

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 7, 2020

H. Chen
Futurewei
M. Toy
Verizon
A. Wang
China Telecom
Z. Li
China Mobile
L. Liu
Fujitsu
X. Liu
Volta Networks
July 6, 2019

SR Path Ingress Protection
draft-chen-pce-sr-ingress-protection-00

Abstract

This document describes protocol extensions and procedures for protecting the ingress node of a Segment Routing (SR) path.

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 <https://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 January 7, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminologies	3
3. SR Path Ingress Protection Example	3
4. Behavior after Ingress Failure	4
5. Extensions to PCE	5
5.1. Capability for SR Path Ingress Protection	5
5.2. SR Path Ingress Protection	6
5.2.1. Traffic-Description sub-TLV	7
5.2.2. Primary-Ingress sub-TLV	10
5.2.3. Service sub-TLV	11
5.2.4. Downstream-Node sub-TLV	12
6. IANA Considerations	13
7. Security Considerations	13
8. Acknowledgements	13
9. References	13
9.1. Normative References	13
9.2. Informative References	14
Authors' Addresses	15

1. Introduction

Fast protection of a transit node of a Segment Routing (SR) path is described in [I-D.bashandy-rtgwg-segment-routing-ti-lfa] and [I-D.hu-spring-segment-routing-proxy-forwarding]. However, these documents do not discuss the procedures for fast protection of the ingress node of an SR path.

This document fills that void and specifies protocol extensions and procedures for fast protection of the ingress node of an SR path. Ingress node and ingress as well as fast protection and protection will be used exchangeably.

2. Terminologies

The following terminologies are used in this document.

SR: Segment Routing

SRv6: SR for IPv6

SRH: Segment Routing Header

SID: Segment Identifier

CE: Customer Edge

PE: Provider Edge

LFA: Loop-Free Alternate

TI-LFA: Topology Independent LFA

TE: Traffic Engineering

BFD: Bidirectional Forwarding Detection

VPN: Virtual Private Network

L3VPN: Layer 3 VPN

FIB: Forwarding Information Base

PLR: Point of Local Repair

BGP: Border Gateway Protocol

IGP: Interior Gateway Protocol

OSPF: Open Shortest Path First

IS-IS: Intermediate System to Intermediate System

3. SR Path Ingress Protection Example

Figure 1 shows an example of protecting ingress PE1 of a SR path, which is from ingress PE1 to egress PE3.

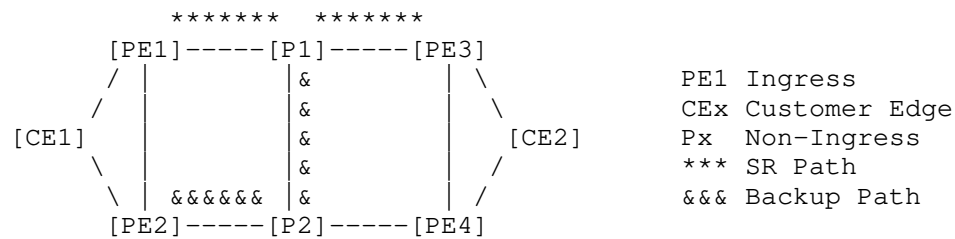


Figure 1: Protecting SR Path Ingress PE1

In normal operations, CE1 sends the traffic with destination PE3 to ingress PE1, which imports the traffic into the SR path.

When CE1 detects the failure of ingress PE1, it switches the traffic to backup ingress PE2, which imports the traffic from CE1 into a backup SR path. In one option, this backup path is from the backup ingress PE2 to ingress PE1's next hop (or endpoint) node P1, where the traffic is "merged" into the SR path, and then sent to egress PE3.

In another option, the backup path is from the backup ingress PE2 to the egress PE3. When the traffic is imported into the backup path, it is sent to the egress PE3 along the path.

4. Behavior after Ingress Failure

After failure of the ingress of an SR path happens, there are a couple of different ways to detect the failure. In each way, there may be some specific behavior for the traffic source (e.g., CE1) and the backup ingress (e.g., PE2).

In one way, the traffic source (e.g., CE1) is responsible for fast detecting the failure of the ingress (e.g., PE1) of an SR path. Fast detecting the failure means detecting the failure in a few or tens of milliseconds. The backup ingress (e.g., PE2) is ready to import the traffic from the traffic source into the backup SR path installed.

In normal operations, the source sends the traffic to the ingress of the SR path. When the source detects the failure of the ingress, it switches the traffic to the backup ingress, which delivers the traffic to the egress of the SR path via the backup SR path.

In another way, both the backup ingress and the traffic source are concurrently responsible for fast detecting the failure of the ingress of an SR path.

In normal operations, the source (e.g., CE1) sends the traffic to the ingress (e.g., PE1). It switches the traffic to the backup ingress (e.g., PE2) when it detects the failure of the ingress.

The backup ingress does not import any traffic from the source into the backup SR path in normal operations. When it detects the failure of the ingress, it imports the traffic from the source into the backup SR path.

5. Extensions to PCE

PCC runs on each of the edge nodes of a network normally. PCE runs on a server as a controller to communicate with PCCs. PCE and PCCs work together to support protection for the ingress of a SR path.

5.1. Capability for SR Path Ingress Protection

When a PCE and a PCC establish a PCEP session between them, they exchange their capabilities of supporting protection for the ingress node of an SR path/tunnel.

A new sub-TLV called SR_INGRESS_PROTECTION_CAPABILITY is defined. It is included in the PATH_SETUP_TYPE_CAPABILITY TLV with PST = TBD1 (suggested value 2 for backup SR path/tunnel) in the OPEN object, which is exchanged in Open messages when a PCC and a PCE establish a PCEP session between them. Its format is illustrated below.

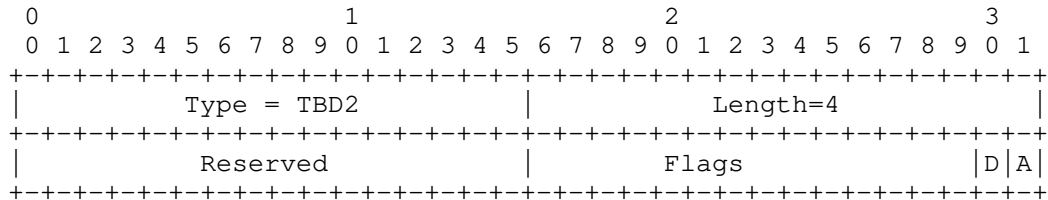


Figure 2: SR_INGRESS_PROTECTION_CAPABILITY sub-TLV

Type: TBD2 is to be assigned by IANA.

Length: 4.

Reserved: 2 octets. Must be set to zero in transmission and ignored on reception.

Flags: 2 octets. Two flags are defined.

- o D flag: A PCC sets this flag to 1 to indicate that it is able to detect its adjacent node's failure quickly.

- o A flag: A PCE sets this flag to 1 to request a PCC to let the forwarding entry for the backup SR path/tunnel be Active.

A PCC, which supports ingress protection for a SR tunnel/path, sends a PCE an Open message containing SR_INGRESS_PROTECTION_CAPABILITY sub-TLV. This sub-TLV indicates that the PCC is capable of supporting the ingress protection for a SR tunnel/path.

A PCE, which supports ingress protection for a SR tunnel/path, sends a PCC an Open message containing SR_INGRESS_PROTECTION_CAPABILITY sub-TLV. This sub-TLV indicates that the PCE is capable of supporting the ingress protection for a SR tunnel/path.

Assume that both a PCC and a PCE support SR_PCE_CAPABILITY, that is that each of the Open messages sent by the PCC and PCE contains PATH-SETUP-TYPE-CAPABILITY TLV with a PST list containing PST=1 and a SR-PCE-CAPABILITY sub-TLV.

If a PCE receives an Open message without a SR_INGRESS_PROTECTION_CAPABILITY sub-TLV from a PCC, then the PCE MUST not send the PCC any request for ingress protection of a SR path/tunnel.

If a PCC receives an Open message without a SR_INGRESS_PROTECTION_CAPABILITY sub-TLV from a PCE, then the PCC MUST ignore any request for ingress protection of a SR path/tunnel from the PCE.

If a PCC sets D flag to zero, then the PCE SHOULD send the PCC an Open message with A flag set to one. When the PCE sends the PCC a message for initiating a backup SR path/tunnel, the PCC SHOULD let the forwarding entry for the backup SR path/tunnel be Active.

5.2. SR Path Ingress Protection

A new sub-TLV called SR_INGRESS_PROTECTION is defined. When a PCE sends a PCC a PCInitiate message for initiating a backup SR path/tunnel to protect the primary ingress node of a primary SR path/tunnel, the message contains this TLV in the RP/SRP object. Its format is illustrated below.

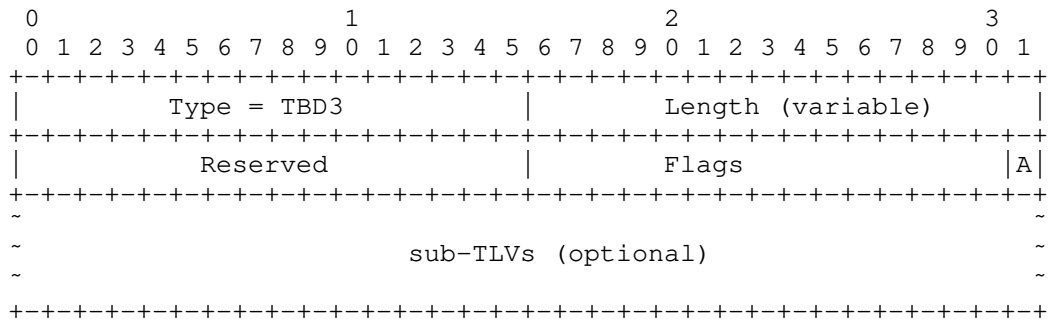


Figure 3: SR_INGRESS_PROTECTION sub-TLV

Type: TBD3 is to be assigned by IANA.

Length: Variable.

Reserved: 2 octets. Must be set to zero in transmission and ignored on reception.

Flags: 2 octets. One flag is defined.

- o A flag: A PCE sets this flag to 1 to request a PCC to let the forwarding entry for the backup SR path/tunnel be Active.

Four optional sub-TLVs are defined.

5.2.1. Traffic-Description sub-TLV

A Traffic-Description sub-TLV describes the traffic to be imported into a backup SR path/tunnel. Its format is illustrated below.

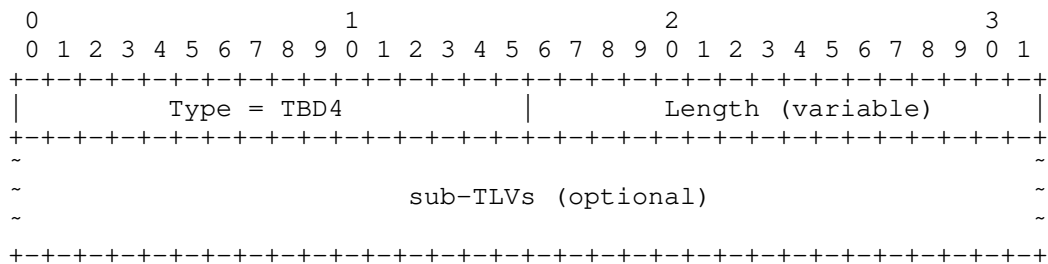


Figure 4: Traffic-Description sub-TLV

Type: TBD4 is to be assigned by IANA.

Length: Variable.

Two optional sub-TLVs are defined. One is FEC sub-TLV and the other interface sub-TLV.

A FEC sub-TLV describes the traffic to be imported into the backup SR path/tunnel. It is an IP prefix with an optional virtual network ID. It has two formats: one for IPv4 and the other for IPv6, which are illustrated below.

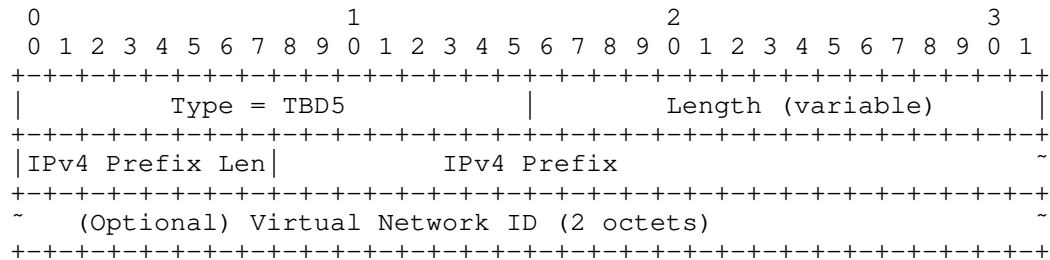


Figure 5: IPv4 FEC sub-TLV

Type: TBD5 is to be assigned by IANA.

Length: Variable.

IPv4 Prefix Len: Indicates the length of the IPv4 Prefix.

IPv4 Prefix: IPv4 Prefix rounded to octets.

Virtual Network ID: 2 octets. This is optional. It indicates the ID of a virtual network.

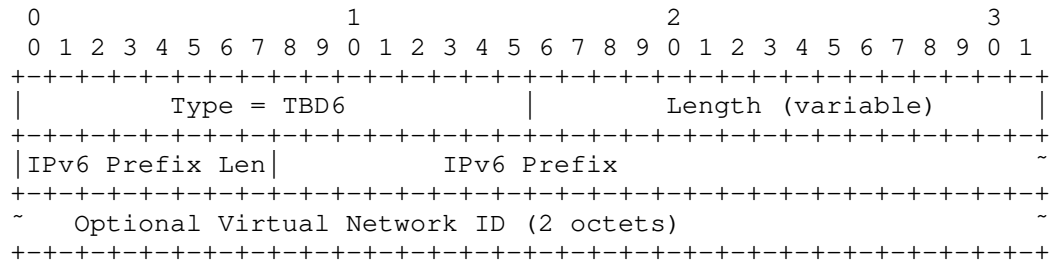


Figure 6: IPv6 FEC sub-TLV

Type: TBD6 is to be assigned by IANA.

Length: Variable.

IPv6 Prefix Len: Indicates the length of the IPv6 Prefix.

IPv6 Prefix: IPv6 Prefix rounded to octets.

Virtual Network ID: 2 octets. This is optional. It indicates the ID of a virtual network.

An Interface sub-TLV indicates the interface from which the traffic is received and imported into the backup SR path/tunnel. It has three formats: one for interface index, the other two for IPv4 and IPv6 address, which are illustrated below.

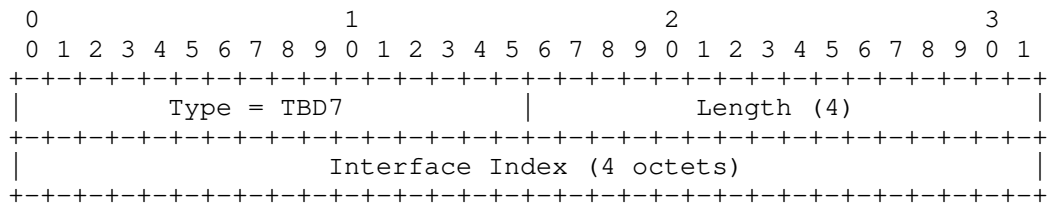


Figure 7: Interface Index sub-TLV

Type: TBD7 is to be assigned by IANA.

Length: 4.

Interface Index: 4 octets. It indicates the index of an interface.

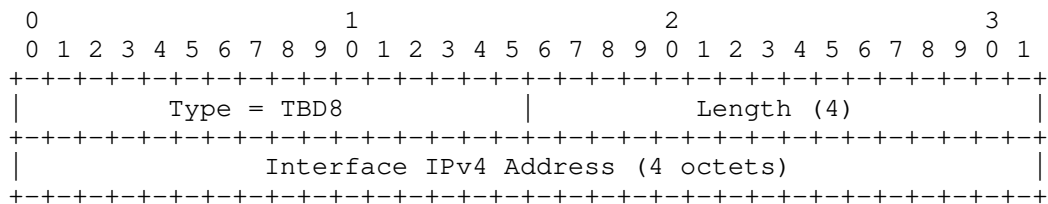


Figure 8: Interface IPv4 Address sub-TLV

Type: TBD8 is to be assigned by IANA.

Length: 4.

Interface IPv4 Address: 4 octets. It represents the IPv4 address of an interface.

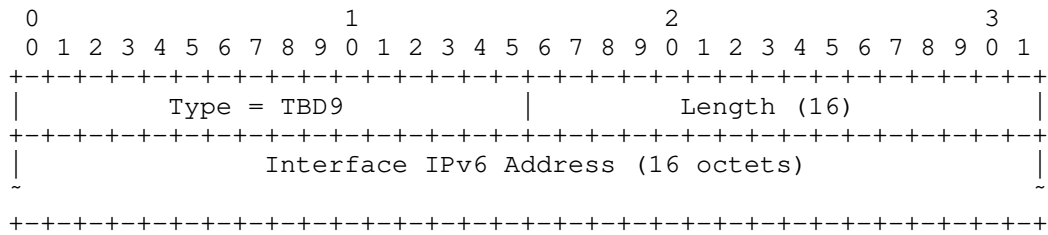


Figure 9: Interface IPv6 Address sub-TLV

Type: TBD9 is to be assigned by IANA.

Length: 16.

Interface IPv6 Address: 16 octets. It represents the IPv6 address of an interface.

5.2.2. Primary-Ingress sub-TLV

A Primary-Ingress sub-TLV indicates the IP address of the primary ingress node of a primary SR path/tunnel. It has two formats: one for primary ingress node IPv4 address and the other for primary ingress node IPv6 address, which are illustrated below.

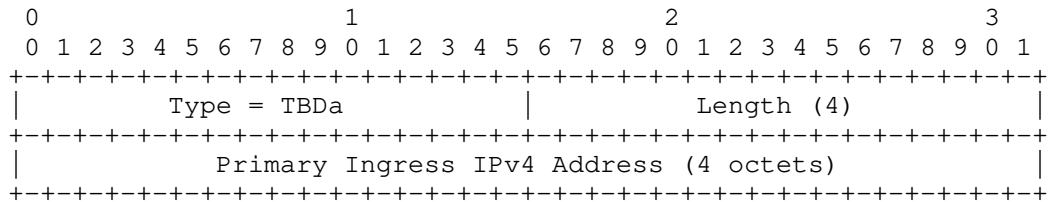


Figure 10: Primary Ingress IPv4 Address sub-TLV

Type: TBDA is to be assigned by IANA.

Length: 4.

Primary Ingress IPv4 Address: 4 octets. It represents an IPv4 host address of the primary ingress node of a SR path/tunnel.

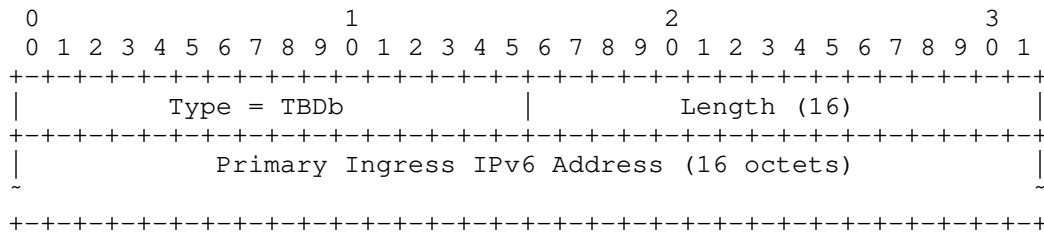


Figure 11: Primary Ingress IPv6 Address sub-TLV

Type: TBDb is to be assigned by IANA.

Length: 16.

Primary Ingress IPv6 Address: 16 octets. It represents an IPv6 host address of the primary ingress node of a SR path/tunnel.

5.2.3. Service sub-TLV

A Service sub-TLV contains a service ID or label to be added into a packet to be carried by a SR path/tunnel. It has two formats: one for the service identified by a label and the other for the service identified by a service identifier (ID) of 32 or 128 bits, which are illustrated below.

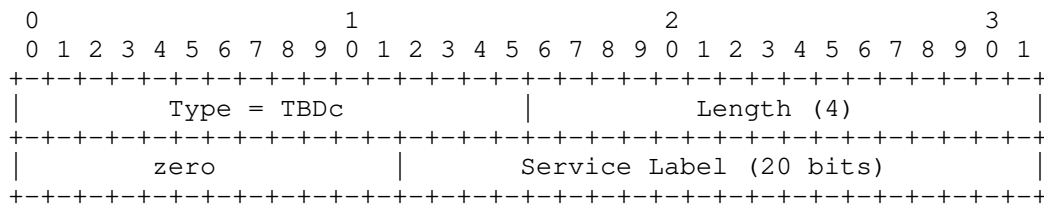


Figure 12: Service Label sub-TLV

Type: TBDC is to be assigned by IANA.

Length: 4.

Service Label: the least significant 20 bits. It represents a label of 20 bits.

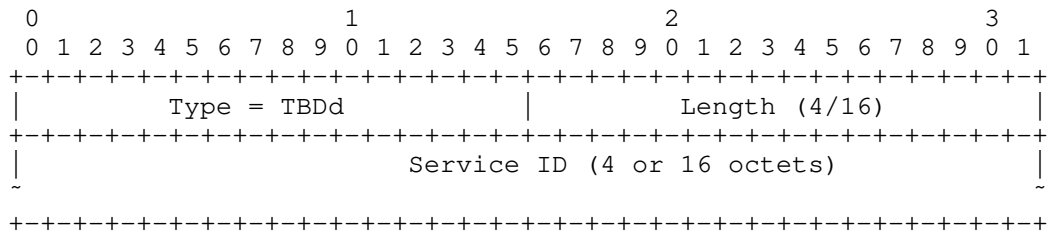


Figure 13: Service ID sub-TLV

Type: TBDD is to be assigned by IANA.

Length: 4 or 16.

Service ID: 4 or 16 octets. It represents Identifier (ID) of a service in 4 or 16 octets.

5.2.4. Downstream-Node sub-TLV

A Downstream-Node sub-TLV gives the IP address of the downstream endpoint node of the primary ingress along the primary SR path/tunnel. It has two formats: one for downstream node IPv4 address and the other for downstream node IPv6 address, which are illustrated below.

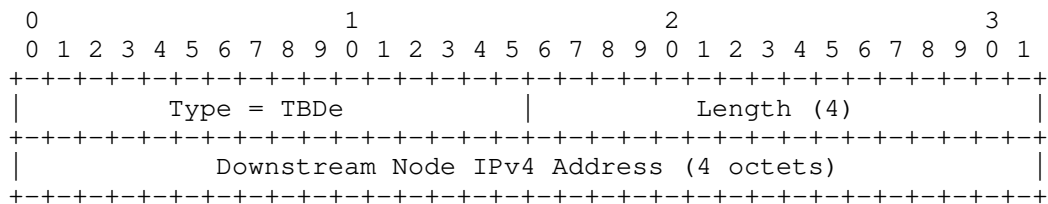


Figure 14: Downstream-Node IPv4 Address sub-TLV

Type: TBDe is to be assigned by IANA.

Length: 4.

Downstream Node IPv4 Address: 4 octets. It represents an IPv4 host address of the downstream endpoint node of the primary ingress node of a SR path/tunnel.

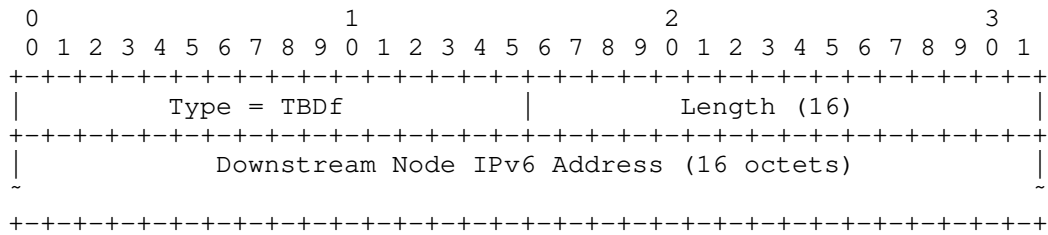


Figure 15: Downstream-Node IPv6 Address sub-TLV

Type: TBDf is to be assigned by IANA.

Length: 16.

Downstream Node IPv6 Address: 4 octets. It represents an IPv6 host address of the downstream endpoint node of the primary ingress node of a SR path/tunnel.

6. IANA Considerations

TBD

7. Security Considerations

TBD

8. Acknowledgements

TBD

9. References

9.1. Normative References

[I-D.bashandy-isis-srv6-extensions]

Psenak, P., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extensions to Support Routing over IPv6 Dataplane", draft-bashandy-isis-srv6-extensions-05 (work in progress), March 2019.

[I-D.hu-spring-segment-routing-proxy-forwarding]

Hu, Z., Chen, H., Yao, J., Bowers, C., and Y. Zhu, "SR-TE Path Midpoint Protection", draft-hu-spring-segment-routing-proxy-forwarding-03 (work in progress), April 2019.

- [I-D.ietf-isis-segment-routing-extensions]
Previdi, S., Ginsberg, L., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-25 (work in progress), May 2019.
- [I-D.ietf-ospf-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-ietf-ospf-segment-routing-extensions-27 (work in progress), December 2018.
- [I-D.li-ospf-ospfv3-srv6-extensions]
Li, Z., Hu, Z., Cheng, D., Talaulikar, K., and P. Psenak, "OSPFv3 Extensions for SRv6", draft-li-ospf-ospfv3-srv6-extensions-03 (work in progress), March 2019.
- [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>>.
- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <<https://www.rfc-editor.org/info/rfc7356>>.

9.2. Informative References

- [I-D.bashandy-rtgwg-segment-routing-ti-lfa]
Bashandy, A., Filsfils, C., Decraene, B., Litkowski, S., Francois, P., daniel.voyer@bell.ca, d., Clad, F., and P. Camarillo, "Topology Independent Fast Reroute using Segment Routing", draft-bashandy-rtgwg-segment-routing-ti-lfa-05 (work in progress), October 2018.
- [I-D.hegde-spring-node-protection-for-sr-te-paths]
Hegde, S., Bowers, C., Litkowski, S., Xu, X., and F. Xu, "Node Protection for SR-TE Paths", draft-hegde-spring-node-protection-for-sr-te-paths-05 (work in progress), July 2019.
- [I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d., bogdanov@google.com, b., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-03 (work in progress), May 2019.

- [I-D.sivabalan-pce-binding-label-sid]
Sivabalan, S., Filsfils, C., Tantsura, J., Hardwick, J.,
Previdi, S., and C. Li, "Carrying Binding Label/Segment-ID
in PCE-based Networks.", draft-sivabalan-pce-binding-
label-sid-06 (work in progress), February 2019.
- [RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching
(MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic
Class" Field", RFC 5462, DOI 10.17487/RFC5462, February
2009, <<https://www.rfc-editor.org/info/rfc5462>>.

Authors' Addresses

Huaimo Chen
Futurewei
Boston, MA
USA

Email: Huaimo.chen@futurewei.com

Mehmet Toy
Verizon
USA

Email: mehmet.toy@verizon.com

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing 102209
China

Email: wangaj.bri@chinatelecom.cn

Zhenqiang Li
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing 100053
China

Email: lizhengqiang@chinamobile.com

Lei Liu
Fujitsu
USA

Email: liulei.kddi@gmail.com

Xufeng Liu
Volta Networks
McLean, VA
USA

Email: xufeng.liu.ietf@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 17 May 2022

H. Chen
M. McBride
Futurewei
M. Toy
G. Mishra
Verizon Inc.
A. Wang
China Telecom
Z. Li
Y. Liu
China Mobile
B. Khasanov
Yandex LLC
L. Liu
Fujitsu
X. Liu
Volta Networks
13 November 2021

Path Ingress Protections
draft-chen-pce-sr-ingress-protection-07

Abstract

This document describes extensions to Path Computation Element (PCE) communication Protocol (PCEP) for fast protecting the ingress nodes of two types of paths or tunnels, which are Segment Routing (SR) paths and Bit Index Explicit Replication Tree/Traffic Engineering (BIER-TE) paths. The extensions comprise a foundation for protecting the ingress nodes of different types of paths. Based on this, the ingress protection of a new type of paths can be easily supported.

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 <https://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 17 May 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Terminologies	3
2. Path Ingress Protection Examples	4
2.1. SR Path Ingress Protection Example	4
2.2. BIER-TE Path Ingress Protection Example	5
3. Behavior around Ingress Failure	6
3.1. Source Detect	6
3.2. Backup Ingress Detect	6
3.3. Both Detect	7
4. Extensions to PCEP	7
4.1. Capabilities for Ingress Protection	7
4.1.1. Capability for Ingress Protection with Backup Ingress	7
4.1.2. Capability for Ingress Protection with Traffic Source	9
4.2. Extensions for Backup Ingress and Traffic Source	10
4.2.1. Extensions for Backup Ingress	10
4.2.2. Extensions for Traffic Source	16
5. Security Considerations	19
6. Acknowledgements	19
7. IANA Considerations	19
8. References	19
8.1. Normative References	19
8.2. Informative References	19
Authors' Addresses	20

1. Introduction

The fast protection of a transit node in each type of paths or tunnels have been proposed. For example, the fast protection of a transit node in a Segment Routing (SR) path or tunnel is described in [I-D.ietf-rtgwg-segment-routing-ti-lfa]. The fast protection of a transit node of a "Bit Index Explicit Replication" (BIER) Traffic Engineering (BIER-TE) path or tunnel is described in [I-D.chen-bier-te-frr]. [RFC8424] presents extensions to RSVP-TE for the fast protection of the ingress node of a traffic engineering (TE) Label Switching Path (LSP). However, these documents do not discuss any protocol extensions for the fast protection of the ingress node of an SR path/tunnel, a BIER-TE path/tunnel, or other type of paths/tunnels.

This document fills that void and specifies protocol extensions to Path Computation Element (PCE) communication Protocol (PCEP) [RFC5440] and [RFC9050] for fast protecting the ingress nodes of two types of paths: SR paths and BIER-TE paths. The extensions comprise a foundation for protecting the ingress nodes of different types of paths. Based on this, the ingress protection of a new type of paths can be easily supported.

Ingress node and ingress, fast protection and protection, path ingress protection and ingress protection, SR path and SR tunnel, as well as BIER-TE path and BIER-TE tunnel will be used exchangeably in the following sections.

1.1. Terminologies

The following terminologies are used in this document.

PCE: Path Computation Element or Path Computation Element server

PCEP: PCE communication Protocol

PCC: Path Computation Client

BIER: Bit Index Explicit Replication

BIFT: Bit Index Forwarding Table

CE: Customer Edge

PE: Provider Edge

TE: Traffic Engineering

SR: Segment Routing
 LFA: Loop-Free Alternate
 TI-LFA: Topology Independent LFA
 BFD: Bidirectional Forwarding Detection
 VPN: Virtual Private Network
 L3VPN: Layer 3 VPN
 FIB: Forwarding Information Base

2. Path Ingress Protection Examples

This section shows two examples of path ingress protection. One is SR path ingress protection, and the other is BIER-TE path ingress protection.

2.1. SR Path Ingress Protection Example

Figure 1 shows an example of protecting ingress PE1 of a SR path, which is from ingress PE1 to egress PE3 and represented by *** in the figure.

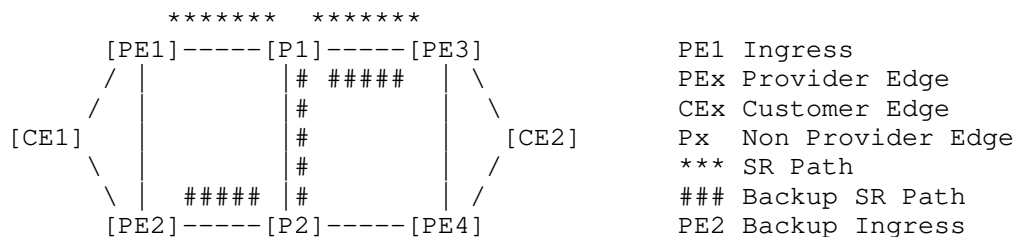


Figure 1: Protecting Ingress PE1 of SR Path

In normal operations, CE1 sends the traffic with destination PE3 to ingress PE1, which imports the traffic into the SR path.

When CE1 detects the failure of ingress PE1, it switches the traffic to backup ingress PE2, which imports the traffic from CE1 into a backup SR path. The backup path is from the backup ingress PE2 to the egress PE3 and represented by ### in the figure. When the traffic is imported into the backup path, it is sent to the egress PE3 along the path.

2.2. BIER-TE Path Ingress Protection Example

Figure 2 shows an example of protecting ingress PE1 of a BIER-TE path, which is from ingress PE1 to egress nodes PE3 and PE4. This primary BIER-TE path is represented by *** in the figure. The ingress of the primary BIER-TE path is called primary ingress.

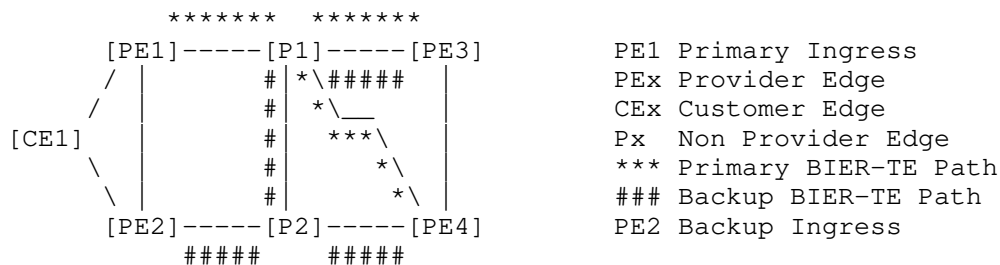


Figure 2: Protecting Ingress PE1 of BIER-TE Path

The backup BIER-TE path is from ingress PE2 to egress nodes PE3 and PE4, which is represented by ### in the figure. The ingress of the backup BIER-TE path is called backup ingress.

In normal operations, CE1 sends the packets with a multicast group and source to ingress PE1, which imports/encapsulates the packets into the BIER-TE path through adding a BIER-TE header. The header contains the BIER-TE path from ingress PE1 to egress nodes PE3 and PE4.

When CE1 detects the failure of ingress PE1 using a failure detection mechanism such as BFD, it switches the traffic to backup ingress PE2, which imports the traffic from CE1 into the backup BIER-TE path. When the traffic is imported into the backup path, it is sent to the egress nodes PE3 and PE4 along the path.

Given the traffic source (e.g., CE1), ingress (e.g., PE1) and egresses (e.g., PE3 and PE4) of the primary BIER-TE path, the PCE computes a backup ingress (e.g., PE2), a backup BIER-TE path from the backup ingress to the egresses, and sends the backup BIER-TE path to the PCC of the backup ingress. It also sends the information about the backup ingress, the primary ingress and the traffic to the PCC of the traffic source (e.g., CE1).

When the PCC of the traffic source receives the information about the backup ingress, the primary ingress and the traffic, it sets up the fast detection of the primary ingress failure and the switch over target backup ingress. This setup lets the traffic source node switch the traffic (to be sent to the primary ingress) to the backup ingress when it detects the failure of the primary ingress.

When the PCC of the backup ingress receives the backup BIER-TE path, it adds a forwarding entry into its BIFT. This entry encapsulates the packets from the traffic source in the backup BIER-TE path. This makes the backup ingress send the traffic received from the traffic source to the egress nodes via the backup BIER-TE path.

3. Behavior around Ingress Failure

This section describes the behavior of some nodes connected to the ingress before and after the ingress fails. These nodes are the traffic source (e.g., CE1) and the backup ingress (e.g., PE2). It presents three ways in which these nodes work together to protect the ingress. The first way is called source detect, where the traffic source is responsible for fast detecting the failure of the ingress. The second way is called backup ingress detect, in which the backup ingress is responsible for fast detecting the failure of the ingress. The third way is called both detect, where both the traffic source and the backup ingress are responsible for fast detecting the failure of the ingress.

3.1. Source Detect

In normal operations, i.e., before the failure of the ingress of a primary path such as a primary BIER-TE path, the traffic source sends the traffic to the ingress of the primary path. The backup ingress (e.g., PE2) is ready to import the traffic from the traffic source into the backup path such as the backup BIER-TE path installed.

When the traffic source detects the failure of the ingress, it switches the traffic to the backup ingress, which delivers the traffic to the egress nodes of the path via the backup path.

3.2. Backup Ingress Detect

The traffic source (e.g., CE1) always sends the traffic to both the ingress (e.g., PE1) of the primary path such as the primary BIER-TE path and the backup ingress (e.g., PE2).

The backup ingress does not import any traffic from the traffic source into the backup path such as the backup BIER-TE path in normal operations. When it detects the failure of the ingress of the primary path, it imports the traffic from the source into the backup path.

For the backup ingress to fast detect the failure of the primary ingress, it SHOULD directly connect to the primary ingress. When a PCE computes a backup ingress and a backup path, it SHOULD consider this.

3.3. Both Detect

In normal operations, i.e., before the failure of the ingress, the traffic source sends the traffic to the ingress of the primary path such as the primary BIER-TE path. When it detects the failure of the ingress, it switches the traffic to the backup ingress.

The backup ingress does not import any traffic from the traffic source into the backup path such as the backup BIER-TE path in normal operations. When it detects the failure of the ingress of the primary path, it imports the traffic from the source into the backup path.

4. Extensions to PCEP

A PCC runs on each of the edge nodes such as PEs of a network normally. A PCE runs on a server as a controller to communicate with PCCs. PCE and PCCs work together to support protection for the ingress of a path. The path is a SR path, a BIER-TE path, or a path of another type.

4.1. Capabilities for Ingress Protection

4.1.1. Capability for Ingress Protection with Backup Ingress

When a PCE and a PCC running on a backup ingress establish a PCEP session between them, they exchange their capabilities of supporting protection for the ingress node of each of different types of paths.

A new sub-TLV called INGRESS_PROTECTION_CAPABILITY is defined. It is included in the PATH_SETUP_TYPE_CAPABILITY TLV with PST = TBD1 (suggested value 2 for path ingress protection) in the OPEN object, which is exchanged in Open messages when a PCC and a PCE establish a PCEP session between them. Its format is illustrated below.

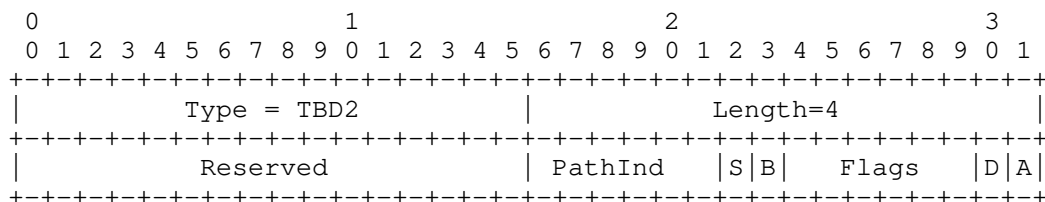


Figure 3: INGRESS_PROTECTION_CAPABILITY sub-TLV

Type: TBD2 is to be assigned by IANA.

Length: 4.

Reserved: 2 octets. MUST be set to zero in transmission and ignored on reception.

PathInd: 1 octet. Indicators for the types of paths whose ingress protections are supported. Two indicators are defined.

- o S : S = 1 indicating that the ingress protection of a SR path is supported.
- o B : B = 1 indicating that the ingress protection of a BIER-TE path is supported.

Flags: 1 octet. Two flags are defined.

- o D flag: A PCC sets this flag to 1 to indicate that it is able to detect its adjacent node's failure quickly.
- o A flag: A PCE sets this flag to 1 to request a PCC to let the forwarding entry for the backup path/tunnel be Active.

A PCC, which supports ingress protection for different types of paths, sends a PCE an Open message containing INGRESS_PROTECTION_CAPABILITY sub-TLV. This sub-TLV indicates that the PCC is capable of supporting the ingress protection for the types of paths.

For example, if a PCC supports ingress protection for SR path and BIER-TE path, the PCC sends a PCE an Open message containing INGRESS_PROTECTION_CAPABILITY sub-TLV with S = 1 and B = 1.

A PCE, which supports ingress protection for different types of paths, sends a PCC an Open message containing INGRESS_PROTECTION_CAPABILITY sub-TLV. This sub-TLV indicates that the PCE is capable of supporting the ingress protection for the types of paths.

If both a PCC and a PCE support INGRESS_PROTECTION_CAPABILITY, each of the Open messages sent by the PCC and PCE contains PATH-SETUP-TYPE-CAPABILITY TLV with a PST list containing PST=TBD1 and an INGRESS_PROTECTION_CAPABILITY sub-TLV.

If a PCE receives an Open message from a PCC without a INGRESS_PROTECTION_CAPABILITY sub-TLV indicating PCC's support for the ingress protection of a type of paths, then the PCE MUST not send the PCC any request for ingress protection of the type of paths.

If a PCC receives an Open message from a PCE without a INGRESS_PROTECTION_CAPABILITY sub-TLV indicating PCE's support for the ingress protection of a type of paths, then the PCC MUST ignore any request for ingress protection of the type of paths from the PCE.

If a PCC sets D flag to zero, then the PCE SHOULD send the PCC an Open message with A flag set to one and the fast detection of the failure of the primary ingress MUST be done by the traffic source. When the PCE sends the PCC a message for initiating a backup path, the PCC MUST let the forwarding entry for the backup path be Active.

4.1.2. Capability for Ingress Protection with Traffic Source

When a PCE and a PCC running on a traffic source node establish a PCEP session between them, they exchange their capabilities of supporting ingress protection.

The PCECC-CAPABILITY sub-TLV defined in [RFC9050] is included in the OPEN object in the PATH-SETUP-TYPE-CAPABILITY TLV, which is exchanged in Open messages when a PCC and a PCE establish a PCEP session between them.

A new flag bit P is defined in the Flags field of the PCECC-CAPABILITY sub-TLV:

- * P flag (for Ingress Protection): if set to 1 by a PCEP speaker, the P flag indicates that the PCEP speaker supports and is willing to handle the PCECC based central controller instructions for ingress protection. The bit MUST be set to 1 by both a PCC and a PCE for the PCECC ingress protection instruction download/report on a PCEP session.

4.2. Extensions for Backup Ingress and Traffic Source

This section specifies the extensions to PCEP for the backup ingress and the traffic source. The extensions let the traffic source

S1: fast detect the failure of the primary ingress and switch the traffic to the backup ingress when the traffic source detects the failure of the primary ingress, or

S2: always send the traffic to both the primary ingress and the backup ingress.

The extensions let the backup ingress

B1: always import the traffic received from the traffic source with possible service ID into the backup path, or

B2: import the traffic with possible service ID into the backup path when the backup ingress detects the failure of the primary ingress.

The following lists the combinations of Si and Bi (i = 1,2) for different ways of failure detects.

Source Detect: S1 and B1.

Backup Ingress Detect: S2 and B2.

Both Detect: S1 and B2.

4.2.1. Extensions for Backup Ingress

For the packets from the traffic source, if the primary ingress (i.e., the ingress of the primary path) encapsulates the packets with a service ID or label into the path, the backup ingress MUST have this service ID or label and encapsulates the packets with the service ID or label into the backup path when the primary ingress fails.

If the backup ingress is requested to detect the failure of the primary ingress, it MUST have the information about the primary ingress such as the address of the primary ingress.

A new sub-TLV called INGRESS_PROTECTION is defined. When a PCE sends a PCC a PCInitiate message for initiating a backup path to protect the primary ingress node of a primary path, the message contains this TLV in the RP/SRP object. Its format is illustrated below.

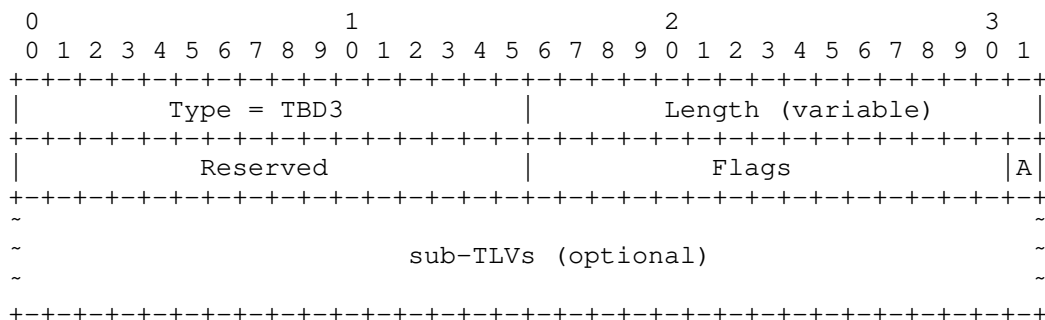


Figure 4: INGRESS_PROTECTION sub-TLV

Type: TBD3 is to be assigned by IANA.

Length: Variable.

Reserved: 2 octets. MUST be set to zero in transmission and ignored on reception.

Flags: 2 octets. One flag is defined.

A flag bit: it is set to 1 or 0 by PCE.

- o 1 is to request the backup ingress to let the forwarding entry for the backup path be Active always. In this case, the traffic source detects the failure of the primary ingress and switches the traffic to the backup ingress when it detects the failure.
- o 0 is to request the backup ingress to detect the failure of the primary ingress and let the forwarding entry for the backup path be Active when the primary ingress fails. In this case, the TLV includes the primary ingress address in a Primary-Ingress sub-TLV. The traffic source can send the traffic to both the primary ingress and the backup ingress. It may switch the traffic to the backup ingress from the primary ingress when it detects the failure of the primary ingress.

Three optional sub-TLVs are defined: Primary-Ingress sub-TLV, Service sub-TLV, and Traffic-Description sub-TLV. The Traffic-Description sub-TLV describes the traffic to be imported into the backup SR path. The Multicast Flow Specification TLV for IPv4 or IPv6, which is defined in [I-D.ietf-pce-pcep-flowspec], is used as a sub-TLV to indicate the traffic to be imported into the backup BIER-TE path.

4.2.1.1. Primary-Ingress sub-TLV

A Primary-Ingress sub-TLV indicates the IP address of the primary ingress node of a primary path. It has two formats: one for primary ingress node IPv4 address and the other for primary ingress node IPv6 address, which are illustrated below.

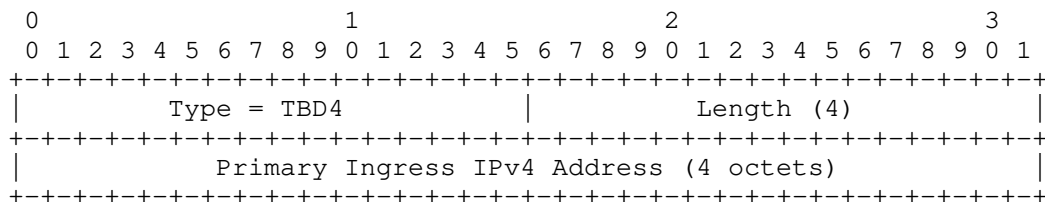


Figure 5: Primary Ingress IPv4 Address sub-TLV

Type: TBD4 is to be assigned by IANA.

Length: 4.

Primary Ingress IPv4 Address: 4 octets. It represents an IPv4 host address of the primary ingress node of a path.

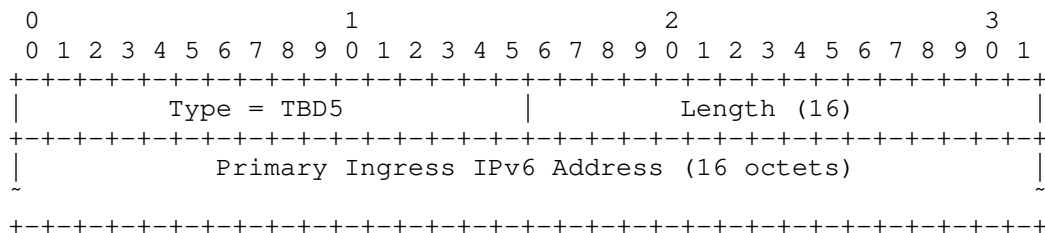


Figure 6: Primary Ingress IPv6 Address sub-TLV

Type: TBD5 is to be assigned by IANA.

Length: 16.

Primary Ingress IPv6 Address: 16 octets. It represents an IPv6 host address of the primary ingress node of a path.

4.2.1.2. Service sub-TLV

A Service sub-TLV contains a service ID or label to be added into a packet to be carried by a path. It has two formats: one for the service identified by a label and the other for the service identified by a service identifier (ID) of 32 or 128 bits, which are illustrated below.

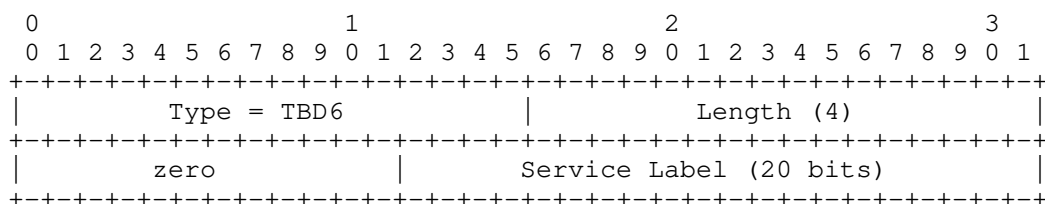


Figure 7: Service Label sub-TLV

Type: TBD6 is to be assigned by IANA.

Length: 4.

Service Label: the least significant 20 bits. It represents a label of 20 bits.

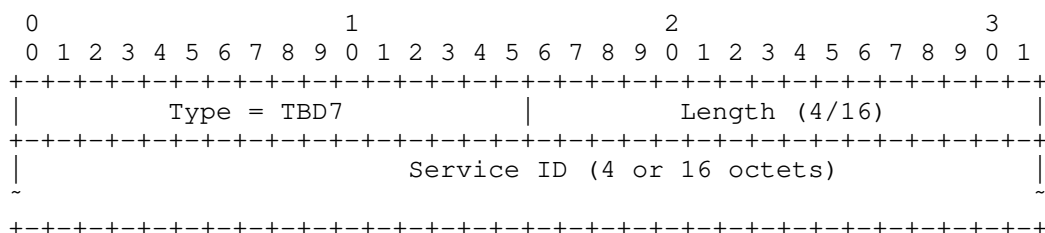


Figure 8: Service ID sub-TLV

Type: TBD7 is to be assigned by IANA.

Length: 4 or 16.

Service ID: 4 or 16 octets. It represents Identifier (ID) of a service in 4 or 16 octets.

4.2.1.3. Traffic-Description sub-TLV

A Traffic-Description sub-TLV describes the traffic to be imported into a backup SR path. Its format is illustrated below.

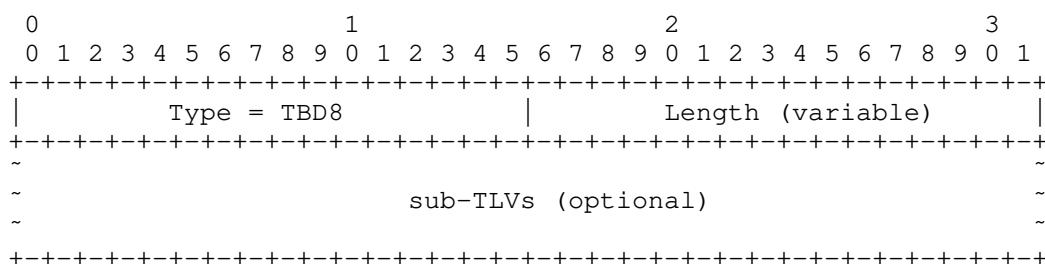


Figure 9: Traffic-Description sub-TLV

Type: TBD8 is to be assigned by IANA.

Length: Variable.

Two optional sub-TLVs are defined. One is FEC sub-TLV and the other interface sub-TLV.

A FEC sub-TLV describes the traffic to be imported into the backup path. It is an IP prefix with an optional virtual network ID. It has two formats: one for IPv4 and the other for IPv6, which are illustrated below.

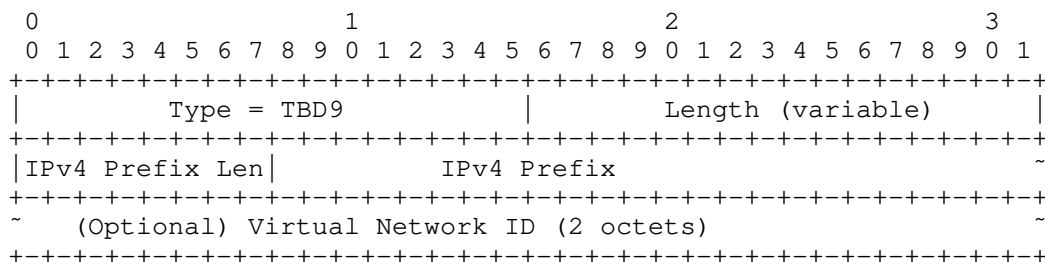


Figure 10: IPv4 FEC sub-TLV

Type: TBD9 is to be assigned by IANA.

Length: Variable.

IPv4 Prefix Len: Indicates the length of the IPv4 Prefix.

IPv4 Prefix: IPv4 Prefix rounded to octets.

Virtual Network ID: 2 octets. This is optional. It indicates the ID of a virtual network.

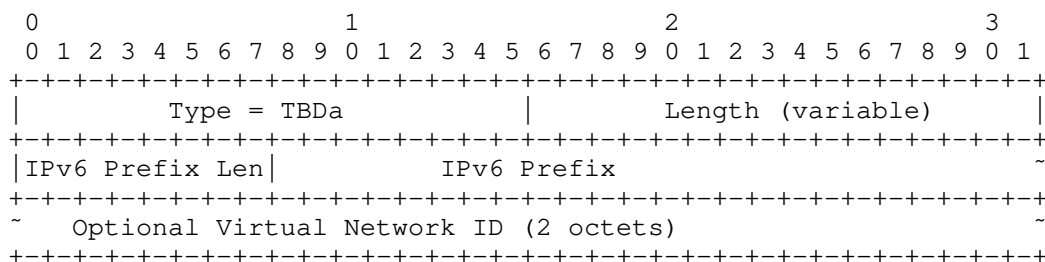


Figure 11: IPv6 FEC sub-TLV

Type: TBDA is to be assigned by IANA.

Length: Variable.

IPv6 Prefix Len: Indicates the length of the IPv6 Prefix.

IPv6 Prefix: IPv6 Prefix rounded to octets.

Virtual Network ID: 2 octets. This is optional. It indicates the ID of a virtual network.

An Interface sub-TLV indicates the interface from which the traffic is received and imported into the backup path. It has three formats: one for interface index, the other two for IPv4 and IPv6 address, which are illustrated below.

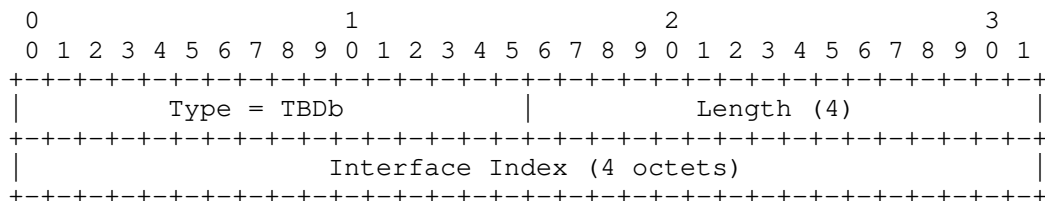


Figure 12: Interface Index sub-TLV

Type: TBDb is to be assigned by IANA.

Length: 4.

Interface Index: 4 octets. It indicates the index of an interface.

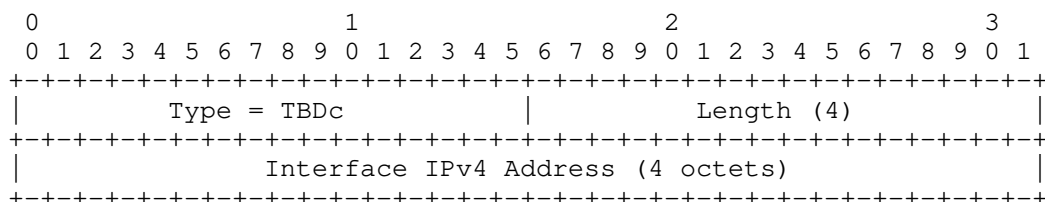


Figure 13: Interface IPv4 Address sub-TLV

Type: TBDc is to be assigned by IANA.

Length: 4.

Interface IPv4 Address: 4 octets. It represents the IPv4 address of an interface.

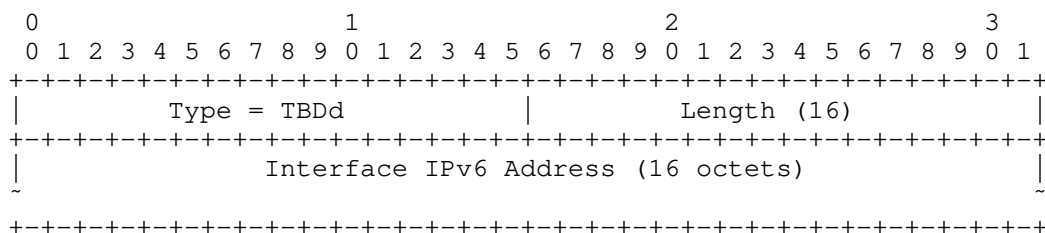


Figure 14: Interface IPv6 Address sub-TLV

Type: TBDd is to be assigned by IANA.

Length: 16.

Interface IPv6 Address: 16 octets. It represents the IPv6 address of an interface.

4.2.2. Extensions for Traffic Source

If the traffic source is requested to detect the failure of the primary ingress and switch the traffic (to be sent to the primary ingress) to the backup ingress when the primary ingress fails, it MUST have the information about the backup ingress, the primary ingress and the traffic. This information may be transferred via a CCI object for INGRESS-PROTECTION to the PCC of the traffic source node from a PCE.

If the traffic source PCC does not accept the request from the PCE or support the extensions, the PCE SHOULD have the information about the behavior of the traffic source configured such as whether it detects the failure of the primary ingress. Based on the information, the PCE instructs the backup ingress accordingly.

The Central Control Instructions (CCI) Object is defined in [RFC9050] for a PCE as a controller to send instructions for LSPs to a PCC. This document defines a new object-type (TBDt) for ingress protection based on the CCI object. The body of the object with the new object-type is illustrated below. The object may be in PCRpt, PCUpd, or PCInitiate message.

