Network Working Group   D. Dukes, Ed.
Internet-Draft   C. Filsfils
Intended status: Standards Track   Cisco Systems
Expires: December 7, 2018   G. Dawra
                                                              LinkedIn
                                                                X. Xu
                                                               Alibaba
                                                              D. Voyer
                                                           Bell Canada
                                                         P. Camarillo
                                                               F. Clad
                                                         Cisco Systems
                                                            S. Salsano
                                              Univ. of Rome Tor Vergata
                                                          June 5, 2018

                      SR For SDWAN: VPN with Underlay SLA
                       draft-dukes-spring-sr-for-sdwan-00

Abstract

   This document describes how SR enables underlay Service Level
   Agreements (SLA) to a VPN with scale and security while ensuring
   service opacity. This solution applies to Over-The-Top VPN (OTT VPN)
   and Software-Defined WAN (SDWAN).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes how SR enables underlay SLA to a VPN with scale and security while ensuring service opacity. This solution applies to Over-The-Top VPN (OTT VPN) with SLA differentiation, and Software-Defined WAN (SDWAN) with SLA differentiation.

The body of this text uses SRv6 for illustration. A similar solution leveraging SR-MPLS is illustrated in an appendix.

This document assumes familiarity with the following IETF documents:

o Segment Routing Architecture [I-D.ietf-spring-segment-routing]

o Segment Routing with MPLS data plane
   [I-D.ietf-spring-segment-routing-mpls]

o IPv6 Segment Routing Header [I-D.ietf-6man-segment-routing-header]

o SRv6 Network Programming
   [I-D.filsfils-spring-srv6-network-programming]

o Segment Routing Policy For Traffic Engineering
   [I-D.filsfils-spring-segment-routing-policy]

o IS-IS Extensions to Support Segment Routing over IPv6 Dataplane
   [I-D.bashandy-isis-srv6-extensions]

For clarity, this version of the document uses the SDWAN example with SRv6 to illustrate how SR can be used to provide underlay SLA to overlay services. The journey of a packet from the left site to the right site of the SDWAN Overlay is described. The solution applies similarly for the return path.

2. Requirements Notation

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 [RFC2119].

3. Single Provider

3.1. Directly Connected CE to PE

```
+-----------+                                  /----------/
|   SDWAN   |                                 /          /
|  Control  |                                / [SDWAN-C]...
+-----------+                               /          / :
                                          /----------/   :
                                                         :
           /------------------------------------------/   :
 +---------+   /                                  /      :
 |  SDWAN  |  /    [A]-----[E1]***********[E2]--------[Z] /      :
 | Overlay | /                :             :           /      :
 +---------+/                 :......       :          /      :
        /----------------------:--------:----------/      :
                               :        :                :
                               :        :                :
              /---------/      :        :                :
 +-----------+   /        /    :        :                :
 |    SP     |  /  [SR-C] /    :        :                :
 |  Control  | /    :    /     :        :                :
 +-----------+ /------:---/    ....:         ..          :
               :        :            :                   :
               :        :            :                   :
 +-----------+     :   /---:------------:------------/      :
 |    SP     |     :  /    :            :           /      :
 |  Underlay |     : /   [C1]----------[C2]        /      :
 +-----------+     :/        \         /          /      :
                   :          \      /--/        /      :
                  /:           \    /          /      :
                 / :...........[C3]...........................:
                /                              /
               /-----------------------------/
```

**** = logical connection
:... = physical connection, between layers
/--\ = physical connection, within a layer

Figure 1: SDWAN Reference Diagram

An SDWAN overlay is composed of two sites A and Z, connected to the
Internet via edge nodes E1 and E2 respectively. E1 and E2 (customer
edge nodes) are connected via a Service Provider (SP) underlay to
form the VPN between the sites.

C1, C2 and C3 are nodes of the SP underlay, where C1 and C2 are
Provider Edge nodes. ISIS is deployed in the SP underlay with the
same cost on each link.

E1 and E2 connect to C1 and C2 respectively. The shortest path from
C1 to C2 is the best-effort path. The explicit path C1-C3-C2 is the

low-latency path. By default, traffic transported from C1 to C2
follows the best-effort path. By default, an SDWAN cannot benefit
from the low-latency path from C1 to C2.

