992, 664} on its IP probe (whose source and destination addresses are the address of AA).

MS sends the probe to its connected router. If the connected router is not SR compliant, a tunneling technique can be used to tunnel the SR-based probe to the first SR router. The SR domain forwards the probe to R2 (72 is the node segment of R2). R2 forwards the probe to R1 over link L1 (adjacency segment 662). R1 forwards the probe to R2 over link L2 (adjacency segment 992). R2 forwards the probe to R1 over link L3 (adjacency segment 664). R1 then forwards the IP probe to AA as per classic IP forwarding.

6.2. Monitoring a remote peering link

In Figure 6, node A can monitor the dataplane liveness of the unidirectional peering link from C to D of AS2 by sending an IP probe with destination address A and segment list {101, 9001}. Node-SID 101 steers the probe to C and External Adj-SID 9001 steers the probe from C over the desired peering link to D of AS2. The SR header is removed by C and D receives a plain IP packet with destination address A. D returns the probe to A through classic IP forwarding. BFD Echo mode ([RFC5880]) would support such liveliness unidirectional link probing application.

7. IANA Considerations

TBD

8. Manageability Considerations

TBD

9. Security Considerations

TBD

10. Acknowledgements

We would like to thank Dave Ward, Dan Frost, Stewart Bryant, Thomas Telkamp, Ruediger Geib and Les Ginsberg for their contribution to the content of this document.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast Services", BCP 126, RFC 4786, December 2006.

[RFC5101] Claise, B., "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", RFC 5101, January 2008.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

[RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, December 2008.

[RFC5443] Jork, M., Atlas, A., and L. Fang, "LDP IGP Synchronization", RFC 5443, March 2009.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.

[RFC6138] Kini, S. and W. Lu, "LDP IGP Synchronization for Broadcast Networks", RFC 6138, February 2011.

11.2. Informative References

[I-D.atlas-mpls-te-express-path]
Atlas, A., Drake, J., Giacalone, S., Ward, D., Previdi, S., and C. Filsfils, "Performance-based Path Selection for Explicitly Routed LSPs", draft-atlas-mpls-te-express-path-02 (work in progress), February 2013.

[I-D.crabbe-pce-pce-initiated-lsp]
Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-crabbe-pce-pce-initiated-lsp-01 (work in progress), April 2013.

[I-D.francois-sr-frr]
Francois, P., Filsfils, C., Bashandy, A., Previdi, S., and B. Decraene, "Segment Routing Fast Reroute", draft-francois-sr-frr-00 (work in progress), July 2013.

[I-D.gredler-isis-label-advertisement]
Gredler, H., Amante, S., Scholl, T., and L. Jalil, "Advertising MPLS labels in IS-IS", draft-gredler-isis-label-advertisement-03 (work in progress), May 2013.

[I-D.ietf-idr-ls-distribution]
Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", draft-ietf-idr-ls-distribution-03 (work in progress), May 2013.

[I-D.ietf-pce-stateful-pce]
Crabbe, E., Medved, J., Minei, I., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-04 (work in progress), May 2013.

[I-D.previdi-isis-segment-routing-extensions]
Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., and S. Litkowski, "IS-IS Extensions for Segment Routing", draft-previdi-isis-segment-routing-extensions-03 (work in progress), October 2013.

[I-D.previdi-isis-te-metric-extensions]
Previdi, S., Giacalone, S., Ward, D., Drake, J., Atlas, A., and C. Filsfils, "IS-IS Traffic Engineering (TE) Metric Extensions",

draft-previdi-isis-te-metric-extensions-03 (work in progress), February 2013.

[I-D.shakir-rtgwg-sr-performance-engineered-lsps]
Shakir, R., Vernals, D., and A. Capello, "Performance Engineered LSPs using the Segment Routing Data-Plane", draft-shakir-rtgwg-sr-performance-engineered-lsps-00 (work in progress), July 2013.

[I-D.sivabalan-pce-segment-routing]
Sivabalan, S., Medved, J., Filsfils, C., Crabbe, E., and R. Raszuk, "PCEP Extensions for Segment Routing", draft-sivabalan-pce-segment-routing-02 (work in progress), October 2013.

[I-D.ward-i2rs-framework]
Atlas, A., Nadeau, T., and D. Ward, "Interface to the Routing System Framework", draft-ward-i2rs-framework-00 (work in progress), February 2013.

[draft-filsfils-rtgwg-segment-routing-01]
Filsfils, C. and S. Previdi, "Segment Routing Architecture", October 2013.

[draft-filsfils-spring-segment-routing-ldp-interop-00]
Filsfils, C. and A. Bashandy, "Segment Routing interoperability with LDP", October 2013.

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

[draft-psenak-ospf-segment-routing-extensions-00]
Psenak, P. and S. Previdi, "OSPF Segment Routing Extensions", May 2013.

[draft-rtgwg-bgp-pic-01.txt]
Filsfils, C., Bashandy, A., and P. Mohapatra, "BGP Prefix Independent Convergence", March 2013.

Authors' Addresses

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

Email: cfilsfil@cisco.com

Pierre Francois (editor)
IMDEA Networks
Leganes,
ES

Email: pierre.francois@imdea.org

Stefano Previdi
Cisco Systems, Inc.
Via Del Serafico, 200
Rome 00142
Italy

Email: sprevidi@cisco.com

Bruno Decraene
Orange
FR

Email: bruno.decraene@orange.com

Stephane Litkowski
Orange
FR

Email: stephane.litkowski@orange.com

Martin Horneffer
Deutsche Telekom
Hammer Str. 216-226
Muenster 48153
DE

Email: Martin.Horneffer@telekom.de

Igor Milojevic
Telekom Srbija
Takovska 2
Belgrade
RS

Email: igormilojevic@telekom.rs

Rob Shakir
British Telecom
London
UK

Email: rob.shakir@bt.com

Saku Ytti
TDC Oy
Mechelininkatu 1a
TDC 00094
FI

Email: saku@ytti.fi

Wim Henderickx
Alcatel-Lucent
Copernicuslaan 50
Antwerp 2018
BE

Email: wim.henderickx@alcatel-lucent.com

Jeff Tantsura
Ericsson
300 Holger Way
San Jose, CA 95134
US

Email: Jeff.Tantsura@ericsson.com

Sriganesh Kini
Ericsson
300 Holger Way
San Jose, CA 95134
US

Email: sriganesh.kini@ericsson.com

Edward Crabbe
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: edc@google.com