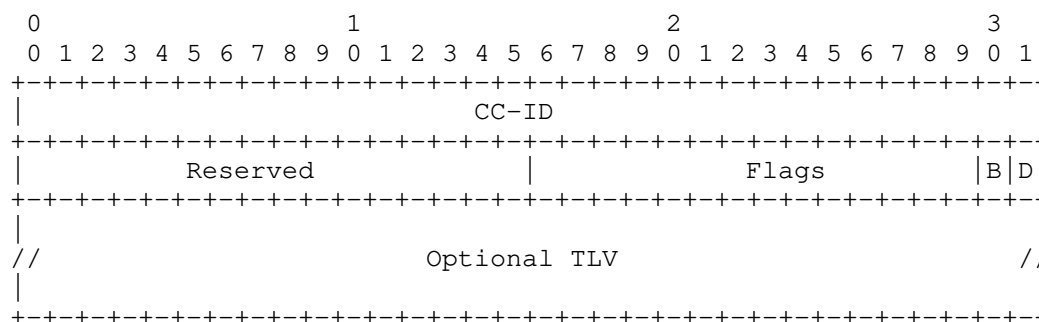


Figure 15: INGRESS-PROTECTION Object Body

CC-ID: It is the same as described in [RFC9050].

Flags: Two flag bits D and B are defined as follows:

D: D = 1 instructs the PCC of the traffic source to Detect the failure of the primary ingress and switch the traffic to the backup ingress when it detects the failure.

B: B = 1 instructs the PCC of the traffic source to send the traffic to Both the primary ingress and the backup ingress.

Optional TLV: Primary ingress TLV, backup ingress TLV, Traffic-Description TLV or Multicast Flow Specification TLV.

The primary ingress sub-TLV defined above is used as a TLV to contain the information about the primary ingress in the object. The Traffic-Description sub-TLV defined above is used as a TLV to contain the information about the traffic for a SR path in the object. The Multicast Flow Specification TLV for IPv4 or IPv6, which is defined in [I-D.ietf-pce-pcep-flowspec], is used to contain the information

about the traffic for a BIER-TE path in the object. A new TLV, called backup ingress TLV, is defined to contain the information about the backup ingress in the object.

4.2.2.1. Backup-Ingress TLV

A Backup-Ingress TLV indicates the IP address of the ingress node of a backup path. It has two formats: one for backup ingress node IPv4 address and the other for backup ingress node IPv6 address, which are illustrated below. They have the same format as the Primary-Ingress sub-TLVs.

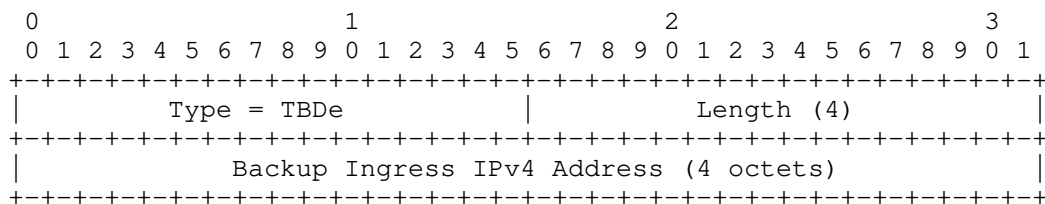


Figure 16: Backup Ingress IPv4 Address TLV

Type: TBDe is to be assigned by IANA.

Length: 4.

Backup Ingress IPv4 Address: 4 octets. It represents an IPv4 host address of the backup ingress.

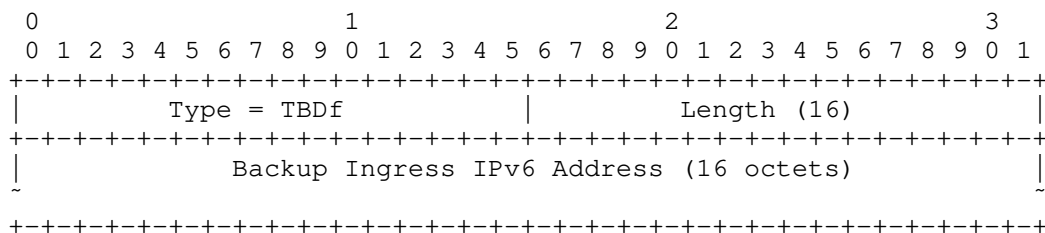


Figure 17: Backup Ingress IPv6 Address TLV

Type: TBDf is to be assigned by IANA.

Length: 16.

Backup Ingress IPv6 Address: 16 octets. It represents an IPv6 host address of the backup ingress node.

5. Security Considerations

TBD

6. Acknowledgements

The authors of this document would like to thank Dhruv Dhody and Robin Li for their reviews and comments.

7. IANA Considerations

TBD

8. References

8.1. Normative References

- [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>>.
- [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>>.
- [RFC7356] Ginsberg, L., Previdi, S., and Y. Yang, "IS-IS Flooding Scope Link State PDUs (LSPs)", RFC 7356, DOI 10.17487/RFC7356, September 2014, <<https://www.rfc-editor.org/info/rfc7356>>.
- [RFC8424] Chen, H., Ed. and R. Torvi, Ed., "Extensions to RSVP-TE for Label Switched Path (LSP) Ingress Fast Reroute (FRR) Protection", RFC 8424, DOI 10.17487/RFC8424, August 2018, <<https://www.rfc-editor.org/info/rfc8424>>.
- [RFC9050] Li, Z., Peng, S., Negi, M., Zhao, Q., and C. Zhou, "Path Computation Element Communication Protocol (PCEP) Procedures and Extensions for Using the PCE as a Central Controller (PCECC) of LSPs", RFC 9050, DOI 10.17487/RFC9050, July 2021, <<https://www.rfc-editor.org/info/rfc9050>>.

8.2. Informative References

[I-D.chen-bier-te-frr]

Chen, H., McBride, M., Liu, Y., Wang, A., Mishra, G. S., Fan, Y., Liu, L., and X. Liu, "BIER-TE Fast ReRoute", Work in Progress, Internet-Draft, draft-chen-bier-te-frr-01, 23 August 2021, <<https://www.ietf.org/archive/id/draft-chen-bier-te-frr-01.txt>>.

[I-D.ietf-pce-pcep-flowspec]

Dhody, D., Farrel, A., and Z. Li, "PCEP Extension for Flow Specification", Work in Progress, Internet-Draft, draft-ietf-pce-pcep-flowspec-13, 14 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-pce-pcep-flowspec-13.txt>>.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]

Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", Work in Progress, Internet-Draft, draft-ietf-rtgwg-segment-routing-ti-lfa-07, 29 June 2021, <<https://www.ietf.org/archive/id/draft-ietf-rtgwg-segment-routing-ti-lfa-07.txt>>.

[RFC5462] Andersson, L. and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field", RFC 5462, DOI 10.17487/RFC5462, February 2009, <<https://www.rfc-editor.org/info/rfc5462>>.

Authors' Addresses

Huaimo Chen
Futurewei
Boston, MA,
United States of America

Email: Huaimo.chen@futurewei.com

Mike McBride
Futurewei

Email: michael.mcbride@futurewei.com

Mehmet Toy
Verizon Inc.
United States of America

Email: mehmet.toy@verizon.com

Gyan S. Mishra
Verizon Inc.
13101 Columbia Pike
Silver Spring, MD 20904
United States of America

Phone: 301 502-1347
Email: gyan.s.mishra@verizon.com

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing
102209
China

Email: wangaj3@chinatelecom.cn

Zhenqiang Li
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing
100053
China

Email: lizhengqiang@chinamobile.com

Yisong Liu
China Mobile

Email: liuyisong@chinamobile.com

Boris Khasanov
Yandex LLC
Moscow

Email: bhassanov@yahoo.com

Lei Liu
Fujitsu
United States of America

Email: liulei.kddi@gmail.com

Xufeng Liu
Volta Networks
McLean, VA
United States of America

Email: xufeng.liu.ietf@gmail.com

PCE Working Group
Internet-Draft
Updates: 8231 (if approved)
Intended status: Standards Track
Expires: January 8, 2020

C. Li
H. Zheng
Huawei Technologies
S. Litkowski
Orange
July 7, 2019

Extension for Stateful PCE to allow Optional Processing of PCEP Objects.
draft-dhody-pce-stateful-pce-optional-04

Abstract

This document introduces a mechanism to mark some Path Computation Element (PCE) Communication Protocol (PCEP) objects as optional during PCEP messages exchange for the Stateful PCE model to allow relaxing some constraints.

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 <https://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 January 8, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. Overview	3
2.1. Usage Example	4
3. PCEP Extension	4
3.1. STATEFUL-PCE-CAPABILITY TLV	4
3.2. Handling of P flag	5
3.2.1. The PCRpt message	5
3.2.2. The PCUpd message and the PCInitiate message	5
3.3. Handling of I flag	5
3.3.1. The PCUpd message	6
3.3.2. The PCRpt message	6
3.3.3. The PCInitiate message	6
3.4. Delegation	6
3.5. Unknown Object Handling	6
4. Security Considerations	7
5. IANA Considerations	7
5.1. STATEFUL-PCE-CAPABILITY TLV	7
6. Manageability Considerations	7
6.1. Control of Function and Policy	7
6.2. Information and Data Models	8
6.3. Liveness Detection and Monitoring	8
6.4. Verify Correct Operations	8
6.5. Requirements On Other Protocols	8
6.6. Impact On Network Operations	8
7. Acknowledgments	8
8. References	8
8.1. Normative References	8
8.2. Informative References	9
Appendix A. Contributors	11
Authors' Addresses	11

1. Introduction

[RFC5440] describes the Path Computation Element communication Protocol (PCEP) which enables the communication between a Path Computation Client (PCC) and a Path Control Element (PCE), or between two PCEs based on the PCE architecture [RFC4655].

PCEP Extensions for Stateful PCE Model [RFC8231] describes a set of extensions to PCEP to enable active control of Multiprotocol Label Switching Traffic Engineering (MPLS-TE) and Generalized MPLS (GMPLS) tunnels. [RFC8281] describes the setup and teardown of PCE-initiated

LSPs under the active stateful PCE model, without the need for local configuration on the PCC, thus allowing for a dynamic network.

[RFC5440] defined P flag (Processing-Rule) as part of Common Object Header to allow a PCC to specify in a Path Computation Request (PCReq) message sent to a PCE whether the object must be taken into account by the PCE during path computation or is just optional. The I flag (Ignore) is used by a PCE in a Path Computation Reply (PCRep) message to indicate to a PCC whether or not an optional object was processed. Stateful PCE [RFC8231] specified that P and I flags of the PCEP objects defined in [RFC8231] is to be set to zero on transmission and ignored on receipt since they are exclusively related to path computation requests. The behavior for P and I flag in other messages defined in [RFC5440] and other extension was not specified. This document clarifies how the P and I flag could be used in the stateful PCE model to identify optional objects in the Path Computation State Report (PCRpt) [RFC8231], the Path Computation Update Request (PCUpd) [RFC8231], and the LSP Initiate Request (PCInitiate) [RFC8281] message.

This document updates [RFC8231] with respect to usage of P and I flag as well as the handling of unknown objects in the stateful PCEP message exchange.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Overview

[RFC5440] describes the handling on unknown objects as per the setting of the P flag for the PCReq message. Further [RFC8231] defined the usage of LSP Error Code TLV in PCRpt message in response to failed LSP Update Request via PCUpd message (for example, due to an unsupported object/TLV).

This document clarifies the procedure for marking some objects as 'optional to be processed' by the PCEP peer in the stateful PCEP messages. Further this document updates the procedure for handling unknown objects in the stateful PCEP messages based on the P flag.

2.1. Usage Example

The PCRpt message is used to report the current state of an LSP. As part of the message both the <intended-attribute-list> and <actual-attribute-list> is encoded (see [RFC8231]). For example, the <intended-attribute-list> could include the METRIC object to indicate a limiting constraint (B flag set) for the Path Delay Variation metric [RFC8233]. In some scenarios it would be useful to state that this limiting constraint can be relaxed by the PCE in case it cannot find a path. Similarly in case of an association groups [I-D.ietf-pce-association-group] such as Disjoint Association [I-D.ietf-pce-association-diversity], the PCE may need to completely relax the disjointness constraint in order to provide a path to all the LSPs that are part of the association. In these case it would be useful to mark the objects as 'optional' and it could be ignored by the PCEP peer. Also it would be used for the PCEP speaker to learn if the PCEP peer has relaxed the constraint and ignored the processing of the PCEP object.

Thus, this document simply clarifies, how the already existing P and I flag in the PCEP common object header could be used during the stateful PCEP message exchanges.

3. PCEP Extension

3.1. STATEFUL-PCE-CAPABILITY TLV

A PCEP speaker indicates its ability to support for handling P and I flag during the stateful PCEP message exchanges during the PCEP initialization phase, as follows. When the PCEP session is created, it sends an Open message with an OPEN object that contains the STATEFUL-PCE-CAPABILITY TLV, as defined in [RFC8231]. A new flag, the R (RELAX) flag, is introduced to this TLV to indicate support for relaxing the processing of some objects via the use of P and I flag in the PCEP common object header.

R (RELAX bit - TBD1): If set to 1 by a PCEP Speaker, the R flag indicates that the PCEP Speaker is willing to send and receive PCEP objects with handling of P and I flags in the PCEP common object header for stateful PCE messages. In case the bit is unset, it indicates that the PCEP Speaker would not handle P and I flags in the PCEP common object header for stateful PCE messages.

The R flag MUST be set by both a PCC and a PCE to indicate support for handling of P and I flag in the PCEP common object header to allow relaxing some constraints by marking objects as optional to process. If the PCEP speaker that did not set R flag but receives

PCEP objects with P or I bit set, MUST behave as per the processing rule in [RFC8231] i.e., the bits are simply ignored.

3.2. Handling of P flag

3.2.1. The PCRpt message

The P flag in the PCRpt message [RFC8231] allows a PCC to specify to a PCE whether the object must be taken into account by the PCE (during path computation or re-optimization) or is just optional. When the P flag is set in PCRpt message received on a PCEP session on which R bit was set by both peers, the object MUST be taken into account by the PCE. Conversely, when the P flag is cleared, the object is optional and the PCE is free to ignore it. The P flag for the mandatory objects LSP and ERO (intended path) MUST be set in the PCRpt message. If the mandatory object is received with the P flag set incorrectly according to the rules stated above, the receiving peer MUST send a PCErr message with Error-Type=10 (Reception of an invalid object) and Error-value=1 (reception of an object with P flag not set). By default, the PCC SHOULD set the P flag, unless a local configuration or local policy indicates that some constraints (corresponding PCEP objects) can be marked as optional and could be ignored by the PCE.

3.2.2. The PCUpd message and the PCInitiate message

The P flag in the PCUpd message [RFC8231] and the PCInitiate message [RFC8281] allows a PCE to specify to a PCC whether the object must be taken into account by the PCC (during path setup) or is just optional. When the P flag is set in PCUpd/PCInitiate message received on a PCEP session on which R bit was set by both peers, the object MUST be taken into account by the PCC. Conversely, when the P flag is cleared, the object is optional and the PCC is free to ignore it. The P flag for the mandatory objects SRP, LSP and ERO MUST be set in the PCUpd/PCInitiate message. If the mandatory object is received with the P flag set incorrectly according to the rules stated above, the receiving peer MUST send a PCErr message with Error-Type=10 (Reception of an invalid object) and Error-value=1 (reception of an object with P flag not set). By default, the PCE SHOULD set the P flag, unless a local configuration or local policy indicates that some constraints (corresponding PCEP objects) can be marked as optional and could be ignored by the PCC.

3.3. Handling of I flag

3.3.1. The PCUpd message

The I flag in the PCUpd message [RFC8231] allows a PCE to indicate to a PCC whether or not an optional object was processed. The PCE MAY include the ignored optional object in its update request and set the I flag to indicate that the optional object was ignored. When the I flag is cleared, the PCE indicates that the optional object was processed.

3.3.2. The PCRpt message

The I flag in the PCRpt message [RFC8231] allows a PCC to indicate to a PCE whether or not an optional object was processed in response to an LSP Update Request (PCUpd) or LSP Initiate Request (PCInitiate). The PCC MAY include the ignored optional object in its report and set the I flag to indicate that the optional object was ignored at PCC. When the I flag is cleared, the PCC indicates that the optional object was processed. The I flag has no meaning if the PCRpt message is not in response to a PCUpd or PCInitiate message (i.e. without the SRP object in the PCRpt message).

3.3.3. The PCInitiate message

The I flag has no meaning in the PCinitiate message [RFC8281] and is ignored.

3.4. Delegation

Delegation is an operation to grant a PCE temporary rights to modify a subset of LSP parameters on one or more LSPs of a PCC as described in [RFC8051]. Note that for the delegated LSPs, the PCE can update and mark some object as ignored even when the PCC had set the P flag during delegation. Similarly, the PCE can update and mark some object as a must to process even when the PCC had not set the P flag during delegation.

The PCC MUST acknowledge this by sending the PCRpt message with the P flag set as per the PCE expectation for the corresponding object. In case PCC cannot except this, it would react as per the processing rules of unacceptable update in [RFC8231].

3.5. Unknown Object Handling

This document updates the handling of unknown objects in stateful PCEP messages as per the setting of P flag in the common object header in a similar way as [RFC5440], i.e. if a PCEP speaker does not understand an object with the P flag set or understands the object but decides to ignore the object, the entire stateful PCEP message

MUST be rejected and the PCE MUST send a PCErr message with Error-Type="Unknown Object" or "Not supported Object" [RFC5440]. In case the P flag is not set, the PCEP speaker is free to ignore the object and continue with the message processing as defined.

[RFC8231] defined LSP Error Code TLV to be carried in PCRpt message in the LSP object to convey error information. This document does not change that procedure.

4. Security Considerations

This document clarifies how the already existing P and I flag in PCEP common object header could be used during stateful PCEP exchanges. It updates the unknown object error handling in stateful PCEP message exchange. These changes on its own do not add any new security concerns. The security considerations identified in [RFC5440], [RFC8231], and [RFC8281].

As per [RFC8231], it is RECOMMENDED that these PCEP extensions only be activated on authenticated and encrypted sessions across PCEs and PCCs belonging to the same administrative authority, using Transport Layer Security (TLS) [RFC8253] as per the recommendations and best current practices in [RFC7525] (unless explicitly set aside in [RFC8253]).

5. IANA Considerations

5.1. STATEFUL-PCE-CAPABILITY TLV

[RFC8231] defines the STATEFUL-PCE-CAPABILITY TLV; per that RFC, IANA created a "STATEFUL-PCE-CAPABILITY TLV Flag Field" subregistry to manage the value of the STATEFUL-PCE-CAPABILITY TLV's Flag field. IANA is requested to allocate a new bit in the subregistry, as follows:

Bit	Description	Reference

TBD1	RELAX bit	[This-I.D.]

6. Manageability Considerations

6.1. Control of Function and Policy

An operator MUST be allowed to configure the capability to support relaxation of constraints in the stateful PCEP message exchange. They SHOULD also allow configuration of related LSP constraints (or parameters) that are optional to process.

6.2. Information and Data Models

An implementation SHOULD allow the operator to view the capability defined in this document. To serve this purpose, the PCEP YANG module [I-D.ietf-pce-pcep-yang] could be extended in future.

6.3. Liveness Detection and Monitoring

Mechanisms defined in this document do not imply any new liveness detection and monitoring requirements in addition to those already listed in [RFC5440].

6.4. Verify Correct Operations

Mechanisms defined in this document do not imply any new operation verification requirements in addition to those already listed in [RFC5440].

6.5. Requirements On Other Protocols

Mechanisms defined in this document do not imply any new requirements on other protocols.

6.6. Impact On Network Operations

Mechanisms defined in this document do not have any impact on network operations in addition to those already listed in [RFC5440].

7. Acknowledgments

Thanks to Jonathan Hardwick for discussion and suggestions around this draft.

8. References

8.1. Normative References

- [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>>.
- [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>>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [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>>.

8.2. Informative References

- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.
- [RFC8051] Zhang, X., Ed. and I. Minei, Ed., "Applicability of a Stateful Path Computation Element (PCE)", RFC 8051, DOI 10.17487/RFC8051, January 2017, <<https://www.rfc-editor.org/info/rfc8051>>.
- [RFC8233] Dhody, D., Wu, Q., Manral, V., Ali, Z., and K. Kumaki, "Extensions to the Path Computation Element Communication Protocol (PCEP) to Compute Service-Aware Label Switched Paths (LSPs)", RFC 8233, DOI 10.17487/RFC8233, September 2017, <<https://www.rfc-editor.org/info/rfc8233>>.
- [RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <<https://www.rfc-editor.org/info/rfc8253>>.
- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.

[I-D.ietf-pce-pcep-yang]

Dhody, D., Hardwick, J., Beeram, V., and J. Tantsura, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", draft-ietf-pce-pcep-yang-12 (work in progress), July 2019.

[I-D.ietf-pce-association-group]

Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Dhody, D., and Y. Tanaka, "PCEP Extensions for Establishing Relationships Between Sets of LSPs", draft-ietf-pce-association-group-09 (work in progress), April 2019.

[I-D.ietf-pce-association-diversity]

Litkowski, S., Sivabalan, S., Barth, C., and M. Negi, "Path Computation Element Communication Protocol (PCEP) Extension for LSP Diversity Constraint Signaling", draft-ietf-pce-association-diversity-08 (work in progress), July 2019.

Appendix A. Contributors

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

Email: dhruv.ietf@gmail.com

Authors' Addresses

Cheng Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: chengli13@huawei.com

Haomian Zheng
Huawei Technologies
H1-1-A043S Huawei Industrial Base, Songshanhu
Dongguan, Guangdong 523808
China

Email: zhenghaomian@huawei.com

Stephane Litkowski
Orange

Email: stephane.litkowski@orange.com

PCE Working Group
Internet-Draft
Updates: 8231 (if approved)
Intended status: Standards Track
Expires: November 6, 2021

C. Li
H. Zheng
Huawei Technologies
S. Litkowski
Cisco
May 5, 2021

Extension for Stateful PCE to allow Optional Processing of PCEP Objects
draft-dhody-pce-stateful-pce-optional-08

Abstract

This document introduces a mechanism to mark some of the Path Computation Element (PCE) Communication Protocol (PCEP) objects as optional during PCEP messages exchange for the Stateful PCE model to allow relaxing some constraints. This document introduces this relaxation and updates RFC 8231.

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 <https://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 November 6, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. Overview	3
2.1. Usage Example	4
3. PCEP Extension	4
3.1. STATEFUL-PCE-CAPABILITY TLV	4
3.2. Handling of P flag	5
3.2.1. The PCRpt Message	5
3.2.2. The PCUpd Message and the PCInitiate Message	5
3.3. Handling of I flag	5
3.3.1. The PCUpd Message	5
3.3.2. The PCRpt Message	6
3.3.3. The PCInitiate Message	6
3.4. Delegation	6
3.5. Unknown Object Handling	6
4. Security Considerations	7
5. IANA Considerations	7
5.1. STATEFUL-PCE-CAPABILITY TLV	7
6. Manageability Considerations	7
6.1. Control of Function and Policy	7
6.2. Information and Data Models	7
6.3. Liveness Detection and Monitoring	8
6.4. Verify Correct Operations	8
6.5. Requirements On Other Protocols	8
6.6. Impact On Network Operations	8
7. Acknowledgments	8
8. References	8
8.1. Normative References	8
8.2. Informative References	9
Appendix A. Contributors	11
Authors' Addresses	11

1. Introduction

[RFC5440] describes the Path Computation Element Communication Protocol (PCEP) which enables the communication between a Path Computation Client (PCC) and a Path Control Element (PCE), or between two PCEs based on the PCE architecture [RFC4655].

PCEP Extensions for Stateful PCE Model [RFC8231] describes a set of extensions to PCEP to enable active control of Multiprotocol Label Switching Traffic Engineering (MPLS-TE) and Generalized MPLS (GMPLS) tunnels. [RFC8281] describes the setup and teardown of PCE-initiated

LSPs under the active stateful PCE model, without the need for local configuration on the PCC, thus allowing for dynamic control.

[RFC5440] defined the P flag (Processing-Rule) in the Common Object Header to allow a PCC to specify in a Path Computation Request (PCReq) message (sent to a PCE) whether the object must be taken into account by the PCE during path computation or is optional. The I flag (Ignore) is used by the PCE in a Path Computation Reply (PCRep) message to indicate to a PCC whether or not an optional object was considered by the PCE during path computation. Stateful PCE [RFC8231] specified that the P and I flags of the PCEP objects defined in [RFC8231] is to be set to zero on transmission and ignored on receipt, since they are exclusively related to path computation requests. The behavior for P and I flag in other messages defined in [RFC5440] and other extension was not specified. This document clarifies how the P and I flag could be used in the stateful PCE model to identify optional objects in the Path Computation State Report (PCRpt) [RFC8231], the Path Computation Update Request (PCUpd) [RFC8231], and the LSP Initiate Request (PCInitiate) [RFC8281] message.

This document updates [RFC8231] with respect to usage of the P and I flag as well as the handling of unknown objects in the stateful PCEP message exchange.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Overview

[RFC5440] describes the handling of unknown objects as per the setting of the P flag for the PCReq message. Further, [RFC8231] defined the usage of the LSP Error Code TLV in the PCRpt message in response to failed LSP Update Request via the PCUpd message (for example, due to an unsupported object/TLV).

This document clarifies the procedure of marking some objects as 'optional to be processed' by the PCEP peer in the stateful PCEP messages. Furthermore, this document updates the procedure for handling unknown objects in the stateful PCEP messages based on the P flag.

2.1. Usage Example

The PCRpt message is used to report the current state of an LSP. As part of the message both the <intended-attribute-list> and <actual-attribute-list> is encoded (see [RFC8231]). For example, the <intended-attribute-list> could include the METRIC object to indicate a limiting constraint (B flag set) for the Path Delay Variation metric [RFC8233]. In some scenarios, it would be useful to state that this limiting constraint can be relaxed by the PCE in case it cannot find a path. Similarly in the case of an association group [RFC8697] such as Disjoint Association [RFC8800], the PCE may need to completely relax the disjointness constraint in order to provide a path to all the LSPs that are part of the association. In these case it would be useful to mark the objects as 'optional' and it could be ignored by the PCEP peer. Also, it would be useful for the PCEP speaker to learn if the PCEP peer has relaxed the constraint and ignored the processing of the PCEP object.

Thus, this document simply clarifies, how the already existing P and I flag in the PCEP common object header could be used during the stateful PCEP message exchange.

3. PCEP Extension

3.1. STATEFUL-PCE-CAPABILITY TLV

A PCEP speaker indicates its ability to support the handling of the P and I flag in the stateful PCEP message exchange during the PCEP initialization phase, as follows. When the PCEP session is established, a PCC sends an Open message with an OPEN object that contains the STATEFUL-PCE-CAPABILITY TLV, as defined in [RFC8231]. A new flag, the R (RELAX) flag, is added in this TLV to indicate the support for relaxing the processing of some objects via the use of the P and I flag in the PCEP common object header.

R (RELAX bit - TBD1): If set to 1 by a PCEP Speaker, the R flag indicates that the PCEP Speaker is willing to send and receive PCEP objects with the P and I flags in the PCEP common object header for the stateful PCE messages. In case the bit is unset, it indicates that the PCEP Speaker would not handle the P and I flags in the PCEP common object header for stateful PCE messages.

The R flag MUST be set by both a PCC and a PCE to indicate support for the handling of the P and I flag in the PCEP common object header to allow relaxing some constraints by marking objects as optional to process. If the PCEP speaker did not set the R flag but receives PCEP objects with P or I bit set, it MUST behave as per the processing rule in [RFC8231] i.e., the bits are simply ignored.

3.2. Handling of P flag

3.2.1. The PCRpt Message

The P flag in the PCRpt message [RFC8231] allows a PCC to specify to a PCE whether the object must be taken into account by the PCE (during path computation, re-optimization, or state maintenance) or is optional to process. When the P flag is set in the PCRpt message received on a PCEP session on which R bit was set by both peers, the object MUST be taken into account by the PCE. Conversely, when the P flag is cleared, the object is optional and the PCE is free to ignore it. The P flag for the mandatory objects such as the LSP and the ERO object (intended path) MUST be set in the PCRpt message. If a mandatory object is received with the P flag set incorrectly according to the rules stated above, the receiving peer MUST send a PCErr message with Error-Type=10 (Reception of an invalid object) and Error-value=1 (reception of an object with P flag not set). On a PCEP session on which R bit was set by both peers, the PCC SHOULD set the P flag by default, unless a local configuration or local policy indicates that some constraints (corresponding PCEP objects) can be marked as optional and could be ignored by the PCE.

3.2.2. The PCUpd Message and the PCInitiate Message

The P flag in the PCUpd message [RFC8231] and the PCInitiate message [RFC8281] allows a PCE to specify to a PCC whether the object must be taken into account by the PCC (during path setup) or is optional to process. When the P flag is set in the PCUpd/PCInitiate message received on a PCEP session on which R bit was set by both peers, the object MUST be taken into account by the PCC. Conversely, when the P flag is cleared, the object is optional and the PCC is free to ignore it. The P flag for the mandatory objects such as the SRP, the LSP and the ERO MUST be set in the PCUpd/PCInitiate message. If a mandatory object is received with the P flag set incorrectly according to the rules stated above, the receiving peer MUST send a PCErr message with Error-Type=10 (Reception of an invalid object) and Error-value=1 (reception of an object with P flag not set). By default, the PCE SHOULD set the P flag, unless a local configuration or local policy indicates that some constraints (corresponding PCEP objects) can be marked as optional and could be ignored by the PCC.

3.3. Handling of I flag

3.3.1. The PCUpd Message

The I flag in the PCUpd message [RFC8231] allows a PCE to indicate to a PCC whether or not an optional object was processed. The PCE MAY include the ignored optional object in its update request and set the

I flag to indicate that the optional object was ignored. When the I flag is cleared, the PCE indicates that the optional object was processed.

3.3.2. The PCRpt Message

The I flag in the PCRpt message [RFC8231] allows a PCC to indicate to a PCE whether or not an optional object was processed in response to an LSP Update Request (PCUpd) or LSP Initiate Request (PCInitiate). The PCC MAY include the ignored optional object in its report and set the I flag to indicate that the optional object was ignored at PCC. When the I flag is cleared, the PCC indicates that the optional object was processed. The I flag has no meaning if the PCRpt message is not in response to a PCUpd or PCInitiate message (i.e. without the SRP object in the PCRpt message).

3.3.3. The PCInitiate Message

The I flag has no meaning in the PCinitiate message [RFC8281] and is ignored.

3.4. Delegation

Delegation is an operation to grant a PCE temporary rights to modify a subset of parameters on one or more LSPs by a PCC as described in [RFC8051]. Note that for the delegated LSPs, the PCE can update and mark some objects as ignored even when the PCC had set the P flag during delegation. Similarly, the PCE can update and mark some object as a must to process even when the PCC had not set the P flag during delegation.

The PCC MUST acknowledge this by sending the PCRpt message with the P flag set as per the PCE expectation for the corresponding object. In case PCC cannot accept this, it would react as per the processing rules of unacceptable update in [RFC8231].

3.5. Unknown Object Handling

This document updates the handling of unknown objects in the stateful PCEP messages as per the setting of the P flag in the common object header in a similar way as [RFC5440], i.e. if a PCEP speaker does not understand an object with the P flag set or understands the object but decides to ignore the object, the entire stateful PCEP message MUST be rejected and the PCE MUST send a PCErr message with Error-Type="Unknown Object" or "Not supported Object" [RFC5440]. In case the P flag is not set, the PCEP speaker is free to ignore the object and continue with message processing as defined.

[RFC8231] defined LSP Error Code TLV to be carried in PCrpt message in the LSP object to convey error information. This document does not change that procedure.

4. Security Considerations

This document clarifies how the already existing P and I flag in PCEP common object header could be used during stateful PCEP exchanges. It updates the unknown object error handling in stateful PCEP message exchange. These changes on their own do not add any new security concerns. The security considerations identified in [RFC5440], [RFC8231], and [RFC8281] continue to apply.

As per [RFC8231], it is RECOMMENDED that these PCEP extensions only be activated on authenticated and encrypted sessions across PCEs and PCCs belonging to the same administrative authority, using Transport Layer Security (TLS) [RFC8253] as per the recommendations and best current practices in [RFC7525] (unless explicitly set aside in [RFC8253]).

5. IANA Considerations

5.1. STATEFUL-PCE-CAPABILITY TLV

[RFC8231] defines the STATEFUL-PCE-CAPABILITY TLV; per that RFC, IANA created a "STATEFUL-PCE-CAPABILITY TLV Flag Field" subregistry to manage the value of the STATEFUL-PCE-CAPABILITY TLV's Flag field. IANA is requested to allocate a new bit in the subregistry, as follows:

Bit	Description	Reference

TBD1	RELAX bit	[This-I.D.]

6. Manageability Considerations

6.1. Control of Function and Policy

An operator MUST be allowed to configure the capability to support relaxation of constraints in the stateful PCEP message exchange. They SHOULD also allow configuration of related LSP constraints (or parameters) that are optional to process.

6.2. Information and Data Models

An implementation SHOULD allow the operator to view the capability defined in this document. To serve this purpose, the PCEP YANG module [I-D.ietf-pce-pcep-yang] could be extended in the future.

6.3. Liveness Detection and Monitoring

Mechanisms defined in this document do not imply any new liveness detection and monitoring requirements in addition to those already listed in [RFC5440].

6.4. Verify Correct Operations

Mechanisms defined in this document do not imply any new operation verification requirements in addition to those already listed in [RFC5440].

6.5. Requirements On Other Protocols

Mechanisms defined in this document do not imply any new requirements on other protocols.

6.6. Impact On Network Operations

Mechanisms defined in this document do not have any impact on network operations in addition to those already listed in [RFC5440].

7. Acknowledgments

Thanks to Jonathan Hardwick for discussion and suggestions around this draft.

8. References

8.1. Normative References

- [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>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [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>>.

8.2. Informative References

- [I-D.ietf-pce-pcep-yang] Dhody, D., Hardwick, J., Beeram, V. P., and J. Tantsura, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", draft-ietf-pce-pcep-yang-16 (work in progress), February 2021.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.
- [RFC8051] Zhang, X., Ed. and I. Minei, Ed., "Applicability of a Stateful Path Computation Element (PCE)", RFC 8051, DOI 10.17487/RFC8051, January 2017, <<https://www.rfc-editor.org/info/rfc8051>>.
- [RFC8233] Dhody, D., Wu, Q., Manral, V., Ali, Z., and K. Kumaki, "Extensions to the Path Computation Element Communication Protocol (PCEP) to Compute Service-Aware Label Switched Paths (LSPs)", RFC 8233, DOI 10.17487/RFC8233, September 2017, <<https://www.rfc-editor.org/info/rfc8233>>.
- [RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <<https://www.rfc-editor.org/info/rfc8253>>.
- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.

- [RFC8697] Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Dhody, D., and Y. Tanaka, "Path Computation Element Communication Protocol (PCEP) Extensions for Establishing Relationships between Sets of Label Switched Paths (LSPs)", RFC 8697, DOI 10.17487/RFC8697, January 2020, <<https://www.rfc-editor.org/info/rfc8697>>.
- [RFC8800] Litkowski, S., Sivabalan, S., Barth, C., and M. Negi, "Path Computation Element Communication Protocol (PCEP) Extension for Label Switched Path (LSP) Diversity Constraint Signaling", RFC 8800, DOI 10.17487/RFC8800, July 2020, <<https://www.rfc-editor.org/info/rfc8800>>.

Appendix A. Contributors

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

Email: dhruv.ietf@gmail.com

Authors' Addresses

Cheng Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: c.l@huawei.com

Haomian Zheng
Huawei Technologies
H1, Huawei Xiliu Beipo Village, Songshan Lake
Dongguan, Guangdong 523808
China

Email: zhenghaomian@huawei.com

Stephane Litkowski
Cisco

Email: slitkows.ietf@gmail.com

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

C. Li
H. Zheng
Huawei Technologies
S. Sivabalan
Cisco Systems, Inc.
July 8, 2019

Conveying Vendor-Specific Information in the Path Computation Element
(PCE) Communication Protocol (PCEP) extensions for stateful PCE.
draft-dhody-pce-stateful-pce-vendor-07

Abstract

A Stateful Path Computation Element (PCE) maintains information on the current network state, including: computed Label Switched Path (LSPs), reserved resources within the network, and pending path computation requests. This information may then be considered when computing new traffic engineered LSPs, and for associated and dependent LSPs, received from Path Computation Clients (PCCs).

RFC 7470 defines a facility to carry vendor-specific information in PCEP.

This document extends this capability for the stateful PCE messages.

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 <https://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 January 9, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. Procedures for the Vendor Information Object	3
3. Procedures for the Vendor Information TLV	5
4. Vendor Information Object and TLV	6
5. Manageability Considerations	6
5.1. Control of Function and Policy	6
5.2. Information and Data Models	6
5.3. Liveness Detection and Monitoring	6
5.4. Verify Correct Operations	6
5.5. Requirements On Other Protocols	6
5.6. Impact On Network Operations	6
6. IANA Considerations	7
7. Security Considerations	7
8. Acknowledgments	7
9. References	7
9.1. Normative References	7
9.2. Informative References	8
Appendix A. Contributor Addresses	9
Authors' Addresses	9

1. Introduction

The Path Computation Element communication Protocol (PCEP) [RFC5440] provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients' (PCCs) requests.

A stateful PCE is capable of considering, for the purposes of path computation, not only the network state in terms of links and nodes (referred to as the Traffic Engineering Database or TED) but also the status of active services (previously computed paths, and currently reserved resources, stored in the Label Switched Paths Database (LSP-DB). [RFC8051] describes general considerations for a stateful PCE deployment and examines its applicability and benefits, as well as its challenges and limitations through a number of use cases.

[RFC8231] describes a set of extensions to PCEP to provide stateful control. A stateful PCE has access to not only the information carried by the network's Interior Gateway Protocol (IGP), but also the set of active paths and their reserved resources for its computations. The additional state allows the PCE to compute constrained paths while considering individual LSPs and their interactions. [RFC8281] describes the set-up, maintenance and teardown of PCE-initiated LSPs under the stateful PCE model. These extensions added new messages in PCEP for stateful PCE.

[RFC7470] defined Vendor Information object that can be used to carry arbitrary, proprietary information such as vendor-specific constraints. It also defined VENDOR-INFORMATION-TLV that can be used to carry arbitrary information within any existing or future PCEP object that supports TLVs.

This document extend the usage of Vendor Information Object and VENDOR-INFORMATION-TLV to stateful PCE. The VENDOR-INFORMATION-TLV can be carried inside any of the new objects added in PCEP for stateful PCE as per [RFC7470], this document extend the PCEP messages to also include the Vendor Information Object as well.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Procedures for the Vendor Information Object

A Path Computation LSP State Report message [RFC8231] (also referred to as PCRpt message) is a PCEP message sent by a PCC to a PCE to report the current state of an LSP. A PCC that wants to convey proprietary or vendor-specific information or metrics to a PCE does so by including a Vendor Information object in the PCRpt message. The contents and format of the object are described in Section 4 of [RFC7470]. The PCE determines how to interpret the information in the Vendor Information object by examining the Enterprise Number it contains.

The Vendor Information object is OPTIONAL in a PCRpt message. Multiple instances of the object MAY be used on a single PCRpt message. Different instances of the object can have different Enterprise Numbers.

The format of the PCRpt message (with [RFC8231] as base) is updated as follows:

```
<PCRpt Message> ::= <Common Header>
                        <state-report-list>
```

Where:

```
<state-report-list> ::= <state-report> [<state-report-list>]
```

```
<state-report> ::= [<SRP>]
                    <LSP>
                    <path>
                    [<vendor-info-list>]
```

Where:

```
<vendor-info-list> ::= <VENDOR-INFORMATION>
                        [<vendor-info-list>]
```

<path> is defined in [RFC8231].

A Path Computation LSP Update Request message (also referred to as PCUpd message) is a PCEP message sent by a PCE to a PCC to update attributes of an LSP. The Vendor Information object can be included in a PCUpd message to convey proprietary or vendor-specific information.

The format of the PCUpd message (with [RFC8231] as base) is updated as follows:

```
<PCUpd Message> ::= <Common Header>
                        <update-request-list>
```

Where:

```
<update-request-list> ::= <update-request>
                           [<update-request-list>]
```

```
<update-request> ::= <SRP>
                     <LSP>
                     <path>
                     [<vendor-info-list>]
```

Where:

```
<vendor-info-list> ::= <VENDOR-INFORMATION>
                        [<vendor-info-list>]
```

<path> is defined in [RFC8231].

A Path Computation LSP Initiate Message (also referred to as PCInitiate message) is a PCEP message sent by a PCE to a PCC to trigger LSP instantiation or deletion. The Vendor Information object

can be included in a PCInitiate message to convey proprietary or vendor-specific information.

The format of the PCInitiate message (with [RFC8281] as base) is updated as follows:

```
<PCInitiate Message> ::= <Common Header>
                           <PCE-initiated-lsp-list>
```

Where:

```
<PCE-initiated-lsp-list> ::= <PCE-initiated-lsp-request>
                              [<PCE-initiated-lsp-list>]
```

```
<PCE-initiated-lsp-request> ::=
    (<PCE-initiated-lsp-instantiation> |
     <PCE-initiated-lsp-deletion>)
```

```
<PCE-initiated-lsp-instantiation> ::= <SRP>
                                         <LSP>
                                         [<END-POINTS>]
                                         <ERO>
                                         [<attribute-list>]
                                         [<vendor-info-list>]
```

Where:

```
<vendor-info-list> ::= <VENDOR-INFORMATION>
                        [<vendor-info-list>]
```

<PCE-initiated-lsp-deletion> and <attribute-list> is as per [RFC8281].

A legacy implementation that does not recognize the Vendor Information object will act according to the procedures set out in [RFC8231] and [RFC8281]. An implementation that supports the Vendor Information object, but receives one carrying an Enterprise Number that it does not support, SHOULD ignore the object in the same way as described in [RFC7470].

3. Procedures for the Vendor Information TLV

The Vendor Information TLV can be used to carry vendor-specific information that applies to a specific PCEP object by including the TLV in the object. This includes objects used in stateful PCE extension such as SRP and LSP object. All the procedures as per section 3 of [RFC7470].

4. Vendor Information Object and TLV

[RFC7470] specify the format of VENDOR-INFORMATION Object and VENDOR-INFORMATION-TLV.

5. Manageability Considerations

All manageability requirements and considerations listed in [RFC5440], [RFC7470] and [RFC8231] apply to PCEP protocol extensions defined in this document. In addition, requirements and considerations listed in this section apply.

5.1. Control of Function and Policy

As stated in [RFC7470], this capability, the associated vendor specific information and policy SHOULD made configurable. This information can be used in stateful messages as well.

5.2. Information and Data Models

The PCEP YANG module is specified in [I-D.ietf-pce-pcep-yang]. It is NOT RECOMMENDED that standard YANG module be augmented with details of vendor information. It MAY be extended to include the use of this information and the Enterprise Numbers that the object and TLVs contain.

5.3. Liveness Detection and Monitoring

Mechanisms defined in this document do not imply any new liveness detection and monitoring requirements in addition to those already listed in [RFC5440].

5.4. Verify Correct Operations

Mechanisms defined in this document do not imply any new operation verification requirements in addition to those already listed in [RFC5440] and [RFC8231].

5.5. Requirements On Other Protocols

Mechanisms defined in this document do not imply any new requirements on other protocols.

5.6. Impact On Network Operations

Mechanisms defined in [RFC5440] and [RFC8231] also apply to PCEP extensions defined in this document. Further, the mechanism

described in this document can help the operator to request control of the LSPs at a particular PCE.

6. IANA Considerations

There are no IANA consideration in this document.

7. Security Considerations

The protocol extensions defined in this document do not change the nature of PCEP. Therefore, the security considerations set out in [RFC5440], [RFC7470], [RFC8231] and [RFC8281] apply unchanged.

As stated in [RFC6952], PCEP implementations SHOULD support the TCP-AO [RFC5925] and not use TCP MD5 because of TCP MD5's known vulnerabilities and weakness. PCEP also support Transport Layer Security (TLS) [RFC8253] as per the recommendations and best current practices in [RFC7525].

8. Acknowledgments

Thanks to Avantika, Mahendra Singh Negi, Udayasree Palle and Swapna K for their suggestions.

9. References

9.1. Normative References

- [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>>.
- [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>>.
- [RFC7470] Zhang, F. and A. Farrel, "Conveying Vendor-Specific Constraints in the Path Computation Element Communication Protocol", RFC 7470, DOI 10.17487/RFC7470, March 2015, <<https://www.rfc-editor.org/info/rfc7470>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [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>>.
- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.

9.2. Informative References

- [I-D.ietf-pce-pcep-yang] Dhody, D., Hardwick, J., Beeram, V., and J. Tantsura, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", draft-ietf-pce-pcep-yang-12 (work in progress), July 2019.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <<https://www.rfc-editor.org/info/rfc5925>>.
- [RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, DOI 10.17487/RFC6952, May 2013, <<https://www.rfc-editor.org/info/rfc6952>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.
- [RFC8051] Zhang, X., Ed. and I. Minei, Ed., "Applicability of a Stateful Path Computation Element (PCE)", RFC 8051, DOI 10.17487/RFC8051, January 2017, <<https://www.rfc-editor.org/info/rfc8051>>.
- [RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <<https://www.rfc-editor.org/info/rfc8253>>.

Appendix A. Contributor Addresses

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

Authors' Addresses

Cheng Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: chengli13@huawei.com

Haomian Zheng
Huawei Technologies
F3 RnD Center, Huawei Industrial Base, Bantian, Longgang District
Shenzhen, Guangdong 518129
P.R.China

Email: zhenghaomian@huawei.com

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

Email: msiva@cisco.com

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: 21 October 2022

C. Li
H. Zheng
Huawei Technologies
S. Sivabalan
Ciena
S. Sidor
Z. Ali
Cisco Systems, Inc.
19 April 2022

Conveying Vendor-Specific Information in the Path Computation Element
(PCE) Communication Protocol (PCEP) extensions for Stateful PCE.
draft-dhody-pce-stateful-pce-vendor-14

Abstract

A Stateful Path Computation Element (PCE) maintains information on the current network state, including: computed Label Switched Path (LSPs), reserved resources within the network, and the pending path computation requests. This information may then be considered when computing new traffic engineered LSPs, and for the associated and the dependent LSPs, received from a Path Computation Client (PCC).

RFC 7470 defines a facility to carry vendor-specific information in Path Computation Element Communication Protocol (PCEP).

This document extends this capability for the Stateful PCEP messages.

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 <https://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 21 October 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. Procedures for the Vendor Information Object	3
3. Procedures for the Vendor Information TLV	6
4. Vendor Information Object and TLV	6
5. Manageability Considerations	7
5.1. Control of Function and Policy	7
5.2. Information and Data Models	7
5.3. Liveness Detection and Monitoring	7
5.4. Verify Correct Operations	7
5.5. Requirements On Other Protocols	7
5.6. Impact On Network Operations	7
6. IANA Considerations	8
7. Implementation Status	8
7.1. Cisco Systems	8
8. Security Considerations	9
9. Acknowledgments	9
10. References	9
10.1. Normative References	9
10.2. Informative References	10
Appendix A. Contributor Addresses	11
Authors' Addresses	11

1. Introduction

The Path Computation Element Communication Protocol (PCEP) [RFC5440] provides mechanisms for a Path Computation Element (PCE) to perform path computation in response to a Path Computation Client (PCC) request.

A Stateful PCE is capable of considering, for the purposes of the path computation, not only the network state in terms of links and nodes (referred to as the Traffic Engineering Database or TED) but also the status of active services (previously computed paths, and currently reserved resources, stored in the Label Switched Paths Database (LSP-DB)). [RFC8051] describes general considerations for a Stateful PCE deployment and examines its applicability and benefits, as well as its challenges and limitations through a number of use cases.

[RFC8231] describes a set of extensions to PCEP to provide stateful control. A Stateful PCE has access to not only the information carried by the network's Interior Gateway Protocol (IGP), but also the set of active paths and their reserved resources for its computations. The additional state allows the PCE to compute constrained paths while considering individual LSPs and their interactions. [RFC8281] describes the set up, maintenance and teardown of PCE-initiated LSPs under the Stateful PCE model. These extensions added new messages in PCEP for Stateful PCE.

[RFC7470] defined Vendor Information object that can be used to carry arbitrary, proprietary information such as vendor-specific constraints. It also defined VENDOR-INFORMATION-TLV that can be used to carry arbitrary information within any existing or future PCEP object that supports TLVs.

This document extend the usage of Vendor Information Object and VENDOR-INFORMATION-TLV to Stateful PCE. The VENDOR-INFORMATION-TLV can be carried inside any of the new objects added in PCEP for Stateful PCE as per [RFC7470], this document extend the stateful PCEP messages to also include the Vendor Information Object as well.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Procedures for the Vendor Information Object

A Path Computation LSP State Report message (also referred to as PCRpt message) [RFC8231] is a PCEP message sent by a PCC to a PCE to report the current state of an LSP. A PCC that wants to convey proprietary or vendor-specific information or metrics to a PCE does so by including a Vendor Information object in the PCRpt message. The contents and format of the object are described in Section 4 of

[RFC7470]. The PCE determines how to interpret the information in the Vendor Information object by examining the Enterprise Number it contains.

The Vendor Information object is OPTIONAL in a PCRpt message. Multiple instances of the object MAY be used on a single PCRpt message. Different instances of the object can have different Enterprise Numbers.

The format of the PCRpt message (with [RFC8231] as base) is updated as follows:

```
<PCRpt Message> ::= <Common Header>
                    <state-report-list>
```

Where:

```
<state-report-list> ::= <state-report>[<state-report-list>]
```

```
<state-report> ::= [<SRP>]
                  <LSP>
                  <path>
                  [<vendor-info-list>]
```

Where:

```
<vendor-info-list> ::= <VENDOR-INFORMATION>
                      [<vendor-info-list>]
```

<path> is defined in [RFC8231].

A Path Computation LSP Update Request message (also referred to as PCUpd message) [RFC8231] is a PCEP message sent by a PCE to a PCC to update attributes of an LSP. The Vendor Information object can be included in a PCUpd message to convey proprietary or vendor-specific information.

The format of the PCUpd message (with [RFC8231] as base) is updated as follows:

```
<PCUpd Message> ::= <Common Header>
                        <update-request-list>
```

Where:

```
<update-request-list> ::= <update-request>
                        [<update-request-list>]
```

```
<update-request> ::= <SRP>
                        <LSP>
                        <path>
                        [<vendor-info-list>]
```

Where:

```
<vendor-info-list> ::= <VENDOR-INFORMATION>
                        [<vendor-info-list>]
```

<path> is defined in [RFC8231].

A Path Computation LSP Initiate Message (also referred to as PCInitiate message) [RFC8281] is a PCEP message sent by a PCE to a PCC to trigger an LSP instantiation or deletion. The Vendor Information object can be included in a PCInitiate message to convey proprietary or vendor-specific information.

The format of the PCInitiate message (with [RFC8281] as base) is updated as follows:

```
<PCInitiate Message> ::= <Common Header>
                           <PCE-initiated-lsp-list>
```

Where:

```
<PCE-initiated-lsp-list> ::= <PCE-initiated-lsp-request>
                              [<PCE-initiated-lsp-list>]
```

```
<PCE-initiated-lsp-request> ::=
    (<PCE-initiated-lsp-instantiation> |
     <PCE-initiated-lsp-deletion>)
```

```
<PCE-initiated-lsp-instantiation> ::= <SRP>
                                         <LSP>
                                         [<END-POINTS>]
                                         <ERO>
                                         [<attribute-list>]
                                         [<vendor-info-list>]
```

Where:

```
<vendor-info-list> ::= <VENDOR-INFORMATION>
                        [<vendor-info-list>]
```

<PCE-initiated-lsp-deletion> and <attribute-list> is as per [RFC8281].

A legacy implementation that does not recognize the Vendor Information object will act according to the procedures set out in [RFC8231] and [RFC8281]. An implementation that supports the Vendor Information object, but receives one carrying an Enterprise Number that it does not support, MUST ignore the object in the same way as described in [RFC7470].

3. Procedures for the Vendor Information TLV

The Vendor Information TLV can be used to carry vendor-specific information that applies to a specific PCEP object by including the TLV in the object. This includes objects used in Stateful PCE extension such as SRP and LSP object. All the procedures as per section 3 of [RFC7470].

4. Vendor Information Object and TLV

[RFC7470] specify the format of VENDOR-INFORMATION Object and VENDOR-INFORMATION-TLV.

5. Manageability Considerations

All manageability requirements and considerations listed in [RFC5440], [RFC7470], [RFC8231], and [RFC8281] apply to PCEP protocol extensions defined in this document. In addition, requirements and considerations listed in this section apply.

5.1. Control of Function and Policy

As stated in [RFC7470], this capability, the associated vendor specific information and policy SHOULD made configurable. This information can be used in Stateful PCEP messages as well.

5.2. Information and Data Models

The PCEP YANG module is specified in [I-D.ietf-pce-pcep-yang]. It is RECOMMENDED that standard YANG module not be augmented with details of vendor information. It MAY be extended to include the use of this information and the Enterprise Numbers that the object and TLVs contain.

5.3. Liveness Detection and Monitoring

Mechanisms defined in this document do not imply any new liveness detection and monitoring requirements in addition to those already listed in [RFC5440].

5.4. Verify Correct Operations

Mechanisms defined in this document do not imply any new operation verification requirements in addition to those already listed in [RFC5440] and [RFC8231].

5.5. Requirements On Other Protocols

Mechanisms defined in this document do not imply any new requirements on other protocols.

5.6. Impact On Network Operations

Mechanisms defined in [RFC5440] and [RFC8231] also apply to PCEP extensions defined in this document. Further, the mechanism described in this document can help the operator to request control of the LSPs at a particular PCE.

6. IANA Considerations

There are no IANA consideration in this document.

7. Implementation Status

[NOTE TO RFC EDITOR : This whole section and the reference to RFC 7942 is to be removed before publication as an RFC]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

7.1. Cisco Systems

- * Organization: Cisco Systems, Inc.
- * Implementation: Cisco IOS-XR PCE and PCC
- * Description: Vendor Information Object used in PCRpt, PCUpd and PCInitiate messages.
- * Maturity Level: Production
- * Coverage: Full
- * Contact: ssidior@cisco.com

8. Security Considerations

The protocol extensions defined in this document do not change the nature of PCEP. Therefore, the security considerations set out in [RFC5440], [RFC7470], [RFC8231] and [RFC8281] apply unchanged.

As stated in [RFC6952], PCEP implementations SHOULD support the TCP-AO [RFC5925] and not use TCP MD5 because of TCP MD5's known vulnerabilities and weakness. PCEP also support Transport Layer Security (TLS) [RFC8253] as per the recommendations and best current practices in [RFC7525].

9. Acknowledgments

Thanks to Avantika, Mahendra Singh Negi, Udayasree Palle and Swapna K for their suggestions.

10. References

10.1. Normative References

- [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>>.
- [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>>.
- [RFC7470] Zhang, F. and A. Farrel, "Conveying Vendor-Specific Constraints in the Path Computation Element Communication Protocol", RFC 7470, DOI 10.17487/RFC7470, March 2015, <<https://www.rfc-editor.org/info/rfc7470>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [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>>.

- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.

10.2. Informative References

- [I-D.ietf-pce-pcep-yang] Dhody, D., Hardwick, J., Beeram, V. P., and J. Tantsura, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", Work in Progress, Internet-Draft, draft-ietf-pce-pcep-yang-18, 25 January 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-pce-pcep-yang-18>>.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <<https://www.rfc-editor.org/info/rfc5925>>.
- [RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, DOI 10.17487/RFC6952, May 2013, <<https://www.rfc-editor.org/info/rfc6952>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [RFC8051] Zhang, X., Ed. and I. Minei, Ed., "Applicability of a Stateful Path Computation Element (PCE)", RFC 8051, DOI 10.17487/RFC8051, January 2017, <<https://www.rfc-editor.org/info/rfc8051>>.
- [RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <<https://www.rfc-editor.org/info/rfc8253>>.

Appendix A. Contributor Addresses

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: dhruv.ietf@gmail.com

Mike Koldychev
Cisco Systems
Kanata, Ontario
Canada

EMail: mkoldych@cisco.com

Authors' Addresses

Cheng Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing
100095
China
Email: c.l@huawei.com

Haomian Zheng
Huawei Technologies
H1, Huawei Xiliu Beipo Village, Songshan Lake
Dongguan
Guangdong, 523808
China
Email: zhenghaomian@huawei.com

Siva Sivabalan
Ciena
385 Terry Fox Drive
Kanata Ontario K2K 0L1
Canada
Email: msiva282@gmail.com

Samuel Sidor
Cisco Systems, Inc.
Email: ssidor@cisco.com

Zafar Ali
Cisco Systems, Inc.
Email: zali@cisco.com

Network Working group
Internet Draft
Intended status: Standard Track

H. Bidgoli, Ed.
Nokia
D. Voyer, Ed.
Bell Canada
S. Rajarathinam
J. Kotalwar
Nokia
S. Sivabalan
Cisco System, Inc.

Expires: January 7, 2020

July 6, 2019

PCEP extensions for p2mp sr policy
draft-hsd-pce-sr-p2mp-policy-00

Abstract

SR P2MP policies are set of policies that enable architecture for P2MP service delivery.

This document specifies extensions to the Path Computation Element Communication Protocol (PCEP) that allow a stateful PCE to compute and initiate P2MP paths from a Root to a set of Leaves.

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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." The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on October 8, 2017.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Conventions used in this document	4
3. Overview of PCEP Operation in SR P2MP Network	4
3.1. High level view of a P2MP Policy Objects	5
3.1.1. Existing drafts used in defining the P2MP Policy	6
3.1.2. P2MP Identification	7
3.2 High-Level Procedures for P2MP SR LSP Instantiation	7
3.2.1 MVPN procedures	8
3.2.2. Global Optimization for P2MP LSPs	10
3.2.3. Fast Reroute	10
3.2.3. Connecting Replication Policy via Segment List	11
3.3. SR P2MP New TLVs and Objects	12
4. Object Format	12
4.1. Open Message and Capability Exchange	12
4.2. Symbolic Name in PCInitiate message from PCC	13
4.3. Replication policy	14
4.3.1 P2MP Policy Association Group	14
4.3.1.1 P2MP SR Policy Association Group Policy Identifiers TLV	14
4.3.1.2 P2MP SR Policy Association Group Candidate Path Identifiers TLV	15
4.3.1.3 P2MP SR Policy Association Group Candidate Path Attributes TLV	16
4.4. PCEP Message Exchanges	17
4.4.1 Extension of the LSP Object, SR-P2MP-LSPID-TLV	17
4.5. SR-P2MP-CCI Object	18
4.5.1 Optional IP-Address TLVs	19
4.6. Root PCE Report message	21

4.6.1 END-POINTS Objects	21
5. Examples of PCEP messages between PCE and PCEP	23
5.1. PCE Initiate and PCC Leaves Update	23
5.2. PCE P2MP LSP Calculation and Replication Policy download	27
5.3. PCC Rpt for PCE Update and Init Messages	35
6. Tree Deletion	37
7. Fragmentation	37
8. Example workflow	37
6. IANA Considerations	37
7. Security Considerations	37
8. References	37
8.1. Normative References	37
8.2. Informative References	38
7. Acknowledgments	38
Authors' Addresses	38

1. Introduction

The draft [draft-voyer-spring-sr-p2mp-policy] defines a variant of the SR Policy [I-D. ietf-spring-segment-routing-policy] for constructing a P2MP segment to support multicast service delivery.