The address of A is 10.10.0.10/32 and the address of Z is
10.26.0.26/32. E1 and E2 respectively advertise 10.10/16 and
10.26/16 to the SDWAN controller SDWAN-C via a secure channel over
the Internet. The solution is applicable to any traffic exchanged
between the sites, including IPv4, IPv6 or L2. For clarity, a single
example with IPv4 in the SDWAN Overlay is used.

The SP operates an SR controller SR-C capable of computing
constrained paths from C1 to C2.

3.2. Best-effort Underlay Transport

Let's consider the path taken by traffic from A to Z, across the
SDWAN, between nodes E1 and E2 with addresses E1:: and E2::
respectively.

Host A sends a packet P to Z via E1. Packet P has source address
10.10.0.10 and destination address 10.26.0.26, illustrated as P
(10.10.0.10,10.26.0.26)(payload). E1, upon receipt of P, determines
E2 is the edge node to be used to reach Z. Edge node E1 encrypts,
encapsulates and forwards the packet P toward E2 and Z, and it is
handled as follow:

o Between A and E1 : P (10.10.0.10,10.26.0.26)(Payload)

o Between E1 and C1 : P
   (E1::,E2::,NH=ESP)(NH=IPv4,(10.10.0.10,10.26.0.26)(Payload))

   * Note that ESP tunnel mode encapsulation, encryption and
      authentication is assumed but not required.

o Between C1 and C2 : P
   (E1::,E2::,NH=ESP)(NH=IPv4,(10.10.0.10,10.26.0.26)(Payload))

o Between C2 and E2 : P (E1::,E2::,NH=ESP)(
   NH=IPv4,(10.10.0.10,10.26.0.26)(Payload))

o Between E2 and Z : P (10.10.0.10,10.26.0.26)(Payload)

This example illustrates that, classically (i.e., without the SR
solution described in this document), the SDWAN cannot leverage the
rich infrastructure of the SP to meet its needs. The SP is
constrained to offer best-effort transit which does not reflect the
capabilities of its infrastructure.

3.3. **SR for Underlay SLA Differentiation**

   SR enables the SDWAN to steer selected flows through selected
   transport paths of the SP, using the same example in Figure 1.

   This small example, with only 3 SP routers, assumes all three support
   SRv6. As explained in
   [I-D.filsfils-spring-srv6-network-programming], a typical deployment
   would only require SRv6 at a few strategic waypoints deployed through
   the network.

   It also assumes ISIS supports the lightweight SRv6 extension
   described in [I-D.bashandy-isis-srv6-extensions].

   The illustration convention from
   [I-D.filsfils-spring-srv6-network-programming] is used such that:

   o SRv6 SID Cj:: is explicitly instantiated at node Cj and bound to
      the END.PSP function.

   o SRv6 SID C1::B21 is a Binding SID (BSID) explicitly instantiated
      at headend C1 and bound to the SRTE policy <C3::, C2::> toward
      endpoint C2.

      * Note the return direction would use a BSID C2::B11, bound at
         headend C2, to the SRTE policy <C3::, C1::> toward endpoint C1.

   The Control-Plane (CP) workflow that leads to the instantiation of
   this Binding SID will be explained in the Control-Plane section.

   Let's again consider the path from A to Z for a packet P, but this
   time E1 has been configured by SDWAN-C to steer packet P into a
   preferred low-latency path of the SP bound to the binding SID C1:B21.

   o Between A and E1

      * P (10.10.0.10,10.26.0.26)(payload)

   o Between E1 and C1

      * P (E1::,C1::B21; NH=SRH)(E2::,C1::B21; SL=1;
         NH=ESP)(NH=IPv4(10.10.0.10,10.26.0.26)(Payload))

   When the Binding SID C1::B21 is processed at C1, the SR TE Policy is
   selected and the SRH for SID list <C3::,C2::> is inserted into P:

   o Between C1 and C3

```
   *  P (E1::,C3::;NH=SRH)(E2::,C2::,C3::; SL=2;NH=ESP)
      (NH=IPv4(10.10.0.10,10.26.0.26)(Payload))
```

At C3, the SegmentsLeft is decremented as the END SID C3:: is
processed, and C2:: is placed in the destination address:

o Between C3 and C2

```
   *  P (E1::,C2::;NH=SRH)(E2::,C2::,C3::; SL=1;NH=ESP)
      (NH=IPv4(10.10.0.10,10.26.0.26)(Payload))
```

At C2, the SegmentsLeft is decremented to 0, and penultimate segment
pop is applied as the END SID C2:: is processed and E2:: is placed in
the destination address while the SRH is removed:

o Between C2 and E2

```
   *  P (E1::,E2::,NH=ESP)(NH=IPv4(10.10.0.10,10.26.0.26)(Payload))
```

Finally, E2 decrypts the packet and strips the outer header to
forward the original packet to Z:

o Between E2 and Z

```
   *  P (10.10.0.10,10.26.0.26)(Payload)
```

The SDWAN edge nodes (E1,E2) maintain their existing behavior of

o Ingress Edge Node: classify ingress traffic, determining the
   egress edge node, selecting a local output interface, secure the
   traffic, and forward to the chosen egress edge node.

o Egress Edge Node: decapsulate, decrypt and forward on the internal
   network.

The only change is that the Ingress node now monitors and selects an
SRv6 binding SID then pushes an SRH with two SIDs.

Note as well that the ingress and egress edge nodes never see the
actual SID list used by the SP to deliver the preferred path. A
variation of this design allows for the BSID to be kept in the packet
so that the egress node can detect which packets have been steered on
which preferred path (for accounting or monitoring purposes).

This is a fairly simple example of how SRv6 binding SIDs and SR TE
policies may be used to provide multiple diverse paths for SDWAN
traffic traversing a single provider network.

3.4. Accounting

As per SRv6 network programming
[I-D.filsfils-spring-srv6-network-programming], each SRTE policy and
its bound BSID is associated with a unique traffic counter. This
allows the SP to implement various forms of billing and reporting to
the customer of the preferred path.

3.5. Security

The domain of trust security solution documented in
[I-D.filsfils-spring-srv6-network-programming] is utilized.

Specifically SEC1, SEC2 and SEC3 guarantee that external traffic to
the SP cannot exercise the SID's of the SP.

The following behavior is added: the ACL implementing SEC1 and SEC2
on node C1 is updated to specifically allow traffic from E1:: to
C1::B21.

Only the SDWAN edge that has ordered the preferential service can use
it.

Any other customer of the SP is unable to use the preferential path
bound to BSID C1::B21.

The SDWAN site that has ordered the preferential service is unable to
directly program the network of the SP using the internal SID's of
the SP. The SDWAN edge node is restricted to the BSID, which
opacifies the SP operation.

3.6. Remotely Connected (to PE)

Well known authentication technology with details provided in
subsequent revisions will be added, detailing the scenario with SDWAN
edge nodes not directly connected to the SP node terminating the
binding SID.

4. Multiple Providers

Well known authentication technology with details provided in
subsequent revisions will be added, detailing the scenario with SDWAN
edge nodes connected to the SP node offering binding SID via an
intermediate SP.

5. Control Plane

The SDWAN overlay in Figure 1 is managed by an SDWAN controller,
SDWAN-C.

The control protocols used by the SDWAN-C to signal the site routes,
the BSID's and the site policies (which traffic on which BSID when)
securely over the SP network to E1 and E2 is outside the scope of
this document.

The SP underlay operates its internal SR deployment with an SR
controller (SR-C). SR-C interacts with the SP's network (Cj) through
standardized protocols (PCE[RFC4674] , PCEP [RFC5440]/[RFC4657], BGP
RR[RFC4456], BGP-TE [I-D.ietf-idr-segment-routing-te-policy], BGP-LS
[RFC7752])

Most likely, the SP would operate its underlay SLA service with a
service controller (SERV-C) that is separate from SR-C. To simplify
the illustration, this text assumes that SERV-C and SR-C are
integrated.