A Point-to-Multipoint (P2MP) segment connects a Root node to a set of Leaf nodes in a Segment Routing Domain. We also define a Replication segment, which corresponds to the state of a P2MP segment on a particular node.

A P2MP segment consists of replication segments for the root, leaves and optionally intermediate replication nodes.

A replication segment defines the forwarding behavior on a particular node on a particular P2MP segment.

For a P2MP segment, a controller may be used to compute paths from a Root node to a set of Leaf nodes, optionally via a set of replication nodes. A packet is replicated at the root node and optionally on Replication nodes towards each Leaf node.

We define two types of a P2MP segment: Spray and Replication.

A Point-to-Multipoint service delivery could be via Ingress Replication (aka Spray in some SR context), i.e., the root unicasts individual copies of traffic to each leaf. The corresponding P2MP segment consists of replication segments only for the root and the leaves.

A Point-to-Multipoint service delivery could also be via Downstream Replication (aka TreeSID in some SR context), i.e., the root and some downstream replication nodes replicate the traffic along the way as it traverses closer to the leaves.

Notice that Spray is actually a special form of TreeSID. Also notice that, the explicit path from the root or a replication node to a leaf or a downstream replication node can optionally be partially or completely specified by the controller PCE or determined locally via static configuration.

A PCE could compute a tree from a Root node to a set of Leaf nodes via a set of Replication nodes. A packet is replicated at the Root node and on Replication nodes towards each Leaf node. It should be noted that two replication nodes can be connected directly, or they can be connected via unicast SR segment or a segment list.

The leaves and the root can be explicitly configured on the PCE or PCC can update the PCE with the information of the discovered root and leaves. As an example Multicast protocols like mvpn procedures or pim can be used to discover the leaves and roots on the PCC and update the PCE with these relevant information. The controller can calculate the P2MP Segment based on these info.

In all of above cases a set of new PCEP object and TLVs are needed to update and instantiate the P2MP tree. This draft explains the procedure needed to instantiate a P2MP TreeSID.

2. Conventions used in this document

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].

3. Overview of PCEP Operation in SR P2MP Network

For a P2MP SR policy, a PCE calculates a P2MP tree and programs the Root, Replication and Leaf nodes with information needed to forward a multicast stream from the root to a set of leaves. The PCE discovers the Root and the set of the Leaves via manual configuration on the PCE. On other hand the PCC (Root of the P2MP Tree) can provide the PCE with the relevant information by discovering the leaves via mvpn procedures or pim.

After discovering the Root and Leaves and computing the MPLS P2MP Tree and identifying the Replication routers, the PCE programs the PCCs with relevant information needed to create a P2MP Tree.

As per draft draft-voyer-spring-sr-p2mp-policy a P2MP Policy is defined by a root and set of leaves. A P2MP Policy contains replication policies. A replication policy is set of forwarding instructions on a specific node. As an example the push information on the Root node or swap and outgoing interface information on the transit nodes or pop information on the bud and leaves nodes. In addition since a P2MP policy is a variant of SR policy it uses the same concept as draft draft-darsh-pce-segment-routing-policy-cp. In short a replication policy uses a collection of SR P2MP Candidate paths. The candidate paths are computed by the PCE and can be used for P2MP LSP redundancy. In short each candidate path in the replication policy is a unique P2MP lsp.

PCE could also calculate and download additional information such as next-hops for link/node protection or initiate a make-before-break procedure for global Path optimization.

3.1. High level view of a P2MP Policy Objects

-SR P2MP Policy:

-Is a policy on PCE which contains information about:

- root node of the P2MP Segment
- leaf nodes of the P2MP Segment
- optionally constrains used to build the tree from leaf nodes to the root node.
- Tree-ID, which is a unique identifier of the P2MP tree on the root

-Replication Policy:

- Is the forwarding information needed on each node and is contained by P2MP policy. It is identified via:

- Its node-ID, the node that replication policy belongs to.
- The root of the P2MP Tree
- Tree-ID, which is a unique identifier of the P2MP Tree on the root. As an example the could be MP-BGP opaque value as per [RFC6513]
- It also contains a set of Candidate paths for P2MP tree

redundancy

-Candidate Path:

- Is used for P2MP Tree redundancy where the P2MP LSP with the highest preference is the active LSP
- It option can contain up to two P2MP LSP global optimization procedures, each identified with their own LSP-ID (i.e. make before break)
- It also contains the forwarding information for the P2MP LSPs, these forwarding information can be a replication SID or a segment list.

3.1.1. Existing drafts used in defining the P2MP Policy

P2MP Policy reuses current drafts and PCEP objects to identify the root and the leaves on the P2MP Segment and update the PCE with these information, and also to have PCE initiate and update P2MP Replication Policies on a the PCC.

In addition this draft will introduced new TLVs and Objects specific to a P2MP Policy.

This draft reuses the following pcep drafts:

- [RFC8231] The bases for a stateful PCE, and reuses the following objects or a variant of them

<SRP Object>

<Lsp Object>

- [RFC8236] P2MP capabilities advertisement

Also a variation of the P2MP LSP Identifier specified in above RFC

- [draft-barth-pce-segment-routing-policy-cp-02] Candidate paths for P2MP Policy is used for Tree Redundancy. As an example a P2MP Policy can have multiple candidate paths each protecting the primary candidate path. The active path is chosen via the precedence of the candidate path.

- [RFC 3209] defines the LSP-ID, LSP-ID is used for global optimization of a candidate path with in a P2MP policy. Each Candidate path can have 2 sub-lsps (LSP-IDs) for MBB and global optimization procedures.

- [draft-ietf-spring-segment-routing-policy] segment-list, used for connecting two non-adjacent replication policy via a unicast binding SID or Segment-list.
- [draft-ietf-pce-pcep-extension-for-pce-controller] a variant of CCI Object, used for downloading the replication policy forwarding instructions. These instructions can be incoming label and set of outgoing labels or fast reroute procedures or even downloading of a segment-list connecting two non-adjacent replication policy.
- [RFC8306] P2MP End Point objects, used for the PCC to update the PCE with discovered Leaves.

It should be noted that the [draft-dhs-spring-sr-p2mp-policy-yang] can provide farther details of the high level P2MP Policy Model.

3.1.2. P2MP Identification

The key to identify a P2MP LSP is in LSP object and is as follow:

PLSP-ID: RFC 8231, is assigned by PCC and is unique per candidate path. It is constant for the lifetime of a PCEP session. Since PLSP-ID is unique per LSP, Stand-by P2MP LSPs will be downloaded with a new PLSP-ID. It should be noted a stand-by LSP is a LSP to protect the primary LSP and can be setup in parallel to the primary LSP. These stand-by LSPs are identified by the candidate path.

LSP-ID: LSP ID Identifier as defined in rfc 3209, and is used for global optimization of a P2MP LSP (Candidate path)

Tree-ID: is equivalent to Tunnel Identifier color which identifies a unique P2MP segment at a ROOT and is advertised via the PTA in the BGP AD route.

Root: is equivalent to the first MPLS node on the path, as per [RFC3209], Section 4.6.2.1

Note that the Tree-ID, Root and LSP-ID are part of a new SR-P2MP-LSP_Identifier TLV which will be identified in this draft.

3.2 High-Level Procedures for P2MP SR LSP Instantiation

A P2MP policy is consistent of Root and a set of Leaves and as such a set of replication policies for each node within the path of the P2MP segment. As mentioned previously a replication policy is a set of forwarding rules used on root, transit nodes and leaves.

The Root and Leaves can be discovered via many methods.

- They can be configured and identified on a controller
- They can be configured on the root node PCC and the root updates the PCE with this information
- They can be discovered via Multicast mechanisms like MVPN procedures or protocols like PIM.

3.2.1 MVPN procedures

In case of MVPN there can be pcc-initiated or pce-initiated p2mp policy. In either case MVPN procedures [RFC6513, RFC6514] are used to discover the leaves on PCC and report them up to the PCE.

1. PCE-initiated Procedure :

PCE is informed of the P2MP request through it's API or configuration mechanism to instantiate a P2MP tunnel. PCE will initiate the P2MP LSP for the request, by sending a PC Initiate message to the Root. The above PC Initiate message to the Root will contain the following information. PLSP-ID = 0, LSP-ID (SR-P2MP-LSP-ID-TLV defined in this document), symbolic path name, association object (association-id defined by PCE, association-type SR-P2MP-PAG), policy identifier TLV(root and tree-id(0)), candidate path identifier tlv(protocol, origin, discriminator (32 bit value, which needs to be generated by PCE, and uniquely identifies a candidate path. This will increment for every candidate path and starts from 0)). The CC-ID list will be empty since PCE has not discovered any leaves yet. Root in response to the PC Initiate message will generate PLSP-ID and tree-id for the LSP Identifier and the candidate path that was downloaded by the PCE for this replication policy. PCC reports back the PLSP-ID, LSP-ID (SR-P2MP-LSP-ID-TLV defined in this document) and tree-id, and any leaves that were discovered until then to PCE. PCE on discovering leaves from the root, will compute the path to the leaves and downloads the label-information by sending PC Initiate message on the transit and leaf nodes connected to PCE and sends a PC Update message to the Root. The PC Initiate message to the transit and leaf is like the PC Initiate message that was sent to the Root, but the Tree-ID will not be 0 and will be the Tree-ID that the Root communicated through the Pc Report.

2. PCC Initiated Procedure:

Root sends a PC Report to PCE including the root, tree-id, PLSP-ID, LSP-ID (SR-P2MP-LSP-ID-TLV defined in this document), symbolic-path-name, and any leaves learnt until then. PCE on receiving this report, will compute the path and download label information on the leaf and

transit using PC Initiate message as PLSP-ID = 0, symbolic path name, association object (association-id defined by PCE, association-type SR-P2MP-PAG), policy identifier TLV(root and tree-id), candidate path identifier tlv(protocol, origin, discriminator (32 bit value, which needs to be generated by PCE, and uniquely identifies a candidate path. This will increment for every candidate path and starts from 0), CC-ID list of label downloads. On the root it will be an update message containing the PLSP-ID and other information that was earlier communicated by the Root. The association-ID used in the Root, transit and leaves will be the same for all candidate paths. Transit and Leaf on receiving the Initiate message will generate a PLSP-ID and report the status of the label downloads.

Beyond this, procedures for (1) and (2) are same.

[draft-ietf-pce-pcep-extension-for-pce-controller] indicates the PLSP-ID used in PC Initiate messages is the PLSP-ID defined at the ingress node, to allow correlation between transit instructions and the ingress LSP entity. For Replication Policy, as defined in the above procedures, other identifiers allow correlation of transit to root/ingress instructions to the downstream so the same PLSP-ID is not required. PLSP-ID's are individually generated by every PCC in the P2MP path.

Any new leaves discovered from here on, are reported to PCE using the PLSP-ID of the active candidate path. If these leaves are discovered on routers that are part of the P2MP LSP path, then PC Update is sent from PCE to necessary PCCs (LEAVES, TRANSIT or ROOT) with the LSPs PLSP-ID. If the new leaves are discovered on routers that are not part of the P2MP Tree yet, then a PC Initiate message is sent down with PLSP-ID=0.

Any new candidate path is downloaded by PCE to its connected Root, transit and leaves by sending a PC Initiate message to them. Every candidate path is a different P2MP LSP which gets a unique PLSP-ID. Multiple candidate path are associated to the same Replication policy and each used as a redundant P2MP LSP.

If a candidate path needs to be removed, PCE sends PC Initiate message, setting the R-flag in the LSP object and R bit in the SRP-object. To remove the entire P2MP-LSP, PCE needs to send PC Initiate remove messages for every candidate path of the SR-P2MP-POLICY to all the PCE connected nodes along the P2MP-LSP path. The R bit in the LSP Object as defined in rfc8231, refers to the removal of the LSP as identified by the SR-P2MP-LSP-ID-TLV (defined in this document). An all zero (SR-P2MP-LSP-ID-TLV defines to remove all the state of the corresponding PLSP-ID.

A candidate path is made active based on the preference of the path. If the Root gets paths one from the PCE and one from the CLI, and based on its tie-breaking rules, if it selects the CLI path, it will send a report to PCE for the PCE path indicating the status of label-download and sets operational bit of the LSP object to UP and Not Active . At any instance, only one path will be active.

3.2.2. Global Optimization for P2MP LSPs

When a P2MP LSP needs to be optimized for any reason (i.e. it is taking a FRR Path or new routers are added to the network) a global optimization is possible. Note that optimization works per candidate path. Each candidate path is capable of global optimization. To do so each candidate path contains two P2MP LSP, each P2MP LSP is identified via the LSP-ID [RFC3209]. After calculating an optimized P2MP LSP path the PCE will program the candidate path with a 2nd LSP-ID and its set of CCI instructions. After the optimized LSP is downloaded a MBB procedure is performed and the previous instance of the P2MP LSP is deleted and removed from the corresponding PCCs. The globally optimized LSP is instantiated via the PCInitiate message. The PLSP-ID of this optimized LSP is same as the Current LSP which is being optimized, this is because both LSPs belong to the same candidate path. That said the LSP-ID of the optimize LSP is uniquely assigned by PCE and is different from that of the current LSP which is being optimized. In short, the LSP-ID uniquely identifies sub-instances of an LSP for optimization with in that candidate path. After the optimized LSP has been downloaded and verified via PCC PCRpt message, the MBB procedure can be performed to switch between the two instances of the LSP. The previous instance will be removed from PCCs.

3.2.3. Fast Reroute

Currently this draft identifies the Facility FRR procedures. In addition, only LINK Protection procedures are defined. How the Facility Path is built and instantiated is beyond the scope of this document.

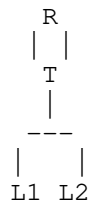


Figure 1

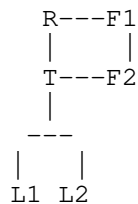


Figure 2

As an example, the bypass path (unicast bypass) between the PLR and MP can be constructed via SR. The PCE needs to only update the PLR PCC with bypass path outgoing label and nexthop information, also PCE needs to update the MP PCC with bypass path ILM information. This information is presented via a P bit in the optional IPv4/IPv6-Address object as per upcoming section.

If one to one FRR is needed, then a second flag O should be defined in the IPv4/IPv6-Address object in future.

As an example, in figure 1 the detour path between R and T is the 2nd fiber between these nodes. As such the bypass path could be setup on the 2nd fiber using treeSID procedures. That said in figure 2 the bypass path is traversing multiple nodes and this example a unicast SR path might be ideal for setting up the detour path. The PCE can download the prefix SID for F2 as a bypass path for R-T to R. Downloading the prefix SID for F2 will ensure an LFA detour for R-T. In addition, PHP procedure and implicit null label on the bypass path can be implemented to reduce the PCE programming on the MP PCC.

3.2.3. Connecting Replication Policy via Segment List

There could be nodes between two replication segment that do not understand P2MP Policy or Replication policy. It is possible to connected two non-adjacent Replication segment via a unicast binding

SID or segment-list.

Replication policy does support the concept of a segment-list. A list of unicast SIDs (Binding SID, Adjacency SIDs or Node SIDs) can be programmed on a Replication segment via the P2MP CCI object.

3.3. SR P2MP New TLVs and Objects

A new object <SR-P2MP-CCI> is defined for the controller to specify the forwarding instructions (label instructions) of the replication policy.

A new Association Type (P2MP SR Policy) is defined. Also a SR-P2MP policy identifiers TLV is defined to communicate the SR-P2MP policy and candidate path information.

A new P2MP-LSP identifier TLV is included to indicate the P2MP identifiers (Root, Tree-ID, LSP-ID).

The above new objects and TLV's defined in this document can be included in PcReport, PcInitiate and PcUpdate messages.

It should be noted that every PcReport, PcInitiate and PCUpdate messages will contain full list of the Leaves and label and forwarding information that is needed to build the P2MP LSP. In short the PCC or PCE should never send the delta information related to the new leaves that need to be added or updated. This is necessary to ensure that PCE or any new PCE is in sync with the PCC.

As such when a PcReport, PcInitiate and PCUpdate messages is send via PCEP it maintains the previous instruction CC-IDs and create new CC-ID for the new instruction. This means the CC-IDs are maintained for each specific forwarding and label instructions until these instructions are deleted. For example, When the first leaf is added PCC gets instructions, CC-ID A on a particular transit node. On a second leaf add, according to the path calculated, PCE might just append the existing instruction A with B. This is done by sending a PC Update with CC-ID A and B.

4. Object Format

4.1. Open Message and Capability Exchange

Format of the open object

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Ver |   Flags |   Keepalive |   DeadTimer |           SID           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
//                               Optional TLVs                               //
|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

All the nodes need to establish a PCEP connection with the PCE.

During PCEP Initialization Phase, PCEP Speakers advertise their support of PCECC extensions to include the new Path Setup Type [draft-ietf-pce-pcep-extension-for-pce-controller]. Also need to set flags N, M, P in the STATEFUL-PCE-CAPABILITY TLV as defined in [draft-ietf-pce-stateful-pce-p2mp] section capability advertisement.

We extend the PCEP OPEN object by defining an optional TLV to indicate the PCE's capability to perform SR-P2MP path computations, New IANA capability type. The inclusion of this TLV in an OPEN object indicates that the sender can perform SR-P2MP path computations. This will be similar to the P2MP-CAPABILITY defined in [RFC8306] section 3.1.2 and a new value needs to be defined for P2MP Policy.

In addition a Assoc-Type-List TLV as per [draft-ietf-pce-association-group-07] section 3.4 should be send via PCEP open object with following association type

Association Type Value	Association Name	Reference
TBD1	P2MP SR Policy Association	This document

OP-CONF-Assoc-RANGE (Operator-configured Association Range) should not be set for this association type and must be ignored.

4.2. Symbolic Name in PCInitiate message from PCC

As per RFC8231 section 7.3.2. a Symbolic Path Name TLV should uniquely identify the P2MP path on the PCC. This symbolic path name is a human-readable string that identifies an P2MP LSP in the network. It needs to be constant through the life time of the P2MP path.

As an example in the case of P2MP LSP the symbolic name can be Root + Tree-ID of the LSP. The Tree-ID is a unique ID that identifies the P2MP LSP on the Root (Source) as such the combination of Root + Tree-ID will provide the P2MP LSP with a unique identification throughout the network. Depending on the Source IP, IPv4 vs. IPv6, the length of the TLV will vary.

- 4.3. Replication policy As per [draft-voyer-spring-sr-p2mp-policy] a replication policy is build of candidate paths. Each candidate path contains p2mp-cci object which may contain a single outgoing label, in case it is directly connected to another replication segment, or a segment list to connect two non adjacent replication segments.

The candidate path and segment list has been described in [draft-ietf-spring-segment-routing-policy].

Candidate paths can be used for P2MP LSP redundancy where the active candidate path in a replication policy has a higher precedence over other candidate paths.

As such and as per [draft-barth-pce-segment-routing-policy-cp] section-3.1 each candidate path of a Replication policy appears as a different SR P2MP LSP (identified via a PLSP-ID) in PCEP, it is useful to group together all the candidate paths that belong to the same Replication policy. Furthermore, it is useful for the PCE to have knowledge of the P2MP SR candidate path parameters such as Root, Tree-ID, protocol origin, discriminator, and preference.

In this draft we define new association group objects to make above possible.

4.3.1 P2MP Policy Association Group

Two ASSOCIATION object types for IPv4 and IPv6 are defined in [I-D.ietf-pce-association-group]. The ASSOCIATION object includes "Association type" indicating the type of the association group. This document adds a new Association type.

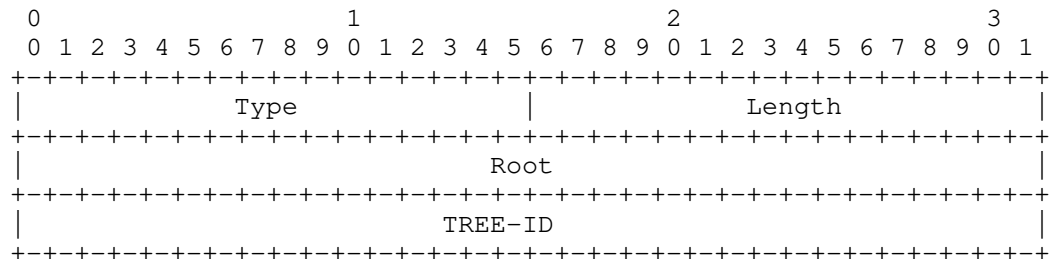
Association type = TBD1 "P2MP SR Policy Association Type" for SR Policy Association Group (P2MP SRPAG).

As per [draft-barth-pce-segment-routing-policy-cp] section 5, three new TLVs are identified to carry association information: P2MP-SRPAG-POL-ID-TLV, P2MP-SRPAG-CPATH-ID-TLV, P2MP-SRPAG-CPATH-ATTR-TLV

4.3.1.1 P2MP SR Policy Association Group Policy Identifiers TLV

The P2MP-SRPOLICY-POL-ID TLV is a mandatory TLV for the P2MP-SRPAG

Association. Only one P2MP-SRPOLICY-POL-ID TLV can be carried and only the first occurrence is processed and any others MUST be ignored.



Type: TBD2 for "P2MP-SRPOLICY-POL-ID" TLV.

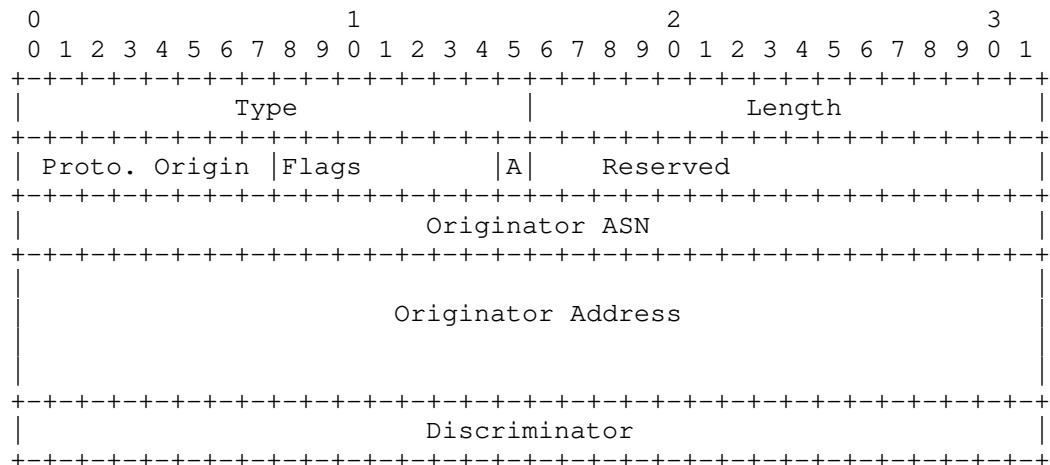
Length: 8 or 20, depending on length of End-point (IPv4 or IPv6)

Tunnel Sender Address : Can be either IPv4 or IPv6, this value is the value of the root loopback IP.

Tree-ID: Tree ID that the replication segment is part of as per draft-ietf-spring-sr-p2mp-policy

4.3.1.2 P2MP SR Policy Association Group Candidate Path Identifiers TLV

The P2MP-SRPOLICY-CPATH-ID TLV is a mandatory TLV for the P2MPSRPAG Association. Only one P2MP-SRPOLICY-CPATH-ID TLV can be carried and only the first occurrence is processed and any others MUST be ignored.



Type: TBD3 for "P2MP-SRPOLICY-CPATH-ID" TLV.

Length: 28.

Protocol Origin: 8-bit value that encodes the protocol origin, as specified in [I-D.ietf-spring-segment-routing-policy] Section 2.3.

Flags : A: This candidate path is active. At any instance only one candidate path can be active. PCC indicates the active candidate path to PCE through this bit.

Reserved: MUST be set to zero on transmission and ignored on receipt.

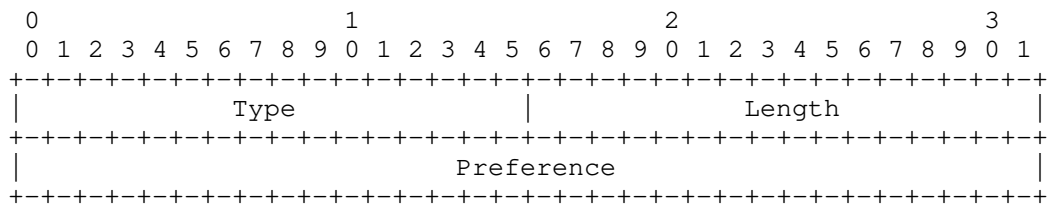
Originator ASN: Represented as 4 byte number, part of the originator identifier, as specified in [I-D.ietf-spring-segment-routing-policy] Section 2.4.

Originator Address: Represented as 128 bit value where IPv4 address are encoded in lowest 32 bits, part of the originator identifier, as specified in [I-D.ietf-spring-segment-routing-policy] Section 2.4.

Discriminator: 32-bit value that encodes the Discriminator of the candidate path.

4.3.1.3 P2MP SR Policy Association Group Candidate Path Attributes TLV

The P2MP-SRPOLICY-CPATH-ATTR TLV is an optional TLV for the SRPAG Association. Only one P2MP-SRPOLICY-CPATH-ATTR TLV can be carried and only the first occurrence is processed and any others MUST be ignored.



Type: TBD4 for "P2MP-SRPOLICY-CPATH-ATTR" TLV.

Length: 4.

Preference: Numerical preference of the candidate path, as specified in [I-D.ietf-spring-segment-routing-policy] Section 2.7.

If the TLV is missing, a default preference of 100 as specified in [I-D.ietf-spring-segment-routing-policy] is used.

4.4. PCEP Message Exchanges

PCE is informed of the P2MP request through manual configuration of root and leaves on the controller or through a Report from Root. On Reception of the P2MP Request, PCE initiates the P2MP LSP on the nodes connected along the P2MP Policy path that are connected to PCE. The object ordering of PC-Init, PC-Update and PC-Report messages are as per [draft-ietf-pce-association-group] section-6.3 (object Encoding in PCEP messages)

Format of PC InitiateMessage:

```
<Common Header>
<SRP>
<LSP>
<association-list>
<SR-P2MP-CCI>
```

Format of PC Update Message:

```
<Common Header>
<SRP>
<LSP>
<association-list>
<SR-P2MP-CCI>
```

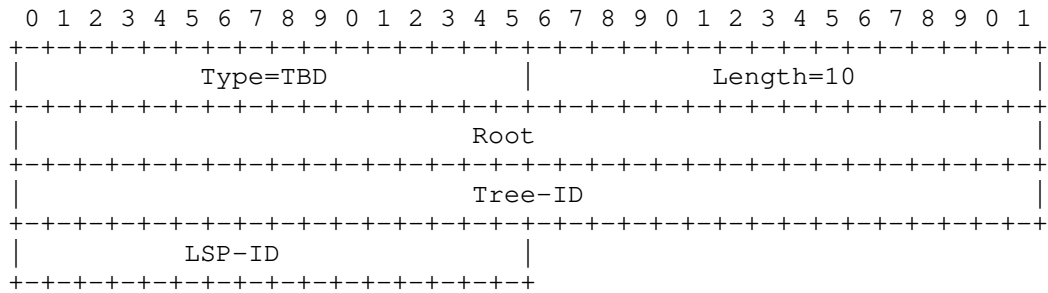
Every SR-P2MP-LSP's LSP Object MUST include the SR-P2MP-LSP-ID-TLV (IPV4/IPv6) which is defined by this document as below. This is a variation to the P2MP object defined in [draft-ietf-pce-stateful-pce-p2mp]

4.4.1 Extension of the LSP Object, SR-P2MP-LSPID-TLV

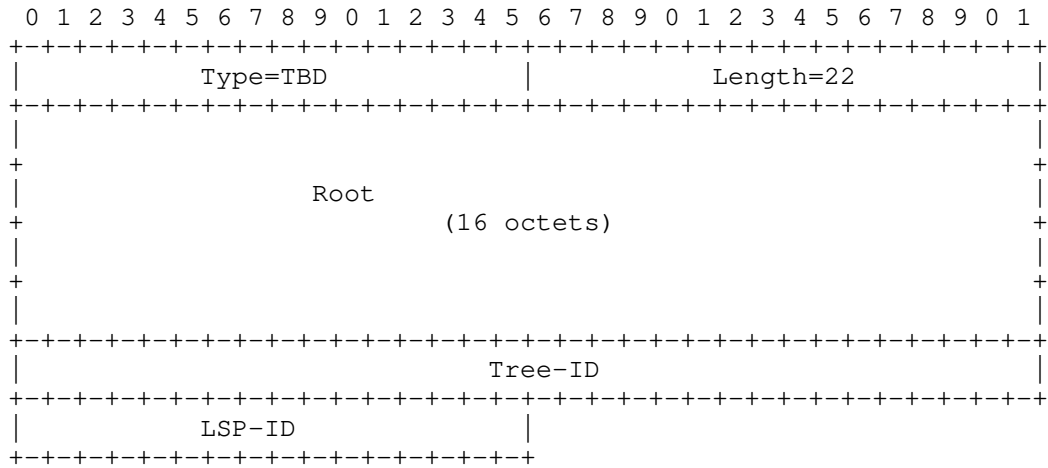
The LSP Object is defined in Section 7.3 of [RFC8231]. It specifies the PLSP-ID to uniquely identify an LSP that is constant for the life time of a PCEP session. Similarly for a P2MP tunnel, the PLSP-ID identify a candidate path (P2MP-LSP) uniquely with in the Replication policy.

The LSP Object MUST include the new SR-P2MP-LSPID-TLV (IPV4/IPv6). This is a variation to the P2MP object defined in [draft-ietf-pce-stateful-pce-p2mp]

SR-IPV4-P2MP-LSP-IDENTIFIER TLV:



SR-IPV6-P2MP-LSP-IDENTIFIER TLV :



The type (16-bit) of the TLV is TBD (need allocation by IANA).

LSP-ID : Contains 16 Bit LSP ID defined in rfc 3209.

Root: Source Router IP Address

Tree-ID: Unique Identifier of this P2MP LSP on the Root.

4.5. SR-P2MP-CCI Object

This is a variation of the CC-ID object defined in [draft-ietf-pce-pcep-extension-for-pce-controller]

The SR-P2MP-CCI is used to download label instruction to the nodes.

It can contain optional IPv4/IPv6 IP-Address TLVs that include forwarding instructions. These instructions includes incoming label (incoming replication SID), or out-going label for adjacent replication policies or Fast Reroute labels. Even segment list labels connecting two non adjacent replication policy can be downloaded via this object.

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     CC-ID                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Reserved                   |           Flags                   | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
//                                     Optional TLV                               //
|                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

CC-ID: A PCEP-specific identifier for the CCI information. A PCE creates an CC-ID for each instruction, the value is unique within the scope of the PCE and is constant for the lifetime of a PCEP session. The values 0 and 0xFFFFFFFF are reserved and MUST NOT be used. Flags:
 0 - Down - means label download was not successful 1 - Up - means label download was successful

4.5.1 Optional IP-Address TLVs

These optional IPv4/IPv6 TLVs can be including in P2MP CCI Object for forwarding information download.

Optional TLV:IPV4-ADDRESS TLV:

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Type=TBD                   | Length = 12                   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           SID-List Size               | Rsvd                         | Flag|I|B|S|E|P|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           IPv4 address                 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Interface ID                 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~                                     labels                                     ~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

SID-List Size:
 is the number of SIDs in the SID List

Flags:

I - Incoming Label:

If set means In Label, If not set means Out Label.

B - Bud Node Label:

If set this label is a bud node, the payload needs to be processed locally and also replicated if the S bit is set. In short if B is set then S needs to set also.

S - SWAP label:

0: If I bit is set and S bit is 0 it means pop the label and if the label's S bit is set do a recursive lookup.

1 - If I bit is set and S bit is 1 it means swap this label with out label.

P - Protection NextHop

0 - Label information is not w.r.t protection next-hop.

1 - Label information is w.r.t protection next-hop.

Note: P flag is used at the PLR and MP to identify the facility tunnel.

E - Protected LTN, This bit is usually set with the an outgoing label, when the outgoing label is protected via a protection nextHop

0 - Label information does not have a protection next-hop.

1 - Label information has a protection next-hop.

IPv4 address and Interface Id:

correspond to the next-hop information in case of an OUT Label, and it corresponds to incoming interface information if it is an IN Label.

Labels:

can be a single label or a list of labels, with the first label in the list being the label on top of the stack and the last label in the list being the label at the bottom of the stack.

IPV6-ADDRESS TLV:

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type=TBD                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     SID-List Size                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Flag                                     |I|B|S|E|P|
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     IPv6 address (16 bytes)                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Interface ID                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
~                                     labels                                     ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

UNNUMBERED-IPV4-ID-ADDRESS TLV:

```

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type=TBD                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     SID-List Size                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Node-ID                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Interface ID                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
~                                     labels                                     ~
+-----+-----+-----+-----+-----+-----+-----+-----+

```

4.6. Root PCE Report message

In order for the Root to indicate operations of its leaves (Add/Remove/Modify/DoNotModify), the PC Report message is extended to include P2MP End Point <P2MP End-points> Object which is defined in [RFC8306]

The format of the PC Report message is as follow:

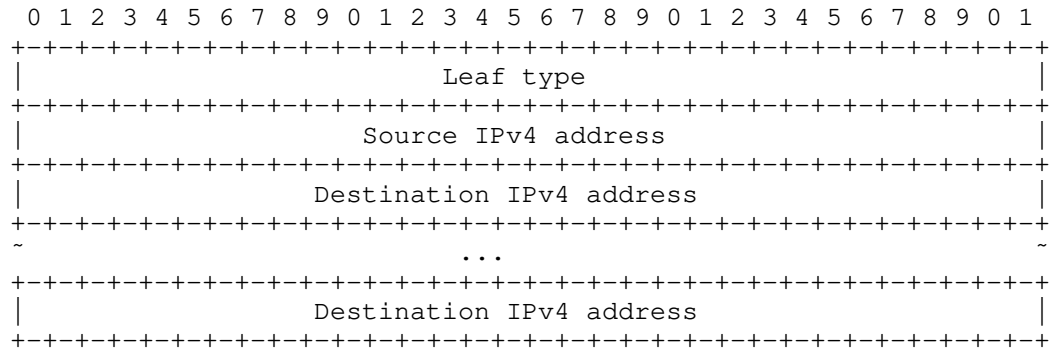
```

<Common Header>
[<SRP>]
<LSP>
[<association-list>]
[<end-points-list> | <SR-P2MP-CCI>]

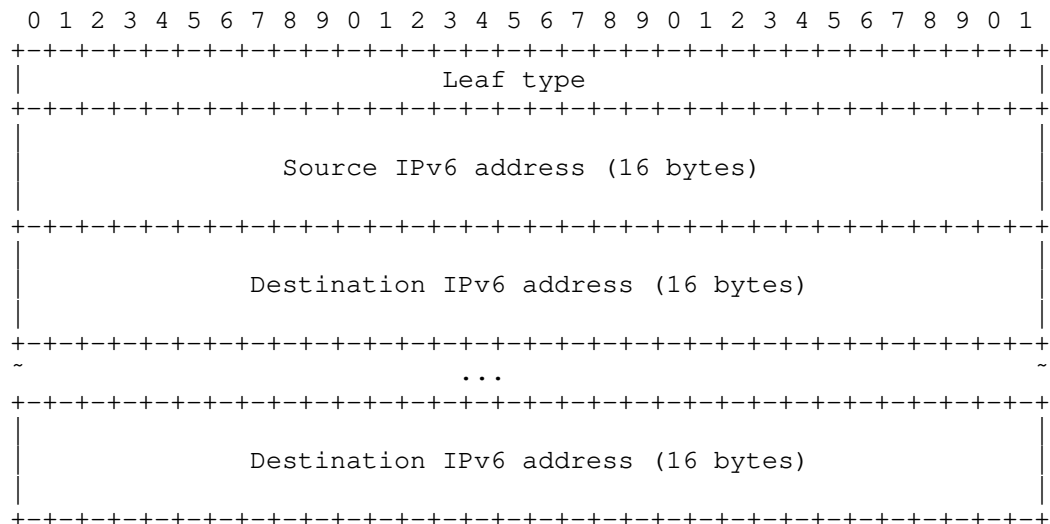
```

4.6.1 END-POINTS Objects

IPv4-P2MP END-POINTS:



IPv6-P2MP END-POINTS:

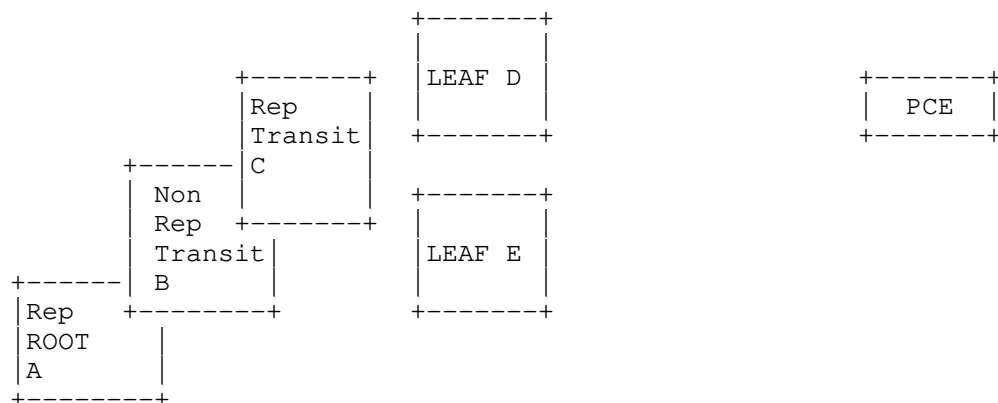


Leaf Types (derived from RFC 8306 section 3.3.2) :

- 1.New leaves to add (leaf type = 1)
- 2.Old leaves to remove (leaf type = 2)
- 3.Old leaves whose path can be modified/reoptimized (leaf type = 3), Future reserved not used for tree SID as of now.
- 4.Old leaves whose path must be left unchanged (leaf type = 4)

A given P2MP END-POINTS object gathers the leaves of a given type. Note that a P2MP report can mix the different types of leaves by including several P2MP END-POINTS objects. The END-POINTS object body has a variable length. These are multiples of 4 bytes for IPv4, multiples of 16 bytes, plus 4 bytes, for IPv6.

5. Examples of PCEP messages between PCE and PCEP



5.1. PCE Initiate and PCC Leaves Update

For a PCE Initiate P2MP Policy a sample PC Initiate message from the PCE to the root is provided below. This is on reception of a P2MP Policy creation on the PCE:


```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
<SRP OBJECT>
+-----+
|                               Flags = 0                               |
+-----+
|                               SRP-ID-number = 1                       |
+-----+
| TLV Type = 28 (PathSetupType) | TLV Len = 4                         |
+-----+
|                               | PST = TBD                             |
+-----+
<LSP OBJECT>
+-----+
|                               PLSP-ID = 0                               |
|                               | A:1,D:1,N:1,C:1                       |
+-----+
|                               Type=17                               |
|                               | Length=<var>                           |
+-----+
|                               symbolic path name                     |
+-----+
|                               Type=TBD                               |
|                               | Length                               |
+-----+
|                               Root = A                               |
+-----+
|                               Tree-ID                               |
+-----+
|                               LSP-ID =L1                             |
+-----+
<ASSOCIATION OBJECT>
+-----+
|                               Reserved                               |
|                               | Flags                               | 0 |
+-----+
| Association type= SR-P2MP-PAG | Association ID = z                   |
+-----+
|                               IPv4 Association Source = <pce-address> |
+-----+
|                               Type                               |
|                               | Length                               |
+-----+
|                               Root = A                               |
+-----+
|                               TREE-ID = 0                           |
+-----+
|                               Type                               |
|                               | Length                               |
+-----+
| ProtOrigin 10 | Flags                               | 0 | Reserved |
+-----+
|                               Originator ASN                         |
+-----+

```

```

+-----+
|                                     |
|                               Originator Address = <pce-address>          |
|                                     |
+-----+
|                                     |
|                               Discriminator = 0                          |
|                                     |
+-----+
|                               Type | Length                             |
+-----+
|                                     |
|                               Preference = 100 <default>                  |
|                                     |
+-----+

```

On Response to the above initiate message, PCC generates a Tree-ID, PLSP-ID for the candidate path identified by the candidate path identifier TLV and sends a report back to PCE. If leaves are discovered by the PCC at that point of time, that is communicated to the PCE in the same report message using the <p2mp-end-point> object in the Report message.

For PCC initiated P2MP Policy, if the Root wants to send a P2MP request to the PCE, the same is achieved through Root sending a PC Report to PCE indicating a P2MP Request.

Sample Report generated by the Root to the PCE for P2MP Request received from the Root Node:

Sample Report generated by the Root to the PCE for Leaf Add

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Flags = 0                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     SRP-ID-number  = 1                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| TLV Type = 28 (PathSetupType) | TLV Len = 4 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     | PST = TBD |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     <LSP OBJECT>                                     |
|                                     PLSP-ID = 1 | A:1,D:1,N:1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type=17 | Length=<var> |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     symbolic path name |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type=TBD | Length |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Root = A |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Tree-ID = Y |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     LSP-ID =L1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     <ASSOCIATION OBJECT>                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Reserved | Flags | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Association type= SR-P2MP-PAG | Association ID = z |
+-----+-----+-----+-----+-----+-----+-----+-----+
| IPv4 Association Source = <pce-address> |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Type | Length |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Root = A |
+-----+-----+-----+-----+-----+-----+-----+-----+
| TREE-ID = Y |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Type | Length |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ProtOrigin 10 | Flags | 0 | Reserved |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Originator ASN |
+-----+-----+-----+-----+-----+-----+-----+-----+
|

```

```

|                               Originator Address = <pce-address>                               |
|-----|
|                               Discriminator = 0                               |
|-----|
|                               Type                               |                               Length                               |
|-----|
|                               Preference = 100 <default>                               |
|-----|
|                               <END POINT OBJECT>                               |
|                               Leaf type =1                               |
|-----|
|                               Source IPv4 address = A                               |
|-----|
|                               Destination IPv4 address = D                               |
|-----|
|                               Destination IPv4 address = E                               |
|-----|

```

5.2. PCE P2MP LSP Calculation and Replication Policy download

Once the PC Report of leaves is sent to the PCE, PCE computes path to the leaf and would send a PC Initiate/ PC Update to the connected PCC's across the path to the leaf along with association object (defining association parameters, SR-P2MP policy identifier TLV, SR-P2MP-candidate path identifier TLV, candidate path attributes TLV) and label download information (SR-P2MP-CCI).

For example, say PCE computed 2 candidate paths <cp1 and cp2> that needs to be downloaded on the transit and root node, sample messages are explained below.

For cp1 -> on the root it will be a PC Update message sent from PCE, updating the empty candidate path it had sent earlier when it had intimated the root about the <root, tree-ID> it had known from NMS. For cp2 -> on the root it will be a PC Initiate messages sent from PCE, initiating the new candidate path and associating it to the same SR-P2MP policy.

On the transit - for cp1, and cp2 since PCE is initiating both newly on those nodes, PCE will send one PC Initiate message with two LSP objects, defining each candidate path. Or PCE can send separate PC Initiate message for every candidate path. As defined in [draft-barth-pce-segment-routing-policy-cp] section multiple candidate paths

A sample PC Update message sent to the Root for cp1 is as follows:

Note Root is connected to the next replication Segment C via non replication segment B. Hence a segment List is used.

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Flags = 0                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     SRP-ID-number  = 2                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| TLV Type = 28 (PathSetupType) | TLV Len = 4 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                                                                   | PST = TBD |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     <LSP OBJECT>                                     |
|                                     PLSP-ID = 1 |                                     A:1,D:1,N:1,C:0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type=17 |                                     Length=<var> |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     symbolic path name |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type=TBD |                                     Length |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Root =A |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Tree-ID = Y |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     LSP-ID = L1 |
+-----+-----+-----+-----+-----+-----+-----+-----+

|                                     <ASSOCIATION OBJECT>                                     |
|                                     Reserved |                                     Flags | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Association type= SR-P2MP-PAG | Association ID = z |
+-----+-----+-----+-----+-----+-----+-----+-----+
| IPv4 Association Source = <pce-address> |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type |                                     Length |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Root = A |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     TREE-ID = Y |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type |                                     Length |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ProtOrigin 10 | Flags | 0 | Reserved |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Originator ASN |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Originator Address = <pce-address> |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

```

|
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Discriminator = 0                                     |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type                                     | Length |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Preference = 100 <default>                                     |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     <CC-ID OBJECT>                                     |
|                                     CC-ID = z                                     |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Reserved                                     | Flags | 0 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type=TBD                                     | Length = 12 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| sid-list-size = 2 | Rsvd | Flag|0|0|0|0|0|0|
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     IPv4 address = NH                                     |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Interface ID                                     |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Label= b,c                                     |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

A sample PC Initiate message to the Root for cp2 is as follows:
Note cp2 can be either on the same path as cp1 or on a seprate path, assuming that there is a 2nd path connecting A to B to C

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Flags = 0                                     |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     SRP-ID-number = 3                                     |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| TLV Type = 28 (PathSetupType) | TLV Len = 4 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     | PST = TBD |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     <LSP OBJECT>                                     |
|                                     PLSP-ID = 0 | A:1,D:1,N:1,C:1 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type=17 | Length=<var> |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     symbolic path name |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type=TBD | Length |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```



```

+-----+
|                                     |
|                               Interface ID                               |
|                                     |
|                               Label= c1, b1                             |
|                                     |
+-----+

```

A sample PC Initiate message to the transit replication policy C for cpl
 Lets assume C is connected to D and C via 2 fiber hence Fast Reroute is possible:

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+
|                                     |
|                               Flags = 0                               |
|                                     |
|                               SRP-ID-number = 4                       |
|                                     |
| TLV Type = 28 (PathSetupType) | TLV Len = 4                         |
|                                     |
|                                     | PST = TBD                       |
|                                     |
+-----+
|                               <LSP OBJECT>                           |
|                               PLSP-ID = 0                             |
|                                     | A:1,D:1,N:1,C:1                 |
|                               Type=17                               |
|                                     | Length=<var>                     |
|                               symbolic path name                     |
|                                     |
|                               Type=TBD                               |
|                                     | Length                           |
|                               Root=A                                   |
|                                     |
|                               Tree-ID                                 |
|                                     |
|                               LSP-ID =L1                             |
|                                     |
+-----+
|                               <ASSOCIATION OBJECT>                   |
|                               Reserved                               |
|                                     | Flags                           | 0 |
|                               Association type= SR-P2MP-PAG         |
|                                     | Association ID = z             |
|                               IPv4 Association Source = <pce-address> |
|                                     |
|                               Type                                   |
|                                     | Length                           |
+-----+

```

```

+-----+
|                                     Root = A                                     |
+-----+
|                                     TREE-ID                                     |
+-----+
|      Type      |      Length      |
+-----+
| ProtOrigin 10 | Flags      | 0 | Reserved |
+-----+
|                                     Originator ASN                             |
+-----+
|                                     Originator Address = <pce-address>             |
+-----+
|                                     Discriminator = 0                           |
+-----+
|      Type      |      Length      |
+-----+
|                                     Preference = 100 <default>                     |
+-----+
|                                     <CC-ID OBJECT>                               |
|                                     CC-ID = z20                                   |
+-----+
|      Reserved      |      Flags      | 0 |
+-----+
|      Type=TBD      |      Length = 12 |
+-----+
| sid-list-size = 1 | Rsvd      | Flag|E|0|0|0|0|
+-----+
|                                     IPv4 address                               |
+-----+
|                                     Interface ID                               |
+-----+
|                                     Label= c1                                   |
+-----+
|                                     <incoming label c1 swap with D1>               |
|                                     CC-ID = z21                                   |
+-----+
|      Reserved      |      Flags      | 0 |
+-----+
|      Type=TBD      |      Length = 12 |
+-----+
| sid-list-size = 1 | Rsvd      | Flag|0|0|S|E|0|
+-----+
|                                     IPv4 address =NHD1                           |
+-----+

```

```

|                                     Interface ID                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Label= D1                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     <With FRR over NH2>                          |
|                                     CC-ID = z22                                |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      Reserved      |      Flags      | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      Type=TBD      |      Length = 12      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  sid-list-size = 1 | Rsvd      | Flag|0|0|0|0|P|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      IPv4 address =NHD2      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Interface ID                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Label= D1Protect                             |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      <incoming label c1 swap with E1>                                         |
|      CC-ID = z23                                                             |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      Reserved      |      Flags      | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      Type=TBD      |      Length = 12      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  sid-list-size = 1 | Rsvd      | Flag|0|0|S|E|0|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      IPv4 address =NHE1      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Interface ID                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Label= E1                                 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      <With FRR over NH2>                                                      |
|      CC-ID = z24                                                            |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      Reserved      |      Flags      | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      Type=TBD      |      Length = 12      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  sid-list-size = 1 | Rsvd      | Flag|0|0|0|0|P|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      IPv4 address =NHE2      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Interface ID                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Label= E1Protect                             |

```

+-----+

5.3. PCC Rpt for PCE Update and Init Messages

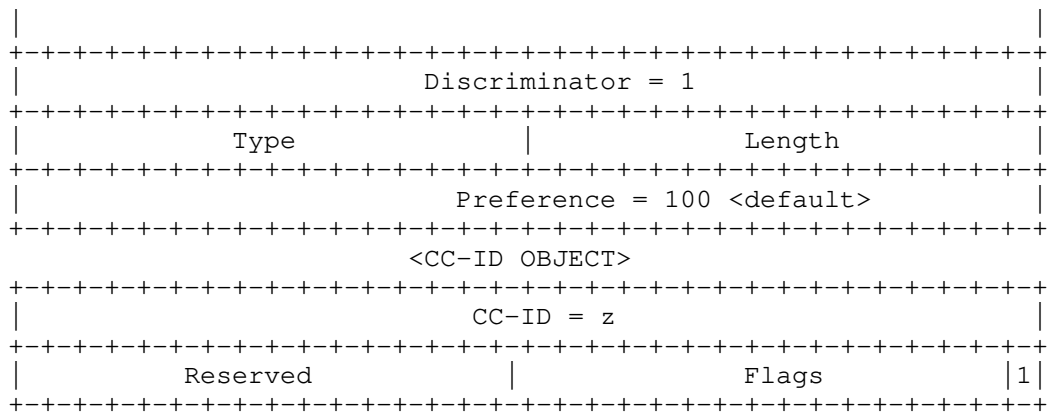
In response to the PC Initiate message / PC Update message , PCC will send PC Reports to PCE indicating the state of the label download for that particular candidate path. PCC's will generate PLSP-ID for newly initiated candidate path.

Here is an PC Report Message send for the root PCE Init message with cp2 on the root.

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Flags = 0                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     SRP-ID-number  = 2                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| TLV Type = 28 (PathSetupType) | TLV Len = 4 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     | PST = TBD |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     <LSP OBJECT>                                     |
|                                     PLSP-ID = 1 | O:1,A:1,D:1,N:1,C:1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type=17 | Length=<var> |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     symbolic path name |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type=TBD | Length |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Root |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Tree-ID |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     LSP-ID = L1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     <ASSOCIATION OBJECT>                                     |
|                                     Reserved | Flags | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Association type= SR-P2MP-PAG | Association ID = z |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     IPv4 Association Source = <pce-address> |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type | Length |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Tunnel Sender Address Ipv4 or Ipv6 =A |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     TREE-ID = Y |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Type | Length |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ProtOrigin 10 | Flags | 1 | Reserved |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Originator ASN |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Originator Address = <pce-address> |
+-----+-----+-----+-----+-----+-----+-----+-----+

```



6. Tree Deletion

To delete the entire tree (P2MP LSP) , Root send a PCRpt message with the R bit of the LSP object set and all the fields of the SR-P2MP-LSP-ID TLV set to 0(indicating to remove all state associated with this P2MP tunnel). The controller in response sends a PCInitiate message with R bit in the SRP object SET to all nodes along the path to indicate deletion of a label entry.

7. Fragmentation

The Fragmentation bit in the LSP object (F bit) can be used to indicate a fragmented PCEP message.

8. Example workflow

As per slides submitted in IETF 105.

6. IANA Considerations

This document contains no actions for IANA.

7. Security Considerations

TBD

8. References

8.1. Normative References

8.2. Informative References

[sr-p2mp-policy] D. Yoyer, Ed., C. Hassen, K. Gillis, C. Filsfils, R. Parekh, H. Bidgoli, "SR Replication Policy for P2MP Service Delivery", draft-yoyer-spring-sr-p2mp-policy-01, April 2019.

7. Acknowledgments

The authors would like to thank Tanmoy Kundu and Stone Andrew at Nokia for Thier feedback and major contribution to this draft.

Authors' Addresses

Hooman Bidgoli
Nokia
600 March Rd.
Ottawa, Ontario K2K 2E6
Canada

Email: hooman.bidgoli@nokia.com

Daniel Voyer
Bell Canada
Montreal
CA

Email: daniel.voyer@bell.ca

Siva Sivabalan
Cisco Systems
Ottawa
Canada

Email: msiva@cisco.com

Jayant Kotalwar
Nokia
380 N Bernardo Ave,
Mountain View, CA 94043
US

Email: jayant.kotalwar@nokia.com

Saranya Rajarathinam
Nokia
380 N Bernardo Ave,
Mountain View, CA 94043
US

Email: saranya.rajarathinam@nokia.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 26, 2021

H. Bidgoli, Ed.
Nokia
V. Voyer
Bell Canada
S. Rajarathinam
Nokia
E. Hemmati
Cisco System
T. Saad
Juniper Networks
S. Sivabalan
Ciena
May 25, 2021

PCEP extensions for p2mp sr policy
draft-hsd-pce-sr-p2mp-policy-03

Abstract

SR P2MP policies are set of policies that enable architecture for P2MP service delivery. This document specifies extensions to the Path Computation Element Communication Protocol (PCEP) that allow a stateful PCE to compute and initiate P2MP paths from a Root to a set of Leaves.

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 <https://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 November 26, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Conventions used in this document	4
3. Overview of PCEP Operation in SR P2MP Network	4
3.1. High level view of P2MP Policy Objects	5
3.1.1. Shared Tree vs Non-Shared Replication Segment	6
3.2. Existing drafts used for defining a P2MP Policy	7
3.2.1. Existing Documents used by this draft	7
3.2.2. P2MP Policy Identification	8
3.2.3. Replication Segment Identification	9
3.2.4. PCECC Use in Replication Segment	9
3.3. High Level Procedures for P2MP SR LSP Instantiation	9
3.3.1. PCE-Init Procedure	9
3.3.2. PCC-Init Procedure	10
3.3.3. Common Procedure	10
3.3.4. Global Optimization of the Candidate Path	11
3.3.5. Fast Reroute	12
3.3.6. Connecting Replication Segment via Segment List	13
3.4. SR P2MP Policy and Replication Segment TLVs and Objects	13
3.4.1. SR P2MP Policy Objects	13
3.4.2. Replication Segment Objects	14
3.4.3. P2MP Policy and Replication Segment general considerations	14
4. Object Format	15
4.1. Open Message and Capability Exchange	15
4.1.1. PCECC Path Setup Capability	15
4.1.2. Association Type Capability	16
4.2. Symbolic Name in PCInit Message from PCC	16
4.3. P2MP Policy Specific Objects and TLVs	16
4.3.1. P2MP Policy Association Group for P2MP Policy	16
4.3.1.1. P2MP SR Policy Association Group Policy Identifiers TLV	16
4.3.1.2. P2MP SR Policy Association Group Candidate Path Identifiers TLV	17
4.3.1.3. P2MP SR Policy Association Group Candidate Path Attributes TLV	18
4.3.2. P2MP-END-POINTS Object	18

4.4. P2MP Policy and Replication Segment Identifier Object and TLV	21
4.4.1. Extension of the LSP Object, SR-P2MP-LSPID-TLV	21
4.5. Replication Segment	22
4.5.1. The format of the replication segment message	23
4.5.2. PCECC	23
4.5.3. Label action rules in replicating segment	26
4.5.4. SR-ERO Rules	27
4.5.4.1. SR-ERO subobject changes	27
5. Tree Deletion	28
6. Fragmentation	28
7. Example Workflows	28
8. IANA Consideration	33
9. Security Considerations	34
10. Acknowledgments	34
11. References	34
11.1. Normative References	34
11.2. Informative References	34
Authors' Addresses	35

1. Introduction

The draft [draft-ietf-pim-sr-p2mp-policy] defines a variant of the SR Policy [draft-ietf-spring-segment-routing-policy] for constructing a P2MP segment to support multicast service delivery.

A Point-to-Multipoint (P2MP) Policy connects a Root node to a set of Leaf nodes, optionally through a set of intermediate replication nodes. A Replication segment [draft-ietf-spring-sr-replication-segment], which corresponds to the state of a P2MP segment on a particular node which provide forwarding instructions for the segment.

A P2MP Policy is relevant on the root of the P2MP Tree and it contains candidate paths. The candidate paths are made of path-instances and each path-instance is constructed via replication segments. These replication segments are programmed on the root, leaves and optionally intermediate replication nodes.

A replication segments MAY be connected directly, or they MAY be connected or steered via unicast SR segment or a segment list.

For a P2MP Tree, a controller may be used to compute paths from a Root node to a set of Leaf nodes, optionally via a set of replication nodes. A packet is replicated at the root node and optionally on Replication nodes towards each Leaf node.

There are two types of a P2MP Tree: Spray and Replication.

A Point-to-Multipoint service delivery could be via Ingress Replication, known as Spray. The root unicasts individual copies of traffic to each leaf. The corresponding P2MP Policy consists of replication segments only for the root and the leaves and they are connected via a unicast SR Segment.

A Point-to-Multipoint service delivery could also be via Downstream Replication, known as Replication. The root and some downstream replication nodes replicate the traffic along the way as it traverses closer to the leaves.

The leaves and the root can be explicitly configured on the PCE or PCC can update the PCE with the information of the discovered root and leaves. As an example Multicast protocols like MVPN procedures [RFC6513] or PIM can be used to discovery the leaves and roots on the PCC and update the PCE with these relevant information. The controller can calculate the P2MP Policy and any of its associated replication segments with these info.

This document defines PCEP objects, TLVs and the procedures to instantiate a P2MP Policy and Replication Segments.

2. Conventions used in this document

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].

3. Overview of PCEP Operation in SR P2MP Network

After discovering the root and the leaves on the PCE (via different mechanism mentioned in previous sections), the PCE computes the P2MP Tree and identifying the relevant Replication routers, then it programs the PCCs with relevant information needed to create a P2MP Tree.

As per draft [draft-ietf-pim-sr-p2mp-policy] a P2MP Policy is defined by Root-ID, Tree-ID and a set of leaves. A P2MP policy is a variant of SR policy as such it uses the same concept as draft [draft-ietf-pce-segment-routing-policy-cp]. A P2MP policy is composed of a collection of SR P2mp Candidate Paths. Candidate paths are computed by the PCE and can be used for P2MP Tree redundancy. Only a single candidate path may be active at each time. Each candidate paths can be globally optimized, therefore it is consists of multiple path-instances. A path-instance can be considered to a P2MP LSP. If a candidate path needs to be globally optimized two path-instances can be programmed on the root node and via make before break procedures the candidate path can be switched from path-

instance 1 to the 2nd path-instance. The forwarding states of these path-instances are build via replication segments, in short each path-instance initiated on the root has its own set of replication segments on the Root, Transit and Leaf nodes.

A replication segment is set of forwarding instructions on a specific node. Each instruction may be a PUSH or SWAP operation before forwarding out of an interface, or a POP action on bud and leaf nodes.

PCE could also calculate and download additional information for the replication segments, such as protections next-hops for link protection (FRR).

3.1. High level view of P2MP Policy Objects

o SR P2MP Policy

- * Is only relevant on the Root of the P2MP path and is a policy on PCE. It is downloaded only on the rootnode and is identified via <Root-ID, Tree-ID> It contains the following information:

- + Root node of the P2MP Segment
- + Leaf nodes of the P2MP Segment
- + Tree-ID, which is a unique identifier of the P2MP tree on the Root
- + A set of Candidate paths belonging to the policy
- + Optional Constraints used to build these candidate paths

o Candidate Path:

- * Is used for P2MP Tree redundancy where the candidate path with the highest preference is the active path.
- * It can contain two path-instance for global optimization procedures (i.e. make before break)
- * Contains information regarding protocol-id, originator, discriminator, preference, path-instances

o Replication Segment:

- * Is the forwarding information needed on each node for building the forwarding path for each path-instance of the P2MP Candidate path.
- * Explained further in upcoming sections, there are 2 ways to identify the replication segment, depending if they are shared and non-shared
 - + It is identified via Tree-ID and Root-ID and path-instance for non-shared replication segment.
 - + It is identified via Node-ID, Replication-ID, for shared replication segment
 - + Contains forwarding instructions, in the form of a list of outgoing segments each of which may be a list
 - + On the forwarding plane the Replication Segment is identified via the incoming Replication SID.
 - + Replication segment information is downloaded on Root, Transit and Leaf nodes respectively.

3.1.1. Shared Tree vs Non-Shared Replication Segment

A non-shared Replication Segment is used when the label field of the PMSI Tunnel Attribute (PTA) is set to zero as per [draft-parekh-bess-mvpn-sr-p2mp]. This is used when there is no upstream assigned label in the PTA (provider tunnel attribute) and aggregate of MVPNs into a single P-Tunnel is not desired.

An alternative shared Replication Segment is used when the label field of the PTA is not set to Zero and there is an upstream assigned label in the PTA. In this case multiple MVPNs (VRFs) can be aggregate into a single Provider Tunnel and the upstream assigned label distinguishes the MVPNs context.

It should be noted that the shared Replication Segment can also be used to build a bypass tunnel for the purpose of fast re-route. This might be desirable if the bypass tunnel is build via the PCE and downloaded to the PCC for link protection. In doing so, multiple non-shared Replication Segments can use the shared replication segment as their bypass tunnel for link protection. The replication segments used in this bypass tunnel should only create a unicast bypass tunnel to protect the link between two replication segments on the primary path.

3.2. Existing drafts used for defining a P2MP Policy

This document attempts to leverage existing IETF draft and RFC documents which define PCEP objects, to update the PCE with Root and Leaves information when PCC Initiated method is used. Similarly, existing documents are utilized where feasible to update the PCC with relevant information to build the P2MP Policy and its Replication Segments. This document introduces new TLVs and Objects specific to a programming P2MP policy and its replication segment.

3.2.1. Existing Documents used by this draft

- o [RFC8231] The bases for a stateful PCE, and reuses the following objects or a variant of them
 - * <SRP Object>
 - * <LSP Object>
 - * A variation of the LSP identifier TLV is defined in this draft, to support P2MP LSP Identifier
- o [RFC8236] P2MP capabilities advertisement
- o [draft-ietf-pce-segment-routing-policy-cp] Candidate paths for P2MP Policy is used for Tree Redundancy. As an example, a P2MP Policy can have multiple candidate paths. Each protecting the primary candidate path. The active path is chosen via the preference of the candidate path.
- o [RFC3209] Defines the instance-ID, instance-ID is used for global optimization of a candidate path with in a P2MP policy. Each Candidate path can have 2 path-instances. These path-instances are equivalent to sub-lsps (instance-IDs). There are used for MBB and global optimization procedures. instance-ID is equivalent to LSP ID
- o [draft-ietf-spring-segment-routing-policy] Segment-list, used for connecting two non-adjacent replication policy via a unicast binding SID or Segment-list.
- o [RFC8306] P2MP End Point objects, used for the PCC to update the PCE with discovered Leaves.
- o [draft-ietf-pce-pcep-extension-for-pce-controller] for programming and identifying the Replication Segment. A new PCE CC Capability sub Tlv is introduced to indicated the support to handle PCE CC based label download for SR P2MP.

- o [draft-ietf-pce-multipath] Forwarding instruction for a P2MP LSP is defined by a set of SR-ERO sub-objects in the ERO object, ERO-ATTRIBUTES object and MULTIPATH-BACKUP TLV as defined in this draft.
- o [RFC8664] SR-ERO Sub Object used in the multipath.

It should be noted that the [draft-dhs-spring-sr-p2mp-policy-yang] can provide further details of the high level P2MP Policy Model.

3.2.2. P2MP Policy Identification

A P2MP Policy and its candidate path can be identified on the root via the P2MP LSP Object. This Object is a variation of the LSP ID Object defined in [RFC8231] and is as follow:

- o PLSP-ID: [RFC8231], is assigned by PCC and is unique per candidate path. It is constant for the lifetime of a PCEP session. Stand-by candidate paths will be assigned a new PLSP-ID by PCC. Stand-by candidate paths can co-exist with the active candidate path.
 - * Note: Every candidate path in the SR-P2MP Policy is unique with its own unique PLSP-ID and Instance-ID. But the same Tree-ID is used for all candidate paths as they are part of the same P2MP Tree.
- o Root-ID: is equivalent to the first node on the P2MP path, as per [RFC3209], Section 4.6.2.1
- o Tree-ID: is equivalent to Tunnel Identifier color which identifies a unique P2MP Policy at a ROOT and is advertised via the PTA in the BGP AD route or can be assigned manually on the root. Tree-ID needs to be unique on the root.
- o Instance-ID: LSP ID Identifier as defined in RFC 3209, is the path-instance identifier and is assigned by the PCC. As it was mentioned the candidate path can have up to two path-instance for global optimization. Note that the Root-ID, Tree-ID and Instance-ID are part of a new SR- P2MP-LSP-IDENTIFIER TLV which will be identified in this draft.
 - * Note: each Path-instance on the Root node is assigned a unique Instance-ID

3.2.3. Replication Segment Identification

The key to identify a replication segment is also a P2MP LSP Object. With varying encoding rules for the SR-P2MP-LSP- IDENTIFIER TLV which will be explained in later sections.

3.2.4. PCECC Use in Replication Segment

PCECC and a variant of CCI object is used in Replication Segment to identify a cross connect. A cross connect is a incoming SID and set of outgoing interfaces and their corresponding SID. The CCI objects contains the incoming SID while the outgoing interfaces are presented via the ERO objects, which each may contain a list of segments.

3.3. High Level Procedures for P2MP SR LSP Instantiation

A P2MP policy can be instantiated via the PCC or the PCE depending on how the root and the leaves are discovered. This document describes two way to discover the root and the leaves:

- o They can be configured and identified on the controller and are considered PCE initiated.
- o They can be discovered on the PCC via MVPN procedures [RFC6513] or legacy multicast protocols like PIM or IGMP etc... and are considered PCC initiated.

3.3.1. PCE-Init Procedure

- o PCE is informed of the P2MP request through its API or configuration mechanism to instantiate a P2MP tunnel.
- o PCE will initiate the P2MP Policy for the request, by sending a PCInitiate message to the Root.
- o Root in response to the PCInitiate message, will generate PLSP-ID for the candidate paths and an Instance-ID for the Path-Instance (LSP-ID) contained with in the candidate path. The tree-id for the P2MP Policy is also filled. PCC will reports back the PLSP-ID, Instance-ID and tree-id via PCRpt message
 - * Optionally, the Root can add any additional leaves that were discovered by multicast procedures in this PCRpt message.
- o PCE will send a PCInitiate message to the Root, Transit and the Leaf nodes to download the Replication Segment information. These messages will utilize the CCI object to encode the forwarding instruction information.

- o PCE will then send a PCUpdate to the root indicating the association information (Candidate path) , and implicitly indicate it to bind to the latest CCI information downloaded.

3.3.2. PCC-Init Procedure

After Root (PCC) discovers the leaves (as an example via MVPN Procedures or other mechanism), the following communication happens between the PCE and PCCs

- o Root sends a PCRpt message for P2MP policy to PCE including the Root-ID, Tree-ID, PLSP-ID, Instance-ID, symbolic-path-name, and any leaves discovered until then.
- o PCE on receiving of this report, will compute the P2MP Policy and its replication segments.
 - * PCE will send a PCInitiate message to the Root, Transit and the Leaf nodes to download the Replication Segment information. These messages will utilize the CCI object to encode the forwarding instruction information.
 - * PCE will then send a PCUpdate to the root indicating the association information (Candidate path) , and implicitly indicate it to bind to the latest CCI information downloaded.

3.3.3. Common Procedure

The following procedures are the same for PCE or PCC Init.

- o PCE will download the replication segments for the Candidate-path's path-instances to all the leaves and transit nodes using PCInitiate message with PLSP-ID = 0, Instance-ID =0, symbolic path name, Root-address, Tree-id(assigned by the root). This PCInitiate message includes the EROs needed for the replication segments. These messages will utilize the CCI object to encode the forwarding instruction information.
- o Any new candidate path for the P2MP Policy is downloaded by PCE to the Root by sending a PCInitiate message
 - * it should be noted, PLSP-ID, Path-Instance ID and the Tree-ID are generated by the PCC for these new candidate paths and their Path-instances
 - * Any update to the Candidate Paths or Replication Segments is done via the PCUpd message. Association object need to be

present for Candidate Path updates and CCI object for the replication segment updates.

- o The PCE will also download the necessary replication segment for the candidate path and its path-instances to the root, leaves and the transit nodes via a PCInit message
- o New leaves can be discovered via Multicast procedures, and new replication segments can be instantiated or existing one updated to reach these leaves
 - * If these leaves reside on routers that are part of the P2MP LSP path, then PCUpd is sent from PCE to necessary PCCs (LEAVES, TRANSIT or ROOT) with the correct PLSP-ID, Instance-ID, Tree-ID and CC-ID.
 - * If the new leaves are residing on routers that are not part of the P2MP Tree yet, then a PCInitiate message is sent down with PLSP-ID=0 and Instance-ID=0 on the corresponding routers.
- o The active candidate-path is indicated by the PCC through the operational bits(Up/Active) of the LSP object in the PCRpt message. If a candidate path needs to be removed, PCE sends PC Initiate message, setting the R-flag in the LSP object and R bit in the SRP-object.
- o To remove the entire P2MP-LSP, PCE needs to send PCInitiate remove messages for every candidate path of the P2MP POLICY to the root and send PCInitiate remove messages for every Replication Segment on all the PCCs on the P2MP Tree. The R bit in the LSP Object as defined in [RFC8231], refers to the removal of the LSP as identified by the SR-P2MP-POLICY-ID-TLV (defined in this document). An all zero (SR-P2MP-LSP-ID-TLV defines to remove all the state of the corresponding PLSP-ID.
- o A candidate path is made active based on the preference of the path. If the Root is programmed with multiple candidate paths from different sources, as an example PCE and CLI, based on its tie-breaking rules, if it selects the CLI path, it will send a report to PCE for the PCE path indicating the status of label-download and sets operational bit of the LSP object to UP and Not Active . At any instance, only one path will be active

3.3.4. Global Optimization of the Candidate Path

When a P2MP LSP needs to be optimized for any reason (i.e. it is taking a FRR tunnel or new routers are added to the network) a global optimization of the candidate path is possible.

Each Candidate Path can contain two Path-Instances. The current unoptimized Path-Instance is the active instance and its replication segments are forwarding the multicast PDUs from the root to the leaves. However the second optimized Path-Instance will be setup with its own unique replication segments throughout the network, from the Root to the leaves. These two Path-Instances can co-exist. The two Path-Instances are uniquely identified by their Instance-ID in the SR-P2MP-POLICY-ID-TLV (defined in this document). After the optimized LSP has been downloaded successfully PCC MUST send two reports, reporting UP of the new path indicating the new LSP-ID, and a second reporting the tear down of the old path with the R bit of the LSP Object SET with the old Instance-ID in the SR-P2MP-POLICY-ID-TLV. This MBB procedure will move the multicast PDUs to the optimized Path-Instance.

The leaf should be able to accept traffic from both Path-Instances to minimize the traffic outage by the Make Before Break process.

3.3.5. Fast Reroute

Currently this draft identifies the Facility FRR procedures. In addition, only LINK Protection procedures are defined. How the Facility Path is built and instantiated is beyond the scope of this document.

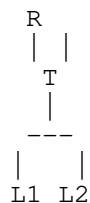


Figure 1

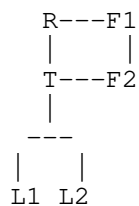


Figure 2

As an example, the bypass path (unicast bypass) between the PLR and MP can be constructed via SR or even via a shared tree (replication segment).

As an example, in figure 1 the detour path between R and T is the 2nd fiber between these nodes. As such the bypass path could be setup on the 2nd fiber. That said in figure 2 the bypass path is traversing multiple nodes and this example a unicast SR path might be ideal for setting up the detour path.

In addition, PHP procedure and implicit null label on the bypass path can be implemented to reduce the PCE programming on the MP PCC.

Optional shared replication segments can be used in networks that do not have unicast SR turned on. These shared replication segments can be programmed on the bypass nodes without a P2MP Policy. The replication segments on primary path can use these shared replication segments as a protection tunnel to protect links.

3.3.6. Connecting Replication Segment via Segment List

There could be nodes between two replication segment that do not support P2MP Policy or Replication segment. It is possible to connect two non-adjacent Replication segments via a unicast segment routing path via a SID list, comprised of any IGP supported segment types (ex: Binding, Adjacency, Node) to forward to the next replicating node. This information is encoded via the SR-ERO sub-objects and ERO-attributes objects. The last segment in an encoding SID list MUST be a replication segment

3.4. SR P2MP Policy and Replication Segment TLVs and Objects

3.4.1. SR P2MP Policy Objects

SR P2MP Policy can be constructed via the following objects

<Common Header>

<SRP>

<P2MP LSP>

[<association-list>]

optionally if the root is updating the PCE with end point list the end-point-list object can be added.

[<end-points-list>]

3.4.2. Replication Segment Objects

Replication segment can be constructed via the following objects

```

<Common Header>
<SRP>
<P2MP LSP>
(<cci-list>|
<CCI><intended-path>))
<cci-list> ::= <CCI>
               [<cci-list>]
<intended-path> ::= ((<PATH-ATTRIB><ERO>)
                    [<intended-path>])

```

Path-attribute as per [draft-ietf-pce-multipath]

3.4.3. P2MP Policy and Replication Segment general considerations

The above new objects and TLV's defined in this document can be included in PCRpt, PCInitiate and PCUpd messages.

It should be noted that every PCRpt, PCInitiate and PCUpd messages will contain full list of the Leaves and segment and forwarding information that is needed to build the Candidate path and its Replication segments. They will never send the delta information related to the new leaves or forwarding information that need to be added or updated. This is necessary to ensure that PCE or any new PCE is in sync with the PCC.

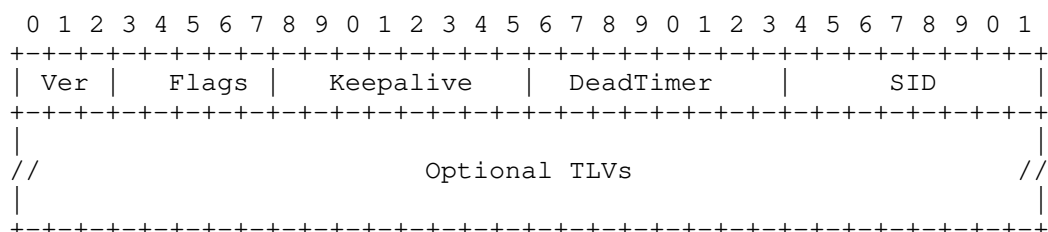
When a PCRpt, PCInitiate and PCUpd messages is sent via PCEP it maintains the previous ERO Path IDs and generates new Path IDs for new instructions, as per [draft-ietf-pce-multipath]. The PATH IDs are maintained for each specific forwarding instructions until the instructions are deleted. For example: When the first leaf is added, the PCE will update with PathID 1 to the PCC. When the second leaf is added, according to the path calculated, PCE might just append the existing instruction Path ID 1 with a new Path ID 2 to construct the new PCUpd message.

The CCI Object is used to identify the entire cross connect of incoming segment and the set of outgoing Interfaces and their corresponding SIDs/SIDList. Any modification to the cross connect should use this CCI ID to identify the cross connect uniquely. Leaves and their corresponding Path IDs can be removed from the cross connect identified via the CCI. The CC-ID is assigned by the PCE.

4. Object Format

4.1. Open Message and Capability Exchange

Format of the open Object:



All the nodes need to establish a PCEP connection with the PCE.

During PCEP Initialization Phase, PCEP Speakers need to set flags N, M, P in the STATEFUL-PCE-CAPABILITY TLV as defined in [draft-ietf-pce-stateful-pce-p2mp] section-5.2

This draft extends the PCEP OPEN object by defining an optional TLV to indicate the PCE's capability to perform SR-P2MP path computations with a new IANA capability type.

The inclusion of this TLV in an OPEN object indicates that the sender can perform SR-P2MP path computations. This will be similar to the P2MP-CAPABILITY defined in [RFC8306] section-3.1.2 and a new value needs to be defined for SR-P2MP.

4.1.1. PCECC Path Setup Capability

A PST of PCECC is also added as per [draft-ietf-pce-pcep-extension-for-pce-controller].

This document also introduces a new bit S in the SR PCECC capacity Sub TLV indicating the support to handle PCECC based label download for Replication segment.



Also, the N,M,P bits in STATEFUL-PCE-CAPABILITY TLV should be SET.

4.1.2. Association Type Capability

A Assoc-Type-List TLV as per [RFC8697] section 3.4 should be send via PCEP open object with following association type

Association Type Value	Association Name	Reference
TBD1	P2MP SR Policy Association	This document

OP-CONF-Assoc-RANGE (Operator-configured Association Range) should not be set for this association type and must be ignored.

The open message MUST include the MULTIPATH CAPABILITY TLV as defined in [draft-ietf-pce-multipath]

4.2. Symbolic Name in PCInit Message from PCC

As per [RFC8231] section 7.3.2. a Symbolic Path Name TLV should uniquely identify the P2MP path on the PCC. This symbolic path name is a human-readable string that identifies an P2MP LSP in the network. It needs to be constant through the lifetime of the P2MP path.

As an example in the case of P2MP LSP the symbolic name can be p2mp policy name + candidate path name of the LSP.

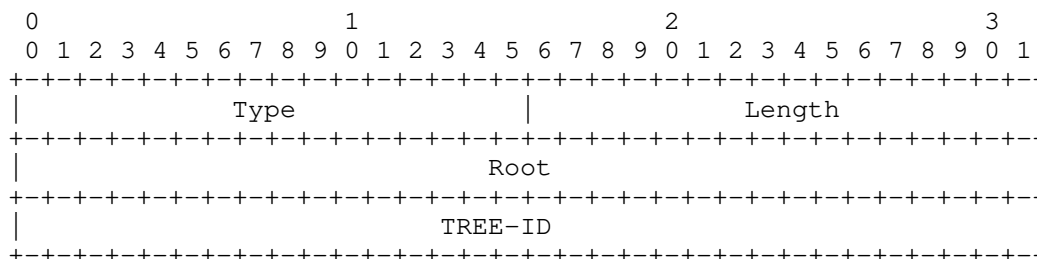
4.3. P2MP Policy Specific Objects and TLVs

4.3.1. P2MP Policy Association Group for P2MP Policy

Two ASSOCIATION object types for IPv4 and IPv6 are defined in [RFC8697]. The ASSOCIATION object includes "Association type" indicating the type of the association group. This document adds a new Association type. Association type = TBD1 "P2MP SR Policy Association Type" for SR Policy Association Group (P2MP SRPAG). As per [draft-barth-pce-segment-routing-policy-cp] section 5, three new TLVs are identified to carry association information: P2MP-SRPAG-POL-ID-TLV, P2MP-SRPAG-CPATH-ID-TLV, P2MP-SRPAG-CPATH-ATTR-TLV

4.3.1.1. P2MP SR Policy Association Group Policy Identifiers TLV

The P2MP-SRPOLICY-POL-ID TLV is a mandatory TLV for the P2MP-SRPAG Association. Only one P2MP-SRPOLICY-POL-ID TLV can be carried and only the first occurrence is processed and any others MUST be ignored.



Type: TBD2 for "P2MP-SR-POLICY-POL-ID" TLV.

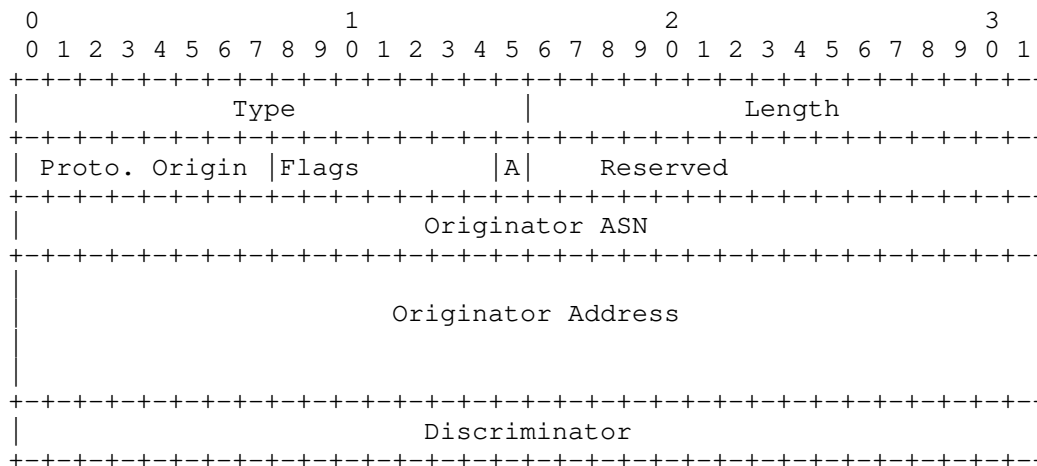
Length: 8 or 20, depending on length of End-point (IPv4 or IPv6)

Tunnel Sender Address : Can be either IPv4 or IPv6, this value is the value of the root loopback IP.

Tree-ID: Tree ID that the replication segment is part of as per draft-ietf-spring-sr-p2mp-policy

4.3.1.2. P2MP SR Policy Association Group Candidate Path Identifiers TLV

The P2MP-SRPOLICY-CPATH-ID TLV is a mandatory TLV for the P2MPSRPAG Association. Only one P2MP-SRPOLICY-CPATH-ID TLV can be carried and only the first occurrence is processed and any others MUST be ignored.



Type: TBD3 for "P2MP-SR-POLICY-CPATH-ID" TLV.

Length: 28.

Protocol Origin: 8-bit value that encodes the protocol origin, as specified in [I-D.ietf-spring-segment-routing-policy] Section 2.3.

Flags : A: This candidate path is active. At any instance only one candidate path can be active. PCC indicates the active candidate path to PCE through this bit. Reserved: MUST be set to zero on transmission and ignored on receipt.

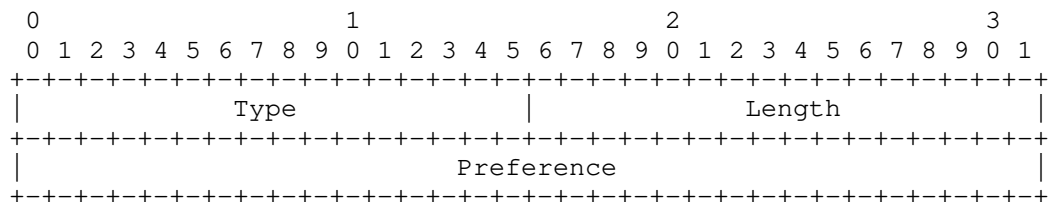
Originator ASN: Represented as 4 byte number, part of the originator identifier, as specified in [draft-ietf-spring-segment-routing-policy] Section 2.4.

Originator Address: Represented as 128 bit value where IPv4 address are encoded in lowest 32 bits, part of the originator identifier, as specified in [draft-ietf-spring-segment-routing-policy] Section 2.4.

Discriminator: 32-bit value that encodes the Discriminator of the candidate path.

4.3.1.3. P2MP SR Policy Association Group Candidate Path Attributes TLV

The P2MP-SRPOLICY-CPATH-ATTR TLV is an optional TLV for the SRPAG Association. Only one P2MP-SRPOLICY-CPATH-ATTR TLV can be carried and only the first occurrence is processed and any others MUST be ignored.



Type: TBD4 for "P2MP-SRPOLICY-CPATH-ATTR" TLV.

Length: 4. Preference: Numerical preference of the candidate path, as specified in [draft-ietf-spring-segment-routing-policy] Section 2.7.

If the TLV is missing, a default preference of 100 as specified in [draft-ietf-spring-segment-routing-policy] is used.

4.3.2. P2MP-END-POINTS Object

In order for the Root to indicate operations of its leaves (Add/Remove/Modify/DoNotModify), the PC Report message is

extended to include P2MP End Point <P2MP End-points> Object which is defined in [RFC8306]

The format of the PC Report message is as follow:

<Common Header>

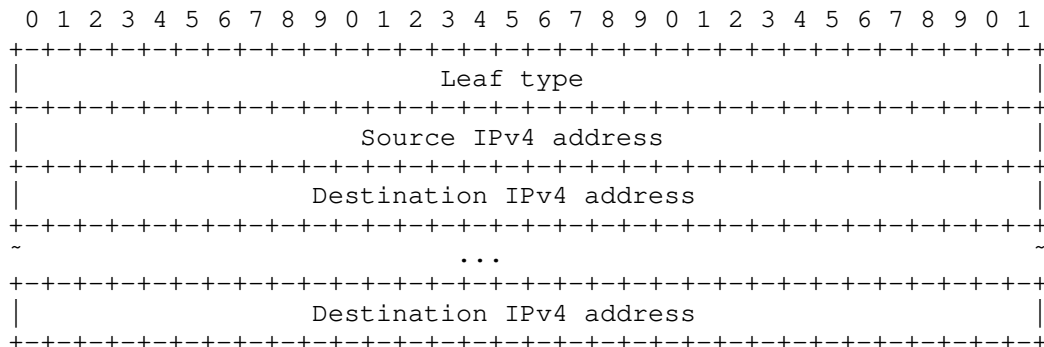
[<SRP>]

<LSP>

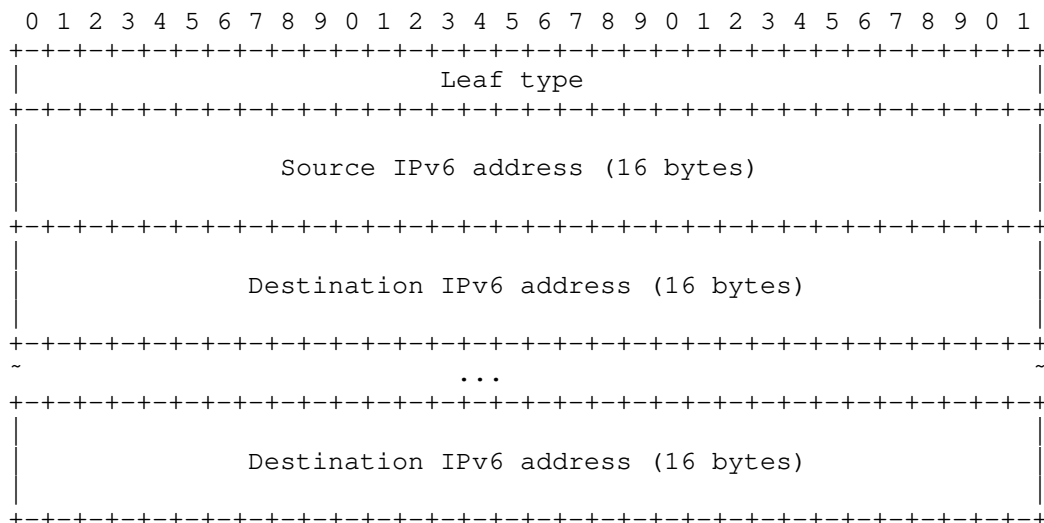
[<association-list>]

[<end-points-list>]

IPv4-P2MP END-POINTS:



IPv6-P2MP END-POINTS:



Leaf Types (derived from [RFC8306] section 3.3.2) :

1. New leaves to add (leaf type = 1)
2. Old leaves to remove (leaf type = 2)
3. Old leaves whose path can be modified/reoptimized (leaf type = 3), Future reserved not used for tree SID as of now.
4. Old leaves whose path must be left unchanged (leaf type = 4)

5. the entire pce leaf list is overwritten and replaced with the new leaf list (leaf type = 5)

A given P2MP END-POINTS object gathers the leaves of a given type. Note that a P2MP report can mix the different types of leaves by including several P2MP END-POINTS objects. The END-POINTS object body has a variable length. These are multiples of 4 bytes for IPv4, multiples of 16 bytes, plus 4 bytes, for IPv6.

4.4. P2MP Policy and Replication Segment Identifier Object and TLV

As it was mentioned previously both P2MP Policy and Replication Segment are identified via the LSP object and more precisely via the SR-P2MP-LSPID-TLV

The P2MP Policy uses the PLSP-ID to identify the Candidate Paths and the Instance-ID to identify a Path-Instance within the Candidate path.

On the other hand the Replication Segment uses the SR-P2MP-LSPID-TLV to identify and correlate a Replication Segment to a P2MP Policy

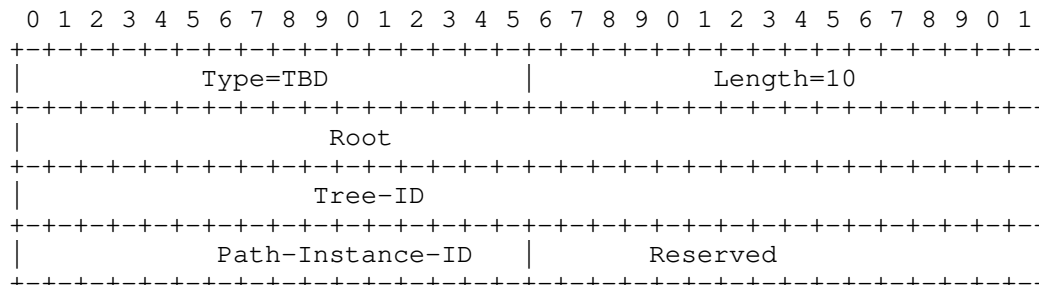
As it was noted previously on the Root, the P2MP Policy and the Replication Segment is downloaded via the same PCUpd message.

4.4.1. Extension of the LSP Object, SR-P2MP-LSPID-TLV

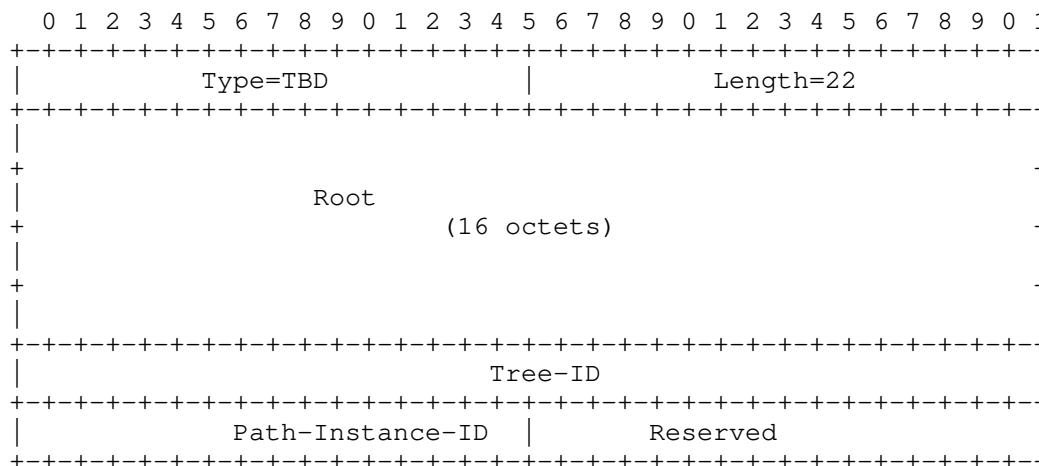
The LSP Object is defined in Section 7.3 of [RFC8231]. It specifies the PLSP-ID to uniquely identify an LSP that is constant for the life time of a PCEP session. Similarly, for a P2MP tunnel, the PLSP-ID identifies a Candidate Path uniquely within the P2MP policy.

The LSP Object MUST include the new SR-P2MP-POLICY-ID-TLV (IPv4/IPv6) defined in this document below. This is a variation to the P2MP object defined in [draft-ietf-pce-stateful-pce-p2mp]

SR-IPV4-P2MP-POLICY-ID TLV:



SR-IPV6-P2MP-POLICY-ID TLV :



The type (16-bit) of the TLV is TBD (need allocation by IANA).

Root: Source Router IP Address

Tree-ID: Unique Identifier of this P2MP LSP on the Root.

Instance-ID : Contains 16 Bit instance ID.

4.5. Replication Segment

As per [draft-ietf-spring-sr-replication-segment] a replication segment has a next-hop-group which MAY contain a single outgoing replication SID or a list of SIDs (sr-policy-sid-list) In either case there needs to be a replication SID at the bottom of the stack. This

means two replication segments can be directly connected or connected via a SR domain.

4.5.1. The format of the replication segment message

The format of a Replication Segment message encoding is similar to P2MP Policy. However, the P2MP Policy contains the association object and the replication segment message does not contain the association object. In addition the replication segment uses the CCI object to identify a P2MP cross connect. The replication segment is downloaded individually to the root, transit and leaf nodes without the P2MP Policy. The P2MP Policy is a Root Concept. The replication segment uses SR-P2MP-LSPID-TLV as its identifier. The TLV is coded differently for shared and non-shared case.

- o In the case of a replication segment being shared, the Tree-ID in the SR-P2MP-POLICY Identifier TLV is the replication-id of the Replication Segment and Root = 0, Instance-Id = 0. When downloading a shared replication segment from PCE through a PCEInitiate message, the SR-P2MP-POLICY Identifier TLV is all 0, and on the report back from PCC, PCC generates PLSP-ID, Replication-id (Tree-id field will be populated with replication-id). Instance-id will be 0.

4.5.2. PCECC

The CCI Object as defined in [draft-ietf-pce-pcep-extension-for-pce-controller] is used to identify a forwarding instruction in the Replication Segment. A forwarding instruction is incoming SID and a set of outgoing branches. The CCI Object-Type of 1 is used for the MPLS Label. The label in the CCI Object is the incoming SID. The outgoing SIDs are defined by the ERO Objects.

The CCI Object can be include in Reports, initiate and Update messages for Replication Segments.

The PCInitiate message defined in [RFC8281] and extended in [draft-ietf-pce-pcep-extension-for-pce-controller] is further extended to support SR-P2MP replication segment based central control instructions.

The format of the extended PCInitiate message is as follows:

```
<PCInitiate Message> ::= <Common Header>
                           <PCE-initiated-lsp-list>
```

Where:

<Common Header> is defined in [RFC5440]

```
<PCE-initiated-lsp-list> ::= <PCE-initiated-lsp-request>
                              [<PCE-initiated-lsp-list>]
```

```
<PCE-initiated-lsp-request> ::=
    (<PCE-initiated-lsp-instantiation> |
     <PCE-initiated-lsp-deletion> |
     <PCE-initiated-lsp-central-control>)
```

```
<PCE-initiated-lsp-central-control> ::= <SRP>
                                         <LSP>
                                         (<cci-list> |
                                         (<CCI><intended-path>))
```

```
<cci-list> ::= <CCI>
               [<cci-list>]
```

```
<intended-path> ::= ((<PATH-ATTRIB><ERO>)
                     [<intended-path>])
```

Where:

<PCE-initiated-lsp-instantiation> and
<PCE-initiated-lsp-deletion> are as per
[RFC8281].

The LSP and SRP object is defined in [RFC8231]. The <intended-path> is as per [RFC8281] [draft-ietf-pce-multipath] (PATH-ATTRIB and ERO).

The format of the PCRpt message is as follows:

```
<PCRpt Message> ::= <Common Header>
                        <state-report-list>
```

Where:

```
<state-report-list> ::= <state-report> [<state-report-list>]
```

```
<state-report> ::= (<lsp-state-report> |
                    <central-control-report>)
```

```
<lsp-state-report> ::= [<SRP>]
                        <LSP>
                        <path>
```

```
<central-control-report> ::= [<SRP>]
                              <LSP>
                              (<cci-list> |
                               (<CCI><intended-path>))
```

```
<cci-list> ::= <CCI>
               [<cci-list>]
```

Where:

<path> is as per [RFC8231] and the LSP and SRP object are also defined in [RFC8231].

The <intended-path> is as per [draft-ietf-pce-multipath] (PATH-ATTRIB and ERO).

This document extends the use of PCUpd message with SR-P2MP CCI as follows:

```
<PCUpd Message> ::= <Common Header>
                        <update-request-list>
```

Where:

```
<update-request-list> ::= <update-request> [<update-request-list>]
```

```
<update-request> ::= (<lsp-update-request> |
                        <central-control-update>)
```

```
<lsp-update-request> ::= <SRP>
                        <LSP>
                        <path>
```

```
<central-control-update> ::= <SRP>
                        <LSP>
                        (<CCI><intended-path>)
```

Where:

<path> is as per [RFC8231] and the LSP and SRP object are also defined in [RFC8231].

The <intended-path> is as per [draft-ietf-pce-multipath] (PATH-ATTRIB and ERO).

4.5.3. Label action rules in replicating segment

The node action and role of ingress, transit, leaf or bud, is indicated via a new Node Role TLV. This document introduces a new SR-P2MP-NODE-ROLE TLV (Type To be assigned by IANA) that will be present in the PATH-ATTRIB object.

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |                                     |
|          Type=TBD                  |          Length=4              |
+-----+-----+-----+-----+-----+-----+-----+-----+
|   Role Type   |                                     | Reserved         |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

- o ingress, role type = 1
- o transit, role type = 2
- o leaf, role type = 3
- o bud, role type = 4

4.5.4. SR-ERO Rules

Forwarding information of a replication segment can be configured and steered via many different mechanisms.

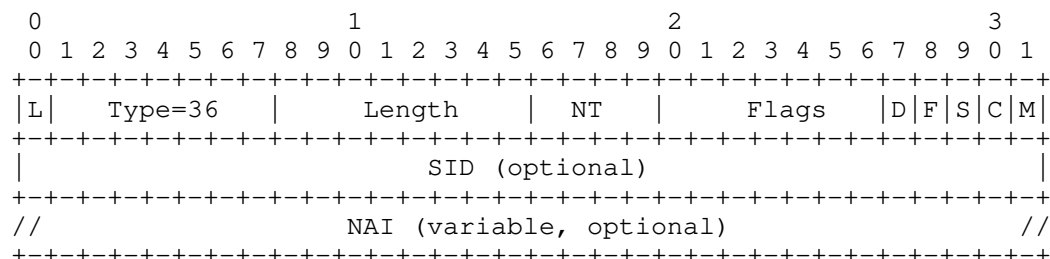
As an example a replication SID can be steered via:

1. Replication SID steered with an IPv4/IPv6 directly connected nexthop
 - * In this case there will be two SR-ERO in the ERO Object, with the Replication SID SR-ERO at the bottom and the IPv4/IPv6 SR-ERO on the top.
2. Replication SID steered with an IPv4/IPv6 loopback address that reside on the directly connected router.
 - * In this case there will be two SR-ERO in the ERO Object, with the Replication SID SR-ERO at the bottom and the IPv4/IPv6 SR-ERO on the top.
 - * In addition a new flag D is added to the SR-ERO to signal that the Loopback nexthop is connected to the directly attached router.
3. Replication SID steered with unnumbered IPv4/IPv6 directly connected Interface
4. Replication SID steered via a SR adjacency or node SID
 - * In this case even a sid-list can be used to traffic engineer the path between two Replication Segment
 - * The Replication SID SR-ERO is at the bottom while the segments describing the path are on top in order.

4.5.4.1. SR-ERO subobject changes

SR-ERO from RFC 8664 is used to construct the forwarding information needed for Replication Segment.

A new D flag was added to indicate a loopback nexthop that is residing on the directly attached router. It should be noted that this flag should be set only for the loopback case and not for a local interface as a nexthop.



Flags : F, S, C, M are already defined in rfc8664.

This document defines a new flag D: If the next-hop in NAI field is system IP or loopback, this bit indicates whether the system IP / loopback is directly connected router or not. If set indicates directly connected address. When this bit is set, F bit should be 0 (meaning NAI should be present)

5. Tree Deletion

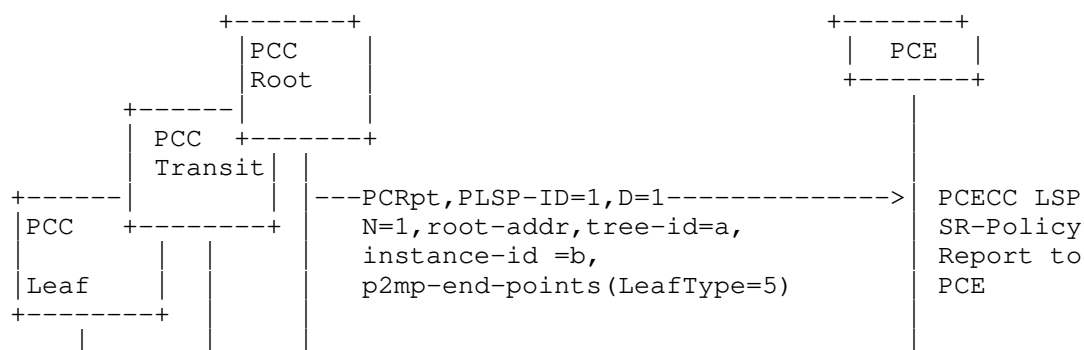
To delete the entire tree (P2MP LSP), Root send a PCRpt message with the R bit of the LSP object set and all the fields of the SR-P2MP-LSP-ID TLV set to 0(indicating to remove all state associated with this P2MP tunnel). The PCE in response sends a PCInitiate message with R bit in the SRP object SET to all nodes along the path to indicate deletion of the entries.

6. Fragmentation

The Fragmentation bit in the LSP object (F bit) can be used to indicate a fragmented PCEP message

7. Example Workflows

PCC-Initiated Workflow

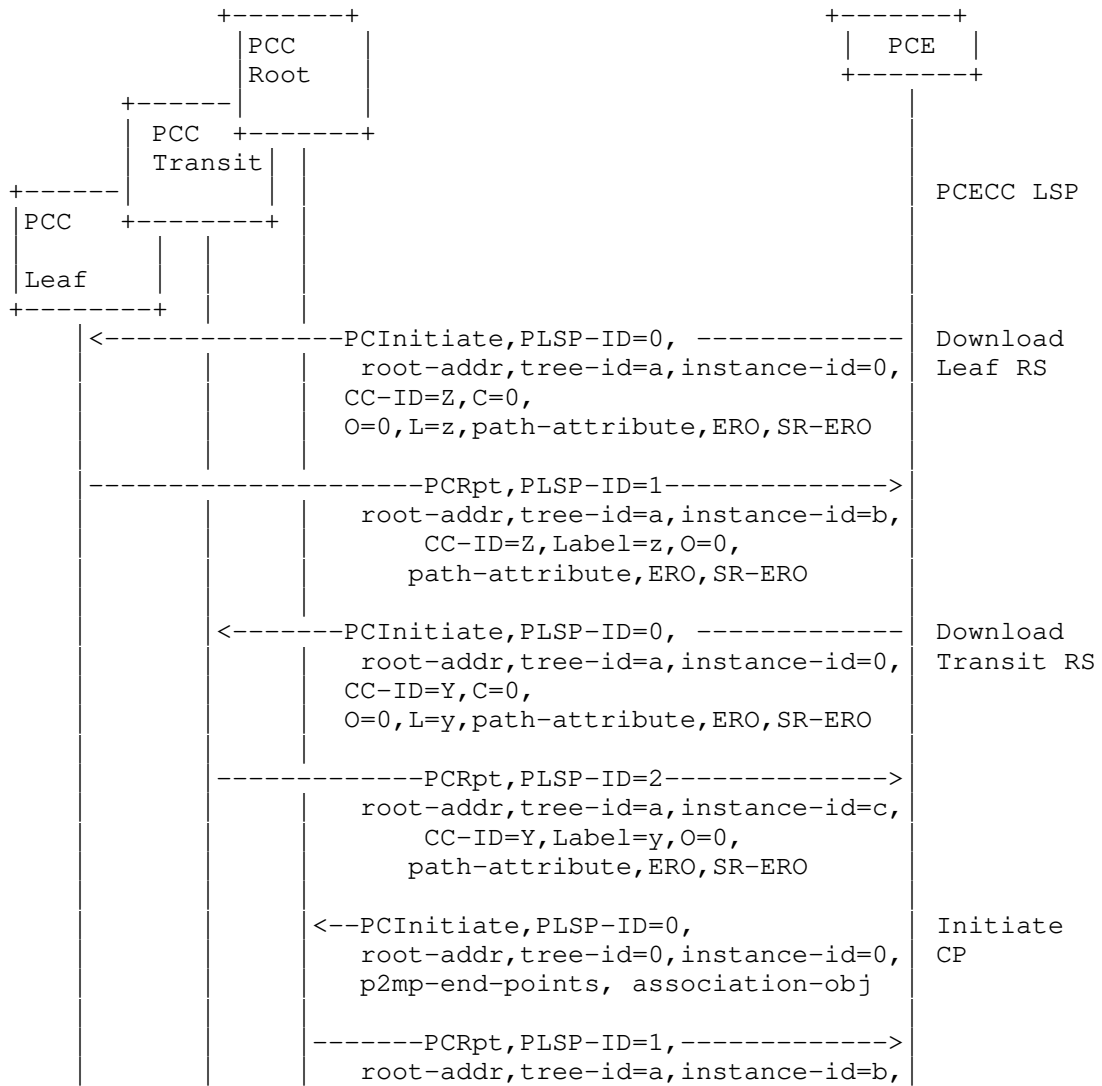


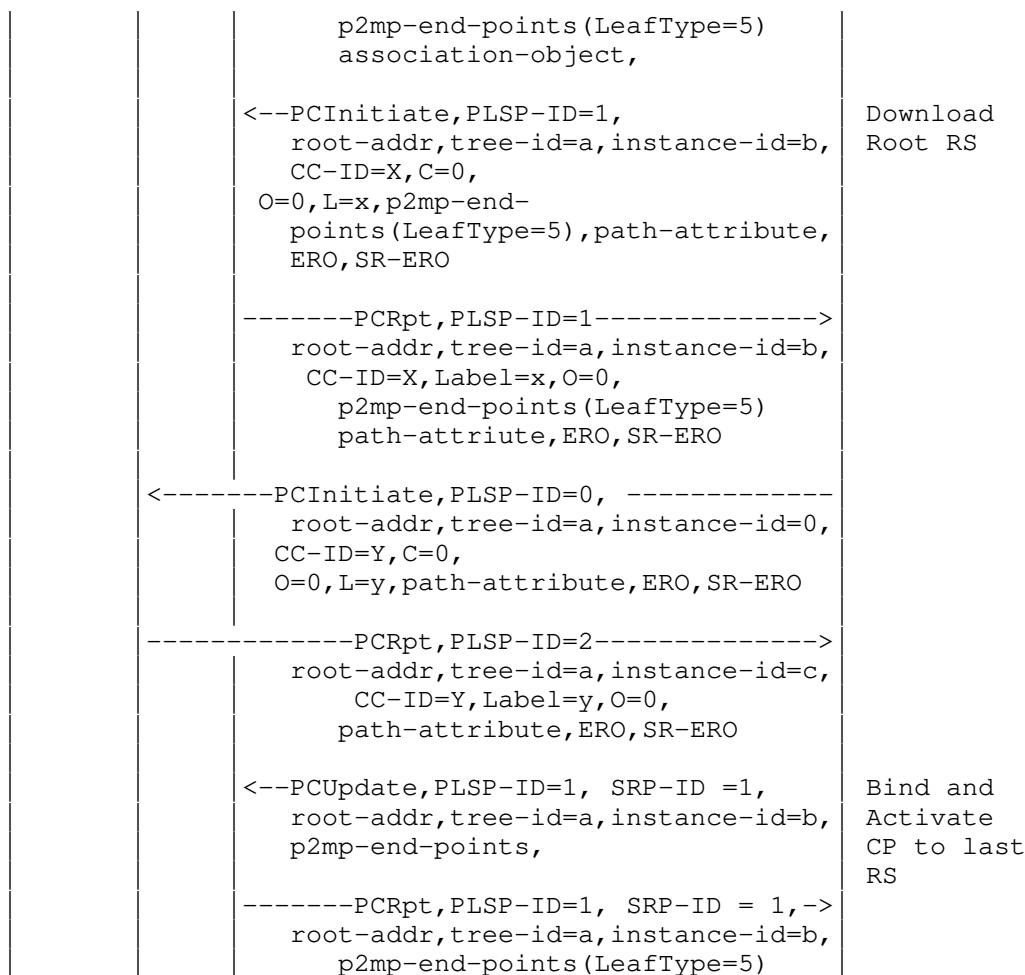
		<pre><--PCUpdate,PLSP-ID=1, SRP-ID =1, root-addr,tree-id=a,instance-id=b, p2mp-end-points, association-obj</pre>	Update CP
		<pre>-----PCRpt,PLSP-ID=1, SRP-ID = 1,-> root-addr,tree-id=a,instance-id=b, p2mp-end-points(LeafType=5) association-object,</pre>	
<-----		<pre>PCInitiate,PLSP-ID=0, ----- root-addr,tree-id=a,instance-id=0, CC-ID=Z,C=0, O=0,L=z,path-attribute,ERO,SR-ERO</pre>	Download Leaf Replication Segment (RS)
		<pre>-----PCRpt,PLSP-ID=1-----> root-addr,tree-id=a,instance-id=b, CC-ID=Z,Label=z,O=0, path-attribute,ERO,SR-ERO</pre>	
	<-----	<pre>PCInitiate,PLSP-ID=0, ----- root-addr,tree-id=a,instance-id=0, CC-ID=Y,C=0, O=0,L=y,path-attribute,ERO,SR-ERO</pre>	Download Transit RS
		<pre>-----PCRpt,PLSP-ID=2-----> root-addr,tree-id=a,instance-id=c, CC-ID=Y,Label=y,O=0, path-attribute,ERO,SR-ERO</pre>	
		<pre><--PCInitiate,PLSP-ID=1, root-addr,tree-id=a,instance-id=b, CC-ID=X,C=0, O=0,L=x,p2mp-end- points(LeafType=5),path-attribute, ERO,SR-ERO</pre>	Download Root RS
		<pre>-----PCRpt,PLSP-ID=1-----> root-addr,tree-id=a,instance-id=b, CC-ID=X,Label=x,O=0, p2mp-end-points(LeafType=5) path-attribute,ERO,SR-ERO</pre>	
		<pre><--PCUpdate,PLSP-ID=1, SRP-ID =2, root-addr,tree-id=a,instance-id=b, p2mp-end-points</pre>	Activate CP to last RS
		<pre>-----PCRpt,PLSP-ID=1, SRP-ID =2, -> root-addr,tree-id=a,instance-id=b,</pre>	

p2mp-end-points (LeafType=5)

Note that on transit / leaf Initiate is with PLSP-ID = 0. Therefore PLSP-ID is locally unique to a node. It should be noted that the CC-ID does not need to be constant across all nodes that make up the path.

PCE-Initiated workflow

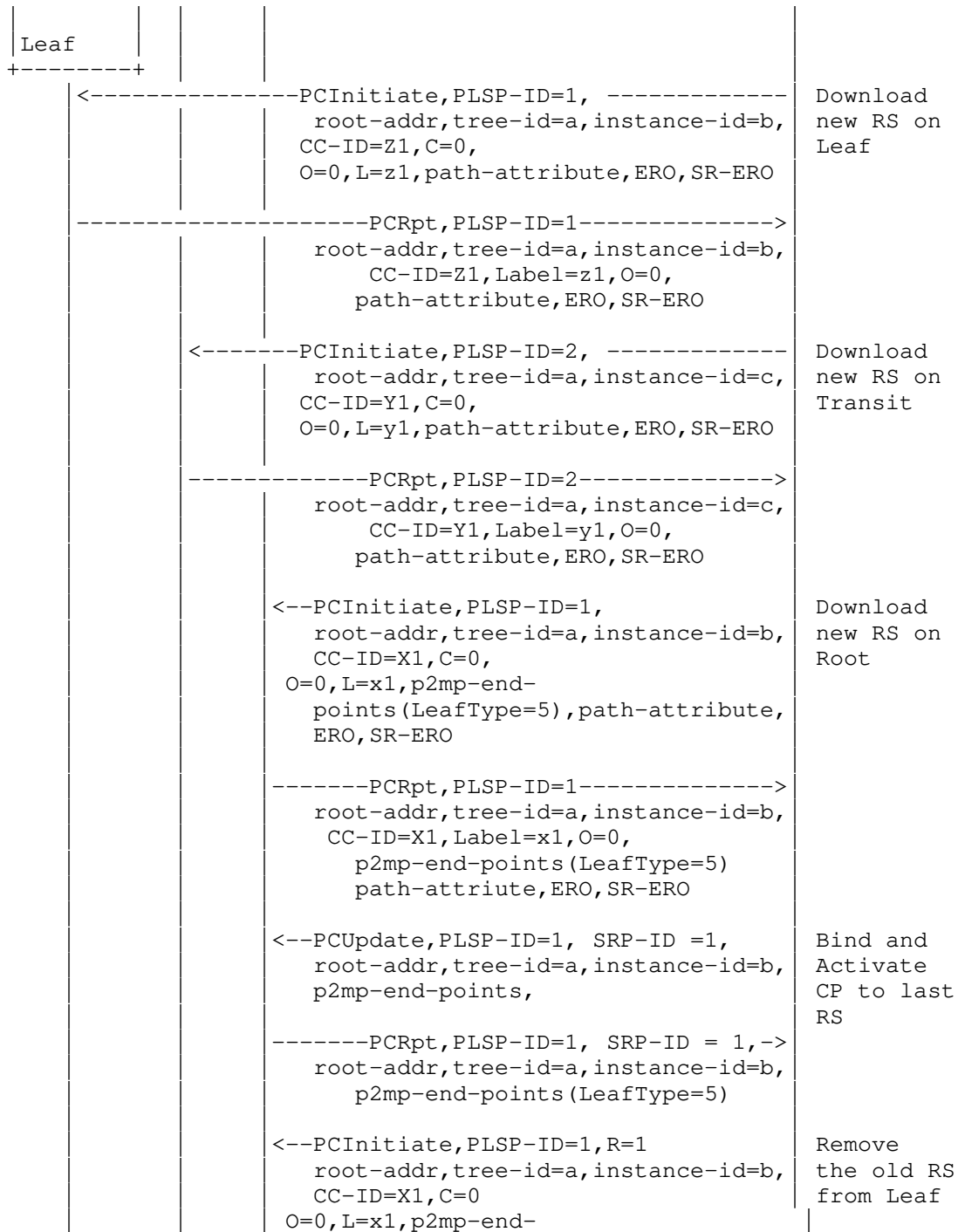


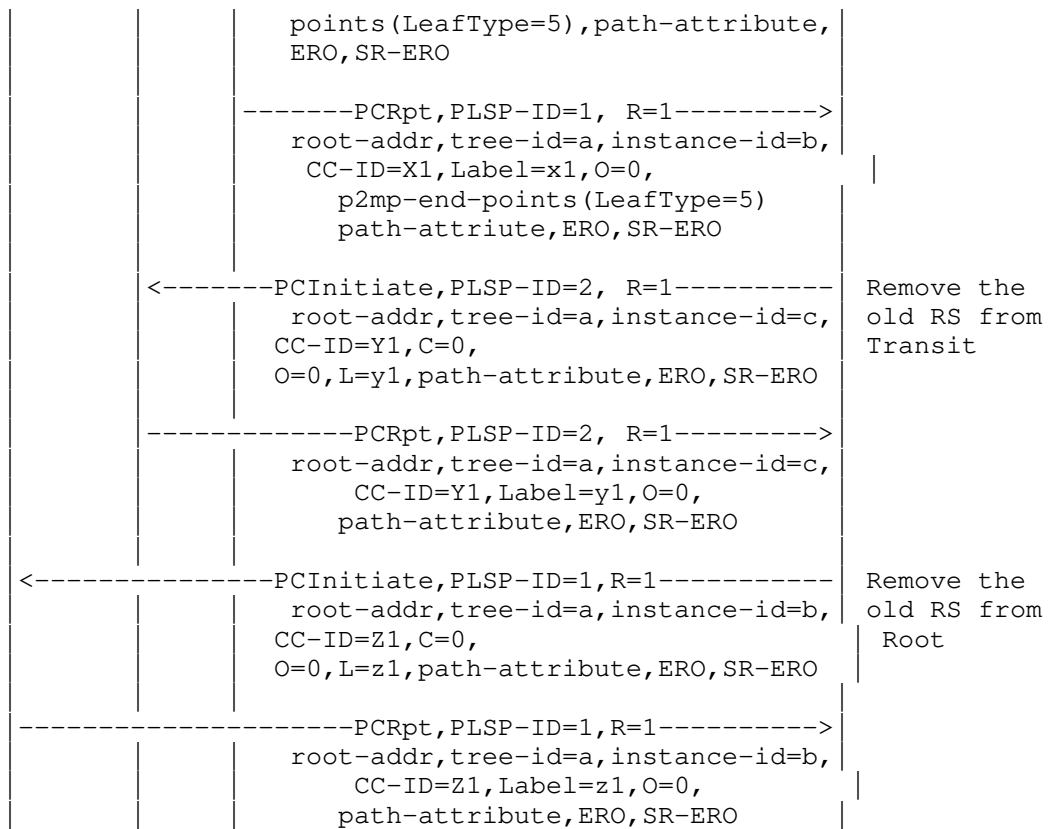


MBB Workflow:

Common (PCE-INIT, PCC-INIT) MBB







8. IANA Consideration

1. This draft extends the PCEP OPEN object by defining an optional TLV to indicate the PCE's capability to perform SR-P2MP path computations with a new IANA capability type (TBD).
2. PCEP open object with a new association type " P2MP SR Policy Association " value (TBD).
3. A new Association type. Association type = TBD1 "P2MP SR Policy Association Type" for SR Policy Association Group (P2MP SRPAG)
 1. three new TLVs are identified to carry association information: P2MP-SRPAG- POL-ID-TLV, P2MP-SRPAG-CPATH-ID-TLV, P2MP-SRPAG-CPATH-ATTR-TLV
4. Two new TLVs for Identifying the P2MP Policy and the Replication segment SR-IPV4-P2MP-POLICY-ID TLV and SR-IPV6-P2MP-POLICY-ID TLV

5. A new SR-P2MP-NODE-ROLE TLV (Type To be assigned by IANA) that will be present in the PATH-ATTRIB object

9. Security Considerations

TBD

10. Acknowledgments

The authors would like to thank Tanmoy Kundu and Stone Andrew at Nokia for their feedback and major contribution to this draft.