This section describes the high-level interaction between these
controllers for the low-latency use-case described in this document,
where an enterprise operator installs a policy in the SDWAN-C
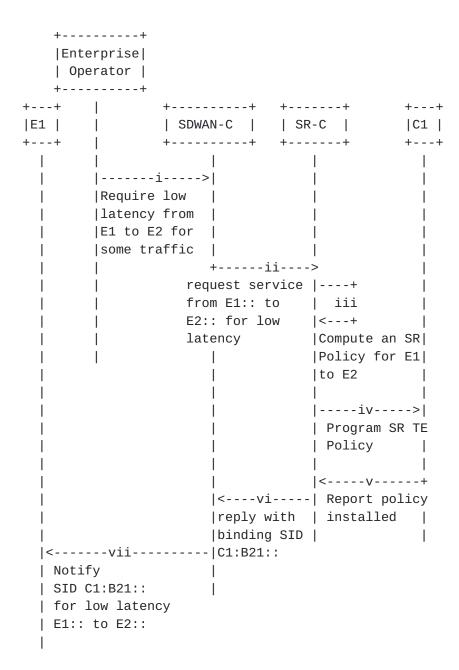requiring a low latency service between E1 and E2.

```
      +----------+
      |Enterprise|
      | Operator |
      +----------+
  +---+     |          +----------+   +-------+      +---+
  |E1 |     |          | SDWAN-C  |   | SR-C  |      |C1 |
  +---+     |          +----------+   +-------+      +---+
    |       |              |              |            |
    |       |-------i----->|              |            |
    |       |Require low   |              |            |
    |       |latency from  |              |            |
    |       |E1 to E2 for  |              |            |
    |       |some traffic  |              |            |
    |       |                  +------ii---->           |
    |       |             request service |----+        |
    |       |             from E1:: to    |  iii        |
    |       |             E2:: for low    |<---+        |
    |       |             latency         |Compute an SR|
    |       |                 |           |Policy for E1|
    |       |                 |           |to E2        |
    |       |                 |           |             |
    |       |                 |           |-----iv----->|
    |       |                 |           | Program SR TE
    |       |                 |           | Policy      |
    |       |                 |           |             |
    |       |                 |           |<-----v------+
    |       |             |<----vi-----| Report policy
    |       |             |reply with  | installed    |
    |       |             |binding SID |              |
    |<-------vii----------|C1:B21::                    |
    | Notify              |
    | SID C1:B21::        |
    | for low latency
    | E1:: to E2::
    |
```

                    Figure 2: Controlplane Flow

   (i)   The enterprise operator requests a low-latency path from site
            E1 to site E2. It defines which traffic needs to be steered
            on this preferred path.

   (ii)   SDWAN-C requests a low-latency service from SR-C for the
            public address of E1 to the public address of E2.

   (iii)   SR-C computes an SR Policy to satisfy SDWAN-C's request:

    A. SR-C maps the E1 and E2 addresses to its managed nodes C1 and C2.

    B. SR-C statefully registers the SRTE policy from C1 to C2 for low-latency.

    C. SR-C computes the SID list fulfilling the SLA requirement (e.g. <C3::, C2::>). The stateful nature of the SRTE policy ensures that the SID list is updated whenever required due to network state change.

    D. SR-C binds a stable Binding SID C1::B21 to the SRTE policy.

(iv)   SR-C programs C1 with the computed SRTE policy and the selected BSID. Standardized protocols such as [I-D.ietf-idr-segment-routing-te-policy] or [RFC5440] are used.

(v)   C1 installs the policy in its dataplane and reports the status of the SRTE policy to SR-C using standardized protocols [RFC7752] or [RFC5440] and [I-D.negi-pce-segment-routing-ipv6].

(vi)   SR-C replies to SDWAN-C with BSID C1::B21

(vii) SDWAN-C programs E1 with the flow classification and steering policy to insert SRv6 SID C1::B21 on the appropriate traffic

6. Benefits

6.1. Scale

The SP network does not hold any per-SDWAN-flow state in the core of its network.

The SP network does not hold any complex L3-L7 flow classification at the edge of its network.

The SP network is unaware of any policy change of the SDWAN instance either in terms of which flow to classify, when to steer it and on which path.

The SP's role only consists in statefully maintaining SRTE policies at the edge of the network and maintaining a few 100's of SID's inside its core network. This is the stateless property of Segment Routing.

6.2. Privacy

The SP network does not share any information of its infrastructure,
topology, capacity, internal SID's.

The SDWAN instance does not share any information on its traffic
classification, steering policy and business logic.

6.3. Flexible Billing

The traffic destined to a BSID is individually accounted
[I-D.filsfils-spring-srv6-network-programming].

The SP and SDWAN instance can agree on various forms of billing for
the usage of the preferential path.

6.4. Security

By default, the SP's SR infrastructure is protected by the simple
domain of trust solution documented in
[I-D.filsfils-spring-srv6-network-programming].

A BSID (and the related preferential path) can only be accessed by
the specific SDWAN instance (and site) that ordered the service.

The security solution supports any SDWAN site connection type:
directly connected to the SP edge or not.

7. Appendix

7.1. Single Provider Example Using End.BM With an MPLS Core

Reusing the example from Section 3.3, with an SP core that supports
SR MPLS as defined in [I-D.ietf-spring-segment-routing-mpls]. Each
node C1, C2 and C3 have node-SIDs defined, resulting in labels 16001,
16002, and 16003 respectively. In such a case a packet from A to Z
has an SRv6 binding SID applied, associated with an SR policy at node
C1. Node C1 translates the binding SID to an MPLS label stack which
is pushed on the packet.