11. References

11.1. Normative References

[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>>.

11.2. Informative References

[draft-barth-pce-segment-routing-policy-cp]

.

[draft-dhs-spring-sr-p2mp-policy-yang]

.

[draft-ietf-pce-multipath]

.

[draft-ietf-pce-pcep-extension-for-pce-controller]

.

[draft-ietf-pce-segment-routing-policy-cp]

.

[draft-ietf-pce-stateful-pce-p2mp]

.

[draft-ietf-pim-sr-p2mp-policy]

"D. Yoyer, C. Filsfils, R.Prekh, H.bidgoli, Z. Zhang,
"draft-voyer-pim-sr-p2mp-policy"", October 2019.

[draft-ietf-spring-segment-routing-policy]

.

[draft-ietf-spring-sr-replication-segment]
"D. Yoyer, C. Filsfils, R.Prekh, H.bidgoli, Z. Zhang,
"draft-voyer-pim-sr-p2mp-policy "draft-voyer-spring-sr-
replication-segment"", July 2020.

[draft-parekh-bess-mvpn-sr-p2mp]

.

[draft-sivabalan-pce-binding-label-sid]

.

[RFC3209] .

[RFC5440] .

[RFC6513] .

[RFC8231] .

[RFC8236] .

[RFC8281] .

[RFC8306] .

[RFC8664] .

[RFC8697] .

Authors' Addresses

Hooman Bidgoli (editor)
Nokia
Ottawa
Canada

Email: hooman.bidgoli@nokia.com

Daniel Voyer
Bell Canada
Montreal
Canada

Email: daniel.yover@bell.ca

Saranya Rajarathinam
Nokia
Mountain View
US

Email: saranya.Rajarathinam@nokia.com

Ehsan Hemmati
Cisco System
San Jose
USA

Email: ehemmati@cisco.com

Tarek Saad
Juniper Networks
Ottawa
Canada

Email: tsaad@juniper.com

Siva Sivabalan
Ciena
Ottawa
Canada

Email: ssivabal@ciena.com

PCE
Internet-Draft
Intended status: Standards Track
Expires: January 8, 2020

Quan Xiong
Greg Mirsky
ZTE Corporation
Fangwei Hu
Individual
Weiqiang Cheng
China Mobile
July 7, 2019

Stitching LSP Association
draft-hu-pce-stitching-lsp-association-01

Abstract

The Path Computation Element Communication Protocol (PCEP) provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients (PCCs) requests. [I-D.ietf-pce-association-group] proposed an association mechanism for a set of LSPs.

This document defines the stitching LSP association type and stitching LSP association TLV for the inter-domain scenario.

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 <https://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 January 8, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
2.1. Terminology	3
2.2. Requirements Language	3
3. Stitching LSPs in SR-MPLS Inter-domain Scenario	3
4. PCEP Extension for Stitching LSP	3
4.1. Stitching LSP Association Type and Group	4
4.2. Stitching LSP Association TLV	4
5. Security Considerations	5
6. Acknowledgements	5
7. IANA Considerations	5
7.1. Association Object Type	5
8. Normative References	5
Authors' Addresses	6

1. Introduction

[RFC5440] describes the Path Computation Element Protocol (PCEP) which is used between a Path Computation Element (PCE) and a Path Computation Client (PCC) (or other PCE) to enable computation of Multi-protocol Label Switching (MPLS) for Traffic Engineering Label Switched Path (TE LSP). [I-D.ietf-pce-association-group] proposed an association mechanism to create a grouping of LSPs in the context of a PCE.

[I-D.xiong-pce-stateful-pce-sr-inter-domain] introduces the procedure and the PCEP extension to form the inter-domain MPLS data entries and the multiple LSPs from multiple contiguous domains need to be stitched to an end-to-end LSP in SR inter-domain scenario.

This document proposes a new association object type called "stitching Association LSP type" and TLV called "Stitching LSP Association TLV" to associate a grouping of LSPs from multiple domains for inter-domain scenario.

2. Conventions used in this document

2.1. Terminology

The terminology is defined as [RFC5440], [I-D.ietf-pce-association-group] and [I-D.xiong-pce-stateful-pce-sr-inter-domain].

2.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Stitching LSPs in SR-MPLS Inter-domain Scenario

As described in [I-D.xiong-pce-stateful-pce-sr-inter-domain], the domains of the networks may be IGP Areas in stitching inter-domain scenario. As Figure 1 shown, the multiple SR-MPLS domains may be interconnect with a ABR within areas. The multiple LSPs in each domain can be stitched to an end-to-end LSP. The LSP-1, LSP-2 and LSP-3 can be associated to a group.

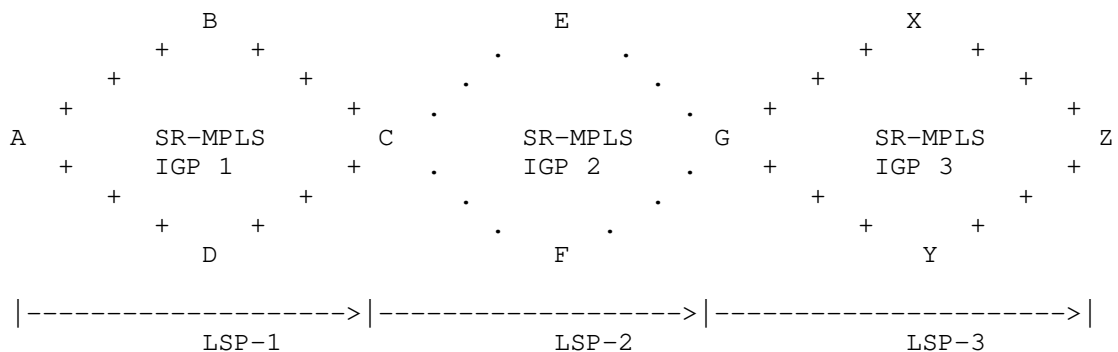


Figure 1: Stitching LSP in SR-MPLS Inter-domain Scenario

4. PCEP Extension for Stitching LSP

4.1. Stitching LSP Association Type and Group

An association ID will be used to identify the group and a new Association Type is defined in this document, based on the generic Association object :

Association Type (TBD) = Stitching LSP Association Group (SLAG).

SLAG may carry optional TLVs including but not limited to :

STITCHING-LSP-ASSOCIATION-TLV: Used to identify the role of stitching LSPs, described in Section 4.2.

As [I-D.ietf-pce-association-group] specified, the capability advertisement of the association types supported by a PCEP speaker is performed by defining a ASSOC-Type-List TLV to be carried within an OPEN object. The association type which defined in this document should be added in the list and be advertised between the PCEP speakers before the stitching LSP association.

Stitching LSP Association could be created dynamically or configured by the operator when operator-configured association is needed.

4.2. Stitching LSP Association TLV

The format of the Stitching LSP Association TLV is shown in Figure 1.

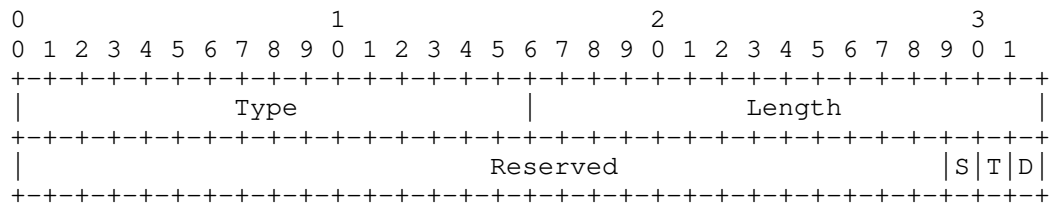


Figure 2: Stitching LSP Association TLV

The fields of the Stitching LSP Association TLV are following:

Type:16bits, it indicates the stitching LSP Association Group

TLV: TBD2, the value is assigned by IANA).

Length: the value is 4, it indicates the length of the TLV is 4 bytes.

Reserved: it is reserved for future use.

Stitching LSP Association Flags-S:1bit, indicates stitching LSP of the source domain when it is set.

Stitching LSP Association Flags-T:1bit, indicates stitching LSP of the transit domain when it is set.

Stitching LSP Association Flags-D:1bit, indicates stitching LSP of the destination layer when it is set.

5. Security Considerations

TBA

6. Acknowledgements

TBA

7. IANA Considerations

7.1. Association Object Type

This document defines a new association type and TLV in Association object which originally defined in [I-D.ietf-pce-association-group]. IANA is requested to make allocations from the registry, as follows:

Value	Name	Reference
TBD	Stitching LSP Association Type	[this document]
TBD	Stitching LSP Association TLV	[this document]

Table 1

8. Normative References

- [I-D.ietf-pce-association-group]
 Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Dhody, D., and Y. Tanaka, "PCEP Extensions for Establishing Relationships Between Sets of LSPs", draft-ietf-pce-association-group-09 (work in progress), April 2019.

- [I-D.xiong-pce-stateful-pce-sr-inter-domain]
Xiong, Q., hu, f., Mirsky, G., and W. Cheng, "Stateful PCE for SR-MPLS-TP Inter-domain", draft-xiong-pce-stateful-pce-sr-inter-domain-00 (work in progress), December 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>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [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

Quan Xiong
ZTE Corporation
No.6 Huashi Park Rd
Wuhan, Hubei 430223
China

Phone: +86 27 83531060
Email: xiong.quan@zte.com.cn

Greg Mirsky
ZTE Corporation
USA

Email: gregimirsky@gmail.com

Fangwei Hu
Individual
China

Email: hufwei@gmail.com

Weiqiang Cheng
China Mobile
Beijing
China

Email: chengweiqiang@chinamobile.com

PCE
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2020

Quan Xiong
Greg Mirsky
ZTE Corporation
Fangwei Hu
Individual
Weiqiang Cheng
China Mobile
October 22, 2019

Stitching LSP Association
draft-hu-pce-stitching-lsp-association-02

Abstract

The Path Computation Element Communication Protocol (PCEP) provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients (PCCs) requests. [I-D.ietf-pce-association-group] proposed an association mechanism for a set of LSPs.

This document defines the stitching LSP association type and stitching LSP association TLV for the inter-domain scenario.

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 <https://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, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
2.1. Terminology	3
2.2. Requirements Language	3
3. Stitching LSPs in SR-MPLS Inter-domain Scenario	3
4. PCEP Extension for Stitching LSP	3
4.1. I-flag in LSP Object	3
4.2. Stitching LSP Association Type and Group	4
4.3. Stitching LSP Association TLV	4
5. Security Considerations	5
6. Acknowledgements	5
7. IANA Considerations	5
7.1. New LSP Flag Registry	5
7.2. Association Object Type	6
8. Normative References	6
Authors' Addresses	7

1. Introduction

[RFC5440] describes the Path Computation Element Protocol (PCEP) which is used between a Path Computation Element (PCE) and a Path Computation Client (PCC) (or other PCE) to enable computation of Multi-protocol Label Switching (MPLS) for Traffic Engineering Label Switched Path (TE LSP). [I-D.ietf-pce-association-group] proposed an association mechanism to create a grouping of LSPs in the context of a PCE.

[I-D.xiong-pce-stateful-pce-sr-inter-domain] introduces the procedure and the PCEP extension to form the inter-domain MPLS data entries and the multiple LSPs from multiple contiguous domains need to be stitched to an end-to-end LSP in SR inter-domain scenario.

This document proposes a new association object type called "stitching Association LSP type" and TLV called "Stitching LSP Association TLV" to associate a grouping of LSPs from multiple domains for inter-domain scenario.

2. Conventions used in this document

2.1. Terminology

The terminology is defined as [RFC5440], [I-D.ietf-pce-association-group] and [I-D.xiong-pce-stateful-pce-sr-inter-domain].

2.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Stitching LSPs in SR-MPLS Inter-domain Scenario

As described in [I-D.xiong-pce-stateful-pce-sr-inter-domain], the domains of the networks may be IGP Areas in stitching inter-domain scenario. As Figure 1 shown, the multiple SR-MPLS domains may be interconnect with a ABR within areas. The multiple LSPs in each domain can be stitched to an inter-domain end-to-end LSP. The LSP-1, LSP-2 and LSP-3 can be associated to a group.

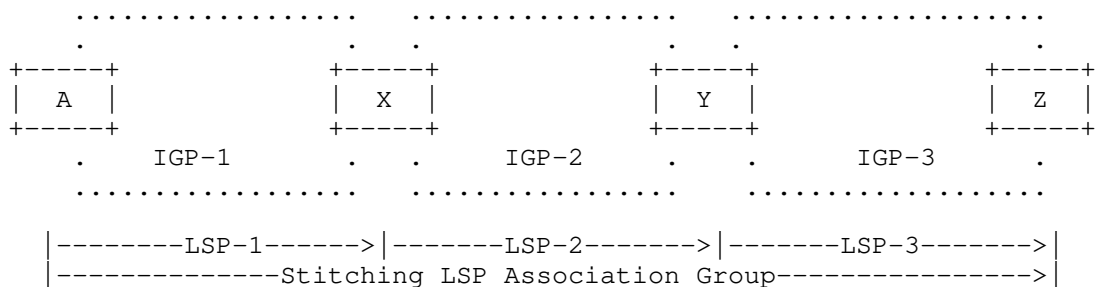


Figure 1: Stitching LSPs in SR-MPLS Inter-domain Scenario

4. PCEP Extension for Stitching LSP

4.1. I-flag in LSP Object

The LSP Object is defined in Section 7.3 of [RFC8231]. This document defiend a new flag (I-flag) for the LSP Object as Figure 2 shown:

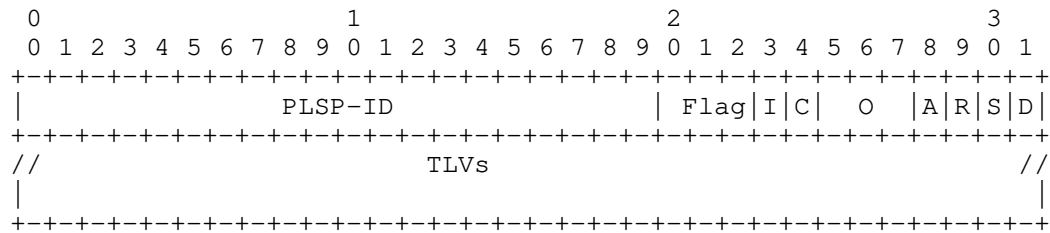


Figure 2: I-flag in LSP Object

I (Request for Inter-domain Path) : If the bit is set to 1, it indicates that the PCC requests PCE to compute the end-to-end path for inter-domain scenario carried in PCReq message. A parent PCE would set this bit to 1 to indicate that it is an end-to-end inter-domain path and a child PCE would set it to 1 to indicate that the path is part of an end-to-end inter-domain path. That may be encoded in the PCRep, PCUpd or PCInitiate message.

4.2. Stitching LSP Association Type and Group

An association ID will be used to identify the group and a new Association Type is defined in this document, based on the generic Association object :

Association Type (TBD) = Stitching LSP Association Group (SLAG).

SLAG may carry optional TLVs including but not limited to :

STITCHING-LSP-ASSOCIATION-TLV: Used to identify the role of stitching LSPs, described in Section 4.3.

As [I-D.ietf-pce-association-group] specified, the capability advertisement of the association types supported by a PCEP speaker is performed by defining a ASSOC-Type-List TLV to be carried within an OPEN object. The association type which defined in this document should be added in the list and be advertised between the PCEP speakers before the stitching LSP association.

Stitching LSP Association could be created dynamically or configured by the operator when operator-configured association is needed.

4.3. Stitching LSP Association TLV

The format of the Stitching LSP Association TLV is shown in Figure 3.

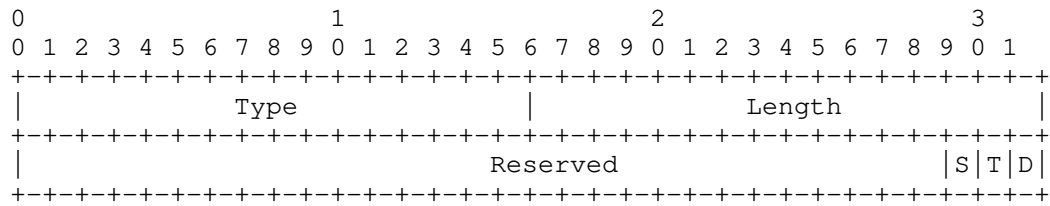


Figure 3: Stitching LSP Association TLV

The fields of the Stitching LSP Association TLV are following:

Type:16 bits, it indicates the stitching LSP Association Group

TLV: TBD2, the value is assigned by IANA).

Length: the value is 4, it indicates the length of the TLV is 4 bytes.

Reserved: it is reserved for future use.

Stitching LSP Association Flags-S:1bit, indicates stitching LSP of the source domain when it is set.

Stitching LSP Association Flags-T:1bit, indicates stitching LSP of the transit domain when it is set.

Stitching LSP Association Flags-D:1bit, indicates stitching LSP of the destination layer when it is set.

5. Security Considerations

TBA

6. Acknowledgements

TBA

7. IANA Considerations

7.1. New LSP Flag Registry

[RFC8231] defines the LSP object; per that RFC, IANA created a registry to manage the value of the LSP object's Flag field. IANA is requested to make allocations from the registry, as follows:

Value	Name	Reference
TBD	Request for Inter-domain Path (I)	[this document]

Table 1

7.2. Association Object Type

This document defines a new association type and TLV in Association object which originally defined in [I-D.ietf-pce-association-group]. IANA is requested to make allocations from the registry, as follows:

Value	Name	Reference
TBD	Stitching LSP Association Type	[this document]
TBD	Stitching LSP Association TLV	[this document]

Table 2

8. Normative References

- [I-D.ietf-pce-association-group]
Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Dhody, D., and Y. Tanaka, "Path Computation Element Communication Protocol (PCEP) Extensions for Establishing Relationships Between Sets of Label Switched Paths (LSPs)", draft-ietf-pce-association-group-10 (work in progress), August 2019.
- [I-D.xiong-pce-stateful-pce-sr-inter-domain]
Xiong, Q., hu, f., Mirsky, G., and W. Cheng, "Stateful PCE for SR-MPLS Inter-domain", draft-xiong-pce-stateful-pce-sr-inter-domain-01 (work in progress), July 2019.
- [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>>.
- [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>>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [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

Quan Xiong
ZTE Corporation
No.6 Huashi Park Rd
Wuhan, Hubei 430223
China

Phone: +86 27 83531060
Email: xiong.quan@zte.com.cn

Greg Mirsky
ZTE Corporation
USA

Email: gregimirsky@gmail.com

Fangwei Hu
Individual
China

Email: hufwei@gmail.com

Weiqiang Cheng
China Mobile
Beijing
China

Email: chengweiqiang@chinamobile.com

PCE working group
Internet-Draft
Updates: 5088,5089 (if approved)
Intended status: Standards Track
Expires: December 4, 2019

D. Lopez
Telefonica I+D
Q. Wu
D. Dhody
Z. Wang
Huawei
D. King
Old Dog Consulting
June 2, 2019

IGP extension for PCEP security capability support in the PCE discovery
draft-ietf-lsr-pce-discovery-security-support-01

Abstract

When a Path Computation Element (PCE) is a Label Switching Router (LSR) participating in the Interior Gateway Protocol (IGP), or even a server participating in IGP, its presence and path computation capabilities can be advertised using IGP flooding. The IGP extensions for PCE discovery (RFC 5088 and RFC 5089) define a method to advertise path computation capabilities using IGP flooding for OSPF and IS-IS respectively. However these specifications lack a method to advertise PCEP security (e.g., Transport Layer Security(TLS), TCP Authentication Option (TCP-AO)) support capability.

This document proposes new capability flag bits for PCE-CAP-FLAGS sub-TLV that can be announced as attribute in the IGP advertisement to distribute PCEP security support information. In addition, this document updates RFC 5088 and RFC 5089 to allow advertisement of Key ID or Key Chain Name Sub-TLV to support TCP AO security capability.

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 <https://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 December 4, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Introduction

As described in [RFC5440], PCEP communication privacy is one importance issue, as an attacker that intercepts a Path Computation Element (PCE) message could obtain sensitive information related to computed paths and resources.

Among the possible solutions mentioned in these documents, Transport Layer Security (TLS) [RFC8446] provides support for peer authentication, and message encryption and integrity while TCP Authentication Option (TCP-AO) [RFC5925] and Cryptographic Algorithms for TCP-AO [RFC5926] offer significantly improved security for applications using TCP. As specified in section 4 of [RFC8253], in order for a Path Computation Client (PCC) to begin a connection with a PCE server using TLS or TCP-AO, PCC needs to know whether PCE server supports TLS or TCP-AO as a secure transport.

[RFC5088] and [RFC5089] define a method to advertise path computation capabilities using IGP flooding for OSPF and IS-IS respectively. However these specifications lack a method to advertise PCEP security (e.g., TLS) support capability.

This document proposes new capability flag bits for PCE-CAP-FLAGS sub-TLV that can be announced as attributes in the IGP advertisement to distribute PCEP security support information. In addition, this document updates RFC5088 and RFC5089 to allow advertisement of Key ID or Key Chain Name Sub-TLV to support TCP AO security capability.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. IGP extension for PCEP security capability support

[RFC5088] defines a PCE Discovery (PCED) TLV carried in an OSPF Router Information Link State Advertisement (LSA) as defined in [RFC7770] to facilitate PCE discovery using OSPF. This document defines two new capability flag bits in the OSPF PCE Capability Flags to indicate TCP Authentication Option (TCP-AO) support [RFC5925][RFC5926], PCEP over TLS support [RFC8253] respectively.

Similarly, [RFC5089] defines the PCED sub-TLV for use in PCE discovery using IS-IS. This document will use the same flag for the OSPF PCE Capability Flags sub-TLV to allow IS-IS to indicate TCP Authentication Option (TCP-AO) support, PCEP over TLS support respectively.

The IANA assignments for shared OSPF and IS-IS Security Capability Flags are documented in Section 8.1 ("OSPF PCE Capability Flag") of this document.

3.1. Use of PCEP security capability support for PCE discovery

TCP-AO, PCEP over TLS support flag bits are advertised using IGP flooding.

- o PCE supports TCP-AO: IGP advertisement SHOULD include TCP-AO support flag bit.
- o PCE supports TLS: IGP advertisement SHOULD include PCEP over TLS support flag bit.

If PCE supports multiple security mechanisms, it SHOULD include all corresponding flag bits in IGP advertisement.

If the client is looking for connecting with PCE server with TCP-AO support, the client MUST check if TCP-AO support flag bit in the PCE-CAP-FLAGS sub-TLV is set. If not, the client SHOULD NOT consider this PCE. If the client is looking for connecting with PCE server using TLS, the client MUST check if PCEP over TLS support flag bit in the PCE-CAP-FLAGS sub-TLV is set. If not, the client SHOULD NOT

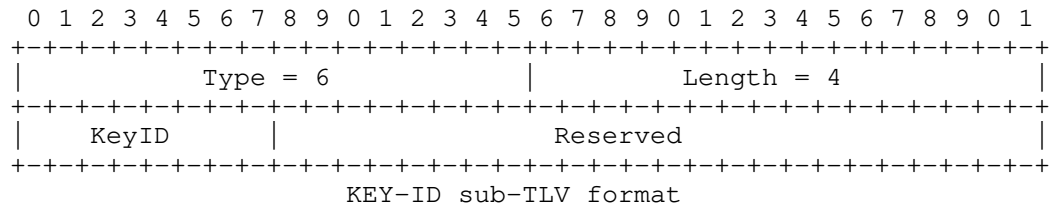
consider this PCE. Note that this can be overridden based on a local policy at the PCC.

3.2. KEY-ID Sub-TLV

The KEY-ID sub-TLV specifies a key that can be used by the PCC to identify the TCP-AO key [RFC5925].

The KEY-ID sub-TLV MAY be present in the PCED sub-TLV carried within the IS-IS Router Information Capability TLV when the capability flag bit of PCE-CAP-FLAGS sub-TLV in IS-IS is set to indicate TCP Authentication Option (TCP-AO) support. Similarly, this sub-TLV MAY be present in the PCED TLV carried within OSPF Router Information LSA when the capability flag bit of PCE-CAP-FLAGS sub-TLV in OSPF is set to indicate TCP-AO support.

The format of the KEY-ID sub-TLV is as follows:



Type: 6

Length: 4

KeyID: The one octet Key ID as per [RFC5925] to uniquely identify the Master Key Tuple (MKT).

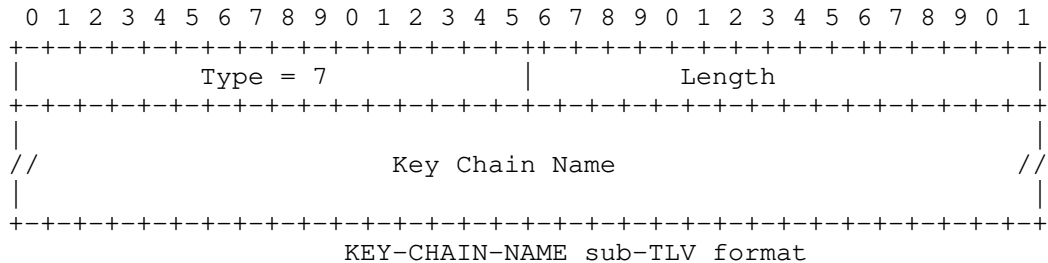
Reserved: MUST be set to zero while sending and ignored on receipt.

3.3. KEY-CHAIN-NAME Sub-TLV

The KEY-CHAIN-NAME sub-TLV specifies a keychain name that can be used by the PCC to identify the keychain [RFC8177].

The KEY-CHAIN-NAME sub-TLV MAY be present in the PCED sub-TLV carried within the IS-IS Router Information Capability TLV when the capability flag bit of PCE-CAP-FLAGS sub-TLV in IS-IS is set to indicate TCP Authentication Option (TCP-AO) support. Similarly, this sub-TLV MAY be present in the PCED TLV carried within OSPF Router Information LSA when the capability flag bit of PCE-CAP-FLAGS sub-TLV in OSPF is set to indicate TCP-AO support.

The format of the KEY-CHAIN-NAME sub-TLV is as follows:



Type: 7

Length: Variable

Key Name: The Key Chain Name contains a string to be used to identify the key chain. It SHOULD be a string of printable ASCII characters, without a NULL terminator. The TLV MUST be zero-padded so that the TLV is 4-octet aligned.

4. Update to RFC5088 and RFC5089

Section 4 of [RFC5088] needs to be updated to allow advertisement of additional PCE information carried in the Router Information LSA. The following is proposed text for this change.

Replace the following paragraph from section 4:

"No additional sub-TLVs will be added to the PCED TLV in the future. If a future application requires the advertisement of additional PCE information in OSPF/ISIS, this will not be carried in the Router Information LSA."

with

"If a future application requires the advertisement of additional PCE information in OSPF, e.g., to facilitate key distribution and cryptographic authentication and message integrity verification, additional sub-TLVs could be added to the PCED TLV and carried in the Router Information LSA."

Section 4 of [RFC5089] needs to be updated to allow advertisement of additional PCE information carried in the Router CAPABILITY TLV. The following is proposed text for this change.

Replace the following paragraph from section 4:

"No additional sub-TLVs will be added to the PCED TLV in the future. If a future application requires the advertisement of additional PCE information in IS-IS, this will not be carried in the CAPABILITY TLV."

with

"If a future application requires the advertisement of additional PCE information in IS-IS, e.g., to facilitate key distribution and cryptographic authentication and message integrity verification, additional sub-TLVs could be added to the PCED sub-TLV and carried in the CAPABILITY TLV."

At a time of publication of [RFC5088] and [RFC5089] there were concerns about advertising non-IGP specific information in OSPF(v3) Router Information LSAs and IS-IS router capability TLV. [RFC7770] added the functionality of advertising multiple instances of the OSPF(v3) Router Information LSA and IS-IS support multiple CAPABILITY TLV [RFC7981].

5. Backward Compatibility Consideration

An LSR that does not support the new IGP PCE capability bits specified in this document silently ignores those bits.

An LSR that does not support the new KEYNAME sub-TLV specified in this document silently ignores the sub-TLV.

IGP extensions defined in this document do not introduce any new interoperability issues.

6. Management Considerations

A configuration option may be provided for advertising and withdrawing PCE security capability via IGP.

7. Security Considerations

This document raises no new security issues beyond those described in [RFC5088] and [RFC5089].

8. IANA Considerations

8.1. OSPF PCE Capability Flag

IANA is requested to allocate new bits assignments for the OSPF Parameters "Path Computation Element (PCE) Capability Flags" registry.

Bit	Meaning	Reference
xx	TCP-AO Support	[This.I.D]
xx	PCEP over TLS support	[This.I.D]

The registry is located at: <https://www.iana.org/assignments/ospfv2-parameters/ospfv2-parameters.xml#ospfv2-parameters-14.xml>

8.2. PCED sub-TLV Type Indicators

The PCED sub-TLVs were defined in [RFC5088] and [RFC5089], but they did not create a registry for it. This document requests IANA to create a new top-level OSPF registry, the "PCED sub-TLV type indicators" registry. This registry should be populated with -

Value	Description	Reference
0	Reserved	[This.I.D] [RFC5088]
1	PCE-ADDRESS	[This.I.D] [RFC5088]
2	PATH-SCOPE	[This.I.D] [RFC5088]
3	PCE-DOMAIN	[This.I.D] [RFC5088]
4	NEIG-PCE-DOMAIN	[This.I.D] [RFC5088]
6	KEY-ID	[This.I.D]
7	KEY-CHAIN-NAME	[This.I.D]

This registry is also used by IS-IS PCED sub-TLV.

9. Acknowledgments

The authors of this document would also like to thank Acee Lindem, Julien Meuric for the review and comments.

10. References

10.1. Normative References

- [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>>.
- [RFC5088] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, DOI 10.17487/RFC5088, January 2008, <<https://www.rfc-editor.org/info/rfc5088>>.
- [RFC5089] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, DOI 10.17487/RFC5089, January 2008, <<https://www.rfc-editor.org/info/rfc5089>>.

- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <<https://www.rfc-editor.org/info/rfc5925>>.
- [RFC5926] Lebovitz, G. and E. Rescorla, "Cryptographic Algorithms for the TCP Authentication Option (TCP-AO)", RFC 5926, DOI 10.17487/RFC5926, June 2010, <<https://www.rfc-editor.org/info/rfc5926>>.
- [RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <<https://www.rfc-editor.org/info/rfc7770>>.
- [RFC7981] Ginsberg, L., Previdi, S., and M. Chen, "IS-IS Extensions for Advertising Router Information", RFC 7981, DOI 10.17487/RFC7981, October 2016, <<https://www.rfc-editor.org/info/rfc7981>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <<https://www.rfc-editor.org/info/rfc8177>>.
- [RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <<https://www.rfc-editor.org/info/rfc8253>>.

10.2. Informative References

- [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>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

Appendix A. No MD5 Capability Support

To be compliant with Section 10.2 of RFC5440, this document doesn't consider to add capability for TCP-MD5. Therefore by default, PCEP Speaker in communication supports capability for TCP-MD5 (See section 10.2, [RFC5440]). A method to advertise TCP-MD5 Capability support using IGP flooding is not required. If the client is looking for connecting with PCE server with other Security capability support (e.g., TLS support) than TCP-MD5, the client MUST check if flag bit in the PCE- CAP-FLAGS sub-TLV for specific capability is set (See section 3.1).

Authors' Addresses

Diego R. Lopez
Telefonica I+D
Spain

Email: diego.r.lopez@telefonica.com

Qin Wu
Huawei Technologies
12 Mozhou East Road, Jiangning District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560037
India

Email: dhruv.ietf@gmail.com

Michael Wang
Huawei
12 Mozhou East Road, Jiangning District
Nanjing, Jiangsu 210012
China

Email: wangzitao@huawei.com

Daniel King
Old Dog Consulting
UK

Email: daniel@olddog.co.uk

PCE working group
Internet-Draft
Updates: 5088, 5089, 8231, 8306, 8623 (if
approved)
Intended status: Standards Track
Expires: 22 February 2022

D. Lopez
Telefonica I+D
Q. Wu
D. Dhody
Q. Ma
Huawei
D. King
Old Dog Consulting
21 August 2021

IGP extension for PCEP security capability support in the PCE discovery
draft-ietf-lsr-pce-discovery-security-support-09

Abstract

When a Path Computation Element (PCE) is a Label Switching Router (LSR) participating in the Interior Gateway Protocol (IGP), or even a server participating in IGP, its presence and path computation capabilities can be advertised using IGP flooding. The IGP extensions for PCE discovery (RFC 5088 and RFC 5089) define a method to advertise path computation capabilities using IGP flooding for OSPF and IS-IS respectively. However these specifications lack a method to advertise PCEP security (e.g., Transport Layer Security (TLS), TCP Authentication Option (TCP-AO)) support capability.

This document defines capability flag bits for PCE-CAP-FLAGS sub-TLV that can be announced as an attribute in the IGP advertisement to distribute PCEP security support information. In addition, this document updates RFC 5088 and RFC 5089 to allow advertisement of Key ID or Key Chain Name Sub-TLV to support TCP-AO security capability. RFC 8231, RFC 8306, and RFC 8623 are also updated to reflect the movement of the IANA "PCE Capability Flags" registry.

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 <https://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 22 February 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions used in this document	3
3. IGP extension for PCEP security capability support	3
3.1. Use of PCEP security capability support for PCE discovery	4
3.2. KEY-ID Sub-TLV	4
3.3. KEY-CHAIN-NAME Sub-TLV	5
4. Update to RFC5088 and RFC5089	5
5. Backward Compatibility Consideration	6
6. Management Considerations	6
7. Security Considerations	6
8. IANA Considerations	7
8.1. PCE Capability Flag	7
8.2. PCED sub-TLV Type Indicators	7
9. Acknowledgments	7
10. References	8
10.1. Normative References	8
10.2. Informative References	9
Appendix A. No MD5 Capability Support	10
Authors' Addresses	10

1. Introduction

As described in [RFC5440], PCEP communication privacy is one importance issue, as an attacker that intercepts a Path Computation Element (PCE) message could obtain sensitive information related to computed paths and resources.

Among the possible solutions mentioned in these documents, Transport Layer Security (TLS) [RFC8446] provides support for peer authentication, and message encryption and integrity while TCP Authentication Option (TCP-AO) [RFC5925] and Cryptographic Algorithms for TCP-AO [RFC5926] offer significantly improved security for applications using TCP. As specified in section 4 of [RFC8253], in order for a Path Computation Client (PCC) to establish a connection with a PCE server using TLS or TCP-AO, PCC needs to know whether PCE server supports TLS or TCP-AO as a secure transport.

[RFC5088] and [RFC5089] define a method to advertise path computation capabilities using IGP flooding for OSPF and IS-IS respectively. However these specifications lack a method to advertise PCEP security (e.g., TLS) support capability.

This document defines capability flag bits for PCE-CAP-FLAGS sub-TLV that can be announced as attributes in the IGP advertisement to distribute PCEP security support information. In addition, this document updates RFC5088 and RFC5089 to allow advertisement of Key ID or Key Chain Name Sub-TLV to support TCP-AO security capability.

Note that the PCEP Open message exchange is another way to discover PCE capabilities information, but in this instance, the TCP security related key parameters need to be known before the PCEP session is established and the PCEP Open messages are exchanged. Thus, the use of the PCE discovery and capabilities advertisement of the IGP needs to be leveraged.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. IGP extension for PCEP security capability support

[RFC5088] defines a PCE Discovery (PCED) TLV carried in an OSPF Router Information Link State Advertisement (LSA) as defined in [RFC7770] to facilitate PCE discovery using OSPF. This document defines two capability flag bits in the OSPF PCE Capability Flags to indicate TCP Authentication Option (TCP-AO) support [RFC5925][RFC5926] and PCEP over TLS support [RFC8253] respectively.

Similarly, [RFC5089] defines the PCED sub-TLV for use in PCE discovery using IS-IS. This document will use the same flag for the OSPF PCE Capability Flags sub-TLV to allow IS-IS to indicate TCP Authentication Option (TCP-AO) support, PCEP over TLS support respectively.

The IANA assignments for shared OSPF and IS-IS Security Capability Flags are documented in Section 8.1 ("OSPF PCE Capability Flags") of this document.

3.1. Use of PCEP security capability support for PCE discovery

TCP-AO, PCEP over TLS support flag bits are advertised using IGP flooding.

- * PCE supports TCP-AO: IGP advertisement SHOULD include TCP-AO support flag bit.
- * PCE supports TLS: IGP advertisement SHOULD include PCEP over TLS support flag bit.

If PCE supports multiple security mechanisms, it SHOULD include all corresponding flag bits in IGP advertisement.

If the client is restricted to a PCE server with TCP-AO support, the client MUST check if TCP-AO support flag bit in the PCE- CAP-FLAGS sub-TLV is set. If not, the client SHOULD NOT consider this PCE. If the client is restricted to a PCE server using TLS, the client MUST check if PCEP over TLS support flag bit in the PCE-CAP-FLAGS sub-TLV is set. If not, the client SHOULD NOT consider this PCE. Note that this can be overridden based on a local policy at the PCC.

3.2. KEY-ID Sub-TLV

The KEY-ID sub-TLV specifies a key that can be used by the PCC to identify the TCP-AO key [RFC5925].

The KEY-ID sub-TLV MAY be present in the PCED sub-TLV carried within the IS-IS Router Information Capability TLV when the capability flag bit of PCE-CAP-FLAGS sub-TLV in IS-IS is set to indicate TCP Authentication Option (TCP-AO) support. Similarly, this sub-TLV MAY be present in the PCED TLV carried within OSPF Router Information LSA when the capability flag bit of PCE-CAP-FLAGS sub-TLV in OSPF is set to indicate TCP-AO support.

The KEY-ID sub-TLV has the following format:

Type: 6

Length: 4

KeyID: The one octet Key ID as per [RFC5925] to uniquely identify the Master Key Tuple (MKT).

Reserved: MUST be set to zero while sending and ignored on receipt.

3.3. KEY-CHAIN-NAME Sub-TLV

The KEY-CHAIN-NAME sub-TLV specifies a keychain name that can be used by the PCC to identify the keychain [RFC8177].

The KEY-CHAIN-NAME sub-TLV MAY be present in the PCED sub-TLV carried within the IS-IS Router Information Capability TLV when the capability flag bit of PCE-CAP-FLAGS sub-TLV in IS-IS is set to indicate TCP Authentication Option (TCP-AO) support. Similarly, this sub-TLV MAY be present in the PCED TLV carried within OSPF Router Information LSA when the capability flag bit of PCE-CAP-FLAGS sub-TLV in OSPF is set to indicate TCP-AO support.

The KEY-CHAIN-NAME sub-TLV has the following format:

Type: 7

Length: Variable

Key Name: The Key Chain Name contains a string to be used to identify the key chain. It SHOULD be a string of printable ASCII characters, without a NULL terminator. The sub-TLV MUST be zero-padded so that the sub-TLV is 4-octet aligned.

4. Update to RFC5088 and RFC5089

Section 4 of [RFC5088] states that no new sub-TLVs will be added to the PCED TLV, and no new PCE information will be carried in the Router Information LSA. This document updates [RFC5088] by allowing the two sub-TLVs defined in this document to be carried in the PCED TLV advertised in the Router Information LSA.

Section 4 of [RFC5089] states that no new sub-TLVs will be added to the PCED TLV, and no new PCE information will be carried in the Router CAPABILITY TLV. This document updates [RFC5089] by allowing the two sub-TLVs defined in this document to be carried in the PCED TLV advertised in the Router CAPABILITY TLV.

The introduction of the additional sub-TLVs should be viewed as an exception to the [RFC5088][RFC5089] policy justified by the requirements to discover the PCEP security support prior to establishing a PCEP session. The restrictions defined in [RFC5089][RFC5089] should still be considered to be in place.

The registry for the PCE Capability Flags assigned in section 8.2 of [RFC8231], section 6.9 of [RFC8306], and section 11.1 of [RFC8623] has changed to the IGP Parameters "Path Computation Element (PCE) Capability Flags" registry created in this document.

5. Backward Compatibility Consideration

An LSR that does not support the IGP PCE capability bits specified in this document silently ignores those bits.

An LSR that does not support the KEYNAME sub-TLV specified in this document silently ignores the sub-TLV.

IGP extensions defined in this document do not introduce any new interoperability issues.

6. Management Considerations

A configuration option may be provided for advertising and withdrawing PCEP security capability via OSPF and IS-IS.

7. Security Considerations

Security considerations as specified by [RFC5088] and [RFC5089] are applicable to this document.

The information related to PCEP security is sensitive and due care needs to be taken by the operator. This document defines new capability bits that are susceptible to a downgrade attack by toggling them. The content of Key ID or Key Chain Name Sub-TLV can be tweaked to enable a man-in-the-middle attack. Thus before advertising the PCEP security parameters, using the mechanism described in this document, the IGP MUST be known to provide authentication and integrity for the PCED TLV using the mechanisms defined in [RFC5304], [RFC5310] or [RFC5709].

Moreover, as stated in [RFC5088] and [RFC5089], if the IGP does not provide any encryption mechanisms to protect the secrecy of the PCED TLV, then the operator must ensure that no private data is carried in the TLV, e.g. that key-ids or key-chain names do not reveal sensitive information about the network.

8. IANA Considerations

8.1. PCE Capability Flag

IANA is requested to move the "PCE Capability Flags" registry from "Open Shortest Path First v2 (OSPFv2) Parameters" to under the IANA Common IGP parameters registry and allocate new bits assignments for the IGP Parameters "Path Computation Element (PCE) Capability Flags" registry.

Bit	Meaning	Reference
xx	TCP-AO Support	[This.I.D]
xx	PCEP over TLS support	[This.I.D]

The registry is located at: <https://www.iana.org/assignments/igp-parameters/igp-parameters.xhtml>

8.2. PCED sub-TLV Type Indicators

The PCED sub-TLVs were defined in [RFC5088] and [RFC5089], but they did not create a registry for it. This document requests IANA to create a new subregistry called "PCED sub-TLV type indicators" under the "Interior Gateway Protocol (IGP) Parameters" registry. The registration policy for this subregistry is "IETF Review" [RFC8126]. Values in this subregistry come from the range 0-65535.

This subregistry should be populated with:

Value	Description	Reference
0	Reserved	[This.I.D] [RFC5088]
1	PCE-ADDRESS	[This.I.D] [RFC5088]
2	PATH-SCOPE	[This.I.D] [RFC5088]
3	PCE-DOMAIN	[This.I.D] [RFC5088]
5	PCE-CAP-FLAGS	[This.I.D] [RFC5088]
4	NEIG-PCE-DOMAIN	[This.I.D] [RFC5088]
6	KEY-ID	[This.I.D]
7	KEY-CHAIN-NAME	[This.I.D]

This registry is located at: <https://www.iana.org/assignments/igp-parameters/igp-parameters.xhtml> and used by both OSPF PCED TLV and IS-IS PCED sub-TLV.

9. Acknowledgments

The authors of this document would also like to thank Acee Lindem, Julien Meuric, Les Ginsberg, Ketan Talaulikar, Yaron Sheffer, Tom Petch, Aijun Wang, Adrian Farrel for the review and comments.

The authors would also like to speical thank Michale Wang for his major contributions to the initial version.

10. References

10.1. Normative References

- [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>>.
- [RFC5088] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, DOI 10.17487/RFC5088, January 2008, <<https://www.rfc-editor.org/info/rfc5088>>.
- [RFC5089] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, DOI 10.17487/RFC5089, January 2008, <<https://www.rfc-editor.org/info/rfc5089>>.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <<https://www.rfc-editor.org/info/rfc5925>>.
- [RFC5926] Lebovitz, G. and E. Rescorla, "Cryptographic Algorithms for the TCP Authentication Option (TCP-AO)", RFC 5926, DOI 10.17487/RFC5926, June 2010, <<https://www.rfc-editor.org/info/rfc5926>>.
- [RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <<https://www.rfc-editor.org/info/rfc8253>>.
- [RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <<https://www.rfc-editor.org/info/rfc8177>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [RFC7770] Lindem, A., Ed., Shen, N., Vasseur, JP., Aggarwal, R., and S. Shaffer, "Extensions to OSPF for Advertising Optional Router Capabilities", RFC 7770, DOI 10.17487/RFC7770, February 2016, <<https://www.rfc-editor.org/info/rfc7770>>.
- [RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <<https://www.rfc-editor.org/info/rfc5304>>.
- [RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, DOI 10.17487/RFC5310, February 2009, <<https://www.rfc-editor.org/info/rfc5310>>.
- [RFC5709] Bhatia, M., Manral, V., Fanto, M., White, R., Barnes, M., Li, T., and R. Atkinson, "OSPFv2 HMAC-SHA Cryptographic Authentication", RFC 5709, DOI 10.17487/RFC5709, October 2009, <<https://www.rfc-editor.org/info/rfc5709>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [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>>.
- [RFC8306] Zhao, Q., Dhody, D., Ed., Palleti, R., and D. King, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", RFC 8306, DOI 10.17487/RFC8306, November 2017, <<https://www.rfc-editor.org/info/rfc8306>>.
- [RFC8623] Palle, U., Dhody, D., Tanaka, Y., and V. Beeram, "Stateful Path Computation Element (PCE) Protocol Extensions for Usage with Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 8623, DOI 10.17487/RFC8623, June 2019, <<https://www.rfc-editor.org/info/rfc8623>>.

10.2. Informative References

- [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>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

Appendix A. No MD5 Capability Support

To be compliant with Section 10.2 of RFC5440, this document doesn't consider adding capability for TCP-MD5. Therefore by default, a PCEP Speaker supports the capability for TCP-MD5 (See section 10.2, [RFC5440]). A method to advertise TCP-MD5 Capability support using IGP flooding is not required. If the client is looking for a PCE server with other Security capability support (e.g., TLS support) than TCP-MD5, the client MUST check if the corresponding flag bit in the PCE-CAP-FLAGS sub-TLV is set (See section 3.1). Irrespective of which security capability (e.g., TCP-MD5) is selected, the same key-ids or key-chain names on the PCC and PCE server should be configured.

Authors' Addresses

Diego R. Lopez
Telefonica I+D
Spain

Email: diego.r.lopez@telefonica.com

Qin Wu
Huawei Technologies
101 Software Avenue, Yuhua District
Nanjing
Jiangsu, 210012
China

Email: bill.wu@huawei.com

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore 560037
Karnataka
India

Email: dhruv.ietf@gmail.com

Qiufang Ma
Huawei
101 Software Avenue, Yuhua District
Nanjing
Jiangsu, 210012
China

Email: maqiufang1@huawei.com

Daniel King
Old Dog Consulting
United Kingdom

Email: daniel@olddog.co.uk

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2020

M. Koldychev
S. Sivabalan
Cisco Systems, Inc.
M. Negi
Huawei Technologies
D. Achaval
Nokia
H. Kotni
Juniper Networks, Inc
July 02, 2019

PCEP Operational Clarification
draft-koldychev-pce-operational-00

Abstract

This document is meant to provide better clarity about how PCEP operates and hence to facilitate better interoperability between different equipment vendors. The content of this document has been compiled based on the feedback from several multi-vendor interop exercises. Several constructs are introduced to facilitate this, such as the LSP-DB and the ASSO-DB.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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 <https://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 January 3, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. PCEP LSP Database	4
3.1. Structure	4
3.2. Synchronization	5
3.3. Stateful Bringup	5
3.4. Successful MBB	7
3.5. Aborted MBB	8
4. PCEP Association Database	9
4.1. 2 LSPs in same Association	9
4.2. Switch Association during MBB	10
5. Acknowledgement	11
6. References	11
6.1. Normative References	11
6.2. Informative References	12
Appendix A. Contributors	12
Authors' Addresses	13

1. Introduction

Path Computation Element (PCE) Communication Protocol (PCEP) [RFC5440] enables the communication between a Path Computation Client (PCC) and a Path Control Element (PCE), or between two PCEs based on the PCE architecture [RFC4655].

PCEP Extensions for the Stateful PCE Model [RFC8231] describes a set of extensions to PCEP to enable active control of Multiprotocol Label Switching Traffic Engineering (MPLS-TE) and Generalized MPLS (GMPLS) tunnels. [RFC8281] describes the setup and teardown of PCE-initiated LSPs under the active stateful PCE model, without the need for local

configuration on the PCC, thus allowing for dynamic centralized control of a network.

PCEP Extensions for Establishing Relationships Between Sets of LSPs [I-D.ietf-pce-association-group] introduces a generic mechanism to create a grouping of LSPs which can then be used to define associations between a set of LSPs and a set of attributes (such as configuration parameters or behaviors) and is equally applicable to stateful PCE (active and passive modes) and stateless PCE.

The PCEP protocol has evolved from a simple stateless model into a stateful model with more features being added. Due to subtle differences in interpretation of existing PCEP standards, it was found that networking equipment vendors often had to adjust their implementations, in order to interoperate. This informational document is meant to clarify these subtle differences and agree on a final model that all major vendors have agreed on and that all other vendors can adopt. This document applies to RSVP-TE and Segment-Routing.

2. Terminology

The following terminologies are used in this document:

PCC: Path Computation Client. Any client application requesting a path computation to be performed by a Path Computation Element.

PCE: Path Computation Element. An entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints.

PCEP: Path Computation Element Protocol.

MBB: Make-Before-Break. A procedure during which the head-end of a traffic-engineered path wishes to move traffic to a new path without losing any traffic, by first "making" a new path and then "breaking" the old path.

Association parameters: As described in [I-D.ietf-pce-association-group], the combination of the mandatory fields Association type, Association ID and Association Source in the ASSOCIATION object uniquely identify the association group. If the optional TLVs - Global Association Source or Extended Association ID are included, then they MUST be included in combination with mandatory fields to uniquely identify the association group.

Association information: As described in [I-D.ietf-pce-association-group], the ASSOCIATION object could also include other optional TLVs based on the association types, that provides 'information' related to the association type.

ERO: Explicit Route Object is the path of the LSP encoded into a PCEP object. To represent an empty ERO object, i.e., without any subobjects, we use the notation "ERO={}". To represent an ERO object containing some given sequence of subobjects, we use the notation "ERO={A}".

3. PCEP LSP Database

We introduce the concept of the LSP-DB, as a database of actual LSP state in the network. This concept is not explicitly defined in [RFC8231], but is fully compatible with it. We use the LSP-DB to describe how certain actions are performed, because it is easier to define actions as a function of database state, rather than as a function of previously received messages. The structure and format of the LSP-DB MUST be common among all dataplane types (i.e., RSVP-TE/SR-TE/SRv6), all instantiation methods (i.e., PCC-initiated/PCE-initiated), all destination types (i.e., point-to-point/point-to-multipoint).

Note that we use the term "Tunnel" somewhat loosely here, to mean "the object identified by the PLSP-ID". It may or may not be an actual tunnel in the implementation. For example, working and protect paths can be implemented as one tunnel interface, but in PCEP we would refer to them as two different Tunnels, because they would have different PLSP-IDs.

Note that the term "LSP", which stands for "Label Switched Path", if taken too literally would restrict our discussion to MPLS dataplane only. In this document, we allow the term "LSP" to refer to any path, regardless of the dataplane format. So that an LSP can refer to MPLS and SRv6 dataplane paths.

3.1. Structure

[RFC8231] states that the LSP-IDENTIFIERS TLV contains the key that MUST be used to differentiate different LSPs during make before break procedure. We further clarify here that PCEP LSPs exist in a 2-tier structure. The top tier is the "Tunnel", identified by the PLSP-ID and/or SYMBOLIC-NAME, while the lower tier is the "LSP", identified by the values in LSP-IDENTIFIERS TLV. A single Tunnel may contain multiple LSPs at the same time, i.e., a Tunnel is a container for LSPs. A Tunnel MUST have at least one LSP and when the last LSP is removed from the Tunnel, the Tunnel itself is removed.

3.2. Synchronization

The stateful PCE MUST maintain the PCE LSP-DB, which stores Tunnels and LSPs. The PCE LSP DB is only modified by PCRpt messages. No other PCEP message may modify the PCE LSP DB. The PCC MUST also maintain the PCC LSP DB, which it MUST synchronize with the PCE LSP DB by sending PCRpt messages.

The PCC adds/removes entries to/from its LSP-DB based on what LSPs it creates/destroys in the network. There can be many trigger types for updating the PCC LSP-DB, some examples include PCUpd messages, local computation on the PCC, local configuration on the PCC, etc. The trigger type does not affect the content of the PCC LSP-DB, i.e., the content of the PCC LSP-DB is updated identically regardless of the trigger type.

Whenever a PCC modifies an entry in its PCC LSP-DB, it MUST send a PCRpt message to the PCE (or multiple PCEs), to synchronize this change. Ensuring this synchronization is always in place allows one to define behavior as a function of LSP-DB state, instead of defining behavior as a function of what PCEP messages were sent or received.

The PCE MUST always act on the latest state of the PCE LSP DB. Note that this does not mean that the PCE cannot use information from outside of LSP-DB. For example, the PCE can use other mechanisms to collect traffic statistics and use them in the computation. However, these traffic statistics are not part of the LSP-DB, but only reference it.

The LSP-DB on both the PCC and the PCE only stores the actual state in the network, it does not store the desired state. For example, consider the case of PCE Initiated LSP, configured on the PCE. When the operator modifies the configuration of this LSP, that is a change in desired state. The actual state has not yet changed, so LSP-DB is not modified yet. The LSP-DB is only modified after the PCE sends PCInit/PCUpd message to the PCC and the PCC decides to act on that message. When the PCC acts on message, it would update its own PCC LSP DB and immediately send PCRpt to the PCE to synchronize the change. When the PCE receives the PCRpt msg, it updates its own PCE LSP DB. After this, the PCC LSP DB and PCE LSP DB are in sync.

3.3. Stateful Bringup

[RFC8231] in section 5.8.2, allows delegation of an LSP in operationally down state, but at the same time mandates the use of PCReq, before sending PCRpt. In this document, we would like to make it clear that sending PCReq is optional.

We shall refer to the process of sending PCReq before PCRpt as "stateless bringup". In reality, stateless bringup introduces overhead and is not possible to enforce from the PCE, because the stateless PCE is not supposed to keep any per-LSP state about previous PCReq messages. It was found that many vendors choose to ignore this requirement and send the PCRpt directly, without going through PCReq. This section will serve to explain and to validate this behavior.

Even though all the major vendors today are moving to the stateful PCE model, it does not deprecate the need for stateless PCEP. The key property of stateless PCEP is that PCReq messages MUST NOT modify the state of the PCE LSP-DB in any way. Therefore, PCReq messages are useful for many OAM ping/traceroute applications where the PCC wishes to probe the network without having any effect on the existing LSPs.

The PCC MAY delegate an empty LSP to the PCE and then wait for the PCE to send PCUpd, without sending PCReq. We shall refer to this process as "stateful bringup". The PCE MUST support the original stateless bringup, for backward compatibility purposes. Supporting stateful bringup should not require introducing any new behavior on the PCE, because as mentioned earlier, the PCE MUST NOT modify LSP-DB state based on PCReq messages. So whether the PCE has received a PCReq or not, it MUST process the PCRpt all the same.

An example of stateful bringup follows. In our example the PCC starts off by using LSP-ID of 0. The value 0 does not hold any special meaning, any other 16-bit value could have been used.

PCC has no LSP yet, but wants to establish a path. PCC sends PCRpt(R-FLAG=0, D-flag=1, OPER-FLAG=DOWN, PLSP-ID=100, LSP-ID=0, ERO={}).

TUNNEL		LSP	
PLSP-ID=100		LSP-ID=0, D-flag=1, OPER=DOWN, ERO={}	

Figure 1: Content of LSP DB

PCC received a PCUpd from the PCE and has decided to install the ERO={A} from that PCUpd. PCC sends PCRpt(R-FLAG=0, D-flag=1, OPER-FLAG=UP, PLSP-ID=100, LSP-ID=0, ERO={A}).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=0, D-flag=1, OPER=UP, ERO={A}

Figure 2: Content of LSP DB

3.4. Successful MBB

Below we give an example of doing MBB to switch the tunnel from one path to another. We represent the path encoded into the ERO object as ERO={A} and ERO={B}.

PCC has an existing LSP in UP state, with LSP-ID=2. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=2, ERO={A}, OPER-FLAG=UP).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=2, ERO={A}, OPER=UP

Figure 3: Content of LSP DB

PCC initiates the MBB procedure by creating a new LSP with LSP-ID=3. It does not matter what triggered the creation of the new LSP, it could have been due to a new path received via PCUpd (if the given tunnel is delegated), or it could have been local computation on the PCC (if the tunnel is locally computed on the PCC), or it could have been a change in configuration on the PCC (if the tunnel's path is explicitly configured on the PCC). It is important to emphasize that the procedure for updating the LSP-DB is common, regardless of the trigger that caused the change.

PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=3, ERO={B}, OPER-FLAG=UP).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=2, ERO={A}, OPER=UP
	LSP-ID=3, ERO={B}, OPER=UP

Figure 4: Content of LSP DB

After some time, the PCC decides to destroy the old LSP. PCC sends PCRpt(R-FLAG=1, PLSP-ID=100, LSP-ID=2).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=3, ERO={B}, OPER=UP

Figure 5: Content of LSP DB

3.5. Aborted MBB

The MBB process can abort when the newly created LSP is destroyed before it is installed as traffic carrying. This scenario is described below.

PCC has an existing LSP in UP state, with LSP-ID=2. PCC sends PCRpt(R-FLAG=0, OPER-FLAG=UP, PLSP-ID=100, LSP-ID=2).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=2, OPER=UP

Figure 6: Content of LSP DB

MBB procedure is initiated, a new LSP is created with LSP-ID=3. LSP is currently being established, so its oper state is DOWN. PCC sends PCRpt(R-FLAG=0, OPER-FLAG=DOWN, PLSP-ID=100, LSP-ID=3).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=2, OPER=UP LSP-ID=3, OPER=DOWN

Figure 7: Content of LSP DB

MBB procedure is aborted. PCC sends PCRpt(R-FLAG=1, PLSP-ID=100, LSP-ID=3).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=2, OPER=UP

Figure 8: Content of LSP DB

4. PCEP Association Database

PCEP Association is a group of zero or more LSPs.

The PCE ASSO DB is populated by PCRpt messages and MAY also be populated via configuration on the PCE itself. An Association is identified by the Association Parameters. The Association parameters contain many fields, so for convenience we will group all the fields into a single value. We will use ASSO_PARAM=A, ASSO_PARAM=B, to refer to different PCEP Associations: A and B, respectively.

4.1. 2 LSPs in same Association

Below, we give an example of LSPs joining the same Association.

PCC creates the first LSP. PCC sends PCRpt (R-FLAG=0, PLSP-ID=100, LSP-ID=1, ASSO_PARAM=A, ASSO_R_FLAG=0).

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1

Figure 9: Content of PCE ASSO DB

PCC creates the second LSP. PCC sends PCRpt (R-FLAG=0, PLSP-ID=200, LSP-ID=1, ASSO_PARAM=A, ASSO_R_FLAG=0).

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1 PLSP-ID=200, LSP-ID=1

Figure 10: Content of PCE ASSO DB

PCC updates the first LSP, the PCC is NOT REQUIRED to send the ASSOCIATION object in this PCRpt, since the LSP is already in the

Association. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=1). The content of the PCE ASSO DB is unchanged. Note that the PCC MUST send the ASSOCIATION OBJECT in the first PCRpt during SYNC state, even if it has already issued a PCRpt with the association object sometime in the past with this PCE. The synchronization steps outlined in [I-D.ietf-pce-association-group] are to be followed.

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1 PLSP-ID=200, LSP-ID=1

Figure 11: Content of PCE ASSO DB

PCC decides to delete the second LSP. PCC sends PCRpt(R-FLAG=1, PLSP-ID=200, LSP-ID=1).

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1

Figure 12: Content of PCE ASSO DB

PCC decides to remove the first LSP from the Association, but not delete the LSP itself. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=1, ASSO_PARAM=A, ASSO_R_FLAG=1). The PCE ASSO DB is now empty.

ASSO	LSP
ASSO_PARAM=A	

Figure 13: Content of PCE ASSO DB

4.2. Switch Association during MBB

Each new LSP (identified by the LSP-ID) does not inherit the Association membership of any previous LSPs within the same Tunnel. This is done so that a Tunnel can have two LSPs that are in different Associations, this may be required when switching from one Association to another.

Below, we give an example a Tunnel going through MBB and switching from Association A to Association B.

PCC creates the first LSP. PCC sends PCRpt (R-FLAG=0, PLSP-ID=100, LSP-ID=1, ASSO_PARAM=A, ASSO_R_FLAG=0).

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1

Figure 14: Content of PCE ASSO DB

PCC creates the MBB LSP in a different Association. PCC sends PCRpt (R-FLAG=0, PLSP-ID=100, LSP-ID=2, ASSO_PARAM=B, ASSO_R_FLAG=0).

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1
ASSO_PARAM=B	PLSP-ID=100, LSP-ID=2

Figure 15: Content of PCE ASSO DB

PCC deletes the old LSP. PCC sends PCRpt (R-FLAG=1, PLSP-ID=100, LSP-ID=1).

ASSO	LSP
ASSO_PARAM=B	PLSP-ID=100, LSP-ID=2

Figure 16: Content of PCE ASSO DB

5. Acknowledgement

6. References

6.1. Normative References

- [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>>.

- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [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>>.
- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.
- [I-D.ietf-pce-association-group]
Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Dhody, D., and Y. Tanaka, "PCEP Extensions for Establishing Relationships Between Sets of LSPs", draft-ietf-pce-association-group-09 (work in progress), April 2019.

6.2. Informative References

- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.

Appendix A. Contributors

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

Email: dhruv.ietf@gmail.com

Andrew Stone
Nokia
Ottawa, Canada

Email: andrew.stone@nokia.com

Authors' Addresses

Mike Koldychev
Cisco Systems, Inc.
2000 Innovation Drive
Kanata, Ontario K2K 3E8
Canada

Email: mkoldych@cisco.com

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

Email: msiva@cisco.com

Mahendra Singh Negi
Huawei Technologies

Email: mahendrasingh@huawei.com

Diego Achaval
Nokia

Email: diego.achaval@nokia.com

Hari Kotni
Juniper Networks, Inc

Email: hkotni@juniper.net

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: 23 August 2022

M. Koldychev
Cisco Systems, Inc.
S. Sivabalan
Ciena Corporation
S. Peng
Huawei Technologies
D. Achaval
Nokia
H. Kotni
Juniper Networks, Inc
February 2022

PCEP Operational Clarification
draft-koldychev-pce-operational-05

Abstract

This document proposes some important simplifications to the original PCEP protocol and also serves to clarify certain aspects of PCEP operation. The content of this document has been compiled based on the feedback from several multi-vendor interop exercises. Several constructs are introduced, such as the LSP-DB and the ASSO-DB.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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 <https://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 5 August 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. PCEP LSP Database	4
3.1. Structure	4
3.2. Synchronization	4
3.3. Stateful Bringup	5
3.4. Successful MBB	7
3.5. Aborted MBB	8
4. PCEP Association Database	8
4.1. 2 LSPs in same Association	9
4.2. Switch Association during MBB	10
5. Computation Constraints	11
6. Use of RRO, SR-RRO and SRv6-RRO objects	12
7. Security Considerations	12
8. IANA Considerations	12
9. Acknowledgement	12
10. Normative References	12
Appendix A. Contributors	13
Authors' Addresses	14

1. Introduction

The PCEP protocol started off being purely stateless with PCReq and PCReply messages, as described in Path Computation Element (PCE) Communication Protocol (PCEP) [RFC5440]. Stateless PCEP operates in a "pull" model, i.e., PCC has to periodically ask the PCE for updates to the path, even if the path has not changed.

Stateful PCEP was later introduced in PCEP Extensions for the Stateful PCE Model [RFC8231]. Stateful PCEP operates in a "push" model, where the PCC can register with PCE to receive future updates about the path, and there is no need to ask the PCE periodically.

The current document serves to optimize the original procedure in [RFC8231] to drop the PCReq and PCReply exchange, which greatly simplifies implementation and optimizes the protocol.

Due to different interpretations of PCEP standards, it was found that implementations often had to adjust their behavior in order to interoperate. The current document serves to clarify certain aspects of PCEP to make it easier to produce interoperable implementations of PCEP.

2. Terminology

The following terminologies are used in this document:

PCC: Path Computation Client. Any client application requesting a path computation to be performed by a Path Computation Element.

PCE: Path Computation Element. An entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints.

PCEP: Path Computation Element Protocol.

MBB: Make-Before-Break. A procedure during which the head-end of a traffic-engineered path wishes to move traffic to a new path without losing any traffic, by first "making" a new path and then "breaking" the old path.

Association parameters: As described in [RFC8697], the combination of the mandatory fields Association type, Association ID and Association Source in the ASSOCIATION object uniquely identify the association group. If the optional TLVs - Global Association Source or Extended Association ID are included, then they MUST be included in combination with mandatory fields to uniquely identify the association group.

Association information: As described in [RFC8697], the ASSOCIATION object could also include other optional TLVs based on the association types, that provides 'information' related to the association type.

ERO: Explicit Route Object is the path of the LSP encoded into a PCEP object. To represent an empty ERO object, i.e., without any subobjects, we use the notation "ERO={}". To represent an ERO object containing some given sequence of subobjects, we use the notation "ERO={A}".

3. PCEP LSP Database

We introduce the concept of the LSP-DB, as a database of actual LSP state in the network. This concept is not explicitly defined in [RFC8231], but is fully compatible with it. We use the LSP-DB to describe how certain actions are performed, because it is easier to define actions as a function of database state, rather than as a function of previously received messages. The structure and format of the LSP-DB MUST be common among all dataplane types (i.e., RSVP-TE/SR-TE/SRv6), all instantiation methods (i.e., PCC-initiated/PCE-initiated), all destination types (i.e., point-to-point/point-to-multipoint).

Note that we use the term "Tunnel" somewhat loosely here, to mean "the object identified by the PLSP-ID". It may or may not be an actual tunnel in the implementation. For example, working and protect paths can be implemented as one tunnel interface, but in PCEP we would refer to them as two different Tunnels, because they would have different PLSP-IDs.

Note that the term "LSP", which stands for "Label Switched Path", if taken too literally would restrict our discussion to MPLS dataplane only. In this document, we allow the term "LSP" to refer to any path, regardless of the dataplane format. So that an LSP can refer to MPLS and SRv6 dataplane paths.

3.1. Structure

[RFC8231] states that the LSP-IDENTIFIERS TLV contains the key that MUST be used to differentiate different LSPs during make before break procedure. We further clarify here that PCEP LSPs exist in a 2-tier structure. The top tier is the "Tunnel", identified by the PLSP-ID and/or SYMBOLIC-NAME, while the lower tier is the "LSP", identified by the values in LSP-IDENTIFIERS TLV. A single Tunnel may contain multiple LSPs at the same time, i.e., a Tunnel is a container for LSPs. A Tunnel MUST have at least one LSP and when the last LSP is removed from the Tunnel, the Tunnel itself is removed.

3.2. Synchronization

The stateful PCE MUST maintain the PCE LSP-DB, which stores Tunnels and LSPs. The PCE LSP DB is only modified by PCRpt messages. No other PCEP message may modify the PCE LSP DB. The PCC MUST also maintain the PCC LSP DB, which it MUST synchronize with the PCE LSP DB by sending PCRpt messages.

The PCC adds/removes entries to/from its LSP-DB based on what LSPs it creates/destroys in the network. There can be many trigger types for updating the PCC LSP-DB, some examples include PCUpd messages, local computation on the PCC, local configuration on the PCC, etc. The trigger type does not affect the content of the PCC LSP-DB, i.e., the content of the PCC LSP-DB is updated identically regardless of the trigger type.

Whenever a PCC modifies an entry in its PCC LSP-DB, it MUST send a PCRpt message to the PCE (or multiple PCEs), to synchronize this change. Ensuring this synchronization is always in place allows one to define behavior as a function of LSP-DB state, instead of defining behavior as a function of what PCEP messages were sent or received.

The PCE MUST always act on the latest state of the PCE LSP DB. Note that this does not mean that the PCE cannot use information from outside of LSP-DB. For example, the PCE can use other mechanisms to collect traffic statistics and use them in the computation. However, these traffic statistics are not part of the LSP-DB, but only reference it.

The LSP-DB on both the PCC and the PCE only stores the actual state in the network, it does not store the desired state. For example, consider the case of PCE Initiated LSP, configured on the PCE. When the operator modifies the configuration of this LSP, that is a change in desired state. The actual state has not yet changed, so LSP-DB is not modified yet. The LSP-DB is only modified after the PCE sends PCInit/PCUpd message to the PCC and the PCC decides to act on that message. When the PCC acts on message, it would update its own PCC LSP DB and immediately send PCRpt to the PCE to synchronize the change. When the PCE receives the PCRpt msg, it updates its own PCE LSP DB. After this, the PCC LSP DB and PCE LSP DB are in sync.

3.3. Stateful Bringup

[RFC8231] in section 5.8.2, allows delegation of an LSP in operationally down state, but at the same time mandates the use of PCReq, before sending PCRpt. In this document, we would like to make it clear that sending PCReq is optional.

We shall refer to the process of sending PCReq before PCRpt as "stateless bringup". In reality, stateless bringup introduces overhead and is not possible to enforce from the PCE, because the stateless PCE is not supposed to keep any per-LSP state about previous PCReq messages. It was found that many vendors choose to ignore this requirement and send the PCRpt directly, without going through PCReq. This section will serve to explain and to validate this behavior.

Even though all the major vendors today are moving to the stateful PCE model, it does not deprecate the need for stateless PCEP. The key property of stateless PCEP is that PCReq messages MUST NOT modify the state of the PCE LSP-DB in any way. Therefore, PCReq messages are useful for many OAM ping/traceroute applications where the PCC wishes to probe the network without having any effect on the existing LSPs.

The PCC MAY delegate an empty LSP to the PCE and then wait for the PCE to send PCUpd, without sending PCReq. We shall refer to this process as "stateful bringup". The PCE MUST support the original stateless bringup, for backward compatibility purposes. Supporting stateful bringup should not require introducing any new behavior on the PCE, because as mentioned earlier, the PCE MUST NOT modify LSP-DB state based on PCReq messages. So whether the PCE has received a PCReq or not, it MUST process the PCRpt all the same.

An example of stateful bringup follows. In our example the PCC starts off by using LSP-ID of 0. The value 0 does not hold any special meaning, any other 16-bit value could have been used.

PCC has no LSP yet, but wants to establish a path. PCC sends PCRpt(R-FLAG=0, D-flag=1, OPER=DOWN, PLSP-ID=100, LSP-ID=0, ERO={}).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=0, D-flag=1, OPER=DOWN, ERO={}

Figure 1: Content of LSP DB

PCC received a PCUpd from the PCE and has decided to install the ERO={A} from that PCUpd. PCC sends PCRpt(R-FLAG=0, D-flag=1, OPER=UP, PLSP-ID=100, LSP-ID=0, ERO={A}).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=0, D-flag=1, OPER=UP, ERO={A}

Figure 2: Content of LSP DB

3.4. Successful MBB

Below we give an example of doing MBB to switch the Tunnel from one path to another. We represent the path encoded into the ERO object as $ERO=\{A\}$ and $ERO=\{B\}$.

PCC has an existing LSP in UP state, with $LSP-ID=2$. PCC sends $PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=2, ERO=\{A\}, OPER-FLAG=UP)$.

TUNNEL	LSP
PLSP-ID=100	LSP-ID=2, ERO={A}, OPER=UP

Figure 3: Content of LSP DB

PCC initiates the MBB procedure by creating a new LSP with $LSP-ID=3$. It does not matter what triggered the creation of the new LSP, it could have been due to a new path received via PCUpd (if the given Tunnel is delegated), or it could have been local computation on the PCC (if the Tunnel is locally computed on the PCC), or it could have been a change in configuration on the PCC (if the Tunnel's path is explicitly configured on the PCC). It is important to emphasize that the procedure for updating the LSP-DB is common, regardless of the trigger that caused the change.

PCC sends $PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=3, ERO=\{B\}, OPER-FLAG=UP)$.

TUNNEL	LSP
PLSP-ID=100	LSP-ID=2, ERO={A}, OPER=UP LSP-ID=3, ERO={B}, OPER=UP

Figure 4: Content of LSP DB

After traffic has successfully switched to the new LSP, the PCC cleans up the old LSP. PCC sends $PCRpt(R-FLAG=1, PLSP-ID=100, LSP-ID=2)$.

TUNNEL	LSP
PLSP-ID=100	LSP-ID=3, ERO={B}, OPER=UP

Figure 5: Content of LSP DB

3.5. Aborted MBB

The MBB process can abort when the newly created LSP is destroyed before it is installed as traffic carrying. This scenario is described below.

PCC has an existing LSP in UP state, with LSP-ID=2. PCC sends PCRpt (R-FLAG=0, OPER-FLAG=UP, PLSP-ID=100, LSP-ID=2).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=2, OPER=UP

Figure 6: Content of LSP DB

MBB procedure is initiated, a new LSP is created with LSP-ID=3. LSP is currently being established, so its oper state is DOWN. PCC sends PCRpt (R-FLAG=0, OPER-FLAG=DOWN, PLSP-ID=100, LSP-ID=3).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=2, OPER=UP LSP-ID=3, OPER=DOWN

Figure 7: Content of LSP DB

MBB procedure is aborted. PCC sends PCRpt (R-FLAG=1, PLSP-ID=100, LSP-ID=3).

TUNNEL	LSP
PLSP-ID=100	LSP-ID=2, OPER=UP

Figure 8: Content of LSP DB

4. PCEP Association Database

PCEP Association is a group of zero or more LSPs.

The PCE ASSO DB is populated by PCRpt messages and MAY also be populated via configuration on the PCE itself. An Association is identified by the Association Parameters. The Association parameters contain many fields, so for convenience we will group all the fields into a single value. We will use ASSO_PARAM=A, ASSO_PARAM=B, to refer to different PCEP Associations: A and B, respectively.

4.1. 2 LSPs in same Association

Below, we give an example of LSPs joining the same Association.

PCC creates the first LSP. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=1, ASSO_PARAM=A, ASSO_R_FLAG=0).

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1

Figure 9: Content of PCE ASSO DB

PCC creates the second LSP. PCC sends PCRpt(R-FLAG=0, PLSP-ID=200, LSP-ID=1, ASSO_PARAM=A, ASSO_R_FLAG=0).

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1
	PLSP-ID=200, LSP-ID=1

Figure 10: Content of PCE ASSO DB

PCC updates the first LSP, the PCC is NOT REQUIRED to send the ASSOCIATION object in this PCRpt, since the LSP is already in the Association. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=1). The content of the PCE ASSO DB is unchanged. Note that the PCC MUST send the ASSOCIATION OBJECT in the first PCRpt during SYNC state, even if it has already issued a PCRpt with the association object sometime in the past with this PCE. The synchronization steps outlined in [RFC8697] are to be followed.

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1 PLSP-ID=200, LSP-ID=1

Figure 11: Content of PCE ASSO DB

PCC decides to delete the second LSP. PCC sends PCRpt(R-FLAG=1, PLSP-ID=200, LSP-ID=1).

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1

Figure 12: Content of PCE ASSO DB

PCC decides to remove the first LSP from the Association, but not delete the LSP itself. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=1, ASSO_PARAM=A, ASSO_R_FLAG=1). The PCE ASSO DB is now empty.

ASSO	LSP
ASSO_PARAM=A	

Figure 13: Content of PCE ASSO DB

4.2. Switch Association during MBB

Each new LSP (identified by the LSP-ID) does not inherit the Association membership of any previous LSPs within the same Tunnel. This is done so that a Tunnel can have two LSPs that are in different Associations, this may be required when switching from one Association to another.

Below, we give an example a Tunnel going through MBB and switching from Association A to Association B.

PCC creates the first LSP. PCC sends PCRpt(R-FLAG=0, PLSP-ID=100, LSP-ID=1, ASSO_PARAM=A, ASSO_R_FLAG=0).

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1

Figure 14: Content of PCE ASSO DB

PCC creates the MBB LSP in a different Association. PCC sends PCRpt (R-FLAG=0, PLSP-ID=100, LSP-ID=2, ASSO_PARAM=B, ASSO_R_FLAG=0).

ASSO	LSP
ASSO_PARAM=A	PLSP-ID=100, LSP-ID=1
ASSO_PARAM=B	PLSP-ID=100, LSP-ID=2

Figure 15: Content of PCE ASSO DB

PCC deletes the old LSP. PCC sends PCRpt (R-FLAG=1, PLSP-ID=100, LSP-ID=1).

ASSO	LSP
ASSO_PARAM=B	PLSP-ID=100, LSP-ID=2

Figure 16: Content of PCE ASSO DB

5. Computation Constraints

For any PCEP object that does not have an explicit removal flag, the absence of that object indicates removal of the constraint specified by that object. For example, suppose the first state-report contains an LSPA object with some affinity constraints. Then if a subsequent state-report does not contain an LSPA object, then this means that the previously specified affinity constraints do not apply anymore. Same applies to all PCEP objects, like METRIC, BANDWIDTH, etc., which do not have an explicit flag for removal. This simply ensures that it is possible to remove a constraint without using an explicit removal flag.

6. Use of RRO, SR-RRO and SRv6-RRO objects

[RFC8231] defines a PCRpt message which contains <intended-path> known as the ERO object and <actual-path> known as the RRO object. [RFC8664] defines SR-ERO and SR-RRO objects for SR-TE LSPs. [I-D.ietf-pce-segment-routing-ipv6] further defines SRv6-ERO and SRv6-RRO objects for SRv6-TE paths.

In practice RRO data set is the result of signalling of the intended path defined in the ERO via protocol such as RSVP. The ERO and RRO values may be different as the path encoded in the ERO may differ than the RRO such as during protection conditions or if the ERO contains loose hops which are expanded upon. As Segment Routing LSP does not perform any signalling, the values of an SR-ERO/SRv6-ERO and SR-RRO/SRv6-RRO (respectively) are in practice the same, therefore some implementations have omitted the SR-RRO/SRv6-RRO when reporting a SR-TE LSP while others continue to send both SR-ERO/SRv6-ERO and SR-RRO/SRv6-RRO values.

A PCC MUST send an (possibly empty) ERO/SR-ERO/SRv6-ERO in the PCRpt message for every LSP. A PCC MAY send an SR-RRO/SRv6-RRO for an SR-TE/SRv6-TE LSP (respectively). A PCE SHOULD interpret the RRO/SR-RRO/SRv6-RRO as the actual path the LSP is taking but MAY interpret only the ERO/SR-ERO/SRv6-ERO as the actual path. In the absence of an RRO/SR-RRO/SRv6-RRO a PCE SHOULD interpret the ERO/SR-ERO/SRv6-ERO (respectively) as the actual path for the LSP.

7. Security Considerations

None at this time.

8. IANA Considerations

None at this time.

9. Acknowledgement

10. Normative References

- [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>>.
- [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>>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [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>>.
- [RFC8664] Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing", RFC 8664, DOI 10.17487/RFC8664, December 2019, <<https://www.rfc-editor.org/info/rfc8664>>.
- [RFC8697] Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Dhody, D., and Y. Tanaka, "Path Computation Element Communication Protocol (PCEP) Extensions for Establishing Relationships between Sets of Label Switched Paths (LSPs)", RFC 8697, DOI 10.17487/RFC8697, January 2020, <<https://www.rfc-editor.org/info/rfc8697>>.
- [I-D.ietf-pce-segment-routing-ipv6]
Li, C., Negi, M., Sivabalan, S., Koldychev, M., Kaladharan, P., and Y. Zhu, "PCEP Extensions for Segment Routing leveraging the IPv6 data plane", Work in Progress, Internet-Draft, draft-ietf-pce-segment-routing-ipv6-11, 10 January 2022, <<https://www.ietf.org/internet-drafts/draft-ietf-pce-segment-routing-ipv6-11.txt>>.

Appendix A. Contributors

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

Email: dhruv.ietf@gmail.com

Andrew Stone
Nokia
Ottawa, Canada

Email: andrew.stone@nokia.com

Samuel Sidor
Cisco Systems
Bratislava, Slovakia

Email: ssidor@cisco.com

Mahendra Singh Negi
RtBrick Inc
N-17L, 18th Cross Rd, HSR Layout
Bangalore, Karnataka 560102
India

Email: mahend.ietf@gmail.com

Authors' Addresses

Mike Koldychev
Cisco Systems, Inc.
2000 Innovation Drive
Kanata Ontario K2K 3E8
Canada
Email: mkoldych@cisco.com

Siva Sivabalan
Ciena Corporation
385 Terry Fox Dr.
Kanata Ontario K2K 0L1
Canada
Email: ssivabal@ciena.com

Shuping Peng
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing
100095
China
Email: pengshuping@huawei.com

Diego Achaval
Nokia
Email: diego.achaval@nokia.com

Hari Kotni
Juniper Networks, Inc

Email: hkotni@juniper.net

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 8, 2020

S. Litkowski
Orange
S. Sivabalan
Cisco
C. Li
H. Zheng
Huawei Technologies
July 7, 2019

Inter Stateful Path Computation Element (PCE) Communication Procedures.
draft-litkowski-pce-state-sync-06

Abstract

The Path Computation Element Communication Protocol (PCEP) provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients (PCCs) requests. The stateful PCE extensions allow stateful control of Multi-Protocol Label Switching (MPLS) Traffic Engineering Label Switched Paths (TE LSPs) using PCEP.

A Path Computation Client (PCC) can synchronize an LSP state information to a Stateful Path Computation Element (PCE). The stateful PCE extension allows a redundancy scenario where a PCC can have redundant PCEP sessions towards multiple PCEs. In such a case, a PCC gives control on a LSP to only a single PCE, and only one PCE is responsible for path computation for this delegated LSP. The document does not state the procedures related to an inter-PCE stateful communication.

There are some use cases, where an inter-PCE stateful communication can bring additional resiliency in the design, for instance when some PCC-PCE sessions fails. The inter-PCE stateful communication may also provide a faster update of the LSP states when such an event occurs. Finally, when, in a redundant PCE scenario, there is a need to compute a set of paths that are part of a group (so there is a dependency between the paths), there may be some cases where the computation of all paths in the group is not handled by the same PCE: this situation is called a split-brain. This split-brain scenario may lead to computation loops between PCEs or suboptimal path computation.

This document describes the procedures to allow a stateful communication between PCEs for various use-cases and also the procedures to prevent computations loops.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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 <https://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 January 8, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction and Problem Statement	3
1.1. Reporting LSP changes	4
1.2. Split-brain	5
1.3. Applicability to H-PCE	12
2. Proposed solution	12
2.1. State-sync session	12

2.2. Master/Slave relationship between PCE	14
3. Procedures and Protocol Extensions	14
3.1. Opening a state-sync session	14
3.1.1. Capability Advertisement	14
3.2. State synchronization	15
3.3. Incremental updates and report forwarding rules	16
3.4. Maintaining LSP states from different sources	17
3.5. Computation priority between PCEs and sub-delegation	18
3.6. Passive stateful procedures	19
3.7. PCE initiation procedures	20
4. Examples	20
4.1. Example 1	20
4.2. Example 2	22
4.3. Example 3	24
5. Using Master/Slave computation and state-sync sessions to increase scaling	25
6. PCEP-PATH-VECTOR-TLV	27
7. Security Considerations	28
8. Acknowledgements	28
9. IANA Considerations	28
9.1. PCEP-Error Object	28
9.2. PCEP TLV Type Indicators	28
9.3. STATEFUL-PCE-CAPABILITY TLV	29
10. References	29
10.1. Normative References	29
10.2. Informative References	29
Appendix A. Contributors	30
Authors' Addresses	30

1. Introduction and Problem Statement

The Path Computation Element communication Protocol (PCEP) [RFC5440] provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients' (PCCs) requests.

A stateful PCE [RFC8231] is capable of considering, for the purposes of path computation, not only the network state in terms of links and nodes (referred to as the Traffic Engineering Database or TED) but also the status of active services (previously computed paths, and currently reserved resources, stored in the Label Switched Paths Database (LSP-DB)).

[RFC8051] describes general considerations for a stateful PCE deployment and examines its applicability and benefits, as well as its challenges and limitations through a number of use cases.

The examples in this section are for illustrative purpose to showcase the need for inter-PCE stateful PCEP sessions.

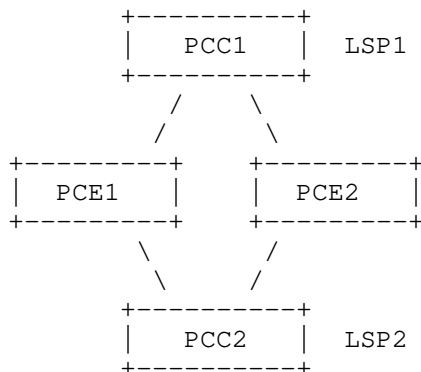
1.1. Reporting LSP changes

When using a stateful PCE ([RFC8231]), a PCC can synchronize an LSP state information to the stateful PCE. If the PCC grants the control on the LSP to the PCE (called delegation [RFC8231]), the PCE can update the LSP parameters at any time.

In a multi PCE deployment (redundancy, loadbalancing...), with the current specification defined in [RFC8231], when a PCE makes an update, it is the PCC that is in charge of reporting the LSP status to all PCEs with LSP parameter change which brings additional hops and delays in notifying the overall network of the LSP parameter change.

This delay may affect the reaction time of the other PCEs, if they need to take action after being notified of the LSP parameter change.

Apart from the synchronization from the PCC, it is also useful if there is synchronization mechanism between the stateful PCEs. As stateful PCE make changes to its delegated LSPs, these changes (pending LSPs and the sticky resources [RFC7399]) can be synchronized immediately to the other PCEs.



In the figure above, we consider a load-balanced PCE architecture, so PCE1 is responsible to compute paths for PCC1 and PCE2 is responsible to compute paths for PCC2. When PCE1 triggers an LSP update for LSP1, it sends a PCUpd message to PCC1 containing the new parameters for LSP1. PCC1 will take the parameters into account and will send a PCRpt message to PCE1 and PCE2 reflecting the changes. PCE2 will so

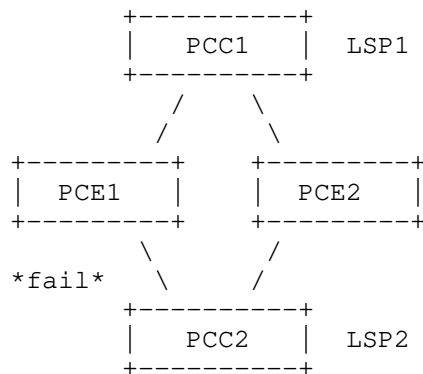
be notified of the change only after receiving the PCRpt message from PCC1.

Let's consider that the LSP1 parameters changed in such a way that LSP1 will take over resources from LSP2 with a higher priority. After receiving the report from PCC1, PCE2 will therefore try to find a new path for LSP2. If we consider that there is a round trip delay of about 150 milliseconds (ms) between the PCEs and PCC1 and a round trip delay of 10 ms between the two PCEs, it will take more than 150 ms for PCE2 to be notified of the change.

Adding a PCEP session between PCE1 and PCE2 may allow to reduce the synchronization time, so PCE2 can react more quickly by taking the pending LSPs and attached resources into account during path computation and reoptimization.

1.2. Split-brain

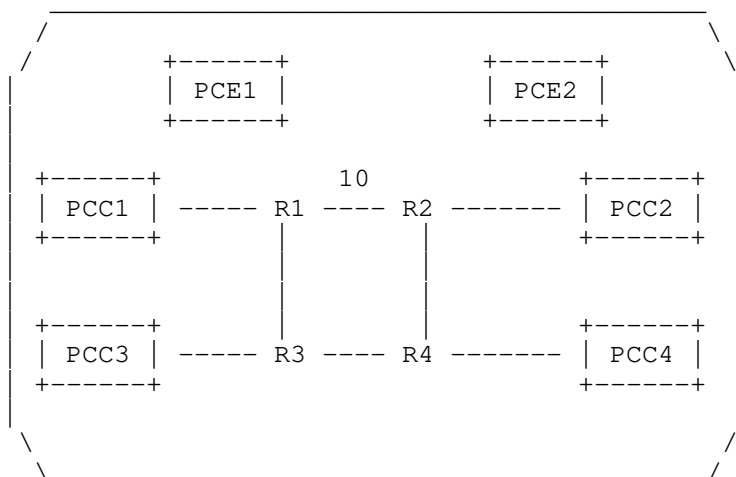
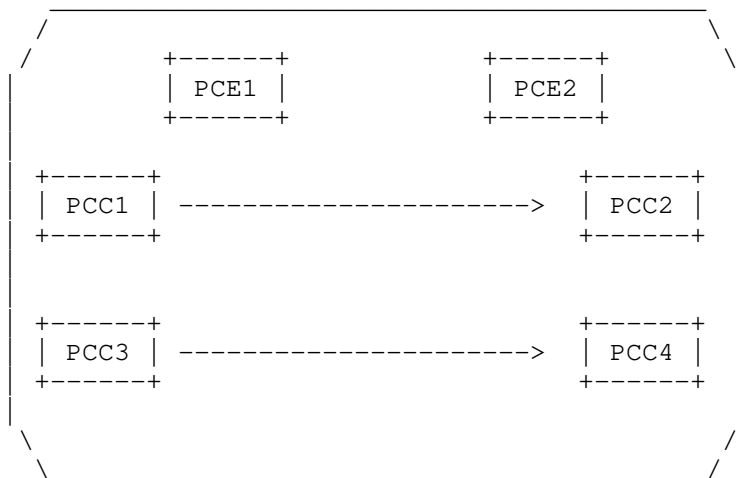
In a resiliency case, a PCC has redundant PCEP sessions towards multiple PCEs. In such a case, a PCC gives control on an LSP to a single PCE only, and only this PCE is responsible for the path computation for the delegated LSP: the PCC achieves this by setting the D flag only towards the active PCE [RFC8231] selected for delegation. The election of the active PCE to delegate an LSP is controlled by each PCC. The PCC usually elects the active PCE by a local configured policy (by setting a priority). Upon PCEP session failure, or active PCE failure, PCC may decide to elect a new active PCE by sending new PCRpt message with D flag set to this new active PCE. When the failed PCE or PCEP session comes back online, it will be up to the implementation to do pre-emption. Doing pre-emption may lead to some disruption on the existing path if path results from both PCEs are not exactly the same. By considering a network with multiple PCCs and implementing multiple stateful PCEs for redundancy purpose, there is no guarantee that at any time all the PCCs delegate their LSPs to the same PCE.



In the example above, we consider that by configuration, both PCCs will firstly delegate their LSPs to PCE1. So, PCE1 is responsible for computing a path for both LSP1 and LSP2. If the PCEP session between PCC2 and PCE1 fails, PCC2 will delegate LSP2 to PCE2. So PCE1 becomes responsible only for LSP1 path computation while PCE2 is responsible for the path computation of LSP2. When the PCC2-PCE1 session is back online, PCC2 will keep using PCE2 as active PCE (consider no pre-emption in this example). So the result is a permanent situation where each PCE is responsible for a subset of path computation.

This situation is called a split-brain scenario, as there are multiple computation brains running at the same time while a central computation unit was required in some deployments/usecases.

Further, there are use cases where a particular LSP path computation is linked to another LSP path computation: the most common use case is path disjointness (see [I-D.ietf-pce-association-diversity]). The set of LSPs that are dependant to each other may start from a different head-end.



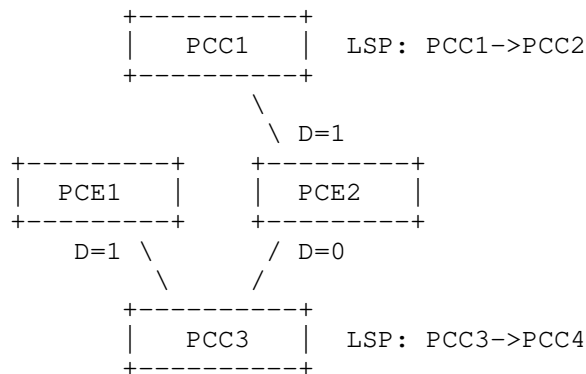
In the figure above, the requirement is to create two link-disjoint LSPs: PCC1->PCC2 and PCC3->PCC4. In the topology, all links cost metric is set to 1 except for the link 'R1-R2' which has a metric of 10. The PCEs are responsible for the path computation and PCE1 is the active primary PCE for all PCCs in the nominal case.

Scenario 1:

In the normal case (PCE1 as active primary PCE), consider that PCC1->PCC2 LSP is configured first with the link disjointness constraint, PCE1 sends a PCUpd message to PCC1 with the ERO: R1->R3->R4->R2->PCC2 (shortest path). PCC1 signals and installs the path. When PCC3->PCC4 is configured, the PCEs already knows the path of PCC1->PCC2 and can compute a link-disjoint path : the solution requires to move PCC1->PCC2 onto a new path to let room for the new LSP. PCE1 sends a PCUpd message to PCC1 with the new ERO: R1->R2->PCC2 and a PCUpd to PCC3 with the following ERO: R3->R4->PCC4. In the normal case, there is no issue for PCE1 to compute a link-disjoint path.

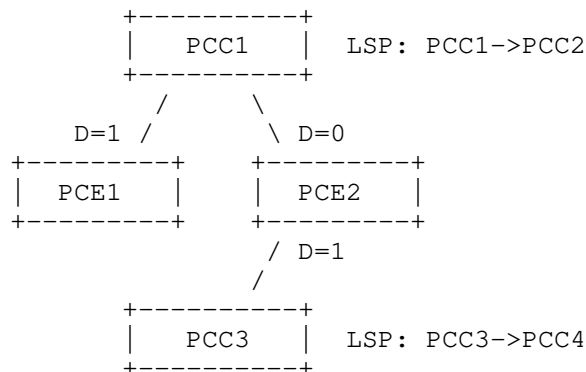
Scenario 2:

Consider that PCC1 lost its PCEP session with PCE1 (all other PCEP sessions are UP). PCC1 delegates its LSP to PCE2.



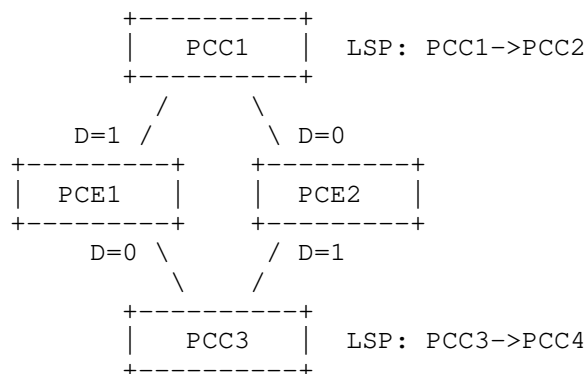
Consider that the PCC1->PCC2 LSP is configured first with the link disjointness constraint, PCE2 (which is the new active primary PCE for PCC1) sends a PCUpd message to PCC1 with the ERO: R1->R3->R4->R2->PCC2 (shortest path). When PCC3->PCC4 is configured, PCE1 is not aware of LSPs from PCC1 any more, so it cannot compute a disjoint path for PCC3->PCC4 and will send a PCUpd message to PCC3 with a shortest path ERO: R3->R4->PCC4. When PCC3->PCC4 LSP will be reported to PCE2 by PCC3, PCE2 will ensure disjointness computation and will correctly move PCC1->PCC2 (as it owns delegation for this LSP) on the following path: R1->R2->PCC2. With this sequence of event and these PCEP sessions, disjointness is ensured.

Scenario 3:



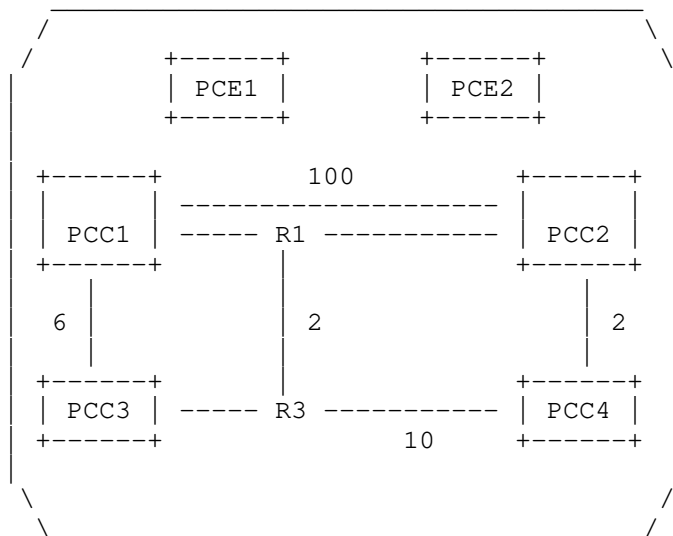
Consider the above PCEP sessions and the PCC1->PCC2 LSP is configured first with the link disjointness constraint, PCE1 computes the shortest path as it is the only LSP in the disjoint association group that it is aware of: R1->R3->R4->R2->PCC2 (shortest path). When PCC3->PCC4 is configured, PCE2 must compute a disjoint path for this LSP. The only solution found is to move PCC1->PCC2 LSP on another path, but PCE2 cannot do it as it does not have delegation for this LSP. In this set-up, PCEs are not able to find a disjoint path.

Scenario 4:



Consider the above PCEP sessions and that PCEs are configured to fallback to shortest path if disjointness cannot be found as described in [I-D.ietf-pce-association-diversity]. The PCC1->PCC2 LSP is configured first, PCE1 computes shortest path as it is the only LSP in the disjoint association group that it is aware of: R1->R3->R4->R2->PCC2 (shortest path). When PCC3->PCC4 is configured, PCE2 must compute a disjoint path for this LSP. The only solution found is to move PCC1->PCC2 LSP on another path, but PCE2 cannot do

Scenario 5:

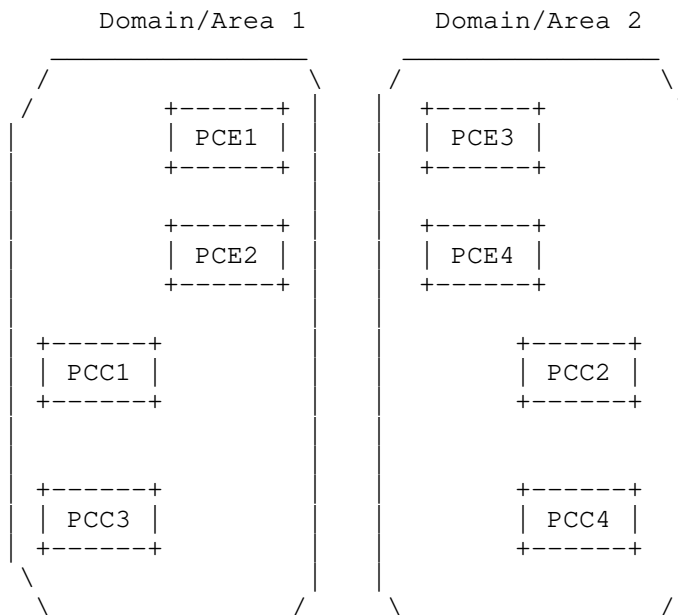


[Page 10]

those paths will be reported to both PCEs, this will trigger CSPF again. An infinite loop of CSPF computation is then happening with a permanent flap of paths because of the split-brain situation.

This permanent computation loop comes from the inconsistency between the state of the LSPs as seen by each PCE due to the split-brain: each PCE is trying to modify at the same time its delegated path based on the last received path information which de facto invalidates this received path information.

Scenario 6: multi-domain



In the example above, suppose that the disjoint LSPs from PCC1 to PCC2 and from PCC4 to PCC3 are created. All the PCEs have the knowledge of both domain topologies (e.g. using BGP-LS [RFC7752]). For operation/management reason, each domain uses its own group of redundant PCEs. PCE1/PCE2 in domain 1 have PCEP sessions with PCC1 and PCC3 while PCE3/PCE4 in domain 2 have PCEP sessions with PCC2 and PCC4. As PCE1/2 do not know about LSPs from PCC2/4 and PCE3/4 do not know about LSPs from PCC1/3, there is no possibility to compute the disjointness constraint. This scenario can also be seen as a split-brain scenario. This multi-domain architecture (with multiple groups of PCEs) can also be used in a single domain, where an operator wants to limit the failure domain by creating multiple groups of PCEs

maintaining a subset of PCCs. As for the multi-domain example, there will be no possibility to compute disjoint path starting from head-ends managed by different PCE groups.

In this document, we propose a solution that address the possibility to compute LSP association based constraints (like disjointness) in split-brain scenarios while preventing computation loops.

1.3. Applicability to H-PCE

[I-D.ietf-pce-stateful-hpce] describes general considerations and use cases for the deployment of Stateful PCE(s) using the Hierarchical PCE [RFC6805] architecture. In this architecture there is a clear need to communicate between a child stateful PCE and a parent stateful PCE. The procedures and extensions as described in Section 3 are equally applicable to H-PCE scenario.

2. Proposed solution

Our solution is based on :

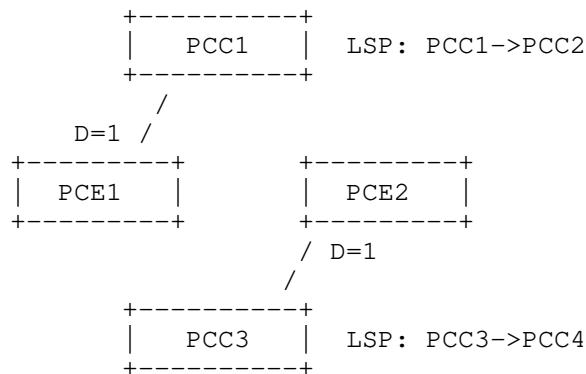
- o The creation of the inter-PCE stateful PCEP session with specific procedures.
- o A Master/Slave relationship between PCEs.

2.1. State-sync session

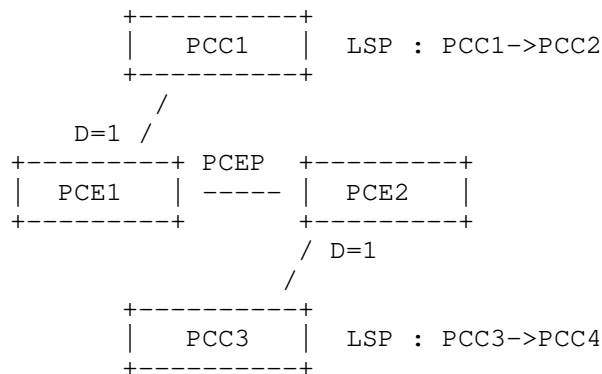
This document proposes to set-up a PCEP session between the stateful PCEs. Creating such a session is already authorized by multiple scenarios like the one described in [RFC4655] (multiple PCEs that are handling part of the path computation) and [RFC6805] (hierarchical PCE) but was only focused on stateless PCEP sessions. As stateful PCE brings additional features (LSP state synchronization, path update, delegation, ...), thus some new behaviours need to be defined.

This inter-PCE PCEP session will allow exchange of LSP states between PCEs that would help some scenario where PCEP sessions are lost between PCC and PCE. This inter-PCE PCEP session is called a state-sync session.

For example, in the scenario below, there is no possibility to compute disjointness as there is no PCE that is aware of both LSPs.



If we add a state-sync session, PCE1 will be able to do state synchronization via PCRpt messages for its LSP to PCE2 and PCE2 will do the same. All the PCEs will be aware of all LSPs even if PCC->PCE session are down. PCEs will then be able to compute disjoint paths.



The procedures associated with this state-sync session are defined in Section 3.

By just adding this state-sync session, it does not ensure that a path with LSP association based constraints can always be computed and does not prevent computation loop, but it increases resiliency and ensures that PCEs will have the state information for all LSPs. In addition, this session will allow for a PCE to update the other PCEs providing a faster synchronization mechanism than relying on PCCs only.

2.2. Master/Slave relationship between PCE

As seen in Section 1, performing a path computation in a split-brain scenario (multiple PCEs responsible for computation) may provide a non optimal LSP placement, no path or computation loops. To provide the best efficiency, an LSP association constraint based computation requires that a single PCE performs the path computation for all LSPs in the association group. Note that, it could be all LSPs belonging to a particular association group, or all LSPs from a particular PCC, or all LSPs in the network that need to be delegated to a single PCE based on the deployment scenarios.

This document propose to add a priority mechanism between PCEs to elect a single computing PCE. Using this priority mechanism, PCEs can agree on the PCE that will be responsible for the computation for a particular association group, or set of LSPs. The priority could be set per association, per PCC, or for all LSPs. How this priority is set or advertised is out of scope of this document. The rest of the text consider association group as an example.

When a single PCE is performing the computation for a particular association group, no computation loop can happen and an optimal placement will be provided. The other PCEs will only act as state collectors and forwarders.

In the scenario described in Section 2.1, PCE1 and PCE2 will decide that PCE1 will be responsible for the path computation of both LSPs. If we first configure PCC1->PCC2, PCE1 computes shortest path at it is the only LSP in the disjoint-group that it is aware of: R1->R3->R4->R2->PCC2 (shortest path). When PCC3->PCC4 is configured, PCE2 will not perform computation even if it has delegation but forwards the delegation via PCRpt message to PCE1 through the state-sync session. PCE1 will then perform disjointness computation and will move PCC1->PCC2 onto R1->R2->PCC2 and provides an ERO to PCE2 for PCC3->PCC4: R3->R4->PCC4. The PCE2 will further update the PCC3 with the new path.

3. Procedures and Protocol Extensions

3.1. Opening a state-sync session

3.1.1. Capability Advertisement

A PCE indicates its support of state-sync procedures during the PCEP Initialization phase [RFC5440]. The OPEN object in the Open message MUST contains the "Stateful PCE Capability" TLV defined in [RFC8231]. A new P (INTER-PCE-CAPABILITY) flag is introduced to indicate the support of state-sync.

This document adds a new bit in the Flags field with :

P (INTER-PCE-CAPABILITY - 1 bit): If set to 1 by a PCEP Speaker, the PCEP speaker indicates that the session MUST follow the state-sync procedures as described in this document. The P bit MUST be set by both speakers: if a PCEP Speaker receives a STATEFUL-PCE-CAPABILITY TLV with P=0 while it advertised P=1 or if both set P flag to 0, the session SHOULD be set-up but the state-sync procedures MUST NOT be applied on this session.

The U flag [RFC8231] MUST be set when sending the STATEFUL-PCE-CAPABILITY TLV with the P flag set. In case the U flag is not set along with the P flag, the state sync capability is not enabled and it is considered as if P flag is not set. The S flag MAY be set if optimized synchronization is required as per [RFC8232].

3.2. State synchronization

When the state sync capability has been negotiated between stateful PCEs, each PCEP speaker will behave as a PCE and as a PCC at the same time regarding the state synchronization as defined in [RFC8231]. This means that each PCEP Speaker:

- o MUST send a PCRpt message towards its neighbour with S flag set for each LSP in its LSP database learned from a PCC. (PCC role)
- o MUST send the End Of Synchronization Marker towards its neighbour when all LSPs have been reported. (PCC role)
- o MUST wait for the LSP synchronization from its neighbour to end (receiving an End Of Synchronization Marker). (PCE role)

The process of synchronization runs in parallel on each PCE (with no defined order).

Optimized state synchronization procedures MAY be used, as defined in [RFC8232].

When a PCEP Speaker sends a PCRpt on a state-sync session, it MUST add the SPEAKER-IDENTITY-TLV (defined in [RFC8232]) in the LSP Object, the value used will refer to the 'owner' PCC of the LSP. If a PCEP Speaker receives a PCRpt on a state-sync session without this TLV, it MUST discard the PCRpt message and it MUST reply with a PCErr message using error-type=6 (Mandatory Object missing) and error-value=TBD1 (SPEAKER-IDENTITY-TLV missing).

3.3. Incremental updates and report forwarding rules

During the life of an LSP, its state may change (path, constraints, operational state...) and a PCC will advertise a new PCRpt to the PCE for each such change.

When propagating LSP state changes from a PCE to other PCEs, it is mandatory to ensure that a PCE always uses the freshest state coming from the PCC.

When a PCE receives a new PCRpt from a PCC with the LSP-DB-VERSION, the PCE MUST forward the PCRpt to all its state-sync sessions and MUST add the appropriate SPEAKER-IDENTITY-TLV in the PCRpt. In addition, it MUST add a new ORIGINAL-LSP-DB-VERSION TLV (described below). The ORIGINAL-LSP-DB-VERSION contains the LSP-DB-VERSION coming from the PCC.

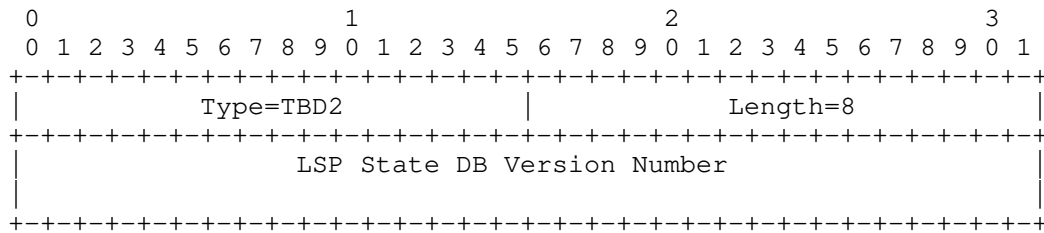
When a PCE receives a new PCRpt from a PCC without the LSP-DB-VERSION, it SHOULD NOT forward the PCRpt on any state-sync sessions and log such an event on the first occurrence.

When a PCE receives a new PCRpt from a PCC with the R flag set and a LSP-DB-VERSION TLV, the PCE MUST forward the PCRpt to all its state-sync sessions keeping the R flag set (Remove) and MUST add the appropriate SPEAKER-IDENTITY-TLV and ORIGINAL-LSP-DB-VERSION TLV in the PCRpt message.

When a PCE receives a PCRpt from a state-sync session, it MUST NOT forward the PCRpt to other state-sync sessions. This helps to prevent message loops between PCEs. As a consequence, a full mesh of PCEP sessions between PCEs is REQUIRED.

When a PCRpt is forwarded, all the original objects and values are kept. As an example, the PLSP-ID used in the forwarded PCRpt will be the same as the original one used by the PCC. Thus an implementation supporting this document MUST consider SPEAKER-IDENTITY-TLV and PLSP-ID together to uniquely identify an LSP on the state-sync session.

The ORIGINAL-LSP-DB-VERSION TLV is encoded as follows and MUST always contain the LSP-DB-VERSION received from the owner PCC of the LSP:



Using the ORIGINAL-LSP-DB-VERSION TLV allows a PCE to keep using optimized synchronization ([RFC8232]) with another PCE. In such a case, the PCE will send a PCRpt to another PCE with both ORIGINAL-LSP-DB-VERSION TLV and LSP-DB-VERSION TLV. The ORIGINAL-LSP-DB-VERSION TLV will contain the version number as allocated by the PCC while the LSP-DB-VERSION will contain the version number allocated by the local PCE.

3.4. Maintaining LSP states from different sources

When a PCE receives a PCRpt on a state-sync session, it stores the LSP information into the original PCC address context (as the LSP belongs to the PCC). A PCE SHOULD maintain a single state for a particular LSP and SHOULD maintain the list of sources it learned a particular state from.

A PCEP speaker may receive a state information for a particular LSP from different sources: the PCC that owns the LSP (through a regular PCEP session) and some PCEs (through PCEP state-sync sessions). A PCEP speaker MUST always keep the freshest state in its LSP database, overriding the previously received information.

A PCE, receiving a PCRpt from a PCC, updates the state of the LSP in its LSP-DB with the new received information. When receiving a PCRpt from another PCE, a PCE SHOULD update the LSP state only if the ORIGINAL-LSP-DB-VERSION present in the PCRpt is greater than the current ORIGINAL-LSP-DB-VERSION of the stored LSP state. This ensures that a PCE never tries to update its stored LSP state with an old information. Each time a PCE updates an LSP state in its LSP-DB, it SHOULD reset the source list associated with the LSP state and SHOULD add the source speaker address in the source list. When a PCE receives a PCRpt which has an ORIGINAL-LSP-DB-VERSION (if coming from a PCE) or an LSP-DB-VERSION (if coming from the PCC) equals to the current ORIGINAL-LSP-DB-VERSION of the stored LSP state, it SHOULD add the source speaker address in the source list.

When a PCE receives a PCRpt requesting an LSP deletion from a particular source, it SHOULD remove this particular source from the list of sources associated with this LSP.

When the list of sources becomes empty for a particular LSP, the LSP state MUST be removed. This means that all the sources must send a PCRpt with R=1 for an LSP to make the PCE remove the LSP state.

3.5. Computation priority between PCEs and sub-delegation

A computation priority is necessary to ensure that a single PCE will perform the computation for all the LSPs in an association group: this will allow for a more optimized LSP placement and will prevent computation loops.

All PCEs in the network that are handling LSPs in a common LSP association group SHOULD be aware of each other including the computation priority of each PCE. Note that there is no need for PCC to be aware of this. The computation priority is a number and the PCE having the highest priority SHOULD be responsible for the computation. If several PCEs have the same priority value, their IP address SHOULD be used as a tie-breaker to provide a rank: the highest IP address has more priority. How PCEs are aware of the priority of each other is out of scope of this document, but as example learning priorities could be done through PCE discovery or local configuration.

The definition of the priority could be global so the highest priority PCE will handle all path computations or more granular, so a PCE may have highest priority for only a subset of LSPs or association-groups.

A PCEP Speaker receiving a PCRpt from a PCC with D flag set that does not have the highest computation priority, SHOULD forward the PCRpt on all state-sync sessions (as per Section 3.3) and SHOULD set D flag on the state-sync session towards the highest priority PCE, D flag will be unset to all other state-sync sessions. This behaviour is similar to the delegation behaviour handled at PCC side and is called a sub-delegation (the PCE sub-delegates the control of the LSP to another PCE). When a PCEP Speaker sub-delegates a LSP to another PCE, it loses the control on the LSP and cannot update it any more by its own decision. When a PCE receives a PCRpt with D flag set on a state-sync session, as a regular PCE, it is granted control over the LSP.

If the highest priority PCE is failing or if the state-sync session between the local PCE and the highest priority PCE failed, the local PCE MAY decide to delegate the LSP to the next highest priority PCE or to take back control on the LSP. It is a local policy decision.

When a PCE has the delegation for an LSP and needs to update this LSP, it MUST send a PCUpd message to all state-sync sessions and to

the PCC session on which it received the delegation. The D-Flag would be unset in the PCUpd for state-sync sessions where as D-Flag would be set for the PCC. In case of sub-delegation, the computing PCE will send the PCUpd only to all state-sync sessions (as it has no direct delegation from a PCC). The D-Flag would be set for the state-sync session to the PCE that sub-delegated this LSP and the D-Flag would be unset for other state-sync sessions.

The PCUpd sent over a state-sync session MUST contain the SPEAKER-IDENTITY-TLV in the LSP Object (the value used must identify the target PCC). The PLSP-ID used is the original PLSP-ID generated by the PCC and learned from the forwarded PCRpt. If a PCE receives a PCUpd on a state-sync session without the SPEAKER-IDENTITY-TLV, it MUST discard the PCUpd and MUST reply with a PCErr message using error-type=6 (Mandatory Object missing) and error-value=TBD1 (SPEAKER-IDENTITY-TLV missing).