For example:

o SRv6 SID C1::B22 is a Binding SID (BSID) explicitly instantiated
   at headend C1 and bound to the SRTE policy <16003,16002> toward
   endpoint C2.

* Note the return direction would use a BSID C2::B12, bound at
  headend C2, to the SRTE policy <16003, 16001> toward endpoint
  C1.

Let's again consider the path from A to Z for a packet P where E1 has
been configured by SDWAN-C to steer packet P into a preferred low-
latency path of the SP bound to the binding SID C1::B22.

o Between A and E1

   * P (10.10.0.10,10.26.0.26)(payload)

o Between E1 and C1

   * P (E1::,C1::B22; NH=SRH)(E2::,C1::B22; SL=1;
     NH=ESP)(NH=IPv4(10.10.0.10,10.26.0.26)(Payload))

When the Binding SID C1::B22 is processed at C1, the SR TE Policy is
selected and the label stack for SID list <16003,16002> is pushed on
P. In practice 16003 is not pushed on the wire since it has been
distributed with PHP:

o Between C1 and C3

   * P (16002)(E1::,E2::;NH=ESP)
     (NH=IPv4(10.10.0.10,10.26.0.26)(Payload))

At C3, 16002 is popped and PHP requires no new label be pushed as P
is forwarded via the link to C2:

o Between C3 and C2

   * P (E1::,E2::;NH=ESP)(NH=IPv4(10.10.0.10,10.26.0.26)(Payload))

At C2, there is no more label stack so it forwards E2:: using the
global IPv6 table toward E2:

o Between C2 and E2

   * P (E1::,E2::,NH=ESP)(NH=IPv4(10.10.0.10,10.26.0.26)(Payload))

Finally, E2 decrypts the packet and strips the outer header to
forward the original packet to Z:

o Between E2 and Z

   * P (10.10.0.10,10.26.0.26)(Payload)

Again the only change is that the Ingress node now selects an MPLS binding SID for traffic taking the lowest latency path. The Ingress node has no knowledge of the SP underlay.

7.1.1. Accounting

As defined in Section 3.4

7.1.2. Security

The SRv6 SID is secured as defined in Section 3.5

MPLS is not enabled on the CE-to-PE link.

7.2. Single Provider Example Using MPLS From CE to PE for BSID

To be completed in future revisions

7.3. Single Provider Example Using SRMPLS Over UDP For CE to PE Not Directly Connected Over Internet

Reusing the example from Section 3.3, with an SP core that supports the SR MPLS extensions defined in [I-D.ietf-isis-segment-routing-extensions]. Each node C1, C2 and C3 have node-SIDs defined, resulting in labels 16001, 16002, and 16003 respectively.

Let's again consider the path from A to Z for a packet P, but this time E1 has been configured by SDWAN-C to steer packet P into a preferred low-latency path of the SP bound to an MPLS binding SID.

For example:

o MPLS label 24102 is a Binding SID (BSID) explicitly instantiated at headend C1 and bound to the SRTE policy <16003,16002> toward endpoint C2.

   * Note the return direction would use a BSID 24201, bound at headend C2, to the SRTE policy <16003, 16001> toward endpoint C1.

Let's again consider the path from A to Z for a packet P where E1 has been configured by SDWAN-C to steer packet P into a preferred low-latency path of the SP bound to the binding SID 24102.

o Between A and E1

   * P (10.10.0.10,10.26.0.26)(payload)

o Between E1 and C1, RFC7510 UDP encapsulation of MPLS is used to
   transport the MPLS labeled traffic.

   * P (E1::,C1::; NH=UDP)(24102,(10.10.0.10,10.26.0.26)(Payload))

When C1 receives P, it decapsulates the UDP header and pops the
Binding SID 24102, the SR TE Policy is selected and the label stack
for SID list <16003,16002> is pushed on P. In practice, 16003 is not
pushed on the wire since it has been distributed with PHP.

The remainder of this use case is identical to Section 7.1. The only
change is that the Ingress node now uses an MPLS binding SID
transported over UDP instead of an SRv6 binding SID, allowing the use
of an IPv6 or IPv4 transport from CE to PE.