When a PCE receives a valid PCUpd on a state-sync session, it SHOULD forward the PCUpd to the appropriate PCC (identified based on the SPEAKER-IDENTITY-TLV value) that delegated the LSP originally and SHOULD remove the SPEAKER-IDENTITY-TLV from the LSP Object. The acknowledgement of the PCUpd is done through a cascaded mechanism, and the PCC is the only responsible of triggering the acknowledgement: when the PCC receives the PCUpd from the local PCE, it acknowledges it with a PCRpt as per [RFC8231]. When receiving the new PCRpt from the PCC, the local PCE uses the defined forwarding rules on the state-sync session so the acknowledgement is relayed to the computing PCE.

A PCE SHOULD NOT compute a path using an association-group constraint if it has delegation for only a subset of LSPs in the group. In this case, an implementation MAY use a local policy on PCE to decide if PCE does not compute path at all for this set of LSP or if it can compute a path by relaxing the association-group constraint.

3.6. Passive stateful procedures

In the passive stateful PCE architecture, the PCC is responsible for triggering a path computation request using a PCReq message to its PCE. Similarly to PCRpt Message, which remains unchanged for passive mode, if a PCE receives a PCReq for an LSP and if this PCE finds that it does not have the highest computation priority of this LSP, or groups..., it MUST forward the PCReq message to the highest priority PCE over the state-sync session. When the highest priority PCE receives the PCReq, it computes the path and generates a PCRep message towards the PCE that made the request. This PCE will then forward the PCRep to the requesting PCC. The handling of LSP object

and the SPEAKER-IDENTITY-TLV in PCReq and PCRep is similar to PCRpt/PCUpd messages.

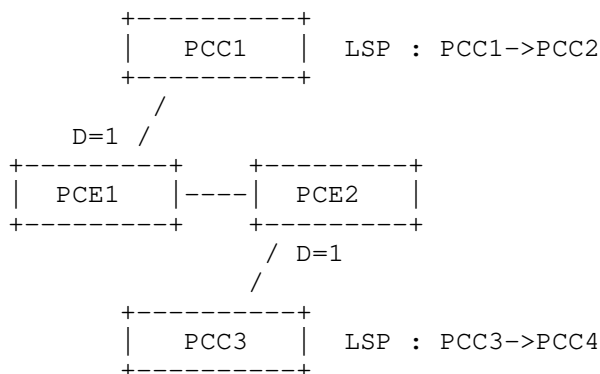
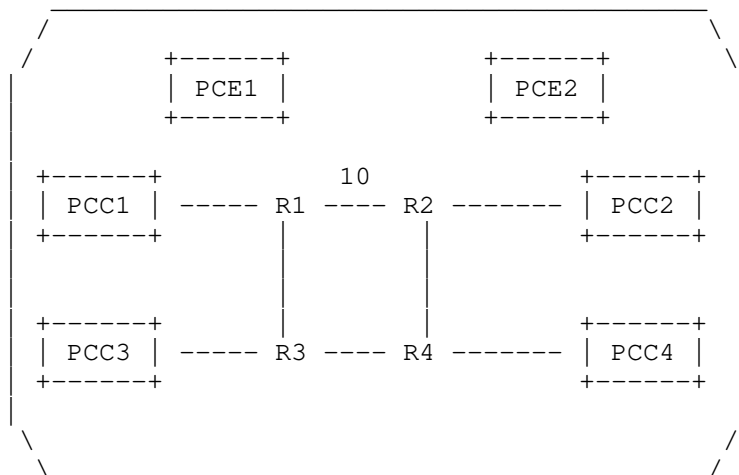
3.7. PCE initiation procedures

TBD

4. Examples

The examples in this section are for illustrative purpose to show how the behaviour of the state sync inter-PCE sessions.

4.1. Example 1



```
PCE1 computation priority 100
PCE2 computation priority 200
```

Consider the PCEP sessions as shown above, where computation priority is global for all the LSPs and link disjoint between LSPs PCC1->PCC2 and PCC3->PCC4 is required.

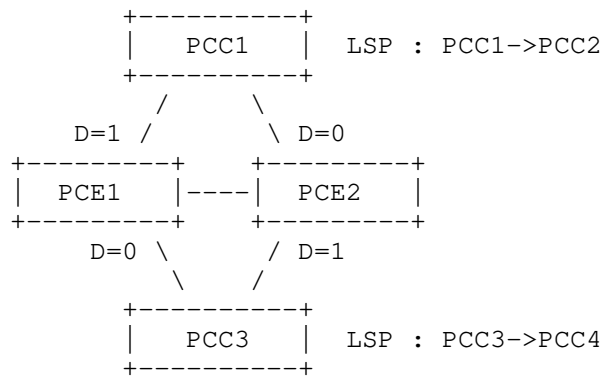
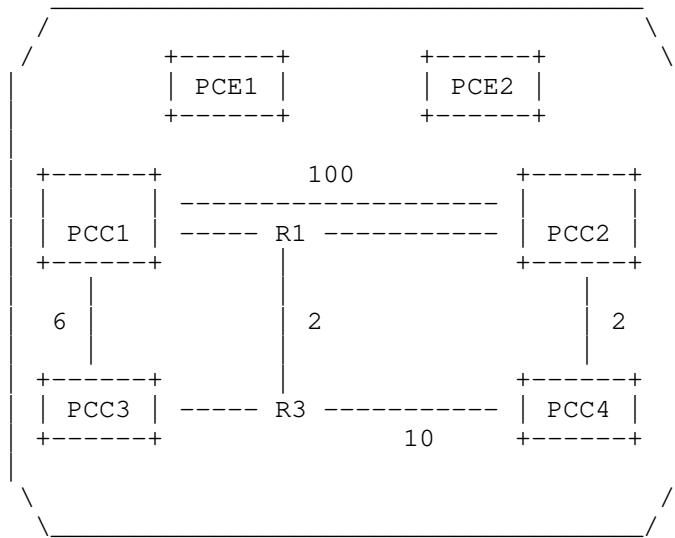
Consider the PCC1->PCC2 is configured first and PCC1 delegates the LSP to PCE1, but as PCE1 does not have the highest computation priority, it sub-delegates the LSP to PCE2 by sending a PCRpt with D=1 and including the SPEAKER-IDENTITY-TLV over the state-sync session. PCE2 receives the PCRpt and as it has delegation for this LSP, it computes the shortest path: R1->R3->R4->R2->PCC2. It then sends a PCUpd to PCE1 (including the SPEAKER-IDENTITY-TLV) with the computed ERO. PCE1 forwards the PCUpd to PCC1 (removing the SPEAKER-

IDENTITY-TLV). PCC1 acknowledges the PCUpd by a PCRpt to PCE1. PCE1 forwards the PCRpt to PCE2.

When PCC3->PCC4 is configured, PCC3 delegates the LSP to PCE2, PCE2 can compute a disjoint path as it has knowledge of both LSPs and has delegation also for both. The only solution found is to move PCC1->PCC2 LSP on another path, PCE2 can move PCC1->PCC2 as it has sub-delegation for it. It creates a new PCUpd with new ERO: R1->R2-PCC2 towards PCE1 which forwards to PCC1. PCE2 sends a PCUpd to PCC3 with the path: R3->R4->PCC4.

In this set-up, PCEs are able to find a disjoint path while without state-sync and computation priority they could not.

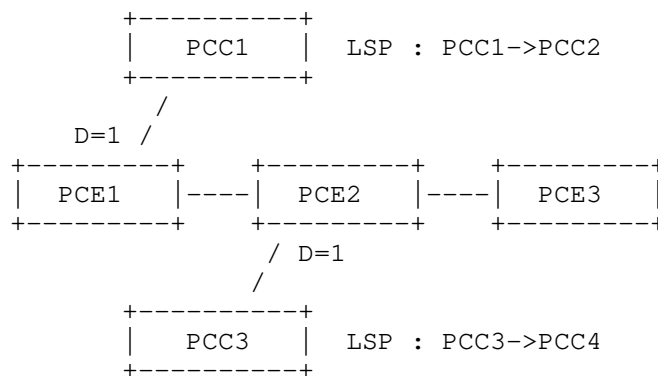
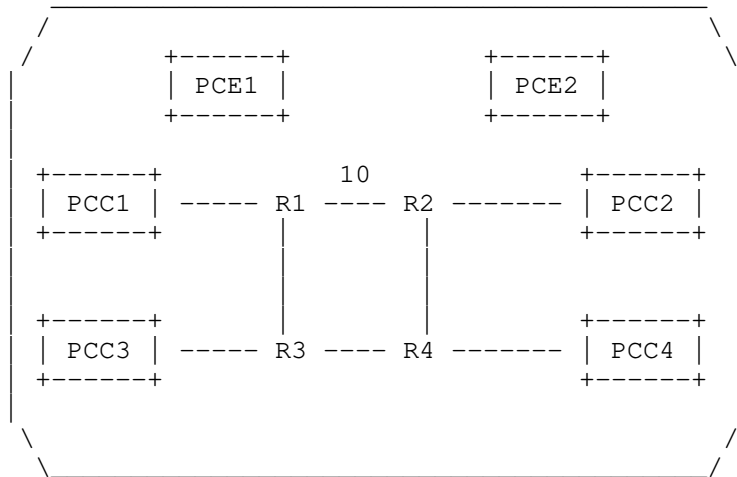
4.2. Example 2



PCE1 computation priority 200
PCE2 computation priority 100

In this example, suppose both LSPs are configured almost at the same time. PCE1 sub-delegates PCC1->PCC2 to PCE2 while PCE2 keeps delegation for PCC3->PCC4, PCE2 computes a path for PCC1->PCC2 and PCC3->PCC4 and can achieve disjointness computation easily. No computation loop happens in this case.

4.3. Example 3



PCE1 computation priority 100
PCE2 computation priority 200
PCE3 computation priority 300

With the PCEP sessions as shown above, consider the need to have link disjoint LSPs PCC1->PCC2 and PCC3->PCC4.

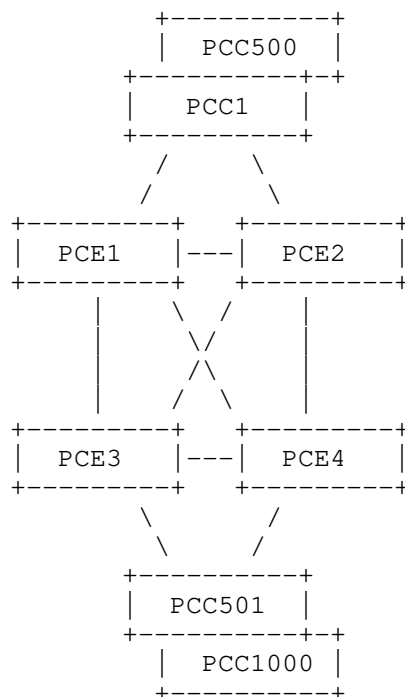
Suppose PCC1->PCC2 is configured first, PCC1 delegates the LSP to PCE1, but as PCE1 does not have the highest computation priority, it will sub-delegate the LSP to PCE2 (as it not aware of PCE3 and has no way to reach it). PCE2 cannot compute a path for PCC1->PCC2 as it does not have the highest priority and is not allowed to sub-delegate the LSP again towards PCE3 as per Section 3.

When PCC3->PCC4 is configured, PCC3 delegates the LSP to PCE2 that performs sub-delegation to PCE3. As PCE3 will have knowledge of only one LSP in the group, it cannot compute disjointness and can decide to fallback to a less constrained computation to provide a path for PCC3->PCC4. In this case, it will send a PCUpd to PCE2 that will be forwarded to PCC3.

Disjointness cannot be achieved in this scenario because of lack of state-sync session between PCE1 and PCE3, but no computation loop happens. Thus it is advised for all PCEs that support state-sync to have a full mesh sessions between each other.

5. Using Master/Slave computation and state-sync sessions to increase scaling

The Primary/Backup computation and state-sync sessions architecture can be used to increase the scaling of the PCE architecture. If the number of PCCs is really high, it may be too resource consuming for a single PCE to maintain all the PCEP sessions while at the same time performing all path computations. Using master/slave computation and state-sync sessions may allow to create groups of PCEs that manage a subset of the PCCs and perform some or no path computations. Decoupling PCEP session maintenance and computation will allow to increase scaling of the PCE architecture.



In the figure above, two groups of PCEs are created: PCE1/2 maintain PCEP sessions with PCC1 up to PCC500, while PCE3/4 maintain PCEP sessions with PCC501 up to PCC1000. A granular master/slave policy is set-up as follows to load-share computation between PCEs:

- o PCE1 has priority 200 for association ID 1 up to 300, association source 0.0.0.0. All other PCEs have a decreasing priority for those associations.
- o PCE3 has priority 200 for association ID 301 up to 500, association source 0.0.0.0. All other PCEs have a decreasing priority for those associations.

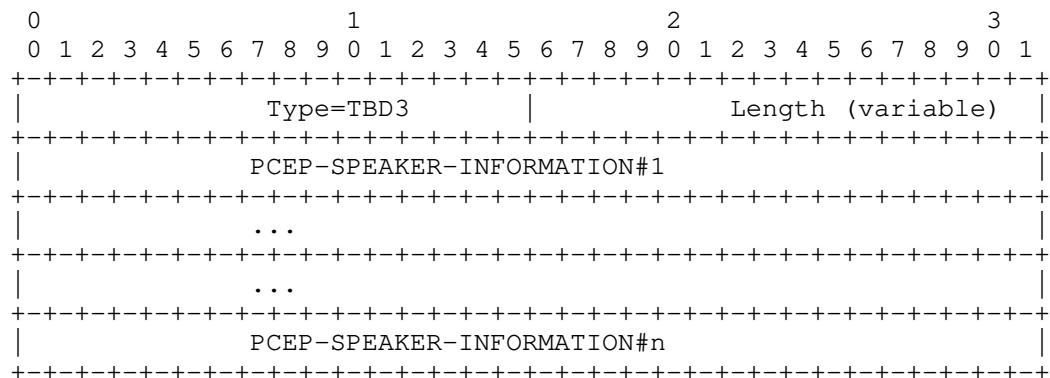
If some PCCs delegate LSPs with association ID 1 up to 300 and association source 0.0.0.0, the receiving PCE (if not PCE1) will sub-delegate the LSPs to PCE1. PCE1 becomes responsible for the computation of these LSP associations while PCE3 is responsible for the computation of another set of associations.

The procedures describe in this document could help greatly in load-sharing between a group of stateful PCEs.

6. PCEP-PATH-VECTOR-TLV

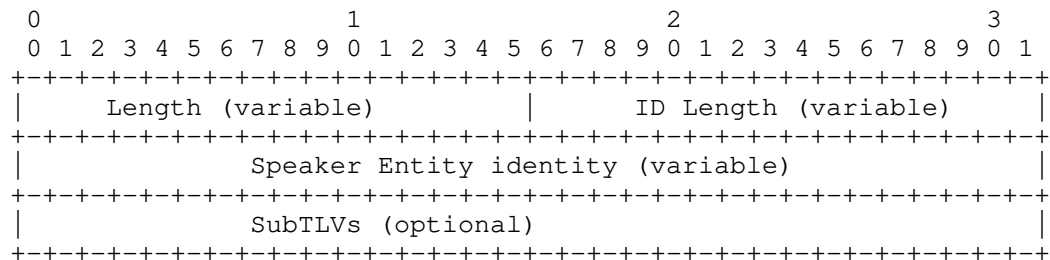
This document allows PCEP messages to be propagated among PCEP speaker. It may be useful to track informations about the propagation of the messages. One of the use case is a message loop detection mechanism, but other use cases like hop by hop information recording may also be implemented.

This document introduces the PCEP-PATH-VECTOR-TLV (type TBD3) with the following format:



The TLV format and padding rules are as per [RFC5440].

The PCEP-SPEAKER-INFORMATION field has the following format:



Length: defines the total length of the PCEP-SPEAKER-INFORMATION field.

ID Length: defines the length of the Speaker identity actual field (non-padded).

Speaker Entity identity: same possible values as the SPEAKER-IDENTIFIER-TLV. Padded with trailing zeroes to a 4-byte boundary.

The PCEP-SPEAKER-INFORMATION may also carry some optional subTLVs so each PCEP speaker can add local informations that could be recorded. This document does not define any subTLV.

The PCEP-PATH-VECTOR-TLV MAY be added in the LSP Object. Its usage is purely optional.

The list of speakers within the PCEP-PATH-VECTOR-TLV MUST be ordered. When sending a PCEP message (PCRpt, PCUpd or PCInitiate), a PCEP Speaker MAY add the PCEP-PATH-VECTOR-TLV with a PCEP-SPEAKER-INFORMATION containing its own informations. If the PCEP message sent is the result of a previously received PCEP message, and if the PCEP-PATH-VECTOR-TLV was already present in the initial message, the PCEP speaker MAY append a new PCEP-SPEAKER-INFORMATION containing its own informations.

7. Security Considerations

TBD.

8. Acknowledgements

TBD.

9. IANA Considerations

This document requests IANA actions to allocate code points for the protocol elements defined in this document.

9.1. PCEP-Error Object

IANA is requested to allocate a new Error Value for the Error Type 9.

Error-Type	Meaning	Reference
6	Mandatory Object Missing	[RFC5440]
	Error-value=TBD1: SPEAKER-IDENTITY-TLV missing	This document

9.2. PCEP TLV Type Indicators

IANA is requested to allocate new TLV Type Indicator values within the "PCEP TLV Type Indicators" sub-registry of the PCEP Numbers registry, as follows:

Value	Meaning	Reference
TBD2	ORIGINAL-LSP-DB-VERSION-TLV	This document
TBD3	PCEP-PATH-VECTOR-TLV	This document

9.3. STATEFUL-PCE-CAPABILITY TLV

IANA is requested to allocate a new bit value in the STATEFUL-PCE-CAPABILITY TLV Flag Field sub-registry.

Bit	Description	Reference
TBD	INTER-PCE-CAPABILITY	This document

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [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>>.
- [RFC8232] Crabbe, E., Minei, I., Medved, J., Varga, R., Zhang, X., and D. Dhody, "Optimizations of Label Switched Path State Synchronization Procedures for a Stateful PCE", RFC 8232, DOI 10.17487/RFC8232, September 2017, <<https://www.rfc-editor.org/info/rfc8232>>.

10.2. Informative References

- [I-D.ietf-pce-association-diversity] Litkowski, S., Sivabalan, S., Barth, C., and M. Negi, "Path Computation Element Communication Protocol (PCEP) Extension for LSP Diversity Constraint Signaling", draft-ietf-pce-association-diversity-08 (work in progress), July 2019.

- [I-D.ietf-pce-stateful-hpce]
Dhody, D., Lee, Y., Ceccarelli, D., Shin, J., and D. King,
"Hierarchical Stateful Path Computation Element (PCE).",
draft-ietf-pce-stateful-hpce-11 (work in progress), July
2019.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation
Element (PCE)-Based Architecture", RFC 4655,
DOI 10.17487/RFC4655, August 2006,
<<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the
Path Computation Element Architecture to the Determination
of a Sequence of Domains in MPLS and GMPLS", RFC 6805,
DOI 10.17487/RFC6805, November 2012,
<<https://www.rfc-editor.org/info/rfc6805>>.
- [RFC7399] Farrel, A. and D. King, "Unanswered Questions in the Path
Computation Element Architecture", RFC 7399,
DOI 10.17487/RFC7399, October 2014,
<<https://www.rfc-editor.org/info/rfc7399>>.
- [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>>.
- [RFC8051] Zhang, X., Ed. and I. Minei, Ed., "Applicability of a
Stateful Path Computation Element (PCE)", RFC 8051,
DOI 10.17487/RFC8051, January 2017,
<<https://www.rfc-editor.org/info/rfc8051>>.

Appendix A. Contributors

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

Email: dhruv.ietf@gmail.com

Authors' Addresses

Stephane Litkowski
Orange

Email: stephane.litkowski@orange.com

Siva Sivabalan
Cisco

Email: msiva@cisco.com

Cheng Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: chengli13@huawei.com

Haomian Zheng
Huawei Technologies
H1-1-A043S Huawei Industrial Base, Songshanhu
Dongguan, Guangdong 523808
China

Email: zhenghaomian@huawei.com

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

S. Litkowski
Cisco
S. Sivabalan
Ciena Corporation
C. Li
H. Zheng
Huawei Technologies
February 22, 2021

Inter Stateful Path Computation Element (PCE) Communication Procedures.
draft-litkowski-pce-state-sync-10

Abstract

The Path Computation Element Communication Protocol (PCEP) provides mechanisms for Path Computation Elements (PCEs) to perform path computation in response to a Path Computation Client (PCC) request. The Stateful PCE extensions allow stateful control of Multi-Protocol Label Switching (MPLS) Traffic Engineering (TE) Label Switched Paths (LSPs) using PCEP.

A Path Computation Client (PCC) can synchronize an LSP state information to a Stateful Path Computation Element (PCE). The stateful PCE extension allows a redundancy scenario where a PCC can have redundant PCEP sessions towards multiple PCEs. In such a case, a PCC gives control of a LSP to only a single PCE, and only one PCE is responsible for path computation for this delegated LSP.

There are some use cases, where an inter-PCE stateful communication can bring additional resiliency in the design, for instance when some PCC-PCE session fails. The inter-PCE stateful communication may also provide a faster update of the LSP states when such an event occurs. Finally, when, in a redundant PCE scenario, there is a need to compute a set of paths that are part of a group (so there is a dependency between the paths), there may be some cases where the computation of all paths in the group is not handled by the same PCE: this situation is called a split-brain. This split-brain scenario may lead to computation loops between PCEs or suboptimal path computation.

This document describes the procedures to allow a stateful communication between PCEs for various use-cases and also the procedures to prevent computations loops.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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 <https://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 August 26, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction and Problem Statement	3
1.1. Reporting LSP Changes	4
1.2. Split-Brain	5
1.3. Applicability to H-PCE	12
2. Proposed solution	12
2.1. State-sync session	12

2.2. Primary/Secondary relationship between PCE	14
3. Procedures and Protocol Extensions	14
3.1. Opening a state-sync session	14
3.1.1. Capability Advertisement	14
3.2. State synchronization	15
3.3. Incremental updates and report forwarding rules	16
3.4. Maintaining LSP states from different sources	17
3.5. Computation priority between PCEs and sub-delegation	18
3.6. Passive stateful procedures	19
3.7. PCE initiation procedures	20
4. Examples	20
4.1. Example 1	20
4.2. Example 2	22
4.3. Example 3	24
5. Using Primary/Secondary Computation and State-sync Sessions to increase Scaling	25
6. PCEP-PATH-VECTOR TLV	27
7. Security Considerations	28
8. Acknowledgements	28
9. IANA Considerations	28
9.1. PCEP-Error Object	28
9.2. PCEP TLV Type Indicators	29
9.3. STATEFUL-PCE-CAPABILITY TLV	29
10. References	29
10.1. Normative References	29
10.2. Informative References	30
Appendix A. Contributors	31
Authors' Addresses	31

1. Introduction and Problem Statement

The Path Computation Element communication Protocol (PCEP) [RFC5440] provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients' (PCCs) requests.

A stateful PCE [RFC8231] is capable of considering, for the purposes of path computation, not only the network state in terms of links and nodes (referred to as the Traffic Engineering Database or TED) but also the status of active services (previously computed paths, and currently reserved resources, stored in the Label Switched Paths Database (LSP-DB).

[RFC8051] describes general considerations for a stateful PCE deployment and examines its applicability and benefits, as well as its challenges and limitations through a number of use cases.

The examples in this section are for illustrative purpose to showcase the need for inter-PCE stateful PCEP sessions.

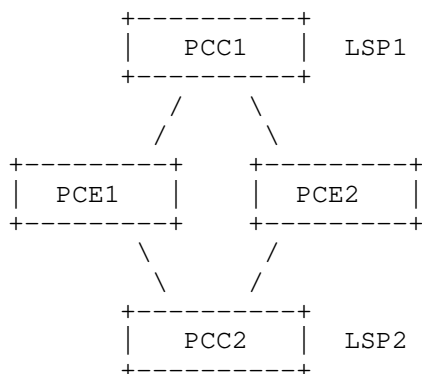
1.1. Reporting LSP Changes

When using a stateful PCE ([RFC8231]), a PCC can synchronize an LSP state information to the stateful PCE. If the PCC grants the control of the LSP to the PCE (called delegation [RFC8231]), the PCE can update the LSP parameters at any time.

In a multi PCE deployment (redundancy, loadbalancing...), with the current specification defined in [RFC8231], when a PCE makes an update, it is the PCC that is in charge of reporting the LSP status to all PCEs with LSP parameter change which brings additional hops and delays in notifying the overall network of the LSP parameter change.

This delay may affect the reaction time of the other PCEs if they need to take action after being notified of the LSP parameter change.

Apart from the synchronization from the PCC, it is also useful if there is a synchronization mechanism between the stateful PCEs. As stateful PCE make changes to its delegated LSPs, these changes (pending LSPs and the sticky resources [RFC7399]) can be synchronized immediately to the other PCEs.



In the figure above, we consider a load-balanced PCE architecture, so PCE1 is responsible to compute paths for PCC1 and PCE2 is responsible to compute paths for PCC2. When PCE1 triggers an LSP update for LSP1, it sends a PCUpd message to PCC1 containing the new parameters for LSP1. PCC1 will take the parameters into account and will send a PCRppt message to PCE1 and PCE2 reflecting the changes. PCE2 will so

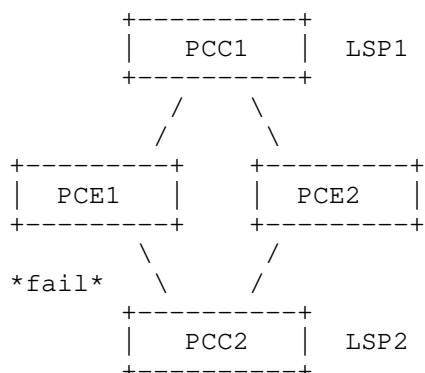
be notified of the change only after receiving the PCRpt message from PCC1.

Let's consider that the LSP1 parameters changed in such a way that LSP1 will take over resources from LSP2 with a higher priority. After receiving the report from PCC1, PCE2 will therefore try to find a new path for LSP2. If we consider that there is a round trip delay of about 150 milliseconds (ms) between the PCEs and PCC1 and a round trip delay of 10 ms between the two PCEs it will take more than 150 ms for PCE2 to be notified of the change.

Adding a PCEP session between PCE1 and PCE2 may allow to reduce the synchronization time, so PCE2 can react more quickly by taking the pending LSPs and attached resources into account during path computation and re-optimization.

1.2. Split-Brain

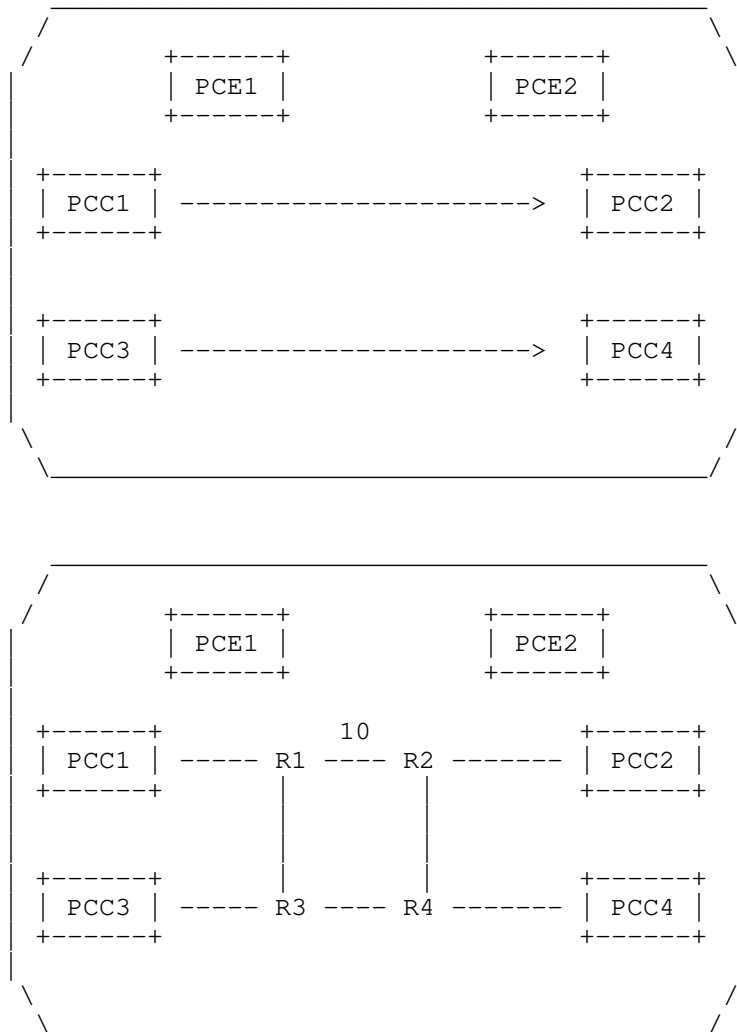
In a resiliency case, a PCC has redundant PCEP sessions towards multiple PCEs. In such a case, a PCC gives control on an LSP to a single PCE only, and only this PCE is responsible for the path computation for the delegated LSP: the PCC achieves this by setting the D flag only towards the active PCE [RFC8231] selected for delegation. The election of the active PCE to delegate an LSP is controlled by each PCC. The PCC usually elects the active PCE by a local configured policy (by setting a priority). Upon PCEP session failure, or active PCE failure, PCC may decide to elect a new active PCE by sending new PCRpt message with D flag set to this new active PCE. When the failed PCE or PCEP session comes back online, it will be up to the implementation to do preemption. Doing preemption may lead to some disruption on the existing path if path results from both PCEs are not exactly the same. By considering a network with multiple PCCs and implementing multiple stateful PCEs for redundancy purpose, there is no guarantee that at any time all the PCCs delegate their LSPs to the same PCE.



In the example above, we consider that by configuration, both PCCs will firstly delegate their LSPs to PCE1. So, PCE1 is responsible for computing a path for both LSP1 and LSP2. If the PCEP session between PCC2 and PCE1 fails, PCC2 will delegate LSP2 to PCE2. So PCE1 becomes responsible only for LSP1 path computation while PCE2 is responsible for the path computation of LSP2. When the PCC2-PCE1 session is back online, PCC2 will keep using PCE2 as active PCE (consider no preemption in this example). So the result is a permanent situation where each PCE is responsible for a subset of path computation.

This situation is called a split-brain scenario, as there are multiple computation brains running at the same time while a central computation unit was required in some deployments/use cases.

Further, there are use cases where a particular LSP path computation is linked to another LSP path computation: the most common use case is path disjointness (see [RFC8800]). The set of LSPs that are dependent to each other may start from a different head-end.



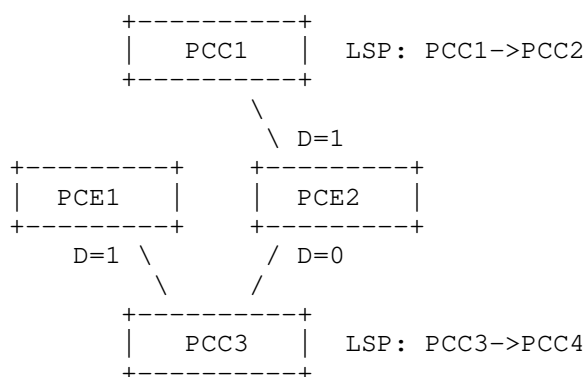
In the figure above, the requirement is to create two link-disjoint LSPs: PCC1->PCC2 and PCC3->PCC4. In the topology, all links cost metric is set to 1 except for the link 'R1-R2' which has a metric of 10. The PCEs are responsible for the path computation and PCE1 is the active primary PCE for all PCCs in the nominal case.

Scenario 1:

In the normal case (PCE1 as active primary PCE), consider that PCC1->PCC2 LSP is configured first with the link disjointness constraint, PCE1 sends a PCUpd message to PCC1 with the ERO: R1->R3->R4->R2->PCC2 (shortest path). PCC1 signals and installs the path. When PCC3->PCC4 is configured, the PCEs already knows the path of PCC1->PCC2 and can compute a link-disjoint path: the solution requires to move PCC1->PCC2 onto a new path to let room for the new LSP. PCE1 sends a PCUpd message to PCC1 with the new ERO: R1->R2->PCC2 and a PCUpd to PCC3 with the following ERO: R3->R4->PCC4. In the normal case, there is no issue for PCE1 to compute a link-disjoint path.

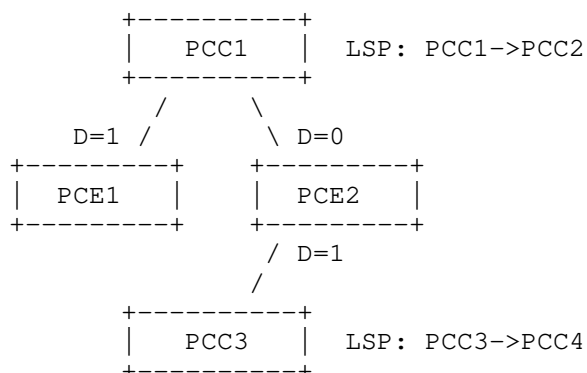
Scenario 2:

Consider that PCC1 lost its PCEP session with PCE1 (all other PCEP sessions are UP). PCC1 delegates its LSP to PCE2.



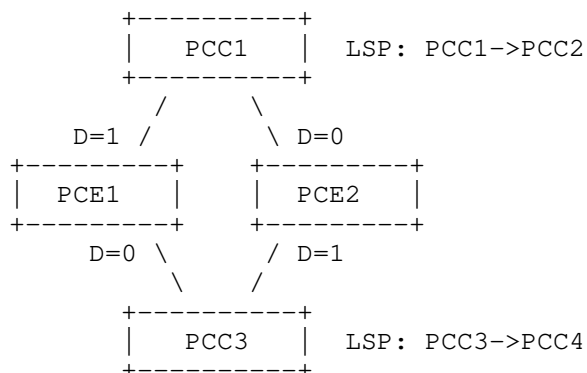
Consider that the PCC1->PCC2 LSP is configured first with the link disjointness constraint, PCE2 (which is the new active primary PCE for PCC1) sends a PCUpd message to PCC1 with the ERO: R1->R3->R4->R2->PCC2 (shortest path). When PCC3->PCC4 is configured, PCE1 is not aware of LSPs from PCC1 any more, so it cannot compute a disjoint path for PCC3->PCC4 and will send a PCUpd message to PCC3 with the shortest path ERO: R3->R4->PCC4. When PCC3->PCC4 LSP will be reported to PCE2 by PCC3, PCE2 will ensure disjointness computation and will correctly move PCC1->PCC2 (as it owns delegation for this LSP) on the following path: R1->R2->PCC2. With this sequence of event and these PCEP sessions, disjointness is ensured.

Scenario 3:



Consider the above PCEP sessions and the PCC1->PCC2 LSP is configured first with the link disjointness constraint, PCE1 computes the shortest path as it is the only LSP in the disjoint association group that it is aware of: R1->R3->R4->R2->PCC2 (shortest path). When PCC3->PCC4 is configured, PCE2 must compute a disjoint path for this LSP. The only solution found is to move PCC1->PCC2 LSP on another path, but PCE2 cannot do it as it does not have delegation for this LSP. In this set-up, PCEs are not able to find a disjoint path.

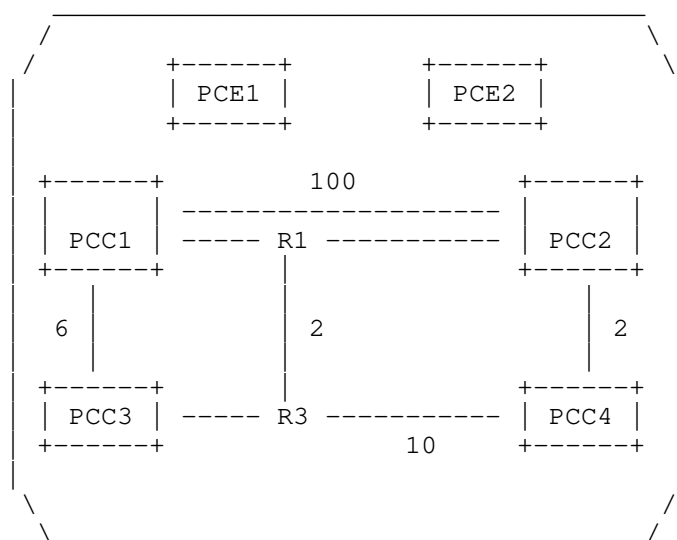
Scenario 4:



Consider the above PCEP sessions and that PCEs are configured to fall-back to the shortest path if disjointness cannot be found as described in [RFC8800]. The PCC1->PCC2 LSP is configured first, PCE1 computes the shortest path as it is the only LSP in the disjoint association group that it is aware of: R1->R3->R4->R2->PCC2 (shortest path). When PCC3->PCC4 is configured, PCE2 must compute a disjoint path for this LSP. The only solution found is to move PCC1->PCC2 LSP on another path, but PCE2 cannot do it as it does not have delegation

for this LSP. PCE2 then provides the shortest path for PCC3->PCC4: R3->R4->PCC4. When PCC3 receives the ERO, it reports it back to both PCEs. When PCE1 becomes aware of the PCC3->PCC4 path, it recomputes the constrained shortest path first (CSPF) algorithm and provides a new path for PCC1->PCC2: R1->R2->PCC2. The new path is reported back to all PCEs by PCC1. PCE2 recomputes also CSPF to take into account the new reported path. The new computation does not lead to any path update.

Scenario 5:

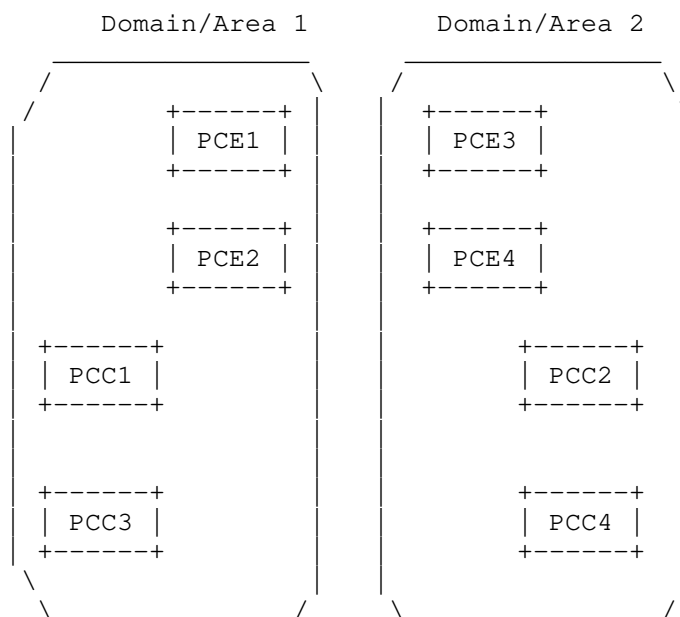


Now, consider a new network topology with the same PCEP sessions as the previous example. Suppose that both LSPs are configured almost at the same time. PCE1 will compute a path for PCC1->PCC2 while PCE2 will compute a path for PCC3->PCC4. As each PCE is not aware of the path of the second LSP in the association group (not reported yet), each PCE is computing the shortest path for the LSP. PCE1 computes ERO: R1->PCC2 for PCC1->PCC2 and PCE2 computes ERO: R3->R1->PCC2->PCC4 for PCC3->PCC4. When these shortest paths will be reported to each PCE. Each PCE will recompute disjointness. PCE1 will provide a new path for PCC1->PCC2 with ERO: PCC1->PCC2. PCE2 will provide also a new path for PCC3->PCC4 with ERO: R3->PCC4. When those new paths will be reported to both PCEs, this will trigger CSFP again. PCE1 will provide a new more optimal path for PCC1->PCC2 with ERO: R1->PCC2 and PCE2 will also provide a more optimal path for PCC3->PCC4 with ERO: R3->R1->PCC2->PCC4. So we come back to the

initial state. When those paths will be reported to both PCEs, this will trigger CSPF again. An infinite loop of CSPF computation is then happening with a permanent flap of paths because of the split-brain situation.

This permanent computation loop comes from the inconsistency between the state of the LSPs as seen by each PCE due to the split-brain: each PCE is trying to modify at the same time its delegated path based on the last received path information which de facto invalidates this received path information.

Scenario 6: multi-domain



In the example above, suppose that the disjoint LSPs from PCC1 to PCC2 and from PCC4 to PCC3 are created. All the PCEs have the knowledge of both domain topologies (e.g. using BGP-LS [RFC7752]). For operation/management reasons, each domain uses its own group of redundant PCEs. PCE1/PCE2 in domain 1 have PCEP sessions with PCC1 and PCC3 while PCE3/PCE4 in domain 2 have PCEP sessions with PCC2 and PCC4. As PCE1/2 does not know about LSPs from PCC2/4 and PCE3/4 do not know about LSPs from PCC1/3, there is no possibility to compute the disjointness constraint. This scenario can also be seen as a split-brain scenario. This multi-domain architecture (with multiple groups of PCEs) can also be used in a single domain, where an

operator wants to limit the failure domain by creating multiple groups of PCEs maintaining a subset of PCCs. As for the multi-domain example, there will be no possibility to compute the disjoint path starting from head-ends managed by different PCE groups.

In this document, we propose a solution that addresses the possibility to compute LSP association based constraints (like disjointness) in split-brain scenarios while preventing computation loops.

1.3. Applicability to H-PCE

[RFC8751] describes general considerations and use cases for the deployment of Stateful PCE(s) using the Hierarchical PCE [RFC6805] architecture. In this architecture, there is a clear need to communicate between a child stateful PCE and a parent stateful PCE. The procedures and extensions as described in Section 3 are equally applicable to the H-PCE scenario.

2. Proposed solution

Our solution is based on :

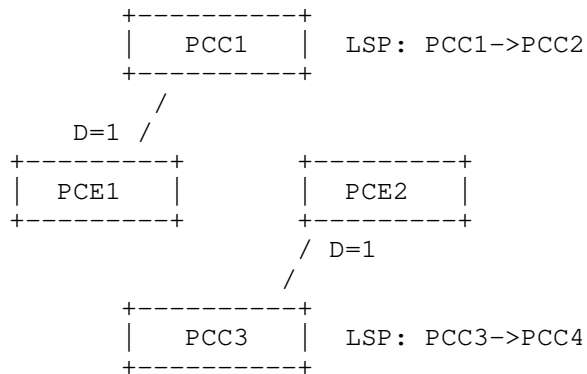
- o The creation of the inter-PCE stateful PCEP session with specific procedures.
- o A Primary/Secondary relationship between PCEs.

2.1. State-sync session

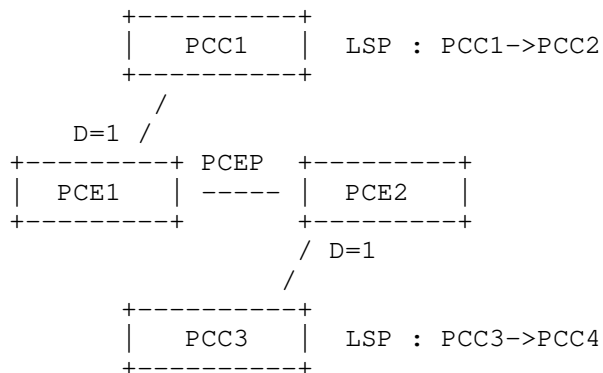
This document proposes to set-up a PCEP session between the stateful PCEs. Creating such a session is already authorized by multiple scenarios like the one described in [RFC4655] (multiple PCEs that are handling part of the path computation) and [RFC6805] (hierarchical PCE) but was only focused on the stateless PCEP sessions. As stateful PCE brings additional features (LSP state synchronization, path update, delegation, ...), thus some new behaviors need to be defined.

This inter-PCE PCEP session will allow the exchange of LSP states between PCEs that would help some scenarios where PCEP sessions are lost between PCC and PCE. This inter-PCE PCEP session is henceforth called a state-sync session.

For example, in the scenario below, there is no possibility to compute disjointness as there is no PCE that is aware of both LSPs.



If we add a state-sync session, PCE1 will be able to do state synchronization via PCRpt messages for its LSP to PCE2 and PCE2 will do the same. All the PCEs will be aware of all LSPs even if a PCC->PCE session is down. PCEs will then be able to compute disjoint paths.



The procedures associated with this state-sync session are defined in Section 3.

By just adding this state-sync session, it does not ensure that a path with LSP association based constraints can always be computed and does not prevent the computation loop, but it increases resiliency and ensures that PCEs will have the state information for all LSPs. Also, this session will allow for a PCE to update the other PCEs providing a faster synchronization mechanism than relying on PCCs only.

2.2. Primary/Secondary relationship between PCE

As seen in Section 1, performing a path computation in a split-brain scenario (multiple PCEs responsible for computation) may provide a non-optimal LSP placement, no path, or computation loops. To provide the best efficiency, an LSP association constraint-based computation requires that a single PCE performs the path computation for all LSPs in the association group. Note that, it could be all LSPs belonging to a particular association group, or all LSPs from a particular PCC, or all LSPs in the network that need to be delegated to a single PCE based on the deployment scenarios.

This document proposes to add a priority mechanism between PCEs to elect a single computing PCE. Using this priority mechanism, PCEs can agree on the PCE that will be responsible for the computation for a particular association group, or set of LSPs. The priority could be set per association, per PCC, or for all LSPs. How this priority is set or advertised is out of the scope of this document. The rest of the text considers the association group as an example.

When a single PCE is performing the computation for a particular association group, no computation loop can happen and an optimal placement will be provided. The other PCEs will only act as state collectors and forwarders.

In the scenario described in Section 2.1, PCE1 and PCE2 will decide that PCE1 will be responsible for the path computation of both LSPs. If we first configure PCC1->PCC2, PCE1 computes the shortest path at it is the only LSP in the disjoint-group that it is aware of: R1->R3->R4->R2->PCC2 (shortest path). When PCC3->PCC4 is configured, PCE2 will not perform computation even if it has delegation but forwards the delegation via PCRpt message to PCE1 through the state-sync session. PCE1 will then perform disjointness computation and will move PCC1->PCC2 onto R1->R2->PCC2 and provides an ERO to PCE2 for PCC3->PCC4: R3->R4->PCC4. The PCE2 will further update the PCC3 with the new path.

3. Procedures and Protocol Extensions

3.1. Opening a state-sync session

3.1.1. Capability Advertisement

A PCE indicates its support of state-sync procedures during the PCEP Initialization phase [RFC5440]. The OPEN object in the Open message MUST contain the "Stateful PCE Capability" TLV defined in [RFC8231]. A new P (INTER-PCE-CAPABILITY) flag is introduced to indicate the support of state-sync.

This document adds a new bit in the Flags field with :

P (INTER-PCE-CAPABILITY - 1 bit - TBD4): If set to 1 by a PCEP Speaker, the PCEP speaker indicates that the session MUST follow the state-sync procedures as described in this document. The P bit MUST be set by both speakers: if a PCEP Speaker receives a STATEFUL-PCE-CAPABILITY TLV with P=0 while it advertised P=1 or if both set P flag to 0, the session SHOULD be set-up but the state-sync procedures MUST NOT be applied on this session.

The U flag [RFC8231] MUST be set when sending the STATEFUL-PCE-CAPABILITY TLV with the P flag set. In case the U flag is not set along with the P flag, the state sync capability is not enabled and it is considered as if the P flag is not set. The S flag MAY be set if optimized synchronization is required as per [RFC8232].

3.2. State synchronization

When the state sync capability has been negotiated between stateful PCEs, each PCEP speaker will behave as a PCE and as a PCC at the same time regarding the state synchronization as defined in [RFC8231]. This means that each PCEP Speaker:

- o MUST send a PCRpt message towards its neighbor with S flag set for each LSP in its LSP database learned from a PCC. (PCC role)
- o MUST send the End Of Synchronization Marker towards its neighbor when all LSPs have been reported. (PCC role)
- o MUST wait for the LSP synchronization from its neighbor to end (receiving an End Of Synchronization Marker). (PCE role)

The process of synchronization runs in parallel on each PCE (with no defined order).

The optimized state synchronization procedures MAY be used, as defined in [RFC8232].

When a PCEP Speaker sends a PCRpt on a state-sync session, it MUST add the SPEAKER-IDENTITY-TLV (defined in [RFC8232]) in the LSP Object, the value used will refer to the 'owner' PCC of the LSP. If a PCEP Speaker receives a PCRpt on a state-sync session without this TLV, it MUST discard the PCRpt message and it MUST reply with a PCErr message using error-type=6 (Mandatory Object missing) and error-value=TBD1 (SPEAKER-IDENTITY-TLV missing).

3.3. Incremental updates and report forwarding rules

During the life of an LSP, its state may change (path, constraints, operational state...) and a PCC will advertise a new PCRpt to the PCE for each such change.

When propagating LSP state changes from a PCE to other PCEs, it is mandatory to ensure that a PCE always uses the freshest state coming from the PCC.

When a PCE receives a new PCRpt from a PCC with the LSP-DB-VERSION, the PCE MUST forward the PCRpt to all its state-sync sessions and MUST add the appropriate SPEAKER-IDENTITY-TLV in the PCRpt. In addition, it MUST add a new ORIGINAL-LSP-DB-VERSION TLV (described below). The ORIGINAL-LSP-DB-VERSION contains the LSP-DB-VERSION coming from the PCC.

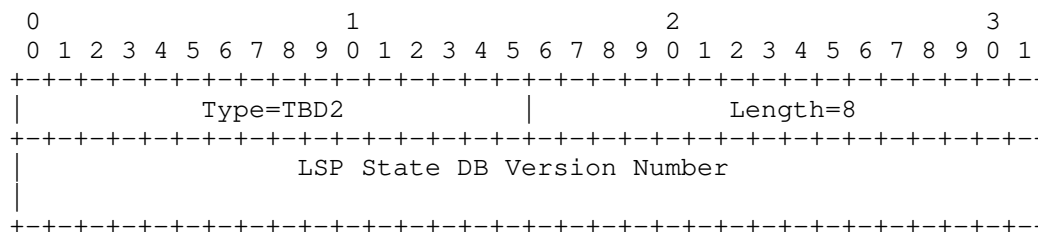
When a PCE receives a new PCRpt from a PCC without the LSP-DB-VERSION, it SHOULD NOT forward the PCRpt on any state-sync sessions and log such an event on the first occurrence.

When a PCE receives a new PCRpt from a PCC with the R flag (Remove) set and an LSP-DB-VERSION TLV, the PCE MUST forward the PCRpt to all its state-sync sessions keeping the R flag set (Remove) and MUST add the appropriate SPEAKER-IDENTITY-TLV and ORIGINAL-LSP-DB-VERSION TLV in the PCRpt message.

When a PCE receives a PCRpt from a state-sync session, it MUST NOT forward the PCRpt to other state-sync sessions. This helps to prevent message loops between PCEs. As a consequence, a full mesh of PCEP sessions between PCEs are REQUIRED.

When a PCRpt is forwarded, all the original objects and values are kept. As an example, the PLSP-ID used in the forwarded PCRpt will be the same as the original one used by the PCC. Thus an implementation supporting this document MUST consider SPEAKER-IDENTITY-TLV and PLSP-ID together to uniquely identify an LSP on the state-sync session.

The ORIGINAL-LSP-DB-VERSION TLV is encoded as follows and MUST always contain the LSP-DB-VERSION received from the owner PCC of the LSP:



Using the ORIGINAL-LSP-DB-VERSION TLV allows a PCE to keep using optimized synchronization ([RFC8232]) with another PCE. In such a case, the PCE will send a PCRpt to another PCE with both ORIGINAL-LSP-DB-VERSION TLV and LSP-DB-VERSION TLV. The ORIGINAL-LSP-DB-VERSION TLV will contain the version number as allocated by the PCC while the LSP-DB-VERSION will contain the version number allocated by the local PCE.

3.4. Maintaining LSP states from different sources

When a PCE receives a PCRpt on a state-sync session, it stores the LSP information into the original PCC address context (as the LSP belongs to the PCC). A PCE SHOULD maintain a single state for a particular LSP and SHOULD maintain the list of sources it learned a particular state from.

A PCEP speaker may receive state information for a particular LSP from different sources: the PCC that owns the LSP (through a regular PCEP session) and some PCEs (through PCEP state-sync sessions). A PCEP speaker MUST always keep the freshest state in its LSP database, overriding the previously received information.

A PCE, receiving a PCRpt from a PCC, updates the state of the LSP in its LSP-DB with the newly received information. When receiving a PCRpt from another PCE, a PCE SHOULD update the LSP state only if the ORIGINAL-LSP-DB-VERSION present in the PCRpt is greater than the current ORIGINAL-LSP-DB-VERSION of the stored LSP state. This ensures that a PCE never tries to update its stored LSP state with an old information. Each time a PCE updates an LSP state in its LSP-DB, it SHOULD reset the source list associated with the LSP state and SHOULD add the source speaker address in the source list. When a PCE receives a PCRpt which has an ORIGINAL-LSP-DB-VERSION (if coming from a PCE) or an LSP-DB-VERSION (if coming from the PCC) equals to the current ORIGINAL-LSP-DB-VERSION of the stored LSP state, it SHOULD add the source speaker address in the source list.

When a PCE receives a PCRpt requesting an LSP deletion from a particular source, it SHOULD remove this particular source from the list of sources associated with this LSP.

When the list of sources becomes empty for a particular LSP, the LSP state MUST be removed. This means that all the sources must send a PCRpt with R=1 for an LSP to make the PCE remove the LSP state.

3.5. Computation priority between PCEs and sub-delegation

A computation priority is necessary to ensure that a single PCE will perform the computation for all the LSPs in an association group: this will allow for a more optimized LSP placement and will prevent computation loops.

All PCEs in the network that are handling LSPs in a common LSP association group SHOULD be aware of each other including the computation priority of each PCE. Note that there is no need for PCC to be aware of this. The computation priority is a number and the PCE having the highest priority SHOULD be responsible for the computation. If several PCEs have the same priority value, their IP address SHOULD be used as a tie-breaker to provide a rank: the highest IP address has more priority. How PCEs are aware of the priority of each other is out of the scope of this document, but as example learning priorities could be done through PCE discovery or local configuration.

The definition of the priority could be global so the highest priority PCE will handle all path computations or more granular, so a PCE may have the highest priority for only a subset of LSPs or association-groups.

A PCEP Speaker receiving a PCRpt from a PCC with the D flag set that does not have the highest computation priority, SHOULD forward the PCRpt on all state-sync sessions (as per Section 3.3) and SHOULD set D flag on the state-sync session towards the highest priority PCE, D flag will be unset to all other state-sync sessions. This behavior is similar to the delegation behavior handled at the PCC side and is called a sub-delegation (the PCE sub-delegates the control of the LSP to another PCE). When a PCEP Speaker sub-delegates an LSP to another PCE, it loose control of the LSP and cannot update it anymore by its own decision. When a PCE receives a PCRpt with D flag set on a state-sync session, as a regular PCE, it is granted control over the LSP.

If the highest priority PCE is failing or if the state-sync session between the local PCE and the highest priority PCE failed, the local PCE MAY decide to delegate the LSP to the next highest priority PCE or to take back control of the LSP. It is a local policy decision.

When a PCE has the delegation for an LSP and needs to update this LSP, it MUST send a PCUpd message to all state-sync sessions and to

the PCC session on which it received the delegation. The D-Flag would be unset in the PCUpd for state-sync sessions whereas the D-Flag would be set for the PCC. In the case of sub-delegation, the computing PCE will send the PCUpd only to all state-sync sessions (as it has no direct delegation from a PCC). The D-Flag would be set for the state-sync session to the PCE that sub-delegated this LSP and the D-Flag would be unset for other state-sync sessions.

The PCUpd sent over a state-sync session MUST contain the SPEAKER-IDENTITY-TLV in the LSP Object (the value used must identify the target PCC). The PLSP-ID used is the original PLSP-ID generated by the PCC and learned from the forwarded PCRpt. If a PCE receives a PCUpd on a state-sync session without the SPEAKER-IDENTITY-TLV, it MUST discard the PCUpd and MUST reply with a PCErr message using error-type=6 (Mandatory Object missing) and error-value=TBD1 (SPEAKER-IDENTITY-TLV missing).

When a PCE receives a valid PCUpd on a state-sync session, it SHOULD forward the PCUpd to the appropriate PCC (identified based on the SPEAKER-IDENTITY-TLV value) that delegated the LSP originally and SHOULD remove the SPEAKER-IDENTITY-TLV from the LSP Object. The acknowledgment of the PCUpd is done through a cascaded mechanism, and the PCC is the only responsible for triggering the acknowledgment: when the PCC receives the PCUpd from the local PCE, it acknowledges it with a PCRpt as per [RFC8231]. When receiving the new PCRpt from the PCC, the local PCE uses the defined forwarding rules on the state-sync session so the acknowledgment is relayed to the computing PCE.

A PCE SHOULD NOT compute a path using an association-group constraint if it has delegation for only a subset of LSPs in the group. In this case, an implementation MAY use a local policy on PCE to decide if PCE does not compute path at all for this set of LSP or if it can compute a path by relaxing the association-group constraint.

3.6. Passive stateful procedures

In the passive stateful PCE architecture, the PCC is responsible for triggering a path computation request using a PCReq message to its PCE. Similarly to PCRpt Message, which remains unchanged for passive mode, if a PCE receives a PCReq for an LSP and if this PCE finds that it does not have the highest computation priority of this LSP, or groups..., it MUST forward the PCReq message to the highest priority PCE over the state-sync session. When the highest priority PCE receives the PCReq, it computes the path and generates a PCRep message towards the PCE that made the request. This PCE will then forward the PCRep to the requesting PCC. The handling of LSP object

and the SPEAKER-IDENTITY-TLV in PCReq and PCRep is similar to PCRpt/PCUpd messages.