7.3.1. Accounting

As defined in Section 3.4

7.3.2. Security

[RFC7510] defines the use of DTLS to authenticate and encrypt the
MPLS in UDP encapsulation between CE and PE. The authentication
ensures the source is authorized to send traffic to a binding SID.

After the source is verified as authorized, the source address and
Binding SID SHOULD be checked to determine if the source is permitted
to use the specific Binding SID in the MPLS label.

MPLS is not enabled on the CE-to-PE link.

8. IANA Considerations

No current considerations.

9. Security Considerations

A domain of trust is secured via methods documented in
[I-D.filsfils-spring-srv6-network-programming]

10. References

10.1. Informative References

   [I-D.bashandy-isis-srv6-extensions]
              Ginsberg, L., Bashandy, A., Filsfils, C., Decraene, B.,
              and Z. Hu, "IS-IS Extensions to Support Routing over IPv6
              Dataplane", draft-bashandy-isis-srv6-extensions-02 (work
              in progress), March 2018.

   [I-D.filsfils-spring-segment-routing-policy]
              Filsfils, C., Sivabalan, S., Hegde, S.,
              daniel.voyer@bell.ca, d., Lin, S., bogdanov@google.com,
              b., Krol, P., Horneffer, M., Steinberg, D., Decraene, B.,
              Litkowski, S., Mattes, P., Ali, Z., Talaulikar, K., Liste,
              J., Clad, F., and K. Raza, "Segment Routing Policy
              Architecture", draft-filsfils-spring-segment-routing-
              policy-06 (work in progress), May 2018.

   [I-D.ietf-6man-segment-routing-header]
              Previdi, S., Filsfils, C., Leddy, J., Matsushima, S., and
              d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header
              (SRH)", draft-ietf-6man-segment-routing-header-13 (work in
              progress), May 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-isis-segment-routing-extensions]
              Previdi, S., Ginsberg, L., Filsfils, C., Bashandy, A.,
              Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura,
              "IS-IS Extensions for Segment Routing", draft-ietf-isis-
              segment-routing-extensions-16 (work in progress), April
              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.

   [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.negi-pce-segment-routing-ipv6]
          Negi, M., Kaladharan, P., Dhody, D., and S. Sivabalan,
          "PCEP Extensions for Segment Routing leveraging the IPv6
          data plane", draft-negi-pce-segment-routing-ipv6-01 (work
          in progress), March 2018.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
          Requirement Levels", BCP 14, RFC 2119,
          DOI 10.17487/RFC2119, March 1997,
          <https://www.rfc-editor.org/info/rfc2119>.

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route
          Reflection: An Alternative to Full Mesh Internal BGP
          (IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006,
          <https://www.rfc-editor.org/info/rfc4456>.

[RFC4657] Ash, J., Ed. and J. Le Roux, Ed., "Path Computation
          Element (PCE) Communication Protocol Generic
          Requirements", RFC 4657, DOI 10.17487/RFC4657, September
          2006, <https://www.rfc-editor.org/info/rfc4657>.

[RFC4674] Le Roux, J., Ed., "Requirements for Path Computation
          Element (PCE) Discovery", RFC 4674, DOI 10.17487/RFC4674,
          October 2006, <https://www.rfc-editor.org/info/rfc4674>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation
          Element (PCE) Communication Protocol (PCEP)", RFC 5440,
          DOI 10.17487/RFC5440, March 2009,
          <https://www.rfc-editor.org/info/rfc5440>.

[RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black,
          "Encapsulating MPLS in UDP", RFC 7510,
          DOI 10.17487/RFC7510, April 2015,
          <https://www.rfc-editor.org/info/rfc7510>.

[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>.

10.2. Normative References

[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.

Authors' Addresses

   Darren Dukes (editor)
   Cisco Systems
   Canada

   Email: ddukes@cisco.com

   Clarence Filsfils
   Cisco Systems
   Belgium

   Email: cfilsfil@cisco.com

   Gaurav Dawra
   LinkedIn
   USA

   Email: gdawra@linkedin.com

   Xiaohu Xu
   Alibaba
   China

   Email: xiaohu.xxh@alibaba-inc.com

   Daniel Voyer
   Bell Canada
   Canada

   Email: daniel.voyer@bell.ca

Pablo Camarillo Garvia
Cisco Systems
Spain

Email: pcamaril@cisco.com

Francois Clad
Cisco Systems
France

Stefano Salsano
Univ. of Rome Tor Vergata
Italy

Email: stefano.salsano@uniroma2.it