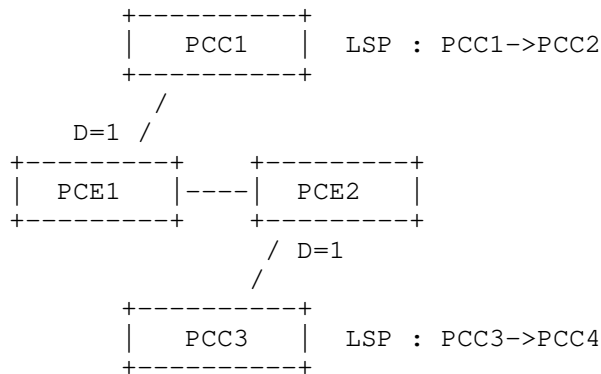
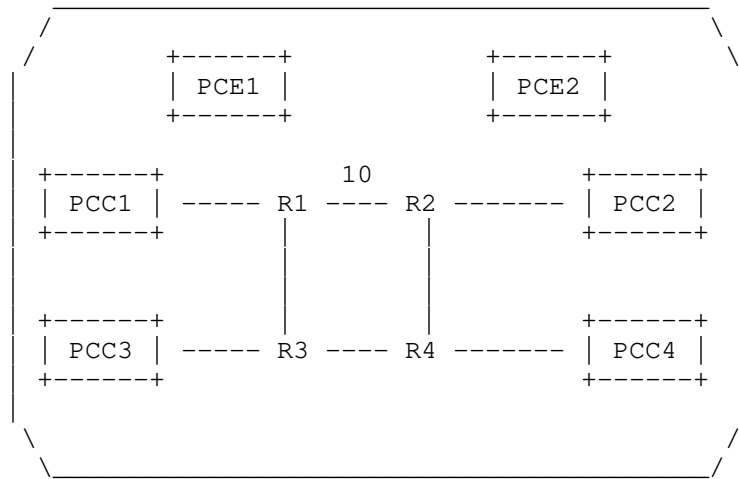
3.7. PCE initiation procedures

It is possible that a PCE does not have a PCEP session with the headend to initiate a LSP as per [RFC8281]. A PCE could send the PCInitiate message on the state-sync sessions to other PCE to request it to create a PCE-Initiated LSP on its behalf. If the PCE is able to initiate the LSP it would report it on the state-sync session via PCRpt message. If the PCE does not have a session to the headend, it MUST send a PCErr message with Error-type=24 (PCE instantiation error) and Error-value=TBD5 (No PCEP session with the headend). PCE could try to initiate via another state-sync PCE if available.

4. Examples

The examples in this section are for illustrative purpose to show how the behavior of the state sync inter-PCE sessions.

4.1. Example 1



PCE1 computation priority 100
PCE2 computation priority 200

Consider the PCEP sessions as shown above, where computation priority is global for all the LSPs and link disjoint between LSPs PCC1->PCC2 and PCC3->PCC4 is required.

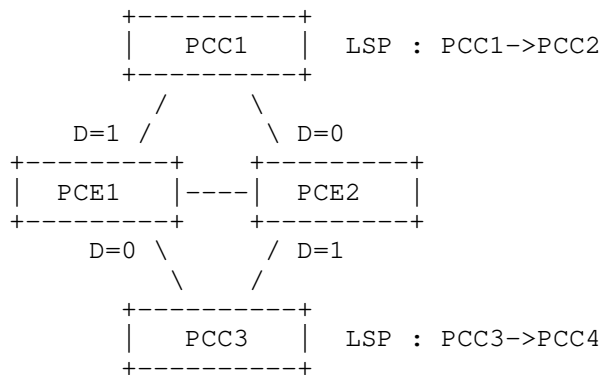
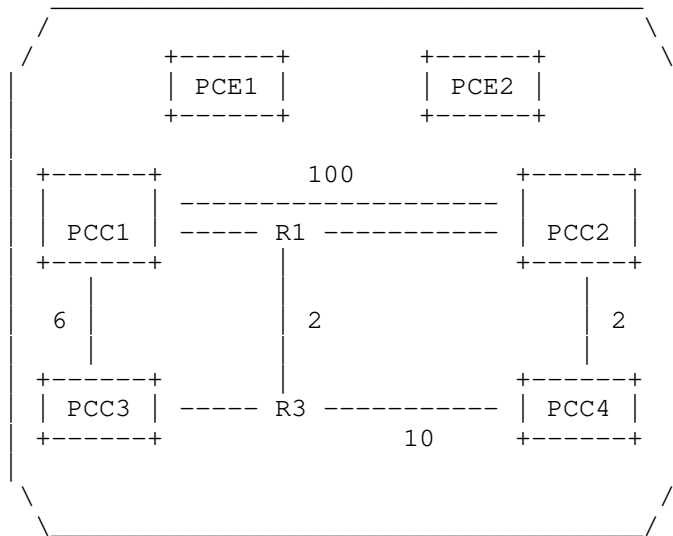
Consider the PCC1->PCC2 is configured first and PCC1 delegates the LSP to PCE1, but as PCE1 does not have the highest computation priority, it sub-delegates the LSP to PCE2 by sending a PCRpt with D=1 and including the SPEAKER-IDENTITY-TLV over the state-sync session. PCE2 receives the PCRpt and as it has delegation for this LSP, it computes the shortest path: R1->R3->R4->R2->PCC2. It then sends a PCUpd to PCE1 (including the SPEAKER-IDENTITY-TLV) with the computed ERO. PCE1 forwards the PCUpd to PCC1 (removing the SPEAKER-

IDENTITY-TLV). PCC1 acknowledges the PCUpd by a PCRpt to PCE1. PCE1 forwards the PCRpt to PCE2.

When PCC3->PCC4 is configured, PCC3 delegates the LSP to PCE2, PCE2 can compute a disjoint path as it has knowledge of both LSPs and has delegation also for both. The only solution found is to move PCC1->PCC2 LSP on another path, PCE2 can move PCC1->PCC2 as it has sub-delegation for it. It creates a new PCUpd with a new ERO: R1->R2-PCC2 towards PCE1 which forwards to PCC1. PCE2 sends a PCUpd to PCC3 with the path: R3->R4->PCC4.

In this set-up, PCEs are able to find a disjoint path while without state-sync and computation priority they could not.

4.2. Example 2

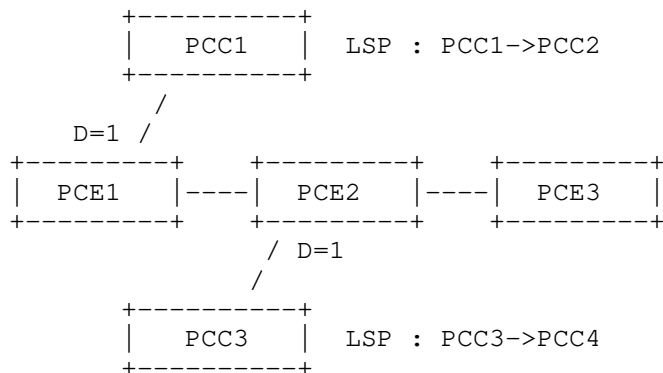
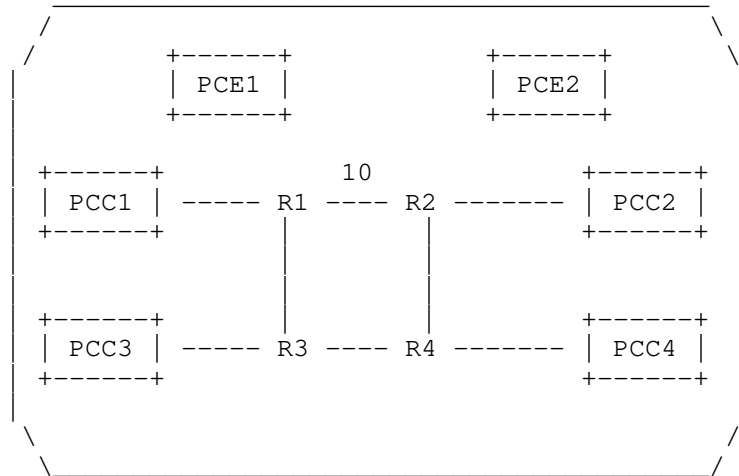


PCE1 computation priority 200

PCE2 computation priority 100

In this example, suppose both LSPs are configured almost at the same time. PCE1 sub-delegates PCC1->PCC2 to PCE2 while PCE2 keeps delegation for PCC3->PCC4, PCE2 computes a path for PCC1->PCC2 and PCC3->PCC4 and can achieve disjointness computation easily. No computation loop happens in this case.

4.3. Example 3



PCE1 computation priority 100
 PCE2 computation priority 200
 PCE3 computation priority 300

With the PCEP sessions as shown above, consider the need to have link disjoint LSPs PCC1->PCC2 and PCC3->PCC4.

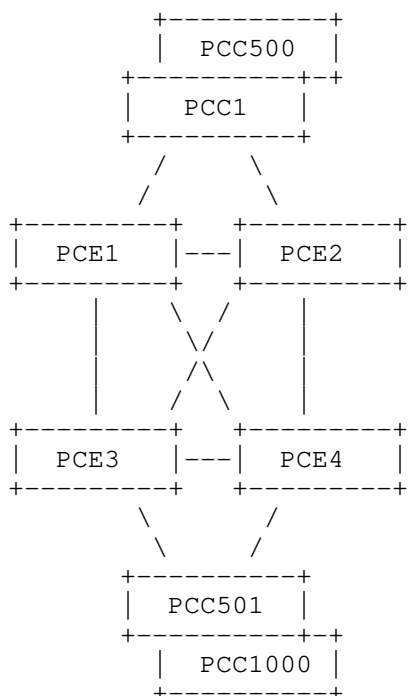
Suppose PCC1->PCC2 is configured first, PCC1 delegates the LSP to PCE1, but as PCE1 does not have the highest computation priority, it will sub-delegate the LSP to PCE2 (as it not aware of PCE3 and has no way to reach it). PCE2 cannot compute a path for PCC1->PCC2 as it does not have the highest priority and is not allowed to sub-delegate the LSP again towards PCE3 as per Section 3.

When PCC3->PCC4 is configured, PCC3 delegates the LSP to PCE2 that performs sub-delegation to PCE3. As PCE3 will have knowledge of only one LSP in the group, it cannot compute disjointness and can decide to fall-back to a less constrained computation to provide a path for PCC3->PCC4. In this case, it will send a PCUpd to PCE2 that will be forwarded to PCC3.

Disjointness cannot be achieved in this scenario because of lack of state-sync session between PCE1 and PCE3, but no computation loop happens. Thus it is advised for all PCEs that support state-sync to have a full mesh sessions between each other.

5. Using Primary/Secondary Computation and State-sync Sessions to increase Scaling

The Primary/Secondary computation and state-sync sessions architecture can be used to increase the scaling of the PCE architecture. If the number of PCCs is really high, it may be too resource consuming for a single PCE to maintain all the PCEP sessions while at the same time performing all path computations. Using primary/secondary computation and state-sync sessions may allow to create groups of PCEs that manage a subset of the PCCs and perform some or no path computations. Decoupling PCEP session maintenance and computation will allow increasing scaling of the PCE architecture.



In the figure above, two groups of PCEs are created: PCE1/2 maintain PCEP sessions with PCC1 up to PCC500, while PCE3/4 maintain PCEP sessions with PCC501 up to PCC1000. A granular primary/secondary policy is set-up as follows to load-share computation between PCEs:

- o PCE1 has priority 200 for association ID 1 up to 300, association source 0.0.0.0. All other PCEs have a decreasing priority for those associations.
- o PCE3 has priority 200 for association ID 301 up to 500, association source 0.0.0.0. All other PCEs have a decreasing priority for those associations.

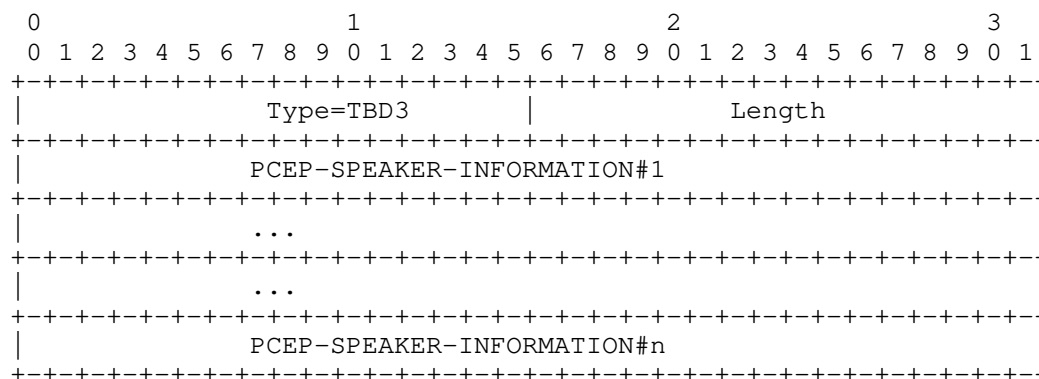
If some PCCs delegate LSPs with association ID 1 up to 300 and association source 0.0.0.0, the receiving PCE (if not PCE1) will sub-delegate the LSPs to PCE1. PCE1 becomes responsible for the computation of these LSP associations while PCE3 is responsible for the computation of another set of associations.

The procedures described in this document could help greatly in load-sharing between a group of stateful PCEs.

6. PCEP-PATH-VECTOR TLV

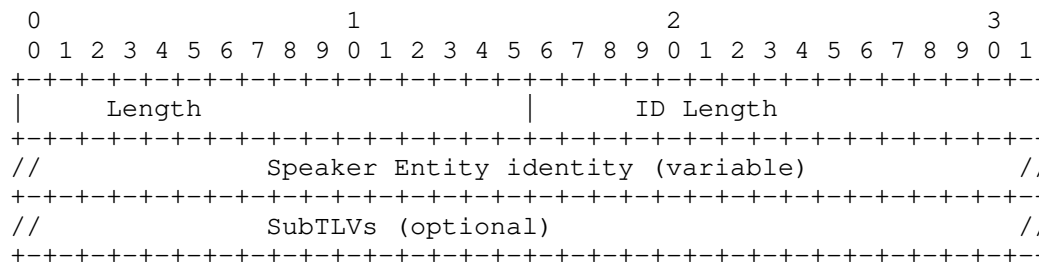
This document allows PCEP messages to be propagated among PCEP speaker. It may be useful to track information about the propagation of the messages. One of the use cases is a message loop detection mechanism, but other use cases like hop by hop information recording may also be implemented.

This document introduces the PCEP-PATH-VECTOR TLV (type TBD3) with the following format:



The TLV format and padding rules are as per [RFC5440].

The PCEP-SPEAKER-INFORMATION field has the following format:



Length: defines the total length of the PCEP-SPEAKER-INFORMATION field.

ID Length: defines the length of the Speaker identity actual field (non-padded).

Speaker Entity identity: same possible values as the SPEAKER-IDENTIFIER-TLV. Padded with trailing zeros to a 4-byte boundary.

The PCEP-SPEAKER-INFORMATION may also carry some optional subTLVs so each PCEP speaker can add local information that could be recorded. This document does not define any sub-TLV.

The PCEP-PATH-VECTOR TLV MAY be carried in the LSP Object. Its usage is purely optional.

The list of speakers within the PCEP-PATH-VECTOR TLV MUST be ordered. When sending a PCEP message (PCRpt, PCUpd, or PCInitiate), a PCEP Speaker MAY add the PCEP-PATH-VECTOR TLV with a PCEP-SPEAKER-INFORMATION containing its own information. If the PCEP message sent is the result of a previously received PCEP message, and if the PCEP-PATH-VECTOR TLV was already present in the initial message, the PCEP speaker MAY append a new PCEP-SPEAKER-INFORMATION containing its own information.

7. Security Considerations

The security considerations described in [RFC8231] and [RFC5440] apply to the extensions described in this document as well. Additional considerations related to state synchronization and sub-delegation between stateful PCEs are introduced, as it could be spoofed and could be used as an attack vector. An attacker could attempt to create too much state in an attempt to load the PCEP peer. The PCEP peer responds with a PCErr message as described in [RFC8231]. An attacker could impact LSP operations by creating bogus state. Further, state synchronization between stateful PCEs could provide an adversary with the opportunity to eavesdrop on the network. Thus, securing the PCEP session using Transport Layer Security (TLS) [RFC8253], as per the recommendations and best current practices in [RFC7525], is RECOMMENDED.

8. Acknowledgements

Thanks to [I-D.knodel-terminology] urging for better use of terms.

9. IANA Considerations

This document requests IANA actions to allocate code points for the protocol elements defined in this document.

9.1. PCEP-Error Object

IANA is requested to allocate a new Error Value for the Error Type 9.

Error-Type	Meaning	Reference
6	Mandatory Object Missing Error-value=TBD1: SPEAKER-IDENTITY-TLV missing	[RFC5440] This document
24	LSP instantiation error Error-value=TBD5: No PCEP session with the headend	[RFC8281] This document

9.2. PCEP TLV Type Indicators

IANA is requested to allocate new TLV Type Indicator values within the "PCEP TLV Type Indicators" sub-registry of the PCEP Numbers registry, as follows:

Value	Meaning	Reference
TBD2	ORIGINAL-LSP-DB-VERSION TLV	This document
TBD3	PCEP-PATH-VECTOR TLV	This document

9.3. STATEFUL-PCE-CAPABILITY TLV

IANA is requested to allocate a new bit value in the STATEFUL-PCE-CAPABILITY TLV Flag Field sub-registry.

Bit	Description	Reference
TBD4	INTER-PCE-CAPABILITY	This document

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [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>>.
- [RFC8232] Crabbe, E., Minei, I., Medved, J., Varga, R., Zhang, X., and D. Dhody, "Optimizations of Label Switched Path State Synchronization Procedures for a Stateful PCE", RFC 8232, DOI 10.17487/RFC8232, September 2017, <<https://www.rfc-editor.org/info/rfc8232>>.
- [RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <<https://www.rfc-editor.org/info/rfc8253>>.

10.2. Informative References

- [I-D.knodel-terminology] Knodel, M. and N. Oever, "Terminology, Power, and Inclusive Language in Internet-Drafts and RFCs", draft-knodel-terminology-04 (work in progress), August 2020.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <<https://www.rfc-editor.org/info/rfc6805>>.
- [RFC7399] Farrel, A. and D. King, "Unanswered Questions in the Path Computation Element Architecture", RFC 7399, DOI 10.17487/RFC7399, October 2014, <<https://www.rfc-editor.org/info/rfc7399>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.

- [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>>.
- [RFC8051] Zhang, X., Ed. and I. Minei, Ed., "Applicability of a Stateful Path Computation Element (PCE)", RFC 8051, DOI 10.17487/RFC8051, January 2017, <<https://www.rfc-editor.org/info/rfc8051>>.
- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.
- [RFC8751] Dhody, D., Lee, Y., Ceccarelli, D., Shin, J., and D. King, "Hierarchical Stateful Path Computation Element (PCE)", RFC 8751, DOI 10.17487/RFC8751, March 2020, <<https://www.rfc-editor.org/info/rfc8751>>.
- [RFC8800] Litkowski, S., Sivabalan, S., Barth, C., and M. Negi, "Path Computation Element Communication Protocol (PCEP) Extension for Label Switched Path (LSP) Diversity Constraint Signaling", RFC 8800, DOI 10.17487/RFC8800, July 2020, <<https://www.rfc-editor.org/info/rfc8800>>.

Appendix A. Contributors

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

Email: dhruv.ietf@gmail.com

Authors' Addresses

Stephane Litkowski
Cisco

Email: slitkows.ietf@gmail.com

Siva Sivabalan
Ciena Corporation

Email: msiva282@gmail.com

Cheng Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: c.l@huawei.com

Haomian Zheng
Huawei Technologies
H1, Huawei Xiliu Beipo Village, Songshan Lake
Dongguan, Guangdong 523808
China

Email: zhenghaomian@huawei.com

PCE WG
Internet-Draft
Intended status: Standards Track
Expires: January 8, 2020

Quan Xiong
Fangwei Hu
Greg Mirsky
ZTE Corporation
Weiqiang Cheng
China Mobile
July 7, 2019

Stateful PCE for SR-MPLS Inter-domain
draft-xiong-pce-stateful-pce-sr-inter-domain-01

Abstract

This document proposes two solutions to perform the Segment Routing with MPLS data plane (SR-MPLS) inter-domain path computation and initiation with stateful PCEs and the use of Path Segment.

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 <https://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 January 8, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
1.2. Terminology	3
2. The SR-MPLS Inter-domain with PCE	3
2.1. The Stitching LSP Association Solution	5
2.2. The Stitching Label Solution	6
3. Inter-domain Path Segment Allocation	6
3.1. PCC Allocated	6
3.2. PCE Allocated	7
4. PCEP Procedure	7
4.1. HPCE-initiated LSP	7
4.2. PCC-initiated LSP	8
5. Security Considerations	8
6. IANA Considerations	9
7. Acknowledgements	9
8. References	9
8.1. Informative References	9
8.2. Normative References	9
Authors' Addresses	11

1. Introduction

The Path Computation Element (PCE) architecture is defined in [RFC4655] for MPLS Traffic Engineering (MPLS-TE) and Generalized MPLS (GMPLS) networks. The Path Computation Element Communication Protocol (PCEP) defined in [RFC5440] provides mechanisms for PCEs to perform path computations in response to Path Computation Clients (PCCs) requests.

[I-D.ietf-pce-segment-routing] proposes extensions to PCEP that allow a stateful PCE to compute TE paths in segment routing (SR) networks. As defined in [I-D.ietf-spring-mpls-path-segment], a path segment is used to identify a SR path and support bidirectional SR paths correlation. [I-D.li-pce-sr-path-segment] proposed the extension for PCEP to operate with Path Segment. [I-D.li-pce-sr-bidir-path] proposed the extension for PCEP to group two unidirectional SR Paths into an Associated Bidirectional SR Path.

[I-D.xiong-spring-path-segment-sr-inter-domain] proposes the use of Path Segment in inter-domain scenarios for SR-MPLS network. It is required to perform the SR inter-domain path computation and initiation with PCE deployment.

The path computation requirements for Label Switched Paths (LSPs) across multiple domains are discussed in [RFC4105] and [RFC4216]. Inter-domain path computation can be performed by a single stateful PCE and multiple stateful PCEs. The PCE may have no ability to collect the topologies all over the domains. So the single PCE model is not applied in deployment. Three multiple PCEs models can be used to perform PCE-based inter-domain path computation including Per-Domain Path Computation [RFC5152], Backward-Recursive PCE-Based Computation (BRPC) [RFC5441] and Hierarchical PCE (H-PCE) [RFC6805]. Computing the optimum inter-domain path requires co-operation between multiple PCEs. But the sequence of domains need to be known before the path computation in BRPC mechanism. Stateful H-PCE architecture is appropriate to compute an optimal end-to-end path across multiple domains.

As defined in [I-D.xiong-spring-path-segment-sr-inter-domain], the SR-MPLS inter-domain models includes stitching and nesting inter-domain models between inter-Area or inter-AS domains. This document proposes two solutions to perform the SR-MPLS inter-domain path computation and initiation with stateful PCEs and the use of Path Segment.

1.1. 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 [RFC2119].

1.2. Terminology

The terminology is defined as [RFC5440], [I-D.ietf-pce-segment-routing], [I-D.ietf-spring-mpls-path-segment].

2. The SR-MPLS Inter-domain with PCE

The SR-MPLS inter-domain scenario is described in [I-D.xiong-spring-path-segment-sr-inter-domain]. The domains of the networks may be IGP Areas or ASes and the inter-domain scenario may be inter-Area or inter-AS. The multiple SR-MPLS domains may be interconnect with a ABR within areas or inter-link between ASes. As the Figure 1 shown, SR-AS1, SR-AS2 and SR-AS3 interconnect with logical links and SR-Area1, SR-Area2 and SR-Area3 interconnect within border nodes. The SR end-to-end bidirectional LSP needs to be provided along the multi-domain paths. The Path 1~5 are forwarding path segments and Path 1'~5' are the related reverse path segments and these are all inter-domain path segments.

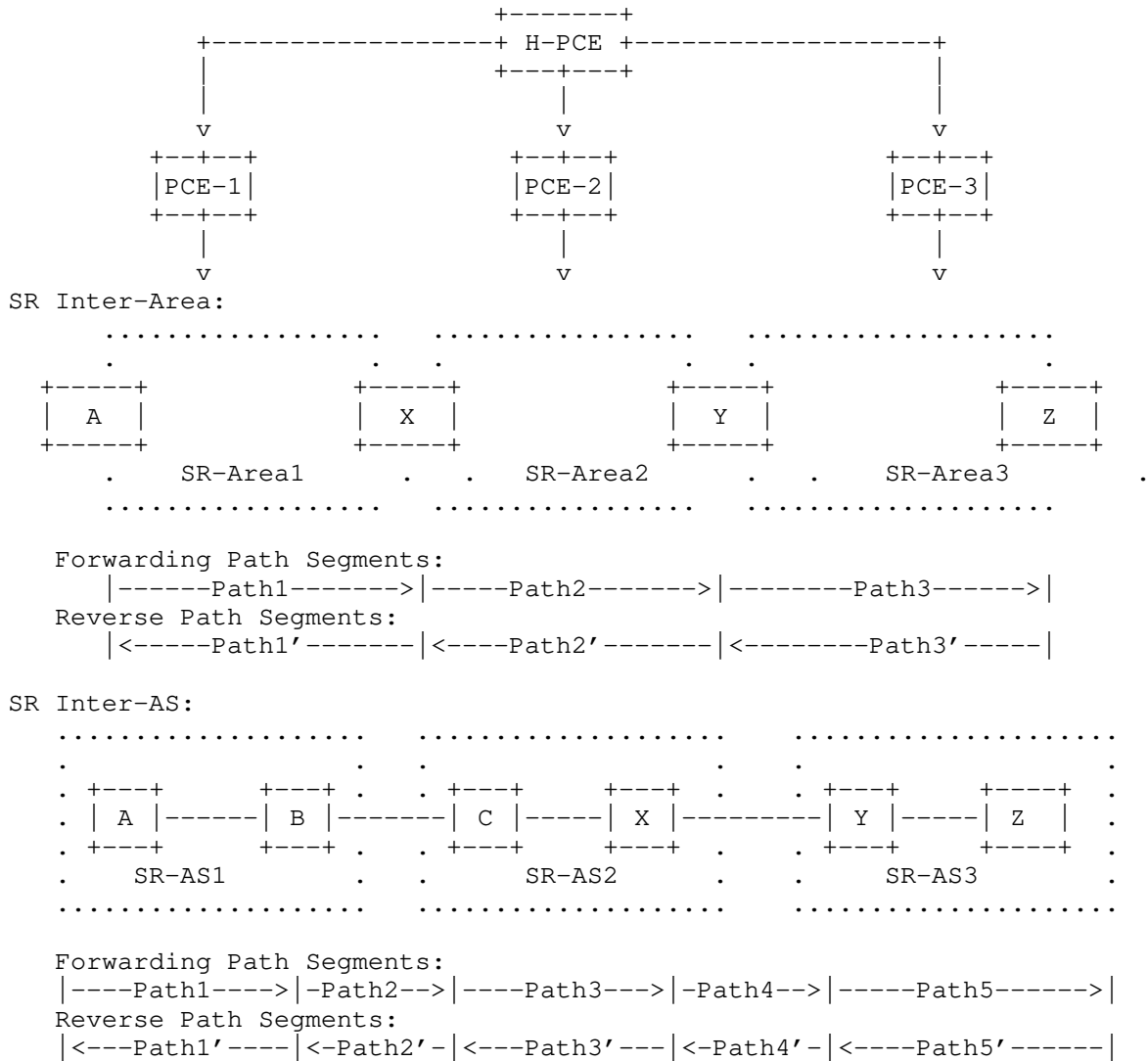


Figure 1 The SR Inter-Domain with H-PCE

The hierarchical PCE architecture is described in [RFC6805], a parent PCE maintains a domain topology map that contains the child domains (seen as vertices in the topology) and their interconnections (links in the topology) but no information about the content of the child domains. Each child domain has one PCE taking in charge of computing paths across its own domain. These PCEs are known as child PCEs and have a relationship with the parent PCE. As the Figure 1 shown,

H-PCE is parent PCE and PCE-1, PCE-2 and PCE-3 are child PCEs which is responsible for each own SR-AS.

When an optimal inter-domain path is required, the ingress PCE sends a request to the parent PCE or the stateful parent PCE itself to initiate the path computation. The parent PCE selects a set of candidate domain paths based on the domain topology and the state of the inter-domain links. It then sends computation requests to the child PCEs responsible for each of the domains on the candidate domain paths. The stateful child PCE in each domain performs active stateful procedure as defined [RFC8231].

2.1. The Stitching LSP Association Solution

The LSPs of multiple domains can be stitched together by adding them to a stitching LSP association group as defined in [I-D.hu-pce-stitching-lsp-association]. As the Figure 2 shown, the stateful H-PCE sends the PCInit message defined in [RFC8281] to initiate the inter-domain path computation adding the forwarding LSP 1~3 to Assoc#1 and reverse LSP 1'~3' to Assoc#2. The child PCEs may initiate the intra-domain LSPs when receiving the message from parent PCE.

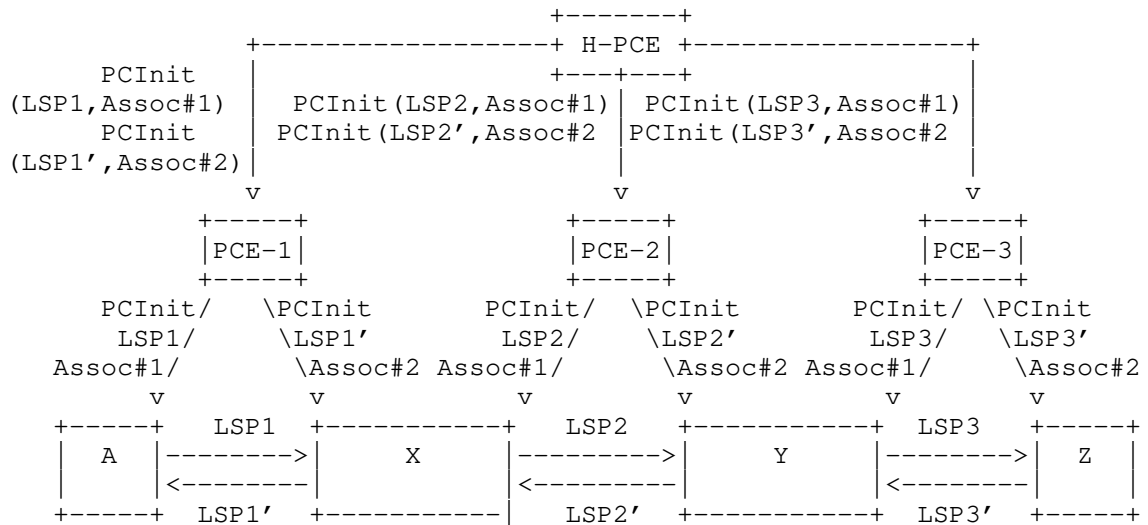
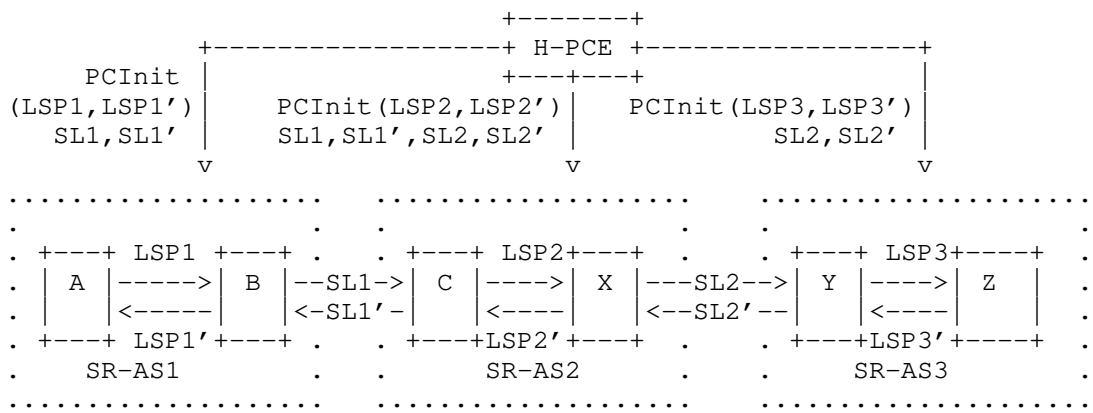


Figure 2 The SR inter-domain Stitching LSP Association

2.2. The Stitching Label Solution

The Path Segment can be used for path stitching. The SR sub-paths can be correlated with the use of Path Segment. This section defined the path segments as stitching Labels which used to stitch per-domain LSP tunnels in order to create end-to-end path that cross multiple domains.

SR intra-domain path is setup as part of inter-domain SR path. When PCC requests the PCE or the PCE itself to initiate The SR path, the inter-domain path segments should be carried as a stitching Label with the associated link.



SL:Stiching Label

Figure 3 The SR Inter-Domain Stitching Label

3. Inter-domain Path Segment Allocation

The inter-domain path segment may be allocated by PCC or PCE. The PCE may be the single domain PCE which taking in charge of the respective domain. The inter-domain path segments is a unique value in the domain which PCC or PCE belongs to. The operation of path segment request and reply may be the same with that in single domain as defined in [I-D.li-pce-sr-path-segment].

3.1. PCC Allocated

As defined in [I-D.xiong-spring-path-segment-sr-inter-domain], an inter-domain path segment can be allocated by egress PCC and may be maintained on the PCC itself. The inter-domain path segment connects

two domains and the ingress and egress PCC are belong to different domains. The ingress and egress PCC need to exchange messages which carrying path segment information between the two PCEs.

The Ingress PCC may request to allocate a path segment from egress PCC. Once egress PCC allocated the inter-domain path segment, it need to inform the PCE in respective domain with the PCRpt message. The PCE need to communicate with the PCE which the ingress PCC belongs to inform the value allocated.

3.2. PCE Allocated

The ingress PCC may request the inter-domain path segment to be allocated by the PCE in PCC-Initiated LSP. The PCE may allocate the inter-domain path segment on its own domain in PCEs-Initiated LSP. The allocated path segment needs to be informed to the ingress and egress PCC.

The inter-domain path segments may be allocated separately by the PCEs which control the ingress and egress PCC along with the LSP initiation.

4. PCEP Procedure

[RFC8281] describes setup, maintenance and teardown of PCE-initiated LSPs under the stateful PCE model, without the need for local configuration on the PCC. Similar to LSP updation, the inter-domain LSP can be initiated by the ingress PCE using the PCInitiate message to the ingress LSR. The inter-domain path segment is viewed as stitching label. Per-domain LSP may also be initiated by respective domain's PCE and stitched together.

4.1. HPCE-initiated LSP

In H-PCE [RFC6805] architecture, the parent PCE is used to compute a multi-domain path based on the domain connectivity information. The stateful H-PCE in active model can be used to initiate the inter-domain bidirectional path for SR networks. PCE sends PCInitiate message to its domain SR nodes with ERO={SID LIST} and carrying stitching association group TLV and path segments. If the SR nodes is the border nodes of the SR domain, it correlates the two path segments and the related SID list if the related association ID is the same value.

The PCEP procedure for the HPCE-initiated LSP is following:

The stateful H-PCE initiates the end-to-end path computation across multiple domains and selects a set of candidate domain paths based on the topology.

The stateful H-PCE sends PCInitiate message to every PCEs which the end-to-end path traversed, carrying inter-domain path segments allocated by H-PCE, stitching LSP association group and the SID list in the ERO object.

The stateful child PCE in each domain perform active stateful procedure as defined in [I-D.li-pce-sr-path-segment].

4.2. PCC-initiated LSP

In case of passive path computation request to the ingress PCE from the ingress LSR, the H-PCE path computation procedure is applied to compute sequence of domains or end-to-end path by using PCReq and PCRep messages among stateful PCEs in passive mode.

In case of delegation to the ingress PCE (active stateful PCE), the ingress child PCE may further delegate to parent PCE as per [I-D.ietf-pce-stateful-hpce]. The parent PCE could update the path of the inter-domain LSP.

The ingress nodes of the source AS sends the PCReq message to its PCE, then the PCE sends PCReq message to the H-PCE or stateful PCEs in other domains. The PECP procedure for the PCC-initiated LSP in H-PCE model is as follow.

The ingress PCC from the ingress domain sends a PCReq request to the PCE which is responsible for the domain containing the destination information.

The ingress PCE sends the path computation request direct to the parent PCE.

The parent PCE computes the optimal end-to-end path and initiates the inter-domain paths to the child PCEs which the path traversed.

Each PCE sends PCInitiate message to ingress or egress nodes of its domain to initiate the LSPs.

5. Security Considerations

TBD.

6. IANA Considerations

TBD.

7. Acknowledgements

TBD.

8. References

8.1. Informative References

- [RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <<https://www.rfc-editor.org/info/rfc6805>>.

8.2. Normative References

- [I-D.hu-pce-stitching-lsp-association]
hu, f., Xiong, Q., Mirsky, G., and W. Cheng, "Stitching LSP Association", draft-hu-pce-stitching-lsp-association-00 (work in progress), December 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-16 (work in progress), March 2019.
- [I-D.ietf-pce-stateful-hpce]
Dhody, D., Lee, Y., Ceccarelli, D., Shin, J., and D. King, "Hierarchical Stateful Path Computation Element (PCE).", draft-ietf-pce-stateful-hpce-11 (work in progress), July 2019.
- [I-D.ietf-spring-mpls-path-segment]
Cheng, W., Li, H., Chen, M., Gandhi, R., and R. Zigler, "Path Segment in MPLS Based Segment Routing Network", draft-ietf-spring-mpls-path-segment-00 (work in progress), March 2019.
- [I-D.li-pce-sr-bidir-path]
Li, C., Chen, M., Cheng, W., Li, Z., Dong, J., Gandhi, R., and Q. Xiong, "PCEP Extensions for Associated Bidirectional Segment Routing (SR) Paths", draft-li-pce-sr-bidir-path-05 (work in progress), March 2019.

- [I-D.li-pce-sr-path-segment]
Li, C., Chen, M., Cheng, W., Dong, J., Li, Z., Gandhi, R.,
and Q. Xiong, "Path Computation Element Communication
Protocol (PCEP) Extension for Path Segment in Segment
Routing (SR)", draft-li-pce-sr-path-segment-05 (work in
progress), March 2019.
- [I-D.xiong-spring-path-segment-sr-inter-domain]
Xiong, Q., Mirsky, G., and W. Cheng, "The Use of Path
Segment in SR Inter-domain Scenarios", draft-xiong-spring-
path-segment-sr-inter-domain-00 (work in progress), July
2019.
- [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>>.
- [RFC4105] Le Roux, J., Ed., Vasseur, J., Ed., and J. Boyle, Ed.,
"Requirements for Inter-Area MPLS Traffic Engineering",
RFC 4105, DOI 10.17487/RFC4105, June 2005,
<<https://www.rfc-editor.org/info/rfc4105>>.
- [RFC4216] Zhang, R., Ed. and J. Vasseur, Ed., "MPLS Inter-Autonomous
System (AS) Traffic Engineering (TE) Requirements",
RFC 4216, DOI 10.17487/RFC4216, November 2005,
<<https://www.rfc-editor.org/info/rfc4216>>.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation
Element (PCE)-Based Architecture", RFC 4655,
DOI 10.17487/RFC4655, August 2006,
<<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC5152] Vasseur, JP., Ed., Ayyangar, A., Ed., and R. Zhang, "A
Per-Domain Path Computation Method for Establishing Inter-
Domain Traffic Engineering (TE) Label Switched Paths
(LSPs)", RFC 5152, DOI 10.17487/RFC5152, February 2008,
<<https://www.rfc-editor.org/info/rfc5152>>.
- [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>>.

- [RFC5441] Vasseur, JP., Ed., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, DOI 10.17487/RFC5441, April 2009, <<https://www.rfc-editor.org/info/rfc5441>>.
- [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>>.
- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.

Authors' Addresses

Quan Xiong
ZTE Corporation
No.6 Huashi Park Rd
Wuhan, Hubei 430223
China

Phone: +86 27 83531060
Email: xiong.quan@zte.com.cn

Fangwei Hu
ZTE Corporation
No.889 Bibo Rd
Shanghai 201203
China

Phone: +86 21 68896273
Email: hu.fangwei@zte.com.cn

Greg Mirsky
ZTE Corporation
USA

Email: gregimirsky@gmail.com

Weiqiang Cheng
China Mobile
Beijing
China

Email: chengweiqiang@chinamobile.com

PCE WG
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2020

Quan Xiong
Greg Mirsky
ZTE Corporation
Fangwei Hu
Individual
Weiqiang Cheng
China Mobile
October 22, 2019

Stateful PCE for SR-MPLS Inter-domain
draft-xiong-pce-stateful-pce-sr-inter-domain-02

Abstract

This document proposes a solution to perform the Segment Routing with MPLS data plane (SR-MPLS) inter-domain path computation and initiation with stateful PCEs and the use of Path Segments.

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 <https://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, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
1.2. Terminology	3
2. The SR-MPLS Inter-domain with Path Segments	3
3. Inter-domain Path Segment Allocation	6
3.1. PCC Allocated	6
3.2. PCE Allocated	7
4. PCEP Procedure	7
4.1. HPCE-initiated LSP	7
4.2. PCC-initiated LSP	8
5. Security Considerations	8
6. IANA Considerations	8
7. Acknowledgements	9
8. References	9
8.1. Informative References	9
8.2. Normative References	9
Authors' Addresses	11

1. Introduction

The Path Computation Element (PCE) architecture is defined in [RFC4655] for MPLS Traffic Engineering (MPLS-TE) and Generalized MPLS (GMPLS) networks. The Path Computation Element Communication Protocol (PCEP) defined in [RFC5440] provides mechanisms for PCEs to perform path computations in response to Path Computation Clients (PCCs) requests.

[I-D.ietf-pce-segment-routing] proposes extensions to PCEP that allow a stateful PCE to compute TE paths in segment routing (SR) networks. As defined in [I-D.ietf-spring-mpls-path-segment], a path segment is used to identify a SR path and support bidirectional SR paths correlation. [I-D.ietf-pce-sr-path-segment] proposed the extension for PCEP to operate with Path Segment. [I-D.li-pce-sr-bidir-path] proposed the extension for PCEP to group two unidirectional SR Paths into an Associated Bidirectional SR Path.

[I-D.xiong-spring-path-segment-sr-inter-domain] proposes the use of Path Segment in inter-domain scenarios for SR-MPLS network. It is required to perform the SR inter-domain path computation and initiation with PCE deployment.

The path computation requirements for Label Switched Paths (LSPs) across multiple domains are discussed in [RFC4105] and [RFC4216]. Inter-domain path computation can be performed by a single stateful

PCE and multiple stateful PCEs. The PCE may have no ability to collect the topologies all over the domains. So the single PCE model is not applied in deployment. Three multiple PCEs models can be used to perform PCE-based inter-domain path computation including Per-Domain Path Computation [RFC5152], Backward-Recursive PCE-Based Computation (BRPC) [RFC5441] and Hierarchical PCE (H-PCE) [RFC6805]. Computing the optimum inter-domain path requires co-operation between multiple PCEs. But the sequence of domains need to be known before the path computation in BRPC mechanism. Stateful H-PCE architecture is appropriate to compute an optimal end-to-end path across multiple domains.

As defined in [I-D.xiong-spring-path-segment-sr-inter-domain], the SR-MPLS inter-domain models includes stitching and nesting inter-domain models between inter-Area or inter-AS domains. This document proposes a solution to perform the SR-MPLS inter-domain path computation and initiation with stateful PCEs and the use of Path Segments.

1.1. 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 [RFC2119].

1.2. Terminology

The terminology is defined as [RFC5440], [I-D.ietf-pce-segment-routing], [I-D.ietf-spring-mpls-path-segment].

2. The SR-MPLS Inter-domain with Path Segments

The SR-MPLS inter-domain scenario is described in [I-D.xiong-spring-path-segment-sr-inter-domain]. The domains of the networks may be IGP Areas or ASes and the inter-domain scenario may be inter-Area or inter-AS. The multiple SR-MPLS domains may be interconnect with a ABR within areas or inter-link between ASes. The hierarchical PCE architecture is described in [RFC6805], a parent PCE maintains a domain topology map that contains the child domains (seen as vertices in the topology) and their interconnections (links in the topology) but no information about the content of the child domains. Each child domain has one PCE taking in charge of computing paths across its own domain. These PCEs are known as child PCEs and have a relationship with the parent PCE.

As the Figure 1 shown, H-PCE is parent PCE and PCE-1, PCE-2 and PCE-3 are child PCEs which is responsible for each own domain. SR-AS1, SR-AS2 and SR-AS3 interconnect with logical links and SR-Area1, SR-Area2

and SR-Area3 interconnect within border nodes. The SR end-to-end bidirectional LSP needs to be provided along the multi-domain paths. The Path 1~5 are forward path segments and Path 1'~5' are the related reverse path segments and these are all inter-domain path segments.

When an optimal inter-domain path is required, the ingress PCE sends a request to the parent PCE or the stateful parent PCE itself to initiate the path computation. The parent PCE selects a set of candidate domain paths based on the domain topology and the state of the inter-domain links. It then sends computation requests to the child PCEs responsible for each of the domains on the candidate domain paths. The stateful child PCE in each domain performs active stateful procedure as defined [RFC8231].

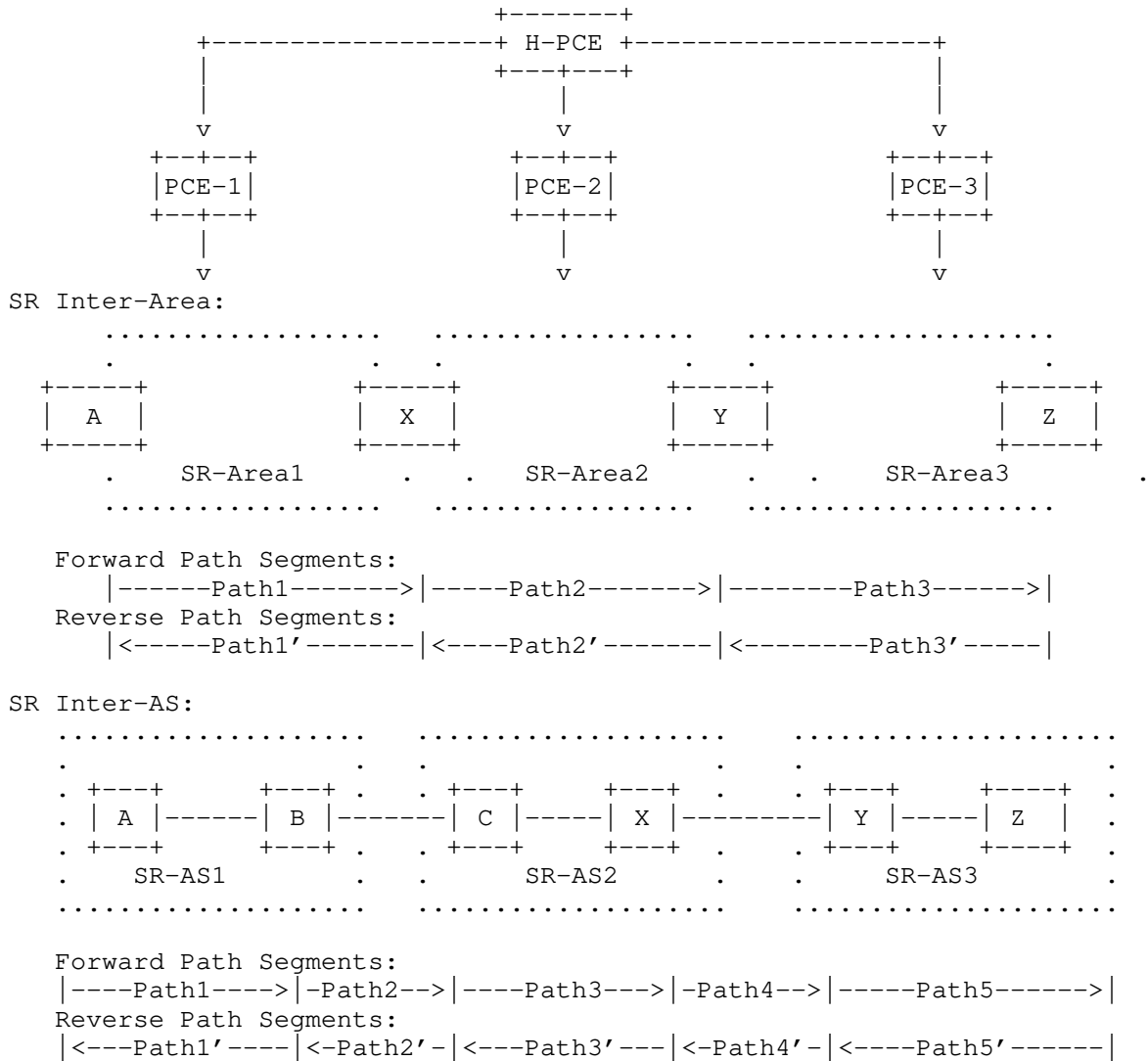


Figure 1 The SR Inter-Domain with H-PCE

The LSPs of multiple domains can be stitched together by adding them to a stitching LSP association group as defined in [I-D.hu-pce-stitching-lsp-association]. As the Figure 2 shown, the stateful H-PCE sends the PCInit message defined in [RFC8281] to initiate the inter-domain path computation adding the forward LSP 1~3 to Assoc#1 and reverse LSP 1'~3' to Assoc#2. The child PCEs may initiate the intra-domain LSPs when receiving the message from parent

PCE. The LSP 1~3 could be stitched to a forward end-to-end LSP and the LSP 1'~3' could be stitched to a reverse end-to-end LSP. Furthermore, the two unidirectional end-to-end LSPs MAY be bound to a bidirectional end-to-end LSP as deccribed in [I-D.li-pce-sr-bidir-path].

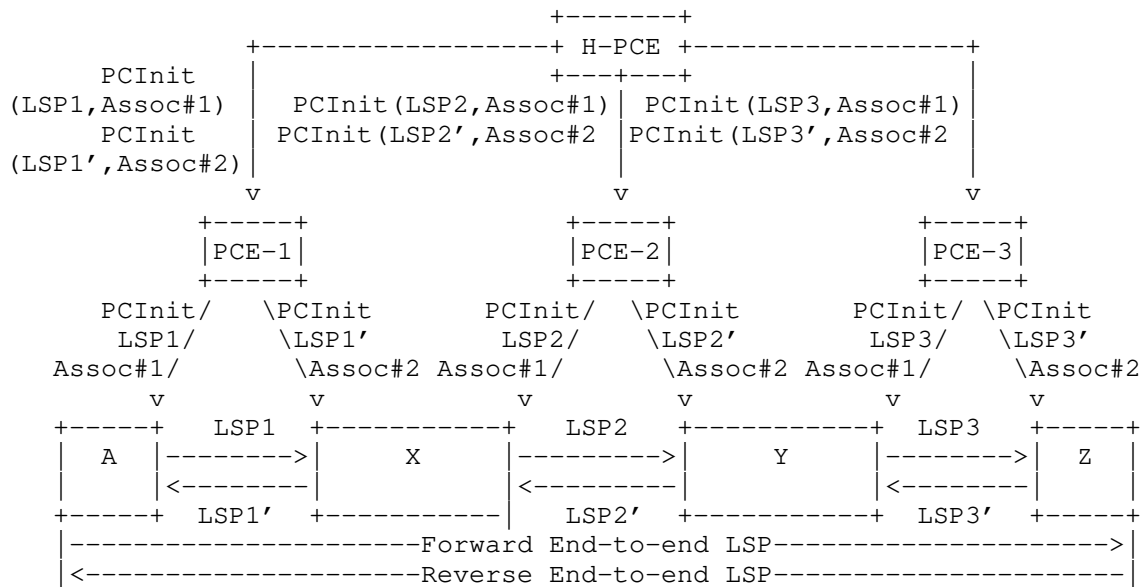


Figure 2 The SR inter-domain Stitching LSP Association

3. Inter-domain Path Segment Allocation

The inter-domain path segment may be allocated by PCC or PCE. The PCE may be the single domain PCE which taking in charge of the respective domain. The inter-domain path segments is a unique value in the domain which PCC or PCE belongs to. The operation of path segment request and reply may be the same with that in single domain as defined in [I-D.ietf-pce-sr-path-segment].

3.1. PCC Allocated

As defined in [I-D.xiong-spring-path-segment-sr-inter-domain], an inter-domain path segment can be allocated by egress PCC and may be maintained on the PCC itself. The inter-domain path segment connects two domains and the ingress and egress PCC are belong to different domains. The ingress and egress PCC need to exchange messages which carrying path segment information between the two PCEs.

The Ingress PCC may request to allocate a path segment from egress PCC. Once egress PCC allocated the inter-domain path segment, it need to inform the PCE in respective domain with the PCRpt message. The PCE need to communicate with the PCE which the ingress PCC belongs to inform the value allocated.

3.2. PCE Allocated

The ingress PCC may request the inter-domain path segment to be allocated by the PCE in PCC-Initiated LSP. The PCE may allocate the inter-domain path segment on its own domain in PCEs-Initiated LSP. The allocated path segment needs to be informed to the ingress and egress PCC.

The inter-domain path segments may be allocated separately by the PCEs which control the ingress and egress PCC along with the LSP initiation.

4. PCEP Procedure

[RFC8281] describes setup, maintenance and teardown of PCE-initiated LSPs under the stateful PCE model, without the need for local configuration on the PCC. Similar to LSP updation, the inter-domain LSP can be initiated by the ingress PCE using the PCInitiate message to the ingress LSR. Per-domain LSP may also be initiated by respective domain's PCE and stitched together.

4.1. HPCE-initiated LSP

In H-PCE [RFC6805] architecture, the parent PCE is used to compute a multi-domain path based on the domain connectivity information. The stateful H-PCE in active model can be used to initiate the inter-domain bidirectional path for SR networks. PCE sends PCInitiate message to its domain SR nodes with ERO={SID LIST} and carrying stitching association group TLV and path segments. If the SR nodes is the border nodes of the SR domain, it correlates the two path segments and the related SID list if the related association ID is the same value.

The PCEP procedure for the HPCE-initiated LSP is following:

The stateful H-PCE initiates the end-to-end path computation across multiple domains and selects a set of candidate domain paths based on the topology.

The stateful H-PCE sends PCInitiate message to every PCEs which the end-to-end path traversed, carrying inter-domain path segments

allocated by H-PCE, stitching LSP association group and the SID list in the ERO object.

The stateful child PCE in each domain perform active stateful procedure as defined in [I-D.ietf-pce-sr-path-segment].

4.2. PCC-initiated LSP

In case of passive path computation request to the ingress PCE from the ingress LSR, the H-PCE path computation procedure is applied to compute sequence of domains or end-to-end path by using PCReq and PCRep messages among stateful PCEs in passive mode.

In case of delegation to the ingress PCE (active stateful PCE), the ingress child PCE may further delegate to parent PCE as per [I-D.ietf-pce-stateful-hpce]. The parent PCE could update the path of the inter-domain LSP.

The ingress nodes of the source AS sends the PCReq message to its PCE, then the PCE sends PCReq message to the H-PCE or stateful PCEs in other domains. The PECP procedure for the PCC-initiated LSP in H-PCE model is as follow.

The ingress PCC from the ingress domain sends a PCReq request to the PCE which is responsible for the domain containing the destination information.

The ingress PCE sends the path computation request direct to the parent PCE.

The parent PCE computes the optimal end-to-end path and initiates the inter-domain paths to the child PCEs which the path traversed.

Each PCE sends PCInitiate message to ingress or egress nodes of its domain to initiate the LSPs.

5. Security Considerations

TBD.

6. IANA Considerations

TBD.

7. Acknowledgements

TBD.

8. References

8.1. Informative References

- [RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <<https://www.rfc-editor.org/info/rfc6805>>.

8.2. Normative References

- [I-D.hu-pce-stitching-lsp-association]
Xiong, Q., Mirsky, G., hu, f., and W. Cheng, "Stitching LSP Association", draft-hu-pce-stitching-lsp-association-01 (work in progress), July 2019.
- [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-16 (work in progress), March 2019.
- [I-D.ietf-pce-sr-path-segment]
Li, C., Chen, M., Cheng, W., Gandhi, R., and Q. Xiong, "Path Computation Element Communication Protocol (PCEP) Extension for Path Segment in Segment Routing (SR)", draft-ietf-pce-sr-path-segment-00 (work in progress), October 2019.
- [I-D.ietf-pce-stateful-hpce]
Dhody, D., Lee, Y., Ceccarelli, D., Shin, J., and D. King, "Hierarchical Stateful Path Computation Element (PCE)", draft-ietf-pce-stateful-hpce-15 (work in progress), October 2019.
- [I-D.ietf-spring-mpls-path-segment]
Cheng, W., Li, H., Chen, M., Gandhi, R., and R. Zigler, "Path Segment in MPLS Based Segment Routing Network", draft-ietf-spring-mpls-path-segment-01 (work in progress), September 2019.

- [I-D.li-pce-sr-bidir-path]
Li, C., Chen, M., Cheng, W., Li, Z., Dong, J., Gandhi, R.,
and Q. Xiong, "PCEP Extensions for Associated
Bidirectional Segment Routing (SR) Paths", draft-li-pce-
sr-bidir-path-06 (work in progress), August 2019.
- [I-D.xiong-spring-path-segment-sr-inter-domain]
Xiong, Q., Mirsky, G., and W. Cheng, "The Use of Path
Segment in SR Inter-domain Scenarios", draft-xiong-spring-
path-segment-sr-inter-domain-00 (work in progress), July
2019.
- [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>>.
- [RFC4105] Le Roux, J., Ed., Vasseur, J., Ed., and J. Boyle, Ed.,
"Requirements for Inter-Area MPLS Traffic Engineering",
RFC 4105, DOI 10.17487/RFC4105, June 2005,
<<https://www.rfc-editor.org/info/rfc4105>>.
- [RFC4216] Zhang, R., Ed. and J. Vasseur, Ed., "MPLS Inter-Autonomous
System (AS) Traffic Engineering (TE) Requirements",
RFC 4216, DOI 10.17487/RFC4216, November 2005,
<<https://www.rfc-editor.org/info/rfc4216>>.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation
Element (PCE)-Based Architecture", RFC 4655,
DOI 10.17487/RFC4655, August 2006,
<<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC5152] Vasseur, JP., Ed., Ayyangar, A., Ed., and R. Zhang, "A
Per-Domain Path Computation Method for Establishing Inter-
Domain Traffic Engineering (TE) Label Switched Paths
(LSPs)", RFC 5152, DOI 10.17487/RFC5152, February 2008,
<<https://www.rfc-editor.org/info/rfc5152>>.
- [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>>.

- [RFC5441] Vasseur, JP., Ed., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, DOI 10.17487/RFC5441, April 2009, <<https://www.rfc-editor.org/info/rfc5441>>.
- [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>>.
- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.

Authors' Addresses

Quan Xiong
ZTE Corporation
No.6 Huashi Park Rd
Wuhan, Hubei 430223
China

Phone: +86 27 83531060
Email: xiong.quan@zte.com.cn

Greg Mirsky
ZTE Corporation
USA

Email: gregimirsky@gmail.com

Fangwei Hu
Individual
China

Email: hufwei@gmail.com

Weiqiang Cheng
China Mobile
Beijing
China

Email: chengweiqiang@chinamobile.com